Enabling Delay-Tolerant Communications for Partially Connected Vehicular Ad Hoc Networks

Kadri Kaan Sevimli [a], *Mujdat Soyturk* [b,*]

[a] Naval Science and Engineering Institute, Tuzla, Istanbul, 93942 – Turkey

[b] *Department of Computer Engineering, Istanbul Technical University, Maslak, Istanbul 34469, Turkey*

Abstract

The packet delivery ratio and throughput in Vehicular Ad Hoc Networks (VANETs) depend on the network connectivity and degrade as the connectivity decreases. A delay-tolerant approach increases the packet delivery ratio in partially connected networks. In the previous studies, the packet lifetime and the vehicle's packet carry time are determined to maximize the packet delivery ratio in partially connected VANETs. However, IPv6 uses the hop count in the Hop Limit field instead of using seconds in the TTL field, as in IPv4. Similarly, although IPv4 packets are designed to carry the packet lifetime in seconds in the TTL field, this field is used for the hop count in practice. In this case, there is no mechanism to determine how long vehicles will carry packets in VANETs when delivering packets in a delay-tolerant fashion. In this study, we propose approaches for the delivery of IPv6 packets in a delay-tolerant fashion in partially connected networks. We also propose another method that enhances these proposed approaches and optimizes the packet delivery ratio. The proposed approaches are simulated to observe performance results. Our analysis shows that the proposed approaches can be easily adopted for IPv6/IPv4 packets to be delivered in a delay-tolerant fashion. Additionally, we observe that our approaches increase the packet delivery ratio in partially connected Vehicular Ad Hoc Networks and can also be used in networks where the nodes are mobile or connections get broken frequently.

Keywords: Delay-Tolerant Communications, IPv6, TTL, Carry Time, Packet Delivery Ratio, Lifetime, Partial Connectivity

[*] **Corresponding author**. Mujdat Soyturk

E-mail addresses: msoyturk@itu.edu.tr, ksevimli@dho.edu.tr

Please note that this version of the paper is the post-peer review, accepted paper submitted for final publication. The printed and reformatted version is accessible at the original source of the publication (http://www.inderscience.com/info/inarticle.php?artid=50264), where cited as "*Kadri Kaan Sevimli, Mujdat Soyturk. Enabling Delay-Tolerant Communications for Partially Connected Vehicular Ad Hoc Networks, International Journal of Ad Hoc and Ubiquitous Computing, Vol.11, No.2/3, pp.157-168, 2012 (2012)*".

1 Introduction

Vehicular Ad Hoc Networks (VANETs) is a good example to the intermittently connected networks where the connections change/break frequently. VANETs have unique features that are very specific and not shared with other networks. Network performance is primarily affected by node densities and vehicle speeds [1]. Node density in VANETs may change very frequently in time and space. Urban areas have heavier traffic with respect to highways and rural areas. While traffic is heavy in highways and urban areas during work hours (and at suitable density for emergency message transfer over multi-hop paths), there is less number of vehicles on those roads at nights which increase the interruptions and route breakages in data transfer. While high node density causes a broadcast storm problem, low density causes disjointed vehicle clusters that partition the network. In highways and rural areas, these scenarios are more problematic than in urban areas. In both cases, the unique feature of intermittent connectivity between vehicles precludes end-to-end connections between source-destination pairs. As a result, main challenge in designing vehicular communication protocols becomes to provide good delay performance and successful packet delivery under the constraints of variable vehicular speeds, unreliable connectivity, and fast topological changes [2].

In such networks, communications become impossible in real time. Messages can be sent in a delay-tolerant fashion by the store-carry-forward method. A message is delayed at a node until an intermediate node is found to forward the message [3]. Delaying packets at intermediate nodes increases the end-to-end delay and makes the usual packet lifetime usage impractical. The lifetime of the packets needs to be determined while considering the delay times at the nodes. On the other hand, a short packet lifetime causes packet losses and a long packet lifetime increases the load at the nodes because of an increased number of buffered packets. Moreover, mobility, network connectivity and Quality of Service requirements of applications force more careful lifetime determination for the packets to be sent in such sparse networks. The packet lifetime that is determined affects the performance of the network. There is a need to determine efficient packet lifetime and vehicles' carry time to maximize the network performance in terms of packet delivery ratio and Quality of Service requirements of applications. In [4], the effects of partial connectivity and vehicle speeds on packet lifetime and vehicle carry times are examined, where the packet lifetime is determined to provide a desired packet delivery ratio for different levels of partial connectivity for sparse VANETs.

The communication protocol stack used in Vehicular Communications is given in **Fig. 1** [5]. Internet Protocol (IP) is the standard protocol for all of the applications, using the TCP and UDP protocols. In Japan, how to treat the IP layer is still an open issue. In the U.S.A., all of the applications other than the safety applications that use Dedicated Short Range Communication (DSRC) [6] use IPv6. Similarly, in Europe, all TCP/UDP traffic uses the IPv6 protocol. In the IPv6 protocol [7], however, the *Hop Limit* field is used to determine a packet's lifetime within the network instead of the *Time to Live (TTL)* field in the IPv4 protocol [8]. The packet lifetime in seconds becomes useless as the *Hop Limit* is decremented once at each hop. Similarly, although IPv4 packets are designed to carry packet lifetime in seconds in the *TTL* field, this field is also used as the hop count in practice. In this case, there is no mechanism to determine how long vehicles will carry packets in VANETs to deliver packets in a delay-tolerant fashion. Hence, IPv6 packets cannot be delivered in a delay-tolerant fashion in VANETs.

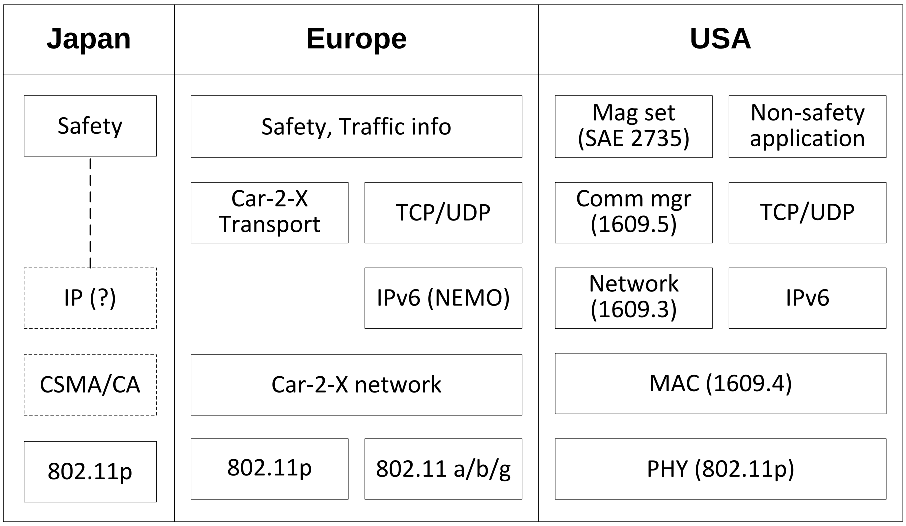

Figure 1 Car-2-X communication layers in Japan, Europe and U.S.A [5].

In this paper, we study the delivery of IPv6/IPv4 packets in a delay-tolerant fashion in partially connected networks. Because the use of lifetime in seconds is not applicable in IP packets, we propose methods which use the *Hop Limit* to maximize the packet delivery ratio. We examined first the effects of the vehicle carry time on the packet delivery ratio and second the relation between the hop limit and the carry time. Based on our analysis, we propose approaches that use the hop limit and the carry time pair and propose another novel method to increase performance results. Proposed methods are not new routing protocols but methods that allow enabling packet forwarding in delayed manner which can be used in conjunction with the routing protocols.

The work in this paper is one of the first studies that examines vehicle's packet carry time in conjunction with the hop limit to maximize the packet delivery ratio. In particular, our contributions are as follows:

- We point out the fact that rather than time in seconds, hop count is used as lifetime which is inserted into the *Hop Limit* field in the header of the IPv6 packets and *TTL* field in the header of IPv4 packets. Based on this fact, there is no use to insert lifetime in seconds which is contrarily common in related studies in the literature. As defined in the IPv6 protocol specification [7], it was left to upper-layer protocol that relies on the internet layer (whether IPv4 or IPv6) to limit packet lifetime and to provide its own mechanisms for detecting and discarding obsolete packets. In [7], maximum hop limit for IPv6 packets was defined as 255. Although, default value was not recommended in [7], 64 and 128 are used in practice. Use of high hop limit value and packet queuing when there is not a connection in the intermittently connected networks, e.g. VANETs, introduce additional delay and cause packet losses due to buffer fill-up.

- Our analysis suggests that lifetime should be determined with respect to current conditions of the network, application area such as urban or highway, mobility, speed of nodes, network connectivity and Quality of Service requirements of the applications. Determining the lifetime carefully will also be useful identifying suitable locations for placing stationary gateways that temporarily buffer messages and thus improve the probability of delivering packets to the destination.

- We present three approaches which define the lifetime in terms of packet carry time and hop limit. We also propose another method that improves packet delivery ratio for intermittently connected networks. The proposed method reduces packet drops considering the lifetime and can be used in conjunction with forwarding/routing schemes to improve the performance results in terms of packet delivery ratio while obeying the delay constraints.

This paper is organized as follows. Studies and related work on vehicular networks and delay-tolerant communication are presented in the second section. In the third section, the effects of the hop count and carry time on the packet delivery ratio are examined and the proposed approaches are demonstrated. Simulation and results are presented in the fourth section, and finally, we conclude our discussion in the last section.

2 Related Work

It is important to provide a reliable transport service over an unreliable network where the connections/routes get broken suddenly and frequently. Delay-Tolerant Networks (DTNs) [9] are the networks to overcome the lack of connectivity and to enable the communications between nodes and applications in such kind of intermittent connectivity. DTNs enable communication where connectivity issues like sparse and intermittent connectivity, long and variable delay, high latency, high error rates, highly asymmetric data rate, and even no connectivity exist between the communicating end-pairs [10]. Vehicular Ad Hoc Networks (VANETs) use data forwarding approaches proposed for DTNs where vehicles communicate with each other as well as fixed nodes placed along the roads or junctions in order to disseminate messages. The store-and-forward method in packet switching networks has evolved to an alternative that is a store, carry and forward method, where packets may also be carried by network nodes from a place to another, increasing communications efficiency. This method allows communication in challenging scenarios where additional delays are acceptable, because carrying is much slower than transmitting, resulting higher latency. Usually, there is a tradeoff between latency and delivery ratio. More messages can be delivered successfully if more time is allowed [10]. To support real-time and multimedia services as well as other type of delay-tolerant services, cross-layer protocol designs spanning transport and network layers can be beneficial and crucial. Therefore, the main challenge in designing forwarding algorithms for VANETs becomes to provide reliable packet transmission with minimum delay, maximum throughput, and low communication overhead [2]. Approaches and protocols for DTNs and VANETs deal only with the forwarding plane where the control plane issues remain open. However, there are a number of studies in the literature to apply routing protocols based on different schemes. Routing protocols and forwarding schemes are surveyed in [2, 3, 11-13]. Some of these routing protocols are summarized in the remaining of this section.

Classical ad hoc routing algorithms (such as Dynamic Source Routing (DSR) [14] and Ad Hoc On-Demand Distance Vector (AODV) [15]) become inadequate and impractical for Vehicular Ad Hoc Networks when the network density is low. The simplest approach that can be used for low network density is to spread the message throughout the network in a controlled manner with TTL usage, to make sure that it reaches its destination. For example, the Epidemic Routing Algorithm [16] uses this method; data packets are spread within the network similar to in the flooding algorithm. Therefore, this approach has all of the unfavorable features of the flooding algorithm. Dissemination of packets through the network is restrained by the Spray&Wait [17] and MoVe [18] algorithms, and unnecessary packet transmissions are reduced in these algorithms. The MoVe algorithm allows nodes to forward packets to vehicles whose direction is towards the destination, while the Spray&Wait algorithm sends packets to a certain number of vehicles chosen at random. The aim in all of these algorithms is to utilize simultaneous multiple paths to make certain that messages reach their destination. With these algorithms, however, it is not possible to communicate between disjoint clusters in a partially connected network. It is impossible to find a path between pairs of nodes that are in disjoint clusters.

Vehicle-Assisted Data Delivery (VADD) [19] protocol aims to forward the packet to the best road with the lowest data-delivery delay for urban road networks where different road segments have different vehicle densities. It attempts to utilize high density segments. However, such an approach increases the utilization of the channel due to high node density where packets can get dropped or incur high delay. Moreover, the approach relies on preloaded traffic statistics such as vehicle speed and traffic density at different time of the day. Therefore, it doesn't propose an approach for intermittently connected sparse networks. An improvement to VADD is SADV (Static-node assisted Adaptive data Dissemination protocol for Vehicular networks) [20] which attempts to avoid selection of non-optimal routes and attempts to reduce the packet delay by deploying static nodes at the intersections. Based on the two forwarding schemes proposed in [20], namely in-road forwarding and intersection forwarding, packets are forwarded either greedily along the road or forwarded to static nodes located at intersections, which compute minimum delay paths based on vehicle densities on different road segments. These static nodes have the ability to store the packets until they find appropriate vehicles for the computed minimum delay paths.

Predictive Graph Relay (PGR) [21], Delay-Bounded Greedy Forwarding (D-Greedy), and Delay-Bounded Minimum Cost Forwarding (D-MinCost) [22] are the geographically based algorithms that aim at transferring the packet by delaying in low density networks. These protocols aim to provide bounded transmission delay while minimizing the bandwidth utilization. D-MinCost has improvement on D-Greedy by incorporating additional factors such as vehicle density into the path selection process. A highly detailed study on packet delivery which is proposed in [23] is the CAN DELIVER, Carry and forwArd mechaNisms for Dependable mEssage deLIvery in VanEts using Rsus. The system designed to be in two parts; first part handles routing from a vehicle to the nearest fixed node placed along the road (called as Road Side Unit (RSU)) and the second part handles routing from RSUs to vehicles. It proposes a routing protocol suitable for both dense and sparse conditions by predicting the location of the destination in a precise manner and making use of all available nodes to reach the destination. Although the data delivery rate is enhanced in these algorithms, they do not address the issues about delays at nodes and packet lifetimes nor do they make an evaluation of these issues.

One early study on the end-to-end delay in delay-tolerant networks is presented in [24]. In this study, which is based on the Mobility Aware Routing Protocol and the Mobility Dissemination Protocol (MARP/MDP) [25] for airborne networks, the authors attempt to predict the minimum end-to-end delay and obtain a corresponding path for airborne networks. However, they do not address issues that are related to carry time and lifetime. Moreover, with regard to different levels of connectivity and speed of vehicles, the time needed to transfer the data packets and hence, the lifetime of the packets to remain in the network will change. The first study to address these issues and to determine the lifetime in sparse vehicular networks is the study presented in [4]. This study provides a substantive understanding of the relationship between the vehicle speeds, their carry time and the packet lifetime. Unfortunately, as described in the following section, the *Hop Limit* field in the IPv6 packets makes lifetime usage impractical. Thus, there is a need for approaches that enable the use of the lifetime in conjunction with the hop limit. Also, studies in the literature are not mature which analyze and propose new approaches that carry IPv6 packets by vehicles to provide a store-carry-forward mechanism. We aim to address these issues. We also propose a novel method to carry IPv6 packets to maximize the packet delivery ratio in such sparse networks. The approaches proposed in this paper are not a kind of forwarding or routing schemes, but help to the forwarding/routing schemes to improve the performance results in terms of packet delivery ratio while obeying the delay constraints.

3 Enabling Delay Tolerance for IPv6 Packets

Vehicles and nodes in sparse networks remain in clusters as partially connected when the full network connectivity is not provided depending on node density and speed of nodes. Sparse and intermittent connectivity affects the data transfer at an instant time causing route breakages and packet losses in end-to-end connections (**Fig. 2 (a-b)**). Store-carry-forward method can be used in order to minimize the packet losses for intermittent connectivity. In this mechanism, in case of no available neighbor nodes to forward the packet, the vehicle *stores* the packet and *carries* until it finds a vehicle/path to send and then *forwards* the packet (**Fig. 2 (c)**). By this way, even if there is not an end-to-end path at an instant time, due to continuous topology change in the network, the message will arrive to the destination in delayed time.

Key consideration for this method is determination of delay time; how long the message will be carried by vehicles (**Fig. 2 (d)**). Depending on the carry time of the message, the successful delivery probability of the message changes. Packet is dropped or is carried by the vehicles according to determined carry time. Depending on this carry time, lifetime of the packets in the networks varies, but using the packet lifetime infinite will also cause buffer fill-up and packet losses at the intermediate nodes.

Vehicular Ad Hoc Networks, DSRC (Dedicated Short Range Communications) [26] standards and 3G/WiMAX communications with roadside units require usage of the IPv6 protocol (**Fig. 1**). In IPv6, the *Hop Limit* field is defined by replacing the *Time To Live* field in IPv4. Although TTL is defined as the packet lifetime in seconds, to limit packet's life within the network, this field is used as a hop count in practice and is decremented once at each hop. The hop count avoids the implementation of a packet lifetime. Questions on how long the vehicles can carry the packets and how long a packet can live within the network remain open issues. The selection of these values randomly cannot provide the expected packet delivery ratio. This problem becomes solidified in vehicular networks, because of the low density property and the partial connectivity in such networks. Packet carrying time and hence packet lifetime depends on the partialness degree of the network. These values affect the packet delivery ratio. As can be seen in **Fig. 3** [4], a successful packet delivery depends on the determined packet lifetime.

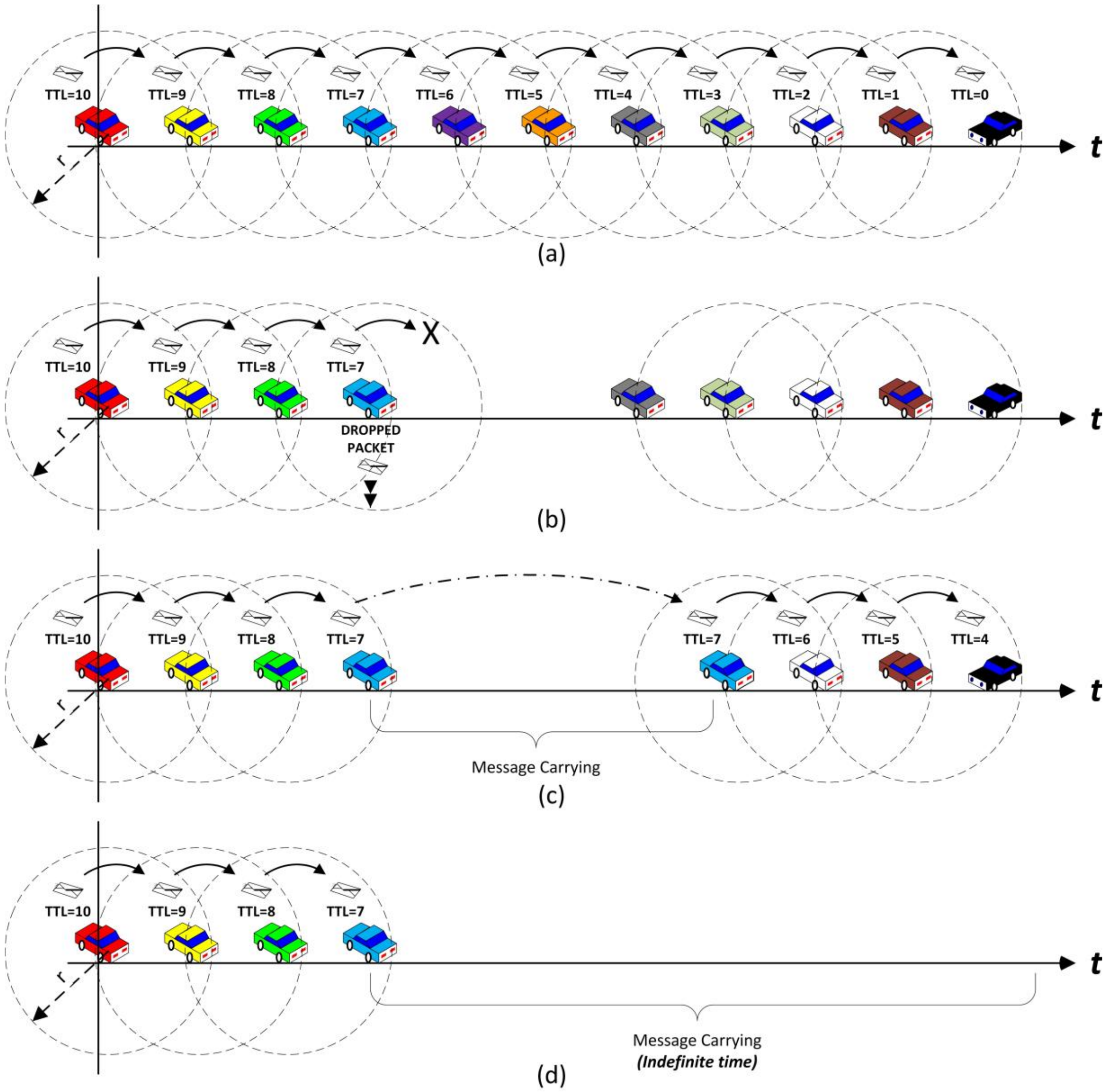

Figure 2 **(a)** Store-and-Forward Method, **(b)** Packet dropping occurs on route breaks/link failures in Store-and-Forward Method, **(c)** Store-Carry-Forward Method, **(d)** Indefinitely packet carrying in Store-Carry-Forward Method fills buffer which causes packet losses.

While the lifetime determines a packet's life within the network, the hop count determines the boundary it can reach. These two values can be used to provide the expected packet delivery ratio. However, there is no way to insert the lifetime into the IPv6 packet header. Rather than use the packet lifetime, one simple solution can be to use the carry time in conjunction with the hop count to generate the packet lifetime. A vehicle can carry packets at most predetermined carry times if it cannot find an intermediate node to forward the packet to. It transmits the packet immediately when it finds a node to forward the packet to, or drops the packet on carry timeout. However, it is not easy to determine the hop count and the carry time for partially connected networks. Partial connectivity affects the determination of both the hop count and the carry time values because the distance a vehicle should take before sending the packet depends on the level of partial connectivity of the network. A small hop count and/or a short carry time increase the packet drops, and a high hop count and/or a long carry time cause overhead in the network. An appropriate approach is to determine the carry time and the hop count by utilizing a balanced approach in which the total time is not allowed to exceed the packet's total lifetime, as given in equation (1).

$$Hop\ Count\ x\ CarryTime\ \leq\ Lifetime \qquad (1)$$

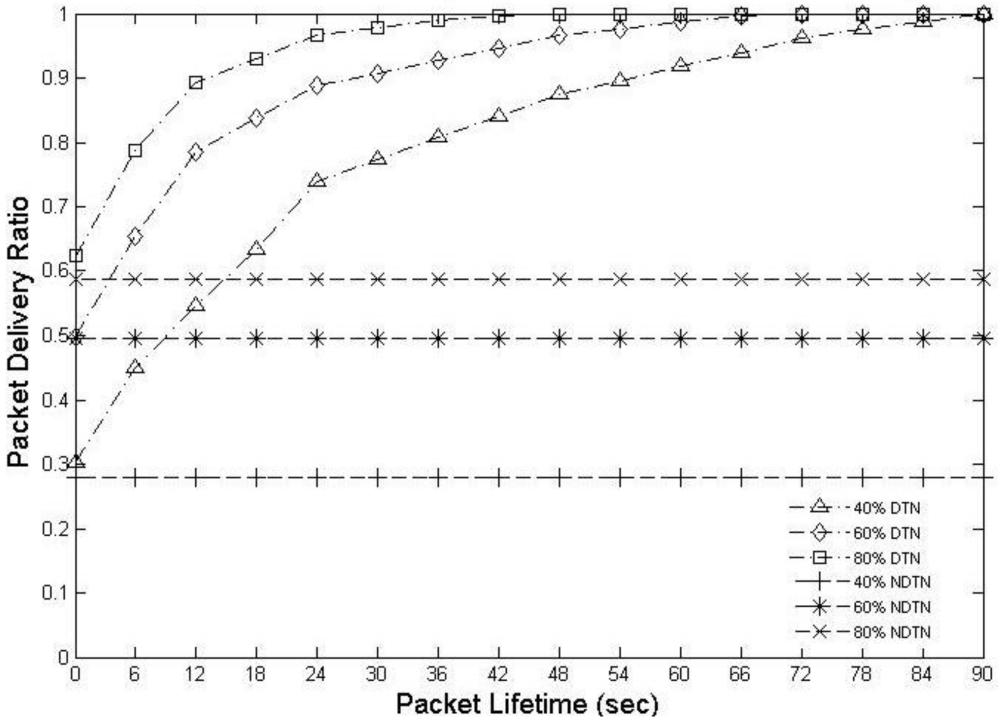

Figure 3 Effects of network connectivity and packet lifetime on the packet delivery ratio for Delay-Tolerant Communication Networks (DTN) and Non-Delay-Tolerant Communication Networks (NDTN) [4]. 40% DTN means Delay-Tolerant Communications Network with 40% connectivity and in this context, it has different meaning than Delay Tolerant Networks which was described in RFC 4838 and RFC 5050.

Equation (1) defines the lifetime in terms of the hop count and the carry time. One of our aims is to find the appropriate hop count and carry time values, which maximize the packet delivery ratio. Different approaches to find such a pair are given and summarized as follows.

After this point, we will use the term *Hops-To-Live (HTL)* instead of *Hop Limit* for IPv6 and instead of *Lifetime* in terms of hops in IPv4, to avoid confusion with TTL and to emphasize the difference.

If the lifetime is sustained at a constant value in equation (1), then an increase in the hop count (HTL) value causes a decrease in the packet carry time, or vice versa. The carry time depends on the selected HTL value. We therefore named the proposed approaches based on the selected HTL values. The first approach aims to find an HTL value that provides the best packet delivery ratio. Increasing the HTL value also increases the packet delivery ratio until it reaches a peak value. The carry time-HTL pair that yields the peak delivery ratio can be set as the best promising pair. We call this approach as the *Best Observed HTL* approach. The *Best Observed HTL* approach aims to utilize the knowledge obtained from past experience of its own and/or other nodes. It is possible for a vehicle to collect its own statistical data from past communications. Such statistical data can be provided to the vehicle from other vehicles in vicinity as well as RSUs with more accurate information.

Our second approach is based on applying an HTL value to allow packets to be transmitted in as many hops as possible. The aim in this approach is to increase the delivery ratio by reducing the drops that arise from HTL exhaustion. We call this approach the *Max HTL*. The *Max HTL* approach aims to find the boundary of the network for the packet to be traversed. The boundary is defined and found as a function of the maximum number of hops that a packet can be transmitted. By this way, a packet cannot be dropped due to HTL exhaustion. Note that increase in the hop count, however, reduces the carry time which may cause packet drops due to carry timeouts. The *Best Observed HTL* and the *Max HTL* approaches require a priori experimentation, knowledge or probing within the network to determine the values of these system-wide parameters (HTL and carry time), which will be used in common by all nodes.

Our third approach aims to reduce the number of drops that arise from the use of pre-determined common HTL values for all of the packets. In this approach, the source predicts the hop count to the destination (based on the distance and the transmission range) and uses this value as the HTL for the packets it sends. This predicted approach is named the *Expected HTL* approach. Each packet is set with a specific HTL value based on the distance between the source and destination pair. The *Expected HTL* approach aims to use more flexible HTL and carry time values depending on the distance between the communicating pairs.

These approaches are summarized below:

- ***Best Observed HTL*** aims to find the HTL and carry time pair that maximizes the delivery ratio.
- ***Max HTL*** aims to reduce drops due to hop count exhaustion and finds the needed maximum hop count.
- ***Expected HTL*** aims to use the value of the variable HTL, which is based on the source and destination pair.

In these three approaches, an appropriate HTL and carry time pair is found to maximize the packet delivery ratio. We can call such an appropriate pair the *Best Hop Count vs. Carry Time* pair. However, even with such appropriate values, packet losses can still be observed. Because of partial connectivity, a predefined carry time may be spent before an intermediate node forwards the packet (**Fig. 4 (b)**), or a predefined hop count may reach zero before the packet reaches its destination. In either case, the packet is dropped at the intermediate node. We propose a new novel method: to reduce drops due to hop count and/or carry time exhaustion and to obtain a packet delivery ratio higher than the maximized value that can be obtained with a *Best Hop Count vs. Carry Time* pair.

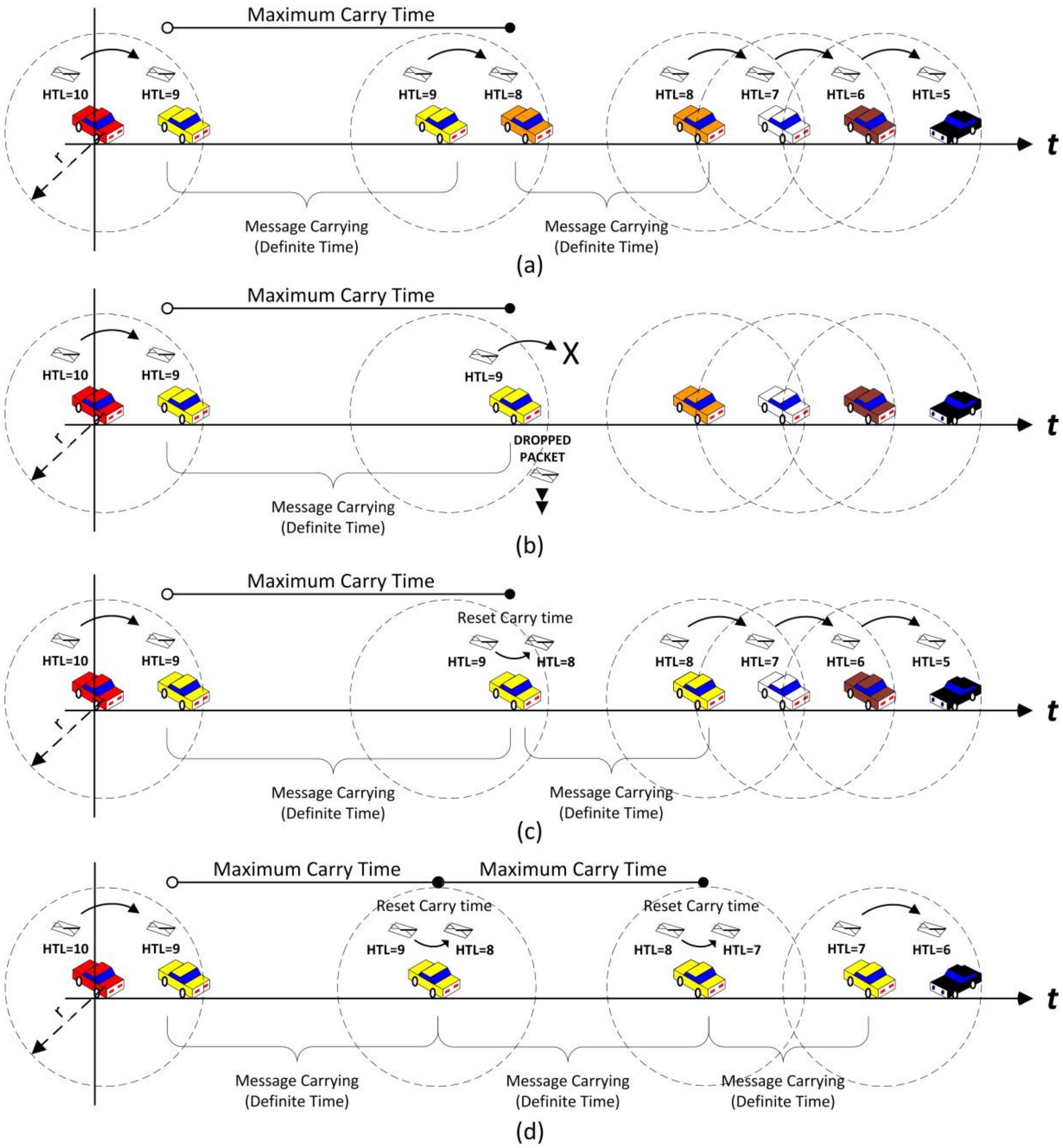

Figure 4 **(a)** Packet delivery with SCT and Store-Carry-Forward Method, **(b)** Packet dropping occurs on carry timeouts with the PCT method, **(c)** PCT with Store-Carry-Forward Method, **(d)** With the use of PCT method, packets never drop due to the carry timeouts. On carry timeouts, HTL value is decremented and the carry time is reset. This process continues until the node transmits the packet or the HTL value exhausts. Packet drops only occur on HTL exhaust.

```
for (each_second){
    decrement_one_value (txpacket_carrytime)
    if ((txpacket_carrytime > 0) && (txpacket_HTL> 0)) {
        keep_packet_in_buffer_to_be_sent()
    }// end of if
    else {
        drop_packet ()
    }// end of else
}// end of for
```

Algorithm 1 Pseudo Code of Steady Carry Time (SCT) Method.

```
for (each_second){
    decrement_one_value (txpacket_carrytime)
    if  (txpacket_carrytime > 0){
        keep_packet_in_buffer_to_be_sent()
    }// end of if
    else if (txpacket_HTL > 0){
        decrement_one_value (txpacket_HTL)
        reset (txpacket_carrytime)
        keep_packet_in_buffer_to_be_sent()
    }// end of else if
    else {
        drop_packet()
    }// end of else
}// end of for
```

Algorithm 2 Pseudo Code of Prolonged Carry Time (PCT) Method.

The proposed method works as follows. If a vehicle cannot find an intermediate node to forward the packet to within the carry time limit, it decreases the hop count by one (if greater than zero) and reinitiates the carry time, rather than dropping the packet (**Fig. 4 (c)**). In this way, it avoids drops due to carry time and continues to carry the packet until it finds a next node (**Fig. 4 (d)**) or until the hop limit (HTL) reaches zero. This method brings more balance and flexibility to the equation (1), providing resilience between the carry time and hop count. We named the former method the *Steady Carry Time (SCT)* method (the one that drops packets after carry time exhausts and given in **Algorithm 1**) and the new method the *Prolonged Carry Time (PCT)* **(Algorithm 2)**. In this new method, PCT, we are expecting to reduce the packet drops because of carry timeouts to a minimum, 0%. In the PCT method, an intermediate node does not drop packets on carry timeouts (**Fig. 4 (d)**). The node experiencing a connection problem, instead of dropping the packet, it continues to carry the packet while decreasing the hop HTL value by one (if the HTL value of the packet is greater than zero), and the carry time is reset. This process continues until it transmits the packet or the HTL value exhausts. In this way, packets never drop due to carry timeout. Moreover, total delay never exceeds the assessed lifetime which is determined at the source node. It never introduces additional delay but increases the packet delivery ratio.

PCT method allows packets to be carried as long as the hop limit is greater than zero, while the hop limit is decremented at each carry timeout. Packet drops due to carry timeouts are avoided with this method. It also helps nodes to keep packets as long as the packet is valid. Therefore, it also helps nodes to manage their queue and keep only recent packets.

4 Simulations and Results

In this section, a simulation environment is described and the simulation parameters are given to evaluate the effectiveness of the approaches and methods that we proposed in the previous section. In our analysis, we compared these approaches and showed the pros and cons of each approach. Then, the proposed methods were applied to these approaches. Simulation results related to the packet delivery ratio are presented.

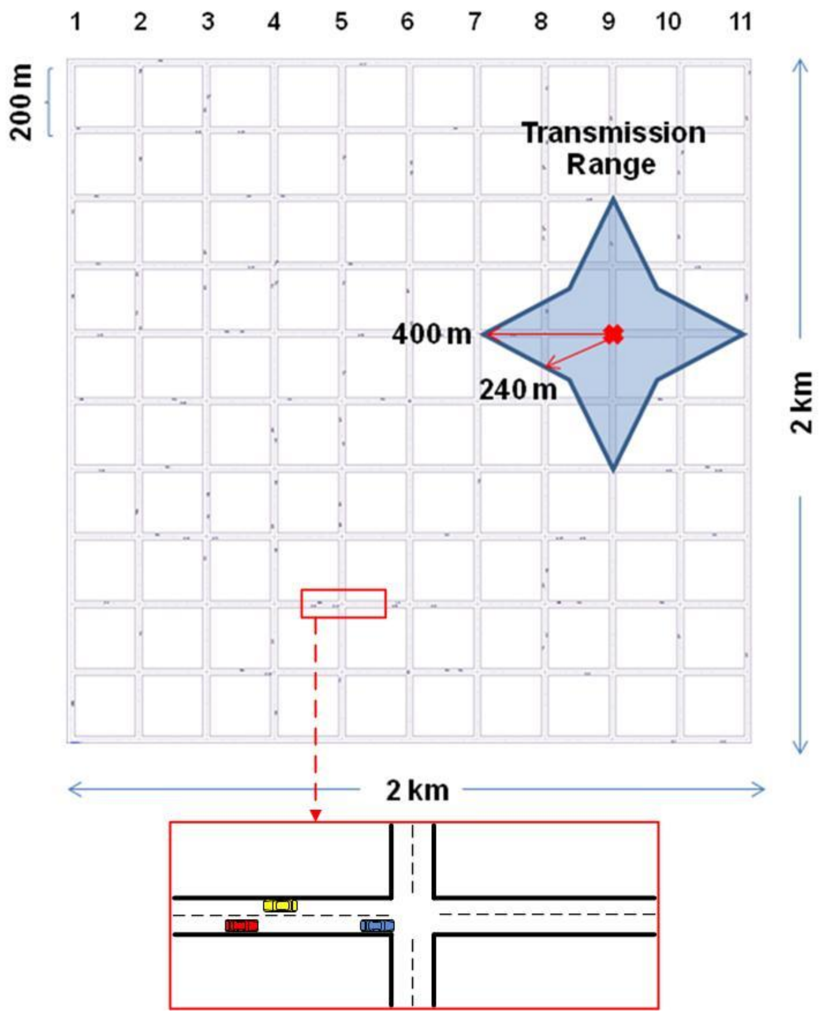

Figure 5 Network topology with Manhattan Grid Structure. Every street has two lanes with opposite directions.

4.1 Simulation Parameters

In the simulation of Vehicular Ad hoc Networks, we used the Manhattan Grid Topology and the Mobility Model [27] for the topology and traffic. **Fig. 5** shows the network topology at an instant time in an urban area. Parameter values are determined to be similar in other related studies [1]. The simulation environment constitutes various components and functions for communications, network construction, applied protocols and network dynamics. Main functions are: Random Vehicle Deployment, Direction and Speed Setting, Vehicle Movement, Connectivity Analyze, Event Generation, Communication related functions, Vehicles Plotting on Map (by Dislin program [28]).

Network Topology: Topology was constructed with 200m x 200m blocks over a 2km x 2km area. Two-way roads were used with one lane each way. Vehicles distributed randomly over the Manhattan Grid structured roads. Collisions in random positioning are avoided during the initial placement phase.

Movement Pattern: Vehicles move at a predetermined constant speed. At junctions, vehicles change their directions randomly either to the left or right or move in the same direction [27]. Although different vehicle speeds affect the performance results [1, 4], change in the average vehicle speed does not have an effect on the performance results [1, 29]. On the other hand, constant speed provides all details on packet drops and control on buffers at nodes. Therefore, vehicles within the network are assumed to travel at a constant speed similar to the studies presented in [1, 29].

Packet Types and Messaging Methods: Two types of packets are generally used in Vehicular Ad Hoc Networks [22]. The first type is a single-hop broadcast (beaconing) packet, and the second type is a multi-hop unicast packet. Beaconing packets containing routine traffic information comprised of location, speed, and direction of vehicles are forwarded as single hop messages. Application specific messages used by commercial applications requiring peer to peer communications use the second type (unicast, multi-hop) packets. In simulations, routine traffic messages broadcast with 3 second intervals as single hop packets. Unicast peer-to-peer data are forwarded with a load ratio of 1 event/minute per vehicle. Peer-to-peer data are sent as 5 successive unicast data packets, which make a load ratio of 5 packets/minute per vehicle in addition to the beaconing packets. We assumed that each node has knowledge of its neighborhood. In the experiments, the buffer size was kept finite but the packet drops due to an insufficient buffer size were not allowed. This approach prevents an insufficient buffer size from affecting the packet delivery ratio.

Communication Model: A 400-m transmission range [30], [31] is used for the vehicles. This value is the accepted communication range according to the DSRC standards [26]. As the signal propagation is affected from obstacles such as buildings, the communication range for the vehicles has been accepted as 240 m (Fig.3).

Measurement Factors: Measurements are observed for different connectivity levels (40%, 60%, 80%, and 100%). Some of the simulation system parameters are summarized in **Table 1**. Network connectivity is defined as the maximum fraction of vehicles that are connected at any given point in time, where any two vehicles can be connected either directly or indirectly (via a multi-hop route) [29]. The network connectivity (*NC*) is given in equation (2) [29], where N is the total number of vehicles in the network and $C(i, j, t)$ is the connectivity indicator. Connectivity indicator takes on the value of *1* if there is a path available from *Vehicle i* to *Vehicle j* at time *t*, and *0* otherwise. Network connectivity levels are shown in **Table 2**.

Table 1 Simulation Parameters

Parameter	Value
Simulation Area	2 km x 2 km
Number of Vehicles	70, 96, 120 vehicles
Transmission Range	400 m, 240 m
Simulation Duration	30 min.
Vehicle Speed	60 km/h
Block Size	200 m x 200 m

$$NC(t) \approx \max_i \left\{ \frac{1}{N} \sum_j C\left(i, j, t\right) \right\} \qquad (2)$$

Table 2 Connectivity Levels and Number of Vehicles

Connectivity Level	Number of Vehicles
40%	70 vehicles (18 vehicle/km2)
60%	96 vehicles (24 vehicle/km2)
80%	120 vehicles (30 vehicle/km2)
100%	180 vehicles (45 vehicle/km2)

4.2 Simulation Results

In this subsection, we examine and present the simulation results for the approaches proposed in the third section, which deliver IPv6 packets in a delay-tolerant fashion in partially connected networks. To compare these approaches, we evaluated the packet delivery ratio and the packet drops.

Simulations are comprised of two parts. In the first part, the appropriate hop count and carry time pair to maximize the packet delivery ratio are found (for the given approaches in Section 3) for different connectivity levels. In the second part, we compare the results of the SCT method and the PCT method.

Table 3 Packet Lifetimes For Different Connectivity Levels

Connectivity	Packet Lifetime
40%	90 sec.
60%	72 sec.
80%	48 sec.

In the simulations, the packet lifetime values given in **Table 3** were used. In partially connected networks, nodes send packets in a delayed manner if there is not any intermediate node to send. The total encountered delay varies depending on the network connectivity level, which determines the required packet lifetime of that network. These values, the required packet lifetime for different connectivity levels, were determined in the [4]. **Table 3** shows the required packet lifetime for different connectivity levels to provide a 100% packet delivery ratio.

Part 1: Finding the Appropriate HTL value and Relationship between the HTL and the Carry Time

The effects of different HTL values on the packet delivery ratio are presented in **Fig. 6**. If we maintain the packet lifetimes given in **Table 3** as constants, then an increase in the HTL value causes a decrease in the packet carry times. The relationship between these two parameters was given in Equation (1). As seen in **Fig. 6,** the packet delivery ratio never reaches 100% for the different connectivity levels (40%, 60% and 80%). An increase in the HTL value also increases the packet delivery ratio until it reaches its peak value, when the HTL is equal to 10, and then it starts to decline for HTL values higher than 10. As the HTL value increases, fewer packets drop because of an insufficient HTL value, which shows an increase for the packet delivery ratio in **Fig. 6**. On the other hand, the increase in the HTL value shortens the carry time, which thereafter causes packet drops because of carry timeouts at nodes. Therefore, the highest packet delivery ratio is observed at HTL 10. We will use this value as the *Best Observed HTL* value in the subsequent measurements. For a fully connected network (100% connectivity), the packet delivery ratio reaches to 100% and remains stable for HTL values of 14 and above. Packet losses resulting from low HTLs are not observed when the HTL is 14 and higher. For such a fully connected network, the HTL value should be at least 14, to provide a higher packet delivery ratio (**Fig. 6**). This value (HTL=14) can be used as the default maximum HTL value to size packets in the dissemination area. We will use this value as the *Max HTL* value for the remainder of the simulations.

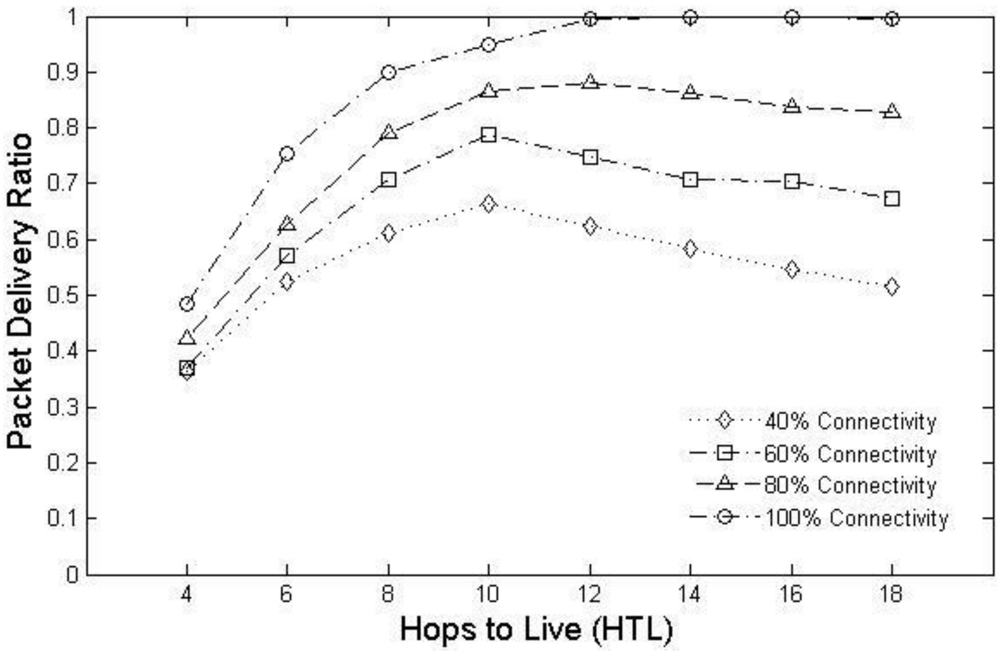

Figure 6 Effects of network connectivity ratio and HTL on the packet delivery ratio.

As presented above, increasing the HTL value does not increase the packet delivery ratio in partially connected networks. For example, the packet delivery ratio with *Best Observed HTL* (HTL=10) is higher than with *Max HTL* (HTL=14). It is related to the packet loss ratios that result from HTL and carry timeouts. As seen in **Fig. 7**, although there are not any packet losses resulting from HTL for the *Max HTL* approach, packet drop ratios from shortened carry times that are inversely proportional to the applied HTL values are high. As seen in **Fig. 7,** the *Best Observed HTL* approach presents the lowest aggregate packet drops (sum of packet drops caused by HTL and carry timeouts) and hence, provides the highest packet delivery ratio.

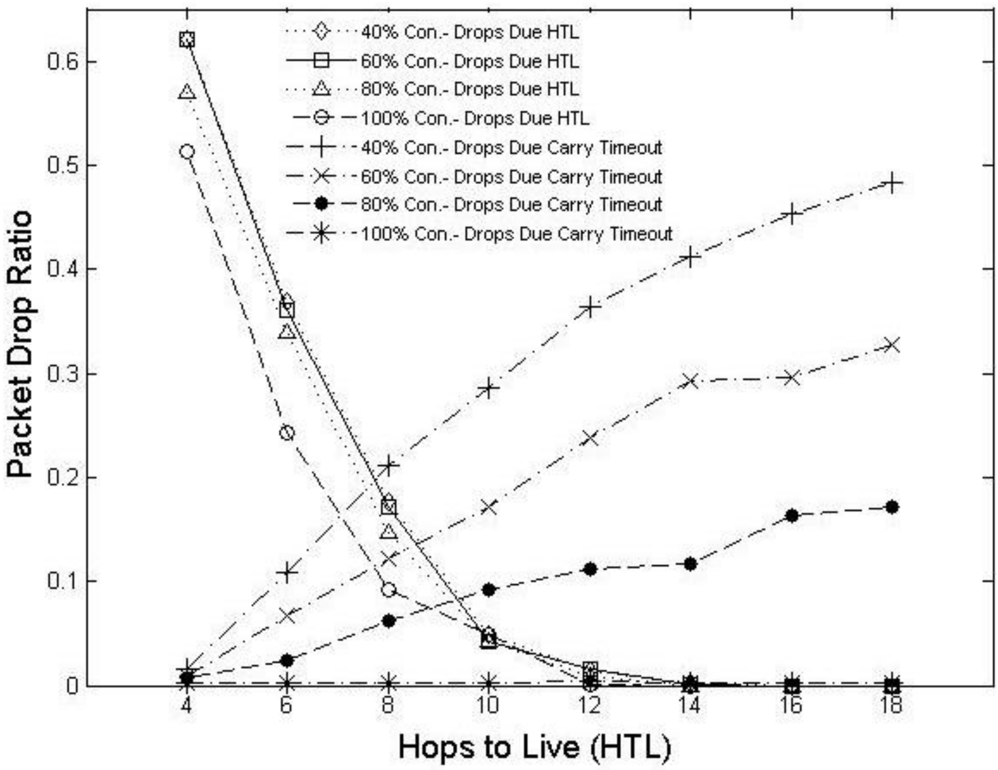

Figure 7 Effects of network connectivity ratio and HTL on the packet drop ratio.

Considering the case defined above, if the packet lifetime that maximizes the packet delivery ratio is known (or an intuitive lifetime is applied), then the pair of HTL-carry time values that preserve/provide the maximum packet delivery ratio can be determined. When the nodes (vehicles) are aware of the carry time (or it is provided to them), the hop limit or the HTL mechanism is applied in a delay-tolerant fashion for IPv6 packets. Consequently, the main aim becomes to decrease the packet drops from HTL and carry timeouts. For the given scenario, the *Best HTL-Carry time* pair is yielded when the HTL value is 10. However, the carry time varies with respect to the applied network connectivity level. For the given scenario, the HTL-carry time pairs that maximize the packet delivery ratios for different network connectivity levels are given in **Table 4**.

Table 4 Best Observed HTL – Carry Time Pairs

Connectivity	Max HTL	Best Observed HTL/ Carry time
40%	14 hop	10 hop / 9 sec.
60%	14 hop	10 hop / 7.2 sec
80%	14 hop	10 hop / 4.8 sec

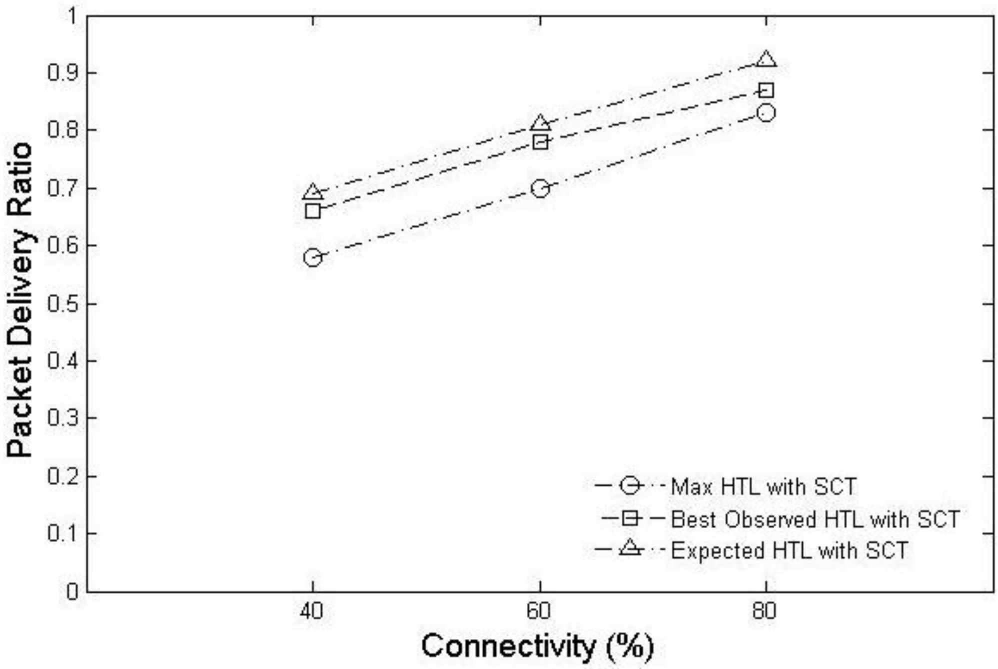

Figure 8 Packet Delivery Ratios of the HTL approaches for different connectivity levels.

Drops that result from the HTL can be reduced using the following approach. If the source predicts the hop count to the destination (based on the distance and transmission range), it may use this value as the HTL in the packets that it sends. We will use this value as the *Expected HTL* value. For a source-destination pair, each packet will be sent with a specific *Expected HTL* value based on the distance between the source and destination pairs. However, carry times cannot change depending on the source-destination pairs. Carry times should be unique for all of the nodes within the network. There is no way to put such information, e.g., the carry time, into the IPv6 packets. Nodes within the network can be informed with this unique value via Road-Side-Units (RSU) or GSM infrastructures. Because the carry time is a system-wide unique value, packet drops arising from carry timeout will always be possible. When this approach is compared with others, it is seen in **Fig. 8** that the packet delivery ratio with *Expected HTL* usage is higher than *Max HTL* and *Best Observed HTL* usages. In this approach, predicting the HTL value is the major issue because the partial connectivity may cause erroneous predictions.

Part 2: Comparison of SCT and PCT

Our next aim is to minimize packet drops due to the HTL and carry timeouts, to provide a 100% (or close to 100%) packet delivery ratio. On the other hand, as shown in **Fig. 6** and **Fig. 7**, there is not a perfect pair of HTL-carry times. Only, packet drops caused by HTL values can be reduced to zero (or close to zero) by using high HTL values. With the new proposed method, PCT, we are decreasing the packet drops - that are attributed to carry timeouts - going to zero. Compared the other approaches, better results are obtained. In the PCT method, packets are not dropped because of carry timeout at intermediate nodes. If the HTL is greater than zero, the HTL value is decremented by one value and the carry time is reset. This process continues until it transmits the packet or the HTL value exhausts. In this way, the packets never drop because of carry timeouts. This method may cause drops attributed to HTL, but the resulting ratio remains negligible. As a result, the HTL and carry time pair (and the lifetime) is used more effectively by the PCT method. The results are shown in **Fig. 9** and **Fig. 10**. **Fig. 9** shows packet delivery ratios of the PCT method with the *Best Observed HTL*, *Max HTL*, and *Expected HTL* approaches and their appropriate HTL-carry time pairs. With the usage of the PCT method, packet delivery ratios are increased to higher values in all of the approaches. For the *Max HTL* approach, a 99% packet delivery ratio is obtained. To compare the PCT with the SCT (which uses steady carry time) for each of HTL approaches, **Fig. 10** is presented. An enhancement of the PCT method to the SCT method is shown in **Fig. 10**. There is a substantial amount of increase in the packet delivery ratio for the PCT method.

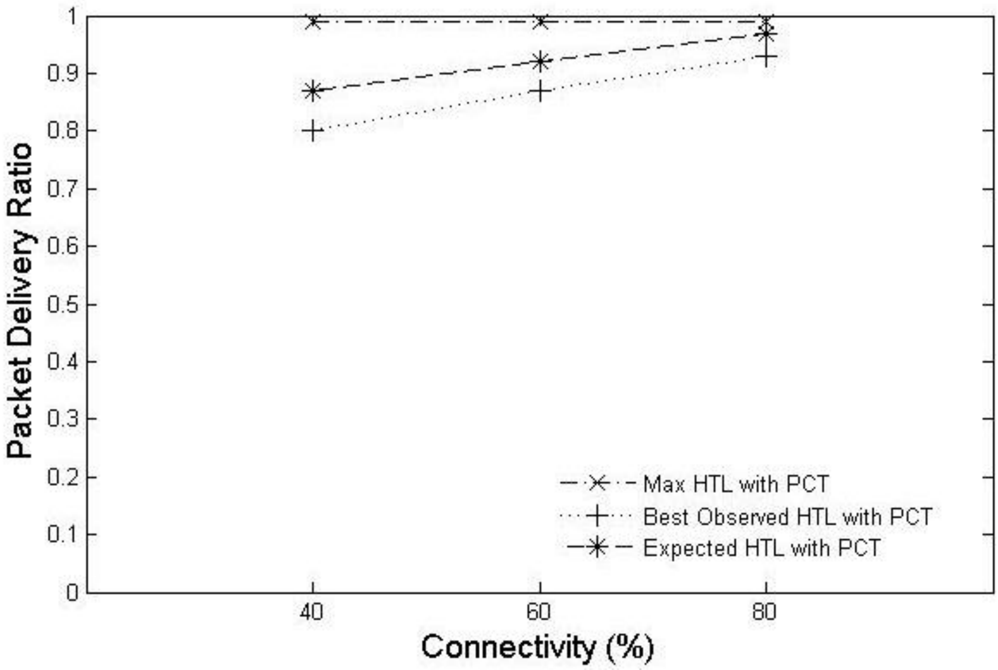

Figure 9 Effects of the PCT method for HTL approaches at different connectivity levels.

In Part 1, we determined the appropriate HTL-carry time pair for the approaches *Best Observed HTL*, *Max HTL* and *Expected HTL*, to maximize the packet delivery ratio for different connectivity levels. In the *Expected HTL* approach, however, determination of the HTL value is the main concern when the network is partially connected. In Part 2, effects of the PCT method, which uses the carry time more effectively and avoids packet drops that result from carry timeout, are shown for the HTL approaches defined in Part 1.

In a partially connected network, each of the three HTL approaches can be used. Of these approaches, *Expected HTL* provides a better packet delivery ratio. As defined before, it is difficult to determine *Expected HTL* in sparse VANETs. Of the other two approaches, the *Best Observed HTL* provides better results than *Max HTL*. It is difficult to determine the correct HTL-carry time pair. This determination may require a priori calculations or probing to find the best pair. The *Max TTL* approach is simple. The highest possible hop count of the path is taken to be the HTL value. In Part 2, we show that the PCT method improves the results of all of these three approaches. The success of the PCT method can be attributed to effective carry time usage in this method. Packet drops resulting from carry timeouts are avoided in PCT. For this reason, better results (99% packet delivery ratio) are obtained when PCT is used with the *Max HTL* approach.

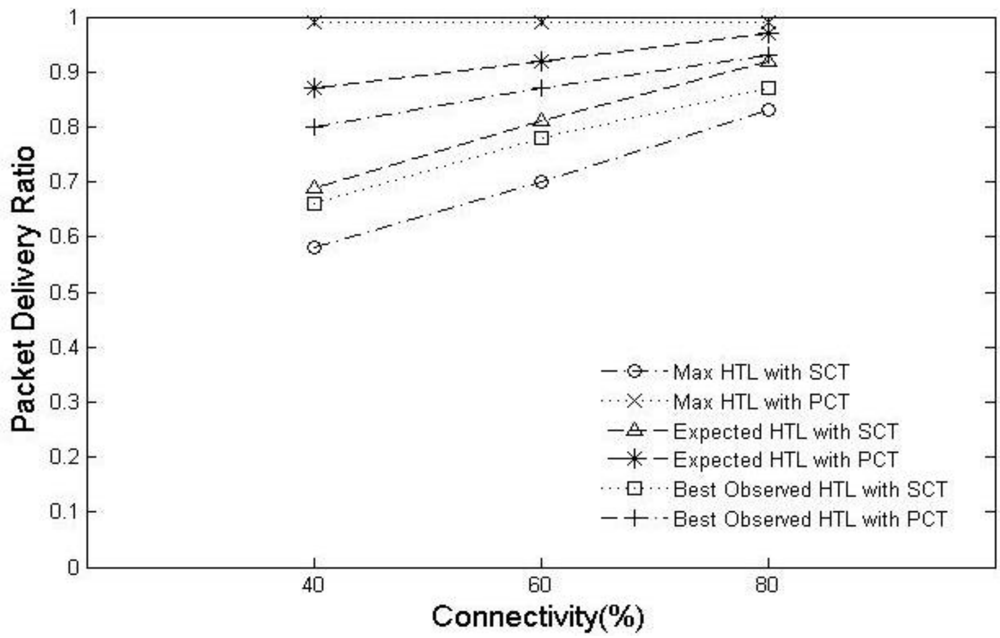

Figure 10 Effects of the PCT and SCT methods for the HTL approaches at different connectivity levels.

In mobile applications, especially during crises and disasters, connections between the nodes sporadically fail within the network, avoiding continuous communications. Sending data in a delay-tolerant fashion provides a connection even though it is delayed. Determining the HTL and carry time for the data that will be sent in a delayed manner affects the packet delivery ratio. By using the proposed method, PCT, whatever the carry time is, the packet delivery ratio is maximized. As a result, this method can be used very efficiently both for VANETs and for mobile and immobile networks, where connections get broken frequently.

5 Conclusions

In Vehicular Ad hoc Networks (VANETs), connections can get broken frequently due to intermittent connectivity. In the case of partial connectivity because of sparseness of the network or features peculiar to VANETs, packets are delivered to destinations by the store-carry-forward method. On the other hand, IPv6 is the standard protocol in IP layers for applications that use UDP/TCP traffic. However, there is no mechanism in the IPv6 header for delivering a packet in a delayed manner because the packet lifetime is determined with the hop limit field. In this paper, we studied the delivery of IPv6/IPv4 packets in a delay-tolerant fashion in partially connected networks. Because the use of lifetime in seconds is not applicable in IP packets, we propose methods which use the Hop Limit to maximize the packet delivery ratio. We also examined the effects of the vehicle carry time on the packet delivery ratio and the relation between the hop limit and the carry time. Based on our analysis, we proposed three approaches that use the hop limit and the carry time pair. Three approaches, namely *Best Observed HTL*, *Max HTL* and *Expected HTL*, are defined to maximize the packet delivery ratio. Because the carry time has a system-wide unique value, packet drops resulting from carry timeout can always occur with these approaches. Therefore, we proposed a novel method, the *Prolonged Carry Time (PCT)*, to avoid drops from carry timeouts. This method uses the carry time and the lifetime very effectively without suffering from carry timeout drops. Our analysis showed that, for all partial connectivity levels, the packet delivery ratio reached 99% for our test topology. This method can also be used in other types of networks where the nodes are mobile or connections break frequently. It will also be useful identifying suitable locations for placing stationary gateways that temporarily buffer messages and thus improve the probability of delivering packets to the destination. Moreover, *Prolonged Carry Time* method can be used in conjunction with routing schemes to improve the performance results in terms of packet delivery ratio while obeying the delay constraints.

References

[1] N. Wisitpongphan, F. Bai, P. Mudalige, V. Sadekar, O. Tonguz, "Routing in Sparse Vehicular Ad Hoc Wireless Networks", *IEEE Journal on Selected Areas in Communications*, Vol. 25, No. 8, October 2007.

[2] G. Karagiannis, O. Altintas, E. Ekici, G. Heijenk, B. Jarupan, K. Lin, T. Weil, "Vehicular Networking: A Survey and Tutorial on Requirements, Architectures, Challenges, Standards and Solutions," *IEEE Communications Surveys & Tutorials,* Vol.13, No.4, Fourth Quarter 2011, 584-616

[3] S. Panichpapiboon, W. Pattara-atikom, "A Review of Information Dissemination Protocols for Vehicular Ad Hoc Networks," *IEEE Communications Surveys & Tutorials,* Vol.PP, No.99, pp.1-15, 2011

[4] K.K. Sevimli, M. Soyturk, "Lifetime Determination for Delay Tolerant Communication in Sparse Vehicular Networks", *International Symposium on Wireless Pervasive Computing ISWPC 2010*, Modena, May 2010.

[5] S. Sai, K.Mase, M. Nishibori, J. Inoue, M. Obuchi, "Field Evaluation of UHF Radio Propogation for an ITS Safety System in an Urban Environment", *IEEE Communications*, November 2009.

[6] (2010) DSRC Website at Berkeley University. [Online] Available: http://path.berkeley.edu/dsrc/

[7] Internet Protocol, Version 6 (IPv6) Specification, RFC 2460, http://www.ietf.org/rfc/rfc2460.txt

[8] Internet Protocol, Darpa Internet Program Protocol Specification, RFC 791, http://www.ietf.org/rfc/rfc0791.txt

[9] K. Fall, S. Farrell, "DTN: an architectural retrospective," *IEEE Journal on Selected Areas in Communications*, Vol.26, No.5, June 2008, 828-836.

[10] P. Pereira, A. Casaca, J. Rodrigues, V. Soares, J. Triay, C. Cervello-Pastor, "From Delay-Tolerant Networks to Vehicular Delay-Tolerant Networks," *IEEE Communications Surveys & Tutorials*, Vol.PP, No.99, 2011

[11] K. C. Lee, U. Lee, M. Gerla, "Survey of Routing Protocols in Vehicular Ad Hoc Networks", *Advances in Vehicular Ad-Hoc Networks: Developments and Challenges, IGI Global*, Oct, 2009.

[12] Y.-W. Lin, Y.-S. Chen, S.-L. Lee, "Routing Protocols in Vehicular Ah Hoc Networks: A Survey and Future Perspectives," *Journal of Information Science and Engineering*, 2009.

[13]T. Spyropoulos, R. N. B. Rais, T. Turletti, K. Obraczka, and A. V. Vasilakos, “Routing for Disruption Tolerant Networks: Taxonomy and Design,” ACM/Springer Wireless Networks, Vol. 16, No. 8, pp. 2349–2370, 2010.

[14]D. B. Johnson, D. A. Maltz, J. Broch,"DSR: The Dynamic Source Routing Protocol For Multi-Hop Wireless Ad Hoc Networks", *Ad Hoc Network*, Chapter 5, Editor CE. Perkins, Addison-Wesley, 2001

[15]C.E. Perkins., E.M. Royer, "Ad-hoc On-Demand Distance Vector Routing", Proceedings of *Mobile Computing Systems and Applications*, pp. 90-100, 1999.

[16]A. Vahdat, D. Becker, “Epidemic Routing for Partially-Connected Ad Hoc Networks”, Duke University, *Technical Report CS-200006*, April 2000.

[17]T. Spyropoulos, K. Psounis, C. S. Raghavendra, “Spray and Wait: Efficient routing scheme for intermittently connected mobile networks”, *ACM SIGCOMM workshop on Delay Tolerant Networking (WDTN)*, 2005.

[18]J. LeBrun, C. N. Chuah, D. Ghosal, “Knowledge- Based Opportunistic Forwarding in Vehicular Wireless Ad Hoc Networks”, *IEEE VTC*, Spring 2005.

[19]J. Zhao and G. Cao, “VADD: Vehicle-Assisted Data Delivery in Vehicular Ad Hoc Networks”, *IEEE Transactions on Vehicular Technology*, Vol. 57, No. 3, pp. 1910–1922, 2008

[20]Y. Ding, C. Wang, L. Xiao. “A static-node assisted adaptive routing protocol in vehicular networks”. In *Proceedings of the fourth ACM International Workshop on Vehicular Ad Hoc Networks* (VANET '07). ACM, 2007

[21]J. Kurhinen, J. Janatuinen, “Delay Tolerant Routing In Sparse Vehicular Ad Hoc Networks”, *Acta Electrotechnica et Informatica*, Vol. 8, No. 3, 2008, 7–13

[22]A. Skordylis, N. Trigoni, “Delay-bounded Routing in Vehicular Ad-hoc Networks”, *MobiHoc’08*, May 2008.

[23]K. Mershad, H. Artail, M. Gerla, "We Can Deliver Messages to Far Vehicles," *IEEE Transactions on Intelligent Transportation Systems*, Vol.PP, No.99, 2012

[24]K. C. Lee, A. Piechowicz, M. Gerla, A. Tiwari, A. Ganguli, D. Krzysiak, "Delay Tolerant Mobility Aware Routing/Mobility Dissemination Protocol for the Airborne Network," *IEEE MILCOM 2009*, Boston, MA, October 2009.

[25]A. Tiwari, A. Ganguli, A. Sampath, D. S. Anderson, B.-h. Shen, N. Krishnamurthi, J. Yadegar, M. Gerla, D. Krzysiak, “Mobility Aware Routing for the Airborne Network Backbone”, *IEEE MILCOM 2008*, San Diego, CA, October 2008.

[26]Vehicle Safety Communications Project Task 3 Final Report, “Identify intelligent vehicle safety applications enabled by dsrc, dot hs 809 859”, *NHTSA*, March 2005. http://www-nrd.nhtsa.dot.gov

[27](2010) ETSI, Universal Mobile Telecommunication System (UMTS), “Selection procedures for the choice of radio transmission Technologies of the UMTS”, *UMTS 30.03 Version 3.2.0,* 1998-04. [Online]. Available: http://www.3gpp.org/ftp/Specs/html-info/3003U.htm

[28]“Dislin.” Online: http://www.dislin.de/

[29]W. Viriyasitavat, F. Bai, O.K. Tonguz. "Dynamics of Network Connectivity in Urban Vehicular Networks", *IEEE Journal on Selected Areas in Communications*, Vol. 29, No. 3, pp.515-533, March 2011.

[30]J. Nzouonta, N. Rajgure, G. Wang, and C. Borcea. "VANET Routing on City Roads using Real-Time Vehicular Traffic Information", *IEEE Transactions on Vehicular Technology*, Vol. 58, No. 7, 2009.

[31]N. Eude, B. Ducourthial, and M. Shawky. "Enhancing NS-2 simulator for high mobility ad hoc networks in car-to-car communication", *7th IFIP International Conference on Mobile and Wireless Communications Networks*, Marrakech, Morocco, September 2005.

Kadri Kaan Sevimli received the B.Sc. degree in Electrical and Electronics Engineering Department from Naval Academy, Istanbul, Turkey, in 2003. He received his M.Sc. degree in Computer Engineering from Naval Science and Engineering Institute, Istanbul, Turkey, in 2010. He is currently working at Turkish Naval Research Center Command (TNRCC) in Istanbul, Turkey. His research interests include Wireless and Mobile Communications and Vehicular Ad Hoc Networks.

Mujdat Soyturk received the B.Sc. degree in Industrial Engineering from Naval Academy, Istanbul, Turkey, in 1994. He received his M.Sc. and Ph.D. degrees both from Istanbul Technical University, Istanbul, Turkey, in Computer Engineering in 2002 and 2007, respectively. He was a visiting researcher for C4I Systems (Command, Control, Computer, Communications and Intelligence Systems) in Naval Postgraduate School (NPS), Monterey, CA, in 2009. His research interests include Wireless and Mobile Communications, Sensor and Ad Hoc Networks, Vehicular Networks, and Net-Centric Communications.

Please note that this version of the paper is the post-peer review, accepted paper submitted for final publication. The printed and reformatted version is accessible at the original source of the publication (http://www.inderscience.com/info/inarticle.php?artid=50264), where cited as "*Kadri Kaan Sevimli, Mujdat Soyturk. Enabling Delay-Tolerant Communications for Partially Connected Vehicular Ad Hoc Networks, International Journal of Ad Hoc and Ubiquitous Computing, Vol.11, No.2/3, pp.157-168, 2012 (2012)*".