

## Research Article

# A Comprehensive Evaluation of RPL under Mobility

**Kevin C. Lee,<sup>1</sup> Raghuram Sudhaakar,<sup>1</sup> Jianxia Ning,<sup>2</sup> Lillian Dai,<sup>1</sup> Sateesh Addepalli,<sup>1</sup>  
J. P. Vasseur,<sup>1</sup> and Mario Gerla<sup>3</sup>**

<sup>1</sup> Cisco Systems, Inc., 170 West Tasman Dr. San Jose, CA 95134, USA

<sup>2</sup> Department of Computer Science and Engineering, University of California, Riverside, 351 Winston Chung Hall, Riverside, CA 92521, USA

<sup>3</sup> Department of Computer Science, University of California, Los Angeles, 4732 Boelter Hall, Los Angeles, CA 90095, USA

Correspondence should be addressed to Kevin C. Lee, kevinl2@cisco.com

Received 15 September 2011; Revised 1 December 2011; Accepted 6 December 2011

Academic Editor: Athanasios Panagopoulos

Copyright © 2012 Kevin C. Lee et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper focuses on routing for vehicles getting access to infrastructure either directly or via multiple hops through other vehicles. We study routing protocol for low-power and lossy networks (RPL), a tree-based routing protocol designed for sensor networks. Many design elements from RPL are transferable to the vehicular environment. We provide a simulation performance study of RPL and RPL tuning in VANETs. More specifically, we seek to study the impact of RPL's various parameters and external factors (e.g., various timers and speeds) on its performance and obtain insights on RPL tuning for its use in VANETs. We then fine tune RPL and obtain performance gain over existing RPL.

## 1. Introduction

The term vehicular ad hoc network (VANET) traditionally refers to pure ad hoc wireless networking between vehicles. We believe that the time is ripe to explore a joint infrastructure and ad hoc wireless networking architecture for consumer vehicles (e.g., sedans, vans, performance cars, buses, trucks, trains, etc.). This is brought on by two recent trends:

- (i) Wireless service providers such as AT and T and Verizon have changed from a fixed monthly fee to a tiered or per bit pricing structure for data usage. At the same time, these service providers have started to roll out their own WiFi networks. As free WiFi networks become more accessible from vehicles, users will have a strong economic incentive to opportunistically offload data traffic from 3G and 4G links to free WiFi links. Since WiFi deployment is not prevalent, enabling vehicles to access roadside WiFi through other vehicles (multihop-to-infrastructure) allows more vehicles to take advantage of data offloading. From the perspective of the service providers, enabling vehicles to multihop-to-infrastructure reduces the number of WiFi access points they need to deploy,

thereby reducing the capital cost of WiFi infrastructure rollout.

- (ii) As more vehicles are connected in the future, social applications for drivers and passengers will become more prevalent. Already, such applications (e.g., traffic alert, speed trap knowledge sharing, etc.) are gaining popularity on smartphones. These applications tend to be strongly location dependent and are relevant to clusters of vehicles in close proximity. Such local, transient, and potentially high volume data traffic is inherently suited for ad hoc wireless networking. On the other hand, infrastructure networks such as 3G, 4G, and WiFi are still more suited for traditional Internet access.

Within the context of vehicular networking with both infrastructure and ad hoc networking capabilities, questions remain as to the appropriate routing protocols to be used and how the two different types of networks can be seamlessly integrated and complementary to each other for cost reduction and performance enhancements. In this paper, we focus on the former routing question for vehicles getting wireless access to road-side infrastructure either directly or via multiple hops through other vehicles.

We envision that future consumer vehicles will be equipped with a mobile onboard unit (OBU) with multiple wireless interfaces (e.g., 3G, 4G, WiFi, DSRC, etc.). Some of these interfaces will be used for direct access between vehicle and road-side infrastructure and some for vehicle-to-vehicle communication. A prototype developed jointly by our research group and our research partners successfully demonstrated in-vehicle wireless and wired connection to the OBU and the OBU achieving seamless mobility (i.e., session continuity) across heterogeneous external wireless networks such as 3G, 4G, and WiFi, fully utilizing all the links for opportunistic throughput improvement, load balancing, and fault tolerance. Each OBU is thus a router with multiple wireless interfaces that can connect to road-side infrastructure and to each other.

Intuitively, a tree topology is best suited for vehicles multihopping to road-side infrastructures. When multiple road-side infrastructures are available, multiple trees may be formed which are not necessarily mutually exclusive. In addition to the complexity of topology formation, an effective routing protocol needs to strike a balance between performance and protocol overhead. Furthermore, a vehicular network is not always highly dynamic as those who live in major metropolis can attest. Frequently, traffic congestion and traffic lights create large and relatively stationary vehicular networks. Hence, a multihop-to-infrastructure protocol for vehicular networks must work well in both static and highly mobile environments.

In this paper, we study Routing Protocol for low-power and lossy networks (RPL) [1], which can be adapted to meeting these requirements. The resulting multihop-to-infrastructure architecture enables large geographical area coverage of connected vehicles with relatively minimal deployment of infrastructure.

RPL was originally designed to meet specific requirements in low-power and lossy networks (LLNs), such as sensor networks. On first glance, this seems to be an irrelevant protocol for vehicular networks. However, those who are familiar with the work know that a common group of researcher/industry practitioners drove the development of RPL and other tree-based protocols, including NEMO [2] and NEMO+ [3]. Many design elements from RPL are transferable to the vehicular environment as we will discuss in detail. This is not to say that we can apply RPL directly to a vehicular environment. Our main contribution in this paper is to provide a simulation performance study of RPL and RPL tuning for vehicles multihopping to infrastructure under several routing metrics. Past evaluation of RPL in [4, 5] has focused on static sensor networks. To the best of our knowledge, this paper is the first to evaluate RPL and RPL tuning in vehicular environments. The enhancements to RPL by no means replace its intended use for sensor networks but beef up the existing protocol for highly mobile vehicular environments.

The rest of the paper is organized as follows. Section 2 describes the motivation for RPL among the already crowded routing protocol space. Section 4 describes RPL and implementation details. Section 5 discusses our modifications to adapt RPL for VANETs. Section 6 evaluates RPL under

several tunable parameters in VANETs. Section 7 concludes the paper and presents the future work.

## 2. Motivation for RPL

RPL is an IPv6 distance vector routing protocol that builds a Destination Oriented Directed Acyclic Graph (DODAG) rooted at a single destination (It is also known as sink or root. For the rest of the paper, we will use root.) (e.g., at a WiFi access point). It is actively maintained by the IETF ROLL Working Group [6]. When there is a single root, the network takes on a tree topology, which can be efficient for multihop routing to a road-side infrastructure such as a WiFi access point. Since RPL forms a tree, it naturally falls in place for the multihop-to-infrastructure routing setting.

In general, multiple roots are possible. The DODAG minimizes the cost of reaching the root from any node in the network as per an objective function (OF). The OF can be set to minimize a particular metric, such as hop count or expected transmission count (ETX). A list of metrics is specified in [7]. Considering the vast number of applications for RPL, the routing protocol has been designed with a great deal of flexibility and supports a variety of OFs in order to build the routing topology according to various link/node metrics and constraints.

As mentioned before, VANETs are at times static and, at other times, highly dynamic. An efficient routing protocol will need to adapt to the changing dynamism of the network in real time. Several attributes of RPL make it a favorable baseline candidate for such applications; enhancements by no means replace the intended use of RPL for sensor networks.

- (i) Compared with reactive routing protocols (e.g., AODV and DSR), RPL may have better response time since routes are available upon request (e.g., always routing through the parent node). Reactive routing protocols are susceptible to route breaks under high mobility. Past simulation results [8] show that most reactive routing protocols (e.g., AODV and DSR) suffer from highly dynamic nature of node mobility because they tend to have poor route convergence and low communication throughput. With RPL, we can modify it relatively easily to adapt the rate at which parent node is updated based on the dynamism of the network.
- (ii) Unlike any other proactive routing protocols, RPL does not flood the network with global network topology information so as to make it unscalable. Rather, local information is exchanged amongst the neighbors. Because routes are established in a tree structure towards the root, the natural usage of it is in a WLAN/cellular and hybrid VANET network architecture [9] where the root is at the APs.
- (iii) RPL may be coupled with another multihop routing protocol to improve routing efficiency for noninfrastructure peer-to-peer communication. While this is

outside of the scope of the current paper, a peer-to-peer routing protocol is important so as not to constrain the communication at the tree boundary (large tree means frequent updates; high mobility may require even more frequent updates; this can interfere with data transmission) or take many more hops than necessary by traveling up or down trees. We will present our joint protocol solution in a future paper.

### 3. Related Work

Proactive routing protocols maintain routes such as next forwarding hop in the background regardless of the communication requests. Control packets are constantly broadcast and flooded among nodes to maintain the paths and the link states between any pair of nodes even though some of paths are never used. Despite protocols such as OLSR [10] or FSR [11] try to reduce the overhead of control messages, they are still more expensive than RPL. It is unavoidable that link state information is exchanged to maintain global topology. Unlike nodes in other proactive routing protocols, RPL nodes only need the rank and ETX information to establish *downward* routes. Reactive routing protocols establish routes upon request. However, they are not the best strategy in vehicular networks because (1) broadcast storm is unavoidable upon every route request (protocols such as AODV [12] or DSR [13] would flood the network during route discovery). and (2) route establishment and breakage due to dynamism of the network make it unsuitable for some applications (i.e., streaming or audio). RPL maintains the route in the background at minimal overhead. Moreover, since RPL is designed with the intention of communicating with the infrastructure, proactive and reactive routing protocols which establish any node to any node path may seem an overkill.

Location and delay-aware cross-layer protocol (LD-CROP) [14] is a V2I multihop routing protocol. It uses beacons, originating from the AP, to establish path information. Each node along the path updates the complete path quality on the beacon with its local traffic statistics before broadcasting the beacon. A node that wishes to send data to the AP can then choose the path with the best quality or minimum delay. Path is defined by a sequence of nodes that make up the road segments toward the AP. Relay node is chosen by the cross-layer approach that favors nodes that either have packets going on the same path, are going in the same direction, or are the furthest from the forwarding node. The protocol is similar to RPL in that RPL also uses Objective Functions to determine path quality. Preferred parent selection can also take direction and distance of the node into consideration. The disadvantage of LD-CROP is the broadcast overhead as each node needs to keep track of the nodes on the path. RPL's broadcast (aka. DIO messages) overhead is fixed. RPL's broadcast assists a node in establishing its parent.

### 4. RPL and Its Implementation

In our architecture, road-side infrastructure such as WiFi access points will be roots. Vehicle OBUs will try to connect to a root or multiple roots using RPL either directly or via

other OBUs. Upon initiation, a root starts topology building by sending out directed acyclic graph (DAG) information option (DIO) messages (msgs). The DIO msgs contain information about the rank of the broadcasting node, with rank = 1 at the root. Once a node receives a DIO msg, it calculates its rank based on the rank in the received DIO msg, and the cost of reaching the node from itself. RPL defines a number of rules for parent selection based on the local quality of the links, the advertised OF, path cost, rank, and so forth. Any node that has lower rank than the node itself is considered as a candidate parent. When a node broadcasts its DIO msg, it includes all information about its rank, OF and the DAG it has joined. In this way, DIO msgs propagate down to the most distant nodes from the root to create a DODAG.

DIO msgs are emitted periodically from each node. The periodic timer  $t$  is set by the trickle timer  $I$  [15] that is bounded by the interval  $[I_{\min}, I_{\max}]$ , where  $I_{\min}$  is the minimum interval size defined in units of time (e.g., 100 milliseconds) and  $I_{\max} = I_{\min} * 2^{I_{\text{doubling}}}$ ,  $I_{\text{doubling}}$ . Suppose  $I_{\min} = 100$  ms and  $I_{\text{doubling}} = 16$ .  $I_{\max} = 100 \text{ ms} * 2^{16} = 6553.6 \text{ s}$ . being some constant (e.g., 16).  $t$  is randomly picked from  $[I/2, I]$ . Whenever  $t$  expires, a DIO msg is sent if the counter  $c$  is less than the redundancy constant  $k$ , a natural number.  $c$  is incremented whenever a node hears a DIO msg that is "consistent," that is, the node does not change its parent set, preferred parent, or rank. If a node hears a DIO msg that makes it inconsistent,  $I$  is set to  $I_{\min}$ ,  $t$  is reset to the interval of the new  $I$ , and  $c$  is reset to 0. This is so that a DIO msg can be quickly transmitted to update the tree. When  $I$  expires, if the node remains consistent, based on the transmission of DIOs from its neighbors,  $I$  is set to  $\min(2 * I, I_{\max})$ ; otherwise,  $I$  is set to  $I_{\min}$ ,  $t$  is reset to the interval of the new  $I$ , and  $c$  is reset to 0.

To avoid sending stale DIO msgs, each node has an ETX periodic timer (we use ETX as the objective function; ETX may very well be replaced with the other metrics such as hop count, latency, throughput, etc.) which probes neighboring nodes for their ETX value. If ETX does change after probes are completed, a new parent may be selected, and DIO msg is delivered to inform neighbors of the node's updated rank. Otherwise, no DIO msg is sent, and timer  $I$  is doubled upon expiry. Note that ETX probing is optional and implementation specific as it is one of the OFs. It is not in the RPL specification [1].

When a node joins the network, it may wait to receive a DIO msg or it may alternatively multicast a solicitation message called the DIS, so that other nodes hearing the DIS start sending DIO msgs, and the newly arrived node can join the DAG. Nodes also multicast Destination Advertisement Option or DAOs for 1-hop reachability and unicast reachable prefix to their parents in the DAG to advertise their addresses and prefixes. The nodes that receive these DAOs update their routing table. When no entry is available in the routing table, or for traffic to the root, a node will forward a packet up to its most preferred parent. Note that while sending packets up the DAG, a node must not forward it to a node with greater rank to prevent a loop in the routing path. In case no parent is available, the node can forward the data to a sibling (node with same rank). In this way, for P2P routing, the packet

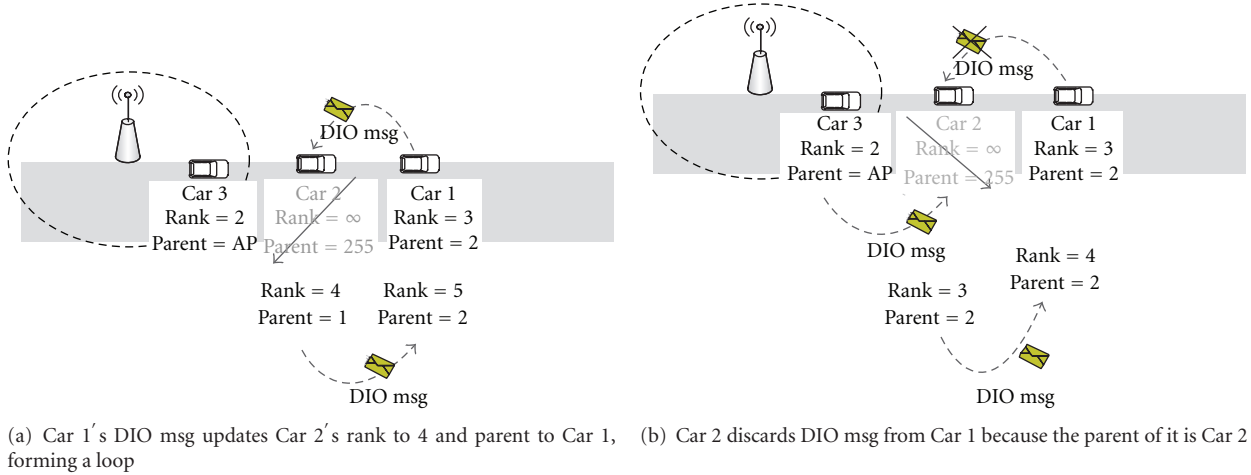


FIGURE 1: Loop avoidance and detection.

goes from the source up to a common ancestor of the source and destination in the DAG, and then it travels down to the destination.

## 5. RPL Modifications for Mobility

This section describes our modifications of RPL for VANET. Since RPL was originally designed for static sensor networks, RPL rank does not update in a timely fashion to reflect frequent topology changes. Slow response to topology changes, as reflected in slow rank update, results in a suboptimal path to the destination or a destination unreachable error. Moreover, a node may lose connection to the road-side infrastructure and then connect to a node in its neighborhood as its parent. This may result in a loop. We describe these problems and propose their solutions below. The first is implementation specific, and the rest is related to the RPL specs.

**5.1. Immediate ETX Probing for a New Neighbor.** When a node discovers a new neighbor, the node schedules PING request messages for its neighbor's ETX value. Based on the ETX value of its new neighbor and its parent, the node determines whether to change its parent. Scheduling of ETX probing may delay the selection of a preferred parent if the new neighbor has a lower ETX value than the existing parent. This problem is especially acute in highly mobile networks as fast moving nodes quickly change their rank. Instead of scheduling ETX probing to start at some future time, we fire off ETX probing immediately. That way, the new neighbor can be considered for preferred parent selection in a timely fashion.

**5.2. Loop Avoidance and Detection.** A node becomes momentarily disconnected to the AP as it moves away from the AP's range. The node tries to reestablish connection to the AP through its neighbors. The node may select a neighbor whose parent was the node itself as its parent since the neighbor's rank is less than the node's current rank (which is at infinity). As shown in Figure 1(a), Node 2 just went out of the AP's

range and has a rank of infinity. When it receives a DIO msg from Node 1, it updates its rank to 4. DIO msg from Node 3 may be lost during transmission, thus failing to update Node 2's parent to Node 3. When Node 2 forwards packets to its parent Node 1, Node 1 will then forward packets to its parent Node 2, forming a loop. Before Node 2 has a chance to re-adjust its parent to Node 3, Node 3 may already get out of the range of the AP. Node 2 will discard the next DIO msg from Node 3 since Node 3's rank is infinity. Node 3 may update its parent to Node 2, further enlarging the loop.

We present the fix by stamping the DIO msg with its parent's ID. When a node receives a DIO msg from its neighbor, it will discard the DIO msg if the parent ID in the DIO msg matches with the node's ID. In the same example, in Figure 1(b), when Node 2 receives a DIO msg from Node 1, since the DIO msg indicates Node 2 was Node 1's parent, Node 2 will drop the DIO msg from Node 1, preventing a loop from happening. The method also detects a loop and breaks the loop.

**5.3. Immediate DIOs upon New Parent Election.** DIOs allow a node, say A, to calculate its rank in relation to the node where A receives the DIO msg from. Rank ordering establishes the parent-children relationship and the tree structure. Upon electing a node as its preferred parent, A sends a DIO msg immediately to notify its new rank. This deviates from the RFC specification in that DIO msg is not sent subject to the expiry of the trickle timer. Although trickle timer does increase DIO msg frequency as topology changes, we adopt an even more aggressive approach. Consequently, the tree can be refreshed to reflect the dynamic topology quickly.

**5.4. Immediate DAOs upon New Parent Election.** DAOs help a node set-up downward routes to its children. As soon as a node, say A, determines who its parent is, a DAO msg is sent immediately to notify its parent routes to A's children. This contrasts the RFC [1] which states that "[a node] should delay sending the DAO message in order to aggregate DAO information from other nodes for which it is a DAO parent."



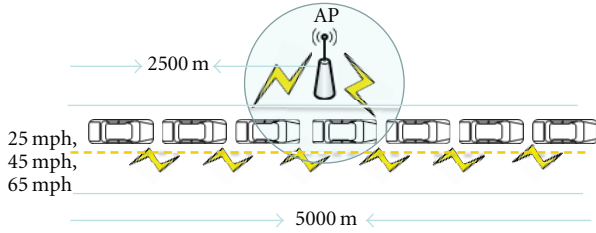


FIGURE 2: Network setup.

However, in a mobile environment such as VANETs, the delay degrades packet delivery as routes are not updated in a timely fashion. The tradeoff for more frequent DAOs, therefore more overhead, improves packet delivery. Intuitively, aggregation makes sense in the sensor networks because nodes hardly move; therefore, routes need not be updated too frequently. Low traffic volume from DAO exchange saves a node's duty cycle and battery power.

## 6. Performance Evaluation

**6.1. Evaluation Setup.** We evaluate modified RPL per Section 5 using Qualnet 4.5 network simulator with 10 nodes traversing a straight line of 5000 m in Figure 2. Nodes are spaced out 250 m apart as they are caravanning from one side of the road to the other side. Transmission power for IEEE 802.11a (We do not have a full implementation of 802.11p; however, we take note of the fact that 802.11a is equivalent to 802.11p without QoS feature.) broadcast is adjusted so that 250 m is the approximate radio range. The access point (AP), which serves as the root in the RPL network, is placed in the middle of the line at 2500 m. Request packets are sent from any car in the caravan to the AP and reply that packets are sent from the AP to the car which makes the requests. The start time of packet transmission is the time when Car 1 gets connected to the AP. Depending on the speed of the caravan, the start time can vary. More specifically, the caravan traveling at 25 mph has a later start time than the caravan traveling at 65 mph because 65 mph caravan will reach the AP faster. Speed is varied from 25 mph, 45 mph, to 65 mph. See Table 1 for details of the simulation parameters.

We first verify modified RPL's correctness by observing cars' rank change with respect to time and rank duration with respect to speed. We then evaluate modified RPL's packet delivery ratio (PDR), overhead, average throughput, and average delay with respect to ETX probe interval. Packet delivery ratio is defined as the number of *reply* packets received over the number of *request* packets sent, all at the sending node. This allows us to evaluate the performance of the bidirectional (upload and download) path. Overhead is the number of ETX packets sent. Throughput is the replies received per second. Delay is the time between the time a reply is sent and a reply is received.

We then evaluate the impact of DIO period and speed on modified RPL's PDR, overhead, delay, and throughput. To illustrate the impact of DIO period, we disable DIO trickle timer. We then compare various DIO fixed timers (2 s, 4 s,

TABLE 1: Simulation parameters.

Parameter	Value
Network simulator	Qualnet 4.5
MAC protocol	802.11a
Transmission rate	24 Mbps
Transmission power	-80 dBm
Transmission range	250 m
Propagation	Two-Ray
Application traffic	FTP
Vehicle speed	25 mph, 45 mph, or 65 mph/run

6 s, 8 s, and 10 s) with modified RPL with trickle timer enabled. The evaluation is to show the impact of speed and DIO msg period on tree formation and to show that DIO trickle timer outperforms DIO fixed timer with reasonable overhead. Lastly, we compare the PDR, throughput, and delay of RPL and modified RPL after changes mentioned in Section 5 are applied.

## 6.2. Evaluation Results

**6.2.1. Rank Change and Duration with Respect to Time and Speed.** Figures 3(a) and 3(b) show the rank versus time graph for Car 1, 5, and 10 traveling from left to right of the road at 25 mph and 65 mph, respectively, DIO period = 2 s and ETX period = 5 s. We can visualize that the modified RPL is able to update the ranks for each car as it moves past the AP and beyond. Take 25 mph, for example, shown in Figure 3(a), when the lead car (Car 1) establishes connection at about 199 s, all the other cars also establish connection to the AP through the car(s) before it. The time each vehicle stays at each rank is about the same, right around 22 s, which is the time it takes for a car to cover the distance between rank changes. In an ideal scenario, the rank duration time should be the same for each car. Since Car 1 will approach the AP first, its rank increases from rank 2 to rank 11; whereas trailing Car 10's rank decreases from rank 11 to rank 2. Rank duration varies as the speed increases. At 65 mph, shown in Figure 3(b), rank duration shrinks down to about 9 s. Rank duration varies as DIO period increases and the speed increases. This is expected because high speed prompts for fast rank change or short rank duration. If DIO period is too long, RPL cannot keep up with the rank change and will skip a rank. Notice that as cars pass the AP and go beyond it, there is a period of disconnection to the AP (e.g., Car 1's steps are not connected by vertical lines). This is due to modified RPL looking for a parent to connect to the AP. For Car 10, this is not the case because its parent is always Car 9 until it goes beyond the range of the AP.

**6.2.2. Impact of ETX Probe Period.** ETX probes are sent periodically by a node to probe its parent's ETX value. Based on the ETX value of its parent, a node can either stay attached to its parent or a new parent. Figure 4(a) shows the PDR of Car 1 as ETX probe interval increases (we only show Car 1's as

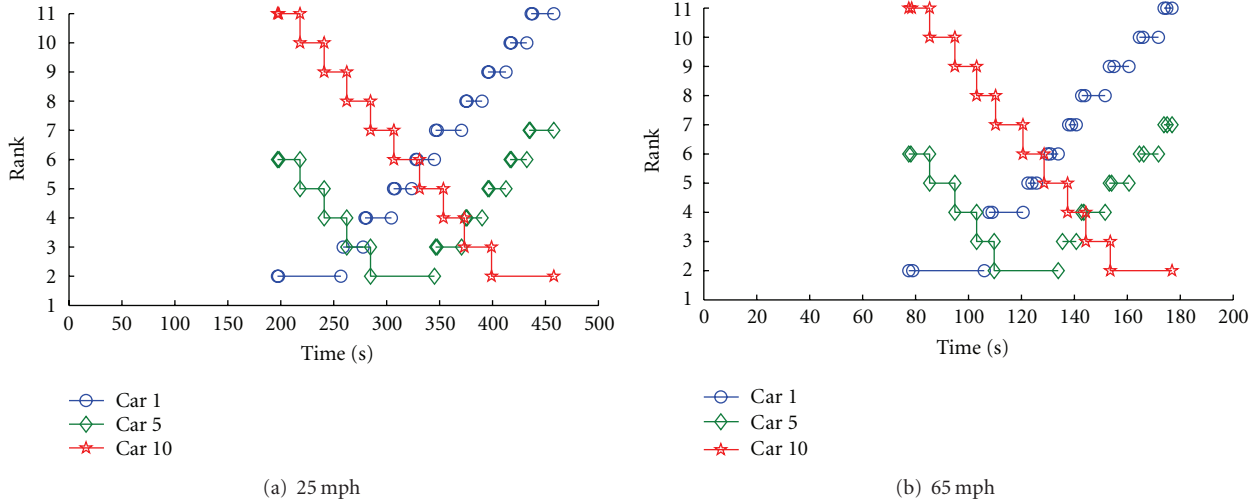


FIGURE 3: Rank versus time for 25 mph and 65 mph. DIO period = 2 s, ETX period = 5 s.

cars in the caravan that exhibit the similar pattern). As ETX probe period increases, PDR drops. This is due to delayed response to changing parenthood as nodes move from the left of the AP to the right of the AP. Parents are no longer cars ahead of nodes as they move past the AP.

Frequent ETX probe comes at a cost. Figure 4(a) shows the number of ETX probes with respect to the ETX probe period. As the probe period increases, the number of ETX probes decreases. Since PDR does not degrade as much for ETX probe period 2 s, 4 s, and 6 s, it might indicate that, for the most part, nodes' parent does not change. This is true as the caravan maintains the same speed and parenthood only changes when nodes move from the left side of the AP to the right. 6 s probe interval may be enough time to capture the topology and determine a node's parent given the particular transmission rate in this particular scenario. An adaptive timer approach for ETX probe, such as trickle timer [15], can be used so as to reduce the overhead. More specifically, probe interval adjusts based on the ETX value and the decision of the parent selection thereof. If the parent remains the same, the probe interval can be prolonged; otherwise, it can be shrunk. To make sure ETX probe reacts in a timely fashion, an ETX probe will be sent immediately before the next scheduled ETX probe.

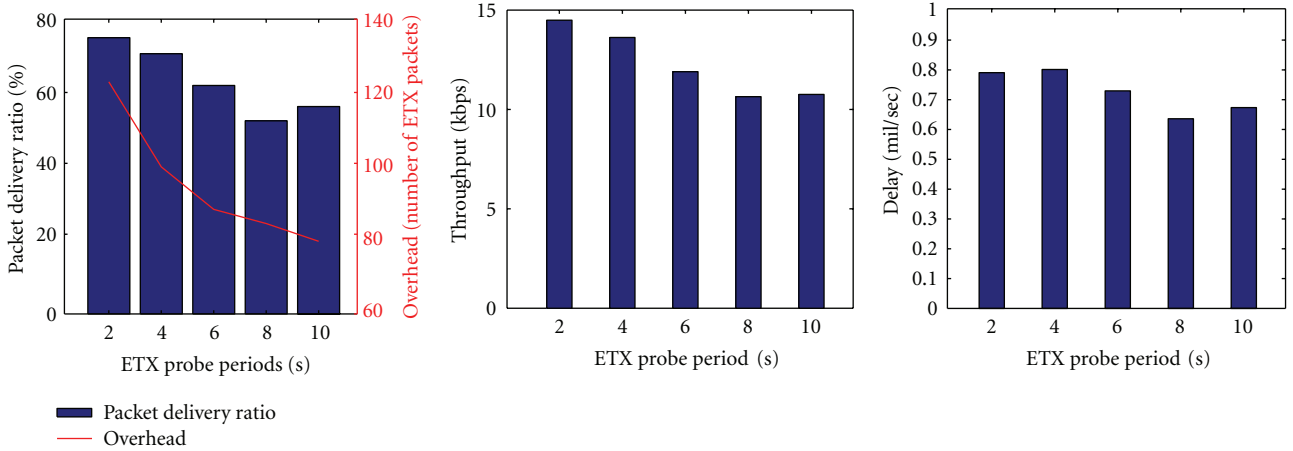
Figure 4(b) shows that as ETX probe period increases, throughput decreases. This is expected as the packet delivery quality decreases with increasing ETX probe period. The delay decreases as the result that packets that are delivered successfully travel fewer hops, shown in Figure 4(c).

**6.2.3. Impact of DIO Msg Period and Speed.** DIO msgs allow a node to quickly identify its rank in the tree. As DIO msgs become more frequent, the tree can be updated quickly to reflect the topology change. In our evaluation, we first disable trickle timer and show the impact of using a fixed timer on RPL's performance under mobility. Then we enable trickle timer and compare its performance with the fixed timer RPL.

Figures 5(a), 5(b), and 5(c) show the PDR of the modified RPL with respect to DIO msg period for Car 1, Car 5, and Car 10, respectively. As DIO msg period increases, PDR drops. Moreover, PDR drops as speed increases for a given DIO msg period. The graph shows the effect of DIO msg period and speed on PDR. At 45 mph, DIO period = 8 s, the PDR, overhead, average throughput, and average delay is lower than at 65 mph as a scenario-dependent case, not warranting a general downward trend from the impact of the speed (Further investigation of the trace shows that it just so happens that when the last two cars (9 and 10) are going past the AP, the last car's DIO msgs are rejected by the car ahead since both cars are in the range of the AP. When Car 9 becomes disconnected beyond the AP, it does not connect to Car 10 right away (since DIO period is so long). But when DIO msg from Car 10 comes, it has already gotten outside the range of the AP. Thus, there is no connection at all for all the cars up to the end of simulation).

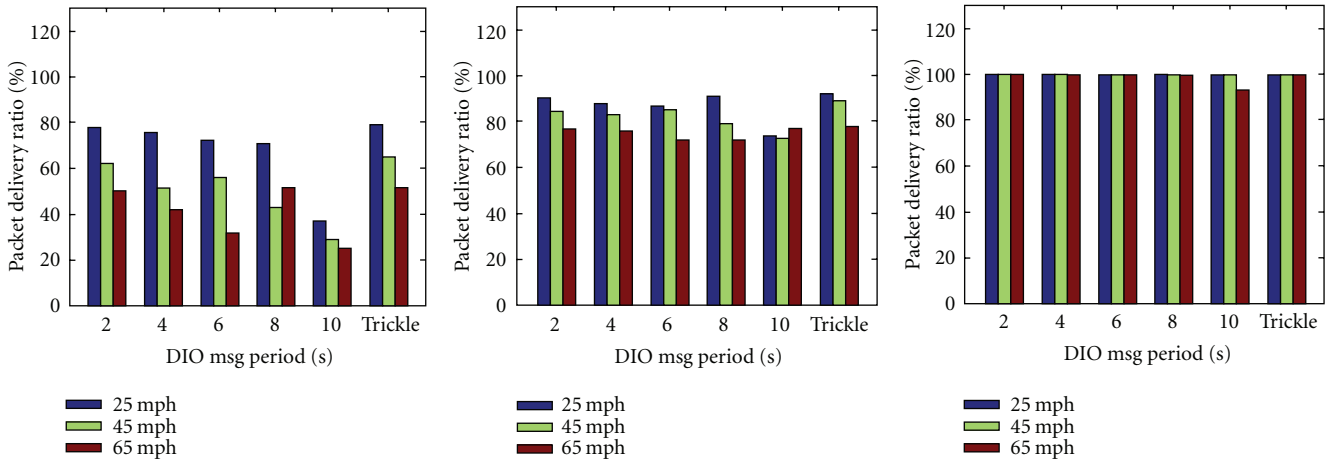
Figures 6(a), 6(b) and, 6(c) show that DIO msg overhead is proportional to the DIO msg period, that is, the lower the DIO msg period, the higher the DIO msg overhead. Moreover, DIO msg overhead increases with increasing speed. This is due to frequent topology changes that prompt for frequent updates by the DIO msg exchange. While high frequency of DIO msg period does improve PDR, there is a diminishing return (e.g., comparing PDR for car 1 at 45 mph, DIO = 2 s and DIO = 4 s, PDR does not differ much yet the overhead at DIO = 2 s is way higher than DIO = 4 s). This shows the advantage of using adaptive timer. Trickle timer is able to adjust DIO msg period based on the topology and yields the equivalent performance as one where DIO period is set to 2 s, yet at the same time controls the overhead.

Figures 7(a), 7(b), and 7(c) show the throughput of the modified RPL with respect to DIO msg period for Car 1, Car 5, and Car 10, respectively. Throughput increases drastically at DIO msg period = 10 s, more apparent for Car 1. A close look at the delay figures (Figures 8(a), 8(b), and 8(c)) indicates that delay is way smaller at DIO msg period = 10 s than



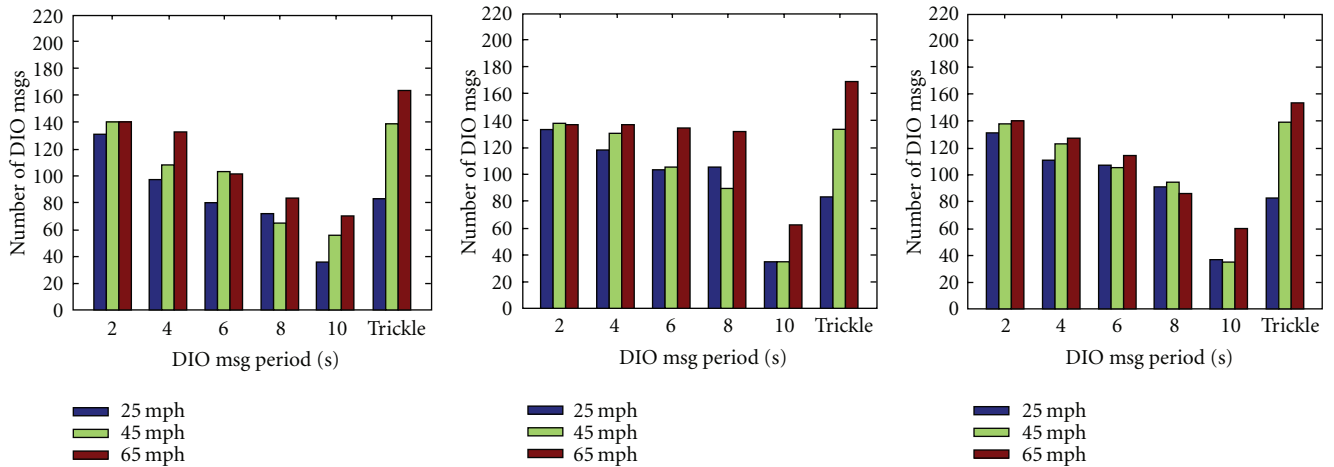
(a) PDR and overhead versus ETX probe period (b) throughput versus ETX probe period (c) delay versus ETX probe period

FIGURE 4: Impact of ETX probe period on Car 1's PDR, overhead, throughput, and delay. Speed = 25 mph, DIO period = 2 s.



(a) Car 1's PDR versus DIO msg period (b) Car 5's PDR versus DIO msg period (c) Car 10's PDR versus DIO msg period

FIGURE 5: Impact of DIO period and speed on the PDR.



(a) Car 1's overhead versus DIO msg period (b) Car 5's overhead versus DIO msg period (c) Car 10's overhead versus DIO msg period

FIGURE 6: Impact of DIO msg period and speed on the overhead.

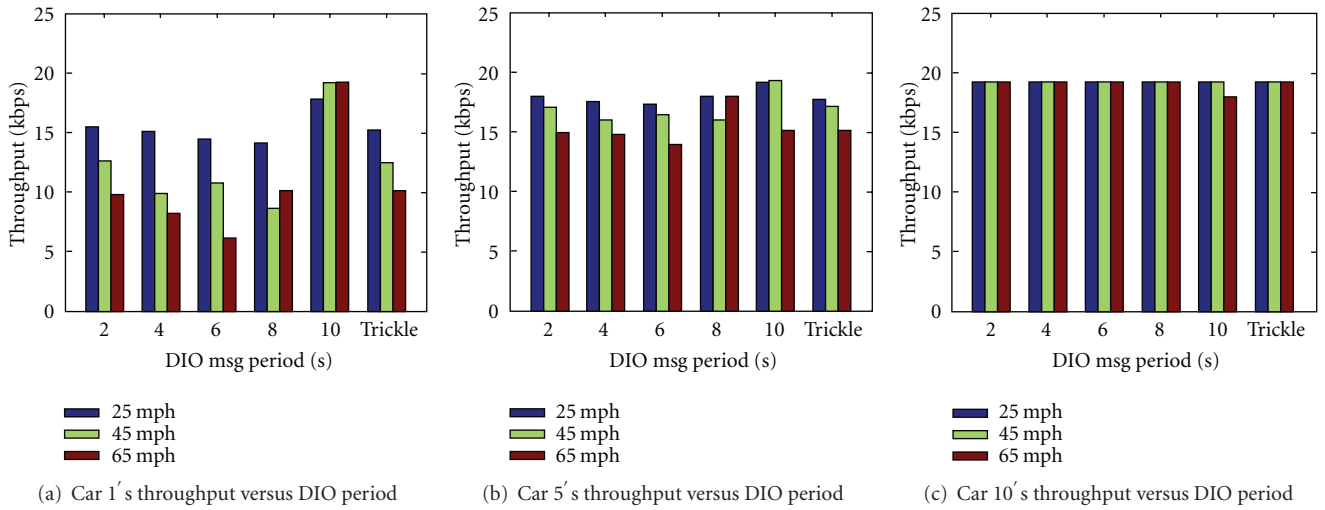


FIGURE 7: Impact of DIO period and speed on the throughput.

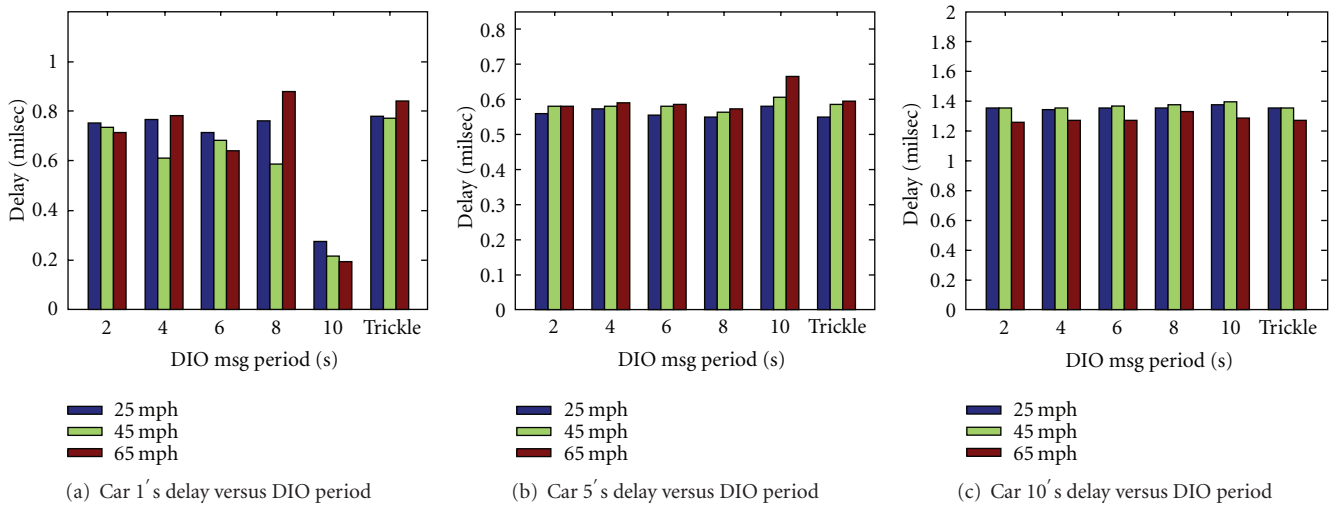


FIGURE 8: Impact of DIO period and speed on the delay.

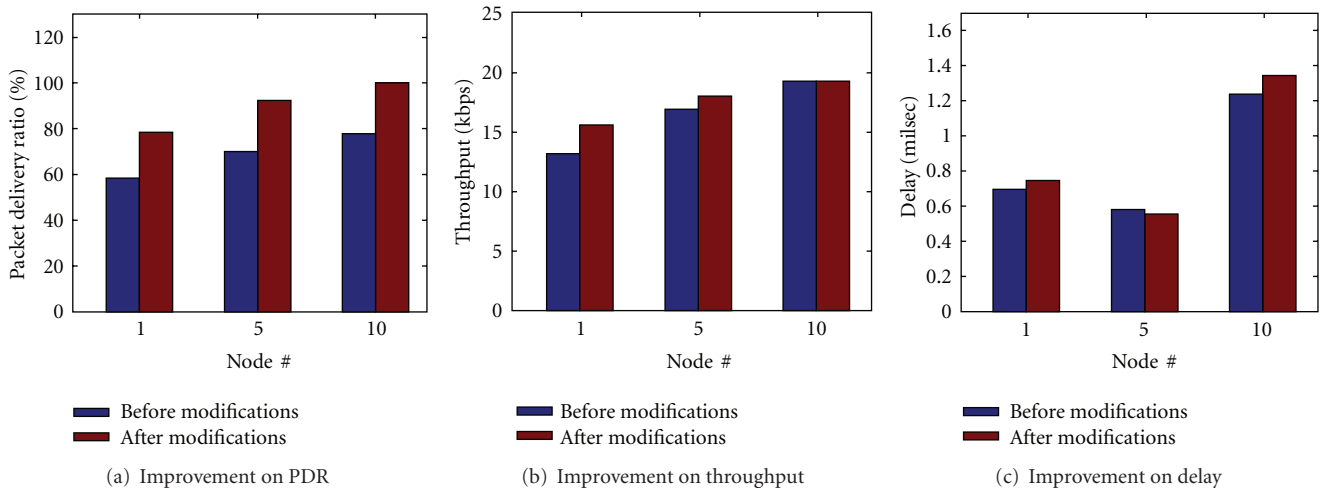


FIGURE 9: Improvement from modifications.



DIO msg period = 2 s. This is because most of the packets that reach the AP are one hop away as Car 1 drives within the range of the AP. When the car is far away from the AP, packets inevitably will travel multiple hops to reach the AP. However, route is not set up quickly enough under infrequent DIO msg period that these packets are dropped. Therefore, the delay here only reflects the packets which are one hop away. Their delay is shorter. This unfairly bumps up the throughput.

Note that PDR increases as Car # increases. This is due to that cars further back in the caravan do not experience disconnection that happens when passing the AP as much as the cars in the front of the caravan.

**6.2.4. Modified RPL Improvement.** Figure 9(a) shows the PDR before and after changes mentioned in Section 5 for speed = 25 mph, ETX = 2 s. Note that both are RPL with trickle timer enabled. Clearly, these changes show much improvement over RPL for static networks. We have fine-tuned RPL so that it can work for highly dynamic vehicular network environment.

Figure 9(b) shows the throughput before and after the modifications. Improvement in throughput can clearly be seen. In Figure 9(c), the average delay before modifications tend to be shorter than the delay after modifications (seen in Car 1 and Car 10) because of disconnection of links before modifications. Packets that are received are when the car is one or two hops away from the AP. Despite the longer delay, modified RPL still performs better in throughput because of more number of packets transmitted successfully.

## 7. Conclusion and Future Work

In this paper, we have studied the impact of DIO msg and ETX probe period and vehicle's speed on RPL in terms of packet delivery ratio (PDR) and overhead. Although the frequency of DIO msg and ETX probe can help RPL in improving PDR, they do increase the overhead. Diminishing return of increasing DIO msg/ETX probe frequency with increasing overhead and similar performance with reasonable overhead from the trickle timer approach shows trickle timer's advantage and value. We then fine tuned RPL based on our finding of protocol's response to mobility and obtained performance improvement over the existing RPL for its use in the vehicular environment. Future work includes an adaptive timer for the ETX probe similar to trickle timer used to send DIO msg and evaluation results for realistic traffic trace with random vehicle speeds and positioning. We have ported RPL on Linux and verified its operation in a 3-node testbed. We plan to deploy RPL on the vehicular testbed at Cisco and evaluate further its performance.

## Acknowledgments

The authors would like to thank Matilde Durvy for the Contiki implementation of RPL, of which we used as a template for our own implementation.

## References

- [1] T. Winter, P. Thubert, A. Brandt et al., RPL: IPv6 Routing Protocol for Low power and Lossy Networks, draft-ietf-roll-rpl-19, Internet Engineering Task Force, 2011, <http://tools.ietf.org/html/draft-ietf-roll-rpl-19>.
- [2] V. Devarapalli, R. Wakikawa, A. Petrescu, and P. Thubert, Network Mobility (NEMO) Basic Support Protocol, RFC 3963, Internet Engineering Task Force, 2005, <http://www.ietf.org/rfc/rfc3963.txt>.
- [3] B. McCarthy, M. Jakeman, C. Edwards, and P. Thubert, "Protocols to efficiently support nested NEMO (NEMO+)," in *Proceedings of the 3rd International Workshop on Mobility in the Evolving Internet Architecture (MobiArch '08)*, pp. 43–48, Association for Computing Machinery, Seattle, Wash, USA, August 2008.
- [4] J. Tripathi, J. C. de Oliveira, and J. P. Vasseur, "A performance evaluation study of RPL: routing protocol for low power and lossy networks," in *Proceedings of the 44th Annual Conference on Information Sciences and Systems (CISS '10)*, pp. 1–6, March 2010.
- [5] T. Clausen and U. Herberg, "Multipoint-to-point and broadcast in RPL," in *Proceedings of the 13th International Conference on Network-Based Information Systems, NBI 2010*, pp. 493–498, September 2010.
- [6] J. Vasseur and D. Culler, "Routing over low power and lossy networks," 2009, <https://datatracker.ietf.org/wg/roll/charter/>.
- [7] J. Vasseur, M. Kim, K. Pister, N. Dejean, and D. Barthel, "Routing metrics used for path calculation in low power and lossy networks," draft-ietf-roll-routing-metrics-10, Internet Engineering Task Force, Apr. 2011, <http://tools.ietf.org/html/draft-ietf-roll-routing-metrics-19>.
- [8] S. Y. Wang, C. C. Lin, Y. W. Hwang, K. C. Tao, and C. L. Chou, "A practical routing protocol for vehicle-formed mobile ad hoc networks on the roads," in *Proceedings of the 8th International IEEE Conference on Intelligent Transportation Systems*, pp. 161–166, September 2005.
- [9] K. C. Lee, U. Lee, and M. Gerla, "Advances in vehicular ad-hoc networks: developments and challenges," in *Survey of Routing Protocols in Vehicular Ad Hoc Networks*, pp. 149–170, IGI Global, Hershey, Pa, USA, 2010.
- [10] P. Jacquet, P. Muhlethaler, T. Clausen, A. Laouiti, A. Qayyum, and L. Viennot, "Optimized link state routing protocol for ad hoc networks," in *Proceedings of the IEEE International Multi Topic Conference, Technology for the 21st Century (IEEE INMIC '01)*, pp. 62–68, 2001.
- [11] G. Pei, M. Gerla, and T.-W. Chen, "Fisheye state routing: a routing scheme for ad hoc wireless networks," in *Proceedings of the IEEE International Conference on Communications (ICC '00)*, vol. 1, pp. 70–74, 2000.
- [12] C. E. Perkins and E. M. Royer, "Ad-hoc on-demand distance vector routing," in *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications (WMCSA '99)*, pp. 90–100, 1999.
- [13] D. B. Johnson and D. A. Maltz, "Dynamic source routing in ad hoc wireless networks," in *Mobile Computing*, T. Imielinski and H. F. Korth, Eds., vol. 353 of *The Kluwer International Series in Engineering and Computer Science*, pp. 153–181, Springer, New York, NY, USA, 1996.
- [14] B. Jarupan and E. Ekici, "Location- and delay-aware cross-layer communication in V2I multihop vehicular networks," *IEEE Communications Magazine*, vol. 47, no. 11, Article ID 5307474, pp. 112–118, 2009.

- [15] P. Levis, T. Clausen, J. Hui, O. Gnawali, and J. Ko, The Trickle Algorithm, RFC 6206, Internet Engineering Task Force, 2011 <http://tools.ietf.org/html/rfc6206> .



**Hindawi**

Submit your manuscripts at  
<http://www.hindawi.com>

