

Analyzing Routing Protocol Performance with NCTUns for Vehicular Networks

Eric Gamess^{1*} and Moises Chachati²

¹Escuela de Computacion, Universidad Central de Venezuela, Caracas, Venezuela;eric.gamess@ciens.ucv.ve

²LACORE, Universidad Central de Venezuela, Caracas, Venezuela

Abstract

Vehicular networks are an emerging field of study that brings many new features to support road transportation. In this paper, we evaluate the performance of three well-known routing protocols in vehicular networks, using NCTUns, a popular network simulator. We study those protocols by changing different metrics such as node mobility, number of vehicles, traffic pattern, and number of RSUs. We also report the total elapsed time for simulations and memory consumption with the aim of guiding researchers in their selection of an adequate routing protocol for large simulation scenarios using NCTUns in vehicular networks.

Keywords: NCTUns, Network Simulators, Performance Evaluation, Routing Protocols, Vehicular Networks

1. Introduction

Vehicular networks are a promising and emerging technology that integrates the capabilities of wireless networks to vehicles. The main objective of this new network is to improve safety in road transportation (e.g., collision avoidance, emergency vehicle management, train crossing). Other important applications include traffic optimization, adaptive traffic light control systems, electronic payment (e.g., electronic toll collection, electronic parking payment), vehicle platooning, taxi management, and comfort applications.

Vehicular networks are distributed and self-organized communication networks established between moving vehicles, and therefore characterized by a very dynamic and large topology, where vehicles have a limited degree of freedom in their mobility patterns. High scalability and rapid topology changes are essential parameters for the routing protocols to be implemented in the Intelligent Transportation System (ITS). Thus, research has been done to study the suitability of using traditional Mobile Adhoc Network (MANET) routing protocols in vehicular networks. It is worth mentioning that most of this

research is done with simulations, since IEEE 802.11p^{1,2} products have just recently been released and are still very expensive (around \$2,000 for an OBU or an RSU). Other important factors that also motivate the usage of simulation tools in vehicular network tests include the lack of test-tracks around the world, the difficulties to conduct experiments due to the large number of vehicles and people that must be involved, the inability to reproduce the same experiment, and safety issues.

The majority of the simulations reported in research papers to evaluate the performance of MANET routing protocols for vehicular networks has a small number of vehicles or is based on the random movement of the vehicles, which are not the real life conditions. In this paper, we make a comparative analysis, focused on vehicular networks, of three well-known routing protocols (ADV, AODV, and DSDV) with a famous network simulator called NCTUns. Unlike most of the previous work, we run our simulations on realistic scenarios with a high number of vehicles.

The results of our simulations seem to indicate that ADV is performing better than the other two routing protocols for vehicular networks. DSDV, which is a proactive routing

*Author for correspondence

protocol, shows a poor performance, since it has problems to update the whole routing table, due to the high number of vehicles and the fast topological changes of the network.

The rest of this paper is organized as follows. In Section 2, we discuss the related work. In Section 3, we briefly introduce the selected simulation environment, routing protocols, and the performance evaluation metrics. We describe our simulation scenarios in Section 4. We present and discuss the obtained results in Section 5. Finally, Section 6 concludes the paper and introduces the extension we plan to do for this research.

2. Related Work

A lot of work done in the area of evaluating the feasibility of using MANET routing protocols for vehicular networks consists of survey papers and comparison studies. For example, the authors of³ present a wide range of protocols for wireless networks, explore the motivations behind their design, and discuss the advantages and disadvantages of using these protocols in vehicular environments.

Other contributions are focused on the performance evaluation. For example, the authors of⁴ make a performance comparison between AODV and OLSR in urban environments. In order to model realistic vehicular motion patterns, they use the Vehicular Mobility Model (VMM), which is part of VanetMobiSim⁵. However, the maximum number of vehicles simulated with ns-2 is relatively low (up to 80 nodes). Ref.⁶ studies the performance of AODV and DSDV for VANETs in urban environments where speeds of nodes are limited by traffic signals and stop signs. To generate the mobility pattern of the vehicles, the authors use the Manhattan grid model implemented in BonnMotion⁷. Simulations are done with ns-2 and the number of vehicles varies from 20 to 40. An evaluation of the performance of AODV, DSR, and DYMO is presented in⁸. The authors also select ns-2 and have a constant number of vehicles (50 nodes) for all the simulations. The performance of ADV and AODV is studied in⁹ using NCTUns. The authors select 4 scenarios: (1) highway with 8 nodes, (2) highway with 16 nodes, (3) city with 20 nodes, and (4) city with 50 nodes. In¹⁰, the authors discuss the performance of ADV and AODV for audio and video transmissions in the context of a vehicular network, using NCTUns. They define a simple road map within an area of 1200m x 680m, where 18 vehicles are placed.

Some other papers do a performance evaluation of MANET routing protocols using NCTUns, but not

specifically focused in vehicular environments. Those papers also use a very limited number of nodes^{11,12}.

Unlike our research, the previous contributions are based on: (1) mobility patterns that do not represent the real movement of vehicles which are constrained by other vehicles, the roads, and traffic lights, and (2) a small number of vehicles that does not correspond to the actual roads where the number of vehicles is huge and increasing dramatically.

3. Simulation Tools and Routing Protocols

In this section, we describe the simulation environment used, we make a small introduction to the selected routing protocols, and we define the performance metrics.

3.1 Simulation Environments

There are several simulation environments offered by the industry (e.g., OPNET, Tetcos NetSim, QualNet, and EstiNet) and the Internet community (e.g., JiST/SWANS, OMNeT++, ns-2, ns-3, GloMoSim, and NCTUns). A comparative study of some of these tools and environments is presented in¹³. A few of these tools have not been updated for many years and seem to be dead, like JiST/SWANS¹⁴ and GloMoSim¹⁵. ns-2¹⁶ is a very popular network simulator. However, it does not scale well¹⁷ and one of the goals of ns-3¹⁸ is to provide a more realistic alternative to ns-2. ns-3¹⁸ and OMNeT++¹⁹ are two good open source environments for network simulations. There are active projects and are widely accepted by the researchers that do network simulation. Since we already presented a performance study of OMNeT++ for vehicular networks in²⁰, we decided not to use it in this paper. In this work, we selected NCTUns for the reasons that are presented in the next paragraph. For us, it was important to select an open source simulator since we wanted to freely distribute our contribution.

The National Chiao Tung University network simulator (NCTUns)^{21,22} is a cooperative network simulator and emulator, that has six main components: GUI, coordinator, job dispatcher, simulation engine, patched Linux kernel, and applications. Through the GUI, users can easily create, set up, and control a simulation/emulation execution, as well as gather the final results. To monitor the participating machines (also called the simulation servers), a coordinator process must be run on each of

them to report their current status (busy or idle) to the job dispatcher program. In general, users create their simulation/emulation cases with the GUI and send a request to the job dispatcher that finds an idle coordinator for the execution. When the idle coordinator receives a job, it forks a simulation engine process to run the simulation case assigned by the job dispatcher. At the end of the simulation, the coordinator reports the results obtained by the engine process to the job dispatcher program, and users can access them through the GUI program.

NCTUns is targeted to be installed on Fedora, a famous Linux distribution. The last version of NCTUns (v6.0) is developed for Fedora Core 12. NCTUns is based on a novel kernel-reentering simulation methodology^{21,22}, so that simulated nodes use the real-world Linux TCP/IP protocol stack and can run real-world applications, with almost no modification, to exchange packets. Due to this property, NCTUns does that simulations are more realistic and the obtained results have a high fidelity.

Currently, most of the available simulation tools do not offer models for IEEE 802.11p^{1,2}. Hence, the VANET community has been using models of IEEE 802.11a (WiFi) for simulation purpose, since IEEE 802.11p is an amendment of IEEE 802.11a. For our simulations, we also use models of IEEE 802.11a for the On Board Units (OBUs) and Road Side Units (RSUs).

To run our simulations, we used a PC (HP xw4600) with an Intel Core 2 Duo E6750 CPU at 2.67 GHz and 8 GB of RAM. We setup Fedora Core 12 as the operating system, and installed NCTUns v6.0 on our machine. It is worth mentioning that the research team that has been developed NCTUns has created a commercial version of this simulator, called EstiNet²², which is based on NCTUns. The simulation results of this paper are limited to NCTUns, since we could not have access to EstiNet.

In this paper, we provide simulation results as the average of 3 to 10 redundant executions with identical parameters but different random seeds.

3.2 Translator from ns-2 Traces to NCTUns Traces

Pre-generated traces are needed to move vehicles during the simulations. For more realistic movements, traffic simulators must be used. Most of the traffic simulators around allow users to generate ns-2 traces, since ns-2 is a well accepted network simulator. Unfortunately, NCTUns does not support ns-2 trace format. Instead, NCTUns

has its own trace format. Then, we developed a tool to translate ns-2 traces to NCTUns traces.

We also used ns-2 Trace Toolkit, a Java based software developed by the Internet community, for analyzing, validating, and debugging the ns-2 traces that were used in the City Road Scenario (see Section 4.2).

3.3 Working with Large Number of Nodes in NCTUns

Most of the documentation of NCTUns is oriented for users that will make a simulation with a moderate number of nodes. That is, the simulation scenario is created with the GUI, where the nodes are placed in the field and configured in the Edit mode. However, for a large number of nodes, this convenient method is not adequate. Instead of using the GUI, we wrote a C program that generates the required files, e.g., the simulation network description file (extension.TCL), the traffic description file (extension.TFC), and the scenario description file (extension.SCE), to describe the simulation case based on the parameters. We manually run the simulations, by bypassing the GUI and feeding the.TCL file generated by our C program, directly to the simulation engine for execution.

We faced some limitations when trying to run simulations with more than 254 nodes. The implementation of some MANET routing protocols (ADV, DSDV, and DSR) in NCTUns v6.0 used a /24 netmask, which limits the number of nodes in the simulations. Hence, all the results presented in this paper are constrained to this upper bound.

3.4 Routing Protocols

NCTUns v6.0 supports four routing protocols: Adhoc On-Demand Distance Vector (AODV), Dynamic Source Routing (DSR), Destination-Sequenced Distance Vector (DSDV), and Adaptive Distance Vector (ADV). AODV²³ is a reactive routing protocol (also called on-demand routing protocol), i.e., a protocol that establishes a route to a destination only on demand, unlike proactive routing protocols which find routing paths independently of their usage. AODV defines three types of messages: RREQ (Route Request), RREP (Route Reply), and RERR (Route Error). AODV is based on UDP and sends broadcast messages with the limited broadcast IP address (255.255.255.255). Sequence numbers help avoid the possibility to rebroadcast the same packet more than once. The range of dissemination is indicated by the TTL field in the

IP header. When a source node has data packets to send to a destination node, it first initiates a route discovery process by broadcasting a RREQ. Neighbor nodes, which do not know an active route for the requested destination node, rebroadcast the RREQ to their neighbors until an active route is found or the maximum number of hops is reached. When an intermediate node knows an active route to the requested destination node, it sends a RREP packet back to the source node as a unicast. The path for the route is established when the source node receives the RREP packet. Maintenance of routes is done by generating and propagating a RERR back to the source node by the upstream neighbor of the node, which has moved out of range. Upon receipt of such a message, the source node can re-initiate the route discovery process if it still desires a route to the destination. To discover when neighbors go out of range, AODV exchanges periodic RREP messages (kind of hello messages) between mobile nodes.

DSR^{24,25} is also a reactive routing protocol and shares many concepts with AODV. The main difference is that DSR provides a “Path Accumulation” mechanism where the RREQ and RREP messages carry cumulative information about intermediate nodes. Similarly to AODV, when a node requires a route to a destination which it does not have in its route cache, it broadcasts a RREQ message. The RREQ is flooded throughout the network and all intermediate nodes in the path add their own IP address to record the route followed by the packet before rebroadcasting the request. RREQ is replied by the destination node or an intermediate node, which knows a route to the destination, using a RREP message. If the network is symmetrical, the return route for the RREP will be the reverse list of IP addresses recorded in the RREQ. In other cases, the node may initiate its own route discovery mechanism and piggyback the RREP packet onto it. Thus, routes may be considered unidirectional or bidirectional. DSR does not enforce any use of periodic messages (hello messages) for maintenance of routes. Instead it uses two types of packets for route maintenance: RERR and ACKs. ACKs are used to acknowledge data packets. Whenever a route becomes invalid (e.g., no ACK is received to acknowledge a sent data packet), the node that discovered the failure sends a RERR message to the source. If the source still requires a route to the destination, it will initiate a new route discovery.

DSDV²⁶ is a proactive routing protocol (also known as table driven routing protocol) created as a variation of the Routing Information Protocol (RIP) for mobile adhoc networks. Hence, it is based on the Bellman-Ford routing

algorithm where the metric is the number of hops. In DSDV, each node maintains a routing table that will be broadcasted periodically (full dump) or incrementally (incremental update). To save bandwidth, incremental updates are more frequent than full dump and only transport information changed since the last full dump. The routing table includes all the reachable destinations, the next-hop to them, the number of hops to them, and a sequence number generated by each destination. Any node announces itself with a metric of 1 to its 1-hop neighbors which update their routing tables. They increment the metric by one and rebroadcast the update packets to their 1-hop neighbors. This process repeats until all nodes in the network receive the update with the corresponding metric. Using a sequence number, the mobile nodes can distinguish stale routes from the new ones, and thus prevent the formation of routing loops. The route labeled with the highest sequence number is always used, and the number of hops is a tiebreaker (a lower value is better). Mobile hosts will cause broken links as they move around in the network and can be detected if no information is received from them after a certain time interval.

ADV²⁷ is a distance vector routing algorithm that exhibits some on-demand characteristics by varying the frequency and the size of the routing updates in response to the network load and mobility conditions.

Our first idea was to test all the four routing protocols. However, we had many problems with the implementation of DSR which seems to be buggy in NCTUns v6.0. Hence we limited our simulations to the other three protocols (ADV, AODV, and DSDV).

3.5 Metrics for Performance Evaluation

For our performance evaluation, we used several important metrics²⁸ that we describe hereby:

- Packet Delivery Ratio (PDR): the ratio of the data packets delivered to destinations and those generated by the sources.
- Average End-to-End Delay: includes all possible delays caused by buffering during route discovery, queuing at the interface queue, retransmission delays at the MAC, propagation and transfer times.
- Average Number of Hops: average number of hops for data packets delivered to destinations.
- Normalized Routing Load (NRL): the number of routing packets transmitted per data packet delivered at the destination. Each hop-wise transmission of a routing packet is counted as one transmission.

- Total Elapsed Time: the overall time required to execute the simulation.
- Memory Consumption: the maximum virtual memory consumption during the execution of the simulation.

4. Simulation Scenarios

Since there are no standard benchmarks to evaluate the performance of routing protocols for vehicular networks, we developed two benchmarks named “Circular Road Scenario” and “City Road Scenario”. In the two cases, the benchmarks report: (1) the packet delivery ratio, (2) the average end-to-end delay, (3) the average number of hops, (4) the normalized routing load, (5) the total number of control messages generated by the routing protocol, (6) the total elapsed time to execute the simulation, and (7) the maximum memory consumption during the execution. The “Circular Road Scenario” is a test bed with a lot of topological changes and where vehicles can move at high speed, since there are no traffic lights, no stop signs, and no traffic jams. In the “City Road Scenario”, topological changes take place slowly since the speed of the vehicles is limited by many factors (intersections, traffic lights, stop signs, neighboring vehicles, etc).

4.1 Circular Road Scenario

Our first benchmark consists of a circular road where vehicles are moving with a constant speed and maintain a constant distance with the vehicle ahead. Vehicles and RSUs send UDP datagrams with a fixed length in a quasi-periodic fashion, similarly to the Constant Bit Rate (CBR) model. Since it is pretty simple to move vehicles in a Circular Road Scenario, we put instructions in our simulation for their displacements, i.e., we did not use pre-generated traces in this case. The benchmark has several parameters and allows users to specify:

- the number of lanes for each way.
- the internal radius (radiusCircle) of the circular road.
- the width of the lanes. For all the simulations reported in this paper, we set this value to 3.6 meters since the most common lane width in the US is 12 feet.
- the speed of the vehicles in each lane, i.e., all the vehicles in the same lane have the same speed, but vehicles in different lanes may have different speeds.
- the distance between vehicles in seconds. For all the simulations reported in this paper, we set this value to two seconds as stated as the minimum distance

between vehicles by the Department of Transportation (DoT) of Florida in good weather and road conditions. It is worth mentioning that in the US, some states apply the three-second rule.

- the time interval that the position of vehicles must be updated. For all the simulations reported in this paper, we set this value to 0.25 seconds.
- the time interval between the emission of consecutive UDP datagrams for vehicles (`sendIntervalCar`). To avoid collision, the first datagram is sent after a random time between 0% and 100% of `sendIntervalCar`. The subsequent datagrams are sent after a random time between 90% and 110% of `sendIntervalCar`.
- the time interval between the emission of consecutive UDP datagrams for RSUs (`sendIntervalRSU`). It follows the same rules as `sendIntervalCar`. For all the simulations reported in this paper, we set this value to 0.1 seconds.
- the length of the payload of UDP datagrams. For all the simulations reported in this paper, we set this value to 512 bytes.
- the transmission range. Even though the transmission range of DSRC is targeted to be up to 1,000 meters, practical experiments show that the transmission range of the actual commercial products for OBUs and RSUs is 300 meters. Hence, for all the simulations reported in this paper, we set this value to 300 meters.
- the number of RSUs in the simulations. RSUs are placed 5 meters from the external lane. Figure 1 shows how the RSUs are positioned when the simulation has two of these devices.
- the simulation time. For all the experiments reported in this paper, we set it to 30 seconds.

4.2 City Road Scenario

For our second scenario, we used Simulation of Urban Mobility (SUMO)²⁹, an open source microscopic road traffic simulator, to generate the traces of vehicles in a

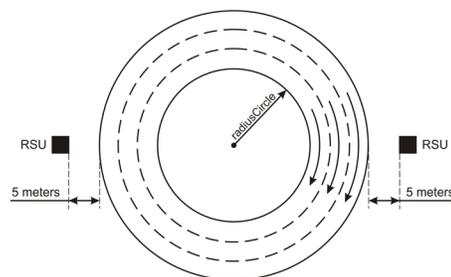


Figure 1. One-way circular road with 3 lanes and 2 RSUs.

neighborhood of the city of Caracas (Figure 2), the capital of Venezuela. We downloaded the digital map of Caracas from the Open Street Map (OSM) project, a collaborative effort to create a free editable map of the world. To achieve our goal, we converted the digital map to a format that can be processed by SUMO with `netconvert` and then initially generate ns-2 traces, before converting them to the NCTUns trace format with our tool. We used the Stefan Krauß car following model³⁰ for the mobility of the vehicles and the open-space propagation model.

5. Results of Our Simulations

In this section, we do several simulation cases and discuss the obtained results. To show how rapid topological changes can be handled by the studied protocols, we start comparing the selected metrics in two different scenarios (Circular and City Road Scenarios) when varying the number of vehicles (see Section 5.1 and Section 5.2). Then, in the next simulation case with 200 vehicles (see Section 5.3), we increase the network traffic in the City Road Scenario to investigate how well the routing protocols can react to this growing demand. Finally in our last simulation case with 200 vehicles (see Section 5.4), we vary the number of RSUs in the City Road Scenario to see the impact of the presence of devices with high network requirements, over the routing protocols.

5.1 Circular Road Scenario when Varying the Number of Vehicles

For this simulation case, we used the Circular Road Scenario with 2 lanes, where vehicles were rotating in the same direction and we varied the number of vehicles (50, 100, 150, 200, and 250 vehicles). The speed of the vehicles was 50 km/h in the internal lane and 60 km/h in the external lane. No RSUs were placed in the simulation.

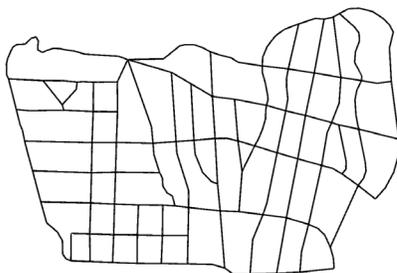


Figure 2. Neighborhood of the city of Caracas (2000m × 1000m).

Figure 3 depicts the total elapsed time for the simulation. As we can see, the simulation with ADV runs faster, while the one with DSDV is the slowest. Also, note that the total elapsed time increases significantly with the number of vehicles.

Figure 4 shows the memory consumption. Results for ADV and AODV are very similar, while DSDV shows the highest one. As expected, memory consumption increases with the number of vehicles.

In Figure 5, we have the PDR. It decreases as the number of vehicles increases. This is due to a more saturated medium as the number of vehicles increases and a longer distance (more hops) to reach the final destination, since the circular road is also bigger. ADV has the better PDR, with is similar to the one shown by AODV. DSDV has a poor PDR that dramatically decreases after 150 vehicles.

Figure 6 shows the average end-to-end delay. Its value increases with the number of vehicles. This is a normal behavior because data packets must travel a longer distance (more hops) to reach the final destination since the circular road is also bigger, and there is more contention for the transmission medium. DSDV has the best end-to-end delay, and it is due to the fact that most of the data packets

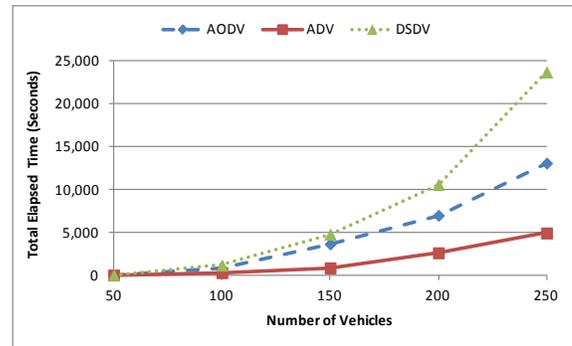


Figure 3. Total elapsed time vs. number of vehicles.

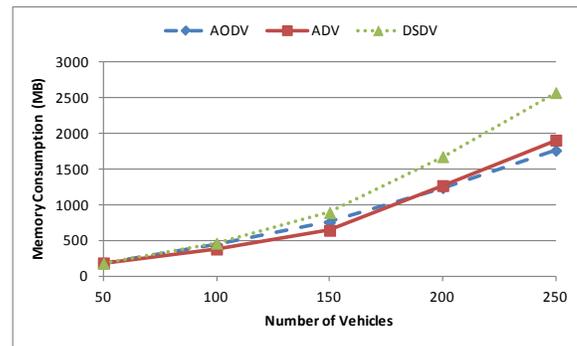


Figure 4. Memory consumption vs. number of vehicles.

that reach their destination in DSDV are 1- or 2-hop away from the source (see Figure 7). ADV has the worst end-to-end delay, since nodes that are up to 6-hop away can be reached with this protocol as shown in Figure 7.

Figure 7 illustrates the average number of hops done by a data packet that arrives to its destination. As we can see, ADV can reach nodes that are 6-hop away from the originator. However, DSDV can only communicate with 1- or 2-hop neighbors. Here, we can infer an important information about using DSDV in the Circular Road Scenario. DSDV is a proactive routing protocol based on RIP. The topological changes in a Circular Road Scenario are so frequent that the information that is not stale in the

routing table of DSDV is about 1- or 2-hop neighbors. This result helps to understand why DSDV has a so poor PDR.

Figure 8 depicts the NRL. It is clear from these curves that the NRL for DSDV has an exponential growth, as a result of the poor performance of this protocol in this scenario.

5.2 City Road Scenario when Varying the Number of Vehicles

For this simulation, we used the City Road Scenario, where vehicles are moving in some neighborhood of the city of Caracas, Venezuela. We varied the number of vehicles (50, 100, 150, 200, and 250 vehicles). No RSUs were placed in the simulation.

Figure 9 depicts the total elapsed time for the simulation. ADV and DSDV have similar results, while AODV is the slowest to be run. Compared to the Circular Road Scenario (Figure 3), the total elapsed time is better here, because we updated the vehicles' position 4 times in a second in the case of the Circular Road Scenario, while the ns-2 traces are updated only once per second in the case of the City Road Scenario.

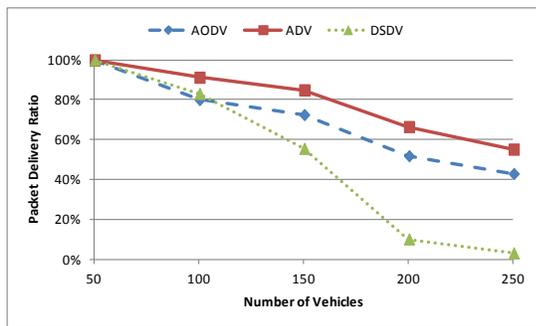


Figure 5. PDR vs number of vehicles.

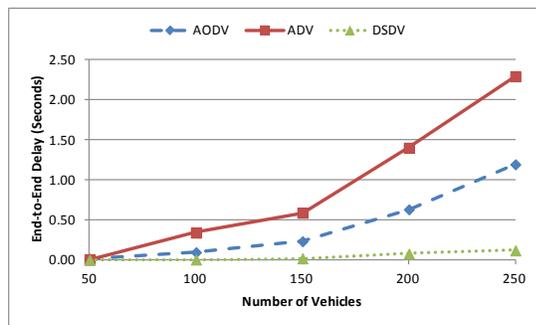


Figure 6. End-to-end delay vs number of vehicles.

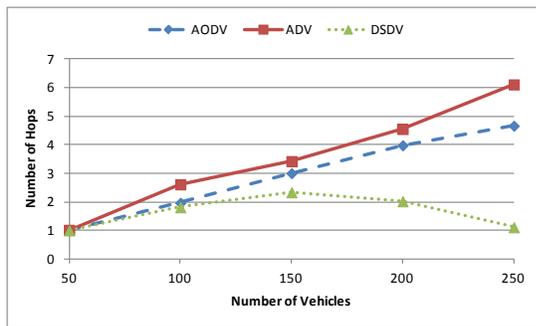


Figure 7. Number of hops vs number of vehicles.

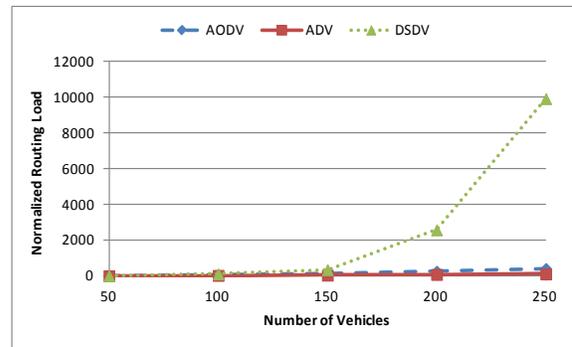


Figure 8. NRL vs number of vehicles.

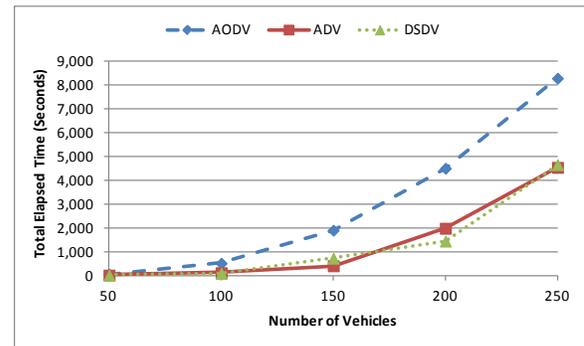


Figure 9. Total elapsed time vs number of vehicles.

The memory consumption is the same for the three protocols as shown by Figure 10 for the City Road Scenario, and lower than the previous scenario (see Figure 4). It increases linearly with the number of vehicles.

Figure 11 depicts the PDR for the City Road Scenario. ADV and DSDV are showing a close behavior. AODV has the lowest PDR of the three routing protocols.

Figure 12 illustrates that AODV has the lowest average end-to-end delay in the City Road Scenario, while ADV and DSDV have a similar behavior.

Figure 13 shows the average number of hops made by datagrams before arriving at their destination. All the

three protocols are maintaining this value between 3 and 4 when varying the number of vehicles.

The NRL is shown in Figure 14. Compared to the NRL of the Circular Road Scenario (see Figure 8), we can observe that ADV and AODV do not change much. However, DSDV significantly improves its values.

DSDV is doing much better in the City Road Scenario than in the Circular Road Scenario. Topological changes are more aggressive and frequent in the Circular Road Scenario resulting in the poor performance of DSDV in this case.

5.3 City Road Scenario when Increasing the Network Traffic

In this simulation, we chose the City Road Scenario with 200 vehicles and varied the traffic pattern by changing the time interval between the emission of consecutive UDP datagrams for vehicles (`sendIntervalCar`). We selected the following values for the number of datagrams to be sent in one second: 0.5, 1, 2, 4, 6, 8, and 10 pps (packets per second).

Figure 15–Figure 18 depict the PDR, end-to-end delay, number of hops, and NRL when the network traffic

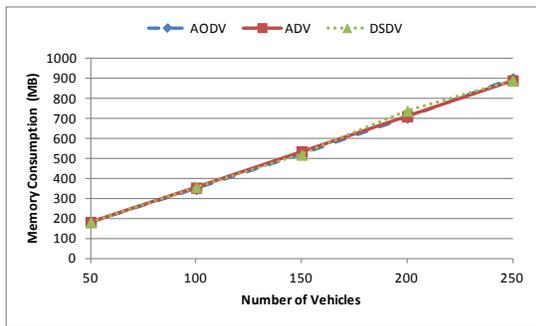


Figure 10. Memory consumption vs number of vehicles.

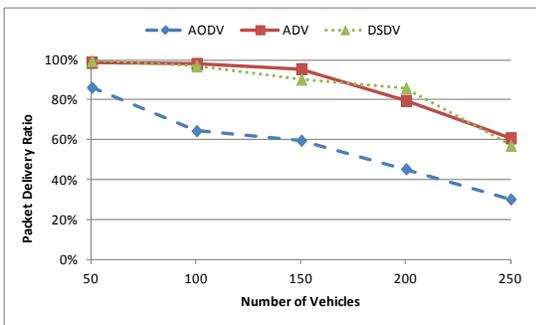


Figure 11. PDR vs number of vehicles.

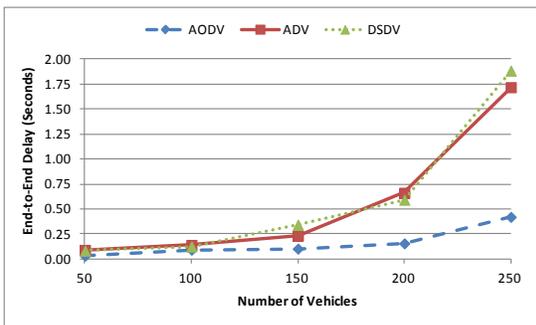


Figure 12. End-to-end delay vs number of vehicles.

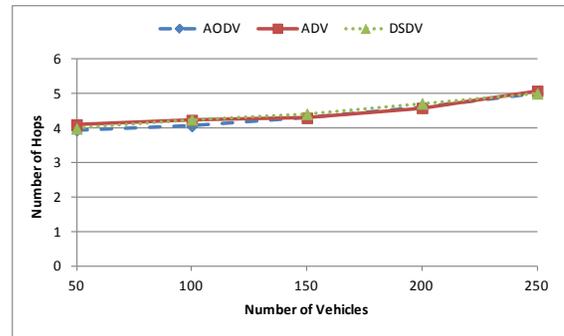


Figure 13. Number of hops vs number of vehicles.

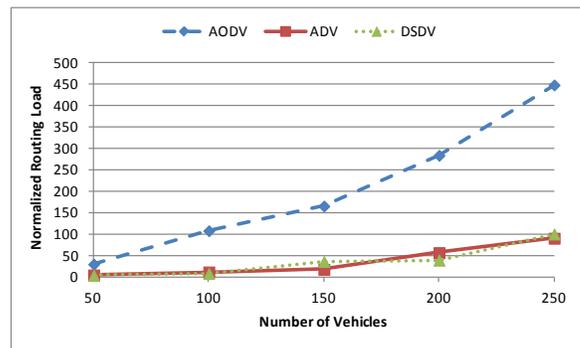


Figure 14. NRL vs number of vehicles.

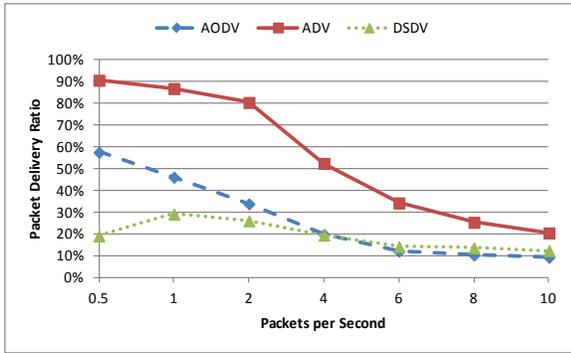


Figure 15. PDR vs packets per second.

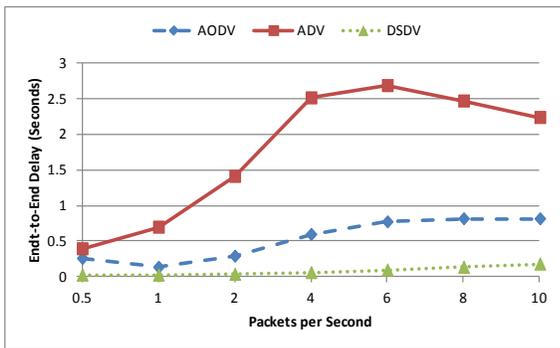


Figure 16. End-to-end delay vs packets per second.

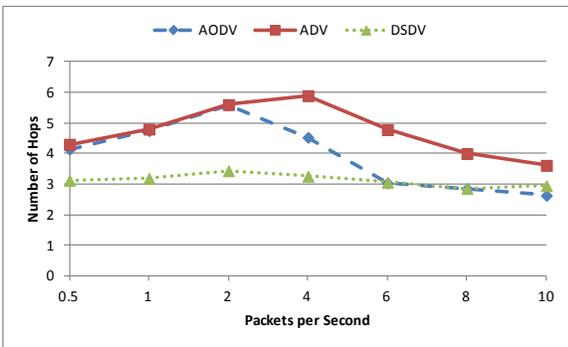


Figure 17. Number of hops vs packets per second.

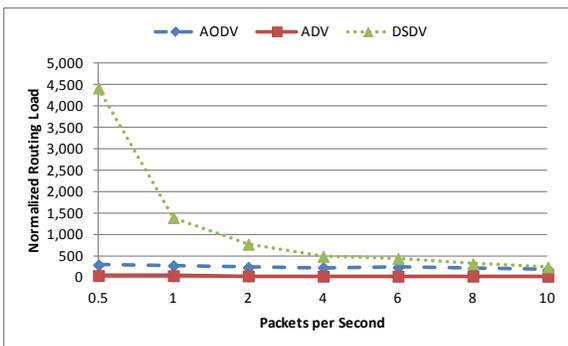


Figure 18. NRL vs packets per second.

is growing. According to our simulations, ADV is doing better than the other two protocols, with the highest PDR and number of hops, and a lowest NRL. The end-to-end delay of ADV is the biggest, and can be seen as a weakness of ADV. However, it is not, since ADV can reach destinations that are further away from the source of a packet, as can be inferred from Figure 17.

5.4 City Road Scenario when Varying the Number of RSUs

In this simulation, we chose the City Road Scenario with 200 vehicles and varied the number of RSUs. RSUs are important elements in vehicular networks as they enable the communication between the moving vehicles and other more traditional networks (e.g., DoT network, the Internet). Hence, it is clear that a high percent of the data traffic that will be sent by a moving vehicle will need to reach those devices. In this simulation, vehicles are sending 1 pps, while RSUs are generating 10 pps. RSUs are always sending their traffic to random vehicles, while vehicles send their traffic to either RSUs (50%) and vehicles (50%).

Figure 19–Figure 22 depicts the PDR, end-to-end delay, number of hops, and NRL when the number of

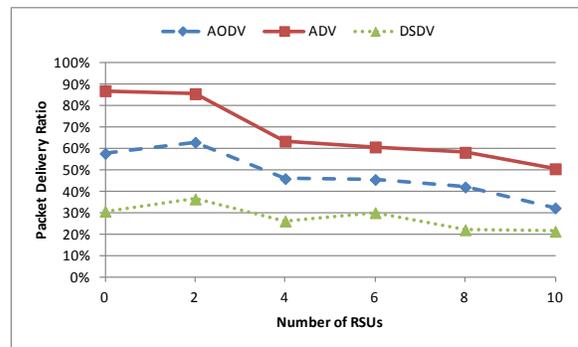


Figure 19. PDR vs number of RSUs.

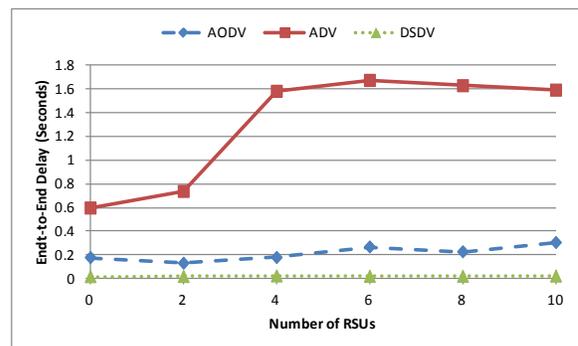


Figure 20. End-to-end delay vs number of RSUs.

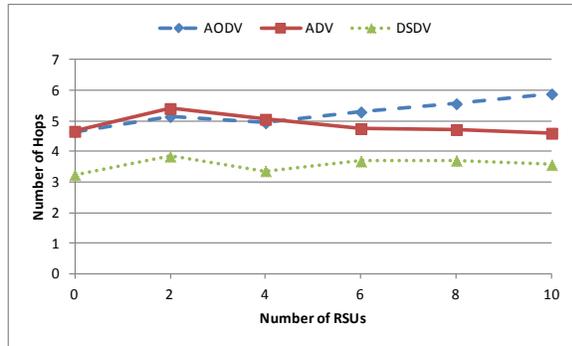


Figure 21. Number of hops vs number of RSUs.

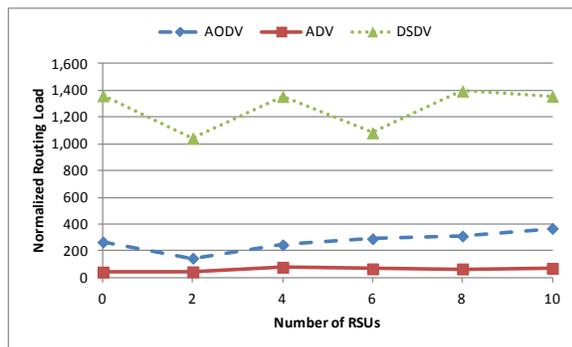


Figure 22. NRL vs number of RSUs.

RSUs is varied between 0 and 10. Again for this scenario, ADV is doing better than the other two protocols, with the highest PDR and a lowest NRL.

6. Conclusions and Future Work

We evaluated the usage of NCTUns to do simulations in the field of vehicular networks. NCTUns v6.0 supports four routing protocols for mobile networks (ADV, AODV, DSDV, and DSR). Our original idea was to evaluate all the available MANET routing protocols, however we faced many problems with the implementation of DSR and then decided to exclude it from this study.

Our simulations seem to indicate that ADV is in general the best routing algorithm for vehicular networks in NCTUns v6.0. The simulations run with ADV had the lowest total elapsed time, the lowest NRL, and the highest PDR. Between AODV and DSDV, the protocol to select will depend on many factors, such as the speed of topological changes in the network. According to our simulations, AODV seems to better handle massive topological changes than DSDV.

Thanks to the kernel-reentering simulation methodology^{21,22}, NCTUns allows simulated nodes to use the real-world Linux TCP/IP protocol stack to generate high-fidelity simulation results. Hence, any real-world program can be run on a simulated network in NCTUns to generate and receive data traffic, without or with a few modifications. Compared to other simulation environments, this is a big strength of NCTUns, however it is also a limitation for scalability. For our simulations, we developed a C program to be run in all the simulated vehicles. Basically, the C program generates UDP packets, receives and processes the UDP packets sent to the node, and cooperates with the report of final results. When running the simulations, a process is created for this C program for each vehicle in the simulation server, as if it were executed on the on-board computer of the vehicle. Therefore, when running a simulation with N vehicles, NCTUns created N processes in the simulation server (one for each vehicle). In simulation tools such as ns-3¹⁸ or OMNeT++¹⁹, we will just have one process or a limited number of processes, when running the simulations. A large number of processes are CPU and memory intensive. As a result, NCTUns does not scale well, and will show limitations when doing simulations with a large number of vehicles.

As future work, we plan to extend our work by modifying the parameters of the routing protocols and study the effect of this variation on the vehicular networks. We also project to further investigate how to take advantage of the usage of multiple channels in IEEE 802.11p, as specified in³¹, to improve the performance of the classical MANET routing protocols when used in the vehicular context.

7. Acknowledgement

We thank the CDCH-UCV (Consejo de Desarrollo Científico y Humanístico) which partially supported this research under grant number: PG 03-8066-2011/1.

8. References

1. IEEE Std. 802.11p. Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 6: Wireless Access in Vehicular Environments. 2010 Jul 15. Available from: <http://ieeexplore.ieee.org/servlet/opac?punumber=5514473>
2. Jiang D, Delgrossi L. IEEE 802.11p: Towards an International Standard for Wireless Access in Vehicular Environments. Proceedings of the 67th IEEE Vehicular

- Technology Conference (VTC Spring 2008); 2008 May 11–14; Singapore. IEEE; 2008. p. 2036–40.
3. Nagaraj U, Kharat M, Dhamal P. Study of various routing protocols in VANET. *IJCST*. 2011 Oct-Dec; 2(4):45–52.
 4. Haerri J, Filali F, Bonnet C. Performance Comparison of AODV and OLSR in VANETs Urban Environments under Realistic Mobility Patterns. *Proceedings of the 5th IFIP Mediterranean Ad-Hoc Networking Workshop (Med-Hoc-Net 2006)*; 2006 Jun; Lipari, Italy.
 5. Harri J, Fiore M, Fethi F, Bonnet C. Vehicular Mobility Simulation with VanetMobiSim. *Simulation: Transactions of the Society for Modeling and Simulation International*. 2011 Apr; 87(4):275–300.
 6. Bharadwaj PS, Rashmi S, Shylaja B. Performance Evaluation of MANET Based Routing Protocols for VANETs in Urban Scenarios. *Proceedings of the 2011 International Conference on Network and Electronics Engineering (ICNEE 2011)*; 2011 Sep 16–18; Singapore.
 7. Aschenbruck N, Ernst R, Gerhards-Padilla E, Schwamborn M. BonnMotion: a Mobility Scenario Generation and Analysis Tool. *Proceedings of the 3rd International ICST Conference on Simulation Tools and Techniques (SIMUTools'10)*; 2010 March 15–19; Torremolinos, Spain.
 8. Shahid S, Din NU, Zafar H, Yahya K, Hasan L. Performance Evaluation of Routing Protocols in Vehicular Ad Hoc Networks. *Pakistan J Sci*. 2012 Mar; 64(1):35–38.
 9. Dalal K, Chaudhary P, Dahiya P. Performance Evaluation of AODV and ADV Protocols in VANET Scenarios. *Int J Comput Appl Tech*. 2012; 3(1):50–5.
 10. Masood O, Akram A, Majeed MN. Performance Evaluation of ADV with AODV for Real-Time and Multimedia Applications in Vehicular Ad-hoc Networks (VANETs). *International Journal of Computing and Network Technology*. 2013 Mar; 1(2):117–25.
 11. Khan K, Zaman R, Reddy K. Performance Comparison of On-Demand and Table Driven Ad Hoc Routing Protocols using NCTUns. *Proceedings of the 10th International Conference on Computer Modeling and Simulation*; 2008 Apr 1–3; Cambridge, UK.
 12. Mumtaz W, Niaz B. Performance Evaluation of Ad-hoc Routing Protocols through NCTUns 6.0. *Int J Comput Appl*. 2011 Apr; 19(5):1–4.
 13. Gamess E, Veracoechea C. A Comparative Analysis of Network Simulation Tools. *Proceedings of the 2010 International Conference on Modeling, Simulation and Visualization Methods (MSV'10)*. 2010 Jul 12–15; Las Vegas, Nevada, USA.
 14. Barr R, Haas Z, Renesse RV. *Handbook on Theoretical and Algorithmic Aspects of Sensor, Ad hoc Wireless, and Peer-to-Peer Networks*. CRC Press; 2005. Chapter 19, Scalable Wireless Ad hoc Network Simulation; p. 297–311.
 15. Bajaj L, Takai M, Ahuja R, Tang K, Bagrodia R, Gerla M. *GloMoSim: A Scalable Network Simulation Environment*. Technical Report. Computer Science Department, University of California at Los Angeles (UCLA); 1999 May. Report No.:#990027
 16. Issariyakul T, Hossain E. *Introduction to Network Simulator NS2*. 2nd ed. Springer; 2011 Dec.
 17. Schoch E, Feiri M, Kargl F, Weber M. Simulation of Ad Hoc Networks: ns-2 compared to JiST/SWANS. *Proceedings of the 2008 International Conference on Simulation Tools and Techniques for Communications, Networks and Systems (SIMUTools'08)*; 2008 Mar 3–7; Marseille, France.
 18. Burbank J. *An Introduction to Network Simulator 3*. 1st ed. Wiley-IEEE Press; 2014 Apr.
 19. Varga A, Hornig R. An Overview of the OMNeT++ Simulation Environment. *Proceedings of the 2008 International Conference on Simulation Tools and Techniques for Communications, Networks and Systems (SIMUTools'08)*; 2008 Mar 3–7; Marseille, France.
 20. Gamess E, Acosta L, Hernandez D. Analyzing Routing Protocol Performance versus Bitrate in Vehicular Networks. *Proceedings of the IEEE Global Information Infrastructure and Networking Symposium 2012 (GIIS 2012)*; 2012 Dec 17–19; Choroní, Venezuela.
 21. Wang S-Y, Huang Y-M. NCTUns distributed network emulator. *Internet Journal*. 2012; 4(2):61–94.
 22. Wang S-Y, Chou C-L, Yang C-M. EstiNet OpenFlow Network Simulator and Emulator. *IEEE Communications Magazine*. 2013 Sep; 51(9):110–7.
 23. Perkins C, Belding-Royer E, Das S. Ad hoc On-Demand Distance Vector (AODV) Routing. RFC 3561; 2003 Jul.
 24. Johnson D, Hu Y, Maltz D. The Dynamic Source Routing Protocol (DSR) for Mobile Ad Hoc Networks for IPv4. RFC 4728. 2007 Feb.
 25. Khan KUR, Zaman R, Reddy AV. Performance Comparison of On-Demand and Table Driven Ad Hoc Routing Protocols using NCTUns. *Proceedings of the Tenth International Conference on Computer Modeling and Simulation*; 2008 Apr 1–3; Cambridge, UK.
 26. Perkins C, Bhagwat P. Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers. *Proceedings of the Conference on Communications Architectures, Protocols and Applications (SIGCOMM 94)*; 1994 Oct 12; London, UK.
 27. Boppana R, Konduru S. An Adaptive Distance Vector Routing Algorithm for Mobile, Ad hoc Networks. *Proceedings of the Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2001)*; 2001 Apr 22–26; Alaska, USA.

28. Zafar H, Harle D, Andonovic I, Khawaja Y. Performance Evaluation of Shortest Multipath Source Routing Scheme. *IET Communications*. 2009 May; 3(5):700–13.
29. Behrisch M, Bieker L, Erdmann J, Krajzewicz D. SUMO - Simulation of Urban MObility: An Overview. *Proceedings of the Third International Conference on Advances in System Simulation (SIMUL 2011)*; 2011 Oct 23–29; Barcelona, Spain.
30. Kraub S, Wagner P, Gawron C. Metastable states in a microscopic model of traffic flow. *Phys Rev E*. 1997 May; 55(304):55–97.
31. IEEE Std. 1609.4. IEEE Trial-Use Standard for Wireless Access in Vehicular Environments (WAVE) - Multi-Channel Operation. 2006 Nov 29. Available from: <http://ieeexplore.ieee.org/servlet/opac?punumber=4025690>