802.11ac and p in a Simulated VANET Environment

Michael Lee
Department of Computer Science
University of Alabama
Tuscaloosa, AL, USA
mslee6@crimson.ua.edu

Beichen Yang
Department of Computer Science
University of Alabama
Tuscaloosa, AL, USA
byang12@crimson.ua.edu

Travis Atkison
Department of Computer Science
University of Alabama
Tuscaloosa, AL, USA
atkison@cs.ua.edu

***Abstract*—In the effort of making a more realistic simulation environment for VANET systems, we employ real traffic data to create a mobility scenario for ns-3. We then test how our simulated system performs when running different variations for network protocols. One version is optimized for general use, while the other is optimized for a VANET environment. We then evaluate how effective the optimizations for VANET are as well as the accuracy of our test system. We found that 802.11p performed better than 802.11ac in our simulated system in terms of both packet delivery and computational efficiency, and that our system seemed to be a reasonable model of the real world.**

I. Introduction

SMALL computing devices are becoming widely available, leading to their integration in many preexisting technologies. Vehicular technologies are one of the many fields seeing a rise in integrated computing devices. As a result, much development is being done on the software side to create technologies that can take advantage of these integrated devices and connect them together into a network that can pass various information of particular relevance to vehicles, including navigation information, safety information, and information for many other convenient services. As these technologies are being developed, the need for an accurate testing environment arises. However, using actual vehicular devices for testing is difficult due to cost constraints on purchasing multiple vehicles that developers can test their software on, as well as issues with creating a well-controlled real world environment and concerns over the safety of vehicles that have software being tested on their devices. As a result, it becomes a large concern to create a virtual environment that accurately models the real world scenario in which a VANET system is being employed.

Resultantly, we endeavor to create a testing system that simulates the real world environment for VANET with a high degree of accuracy. Our system will be run in ns-3, a commonly used network simulator that provides a very accurate approximation of network performance [1]. In order to make the system more closely model the real world environment, we will use real traffic data collected from intersections to generate mobility pattern. This data will approximate the movements of the vehicles through the intersection and then be used to describe the movement of nodes in the ns-3 simulation. We can combine the accurate network simulations of ns-3 with the accurate mobility patterns we generate, creating a very realistic approximation of vehicle mobility at a real world intersection. We will then use this testing environment to test the different performances of 802.11ac and 802.11p. These protocols were chosen because 802.11ac has better general performance that 802.11p, but 802.11p was proposed as a variant that gives a better performance in an ad-hoc vehicular environment [3]. By testing these two, we can see what aspects of 802.11p perform better in our simulated VANET environment and whether it performs better than 802.11ac for this situation.

978-1-7281-0858-2/19/$31.00 © 2019 IEEE

II. Background & Related Works

A. 802.11ac

First, let us begin by discussing 802.11ac. 802.11ac is the modern general purpose version of the original network standard IEEE 802.11. It was originally developed as a way to increase the throughput of current network systems, with hopes of reaching a gigabit wireless connection. It was based on 802.11n and sought to enhance the notions of a multiple input, multiple output transmission system while also increasing channel bandwidth [2].

802.11ac maintained a number of the mandatory physical features required in 802.11n, including using binary convolutional coding for forward error correction, basic multiple input multiple output, modulation and coding schemes, and the regular guard interval of 800 nanoseconds. One of the key changes 802.11ac made was adding in support for 80 MHz channel bandwidth, which allows for the maximum data rate to approximately double from what 802.11n offered. This allows 802.11ac to only require systems to create a single spatial stream, while 802.11n required systems to use one or two different spacial streams for some modes of operations. In addition, it added two new modulation and coding schemes which provide a further 20% and 33% improvement in the data rate [2]. Additionally, 802.11ac updated handshaking mechanisms to provide increased security and allow nodes to decide on a bandwidth that they can transmit on, as well as informing nodes to the secondary channels each device has available.

B. 802.11p

The other algorithm we are comparing against is 802.11p. We chose to investigate the effectiveness of 802.11p in a simu-

978-1-7281-0858-2/19/$31.00 ©2019 IEEE

lated VANET environment due to the protocol being described as the de facto network protocol for VANET communications [3] [4]. But what elements of 802.11p make it advantageous over general purpose algorithms for a VANET environment? We will begin discussing these advantages here.

One of the key aspects of 802.11 that vehicles can take advantage of is the Infrastructural Basic Service Set. The Basic Service Set (BSS) allows a group of computers to communicate with each other over one shared channel. This feature is very convenient for vehicular networks as it allows many nodes to communicate with each other easily and can be modified slightly to no longer require a definitive access point. However, these BSS have the distinct disadvantage of requiring a large amount of time for the connection to be initialized, as a connecting node must listen for beacons from an access point, then on finding the BSS perform a number of interactive steps, including authenticating itself and the other devices and associating with the set. There exists a variation of this defined for ad-hoc operations called Independent BSS (IBSS), which requires the same establishment process that BSS does [3]. However, these systems are not well suited for a VANET environment, as these procedures can consume a large amount of time, and when vehicles are moving at high speed in opposing directions, their window for communications can be very short [3] [4]. As a result, these systems must be modified in order to be successfully used with VANET systems [3].

In order to further the utility of these systems for VANET environments, 802.11p made some modifications to the system. The first goal of these modifications was to reduce the amount of time required for a vehicle to join a BSS by decreasing the need to exchange packets. A BSS will be configured to always be on the same frequency so that vehicles do not have to scan for it, and will use the same Basic Service Set Identifier (BSSID). As the name might suggest, a BSSID is the name of a BSS, and vehicles use this part of a packet to decide whether they should receive a packet or ignore it. This allows for vehicles to attempt to pass messages at any point in their travels, allowing safety information to be broadcasted to all vehicles on the highway without any time lag while a new node attempts to connect to other nodes on the network. Since IEEE 802.11 reserves a wildcard BSSID composed of all ones for exclusively management frames, 802.11p can repurpose this wildcard BSSID for security messages and management frames without affecting the number of unique BSSID values available to the user. This mode of communications is referred to as Wireless Access in Vehicular Environments (WAVE) mode [3].

While WAVE mode provides a quick and easy way for safety information to be passed across the network, even other applications may suffer greatly from having to use the traditional 802.11 exchange to set up their communications. As such, 802.11p offers another new system of communication that applications can employ. This system is a new form of a Basic Service Set (BSS) called a WAVE BSS (WBSS). To form this WBSS, a station first transmits an on demand beacon. This form of beacon does not need to be periodically repeated and advertises the services offered by the WBSS. A receiver station can receive the beacon, consider whether it wants to join based on the services provided, and configure itself to enter the WBSS from the information contained in the WBSS alone. This allows the station to join the network after only receiving a single beacon, removing the need for a back and forth handshake entirely, greatly reducing the lag between location the BSS and joining it [3]. This decrease in overhead time increases the amount of time the vehicle can spend connected to the network, and since this may be short in the first place it can cause a very large percentage-wise increase in total application data throughput [3] [4].

802.11p also makes some changes at the physical level, though these changes are more limited since they want to limit the number of hardware changes necessary to implement 802.11p. The first of these changes is the change in width of the channel. Most versions of 802.11 have a 20 MHz wide channel, while 802.11p has a 10 MHz wide channel. This means that the range of frequencies dedicated to a single channel is doubled for 802.11p, so a single channel will eat up as much of the available broadcasting frequencies as two normal channels [3]. This change was necessary since the Doppler effect could distort the transmissions of vehicles to be outside of the 20 MHz range [3] [5]. This would cause a vehicle's message to be received on a different channel from the one it was broadcasting on, preventing the intended received from reading it while also possibly interfering with the communications on another channel. The switch to 10 MHz appears to be sufficient to offset this problem while also remaining a scaled version of 802.11a [3] [5].

Due to the high proximity of vehicles in many applications of a VANET system, cross channel interference could cause a large reduction in performance of a VANET. As a result, 802.11p introduces mandatory levels of channel management policies to reduce the amount of this interference. Generally these policies are considered to be outside of the scope of 802.11 and handled at higher levels, 802.11p made these mandatory due to the high importance of such policies for this system [3]. The last change made at the physical layer is the modification of four spectrum masks for different classes of operations. These spectrum masks were made to be more stringent than the masks used in current 802.11 systems [3].

C. Related Works

In order to accurately portray a VANET environment, one must have a way to simulate both vehicle mobility traces as well as the network environment. Some existing simulators for VANETs only generate one of these aspects, while others attempt to provide an integrated framework that will simulate both of them together [6]. As such, our first task was to identify which systems were best for our process as well as what conbined simulators already existed for our purpose.

When investigating vehicle mobility pattern generators, we considered many different available services for this purpose. Some of the ones we considered included SUMO, MOVE, CityMob, FreeSim, and VanetMobiSim [7] [8] [9] [10]. However, the ones that met our requirement of being open-source, able to input real world maps, and able to model intersections were only SUMO and VanetMobiSim, and we elected to use SUMO of these two due to familiarity with the software.

There were also a large number of different simulators that we could employ as our network simulator, including ns-2, ns-3, GloMoSim, JiST with the SWANS add-on, and GTNetS [11] [1] [12] [13] [14]. As there weren't any particular necessary features for our research that only one simulator supported, we elected to use ns-3 due to availability as well as our familiarly with coding in C++.

There are a number of different testing software that seek to create a realistic VANET simulation by combining the efforts of generating traffic data and simulating the network. We chose to investigate NCTUns, GrooveNet, and MobiREAL. These software provide a number of benefits that can be helpful when creating a realistic VANET simulation. NCTUns provides the additional feature of simulating the systems that the network being tested will be run on, allowing one to more accurately determine the impact of their systems on the devices' limited processing power. GrooveNet had the most sophisticated environment of these, with a large number of different test modes to run your system on. Finally, MobiREAL can simulate both VANET and MANET systems, which is convenient for cross-platform studies. However, none of these provided a convenient interface for inputting real-world traffic data, which was the defining detail of our newly created simluation system [15] [16] [17].

III. Our Research

In order to create our realistic testing environment, our first step was to procure traffic data collected from local intersections. By working together with the Alabama Department of Transportation, we were able to obtain traffic data for multiple intersections. This data included the time at which vehicles entered the intersections, the lane the vehicles passed through, an intersection ID, and the name and location of the intersection. We fed this data into SUMO, a vehicle traffic simulator that allows us to create realistic movement patterns using only the arrival times of vehicles to the intersections. In order to accomplish this, we choose to modeling the traffic volume at the intersection of Hackberry Lane and University Blvd, as you can see in Fig. 1. These roads were chosen due to their relatively high amount of traffic, and we took data from the time period of 12:00pm-1:00pm. This time period was similarly chosen because it was somewhat on the high side for traffic, but not the highest point, so that we could test in conditions of reasonably high traffic.

We first analyzed the traffic volume at the 4 different approaches at this intersections, Hackberry Lane North, Hackberry Lane South, University Blvd East, and University Blvd West. Then we derived the vehicle arrival rates for these approach from traffic volume. We discovered that the arrival rate for northern and southern approaches was around 0.0663 vehicle per second, taking up 41.9% of the total traffic volume passing through this intersection. While for the western and eastern approaches, the arrival rate was around 0.0920 vehicles per second, taking up 58.1% of entire traffic volume. The reason why the eastern and western approaches saw a higher traffic volume is that the University Blvd is the major East-West road for the campus, so as a result it experiences more traffic throughout the day than other streets in the area. We then used OpenStreetMap, a project which describes the location and intersections of roads around the world, to acquire the road network around this intersection and put the map with the approach arrival rate data into SUMO to conduct our simulations [18], as it is showed in Fig. 1. We also assigned the intersection a 4 phases signal plan based on the observation in our data. After finishing the simulation in SUMO, we retrieved the data of the vehicles' positions for each different second in the simulation process. We later use this data to configure the mobility patterns of our nodes in an ns-3 simulation.

Fig. 1. SUMO representation of the intersection (in black) overlaid over same intersection in Google Maps

One of our goal for this research project is that we want to investigate how vehicle mobility could influence the performance of the wireless protocol. Here we choose the 802.11p and 802.11ac to start our investigation. 802.11p has several major modifications in physical layer from 802.11a to suit the need for vehicular communication scenario. The most important modification is that 802.11p reduce the bandwidth from 20MHz to 10MHz in the 5.9 GHz band compared with 802.11ac [20]. This reduction of bandwidth will inevitably lower down the total throughput. However, if we take dynamic nature of the communication channel in Vehicular Network into account, this sacrifice of bandwidth is necessary. A smaller bandwidth makes the signal more robust against fading and increase the tolerance for multipath propagation effects of signals in a vehicular environment. Another difference between the 802.11p and 802.11a is that there is no authentication/association process in 802.11p, it allows vehicle nodes can set up a connection with the AP much faster than 802.11a can [22]. 802.11ac, on the other hand, has significant higher bandwidth up to 160MHz compared with 802.11p [21]. The MIMO-OFDM technology adopted by 802.11ac standard can support for up to eight spatial streams. These two features greatly expand the maximum throughput of mobile nodes and make network resource consuming applications such as HD video streaming become possible. However, 802.11ac protocol does not stress the issue of high mobility scenario. Thus, we can reasonably assume that the performance for 802.11ac could be substantially compromised in VANET. Our experiment in this project aims at figuring out whether the jeopardized 802.11ac protocol or the less capable but more reliable 802.11p protocol could perform better based on our vehicle mobility data.

In order to test the performance of 802.11p against 802.11ac, we had to create two different simulations in ns-3, one for each protocol. There are a number of parameters that we had to configure that will remain constant across both simulations. First, we had to create the nodes that represent each individual car travelling through the intersection. By looking through the data output from SUMO, we determined that there were 570 unique vehicles that travelled across the intersection during our one hour observation period. As a result, we created 570 nodes to model these vehicles. We then used the data collected from SUMO to assign the corresponding movement patterns to each of these vehicles. In order to accomplish this, we used the class called "Ns2MobilityHelper". Ns2MobilityHelper is a class that takes the name of a file that contains mobility data in the format used as a standard for ns-2 [19]. Fortunately, the output from SUMO followed this format, so by pointing this class at our stored mobility data we were able to correctly correlate the movement of each of our nodes to the vehicle they are meant to model. As a result, our network topology configuration was finished after these two steps.

We then created the simulated versions of our devices in each vehicle. To accomplish this, we used the default settings for ns-3's YansWifiPhyHelper and YansWifiChannelHelper. After the physical devices were created, we configured which protocol they would use by using the default settings for ns-3's respective representations of 802.11ac and 802.11p with YansWifiHelper. We then assigned each device a unique IP address and opened port 80 for communications. The devices created by this method were each assigned to one node, with the exception of one extra node that we had installed in the center of the intersection for data collection purposes.

Now that we had all of our nodes set up with network devices using the network protocol associated with that test run, we created traffic to test our system. We gave each node 10 single kilobyte packets that they wanted to transmit while in the proximity of the intersection. These packets were made to be broadcasted to every available node. We chose to make these broadcasts in order to represent a VANET environment more accurately, as many of the messages sent in a VANET are indeed broadcast packets. We then modified the delay time for each node so that they would begin transmitting their packets on some section of the intersection. We created two different timings for nodes to begin transmitting to the different test cases. Our two test cases for this scenario were transmitting around when they reached the intersection and sending a long time after they'd reached the intersection. The first test case will have the vehicles transmit when they are around the most other vehicles, as they can transmit to vehicles from any of the four approaches to the intersection. The second test case will only allow vehicles to meaningfully transmit if they were stopped by the intersection, so it will represent a sparser VANET environment, while the first test case represents a dense VANET environment. We then added in a timer for the simulation so that we could determine how long it took the program to run. By looking at runtime, we could determine the approximate computational complexity of running the simulation with the given protocol.

We also created a second test scenario with a longer distance from where the nodes are created to when they reach the intersection. This longer distance to the intersection allows us to test a wider variety of VANET scenarios. Since we had extra flexibility in transmission times, we created three test cases for this scenario: 1) a test case where the vehicles transmitted their packets very early, generally finishing their transmissions before the vehicles reached the intersection. 2) a test case where the vehicles transmitted their packages near the intersection. 3) a test case where the vehicles transmitted their messages only after a long time had elapsed.

IV. Results

A. Network Performance

For each of our test cases, we measured the number of packets that the network managed to deliver during the duration of the simulation. By looking at this metric, we can get an approximate feeling of how much messages interfere with each other in the network and slow down the connection, as well as determine the approximate differences in throughput between the two algorithms we are testing, 802.11p and 802.11ac. Then by comparing how their performance varies across the test cases we can investigate some of the situations where one of the algorithm performs better than the other.

In our first series of tests, the tests with the smaller distance between where the nodes originate and the intersection, we ran two tests. The first of these, the test with the vehicles sending their messages near the intersection, had the version of the program using 802.11p successfully receive more of the messages. The 802.11p version received 23,587 packets, while the 802.11ac version received 21,179 packets. This means that the 802.11ac version successfully transmitted 11.37% fewer messages than 802.11p managed to transmit. As such, we can see how 802.11p already seems as though it is better suited to transmitting messages in a VANET environment. The results of this test indicate that we can expect the use of 802.11p to lead to increased throughput in VANET systems for at least usage at intersection over 802.11ac.

The second test had similar results, with 802.11p successfully receiving 22,367 packets and 802.11ac receiving 20,031 packets. The difference between the values is very similar, with 802.11ac reaching 11.66% less than what 802.11p was able to reach. Since this test had a slightly more dense mobility scenario than when the vehicles were sending their messages at the intersection, the slight decrease in performance of 802.11ac when compared to 802.11p may be indicative of 802.11ac having a worse performance in low density scenarios. While we can not yet call this difference conclusive, it would be somewhat expected if this were the case since 802.11p has a longer transmission range than 802.11ac. This should allow 802.11p to outperform 802.11ac to a larger degree in VANET scenarios when the density of nodes is low.

Our second test scenario's first test, the test with nodes sending their packets very early, saw the highest number of packets received of any of our tests. The version using 802.11p managed to receive 8,951 packets while the version using 802.11ac received 7,677 packets. This also is the test in which 802.11ac got the furthest from reaching the performance of 802.11p, with it falling a whole 16.60% short of the performance of 802.11p's program. Since this was the least dense network traffic of any of our scenarios so far, it supports our hypothesis that 802.11p has the best performance when compared to 802.11ac for networks that are more sparsely populated.

The second test case for the second scenario created results more similar to the results we saw in the first scenario. This test, which had the vehicles send their packets near the intersection, resulted in 802.11p's program receiving 15,562 packets, while 802.11ac's received 13,665 packets. This means that 802.11ac's performance fell 13.88% behind that of 802.11p's, which is only marginally closer than the performance of our first test case.

The third test case of the second scenario had the fewest messages received of all of our test cases, with 802.11p receiving 16,452 packets while 802.11ac received 15,107 packets. This supports our hypothesis that 802.11p's advantage for VANET scenarios is increased in low density environments, as 802.11ac fell 8.90% short of 802.11p. Once again, the difference between 802.11ac and 802.11p decreased slightly as the density of the network dropped. As a result, it would appear that 802.11p will outperform 802.11ac most significantly when the network is in a low density scenario. If there are any cases where the network is so dense that 802.11ac begins to outperform 802.11p, our test cases did not capture such a scenario. Interestingly, when connections were initiated before the intersection, the node we created as a sink in the middle of the intersection received more packets when running the 802.11ac version than when we ran the 802.11p version. This is indicative of 802.11ac taking longer to send its packets. We can attribute this to an increased amount of interference and backoff in the version using 802.11ac, which may be one of the major causes of 802.11ac's decreased ability to successfully receive packets in our simulated network.

In Fig. 2 we can see the bell curve of number of packets received against the delay time between when vehicles started moving and when they first began sending packets. We chose to use the second test scenario for this graph as the extended roads allow us to have the largest amount of detail on how the behavior of this system varies with different message start times. As we can see from this curve, the throughput increased when the messages were sent as the vehicle approached the intersection. After the throughput reached the peak, the total number of packets delivered decreased gradually when the messages were initiated later in their movement. This was because the vehicles sending the messages had already left the intersection, travelling through the open road which has a lower density of vehicles. Another interesting finding was that the throughput increased more rapidly as the transmissions began closer to the intersection but decreased more when being sent after the intersection. The reason for this difference was that vehicles would approach the intersection at a constant rate until they reach the intersection. However, due to the effect of the traffic light, the amount of time any vehicle may spend at an intersection may vary, and some will remain in this high traffic region for an extended period of time. The total amount of time each vehicle spent at the intersection could range from 0 second to 90 seconds due to the way we set the cycle length. Therefore, this slow decrease of throughput shows how the traffic light causes traffic to accumulate and have more time to send messages. Additionally, after leaving the traffic light, the vehicles will remain in groups that can transmit packets to each other, which causes the total packets that can be sent to not decrease to the same level as it was before the vehicles were grouped by the intersection, as by our settings they entered the intersection randomly.

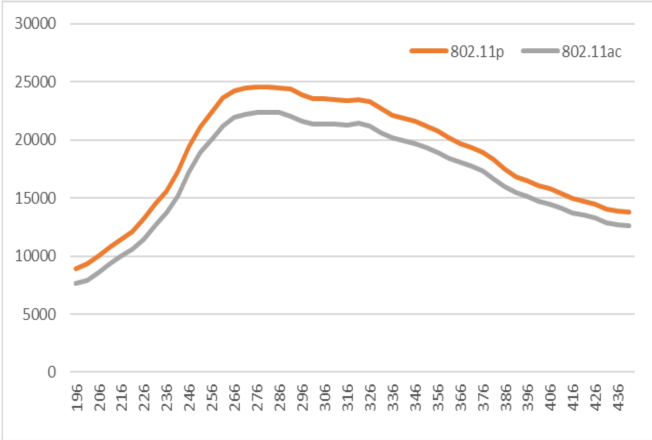

Fig. 2. Packets delivered compared to start time delay across 802.11p and 802.11ac in the long road segment scenario

B. Computational Efficiency

The next metric we tested was the computational requirements of 802.11p and 802.11ac in our simulated environment. We used running time of the simulation as our metric to determine the amount of computation used by the algorithms. We know that 802.11ac has the more efficient algorithm for the general case, but the increased computation time required for backoffs and retransmissions could cause the total amount of computation required for 802.11ac to increase to be greater than that of 802.11p.

For this evaluation, we ran the second test case of the first scenario ten times for each of the two programs. We then recorded the runtime of each run of the program and calculated the mean and standard deviation of the runtimes for each of the programs. 802.11p had a mean of 201.49 seconds, with a standard deviation of 0.5141 seconds. 802.11ac had a mean of 196.61 seconds and a standard deviation of 0.5198 seconds. We then set our null hypothesis to be that 802.11ac and 802.11p had the same runtime, and our alternative hypothesis to be that 802.11p has a shorter runtime than 802.11ac on average. We set our alpha to be 0.05, or that we'd only reject the null hypothesis if the odds of seeing our result if the null hypothesis were true were less than 5%. We then conducted a two-proportion z-test on our collected data.

The two-proportion z-test resulted in a p-value of 0.0175, which was less than our alpha of 0.05. As a result, we rejected the null hypothesis that 802.11ac and 802.11p have the same runtime in favor of the alternative hypothesis that 802.11p has a shorter runtime than 802.11ac. We would conclude that 802.11p has a lower requirement for computational power for a VANET system. We primarily attribute this to the decrease in retransmitted packets, which will allow the computer to perform other computations during the time period that it would be retransmitting packets.

V. Conclusion

In our study, we first created a simulated VANET environment using real world data, SUMO, and ns-3. Afterwards, we used this simulation to perform tests on two different network protocols: one of which was developed for general use, and the other of which was developed to be deployed in VANET environments. We then ran two batteries of tests to evaluate how well the network protocol developed for VANET use was able to run in a VANET scenario. We evaluated two primary metrics: the number of packets that were successfully passed across the network connection and the amount of time it took the simulation to pass all of the scheduled packets across the network. In these tests we discovered that 802.11p, the system developed for usage with VANETs, was able to pass more messages successfully across the network in all our tested scenarios. Additionally, it outperformed the general purpose method further the lower the density of the environment we were running the test in was. On top of this, 802.11p had the shorter runtime for the simulation. This indicates that for VANET environments, using the 802.11p protocol will require less computational power than using the general purpose protocol, 802.11ac. Since we found that 802.11p performed better that 802.11ac in both of our measured metrics this is indicative of both that 802.11p is better suited for a VANET environment than 802.11ac is, as well as that our simulation did accurately provide a realistic testing scenario to resembles a VANET environment. As a result, it would appear that our methodology for creating a realistic VANET scenario based on real-world data was a successful effort, and we can use such methodologies going forward to evaluated other protocol's effectiveness in a VANET environment.

References

[1] Riley, George F., and Thomas R. Henderson. "The ns-3 network simulator." Modeling and tools for network simulation. Springer, Berlin, Heidelberg, 2010. 15-34.

[2] Eng Hwee Ong, J. Kneckt, O. Alanen, Z. Chang, T. Huovinen and T. Nihtilä, "IEEE 802.11ac: Enhancements for very high throughput WLANs," *2011 IEEE 22nd International Symposium on Personal, Indoor and Mobile Radio Communications*, Toronto, ON, 2011, pp. 849-853.

[3] Jiang, Daniel, and Luca Delgrossi. "IEEE 802.11 p: Towards an international standard for wireless access in vehicular environments." VTC Spring 2008-IEEE Vehicular Technology Conference. IEEE, 2008.

[4] Eichler, Stephan. "Performance evaluation of the IEEE 802.11 p WAVE communication standard." *2007 IEEE 66th Vehicular Technology Conference*. IEEE, 2007.

[5] D. Stancil, L. Cheng, B. Henty and F. Bai, "Performance of 802.1 Ip Waveforms over the Vehicle-to-Vehicle Channel at 5.9 GHz ", IEEE 802.11 Task Group p report, September 2007

[6] Lan, Kun-chan, and Chien-Ming Chou. "Realistic mobility models for vehicular ad hoc network (VANET) simulations." 2008 8th International Conference on ITS Telecommunications. IEEE, 2008.

[7] Behrisch, Michael, et al. "SUMO–simulation of urban mobility: an overview." Proceedings of SIMUL 2011, The Third International Conference on Advances in System Simulation. ThinkMind, 2011.

[8] Karnadi, Feliz Kristianto, Zhi Hai Mo, and Kun-chan Lan. "Rapid generation of realistic mobility models for VANET." 2007 IEEE wireless communications and networking conference. IEEE, 2007.

[9] Martinez, Francisco J., et al. "Citymob: a mobility model pattern generator for VANETs." ICC Workshops-2008 IEEE International Conference on Communications Workshops. IEEE, 2008.

[10] Miller, Jeffrey, and Ellis Horowitz. "FreeSim-a free real-time freeway traffic simulator." 2007 IEEE Intelligent Transportation Systems Conference. IEEE, 2007.

[11] Issariyakul, Teerawat, and Ekram Hossain. "Introduction to Network Simulator 2 (NS2)." Introduction to Network Simulator NS2. Springer, Boston, MA, 2009. 1-18.

[12] Zeng, Xiang, Rajive Bagrodia, and Mario Gerla. "GloMoSim: a library for parallel simulation of large-scale wireless networks." Proceedings. Twelfth Workshop on Parallel and Distributed Simulation PADS'98 (Cat. No. 98TB100233). IEEE, 1998.

[13] Barr, Rimon, Zygmunt J. Haas, and R. Van Renesse. "Jist/swans." Wireless Networks Laboratory, Cornell University. http://jist. ece. cornell. edu (2005).

[14] Murray, B. "GTNetS-Home."

[15] Wang, Shie-Yuan, et al. "The design and implementation of the NCTUns 1.0 network simulator." Computer networks 42.2 (2003): 175-197.

[16] Mangharam, Rahul, et al. "Groovenet: A hybrid simulator for vehicle-to-vehicle networks." 2006 Third Annual International Conference on Mobile and Ubiquitous Systems: Networking & Services. IEEE, 2006.

[17] Konishi, Kazuki, et al. "Mobireal simulator-evaluating manet applications in real environments." 13th IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems. IEEE, 2005.

[18] Haklay, Mordechai, and Patrick Weber. "Openstreetmap: User-generated street maps." IEEE Pervasive Computing 7.4 (2008): 12-18.

[19] Raj, Chitraxi, et al. "Simulation of VANET using ns-3 and SUMO." International Journal of Advanced Research in Computer Science and Software Engineering 4.4 (2014).

[20] Abdeldime M.S. Abdelgader, W. Lenan, "The physical layer of IEEE 802.11p WAVE communication standard: The specifications and Challenges. Proceedings of the World Congress on Engineering and Computer Science 2014 Vol II WCECS 2014, 22-24 October, 2014, San Francisco, USA

[21] "802.11ac: The Fifth Generation of Wi-Fi Technical White Paper." CISCO, 2018

[22] Lin WY., Li MW., Lan KC., Hsu CH. (2012) A Comparison of 802.11a and 802.11p for V-to-I Communication: A Measurement Study. In: Zhang X., Qiao D. (eds) Quality, Reliability, Security and Robustness in Heterogeneous Networks. QShine 2010. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol 74. Springer, Berlin, Heidelberg.