RPL under Mobility

Kevin C. Lee*, Raghuram Sudhaakar*, Lillian Dai*, Sateesh Addepalli*, Mario Gerla†

Cisco Systems, Inc.*
170 West Tasman Dr.
San Jose, CA 95134
{kevinl2,rsudhaak,lildai,sateeshk}@cisco.com

University of California, Los Angeles†
Department of Computer Science
Los Angeles, CA 90095
gerla@cs.ucla.edu

***Abstract*—This paper focuses on routing for vehicles getting access to infrastructure either directly or via multiple hops though 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.**

I. Introduction

Vehicular networking (VANET) technologies enable a wide range of communication applications. In light of recent changes in wireless service provider deployments and pricing structure and the tremendous growth in the mobile wireless sector, we believe that the time is ripe for exploring the interplay between infrastructure and ad hoc wireless networking for consumer vehicles (e.g., sedans, vans, performance cars, etc.) and commercial vehicles (e.g., buses, trains, boats, etc.).

We envision that future consumer and commercial vehicles will be equipped with a mobile onboard unit (OBU) with multiple wireless interfaces. 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 fail tolerance.

Initially, when few vehicles on the road are equipped with OBUs, those that do will likely access available infrastructure such as 3G, 4G, and WiFi networks. As the penetration of OBUs increases in the future, vehicle-to-vehicle networking may become important for two reasons: cost savings and performance enhancement for location applications. While 3G and 4G networks are pervasive, service providers are increasingly moving towards a tiered pricing model for data consumption. As it becomes expensive to use 3G and 4G services, users will have a strong economic incentive to utilize free services or other alternative low cost services whenever possible. In addition, major wireless service providers such as AT&T and Verizon have recently announced rollout of their own WiFi networks. These trends drive the vehicular networking architecture to one of opportunistic offloading of communication traffic from 3G and 4G links to free WiFi links, possibly via multiple hops. 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 traffic is inherently suited for ad hoc wireless networking.

Within the context of vehicular networking with both infrastructure and ad hoc networking capabilities, questions remain as to the appropriate routing protocol and how the two different types of networks can complement one another for cost reduction and performance enhancements. In this paper, we focus on the former routing question for vehicles getting access to road-side infrastructure either directly or via multiple hops through wirelessly connected OBUs. Each OBU is thus a router with multiple wireless interfaces, some for internal wireless connection to in-vehicle devices (e.g., laptops, smartphones, etc.), some to external infrastructure networks such as 3G, 4G, and WiFi, and some to other OBUs in an ad hoc fashion.

In contrast to routing protocols such as NEMO [1], NEMO+ [2], and MANEMO [3] which are designed for Mobile IP (MIP), in this paper we focus on a protocol that is complementary to global mobility protocols such as the Locator/ID Separation Protocol (LISP) [4], thereby bypassing numerous inefficiencies inherent in marrying edge routing with global MIP mobility [2]. The protocol we study is Routing Protocol for Low power and Lossy Networks (RPL) [7] which was 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 NEMO, NEMO+, and RPL. Many design elements from RPL are transferable to the vehicular environment as we will discuss in detail. 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.

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 in VANET under several routing metrics. Past evaluation of RPL in [5], [6] have focused on static sensor networks. To the best of our knowledge, this paper is the first to evaluate RPL and RPL tuning for VANETs.

The rest of the paper is organized as follows. Section II describes the motivation for RPL among the already crowded routing protocol space. Section III describes RPL and implementation details. Section IV discusses the modifications to adapt RPL for VANETs. Section V evaluates RPL under several tunable parameters in VANETs. Section VI concludes the paper and presents the future work.

II. 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[1] (i.e., at a single root with no outgoing edges). It is actively maintained by the IETF ROLL Working Group [7]. 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. 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 [8]. 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.

On its own, RPL simply constructs a multihop DODAG topology to reach road-side infrastructure, but does not handle seamless mobility when vehicles move from the coverage area of one infrastructure device to another. However, with our vehicular networking architecture, an OBU and an in-network appliance, such as a controller or gateway, can implement a global mobility solution such as LISP [4]. For example, the OBU and the in-network appliance can act as ingress and egress routers in LISP parlance. In general, any similar mobility solution that hides the underlying mobility event from end-user devices between the OBU and the in-network appliance can be used with RPL.

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:

- 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 [9] 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.
- Unlike any other link-state 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 [10] where the root is at the APs.
- RPL may be coupled with another multihop routing protocol to improve routing efficiency for 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 multi-hop to infrastructure to peer-to-peer routing protocol solution in a future paper.

III. 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, etc. 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. 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 [11] 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 $I_{max} = I_{min} * 2^{I_{doubling}}$, $I_{doubling}$[2] 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 every time a DIO msg, which is received from its neighbors,

[1]It is also known as sink or root. For the rest of the paper, we will use root.

[2]Suppose $I_{min} = 100ms$ and $I_{doubling} = 16$. $I_{max} = 100ms * 2^{16} = 6553.6s$.

does not change the node's rank. When I expires, if the node's rank did not change from the transmission of DIOs from its neighbors, I is set to $min(2*I, I_{max})$; otherwise, if the node's rank is changed, I is set to I_{min}. t is then reset to the interval of the new I. Note that $t < I$ and t is never reset unless I expires.

To avoid sending stale DIO msgs, each node has an ETX periodic timer[3] 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 t is reset to its old value upon expiry (rather than doubled based on the RFC). We disabled trickle timer in the implementation to study the effects of DIO periods on RPL's performance (See Section IV).

IV. 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 sub-optimal 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 two problems and propose their solutions below.

A. Disabling DIO Trickle Timer

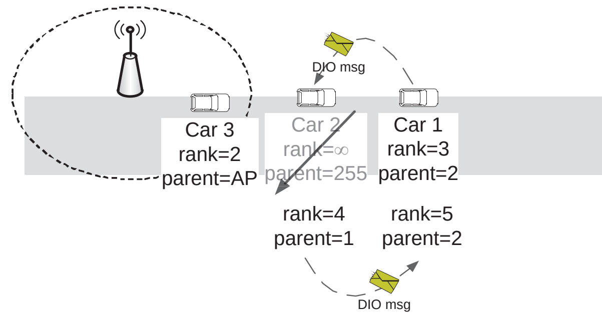
In our implementation, the trickle timer is disabled for the reason of studying the impact of using a fixed timer on RPL's performance under mobility. Since the topology changes frequently for mobile networks, trickle timer, which is designed for static networks, may not be suitable. Node's rank may not change from one DIO msg to the next DIO msg during one t period under low speed, doubling t's value may cause the node to send its DIO msg later. This may cause the other nodes to double their timer because there is no DIO msg to suggest a rank change. In the meantime, the current node may send a stale DIO msg and double its timer again since it does not receive any DIO msg from its neighbors. The situation worsens as nodes exchange stale DIO msgs and continue to increase their t timer. Outdated DIO msgs can be exchanged amongst nodes in a prolonged period that ranks of nodes no longer reflect the tree structure of the current network topology.

We realize that the design of the trickle timer is for LLNs where network topology does not change often. We therefore disable trickle timer and study various fundamental factors (DIO periods, ETX periods, and ETX) that impact RPL performance. Note that a holistic solution is to dynamically tune the trickle timer for static and mobile scenarios.

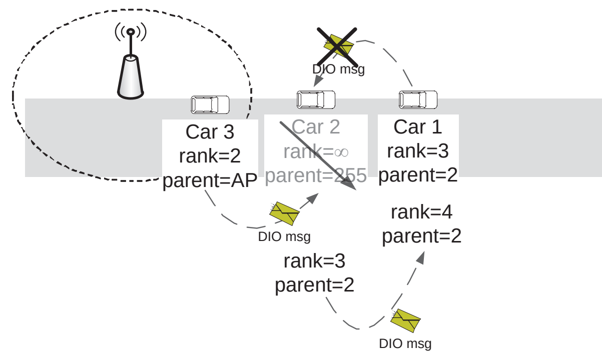
B. Immediate ETX Probing for a New Neighbor

When a node discovers a new neighbor, the node schedules PING request messages for its neighbor's ETX value. Based

[3]We use ETX as the objective function; ETX may very well be replaced with the other metrics such as hop count, latency, throughput, etc.



(a) Car 1's DIO msg updates Car 2's rank to 4 and parent to Car 1, forming a loop



(b) Car 2 discards DIO msg from Car 1 because the parent of it is Car 2

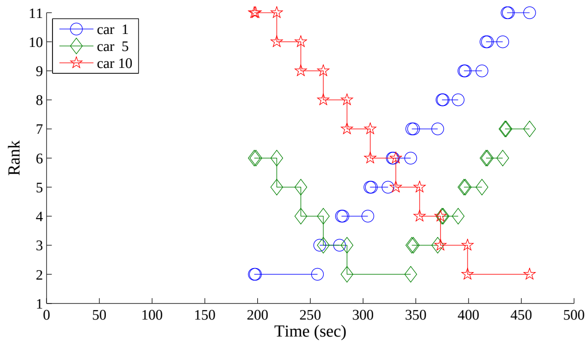
Fig. 1. Loop Avoidance and Detection.

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.

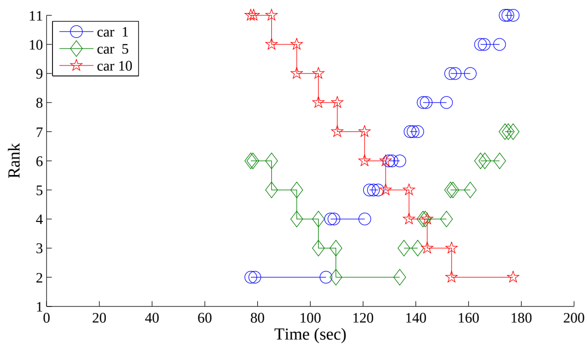
C. Loop Avoidance and Detection

A node becomes momentarily disconnected to the AP as it moves away from the AP's range. The node tries to re-establish 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 1 has a chance to readjust 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



(a) 25mph



(b) 65mph

Fig. 2. Rank vs. Time for 25mph and 65mph. DIO period = 2s, ETX period = 5s

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.

V. Performance Evaluation

A. Evaluation Setup

We evaluate modified RPL per Section IV using Qualnet 4.5 network simulator with 10 nodes traversing a straight line of 5000m. Nodes are spaced out 250m apart as they are caravanning from one side of the road to the other side. Transmission power for IEEE 802.11b broadcast is adjusted so that 250m 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 2500m. We evaluate modified RPL connectivity duration and DIO msg overhead by varying the speed of the vehicles and the period of DIO msgs. Speed is varied from 25mph, 45mph, to 65mph and DIO msg period is varied from 2s to 10s, in an increment of 2s.

B. Evaluation Results

Figures 2(a) and 2(b) show the Rank vs. Time graph for Car 1, 5, and 10 at 25mph and 65mph, respectively, DIO period = 2s and ETX period = 5s (due to space constraint, but a general pattern can still be obtained). The figures allow us to visualize that the modified RPL is able to update the ranks for each car as it moves past the AP and beyond. Take 25mph for example, shown in Figure 2(a), when the lead car (Car 1) establishes connection at about 199s, 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 22s, 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 65mph, shown in Figure 2(b), rank duration shrinks down to about 9s. 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.

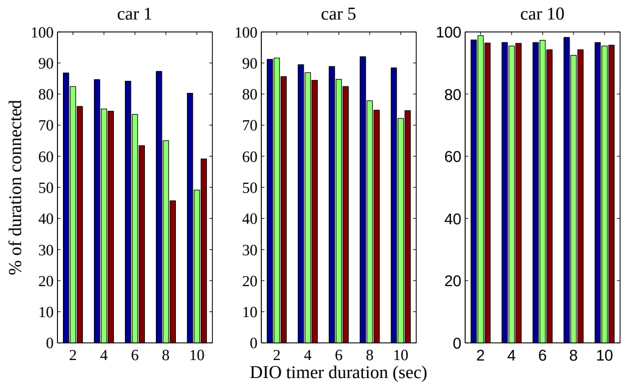
Figure 3(a) shows the percentage connectivity duration (PCD) of cars 1, 5, and 10 based on speed and DIO msg period (only these 3 representative cars are shown due to space constraint). Connectivity duration (CD) is defined as the total time a car, using modified RPL, is connected to the AP during its trip. Theoretical connectivity duration (TCD) is defined as the total time a car, using an ideal protocol, is connected to the AP during its trip. PCD is the ratio between CD and TCD. This is the metric that measures the performance of the modified RPL. There is a decrease in PCD as speed increases for Car 1 and Car 5, all DIO periods except DIO period = 10s[4].

PCD increases as Car # increases. This is due to 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.

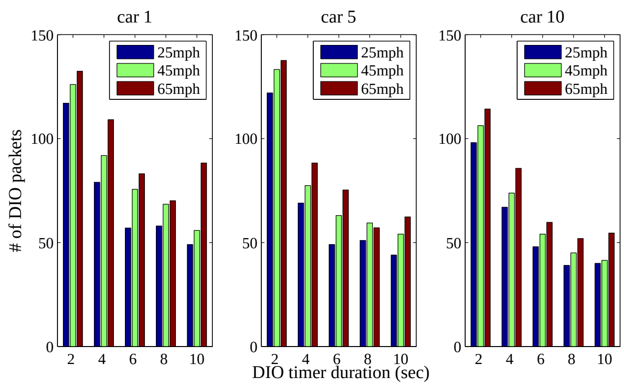
Figure 3(b) shows the DIO msg overhead normalized by the simulation time for each different DIO period and different speed, corresponding closely with % connectivity duration in Figure 3(a). DIO msg overhead is proportional to the DIO period, i.e., the lower the DIO period, the higher the DIO msg overhead. 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 DIO period does improve PCD, there is a diminishing return (e.g., comparing PCD for car 5 at 45mph, DIO = 2 and DIO = 4, PCD does not differ much yet the overhead at DIO = 2s is way higher than DIO = 4s). This shows that DIO period fine-tuning should consider the tradeoff between PCD and overhead gain.

Figure 4 shows the standard deviation of percentage connectivity duration across different DIO periods for each car at all

[4]At 45mph, DIO period = 10s, 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.



(a) % of connectivity duration (PCD) in which a car is connected to the AP, based on different speeds.



(b) Overhead for each DIO period, based on different speeds.

Fig. 3. % connectivity time and overhead graph, ETX period = 5s

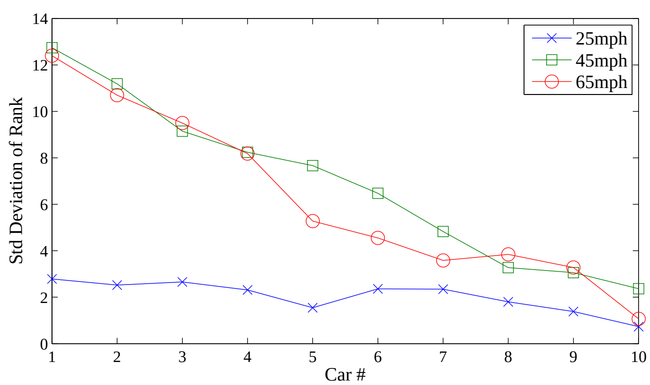


Fig. 4. Standard deviation of PCD with different DIO periods for ETX=5s.

three speeds. The graph shows that speed plays an important role in the variation of connectivity duration up to a certain speed (e.g., 45mph. Beyond 45mph, std of 45mph and 65mph coincide). The higher the speed, the higher this variation and vice versa. Since the faster the vehicle, the more frequently the topology changes (as shown in Figure 2(b) where the rank duration for 65mph is shorter than for 25mph for each vehicle). This results in wide connectivity duration fluctuation with different DIO periods. This indicates the importance of adaptively adjusting DIO period in order to narrow the PCD variation at high speed. The downward trend in std as Car # increases is due to a period of disconnection as cars move past the AP (explained in Figure 3(a)).

VI. Conclusion and Future Work

In this paper, we have studied various parameters (e.g., various timers and speeds) on modified RPL in terms of connectivity duration and overhead as RPL in its original form does not adapt to the high speed vehicular environment. High speed decreases connectivity duration since periodic updates cannot keep up with the topology change. Trickle timer may not be suitable as it works best for sensor networks. Lastly, loops are observed and can be avoided and detected with simple hints in DIO msgs. Future work includes proposing an adaptive timer that is cognizant of vehicle speed and weigh in the tradeoff between performance and overhead gain and evaluation results for realistic traffic trace with random vehicle speeds and positioning.

Acknowledgements

We 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] V. Devarapalli, R. Wakikawa, A. Petrescu, and P. Thubert, “Network Mobility (NEMO) Basic Support Protocol,” RFC 3963, Internet Engineering Task Force, Jan. 2005. [Online]. Available: http://www.ietf.org/rfc/rfc3963.txt

[2] 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*, ser. MobiArch ’08. ACM, 2008, pp. 43–48.

[3] B. McCarthy, C. Edwards, and M. Dunmore, “Using nemo to support the global reachability of manet nodes,” in *INFOCOM 2009, IEEE*, April 2009, pp. 2097–2105.

[4] D. Farinacci, V. Fuller, D. Meyer, and D. Lewis, “Locator/ID Separation Protocol (LISP),” draft-ietf-lisp-11, Internet Engineering Task Force, Sep. 2011. [Online]. Available: http://tools.ietf.org/html/draft-ietf-lisp-11

[5] J. Tripathi, J. de Oliveira, and J. Vasseur, “A performance evaluation study of RPL: Routing Protocol for Low power and Lossy Networks,” in *Information Sciences and Systems (CISS), 2010 44th Annual Conference on*, March 2010, pp. 1–6.

[6] T. Clausen and U. Herberg, “Multipoint-to-point and broadcast in rpl,” in *Network-Based Information Systems (NBiS), 2010 13th International Conference on*, September 2010, pp. 493–498.

[7] J. Vasseur and D. Culler. (2009, Jun.) Routing Over Low power and Lossy networks. [Online]. Available: http://datatracker.ietf.org/wg/roll/charter

[8] 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. [Online]. Available: http://tools.ietf.org/html/draft-ietf-roll-routing-metrics-10

[9] S. Wang, C. Lin, Y. Hwang, K. Tao, and C. Chou, “A practical routing protocol for vehicle-formed mobile ad hoc networks on the roads,” in *Intelligent Transportation Systems, 2005. Proceedings. 2005 IEEE*, September 2005, pp. 161–166.

[10] 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*. IGI Global, June 2010, pp. 149–170.

[11] P. Levis, T. Clausen, J. Hui, O. Gnawali, and J. Ko, “The Trickle Algorithm,” RFC 6206, Internet Engineering Task Force, Mar. 2011. [Online]. Available: http://tools.ietf.org/html/rfc6206