The University of Hong Kong

Faculty of Engineering

Department of Computer Science

COMP7704
Dissertation Title
Research on Trust Level Warning System in Vehicular Ad-hoc Network

Submitted in partial fulfillment of the requirements for the admission to the degree of Master of Science in Computer Science

By
Jingyue Chen
3035249162

Supervisor's title and name: Dr. Lucas C.K. Hui
Date of submission: 01/05/2017

Abstract

This report focuses on the behavior of normal drivers in responding to warning systems in three different circumstances and variety of testing maps. The result of this experiment shows the efficiency and credibility of three different methods.
(a) Without collusion team of malicious users, the fewer certificates the system needs the quicker the respond will be received. (b) Fake warning in an unobstructed road, rapid response mechanism causes the lack of alertness for normal vehicles. (c) Real warning and collusion team of malicious users which gives wrong road situation message, the number of collusion teams induce the congregate delay of responds. (d) The generate rule in this system, excessive trust both improving the efficiency of reflection and decreasing the credibility of the system.

In this project, author uses Vehicular Ad-Hoc Networks (VANETs) provides network for Vehicles to communicate with other vehicles and roadside infrastructures and Traffic Control Interface (TraCI) for interlinking road traffic and network simulators. And author finally built the special Trust level system by three methods, such as threshold method, parallel threshold method and Conditionality distinguishable pseudo identities approach.

Keywords：Vehicular Ad-Hoc Networks, Traffic Control Interface, Trust level warning system

Declaration

I declare that this thesis “Research on Trust Level Warning System in Vehicular Ad-hoc Network” has been composed completely by me. And it has not been submitted in any other application for a degree.

Because this project was a research project, some methodology may be built base on some previous papers. All the approaches and results analysis in this report improved or decided by my research. Except the states particularly through acknowledgement, the work presented is entirely my own.

Acknowledgements

I would like to thank my supervisor Professor Lucas C.K. Hui for providing this researching topic and many valuable instructions. I would like to thank Leo Yeung for providing help and patient explanations on the problems I have encountered. And I would like to thank all the people who gave me useful suggestions and instructive comments during the project.

Contents

1 Introduction

In recent years, the utilization of intelligent transportation system (ITS) increases a lot. Vehicular Ad-hoc Network (VANETs) becomes an important part of future traffic research. How to avoid potential hazardous situation is the most significant task considered by most normal users. To decrease incidents and improve safety, a new version of trust level warning system has been put forward in this report. Suppose every message broadcast in the Vehicular Ad-hoc Network (VANETs) has its own trust level, while the later normal vehicle receiving the warning message with trust level, it can be considered as a significant factor to decide to trust or not. And the trust level system can also show an excellent result on picking and removing collusion team of malicious user, detection unit error and inevitable message delay error.

Previous research has already done some task on Vehicular Ad-hoc Network (VANETs), Traffic Control Interface (TraCI) and trust level warning system. But they all have some shortcomings about detect efficiency or system safety. Without VANETs, later vehicle can only detect rear obstacle in a short range by them, even if this method come into our real life, there are still a lots of victims fall into the traffic incident. And previously trust level warning system gives every vehicle a trust level. The rear vehicle trusts them message or not is all depends on their trust level. This protocol makes system overload to calculate the trust level of every vehicle by every step and this will cause delays and mistakes happened very frequent.

Due to these shortcomings the previous protocol has, this report tries to figure them out by the following measure. Every message broadcast in the VANETs with its position has a trust level and position number is its unique identity number. Therefore, the calculation method will be called only if a crash happened and at least some vehicle (the number of the vehicles depends on the specific method) has detected the crash position and broadcasted the crash position message into VANETs. After this step, VANETs can help rear vehicle receive the message by network. In this way, the range of the crash travels inside VANETs become larger and larger. And then, inside of the trust level warning system, the most significant part is to compare the behavior of three methods, such as Threshold Method (TS), parallel threshold method (PTS) and Conditionality distinguishable pseudo identities approach (CDPD); find out the best serving method to complete my assumption. This result will be given out by several tests in variable maps cover the both efficiency and safety. The last result is the most synthesized one compare with the entire current VANETs project. And it will help the driverless vehicle do a better job in their daily operation.

This article aims to how the behavior of drivers in response to warning message is influenced by the message's trust level in the warning systems. Also, gives a complete set of methodology in trust level calculation. Furthermore, this project tries variable models to do the test, such as the model compares the situation with or without malicious, the model compares real crash and fake crash on the main road and the model compare different type of road (simple road, block road and real road). By the help of different set of test settings, the last result and conclusion is the most synthesized one of the current research.

2 Background Research

2.1 VANET

2.1.1 Introduction of VANETs

Vehicular ad-hoc network is a subset of mobile as-hoc network which offer networks for communicate between vehicle-vehicle and vehicle-roadside infrastructures. It is the significant part of intelligent traffic system (ITS). The principle of VANETs is a spontaneous behavior of create a wireless network for data exchange. It can be considered as a lifestyle utilization of mobile as-hoc network (MANETs) and VANET also became the mostly use of inter-vehicle communication (IVC).

Inside of VANETs, there are three important roles On-Board Unit (OBU), Road-Side Unit (RSU) and Trusted Authority (TA). OBU is installed on every vehicle and RSU is deployed on road-side infrastructures in order to let them communicate by Dedicated Short Range Communications (DSRC) protocol over the wireless channel. TA is the judgment department to use some security technology to make sure the

2.1.2 Safety of the whole VANETs.

In-vehicle Domain	In-vehicle is the communication between OBU and application units (AU). AU can be specific equipment. The connection can be wireless or wired, for example, WIFI and Bluetooth.
Ad Hoc Domain	This is the wireless communication between OBU and RSUs. It can be both of Single-hop network and Multi-

	hop network. Such as, V2V and V2E communications.
Infrastructure Domain	The communication amount OBU-infrastructures and RSU-infrastructures to complete the connection function for them into internet. For example, satellite, hot spot, 3G, 4G etc. Specially, the connection for RSU can be wired.

Table 2.1 Safety of the whole VANETs

2.1.3 Motivation of VANETs

By the development of society, tradition traffic system become more and more not adapt with modern urban structure and people travel habits. Urbanization phenomenon leads amount of people live far away from the downtown and every person's daily mileage has increased significantly. Since route becomes overloaded previous rush hour changed from one to two hours into three to four hours, congestion becomes the most serious problems in traffic system. Due to this realistic situation, the original motivation of VANETs is in order to enhance the safety of vehicle driving and improve the effectiveness of road traffic.

For safety reasons, traffic incidents not only create loss of life and property directly but also cause road congestion and make deterioration of the road traffic environment indirectly. The traditional equipment, safety belts, airbags, bumper, anti-lock system and other passive safety system become not enough for real traffic situation. In recent years, active safety systems based on sensors such as radar, laser and camera are also installed in large quantities, providing a reference for the driver and issuing an alarm in a dangerous situation. For the near future, through wireless telecommunication, VANETs try to the use of different vehicles

to improve the sensor capacity and achieve collaborative security system to improve the safety factor of traffic.

Furthermore, people have already recognized that it will be impossible to build new road infrastructure to stop the rate of increasing congestion, so people's attention turn into how to make more efficient use of existing road facilities. Take advantage of VANETs, to composition vehicular sensor network can easily improve the efficiency of modern traffic situation.

Above all, VANET can provide an efficient and unified solution platform for improving traffic efficiency.

2.1.4 Security of VANETs

For VANET, the part people concern a lot is its safety, because it always associates with people's life safety. In traditional network, the most significant security problems are confidentiality, integrity and availability, none of them relate to real people's life safety. Moreover, because the unique characters in VANET, such as high mobility, dynamic topology, short connection duration and frequent disconnection, these unique function brings some short coming to VANET. To overcome these byproducts, this report modifies some of the figures in VANET, I will write more details in the later chapters

2.1.5 Comparison of VANETs and traditional wireless network

Advantage part	Unlimited Transmit Power: The power of each vehicle is always enough, because of the reason that vehicle can generate electricity by themselves.
	Higher Computational Capability: large space of equipment

	and strong power of computation.
	Predictable Mobility: the movement of vehicle is regularly, this makes it easy to be used.
	Well Equipped Cars: most of vehicles already have some kinds of sensor and communication equipment.
Disadvantage part	Large Scale: Because of it occupies the whole Route Network, the scalability of VANET protocol maybe a big problem.
	Partitioned Network: The entire network is not necessarily connected. So in the sparsely populated scenario, it may become unique small network one by one.
	High Mobility: The absolute and relative changes in velocity are large, the movement of each vehicle is complicated, channel changes violently, range of wireless links and node's density impact the degree of connection to the network a lot, the connect and disconnect between vehicle and vehicle change very frequent, the topology structure changes rapidly.

Table 2.2 Comparison of VANETs and traditional wireless network

2.2 Traffic Control Interface (TraCI)

TraCI is a traffic control interface which gives access to run the traffic simulation and allows controlling and managing vehicles' behavior by its own "network" during the simulation time.

TraCI also has a GUI interface to make the simulation result more directly like SUMO. And by using TraCI, SUMO can do more function of traffic simulation, like traffic light task, crossroad task and “stop” sign etc.

2.3 Simulation of Urban Mobility (SUMO)

SUMO is an open source traffic simulation package including net import and demand modeling components, it helps do the research on these several topics, route change selection, algorithms of traffic light and simulate the communication between vehicle and vehicle. Due to these advantages, SUMO has been considered as a framework and frequently used to simulate automatic driving and traffic management strategies.

Node file: Every figure has junctions in order to depict a specific shape. And this node file defines the junction as several nodes and set them in to an x-y coordinate system. And each node has its own unique identical number.

Edge file: The edge file consist every two nodes into a lane. And each edge defines a relationship between every two adjacent nodes (connect or disconnect). Furthermore, same to the node file, each edge has its own unique identical number.

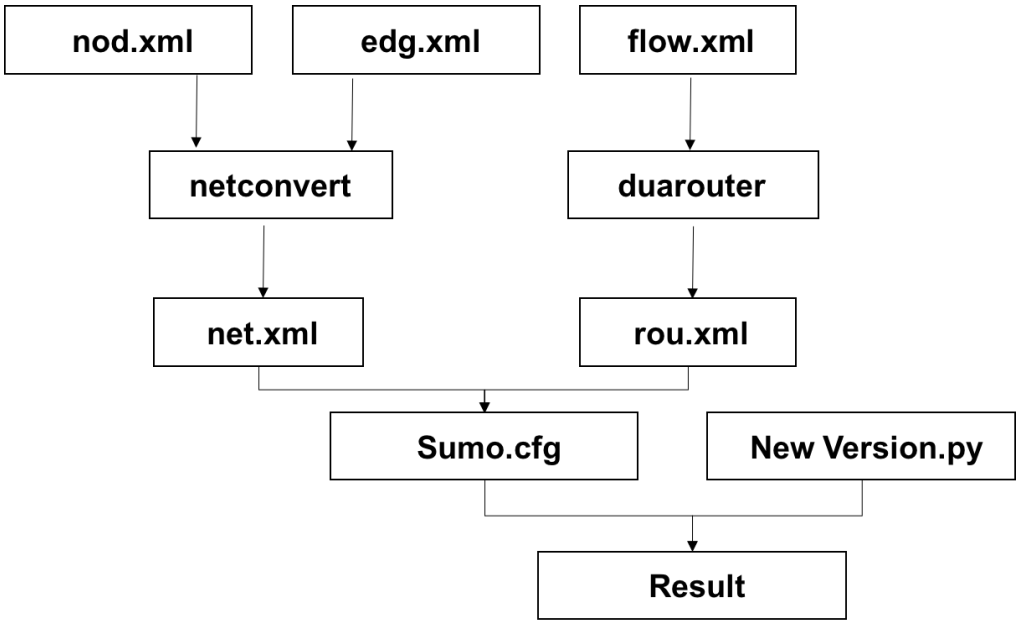


Figure 2.1 Simulation of Urban Mobility

Net file: Netconvert is a useful tool in SUMO. By using this tool, simulator can create an initial complete map file by combining the node file and edge file together. After finish manufacturing the net file, node file and edge file need not to load again while doing the simulation.

Route file: Inside of this file, each vehicle with them unique ID will be defined with fixed route. No matter what situation it meets darning the simulation, it would not change any part of the initial route. Fixed route can be defined in this xml file in order to do the testing task by same variable.

Sumocfg: Sumocfg file is the SUMO project which combine the net file and route file together to run the simulation. By using different define inside Sumocfg, people can do different set of traffic simulation, like variable vehicles with different route set.

New Version.py file: Depends on the methodology which users want to do with traffic simulation, the python can be written variable. In this report, I define a

specific trust level warning system and fixed route changing in order to control the simulation in all aspect.

Result: The result will be shown as two type of values (number of victim and vehicle id of first Beneficiary). In different situation, this value will be considered as different meanings. “The number of victim” count the number of total victims which fall into the traffic crash during the whole traffic simulation. “Vehicle ID of first beneficiary” means the first Vehicle’s ID who has detects the exit warning and turn to another to avoid the crash. Or in another situation, while a fake warring has been broadcasted through the VANET, once the later vehicle recognizes the fake warning, it would send one message to cancel the fake warning. And after the fake warning has been cancelled, the id of first vehicle does not trust the fake warring and keep going on their initial route will be considered as “Vehicle ID of first beneficiary” in this situation.

2.4 Trust Level System

In this report, I define a specific trust level warring system by three different methods, such as, threshold method, parallel threshold method and conditionality distinguishable pseudo identities approach to define the trust level of each warning message.

During the traffic simulation, each message broadcasted through VANET will have its own trust level. And this trust level is calculated by the three algorithm (I will declare more details about the algorithm in the following chapter). After the message has been received by the later vehicle, whether the vehicle trust the

message and turn to the advice route or not is all depends on its trust level rating. The message with a high rating of trust level will get more worthy of trust and there will be more or later vehicle trust it and try to change their route to the advice one. The rating of one's message's trust level is a dynamic value and it can be increase and decrease of other vehicles. If the more vehicles detect the exit crash position the rating of this message's trust level will higher. If the warning has been detected as a fake warning, it's rating of trust level will be decreased by the other vehicles. And after the rating has been decreased less rear vehicles would trust it.

3. Simulation method

3.1 Introduction of Trust Level Warning System:

3.1.1 Threshold Method TS[n]

In Threshold method, at least n verifiers are required for an authenticated warning message and n is the threshold in this method. After enough verifiers have been received by the later vehicles, the system will give out an authenticated message and broadcast the crash position through the vehicular Ad-hoc network. After the rear vehicle received the verified warning message, there is 80% of trust it will worthy. So, in this method, threshold is the most significant thing to control the efficiency and safety. The higher efficiency the method has the less safety it will be. And more over, in this report, I have set 3, 5 and 8 to be the parameter of TS[n] method. And moreover, I will show the selection of these three parameters in the following chapter.

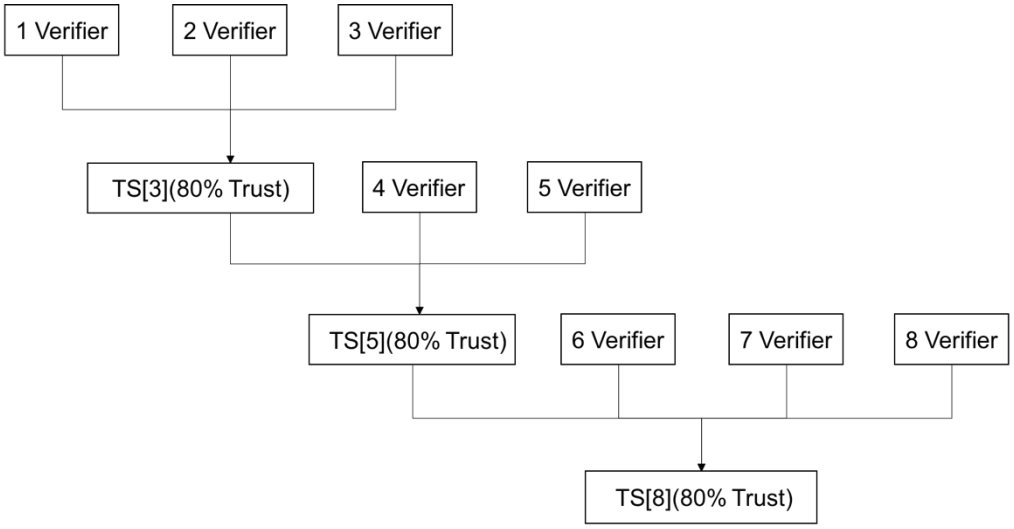


Figure 3.1 Threshold Method

3.1.2 Parallel Threshold Method

Parallel threshold method is a multi-thresholds method. It will give out three parameters like the TS[n] method. But compare to TS method, it will set low, medium and high trust level on each warning message. In the configuration of this report, I set 30% trustable for parameter TS [3], 50% trustable for parameter TS [5] and 80% trustable for parameter TS [8]. It means, while three verifiers have been sent and rear vehicles received a TS [3] warning message broadcasted in the network, the percentages of trusting the warning message is 30%, and the percentage of trust while meeting the threshold TS [5] is 50% and so do TS [8]. Compare with threshold method, parallel threshold method is a more perfect method. PTS [3,5,8] has a high efficiency than TS [8], because it needs less verifiers to finish the authentication. Due to this feature, three verifiers are enough to send a 30% trust, there will be a car to take safe action earlier. And because of its less worthy of trust than TS [3], it is more secure than TS [3] in the set with collusion team. Later and less vehicle trust the warning message may pick out the fake warning more efficiency than the normal threshold method.

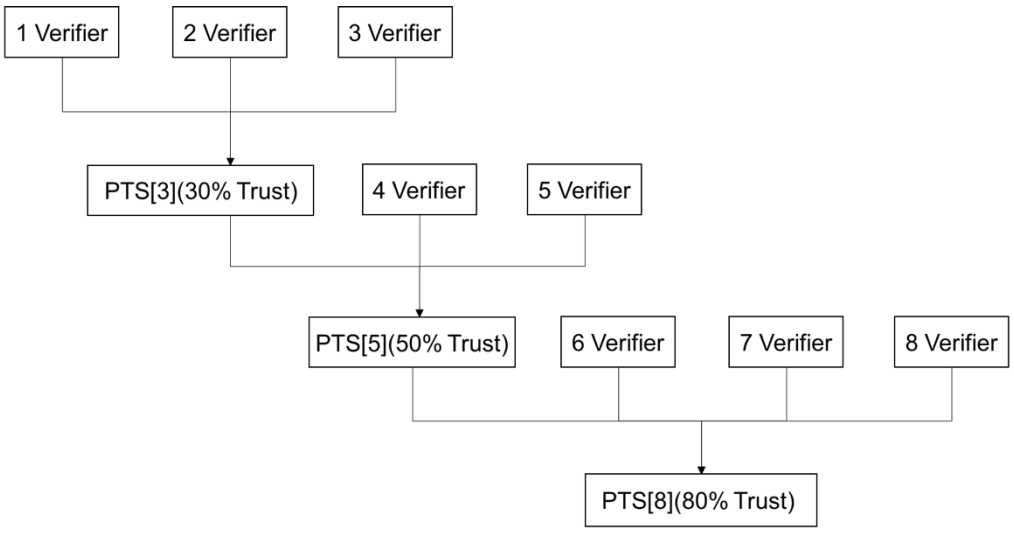


Figure 3.2 Parallel Threshold Method

3.1.3 Conditionality distinguishable pseudo identities approach

Conditionality distinguishable pseudo identities approach [n, p] is a dynamic method which can verify a real warning and withdraw a fake warning more efficiently. N is the parameter that at least n messages should be received before running the CDPD [n, p] method. And p is the minimal percentage of the proportion of positive messages in the number of total messages, if the result computed by positive message and number if total message is more than the parameter then this warning will be authenticated by the network and broadcasted through the whole Ad-hoc network. In this method, all the message sent from the main road (I assume crash in on the main road) will be counted as a denominator and the number of positive message will be the numerator. While the later vehicles received the CDPD [n, p] warning message, they will trust the warning message and 80% of them will follow the guidance of the system to turn their way into the pass way. I set the parameter of CDPD [n, p] method into [5,0.7] and [5,0.9] in this report.

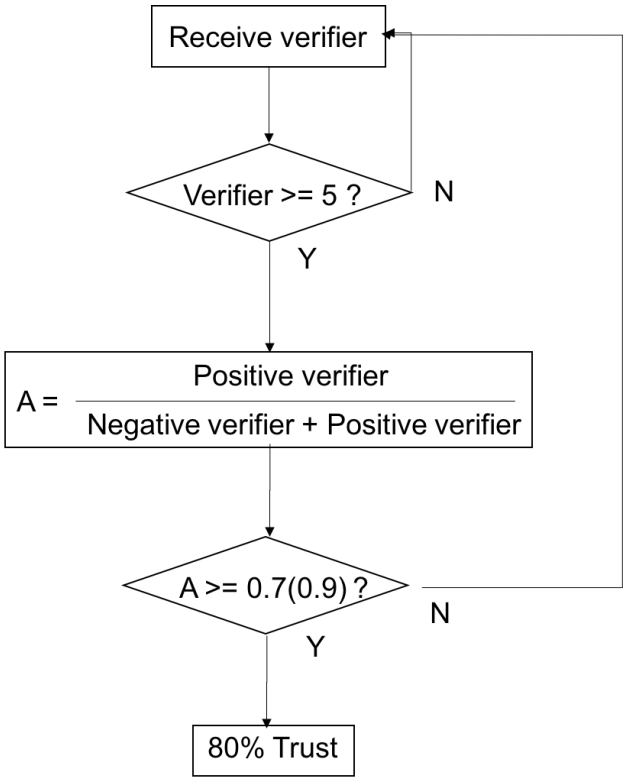


Figure 3.3 Conditionality distinguishable pseudo identities approach

4 Scenario

4.1 Situation without the collusion team of malicious users

In this set, my assumption is that, there is no malicious user in the traffic simulation of these 6 methods. And the test result will show the efficiency of these 6 methods' ability of verifying a real warning message in Ad-hoc network. In this set of simulation, vehicles are set into two types, testing vehicles and crash generator vehicle. The traffic crash generator would not be collected during the traffic simulation time. The distance from the intersection to the crash position is the only variable in set 1. I set five different random route files for each testing vehicles into five different rou.xml file to control the variables in the experiment. And the result will show the efficiency by the two values— "number of victims" and "first non-victim". In this set, excluding the impact of random factors, the advantages and disadvantages in each method will be best shown in this set. If a verified warning massage have been broadcasted in the network, all the later vehicle which received the message will turn their initial route into the advice route. So, the two values of "number of victims" and "first non-victim" is all depends on the efficiency of these three methods. The smaller the two value is the higher efficiency the method is.

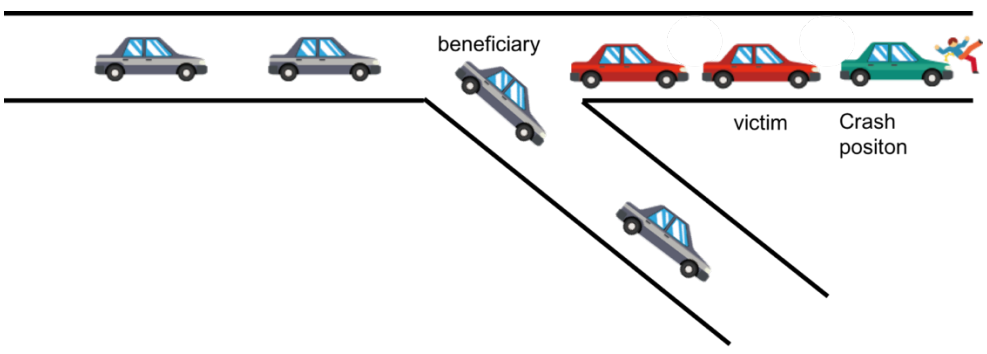


Figure 4.1 Situation without malicious users

4.2 Situation without the collusion team of malicious users (80%)

Due to the situation in the real world, users sometimes do not follow the guideline of network system. So, in this report, I modified the scenes in set1 and set a solution that only 80% of the normal users will trust the warning given by the Ad-hoc system and 20% of the normal users would not turn into the advice route. So, in this set, the value of "the number of victims" will randomly larger than the value I got in set 1. But, objectively speaking, this results of set 1.5 are more realistic.

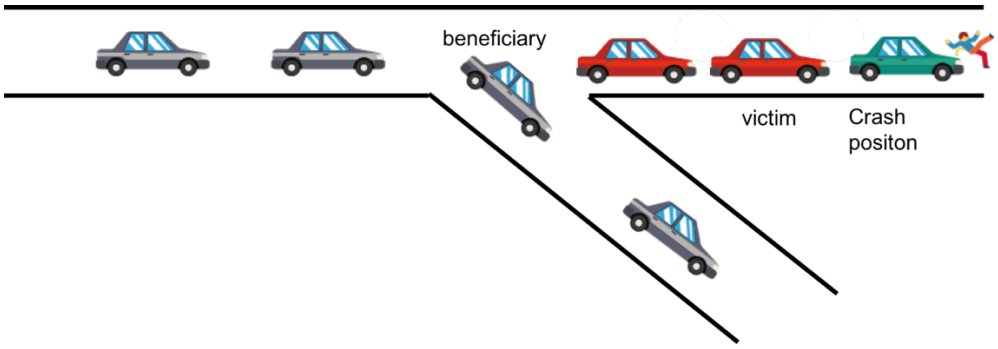


Figure 4.2 Situation without malicious users (80%)

4.3 Situation with collusion team and a fake positive warning message

For set 2, resemble the situation in real world more, this report considers more about the influence of the team of malicious users. Moreover, this set focus on the

ability of Ad-hoc network retrieving an existing fake warning message. There is a verified warning message broadcast in the network before the start of the simulation (I assume this fake warning message is given by a collusion team of malicious users and all the testing users in the simulation are normal users) and the result is to find how long will it cost to change the fake positive warning message into negative. In this set, the distance between the intersection and the crash position is a fixed value equal to 240 (240 is the best testing setting values got from set 1 and set 1.5) and the number of the malicious users inside the collusion team is the only variable in set 2. Also, in order to control the variables in the experiment the route has been defined to a five fixed rou.xml like set 1 and set 1.5. The efficiency will be shown as the values of "the number of victims" and "the vehicle ID of first non-victim". Unlike the previous set, "the number of victims" is the total number of the vehicle who trusts the fake warning message and turn their initial route in the system advice one (The victim vehicle would not recognize the fake warning or offer any helps in changing the positive message into negative). And "the vehicle ID of first non-victim" is the first vehicle ID who does not trust the fake warning and run through the crash position (This crash position with its warning message must has been verified and broadcasted through the whole VANET). So, this vehicle is the first beneficiary and it must give a right respond in order to cancel the fake warning during the simulation time.

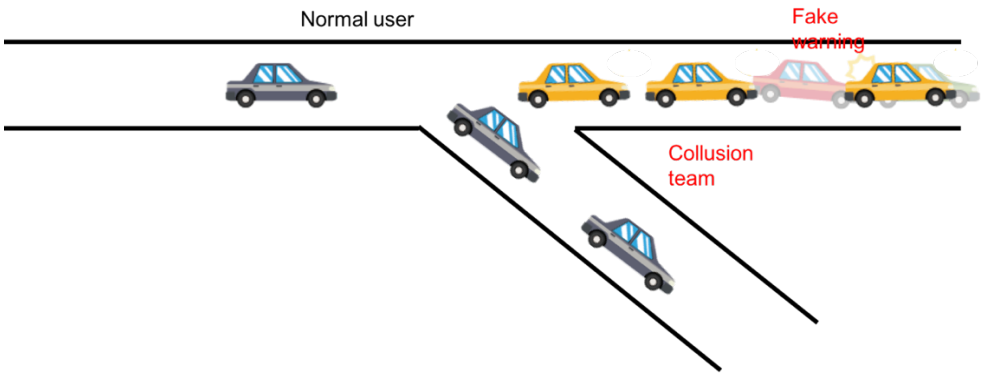


Figure 4.3 Situation with fake positive warning message

4.4 Situation with collusion team and a fake negative warning message

In this set, this report also considers about the impact of malicious users. And it sets a situation that a real crash exit on the road but no initial warning message exist on the road. In this set, the distance between the intersection and the crash position is also a fixed value equal to 240 (240 is the best testing setting values got from set 1 and set 1.5) and the number of the malicious users inside the collusion team is the only variable in this set 3 (The number of malicious users is from 1 to 10). The fake negative warning message is generated and broadcasted by a collusion team of malicious user in the traffic simulation. They will give a negative reflect as a fake message to influent the warning broadcasted through the whole network. There is a real cash exits on the road during the simulation time. The safety will be shown as the values of “the number of victims” and “the vehicle ID of first non-victim”. “The number of victims”, like the result values got in set 1 and set 1.5, is the total number of testing vehicles which fall into the

crash. Due to the impact of malicious user in collusion team, the value of "the number of victims" all larger than the values in set 1 and set 1.5. During the simulation time, after the fake negative warning has been cancelled and a new true positive warning has been generated by the testing vehicle, the id of first vehicle trust the warning message and change their route to the advice route will be recorded as "the vehicle ID of first non-victim" value. This set of test is aimed at testing of robustness of the Ad-hoc network system under the impact of the malicious users.

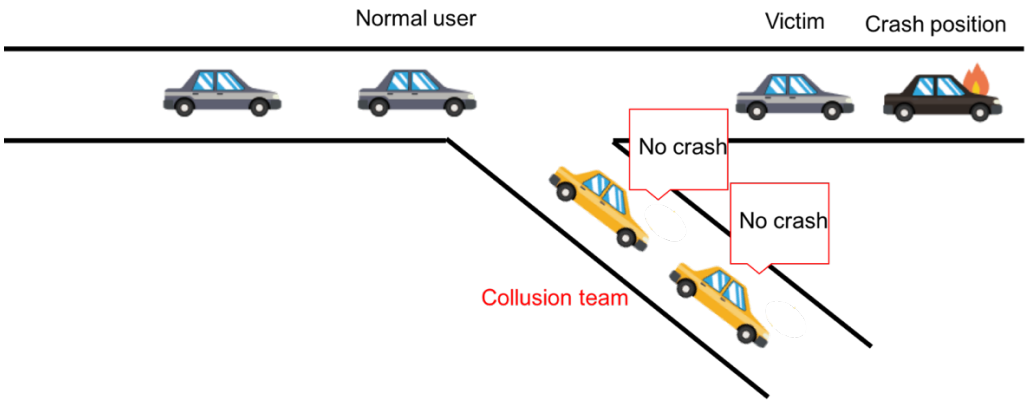


Figure 4.4 Situation with fake negative warning message

5 Traffic Simulation Map

5.1 Simple map

In order to compare the different results of these six methods fairly, I make the simulation test standardized. During the whole dissertation, I will use 5 random files to do the test. In each random file, 80% of all the vehicles run on the main road, 20% run on the branch way. Every vehicle would not change their Route unless meeting any verified warning message.

The road is one-way lane, the vehicles in behind cannot outstrip the car in front. There are lots of pathways that allows the vehicles to arrive their destination but main road should be the shortest path. I named the lane from start point to the intersection is lane “1 to 2”, the lane from intersection to the pass way is “2 to 3” and the intersection to the end of the main road is “2 to 4”. “1 to 2” is equal to 350 units (unit is a special unit defined by SUMO), “2 to 3” is equal to 206 units and “2 to 4” is equal to 400 units.

The first vehicle will be the crash generator in the test, the distance between intersection and crash position is the parameter in this simulation. The detection range and receiving range of each vehicle in this test is a statistic number which is 10 units (unit is a special unit which define in SUMO) and 100 units.

In this trust level simulation, I set 100 vehicles and the simulation will finish within 500 steps. As a traffic simulation application, SUMO has its own unit to depict the time and distance. And in order to relate the simulation result to the real situation in real world. I assumed that 1 range per step is equal to 5 km per hour. All the speed of vehicles will appear in the simulation with a speed of 0(unit) and

the maximum speed of each vehicles is 13.9(unit). Due to the test setting I gave out previously, this result can be interpreted that the speed of the vehicles is from 0 km/h to 69.5 km/h.

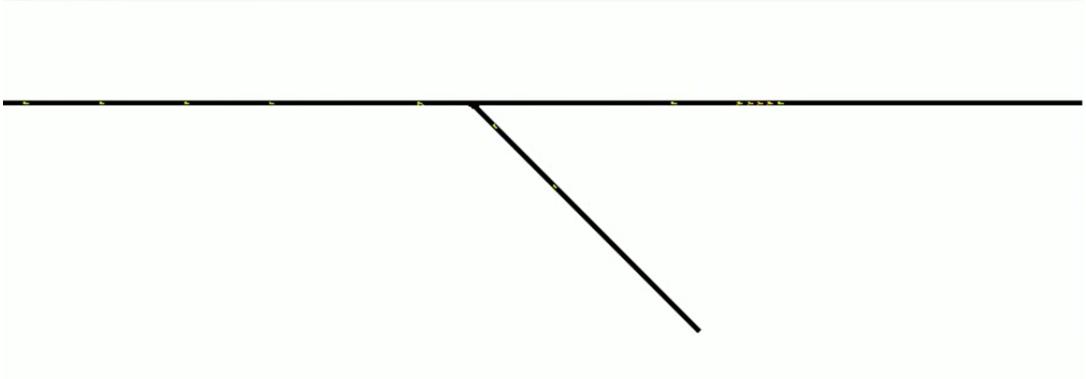


Figure 5.1 simple map

5.2 Complex map

5.2.1 Triangle map

During the implement to suit the situation in the real life traffic situation, this report sets up a complex roundabout map as a transitional phase. This triangle map is a transition from the simple map to the real map.

In the node file, I set 9 nodes to depict the main shape and size of this map. And in edge file, there are 15 edges which means fifteen different relationships between fifteen pairs of two nodes. Moreover, by using the netconvert function of SUMO the net.xml has been created and the figure of the triangle map is on the following figure.

There are five different routes for each vehicle from the random start place to the variable destination, also which way to go is random and average for each

testing vehicle. Normally, the fixed five route is the shortest way for vehicles to go from the start point to the destination point, but there will be a crash happens on this short route. Due to this situation, if the vehicle does not change their initial route to the advice route, it would fall into the crash during the simulation time.

By the help of the three methods, all the following vehicles will try to avoid the crash position by turn into other routes as soon as the warning message has been received. Same with the previously simulation map, all the speed of vehicles will appear in the simulation with a speed of 0 km/h to 69.5 km/h.

Because this map is just a transition map. The main aim is to test whether the three methods' work condition is good or not. Then, the result got from this map would not be analyzed by this report.

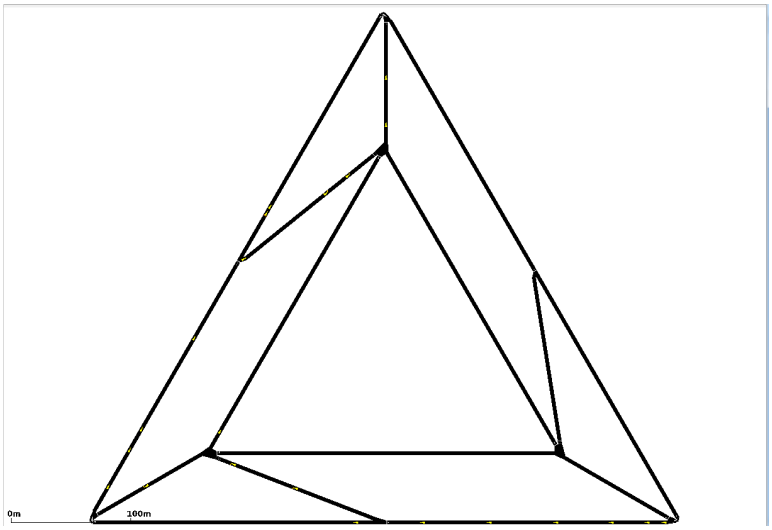


Figure 5.2 Triangle map

5.2.2 Roundabout map

After modifying some code of the three methods, the simulation of these three methods in complex map (The map has more than two different ways to go for

normal testing vehicles) can be accessed. So this project tries to do the simulation in a small road setting of real map.

First, by using “openstreetmap” which is a web site shows every detail of road information to find a suitable road place to be the test road for this map. I choose the part of map sets up a real block map in “central, Hong Kong Island, Hong Kong” which has a small roundabout road near the Victoria Harbor. And map project will be saved as a OSM file.

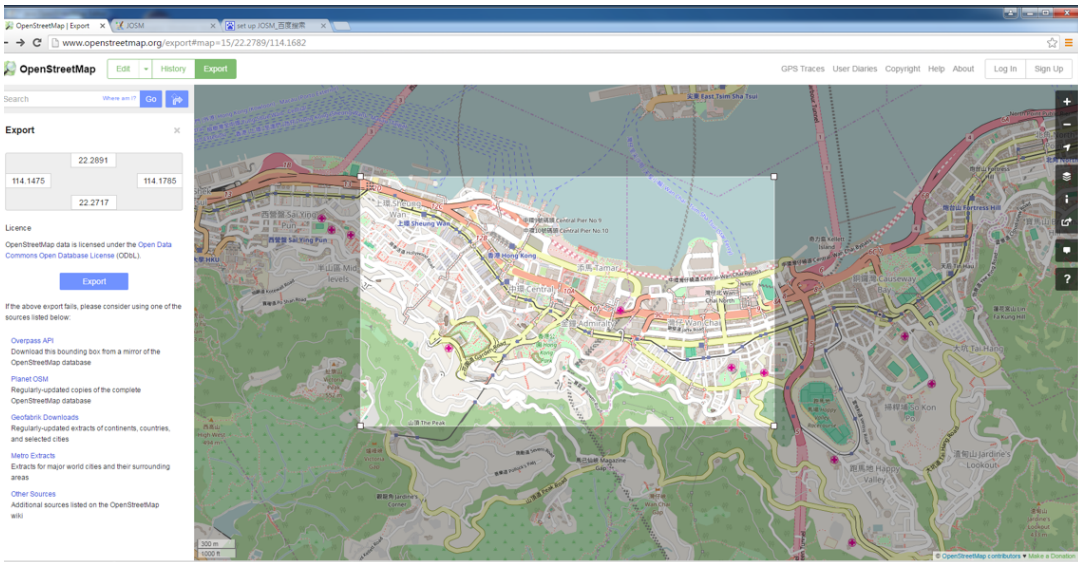


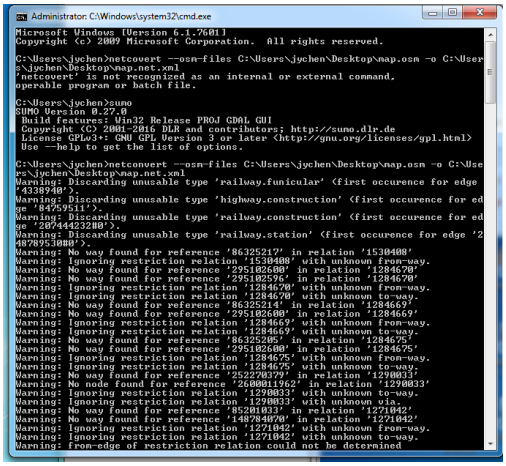
Figure 5.3 Real street map in Hong Kong Island

And then, open the map.osm file in JOSM. In order to simplify this map, I delete almost all the roads except the road setting of roundabout road like the following figure 5.4 and figure 5.5.

Figure 5.4 OSM file in Hong Kong Island

Figure 5.5 Modification of OSM file

And in next step, I translate the JOSM file into SUMO map file and do some further modification. Only left one roundabout road part behind.



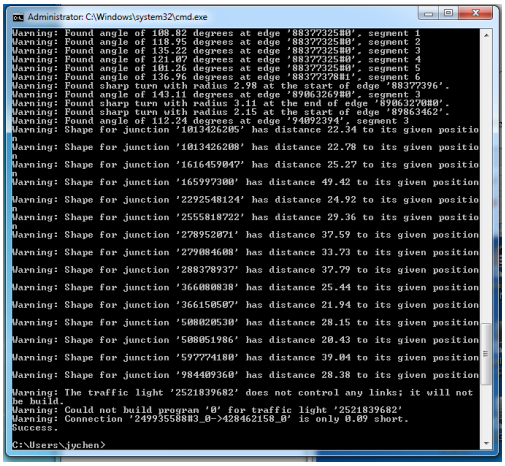


Figure 5.6(a) and Figure 5.6(b) translation from OSM file to SUMO file

Finally, after all modifying task, roundabout map is like the following figure 5.7. By using this map of roundabout map testing, the code of the simulation is more suitable for real life situation. But compare with the road setting in real life, this small roundabout map is still not completed.

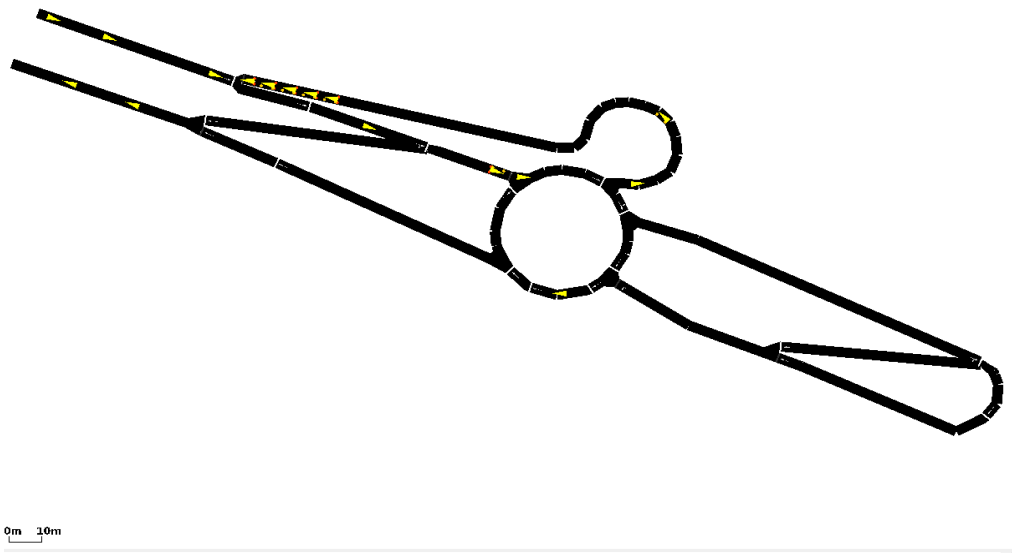


Figure 5.7 Roundabout map

This map is only the special edition of triangle map. It is still far away from the real life situation. So the result got from this map would not be analyzed by this report.

5.3 Block map

Connecting to web site http://www.openstreetmap.org. And search a real map of Excelsior, California, USA. After I select the area I want to test in this report, I save the map as a OSM file.

By the help of JOSM, I translate all the road information into more detailed lines, and show on the interface. And by the help of JOSM, I delete some route information such as non-vehicle route and shipping route. After I modified the OSM file step by step. And finally, I only left 200 vehicles and 100 intersections in order to do the comparison with real map in the later chapter.

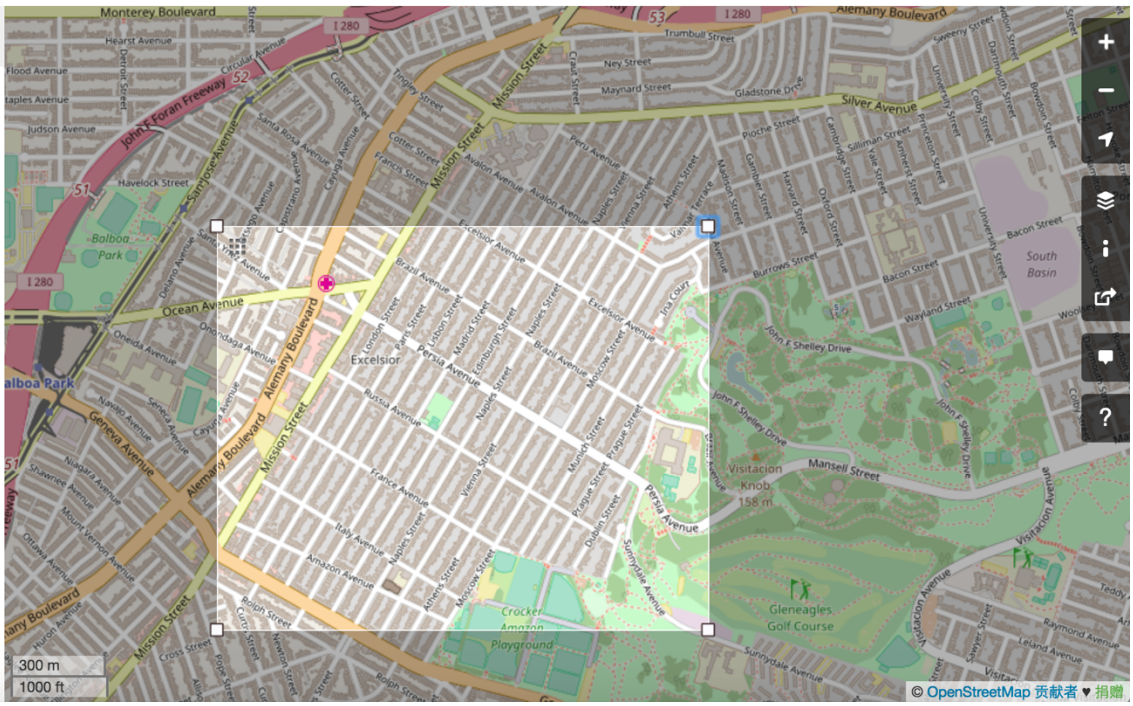


Figure 5.8 Typical block map in real world

Netconvert can change the OSM file into SUMO project by call it in command line. And by the help of OSM, it is obvious that, there are some road traffic facilities which not as same as the previously maps. Due to this characters, this traffic simulation is more likely to that real life situation. For example, the road in this map has different road priorities and there are stop signs on the intersections, in order to ensure the safety of the road, and so on.

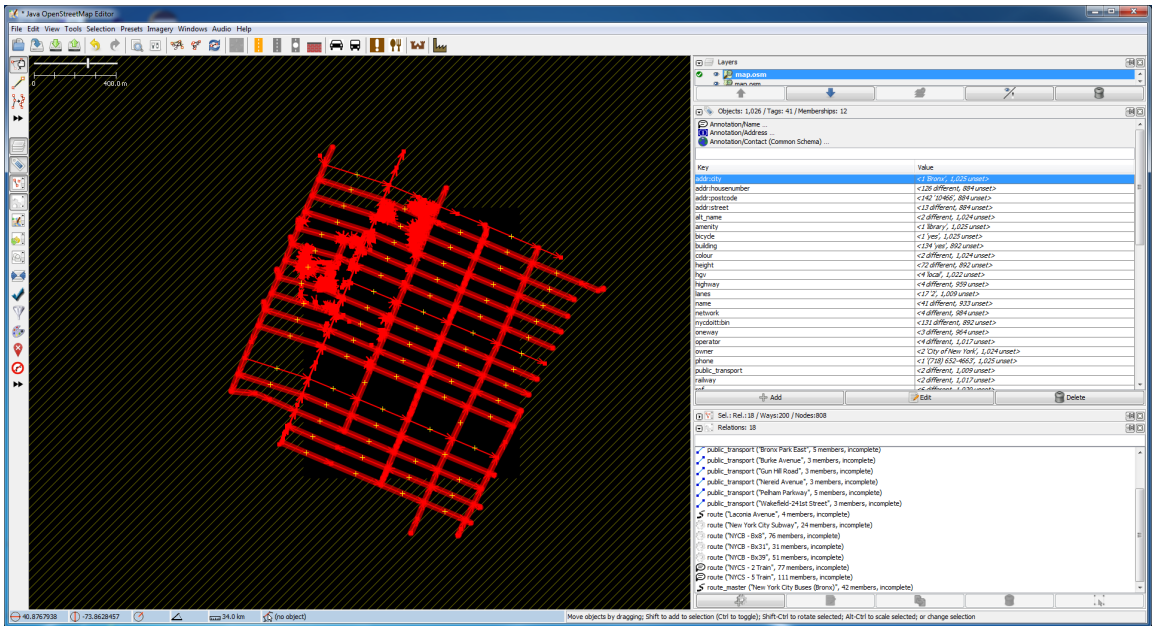


Figure 5.9 OSM file of block map

The final edition of block map is like the following figure 5.10. And in this map, I suppose that there are three density of vehicles such as 100 vehicles, 200 vehicles and 500 vehicles. And all the vehicles must follow the road regulations. Except malicious vehicles, all the testing vehicle must stop for a little while at the junction of the map and the vehicle goes on the east - west direction must let the vehicles goes on north-south direction first. So the result of this block map is as much as possible to be similarity with the situation in real life.

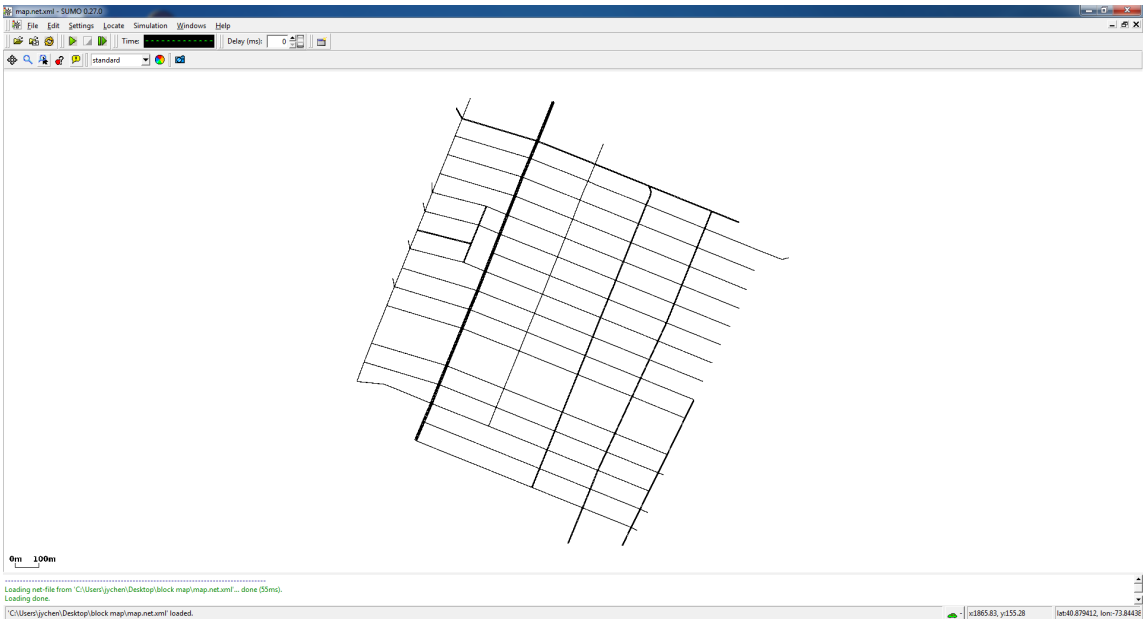


Figure 5.10 SUMO file of block map

5.4 Real Map

Connecting to web site http://www.openstreetmap.org. And search a real map of New York City, New York, USA. After I select the area I want to test in my project, I save the map as a OSM file.

By the help of JOSM, I translate all the road information into more detailed lines, and show on the interface. And by the help of JOSM, I delete some route information such as non-vehicle route and shipping route. After I modified the OSM file step by step. And finally, I only left 200 vehicles and 100 intersections in order to do the comparison with real map in the later chapter.

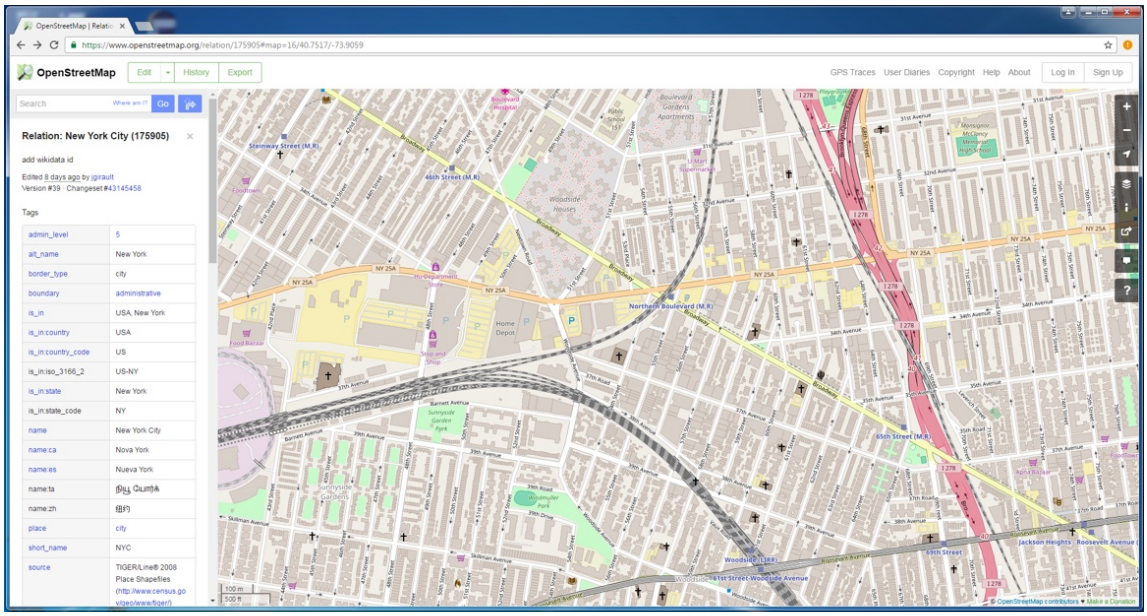


Figure 5.11 Real world map

Netconvert can change the OSM file into SUMO file by call it in command line. Due to the special traffic facilities which simple map does not have, this traffic simulation is more likely to that real life situation. For example, the road in this map has different one-way or roundabout road and there are some traffic lights, in order to ensure the safety and efficiency of the road.

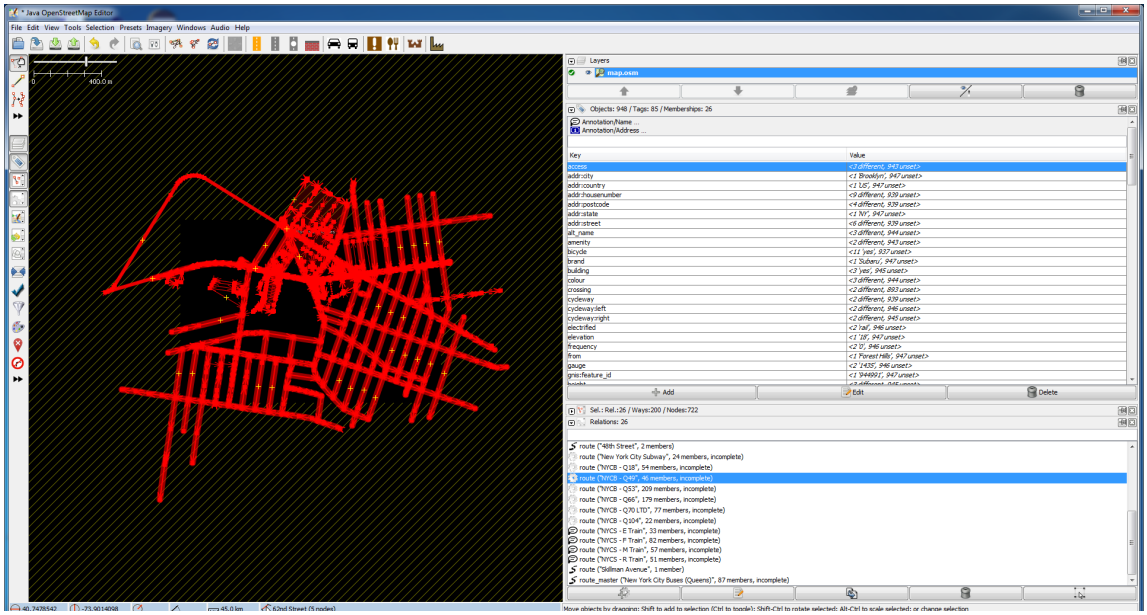


Figure 5.12 OSM file of real map

The final edition of real map is like the following figure 5.13. And in this map, I suppose that there are three different density of vehicles such as 100 vehicles, 200 vehicles and 500 vehicles. And all the vehicles must follow the road regulations. Except malicious vehicles, all the testing vehicle must wait on the junction while the traffic light turns into red and the vehicle goes on some direction must let the vehicles goes on the other direction first (The direction is all depends on the priority of each road). So the result of this real map is as much as possible to be similarity with the situation in real life.

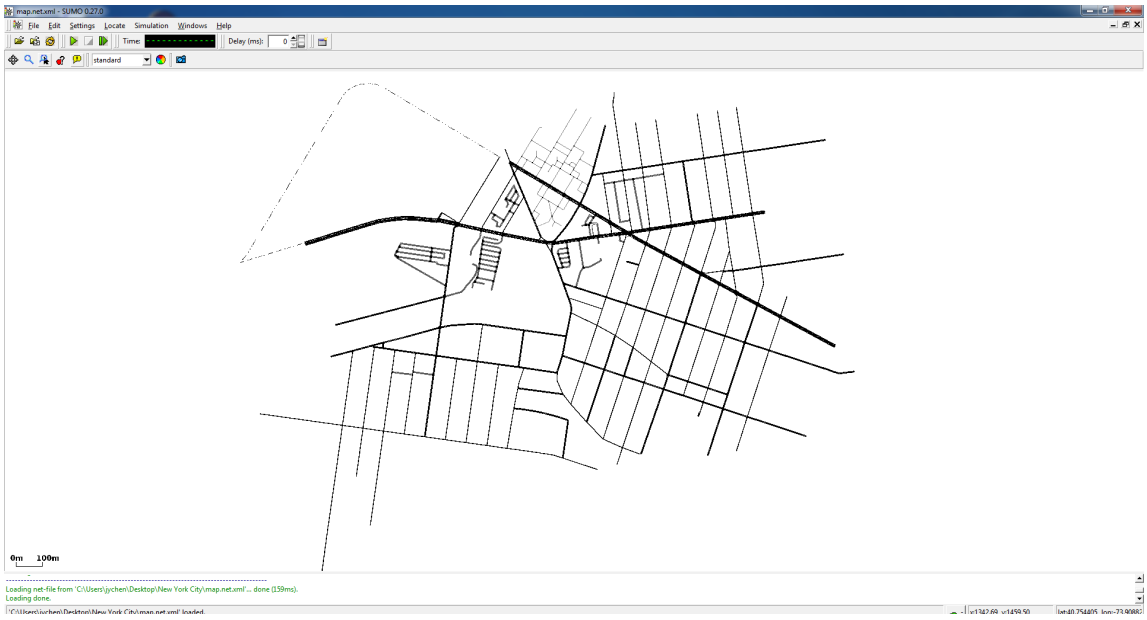


Figure 5.13 SUMO file of real map

5.5 Summary of all types of maps

And the following table 5.1 is the total comparison of the three important map simulation (For the type of simple map, roundabout road and triangle map would not be considered as one of the comparison task). The setting is all about every

detail inside the traffic simulation. For example, “number of intersection”, “number of road”, “number of vehicles”, “vehicle speed”, “detection radius” and so on.

Due to the limitation of simple map, it only simulated with 100 vehicles and short simulation time. But for the convenience of subsequent comparison action, the setting of block map is as same as the real map. And for block map and real map section, there are three set of test data. There are several simulations, such as 100 vehicles testing, 200 vehicles testing and 500 vehicles testing. It aims to testing the impact of vehicle density to the three different methods.

Parameters	Simple map simulation	Block map simulation	Real map simulation
Map type	Simple map	Block map	Real Map
Number of road	3	200	200
Number of intersections	1	100	100
Number of vehicles	100	100/200/500	100/200/500
Vehicle speed	0 ~ 69.5 km/h	0 ~ 69.5 km/h	0 ~ 69.5 km/h
Simulation time	300s	550s	750s
Detection radius	100 m	1000 m	1000 m
Red signal lasting time	N/A	N/A	0 ~ 2.5s
The probability of running red lights	N/A	N/A	20% ~ 25%

Table 5.1 Summary of all types of maps

6 Experiment

6.1 System configuration

Trust Level System can efficiently reduce the impact of malicious user and improve the reaction time of detection in the Ac-hoc Network. To build the simulation system, I try different sets of configuration (Linux plus SUMO 0.25.0 plus python 3.0 and Linux plus SUMO 0.23.0 plus python 3.0) and I choose to use the set of test which is windows7 plus python 2.0 and SUMO 0.27.0 finally.

Environment variable:

PATH: C:\Python27\; C:\Python27\Scripts; C:\Program Files\DLR\Sumo\tools\ C:\Program Files\DLR\Sumo\bin\

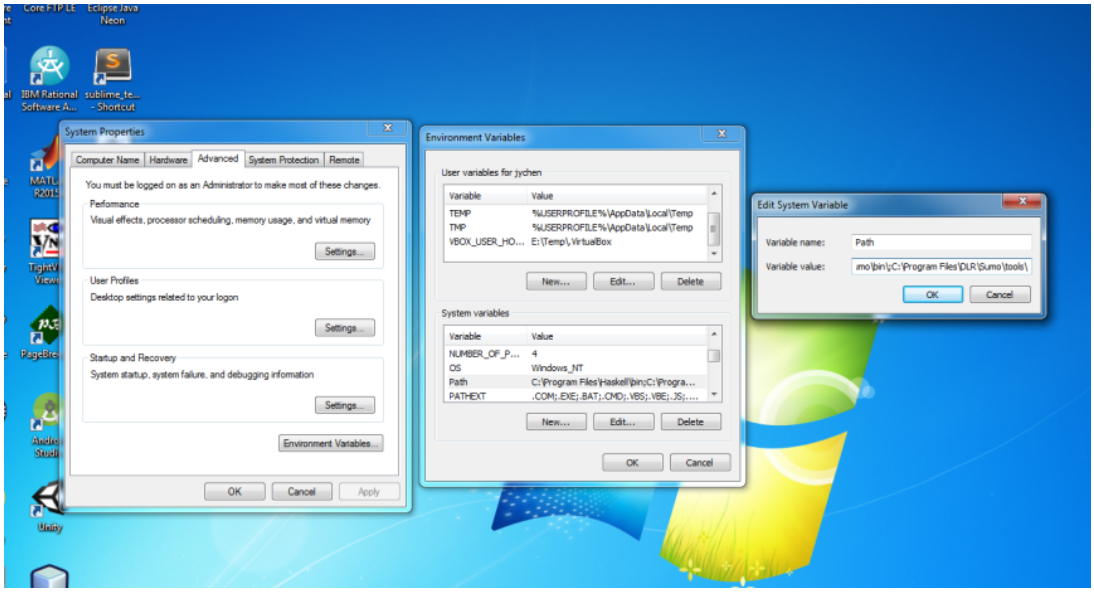


Figure 6.1 Environment variable (1)

SUMO_HOME: C:\Program Files\DLR\Sumo

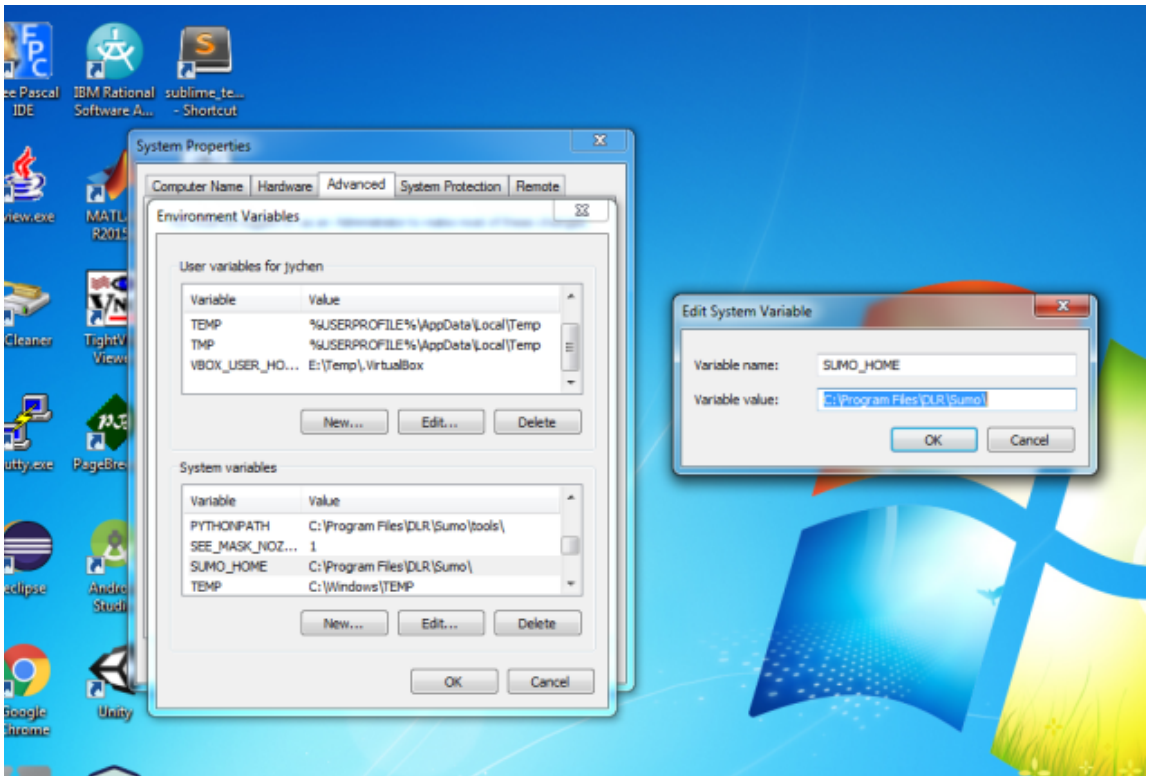


Figure 6.2 Environment variable (2)

Code in the following image is the test demo for SUMO environment.

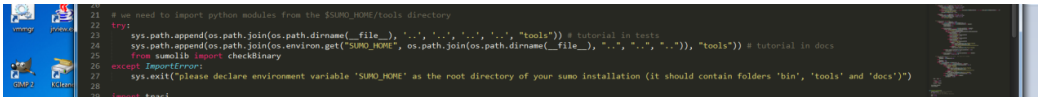


Figure 6.3 Test code of system configuration

If the port used in the test is occupied, then the command line will display the following bug warning and the SUMO-Gui will show up a socket error warning.

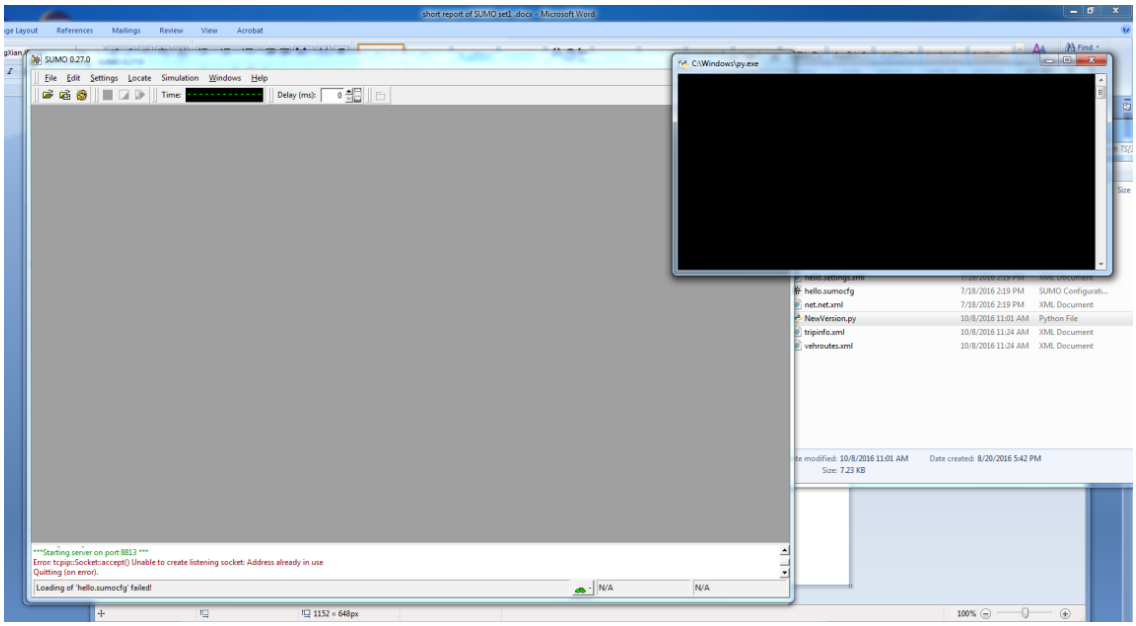


Figure 6.4 Test step of system configuration

If the combination of the version of python, SUMO and system is not suitable, then the command line will give the following warning out like the following figure 6.5 and figure 6.6.

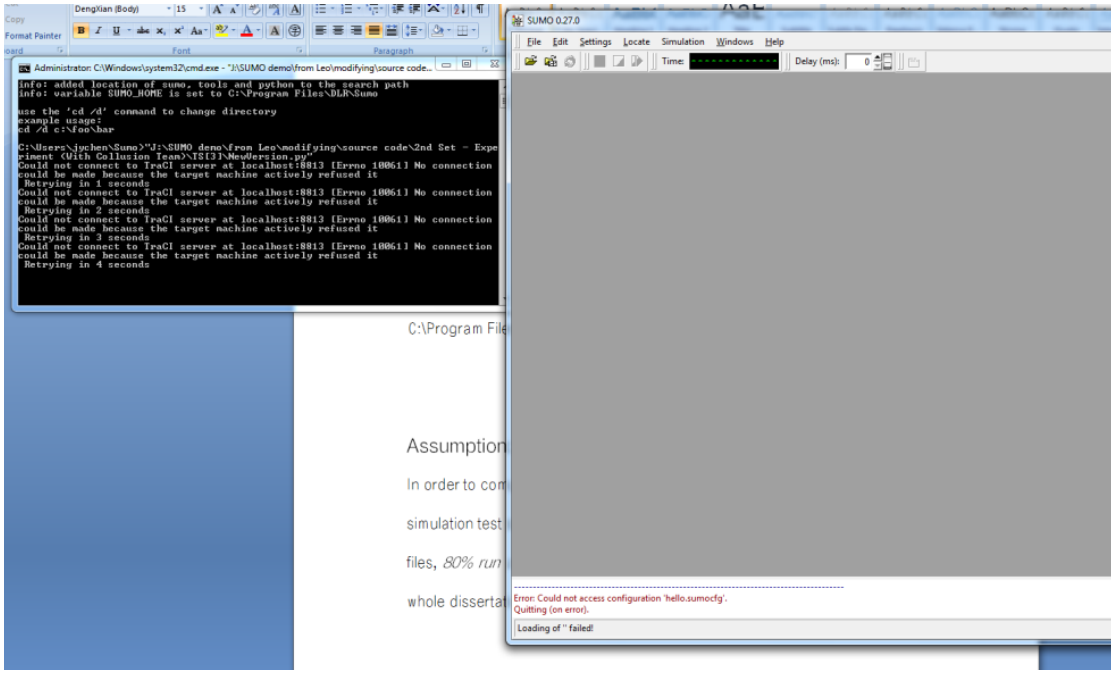


Figure 6.5 Test result of problem situation

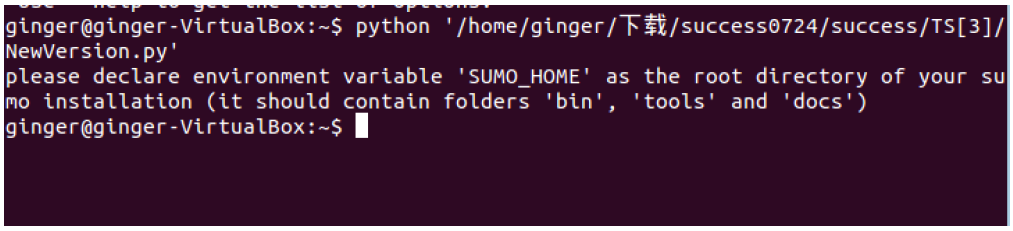


Figure 6.6 Error message of SUMO system configuration

If the SUMO test system set up successfully, the command line won't display anything and SUMO-Gui will be launched up.

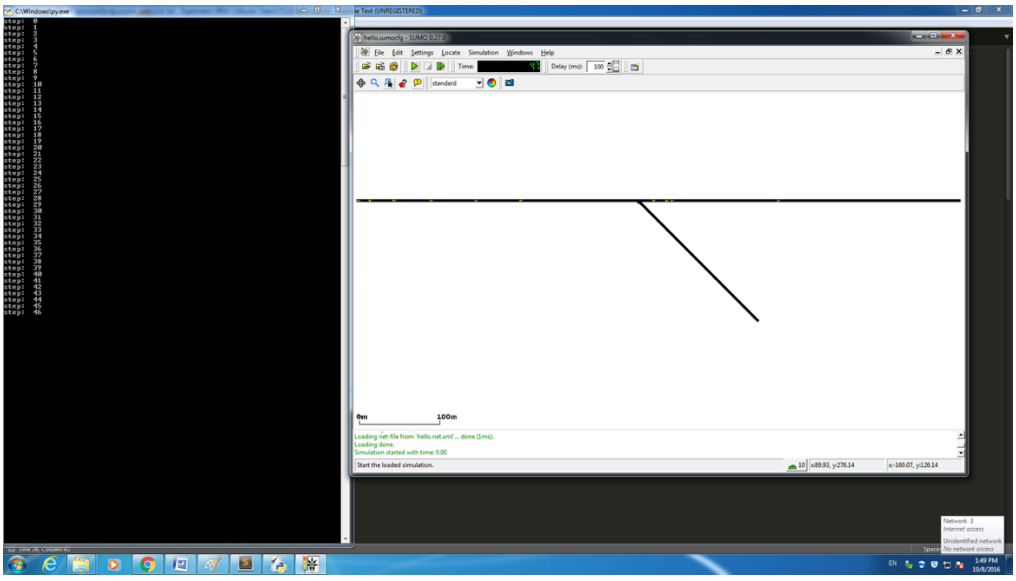


Figure 6.7 Successful test result

6.2 Preparation of test threshold

For the three method I descript in the previously chapter, threshold method is a broadly method. Due to this situation, I try to pick up three suitable thresholds during the main situation of this threshold method. And the following reparation of the experiment is the selection of the three threshold.

In this part of preparation of experiment, I test the variables from 1 to 8 to be the threshold of threshold methods separately. And set up the graph of “distance-number of victims” and “distance-first of non-victim” separately, in order to see the different trends of different thresholds. The result of this test is on the following figure 6.8 and figure 6.9.

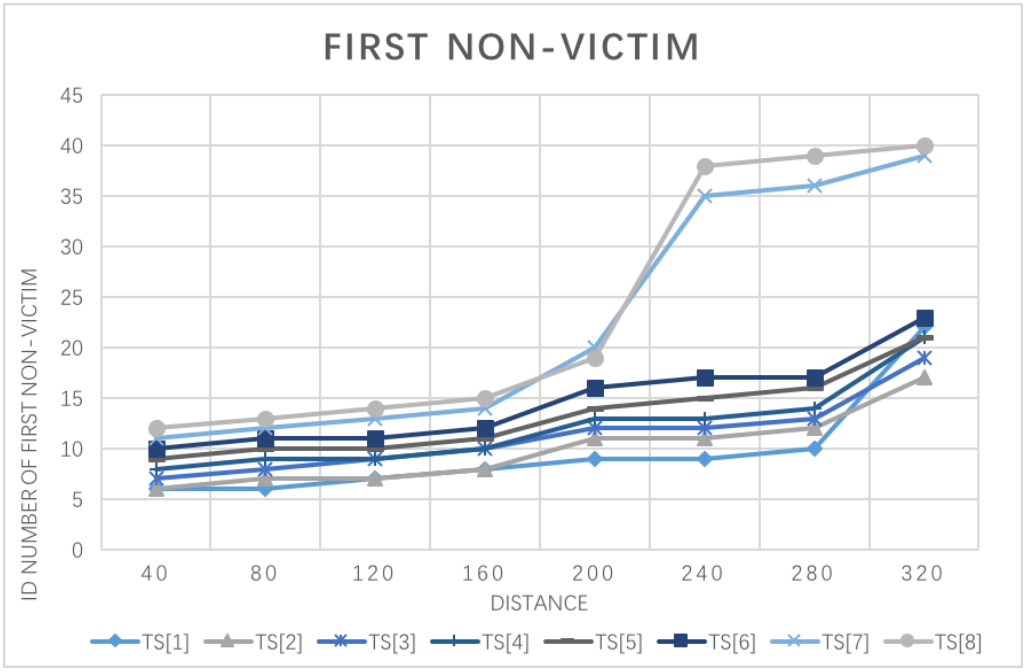


Figure 6.8 Result of re-test in vehicle id of first beneficiary

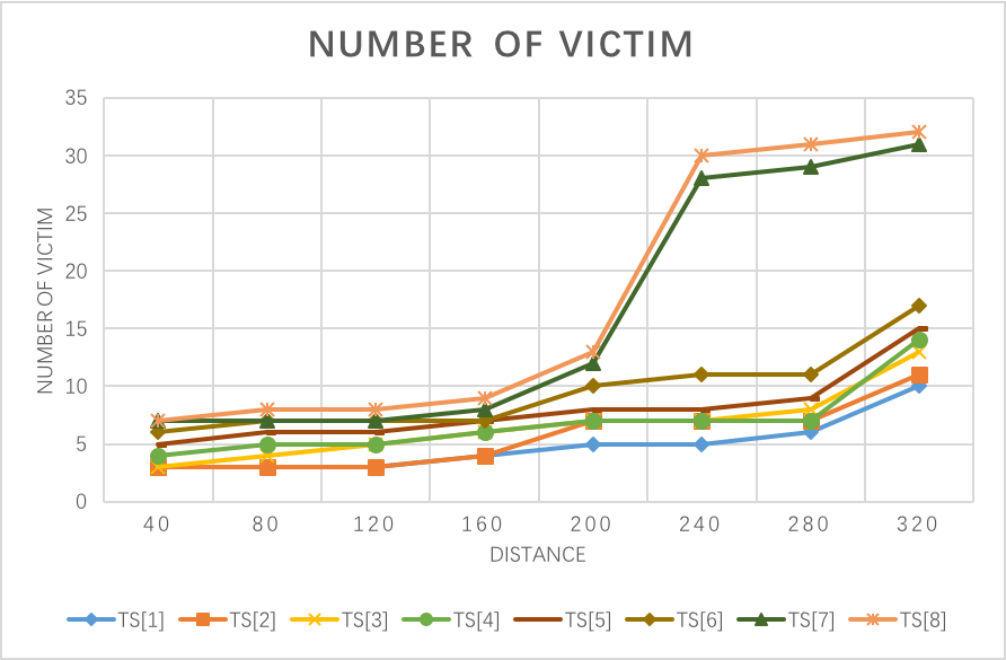


Figure 6.9 Result of re-test in the number of victim

By the visually help of the above figures, I get the knowledge that by the increase of thresholds that trend increased follow a certain law. Due to this findings, I can simplify the future task by not consider about the threshold shows the similar result. And in the next sets I will only test the threshold at TS [3], TS [5], TS [8] while I compare with the three methods.

Furthermore, I also confused by the great increasing of the TS [7] and TS [8] while the distance parameter from 240 to 320. To get the reason of these confusion, I make the two following assumptions.

6.2.1 Assumption one

For the preparation of the experiment part, it is not necessary to get the result from all of the five random files. I only use one of the five random files. Due to this reason, the great increasing of the result might because of the vehicles' trip information I design in the random file.

```
46    <vehicle id="36" type="Car" route="route1" depart="149" />
47    <vehicle id="37" type="Car" route="route0" depart="150" />
48    <vehicle id="38" type="Car" route="route0" depart="153" />
49    <vehicle id="39" type="Car" route="route0" depart="160" />
50    <vehicle id="40" type="Car" route="route0" depart="163" />
51    <vehicle id="41" type="Car" route="route0" depart="172" />
52    <vehicle id="42" type="Car" route="route0" depart="180" />
53    <vehicle id="43" type="Car" route="route0" depart="185" />
54    <vehicle id="44" type="Car" route="route0" depart="194" />
55    <vehicle id="45" type="Car" route="route0" depart="198" />
56    <vehicle id="46" type="Car" route="route0" depart="204" />
57    <vehicle id="47" type="Car" route="route0" depart="208" />
```

Figure 6.10 Assumption one

Figure 6.10 is the trip information from vehicle 37 to vehicle 47 which the great increasing happened. And I find that during this time in the situation, all the normal vehicles randomly choose to go on the main road. By this situation, before the warning message had been received by the after vehicles, there must be a great increasing of victims and so do the result of "first non-victims".

6.2.2 Assumption two

In order to avoid the influence caused by the random file, I test TS [5], TS [6], TS [7] and TS [8] by another random file.

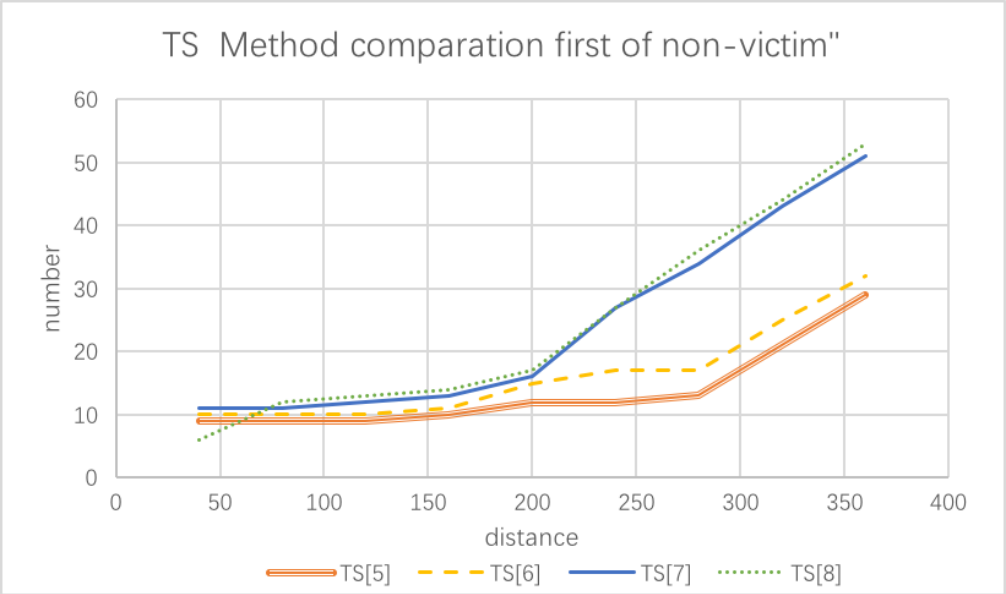


Figure 6.11 Assumption two in the number of victim

By the help of the new test, TS [5] and TS [6] also has an increase, so I add the "distance" (distance is the length road from the intersection to the crash position) parameter to 360 units in order to find the potential rules.

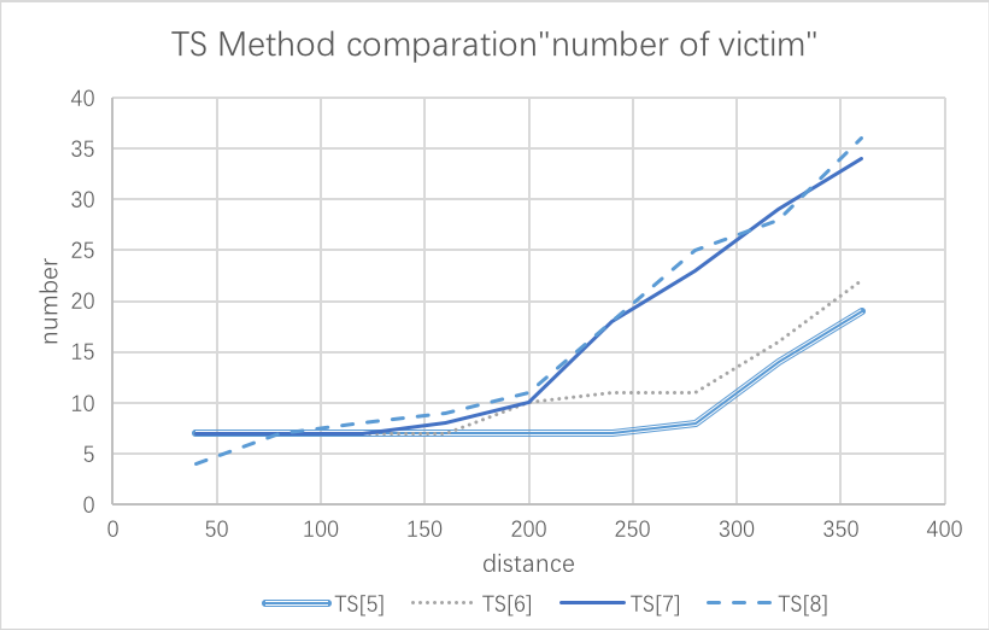


Figure 6.12 Assumption two in vehicle id of first beneficiary

After I modified the test configuration, I got the result and draw a new chart above. Both of "number of victims" and "first non-victim" for TS [5] and TS [6] has an obvious increasing as same the TS [7] and TS [8] after "distance" grows to 360 units. And the previous problem is because of all the threshold methods have an increasing result but the TS [7] and TS [8]'s increasing appears earlier than the other methods.

So by the verification of these two assumptions, the previous confusion is both because of the random file and the "distance" parameter.

6.3 Simple map

6.3.1 Situation without the collusion team of malicious users

In this set, I assume there is no malicious user in these 6 test methods. And the test result will show the efficiency of these 6 methods' ability of verifying a real warning message in Ad-hoc network. In this set of simulation, I set 101 vehicles, the first is defined as the traffic crash generator, I won't collect the information of the first vehicles. And the result will show the efficiency by the two values "number of victims" and "first non-victim".

Number of road	3
Number of intersection	1
Number of normal vehicles	100
Number of crash generator	1
Number of malicious users	0
Situation in the simulation	None
Crash duration time	500
Percentage of the verified warning	100%

Table 6.1 Test setting of simple map

I set up five random travelling route files. The speed of each vehicles is a random value from 50 Km/h to 60 Km/h (the speed is an assumption I wrote previously). The first vehicle with a vehicle ID 0 will be seemed as a crash generator and all the messages send by the generator would not be counted in the Trust Level Warning System. And in this set, all the 100 vehicles are normal users. Due to these prerequisites, the result in this set can only show the responding speed of each methods which is the efficiency of each methods.

The results are the average value of these five files. The red line is the result of the id number of first non-victim. The blue line is the number of the victims in crash. X-axis is the distance between the intersection and the crash position. Y-axis is the number of the victims or the id number of the first beneficiary.

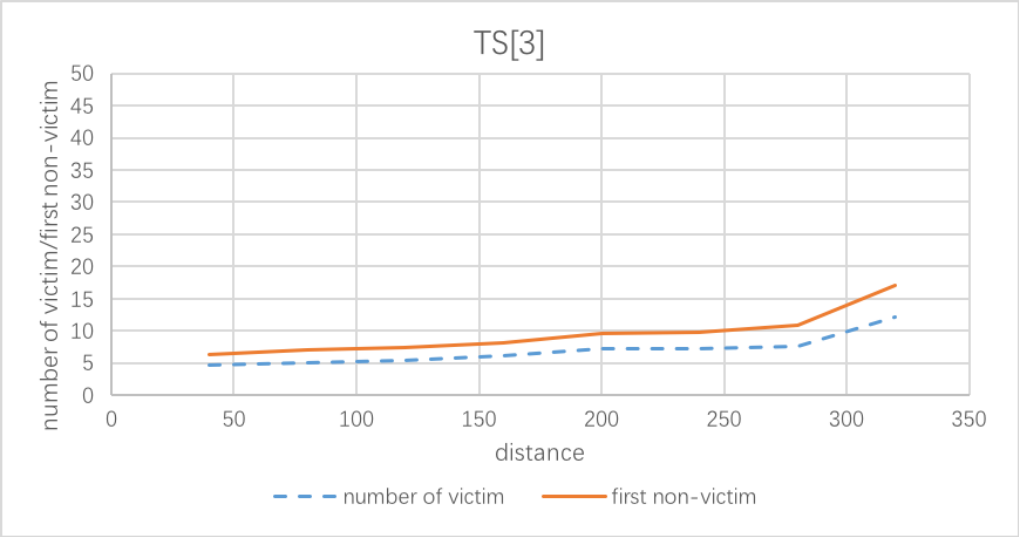


Figure 6.13 Result of simple map with scenario one (TS [3])

In the figure 6.13 above, it's obvious that if the distance larger than 275, there will be a surge in the rate of growth of "the number of victims" and "first beneficiary". And while the distance is smaller than 300, the value of "number of victims" is all smaller than 10.

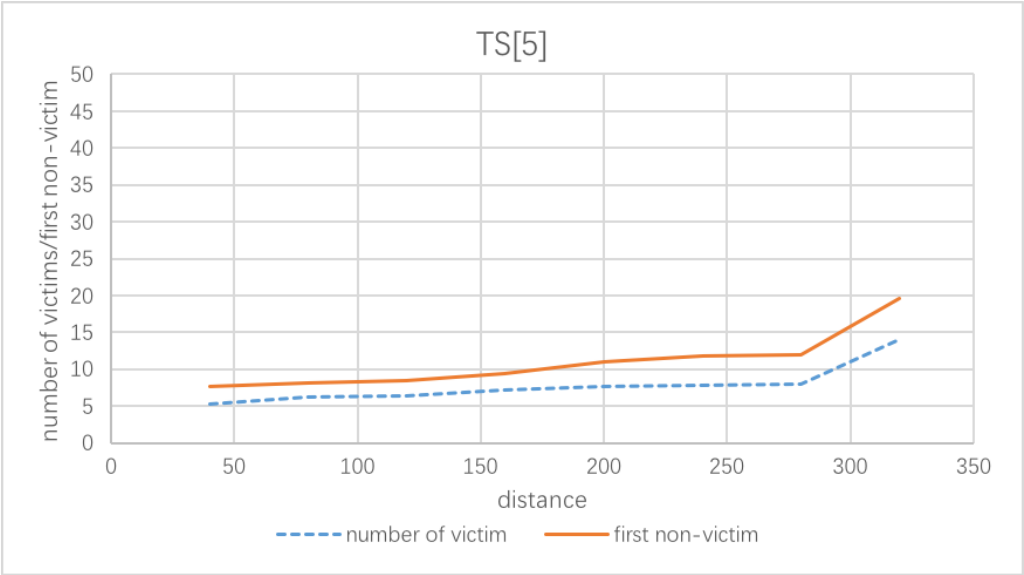


Figure 6.14 Result of simple map with scenario one (TS [5])

In the figure 6.14 above, it's obvious that the distance of 280 is a turning point, the slope grows a lot in "the number of victims" and "first beneficiary". Compare with the result of TS [3], TS [5] has a larger value of both of "the number of victims" and "first beneficiary".

Figure 6.15 Result of simple map with scenario one (TS [8])

In the figure 6.15 above, the turning point of TS [8] appears earlier than the two methods above. The surging point is while the distance is equal to 200. And after the distance is larger than 250, the slope decreases but the two values of "the number of victims" and "first beneficiary" are much bigger than the previously two thresholds.

By the figure of the above three thresholds, it is obviously that, as the threshold increases both of the "the number of victims" and "first beneficiary" has a larger value than before. So it can be proved that, the larger the threshold value the method has, the less efficiency it is.

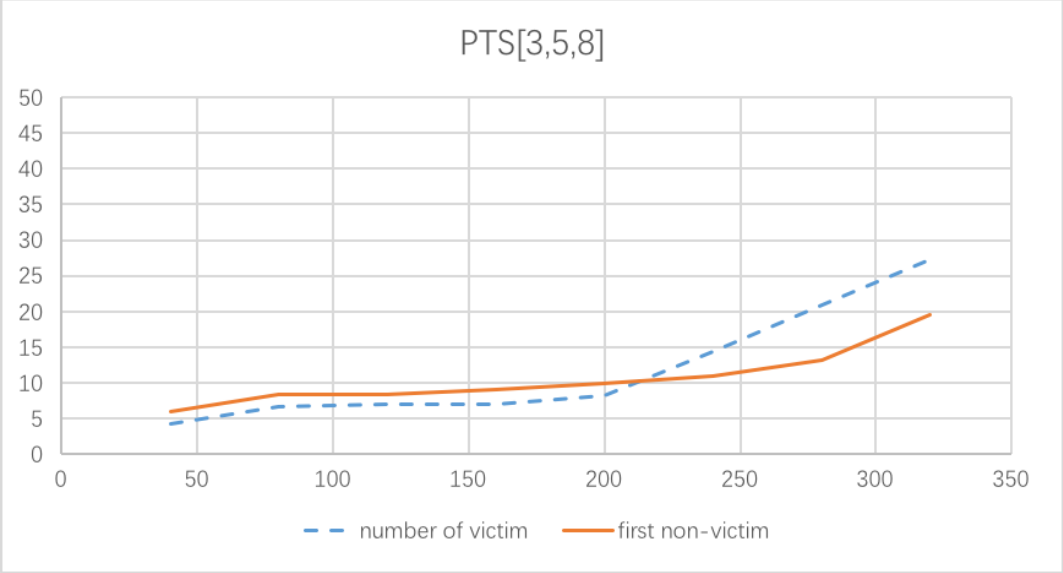


Figure 6.16 Result of simple map with scenario one (PTS [3,5,8])

Compare with threshold method, parallel threshold method is a complicated method. It starts sending warning message earlier than TS [5] and TS [8], but it is opposite that only 30% of rear vehicles trusts the warning message while three verifiers have been received. The result is shown that the values of "the number of victims" in PTS [3,5,8] is not much difference than TS [8]. And the values of

"first beneficiary" is almost equal to TS [5]. In conclusion, for the situation without collusion team, the efficiency of parallel threshold method is worse than threshold method while the it's threshold is less than 5.

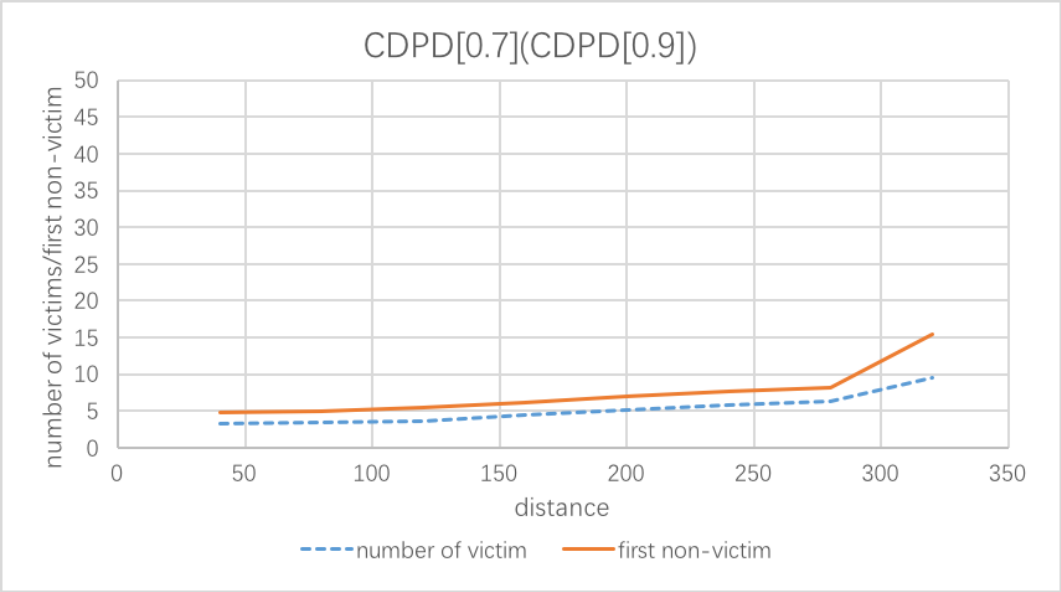


Figure 6.17 Result of simple map with scenario one (CDPD method)

In the situation that there is no malicious user inside the traffic simulation, CDPD [0.7] is equal to CDPD [0.9], because all the message broadcasted through the VANET is positive and true. Once a verifier has been received by the following vehicle, this verifier will turn into a real warning message and broadcast in the trust level system. Due to this reason, the special result come from the CDPD is show on the above figure 6.17.

6.3.2 Situation without the collusion team of malicious users (80%)

For the situation in the real world, users sometimes do not follow the GPS's guideline. So I modified the scenes in Set1 and set a rule that only 80% of the normal users will trust the warning given by the Ad-hoc system and 20% of the

normal users won't turn into the pass way. So in this set, the value of "the number of victims" will randomly larger than the value I got in set 1.

Number of road	3
Number of intersection	1
Number of normal vehicles	100
Number of crash generator	1
Number of malicious users	0
Situation in the simulation	None
Crash duration time	Whole simulation
Percentage of the verified warning	80%

Table 6.2 Test setting of simple map with scenario one

This set is a modification of set 1. I also set up five random travelling route files. In these files, there are 101 vehicles in each file and the speed of each vehicles is a random value from 50 Km/h to 60 Km/h (the speed is an assumption I wrote in the previous chapter). The first vehicle with a vehicle ID 0 will be seemed as a crash generator and all the messages send by the generator would not be counted in the Trust Level Warning System. And in this set, all the 100 test vehicles are normal users. But different from set 1, even if the result counted by the trust level system meets the predefined threshold, there will be only 80% of the later vehicles trust and turn into the pass way. So in this set, all the value of the result is larger than the value I got from set 1. That is all due to the 80% random situation. In this way, this set may similar to the real world situation.

The results are the average value of these five files. The red line is the result of the id number of first non-victim. The blue line is the number of the victims in

crash. X-axis is the distance between the intersection and the crash position. Y-axis is the number of the victims or the id number of the first non-victim.

It is obvious that all the six methods have a very large start of value “number of victims”. Due to the reason 80% trusted of the warning message sent by the Ad-hoc system, if the distance from the intersection to the crash position is small enough (like the parameter of the start of the set test is only 40 units) it will cause all the later vehicles fall into the crash. Because before the verified message has been received by the later vehicles, they will miss the intersection already. And before the distance smaller than 120 units, both of value “the number of victims” and “vehicle ID of first beneficiary” are both larger than the same value in set 1.

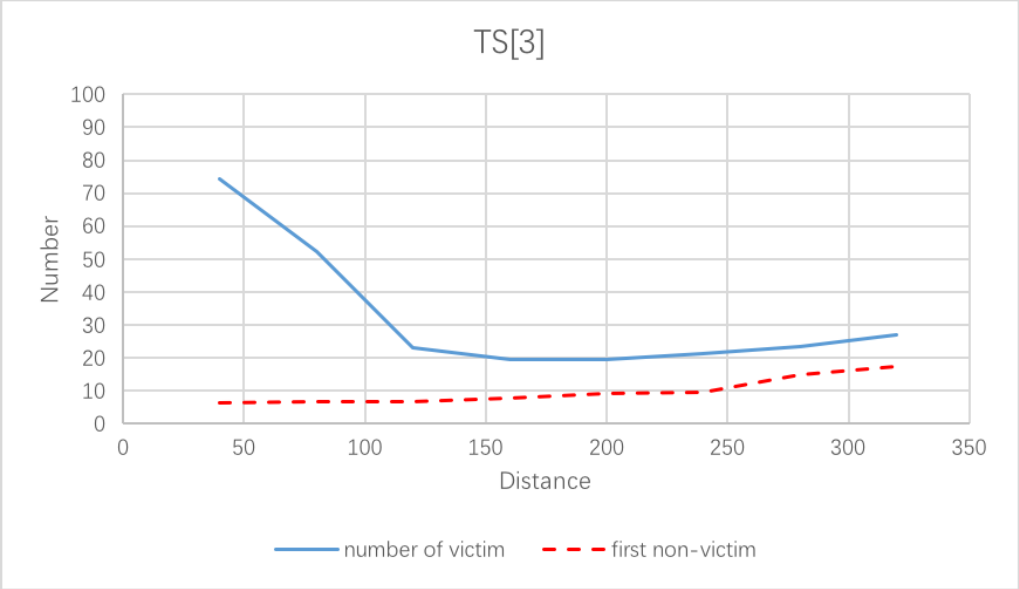


Figure 6.18 Result of simple map with scenario one (TS [3]) 80%

Compare with the result got from previously simulation in figure 6.18, this traffic simulation gets a worse result. When the distance is less than 120, the figure shows a great value of result in “the number of victims”, it means almost all the rear vehicles fall into the crash position. And for the result while the

distance is larger than 120, it is a parallel line with the previous situation but higher than it.

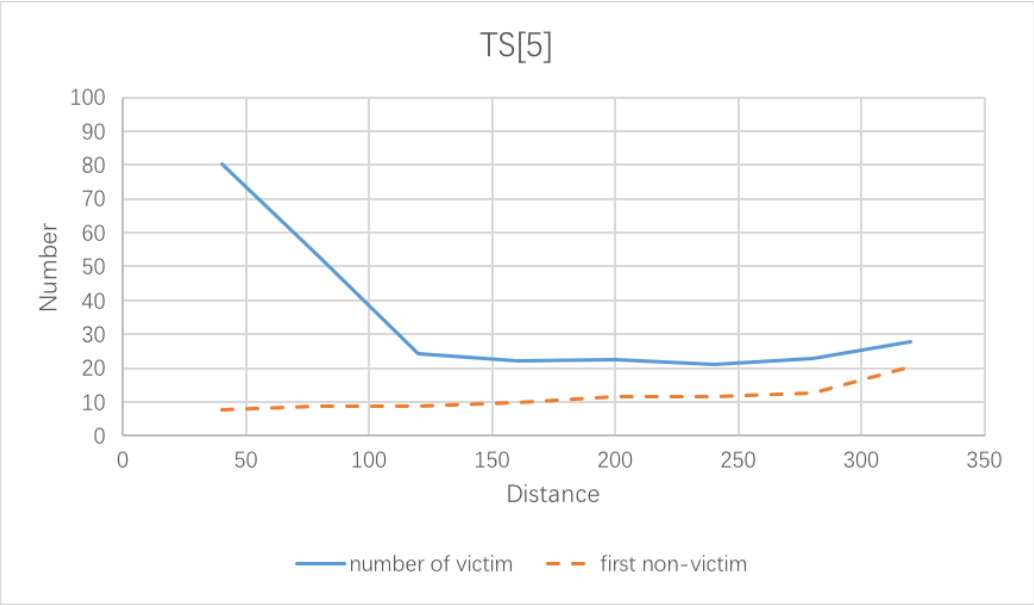


Figure 6.19 Result of simple map with scenario one (TS [5]) 80%

And for the situation for TS [5], it has a same behavior with TS [3]. All victims before the distance large than 120 fall in to the traffic crash. And this trouble becomes better after the distance is larger than 150.

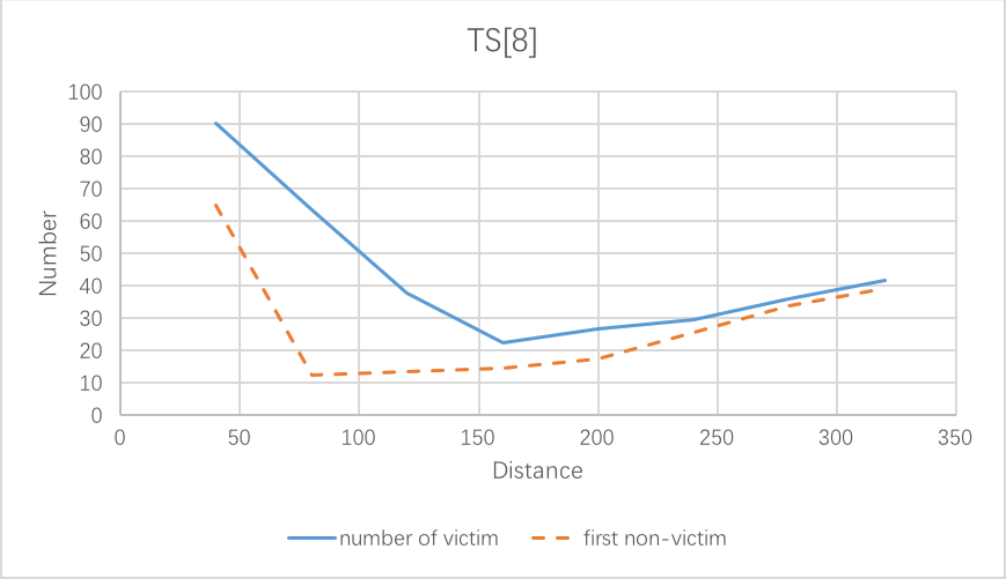


Figure 6.20 Result of simple map with scenario one (TS [8]) 80%

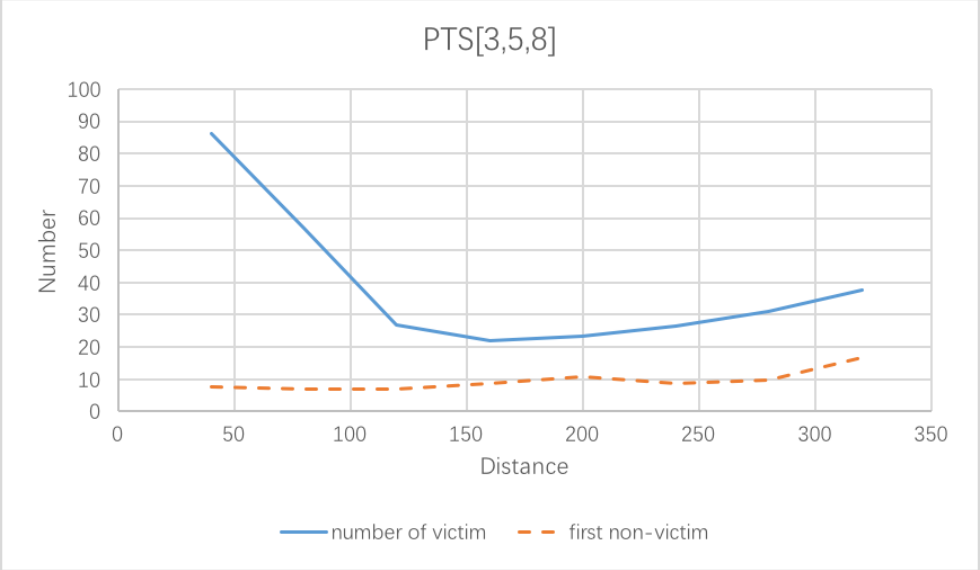


Figure 6.21 Result of simple map with scenario one (PTS [3,5,8]) 80%

The same situation also happened in PTS [3,5,8] too.

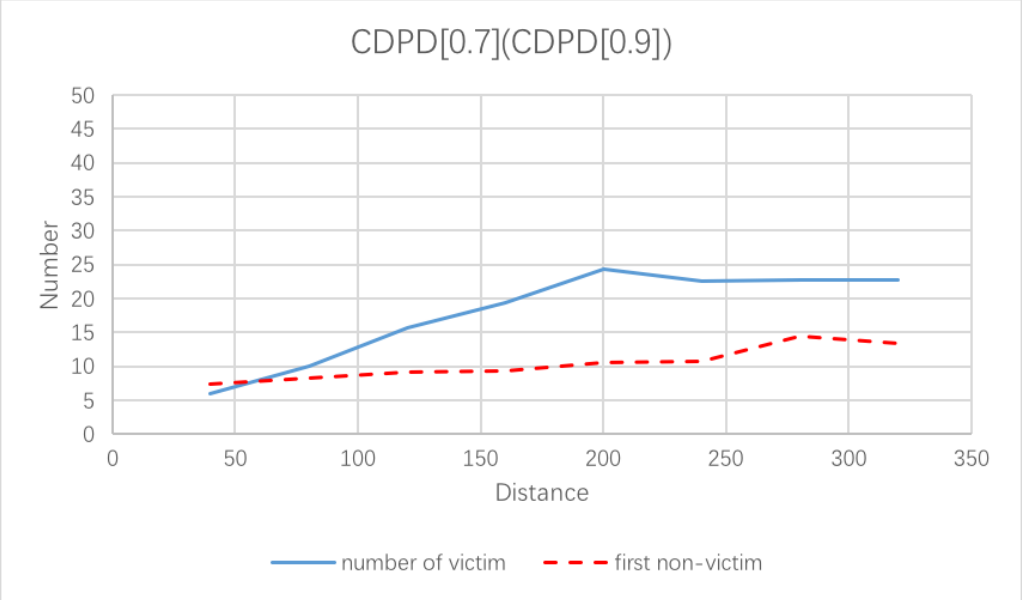


Figure 6.22 Result of simple map with scenario one (CDPD method) 80%

Due to the special assumption that there is no malicious user inside of this simulation, CDPD is no depending on its threshold and all the result is equal to TS [1].

In conclusion, the situation without malicious users and result will not be affected by collusion teams. So, in this situation, I would not consider about the impact of the trustable ability of the system. So the CDPD [0.7] and CDPD [0.9] (which equal to TS1[1]) has the highest efficiency than any other methods during the simulation time.

6.3.3 Situation with collusion team and a fake positive warning message

For situation with collusion team and a fake positive warning message, resemble the situation in real world more, this report considers more about the influence of the team of malicious users or even a detection error. This set focus on the ability of Ad-hoc network retrieving an existing fake warning message. I broadcast a warning message in the network before the start of the simulation (I assume this fake warning message is given by a team of malicious users and all the users in the simulation is normal) and the result is to find how long will it cost to change the fake positive warning message into negative. The efficiency and trust will be shown as the values of “the number of victims” and “the vehicle ID of first beneficiary”.

Number of road	3
Number of intersection	1
Number of normal vehicles	100

Number of crash generator	Depend on the verified method
Number of malicious users	Depend on the verified method
Situation in the simulation	Fake warning without real crash
Crash duration time	Whole simulation
Percentage of the verified warning	80%

Table 6.3 Test setting of simple map with scenario two

In this set, I only set a fake warning message to test the threshold method. For parallel threshold method, the parameter is warning message with threshold TS [3], TS [5], TS [8]. The parameter in CDPD method is the number of malicious users, from 5 malicious users to 7 malicious users. All the result is the average value from the five random files.

The results are the average value of these five files. The red line is the result of the id number of first non-victim. The blue line is the number of the victims in crash. X-axis is the distance between the intersection and the crash position. Y-axis is the number of the victims or the id number of the first beneficiary.

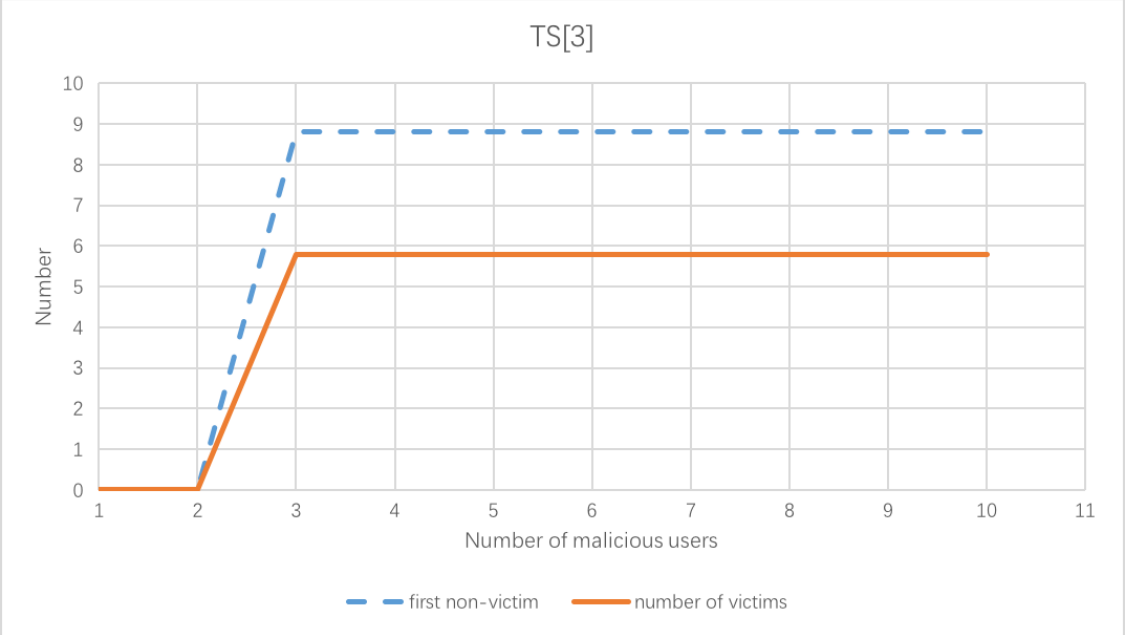


Figure 6.23 Result of simple map with scenario two (TS [3])

Due to the restrictions of malicious users and the crash position. TS [3] shows a really stable result that from the beginning to the end of the traffic simulation, the number of the victims fall into the crash does not change.

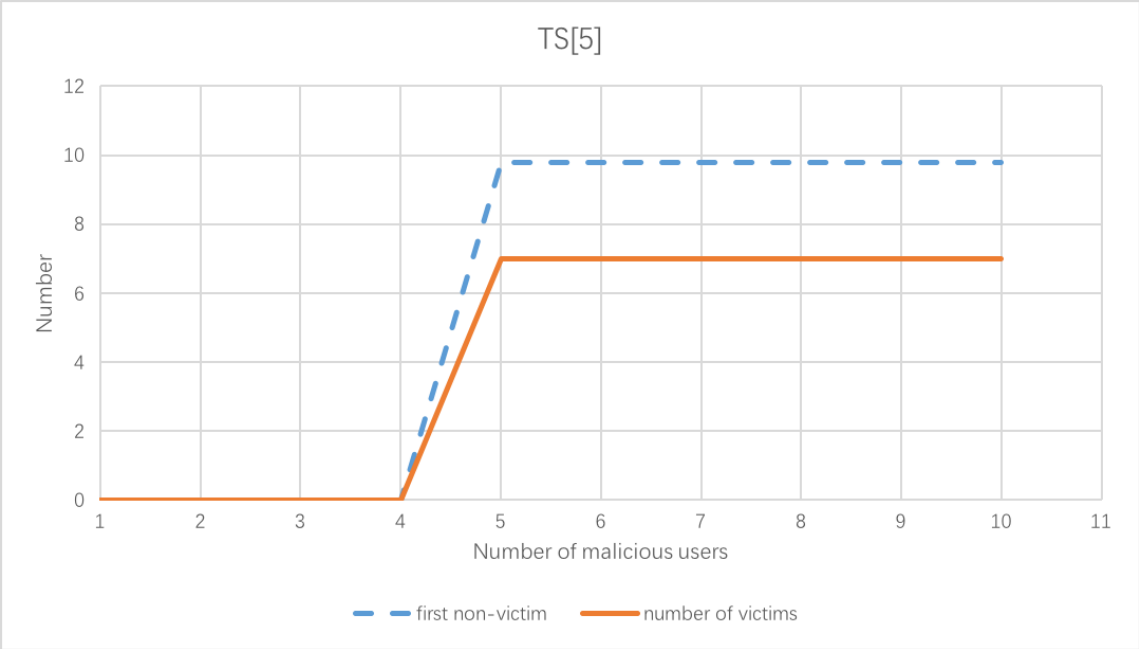


Figure 6.24 Result of simple map with scenario two (TS [5])

Furthermore, by the result of TS [5], after the number of malicious users is enough to send a verified warning message, the number of victim will keep at a fixed value. The fixed values are both larger than TS [3] in “the number of victims” and “vehicle ID of first beneficiary”.

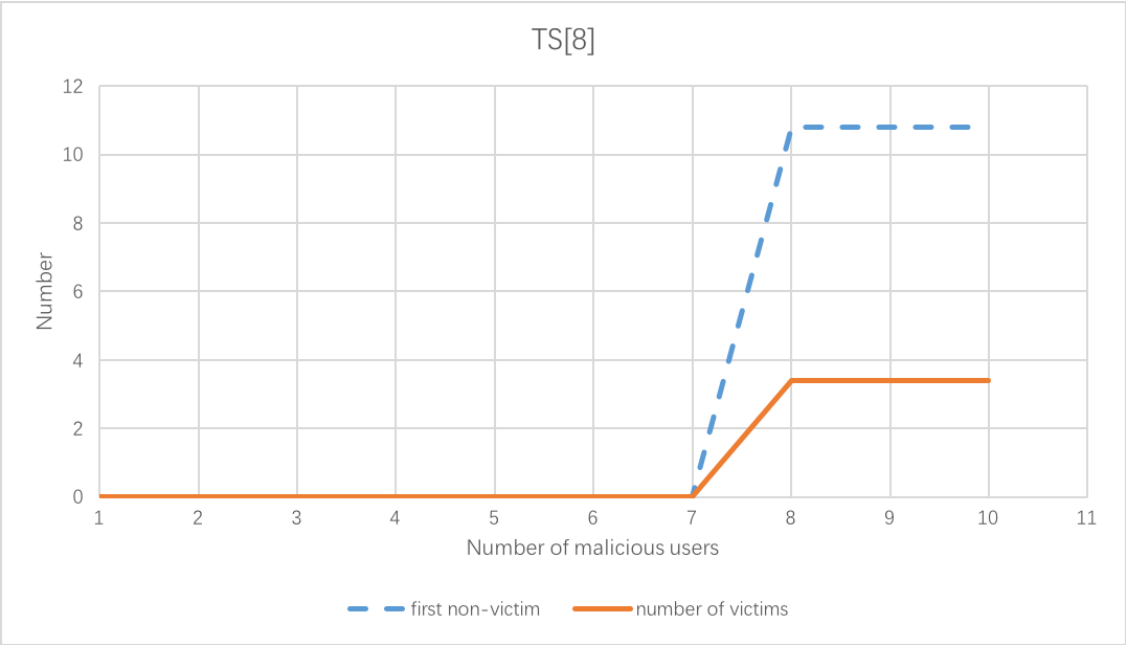


Figure 6.25 Result of simple map with scenario two (TS [8])

Due to the situation with collusion teams, the safety of the system must be considered on more proportion. Compare with the method with threshold 3 and 5, TS [8] have a less value of “the number of victims”, it shows a much better responds of the safety of the system. And the figure shows a high value of “vehicle id of first beneficiary”, it is due to this assumption of this situation, the result number must subtract the number of malicious users in this traffic simulation. After the further calculation, it can be proved that, in the situation with malicious users, the higher the threshold in threshold method the better result of “the number of victims” and “vehicle ID of first beneficiary” it has.

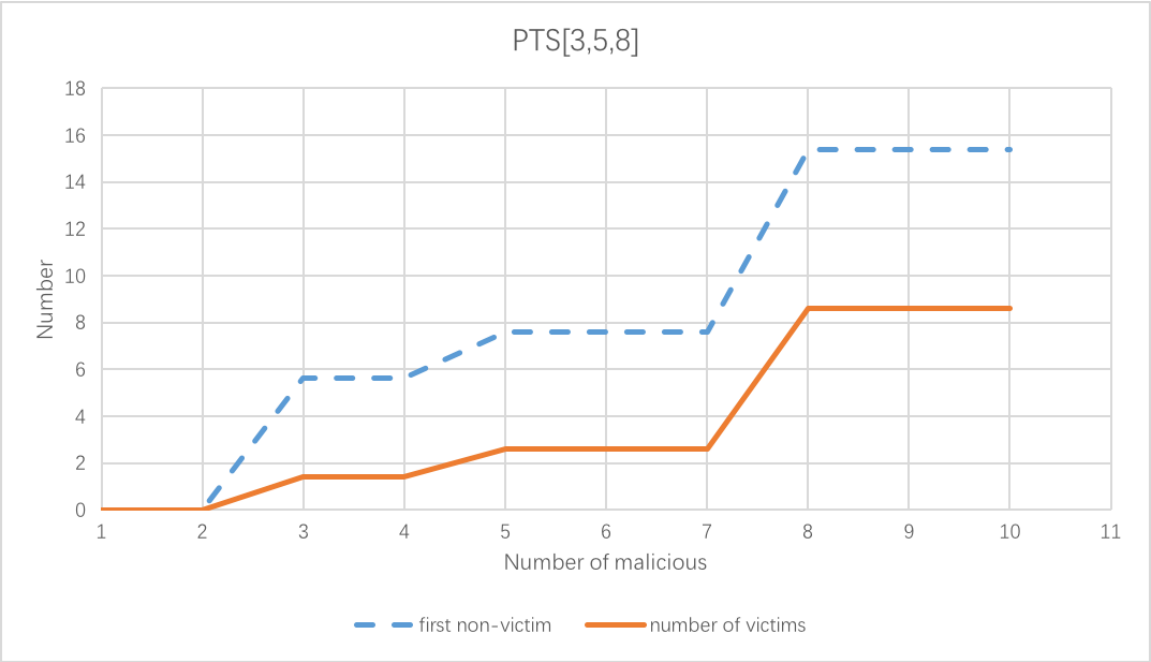


Figure 6.26 Result of simple map with scenario two (PTS [3,5,8])

Compared with the previous method, PTS does not have an obvious numerical advantage, but to compare the result in each part (The different parts are divided by different thresholds such as 3, 5 and 8) and consider both of the two values of “the number of victims” and “vehicle ID of first beneficiary”, PTS really has a stable, trustable and safe warning verify abilities.

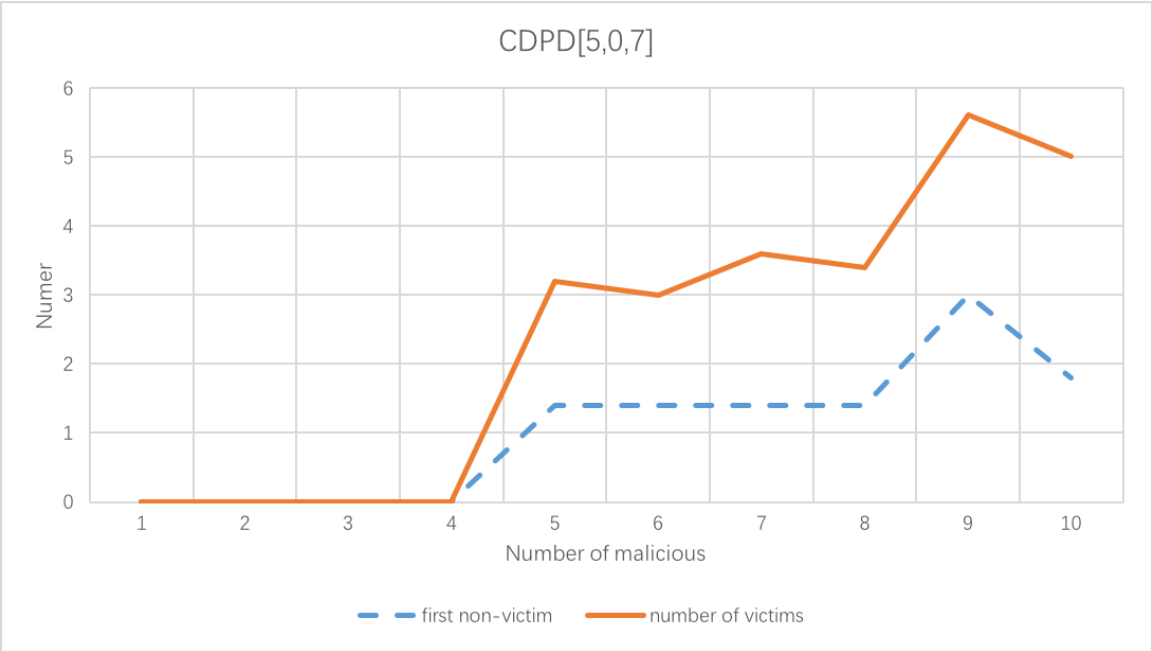


Figure 6.27 Result of simple map with scenario two (CDPD [0.7])

In CDPD method, the result has a value from the start to the end of the simulation. The maximum value in the whole simulation is all smaller than the result in the previously methods. So the efficiency and credibility of this method is better than the previous method.

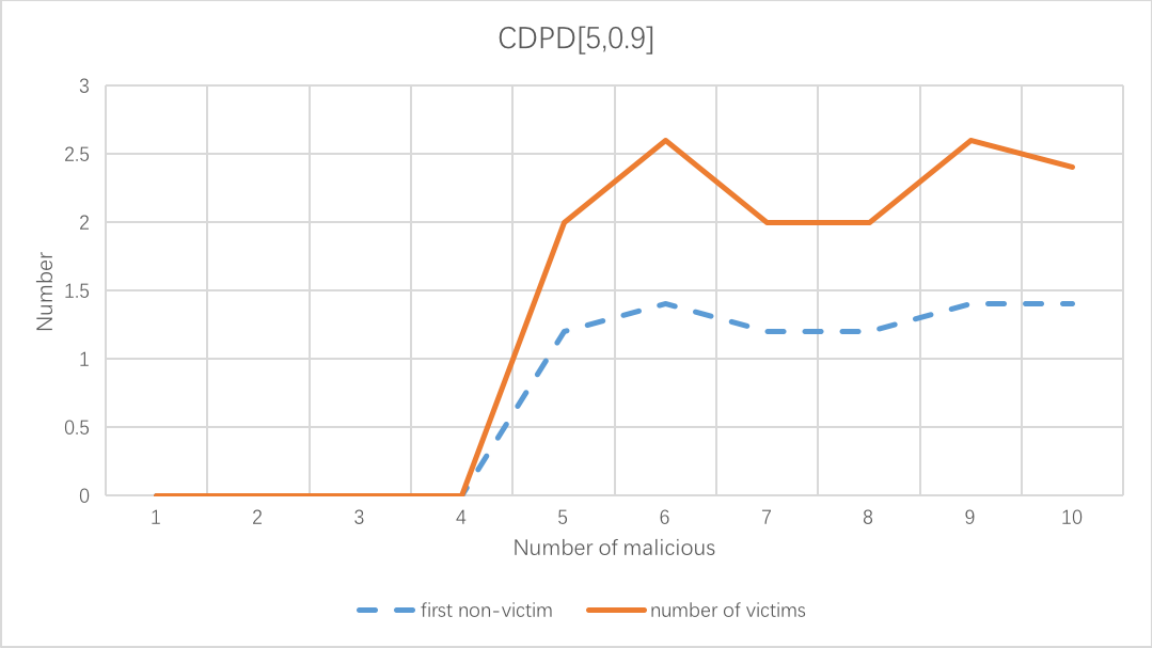


Figure 6.28 Result of simple map with scenario two (CDPD [0.9])

By compare the CDPD [5,0.7] and CDPD [5,0.9], I can get the potential rule that in the situation without malicious users, the higher the threshold the method has the better values in “the number of victims” and “vehicle ID of first beneficiary”.

6.3.4 Situation with collusion team and a fake negative warning message

In situation with collusion team and a fake negative warning message, I also consider about the impact of malicious users, I set a situation that there is no initial warning message exist on the road but there is a team of malicious user in the traffic simulation. They will give a negative reflect as a fake messages to influent the warning broadcasted in the whole network. This set of test is aimed at testing of robustness of the Ad-hoc network system under the impact of the malicious users.

Number of road	3
Number of intersection	1
Number of normal vehicles	100
Number of crash generator	Depend on the verified method
Number of malicious users	Depend on the verified method
Situation in the simulation	Fake non-warning with real warning
Crash duration time	Whole simulation
Percentage of the verified warning	80%

Table 6.4 Test setting of simple map with scenario three

The variable in this set is the number of malicious users. And the result is got from the five random files.

The results are the average value of these five files. The red line is the result of the id number of first non-victim. The blue line is the number of the victims in crash. X-axis is the distance between the intersection and the crash position. Y-axis is the number of the victims or the id number of the first beneficiary.

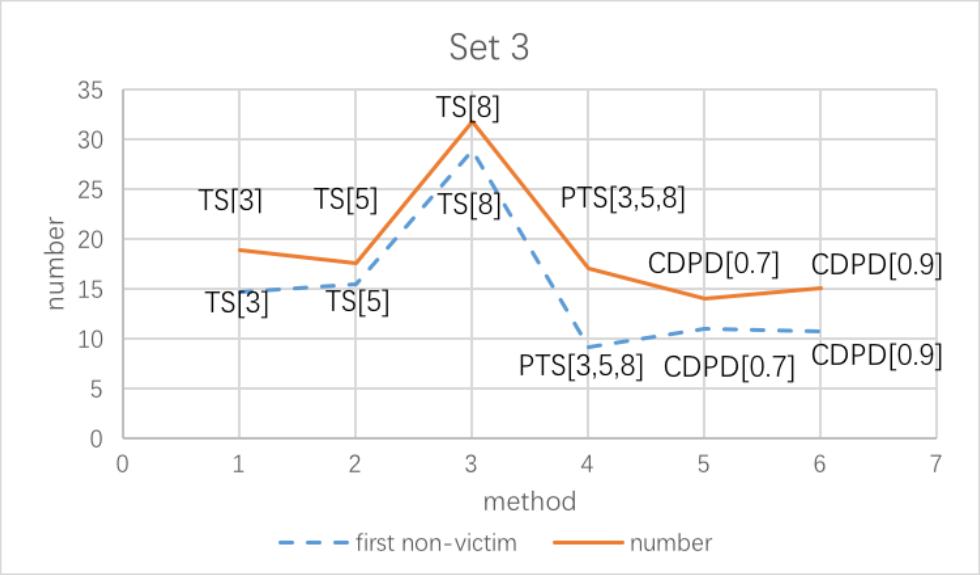


Figure 6.29 Result of simple map with scenario three (a)

The above figure shows the comparison of these six methods with a negative warning created by collusion teams. Even if in this comparison, threshold method has a small enough value which equal to three, CDPD still has a really obvious great result. And another rule same to previously simulation that the higher the threshold is the worse behavior it responds.

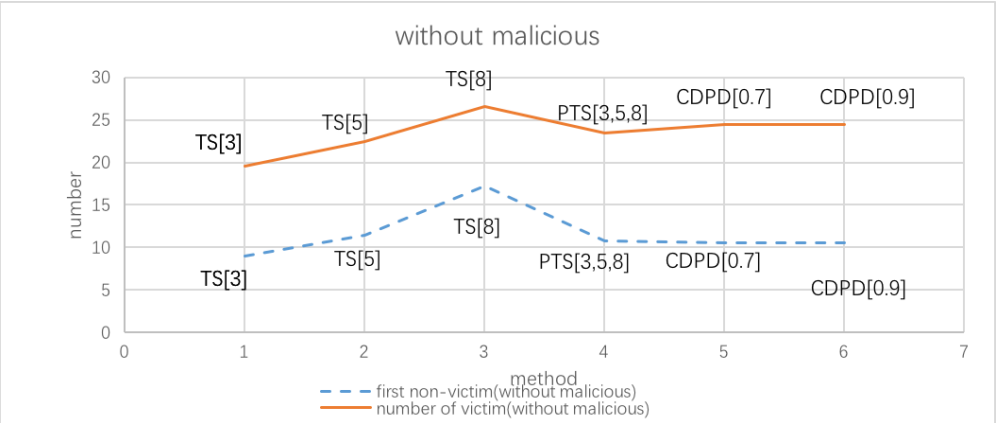


Figure 6.30 Result of simple map with scenario three (b)

And the figure 6.30 above is the result of the situation without malicious users. And in this assumption, this report gets the conclusion that in the situation without collusion teams, threshold method with a lower threshold has a better result and in the situation with malicious users CDPD method with a higher threshold has the better behavior.

6.4 Block map

In this block map section, I build a typical block map model from a real map in Excelsior, California, USA. Due to this block map is a really neat and typical block, so it can be seemed as both of real map in real life and a multiple set of simple map. And in the case of foreign road traffic, block map is the most frequent of a map. So after get the conclusion of the simple map, block map will a lifestyle application of the trust level warning system I supposed before and re-test the previous simulation results.

6.4.1 Situation without the collusion team of malicious users (80%)

Number of road	200
Number of intersection	100
Number of normal vehicles	200
Number of crash position	1/2/3/4/5
Number of crash generator	1/2/3/4/5
Number of malicious users	0
Situation in the simulation	No collusion team and fake warning
Crash duration time	100
Percentage of the verified warning	80%

Table 6.5 Test setting of block map with scenario one

In this section of trust level warning system, the distance between the intersection and crash position and the number of crash position is the two variables in this traffic simulation.

Compare with previously simulations, “the number of crash position” is the most significant parameter in this section. This variable can prove the investigation and resistance abilities of these three methods while there are more than one crash positions during the simulation time.

And for the no malicious user enactment, the level of practical efficiency is the only conclusion that can be obtained.

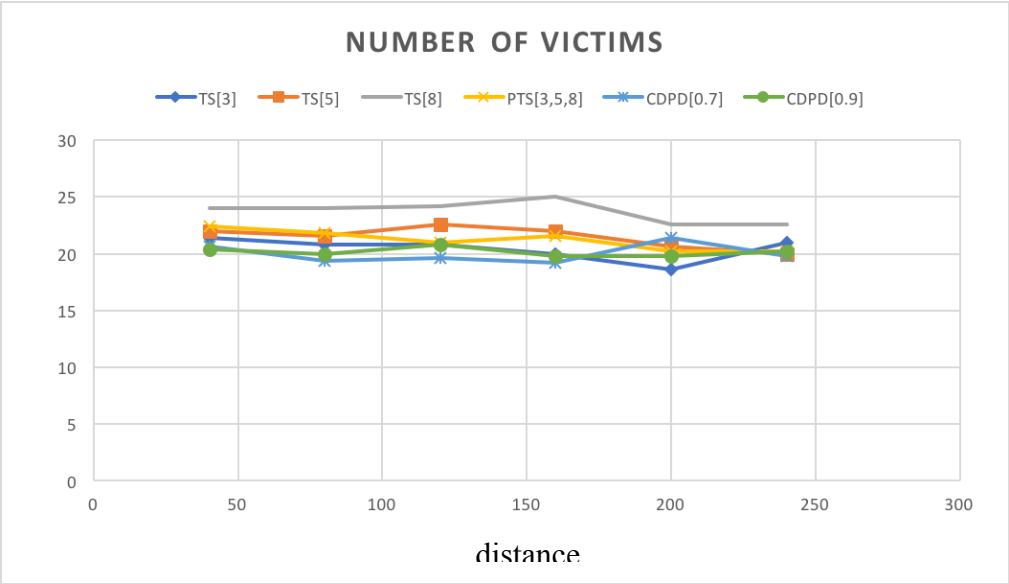


Figure 6.31 Result of block map with scenario one – one crash point

In the one crash position section, all the man-made settings are the same with simple map. But because of the setting of block map itself, the vehicle goes on the east - west direction must let the vehicles goes on north-south direction first and there is a stop sign on every junctions of the block map. Comparing with the figure 6.13 above and the result get from simple map, this report can get the difference after add the impacts of stop sign and road priority. It is obvious that, for the situation without malicious users and fake warnings threshold method still has the best behaviors of efficiency. More over the smaller the threshold value in threshold method, the better respond behavior it will give out.

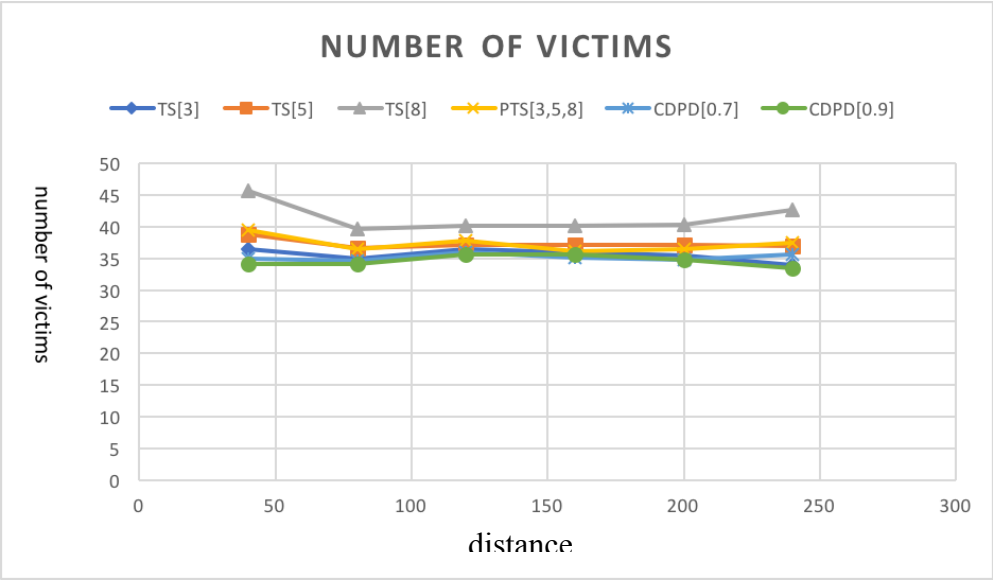


Figure 6.32 Result of block map with scenario one – two crash points

Above figure 6.32 shows if the crash position increases from one point to two points. The number of victims which fall into the crash after they received the verified warning message also with the increase in the relative.

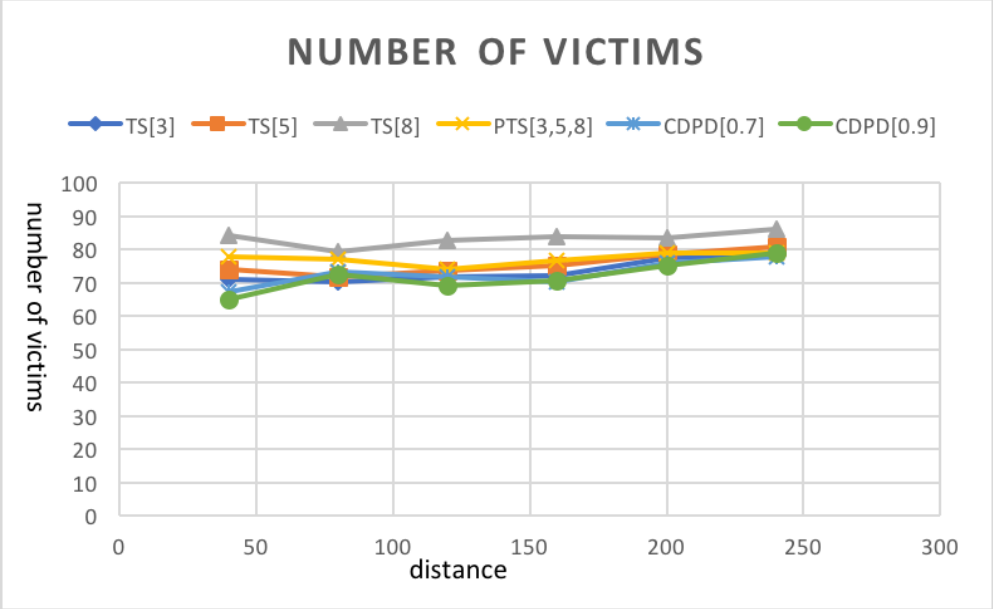


Figure 6.33 Result of block map with scenario one – three crash points

By the increase of crash position, the value of each growth becomes smaller and smaller which means that by the growing of the crash point, the advantage of threshold methods become less obvious. In conclusion, by helps of this finding, we can find a better method while we comparing the trustable and efficiency in the later situation.

6.4.2 Situation with collusion team and a fake positive warning message

Number of road	200
Number of intersection	100
Number of normal vehicles	200
Number of crash position	1/2/3/4/5
Number of crash generator	1/2/3/4/5
Number of malicious users	0
Situation in the simulation	Fake positive warning
Crash duration time	100
Percentage of the verified warning	80%

Table 6.6 Test setting of block map with scenario two

In this situation, this report pays more attention on aware and cancel an exit fake warning message inside the traffic simulation. And this situation supposes that there is no crash inside the simulation and a fake positive warning has been created by a collusion teams.

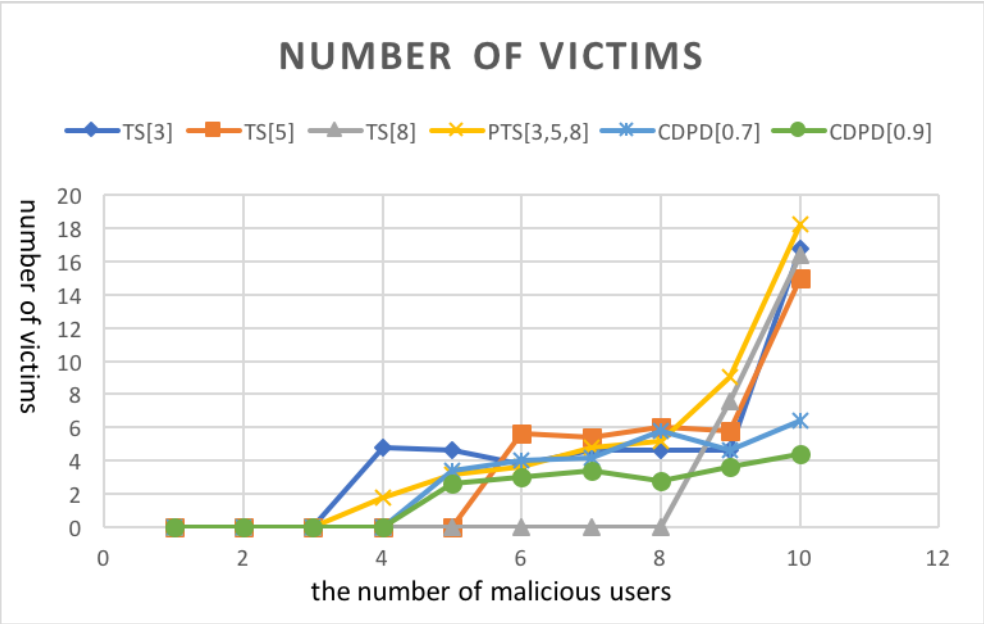


Figure 6.34 Result of block map with scenario two – one crash point

Above figure 6.34 shows the results of the simulation that crash position is equal to one. The impaction cause by malicious users will start after the number of malicious users bigger than the threshold in each method. Due to this reason, TS [8] is the last one being attacked. But while the number of malicious user is larger than its threshold in TS methods, the value of "the number of victims" grows up rapidly. So do PTS method. The opposite of TS method and PTS method, CDPD method shows a much more stable result, even if it being attacked earlier than TS [8], TS [5] and PTS [3,5,8].

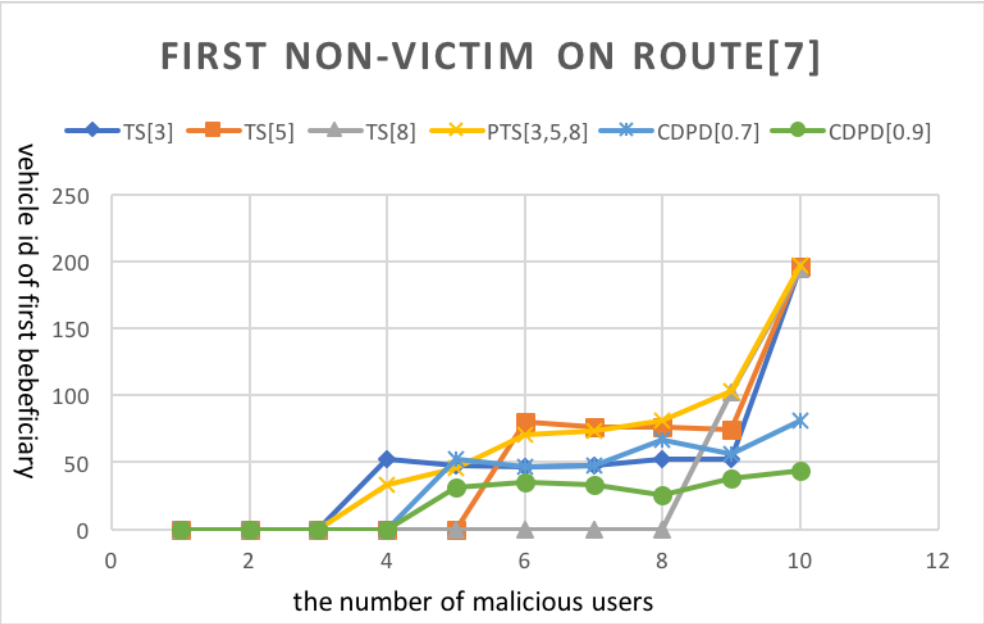


Figure 6.35 Result of block map with scenario two – two crash points

For the value of "vehicle id of first beneficiary", it is difficult to compare the measurement of result of each method in this section. And after the last finding that the trend of monotonically increasing the number of each line will not change. So this report tries to use the slope of each line to compare between the methods. The line drew with the smallest slope means this method has the better behavior in this situation of trust level warning system. By the help of figure 6.35 above, it is obvious that CDPD method with threshold 0.9 have the best efficiency and safety and the second is PTS [3,5,8]. In this situation TS [5] and CDPD method with threshold 0.7 both has a similar respond with the fake positive warning message.

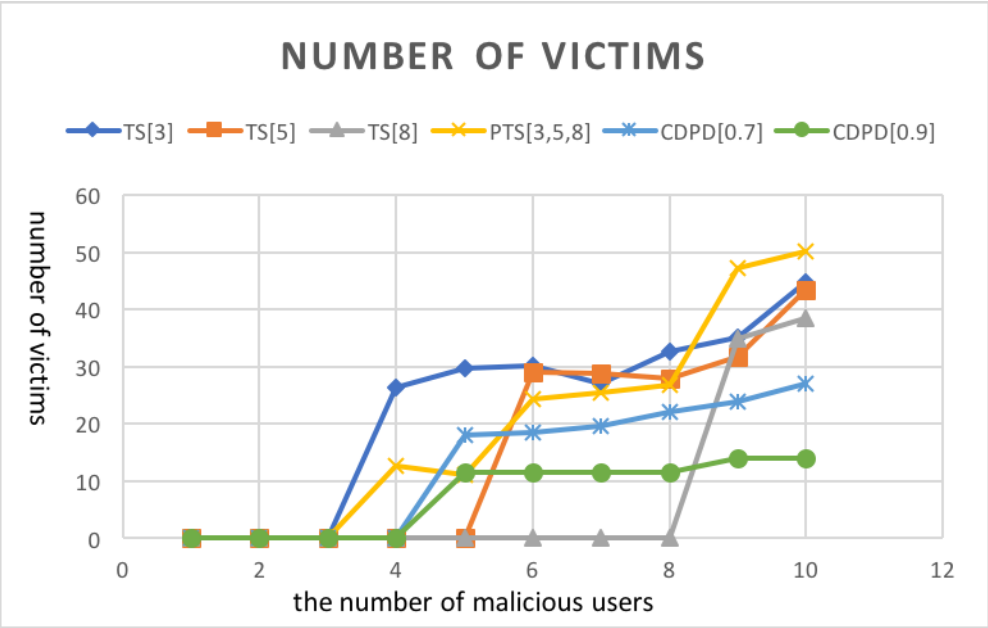


Figure 6.36 Result of block map with scenario two – three crash points

This situation is set to be the multiple form of the simple map but the result shows the result respond a different behavior. Compare with results I have got from one crash position in this situation, each method shows a huge of difference at very early times. For example, in the one crash position set, the variable of malicious users from equal to 4 to equal to 8, the result in this interval almost have the similar value. On the opposite of this situation while the number of crash point is equal to five, the values of different lines present a very distributed state after the number of malicious users is bigger than the threshold in each method during the simulation time. So in this set, it is easier to compare the six method to the previously settings.

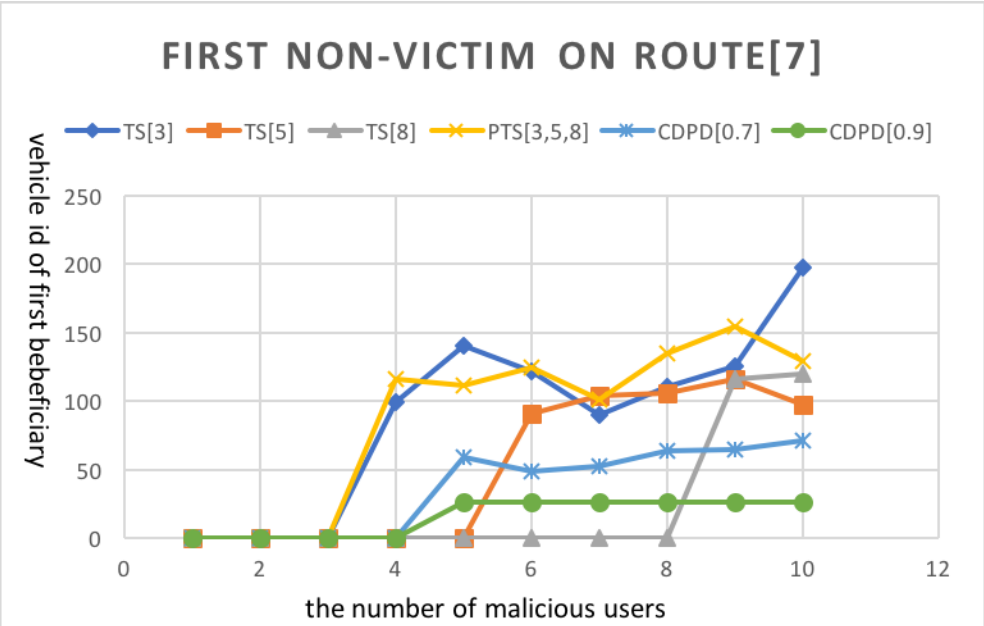


Figure 6.37 Result of block map with scenario two – Route [7]

6.4.3 Situation with collusion team and a fake negative warning message

Number of road	200
Number of intersection	100
Number of normal vehicles	200
Number of crash position	1/2/3/4/5
Number of crash generator	1/2/3/4/5
Number of malicious users	0
Situation in the simulation	Fake negative warning
Crash duration time	100
Percentage of the verified warning	80%

Table 6.7 Test setting of block map with scenario three

In this situation, this report pays more attention on aware and cancel an exit fake warning message inside the traffic simulation. And this situation supposes

that there is a real crash inside the simulation and a fake negative warning has been created by a collusion teams of malicious users.

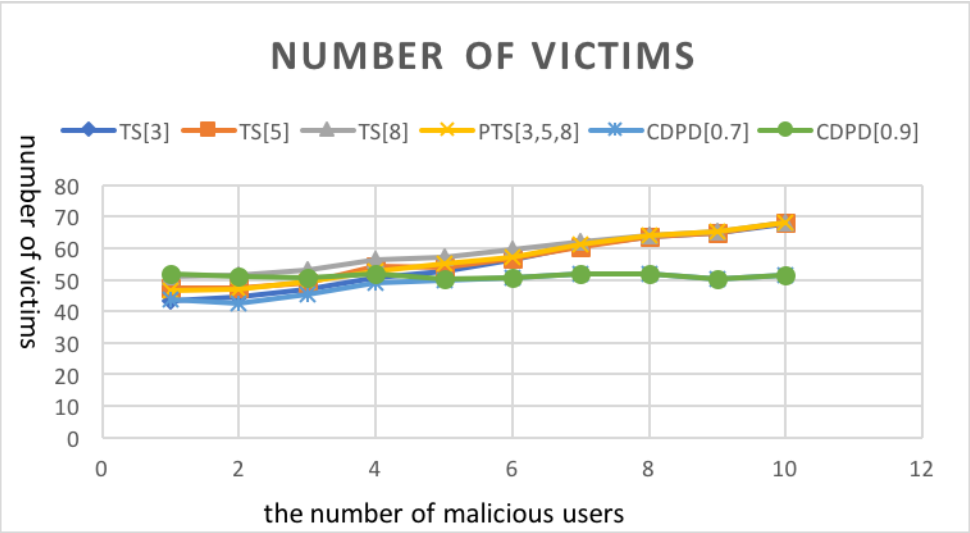


Figure 6.38 Result of block map with scenario three – one crash point

The assumption of figure 6.38 on above is fake negative warning with 100 test vehicles. Both of CDPD [5, 0.7] and CDPD [5, 0.9] show a good result by the increasing of the number of malicious users. Even if the number of malicious users become larger, the result of CDPD method does not add much.

To consider about the real life more, this report also tests a setting of 200 vehicles and the following figure 6.39 is the result of this setting. But in this simulation, CDPD method does not responds the best behavior than the previously one. But as same as the result got in 100 vehicles setting, the result of CDPD method has a really stable result. The simulation this report has done only has an interval from 1 malicious user to 10 malicious users. All the other methods except CDPD method, shows a monotonically increasing feature and there are more than ten malicious inside collusion team in real life. So by the higher malicious user it will have in the real life, the advantages of CDPD method might

come out later. By the limitation of the three map, the maximum of the number of malicious user is no more than ten. I will do the further test in the future task in order to prove the advantages of this method in a long-term.

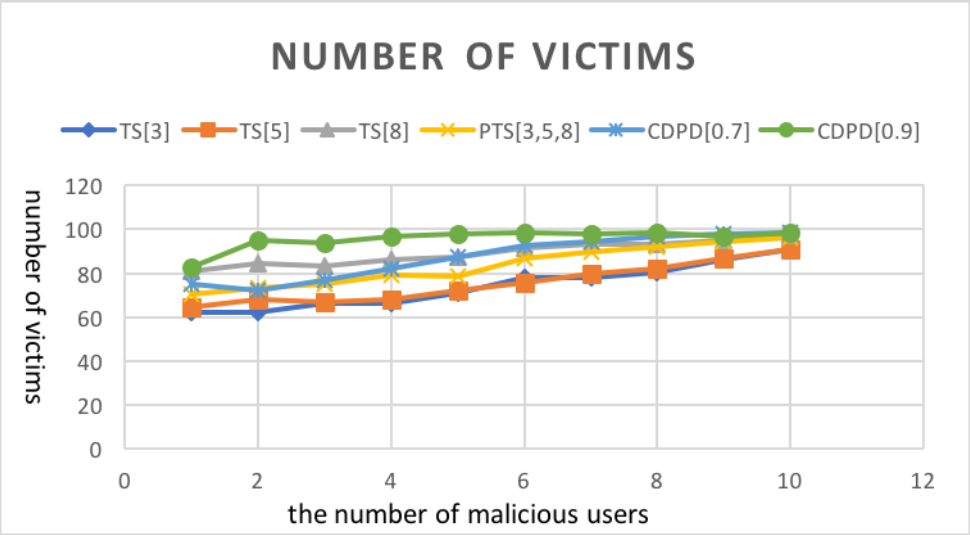


Figure 6.39 Result of block map with scenario three – two crash points

As I describe in the front chapter, this situation is about a collusion of malicious users which run on the pass-way and there is a real crash exited on the main road. Malicious users try to create a fake negation warning to let more test vehicles fall in to the crash on the main road. So in contrast to the reality, the probability of this situation is very small. As far as possible to include all the actual situation may occur in the real life, this report also does the simulation of this situation. So even if CDPD method does not show an excellent performance like the previous scenario by the increasing of the number of crash point. It can also be proved that in combination, CDPD method is the best method during the whole simulation. And CDPD [5, 0.9] has a better behavior than CDPD [5, 0.7], even if the result is not that good, but it is a really stable result than any others.

6.5 Real map

In this block map section, I build a typical block map model from a real map in New York City, New York, USA. The simulation has done by three different situations this report has mention in the previously chapters.

And this real map section is an examination of the conclusion gotten in the previously simple map and block map. Moreover, compare the difference between the result of real map and simple map, this report can get the impaction of the traffic light with the trust level warning system. But by the data analysis of this section, it has a similar result with block map. So this report would not illustrate the result of real map again. All the examination of the real map can be find in the appendix of this report.

7 Performance Evaluation

7.1 Simple map:

7.1.1 Situation without the collusion team of malicious users

Due to the reason that both of number of victims" and "first non-victim" are important parameters, the final method I figure out must let both of these two parameter get the best result (the smaller the better).

In the situation without malicious users, CDPD (CDPD [0.7] is equal to CDPD [0.9]) shows the best efficient and correct rate.

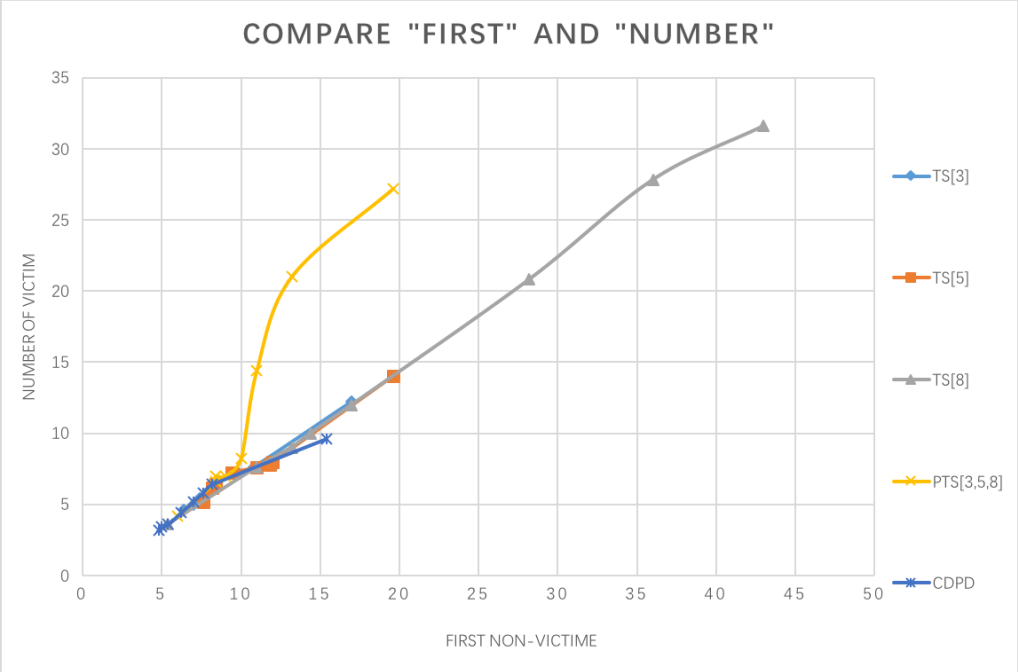


Figure 7.1 Analysis of simple map with scenario one

To explain the above graph. I combine the table "distance-number of victims" and "distance-first of non-victim" by the same coordinate "distance". In this assumption, I set the parameter "the ID of first non-victim" be the x-axis and the

parameter “the number of victims” be the y-axis and combine the two points into one point by omission the value of “distance”.

So for the graph I design, it has obvious advantage feature that making the comparison between these 6 methods much more clearly. The efficient method I’m doing research to find must has a smaller value of “distance-number of victims” and “distance-first of non-victim”, this feature will show in the graph as the point has both a smaller x-axis value and a small y-axis value. A great or a tiny slop will not be considered as a good enough method of the set1 Trust Level System.

7.1.2 Situation without the collusion team of malicious users (80%)

Modified by the previous set, I set only 80% of the users will trust and follow the guidance from Ad-hoc system. And I got the following results.

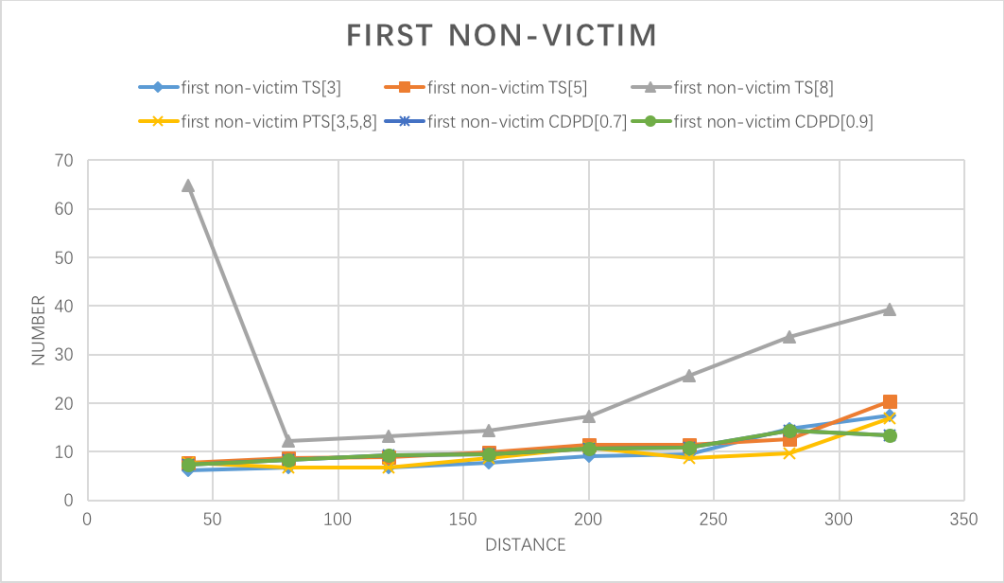


Figure 7.2 Analysis of simple map with scenario one in first beneficiary (80%)

The great X value of the beginning of TS [8] is due to the parameter of distance.

A short of “distance” like 40 units, will cause all the vehicle miss the real warning message before they pass the intersection. So all the vehicles will fall into the traffic crash. Compare with other methods, TS [8] needs more time to verified the positive warning message.

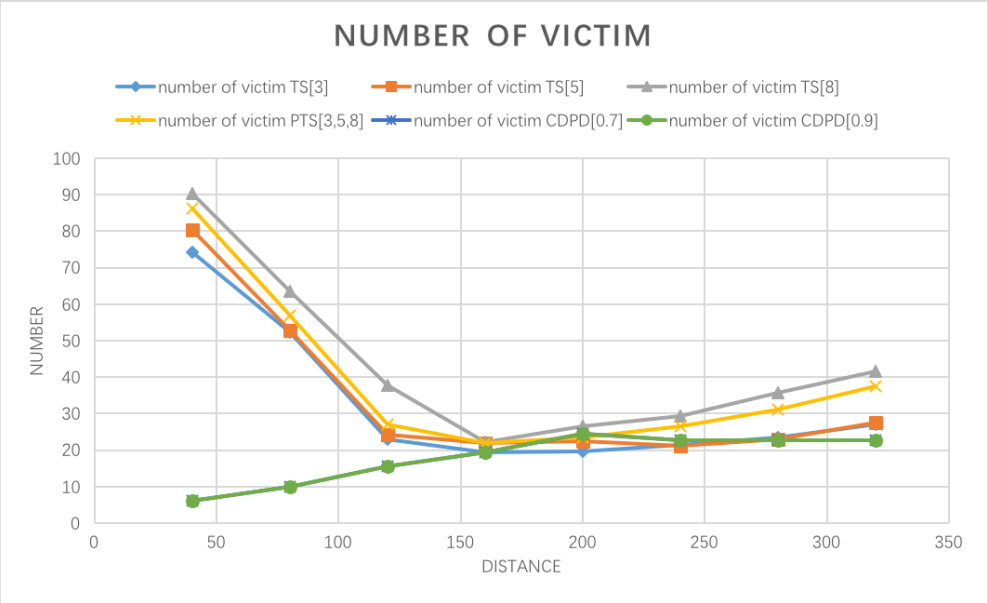


Figure 7.3 Analysis of simple map with scenario one in number of victim (80%)

The image above is the value “number of victim” of set 1.5. The result shows the figure of Set 1.5 is obvious that all of TS [3], TS [5], TS [8] and PTS [3,5,8] all has a special beginning in this set of test. And for the CDPD method, in set1’s assumption, without malicious users, it works like Threshold method with threshold equal to 1. Due to this reason, it shows a more efficiency than other methods.

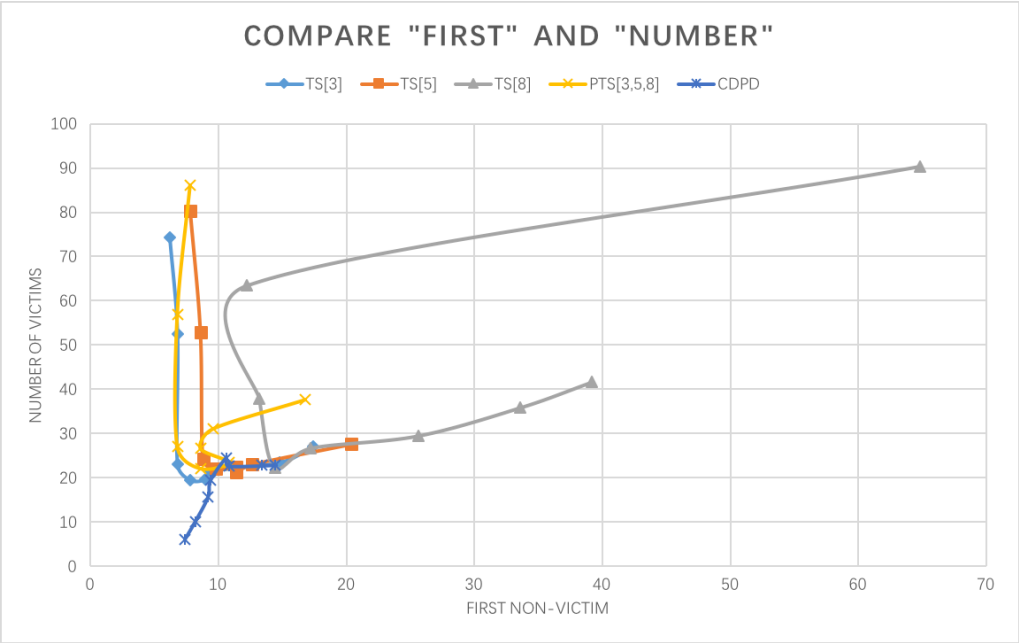


Figure 7.4 Analysis of simple map with scenario one (80%)

By the previous conclusion, I won't consider the efficiency of CDPD method. Compare the other methods, Both of TS [3] and PTS [3,5,8] has a better result than other methods. TS [3] and PTS [3,5,8] show their advantages differently in "Vehicle ID of non-victim" and "number of victims" separately.

7.1.3 Situation with collusion team and a fake positive warning message

In this set, the advantage of CDPD method shows out obviously. Because in this set, the fake warning message can be cancelled only by the 20% of the normal users does not trust the guidance from the Ad-hoc warning system. In this way, the less present of trust the less time it needs to cancel the fake warning message. If the 20% of the normal user does not trust the guidance from the Ad-hoc warning system and keep go on their route on the main road, after they pass the

crash position and find there is no crash at that position broadcasted by the Ad-hoc warning system, they will send a negative message to cancel the warning message as soon as possible.

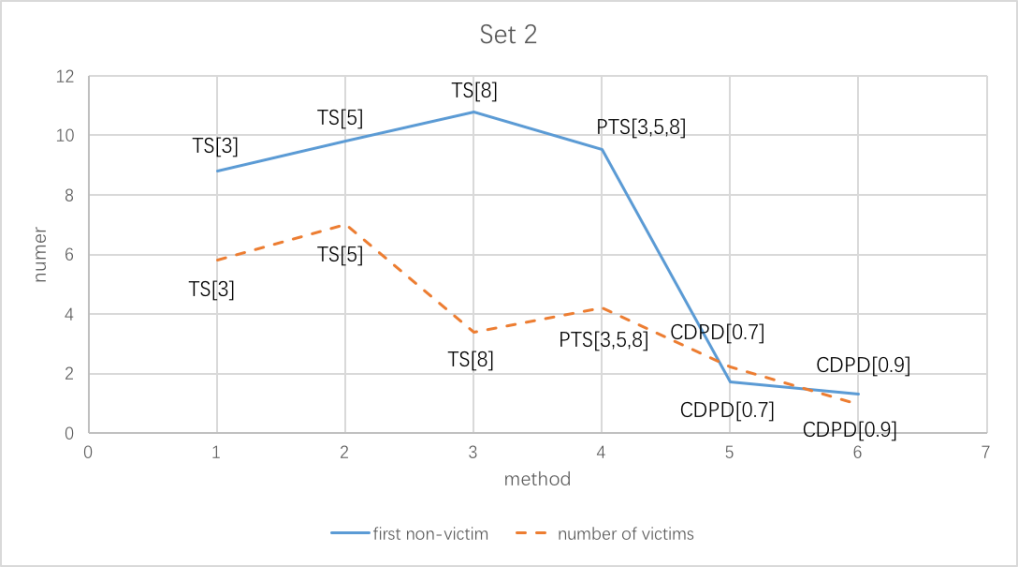


Figure 7.5 Analysis of simple map with scenario two

Furthermore, I compare CDPD [5, 0.7] and CDPD [5, 0.9], in order to find the impact of the parameter. As shown in the following figure, CDPD [5,0.9] performs a better result which both have a less value of “vehicle ID of first beneficiary” and value of “number of victims”. This efficiency is due to the higher the threshold is the less credibility it has, compare with CDPD [5,0.7], a warning message will cost long time to be verified for CDPD [5,0.9].

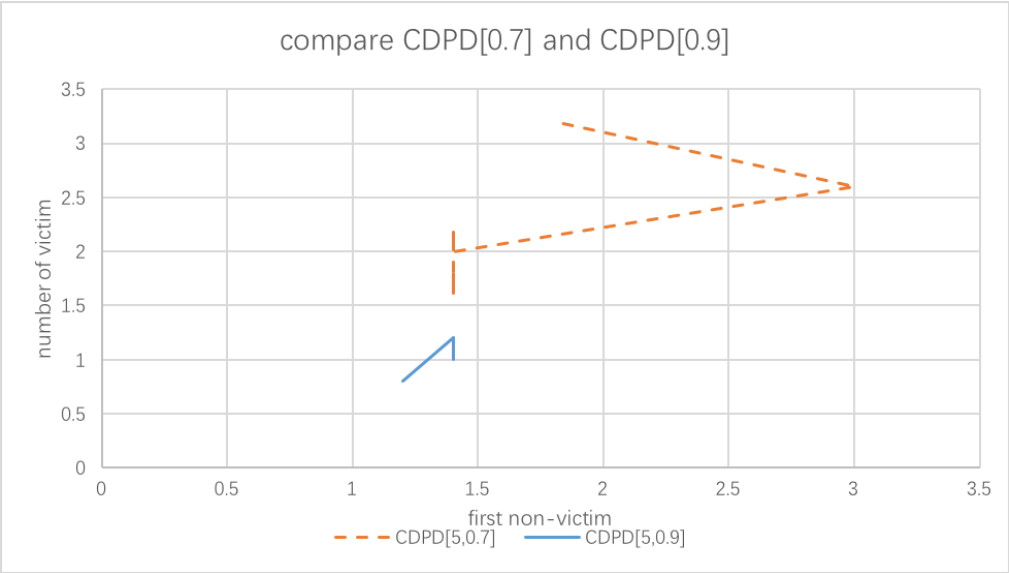


Figure 7.6 Analysis of simple map with scenario two (CDPD method)

7.1.4 Situation with collusion team and a fake negative warning message

In this set of test. These six method will be tested to verify a real warning under the impaction from collusion teams.

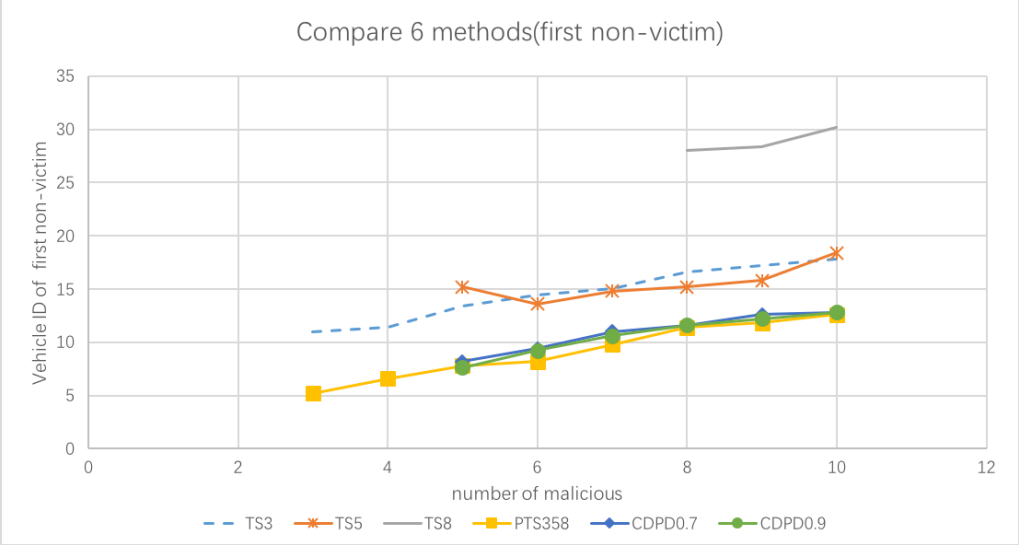


Figure 7.7 Analysis of simple map with scenario three in first beneficiary

The above figure shows that the parallel threshold method has shorter time to verify a real crash under the impaction of collusion teams. And by the grow up of malicious users, CDPD method also show a better result.

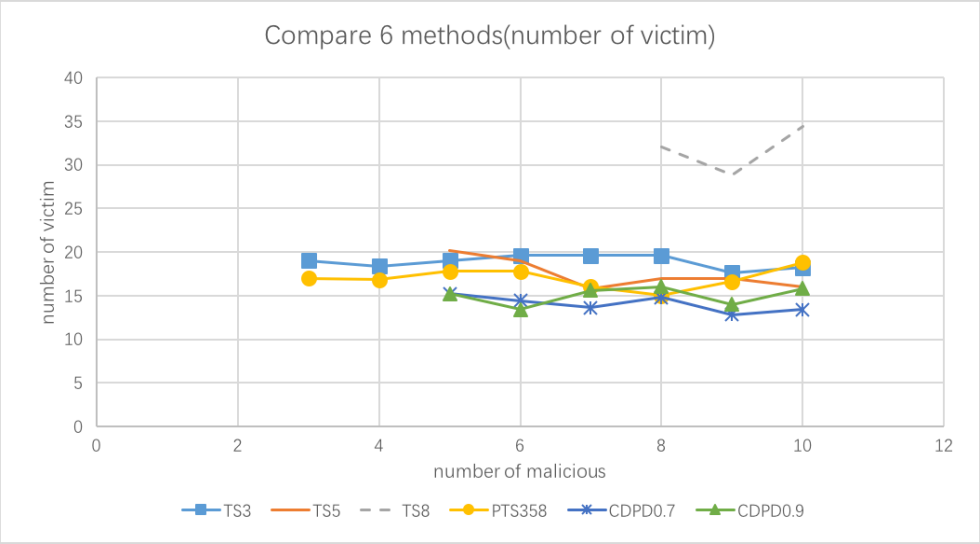


Figure 7.8 Analysis of simple map with scenario three in the number of victim

For “number of victims”, CDPD method performs a great result than any other method. Due to the dynamically authentication mechanism, it can detect malicious users and give a positive feedback efficiently.

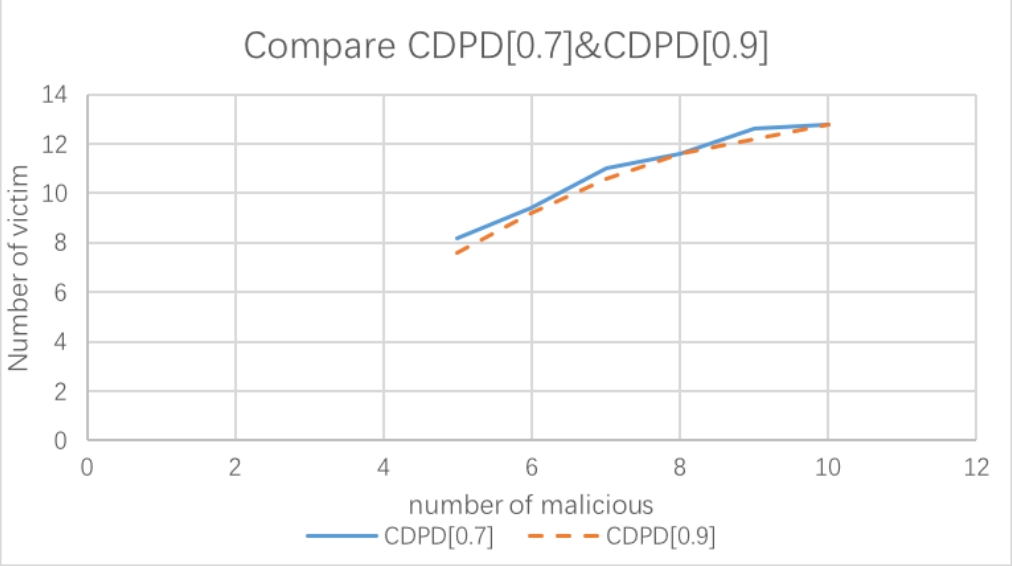


Figure 7.9 Analysis of simple map with scenario three (CDPD method)

The parameter of CDPD method does not impact the test result obviously, but compare with another method. CDPD [5, 0.9] give a better result.

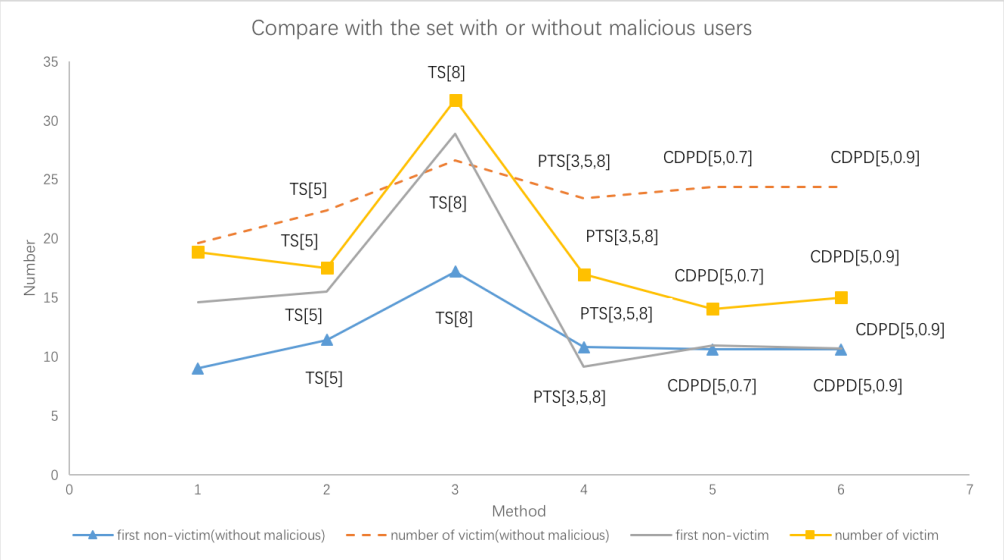


Figure 7.10 Analysis of simple map with scenario three

In above figure, the six points in the same line means six different methods. Compare with the result with and without collusion team, CDPD method performs a better result to resist the impaction from malicious users.

7.2 Block map:

7.2.1 Situation without the collusion team of malicious users (80%)

According to previous research results, the system with a higher safety level will have a lower efficiency in relatively. And in this situation, due to the scenario that there is no malicious user, this result can only be considered as an examination of efficiency.

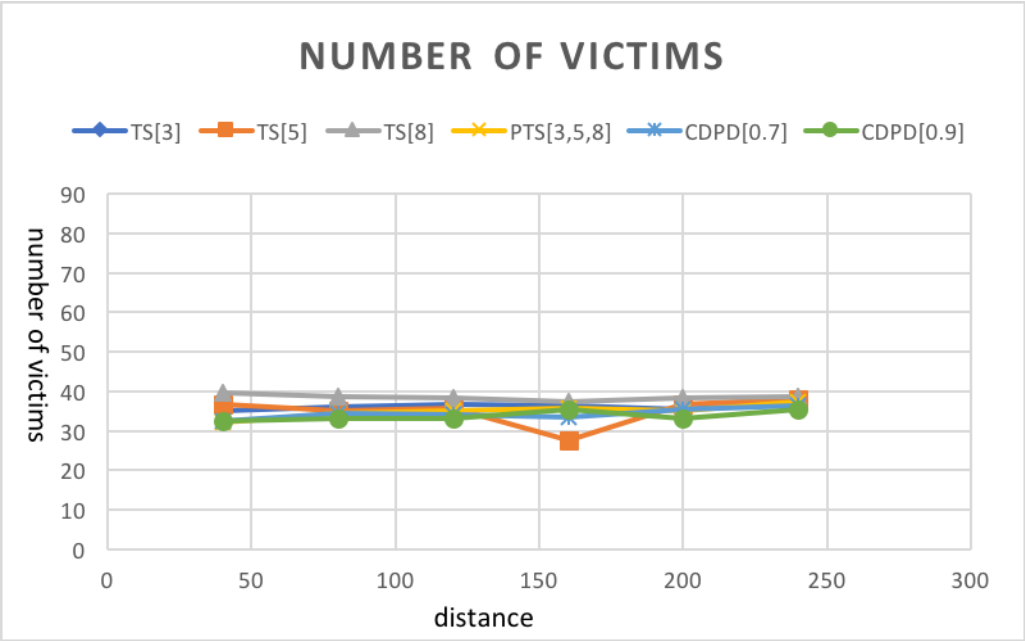


Figure 7.11 Analysis of block map with scenario one (100 test vehicles)

In the setting of 100 test vehicles, the difference between each method is not obvious. In the above figure 7.11, it is obvious that the percentage of victims relative to the total number of test vehicles for all the trust verify method is under 40%.

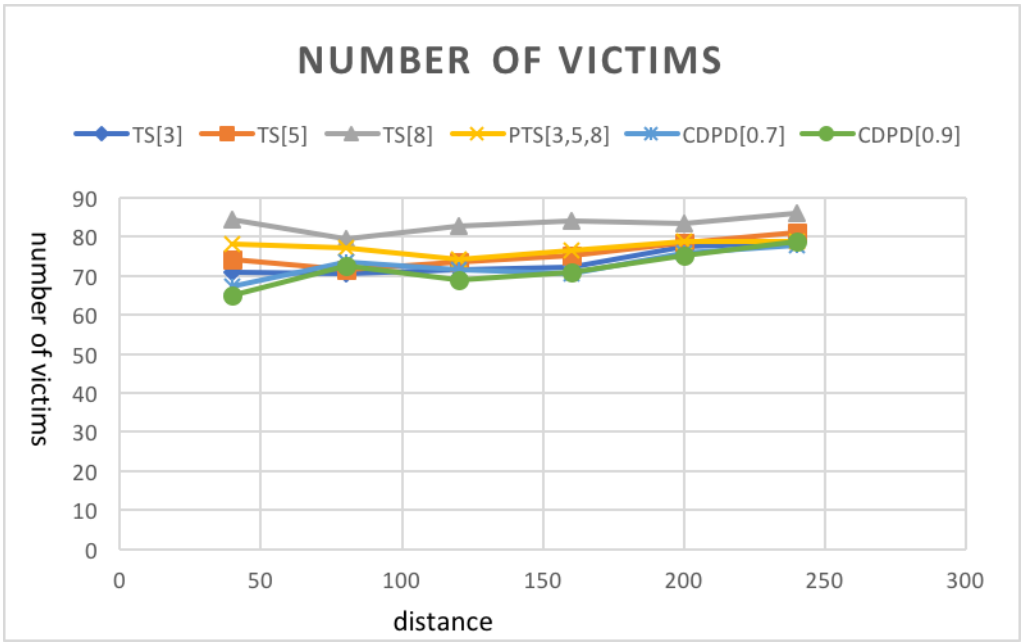


Figure 7.12 Analysis of block map with scenario one (200 test vehicles)

Comparison of the above two figures, by the increasing of the number of test vehicles, the CDPD method shows the advantage of the efficiency much better. And consider the situation in the real life traffic, the traffic flow is certainly huge enough to reflect the superiority of CDPD method.

In conclusion, if only consider the efficiency of each method regardless of their safety, CDPD method is the best method and the higher the threshold in this method. And the TS method with a threshold less than 5 also shows a good result. Moreover, to compare the safety of each method in this system, I will do the analysis later.

7.2.2 Situation with collusion team and a fake positive warning message

For the setting of each parameter in this experiment, this report aim to find the better method which has a both high efficiency and safety (even if this method may not have the highest efficiency or the highest of safety). So for this situation, the most comprehensive approach is the best.

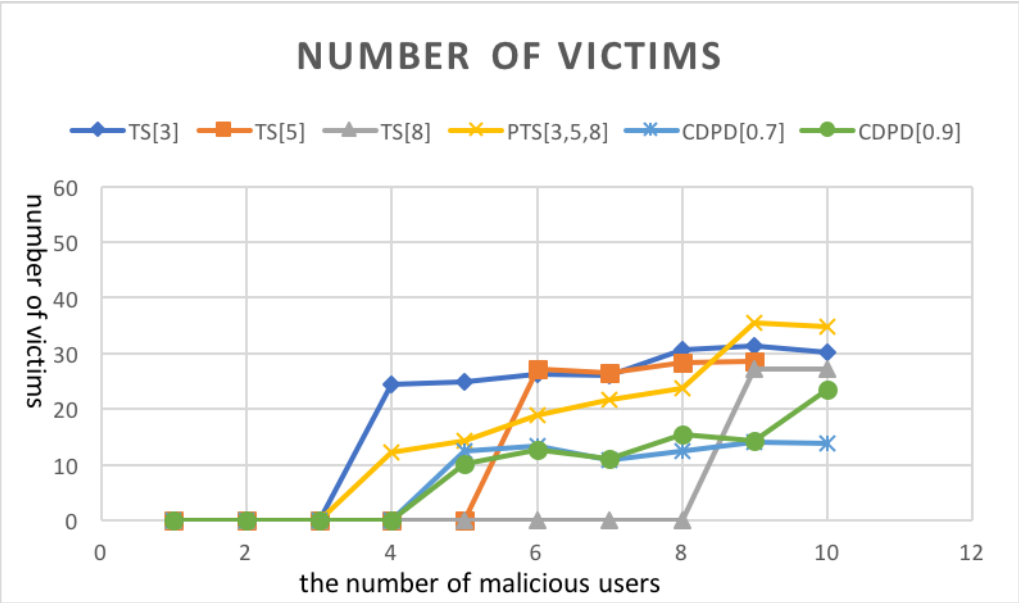


Figure 7.13 Analysis of block map with scenario two (100 test vehicles)

Compare the two settings 100 test vehicles and 200 test vehicles, by the increasing of the number of test vehicles, CDPD [0.9] shows a better reflect of efficiency like the previously conclusion. TS [8] method ranks the third place in terms of computational efficiency. And PTS [3,5,8] is ranked as the forth place.

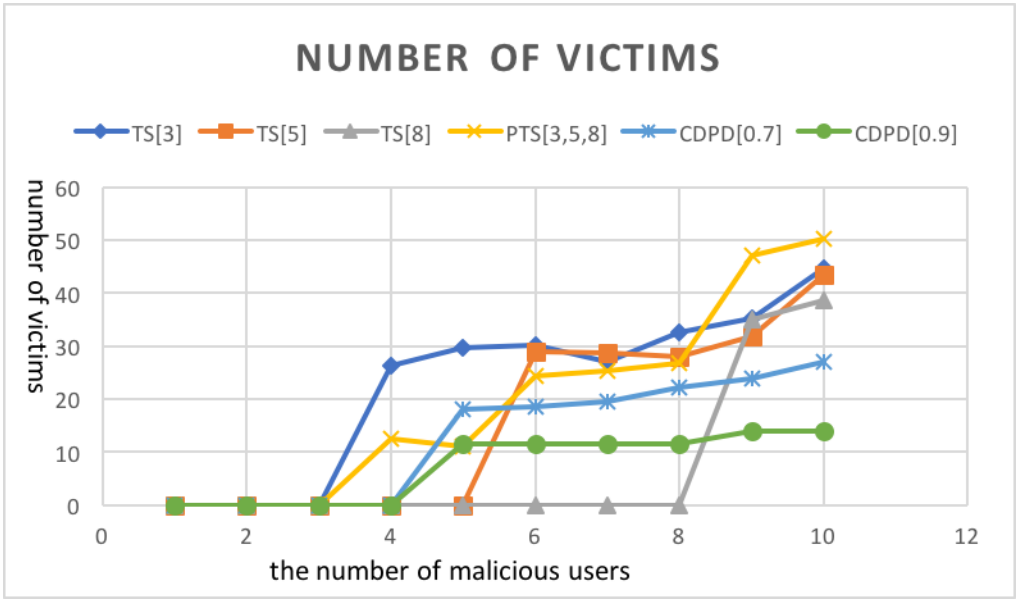


Figure 7.14 Analysis of block map with scenario two – (200 test vehicles)

Moreover, this report also does the comparison of value of "the vehicle id of first beneficiary". The results are on the following figures.

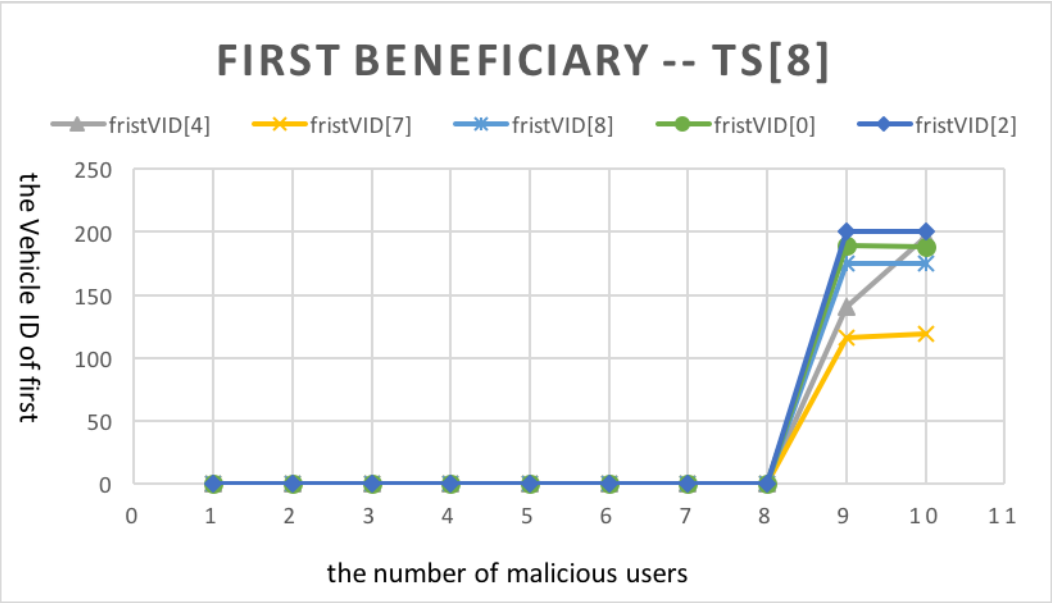


Figure 7.15 Analysis of block map with scenario two – TS [8]

By analyzing he figure 7.15 above, TS [8] shows 0 victim before the malicious user is less than the threshold in TS method. But after the number of malicious users is bigger than 8 (8 is the threshold in TS [8] method), the number of the victims rapidly growing to 200 which is the number of all tested users. This means after the number of malicious users is bigger than 8, there is no beneficiary in this traffic simulation.

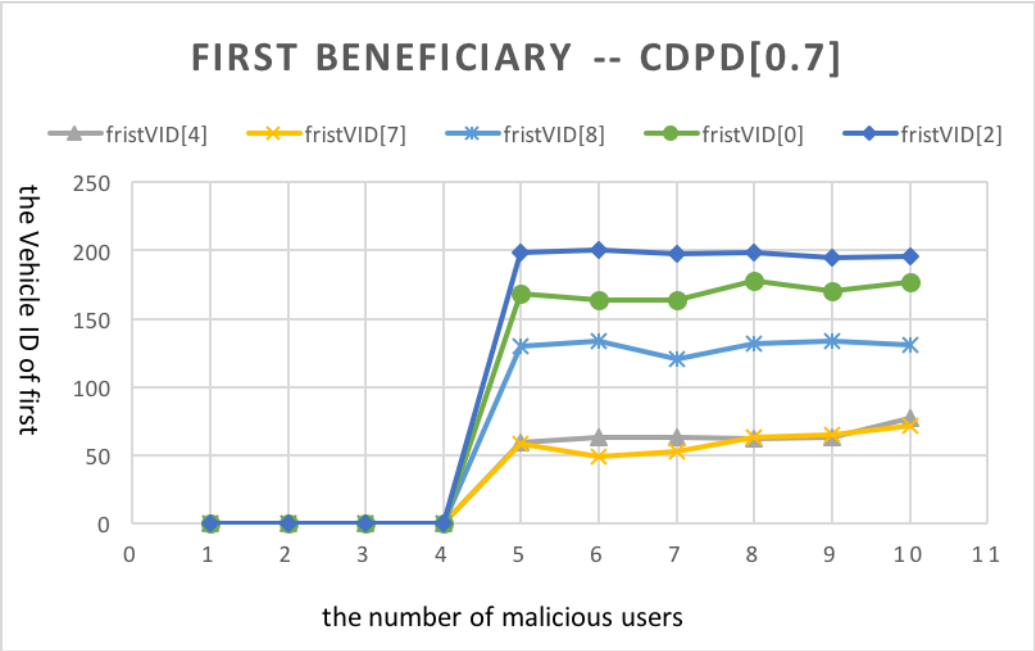


Figure 7.16 Analysis of block map with scenario two – CDPD [0.7]

Compare the figure 7.17 and figure 7.16, it is evident that the first beneficiary of CDPD [0.9] appears earlier than CDPD [0.7]. And plus the previous conclusion, these two advantage can best prove that CDPD [0.9] is better than CDPD [0.7] in this situation with collusion teams.

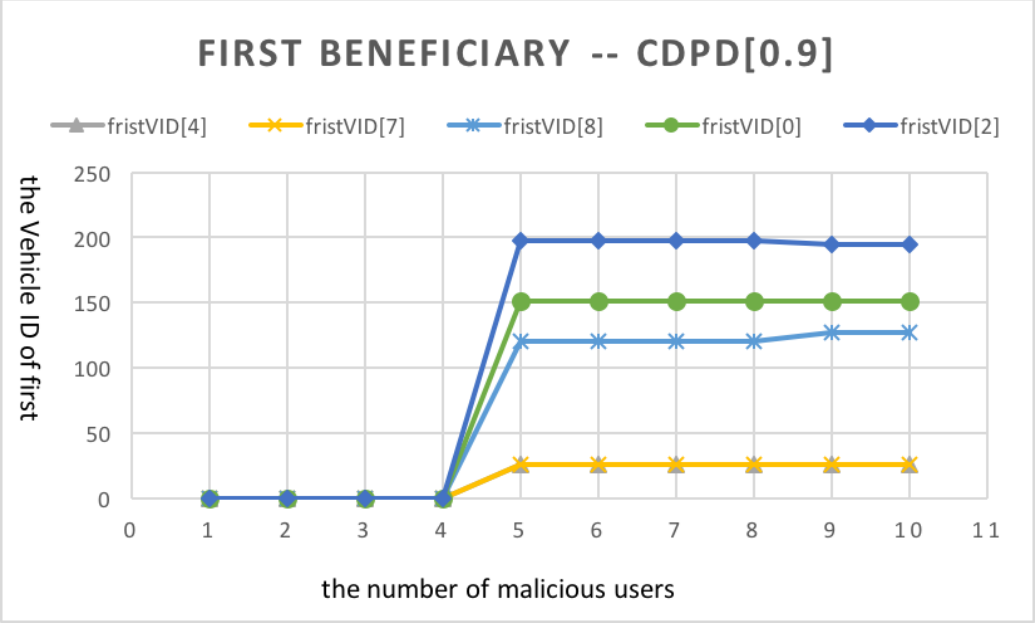


Figure 7.17 Analysis of block map with scenario two – CDPD [0.9]

For PTS method which is a special edition of TS method, it has a better and more comprehensive behavior. But contrast with CDPD method which is a high dynamic method, PTS method is not good enough to be implemented in the trust level warning system.

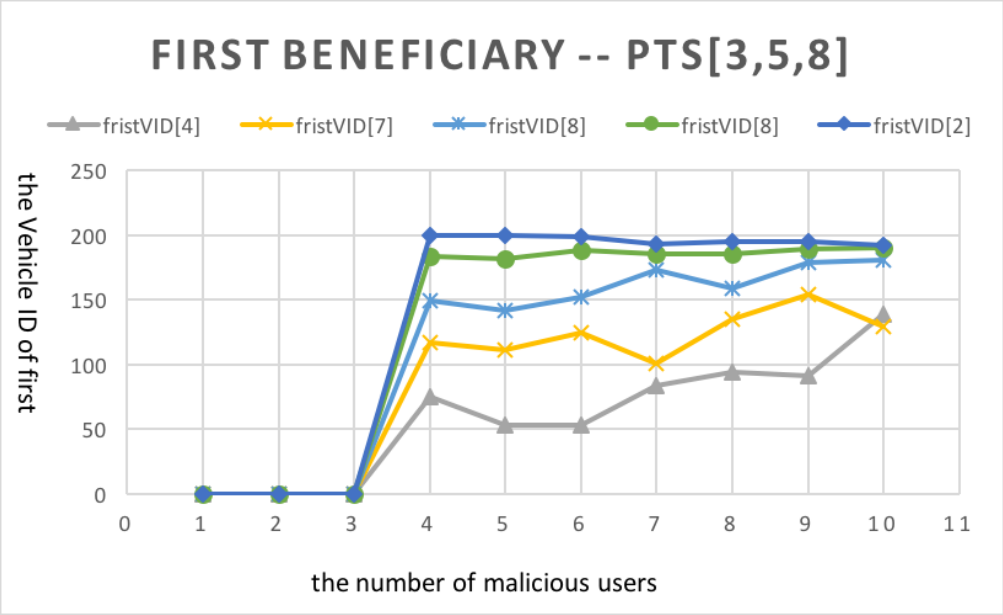


Figure 7.18 Analysis of block map with scenario two – PTS [3,5,8]

7.2.3 Situation with collusion team and a fake negative warning message

For the unique character of the settings in this situation (malicious users detect the exit crash first and after they turn their way into pass-way they leave a fake negative warning message on the main-road instead), all the value of "the number of victims" shows a strong regularity of grows up multiply.

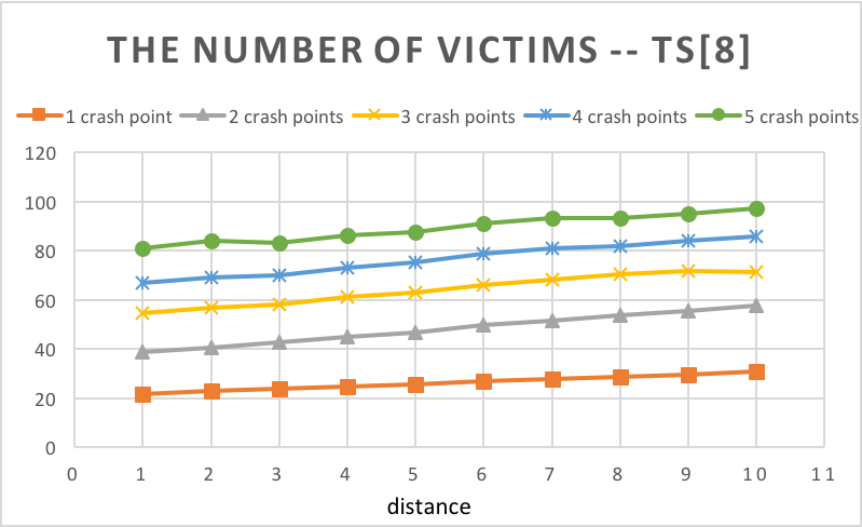


Figure 7.19 Analysis of block map with scenario three – TS [8]

For TS [8] method, the slope of each line is the same. And the difference of each line is depends on the number of the crash points.

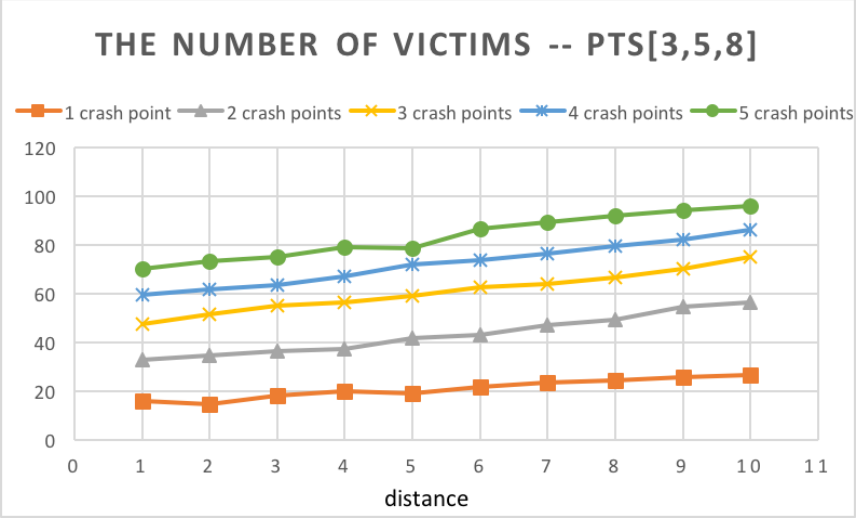


Figure 7.20 Analysis of block map with scenario three – PTS [3,5,8]

Also PTS [3,5,8] has the same features of the value of "the number of victims".

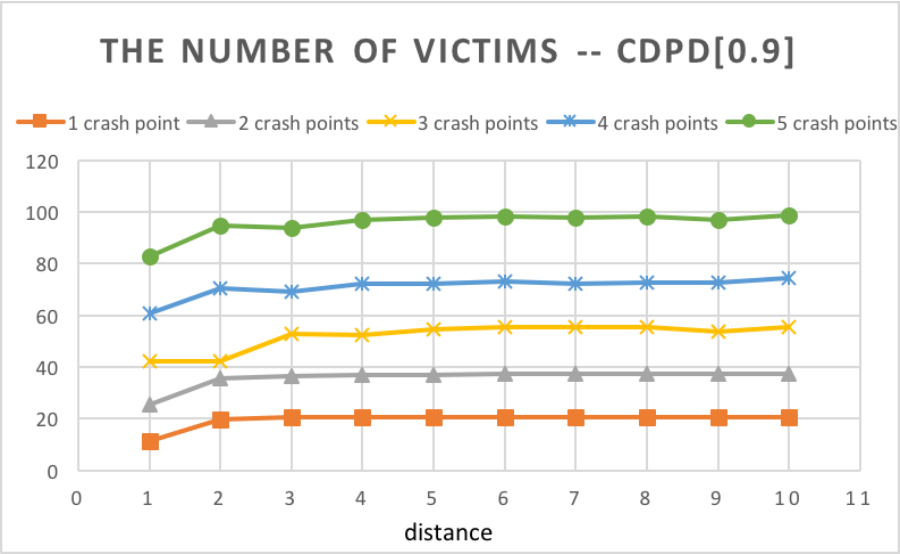


Figure 7.21 Analysis of block map with scenario three – CDPD [0.9]

The above figures 7.21 is the values of methods CDPD [0.9], the regular increasing mean the grow up of the number of crash points in this situation makes the almost no difference in the trust level warning system simulations. By the characteristic of the result of this situation, I would not do the further analyze. And all the conclusion of this block map will be given by the previously two situations.

7.3 Real map:

In this report, the real map part is just an examination section to test the conclusion got in the previous simple map and block map. As expected, the result is also as same as my assumption. So in this report, the result of real map section would be analyzed again in this chapter and the figure of all the result can be find in appendix.

8 Conclusion

In this project, I proposed an edition version of trust level warning system and combine the trust level warning system with Ad-hoc Network. To complete the traffic simulation of trust level warning system, I set up a traffic simulation project on Simulation of Urban Mobility.

To make the whole experiment more complete and comprehensive. I set up three representative scenarios which may happens in real world frequently. One is the situation without malicious user, one is a fake positive warning created by collusion team of malicious users, other is a fake negative warning created by collusion team of malicious users. Furthermore, to test the three typical scenarios, I also build three different type of map, such as simple map, block map and real map. So There are nine combinations of the above three scenarios and three maps.

For the characteristics of the three scenarios, the situation without malicious show the efficiency most obviously. And to detect the security of this method, I consider the second more. And in the third scenario, I test the conclusion of the preciously two situations.

Moreover, due to the simple structure of the simple, all the behavior of safety and efficiency can be shown very obvious. And block map is both a special type of real map and a multiple type of simple map but some traffic rules in the real traffic situation. So by the help of block map, I got some knowledge of the impaction of the road priority, stop sign and traffic light. And using the real map, I do the final test of all the conclusion gotten from the preciously two maps.

And finally I get the conclusion that (a) without collusion team of malicious users, the less certificates the system needs the quicker the respond will be received. (b) fake warning in an unobstructed road, rapid response mechanism causes the lack of alertness for normal vehicles. (c) real warning and collusion team of malicious users which gives wrong road situation message, the number of collusion teams induce the congregate delay of responds. (d) the generate rule in this system, excessive trust both improving the efficiency of reflection and decreasing the credibility of the system. Summed up the whole project, conditionality distinguishable pseudo identities approach with a higher threshold gives the best behavior in the trust level warning system both for the method's efficiency and safety.

9 Future Work

During the traffic simulation of real map part, I find another interesting aspect. While one test vehicle is waiting for the green light at a junction (I assume that every normal vehicle should stop and wait while the traffic light become red and yellow), there is another action, turn their way to another route, they can do instead. And by the function of Simulation of Urban Mobility, the information of later five junctions the vehicle will meet can be get from the given SUMO port. The information of the later five junctions will be shown as [(traffic light ID, traffic light index, distance between this position to the traffic light, the state of the traffic light), ...].

By the researches above, I think I can do some further work about the prediction of vehicle behavior at traffic light.

10 Reference

[1] Chim, T. W., et al. "Secure, privacy-preserving, distributed motor vehicle event data recorder." Connected Vehicles and Expo (ICCVE), 2013 International Conference on. IEEE, 2013.

[2] Sumra, Irshad Ahmed, et al. "Trust and trusted computing in VANET." Computer Science Journal 1.1 (2011)。

[3] Tsutsumi, Shigeyoshi, Takahiro Wada, and Shun'ichi Doi. "A methodology to increase driver trust in rear-obstacle warning systems with imperfect sensing results—Proposal for a warning system using sensor reliability information." IATSS research 35.2 (2012): 71-78.

[4] Yeung, C. Y., et al. "Distributing blackbox data to multiple vehicles in a secure and privacy-preserving manner." Connected Vehicles and Expo (ICCVE), 2014 International Conference on. IEEE, 2014.

[5] Chim, Tat Wing, et al. "SPECS: Secure and privacy enhancing communications schemes for VANETs." Ad Hoc Networks 9.2 (2011): 189-203.

[6] Chim, Tat Wing, et al. "Security and privacy issues for inter-vehicle communications in VANETs." Sensor, Mesh and Ad Hoc Communications and Networks Workshops, 2009. SECON Workshops' 09. 6th Annual IEEE Communications Society Conference on. IEEE, 2009.

[7] Raj, Chitraxi, et al. "Simulation of VANET using ns-3 and SUMO." International Journal 4.4 (2014).

[8] Neale, Vicki L., et al. "Investigation of driver-infrastructure and driver-vehicle interfaces for an intersection violation warning system." Journal of Intelligent Transportation Systems 11.3 (2007): 133-142.

[9] Simulation of Urban Mobility http://sumo.dlr.de/wiki/TraCI

[10] Python https://www.python.org/

[11] Traffic collision https://en.wikipedia.org/wiki/Traffic_collision

[12] Autonomous car https://en.wikipedia.org/wiki/Autonomous_car

[13] Python http://www.runoob.com/python/python-tutorial.html

[14] Wu, Qiong, et al. "Early car collision prediction in VANET." Connected Vehicles and Expo (ICCVE), 2015 International Conference on. IEEE, 2015.

[15] Jaap, Sven, Marc Bechler, and Lars Wolf. "Evaluation of routing protocols for vehicular ad hoc networks in typical road traffic scenarios." Proc of the 11th EUNICE Open European Summer School on Networked Applications (2005): 584-602.

[16] Cheuk Yu Yeung “Anonymous Counting Problem in Trust Level Warning System for VANET” submitted for publication.

[17] Zhang, Chenxi, et al. "An efficient identity-based batch verification scheme for vehicular sensor networks." INFOCOM 2008. The 27th Conference on Computer Communications. IEEE. IEEE, 2008.

[18] Zhang, Chenxi, et al. "RAISE: An efficient RSU-aided message authentication scheme in vehicular communication networks." Communications, 2008. ICC'08. IEEE International Conference on. IEEE, 2008.

Appendix A:

Source code:

```
    if(step >= 50):
        for i in range(100):
            if(str(i) in NewVeh):
                speed[i] = traci.vehicle.getSpeed(str(i))
                position.insert(i,traci.vehicle.getPosition(str(i)))
                edge[i] = traci.vehicle.getRoadID(str(i))
            else:
                speed[i] = -1
                position.insert(i,[-1,-1])
                edge[i] = "-1"
            TmpPos = position[i]
```

Vehicle information initialization

```
#if a car is in an accident ,this defines its behavior
    if(speed[i] < 0.1 and TmpPos[0] > 10 and i!= 0 ):
        if(crash_pos[int(TmpPos[0])] == [-1,0,-1,-1,-1]):
            crash_pos[int(TmpPos[0])] = [i,1,-1,-1,-1]
            print i,"crash",TmpPos
            print "crash position",int(TmpPos[0])
            print "array",crash_pos[j][0]
        elif(crash_pos[int(TmpPos[0])][2] != -1 and crash_pos[int(TmpPos[0])][3] != -1 and crash_pos[int(TmpPos[0])][4] != -1):
            print "verified,send warning"
            warning[int(TmpPos[0])] = [i,1]

    if(speed[i] < 0.1 and TmpPos[0] > 10 and i == 0 ):
        if(crash_pos[int(TmpPos[0])] == [-1,0,-1,-1,-1]):
            crash_pos[int(TmpPos[0])] = [i,0,-1,-1,-1]
            print i,"crash",TmpPos
            print "crash position",int(TmpPos[0])
            print "array",crash_pos[j][0]
```

Define the behavior of vehicle traffic accidents (TS and PTS)

```
        #detect(10 meters) and verify
            if(TmpPos[0]!= -1 and TmpPos[0] < 710):
                for j in range(int(TmpPos[0])+40,int(TmpPos[0]),-1):
                    if(crash_pos[j][1] == 0 and str(crash_pos[j][0]) in NewVeh):
                        crash_pos[j][1] = 1
                        crash_pos[j][2] = i

                    elif(crash_pos[j][1] == 1 and str(crash_pos[j][0]) in NewVeh):
                        if(traci.vehicle.getSpeed(str(crash_pos[j][0])) < 0.1 ):
                            print i,"detected",crash_pos[j][0]
                            if(i not in [crash_pos[j][2],crash_pos[j][3],crash_pos[j][4]]):
                                if(crash_pos[j][2] == -1):
                                    crash_pos[j][2] = i
                                    print i,"verified",crash_pos[j][0]
                                elif(crash_pos[j][3] == -1):
                                    crash_pos[j][3] = i
                                    print i,"verified",crash_pos[j][0]
                                elif(crash_pos[j][4] == -1):
                                    crash_pos[j][4] = i
                                    print i,"verified",crash_pos[j][0]
                                    if(edge[i] == "1to2"):
                                        traci.vehicle.setRoute(str(i),["1to2","2to4"])
                                        print traci.vehicle.getRoute(str(i))
                                    elif(edge[i] == "2_1"):
                                        traci.vehicle.setRoute(str(i),["2_1","2to4"])

                        elif(traci.vehicle.getSpeed(str(crash_pos[j][0])) > 0.1):
                            crash_pos[j] = [-1,0,-1,-1,-1]

                    elif(crash_pos[j][1] == 1 and str(crash_pos[j][0]) not in NewVeh):
                        crash_pos[j] = [-1,0,-1,-1,-1]
```

Define the detection range of rear vehicles (TS and PTS)

```
#receive warningy(100 meters) and verify   the range can be changed
    if (ra[i] == 0):
        r = random.randint(0, 9)
        rr[i] = r
        ra[i] = 1
    else:
        r = rr[i]

    if(TmpPos[0] < 450):
        MaxRange = int(TmpPos[0])+300
    else:
        MaxRange = 749
    if(TmpPos[0]!= -1):
       for j in range(MaxRange,int(TmpPos[0]),-1):
           if(warning[j][1] == 1 and str(warning[j][0]) in NewVeh):
               if(traci.vehicle.getSpeed(str(warning[j][0])) < 0.1):
                   print i,"warning received"
                   if (r < 8):
                       if(edge[i] == "1to2"):
                           traci.vehicle.setRoute(str(i),["1to2","2to4"])
                           print traci.vehicle.getRoute(str(i))
                       elif(edge[i] == "2_1"):
                           traci.vehicle.setRoute(str(i),["2_1","2to4"])
               elif(traci.vehicle.getSpeed(str(warning[j][0])) > 0.1):
                       warning[j] = [-1,0]
               elif(str(warning[j][0]) not in NewVeh):
                   warning[j] = [-1,0]
```

Define the receive range of rear vehicles (TS and PTS)

```
#if a car is in an accident ,this defines its behavior
y = len(NewVeh)

    if(speed[i] < 0.1 and TmpPos[0] > 10 and i != 0):
        if(crash_pos[int(TmpPos[0])] == [-1,0,-1]):
            if (unc[i] == 0):
                x += 1
                unc[i] = 1

            crash_pos[int(TmpPos[0])] = [i,1,-1]
            print i,"crash",TmpPos
            q = q + 1

        elif(crash_pos[int(TmpPos[0])][2] != -1):
            print "verified,send warning"
            warning[int(TmpPos[0])] = [i,1]


    if(speed[i] < 0.1 and TmpPos[0] > 10 and i == 0 ):
        if(crash_pos[int(TmpPos[0])] == [-1,0,-1]):
            crash_pos[int(TmpPos[0])] = [i,0,-1]
            print i,"crash",TmpPos
            q = q + 1
```

Define the behavior of vehicle traffic accidents (CDPD)

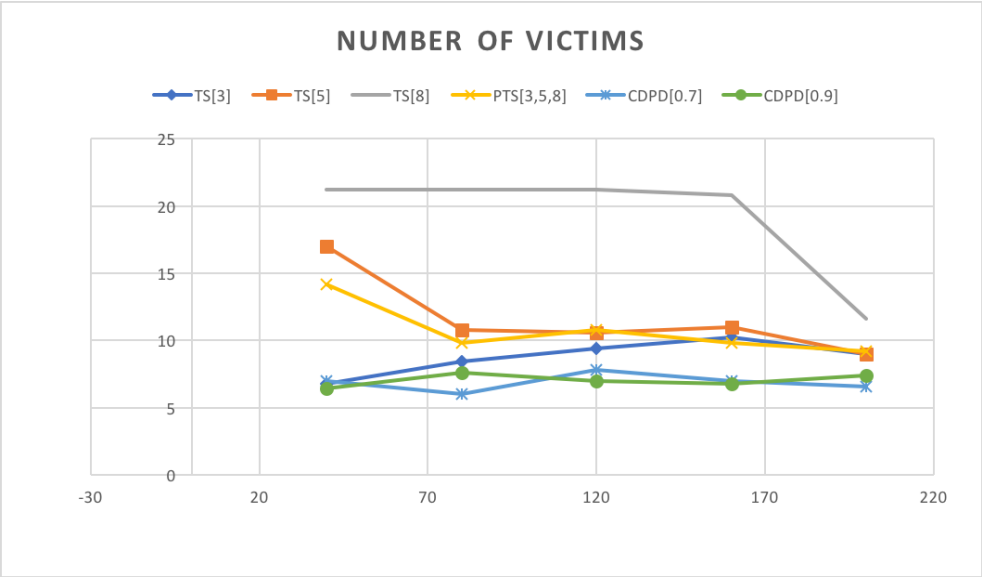
```
#detect(10 meters) and verify
    if(TmpPos[0]!= -1 and TmpPos[0] < 710):
        for j in range(int(TmpPos[0])+40,int(TmpPos[0]),-1):
            if(crash_pos[j][1] == 0 and str(crash_pos[j][0]) in NewVeh):
            print i,"detected",crash_pos[j][0],"nooo"
                crash_pos[j][1] = 1
                crash_pos[j][2] = i
            elif(crash_pos[j][1] == 1 and str(crash_pos[j][0]) in NewVeh):
                if(traci.vehicle.getSpeed(str(crash_pos[j][0])) < 0.1 ):
                    print i,"detected",crash_pos[j][0]
                    if (crash_pos[j][2] == -1):
                        crash_pos[j][2] = 1
                        print i, "verified", j
                    if(q > 5):
                        if((float(x) / float(x)) > 0.7):
                            if(edge[i] == "1to2"):
                                traci.vehicle.setRoute(str(i),["1to2","2to4"])
                                print traci.vehicle.getRoute(str(i))
                            elif(edge[i] == "2_1"):
                                traci.vehicle.setRoute(str(i),["2_1","2to4"])

                elif(traci.vehicle.getSpeed(str(crash_pos[j][0])) > 0.1):
                    crash_pos[j] = [-1,0,-1]
                    x -= 1
                    print x
            elif(crash_pos[j][1] == 1 and str(crash_pos[j][0]) not in NewVeh):
                crash_pos[j] = [-1,0,-1]
```

Define the detection range of rear vehicles (CDPD)

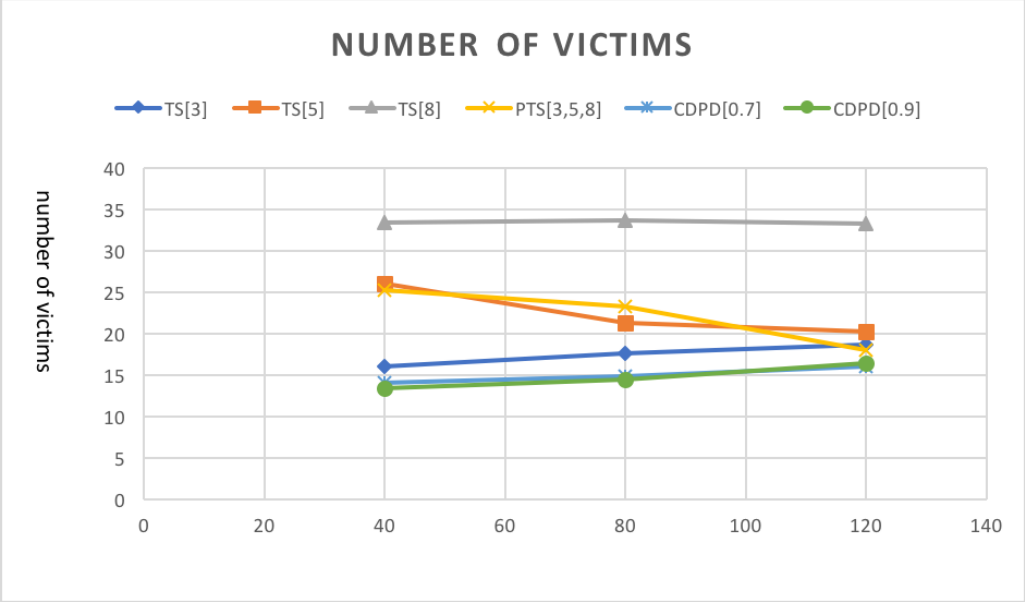
```
#receive warningy(100 meters) and verify   the range can be changed
    if(TmpPos[0] < 450):
        MaxRange = int(TmpPos[0])+300
    else:
        MaxRange = 749
    if(TmpPos[0]!= -1):
    for j in range(MaxRange,int(TmpPos[0]),-1):
        if(warning[j][1] == 1 and str(warning[j][0]) in NewVeh):
            if(traci.vehicle.getSpeed(str(warning[j][0])) < 0.1 ):
                if(q > 5):
                    if((float(x) / float(x)) > 0.7):
                        if(edge[i] == "1to2"):
                            traci.vehicle.setRoute(str(i),["1to2","2to4"])
                            print traci.vehicle.getRoute(str(i))
                        elif(edge[i] == "2_1"):
                            traci.vehicle.setRoute(str(i),["2_1","2to4"])
        elif(str(warning[j][0]) not in NewVeh):
            warning[j] = [-1,0]
```

Define the receive range of rear vehicles (CDPD)

result of real map

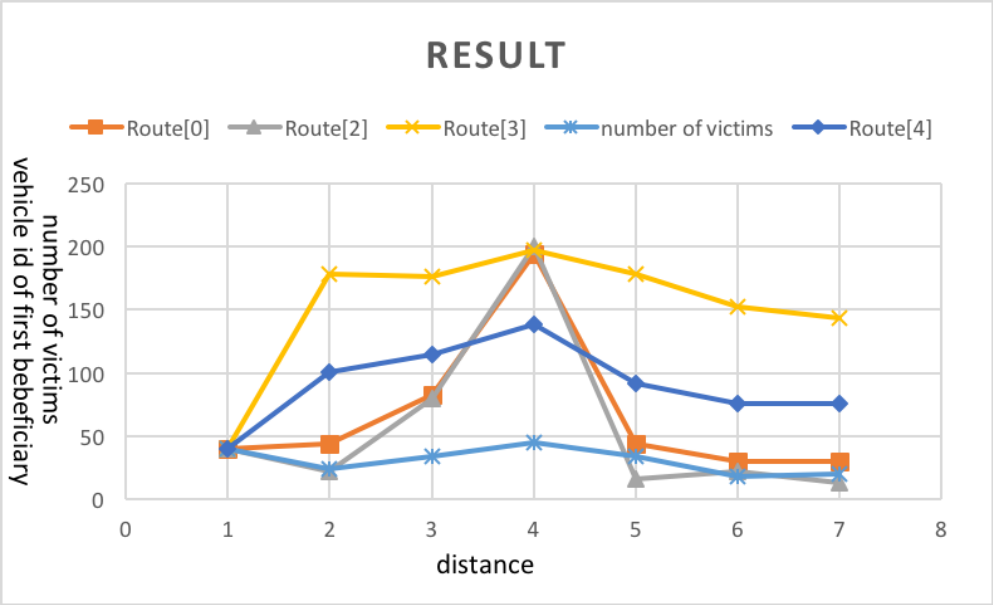


number of victims -- one point in scenario one

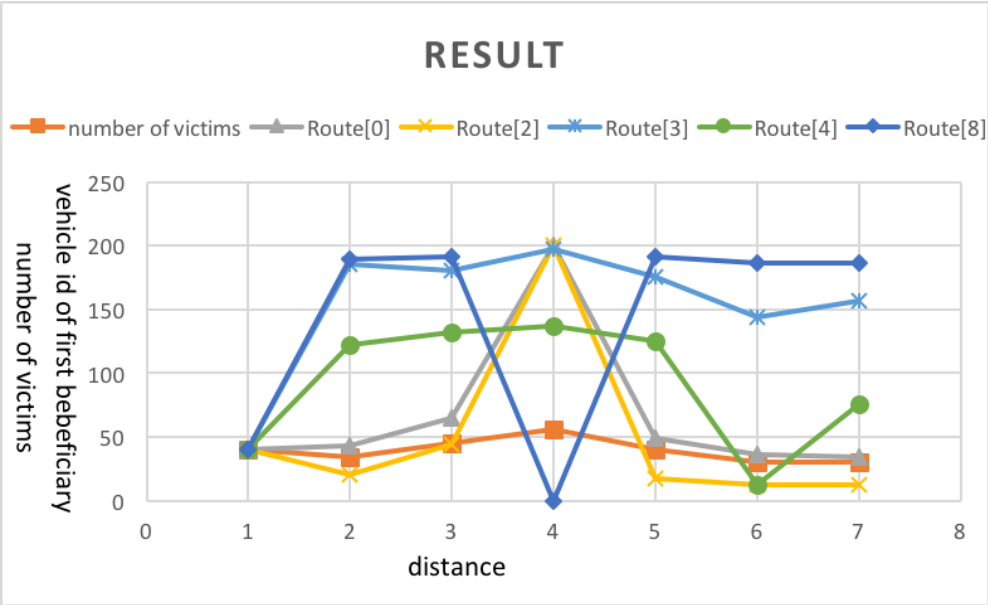


number of victims -- two point in scenario one

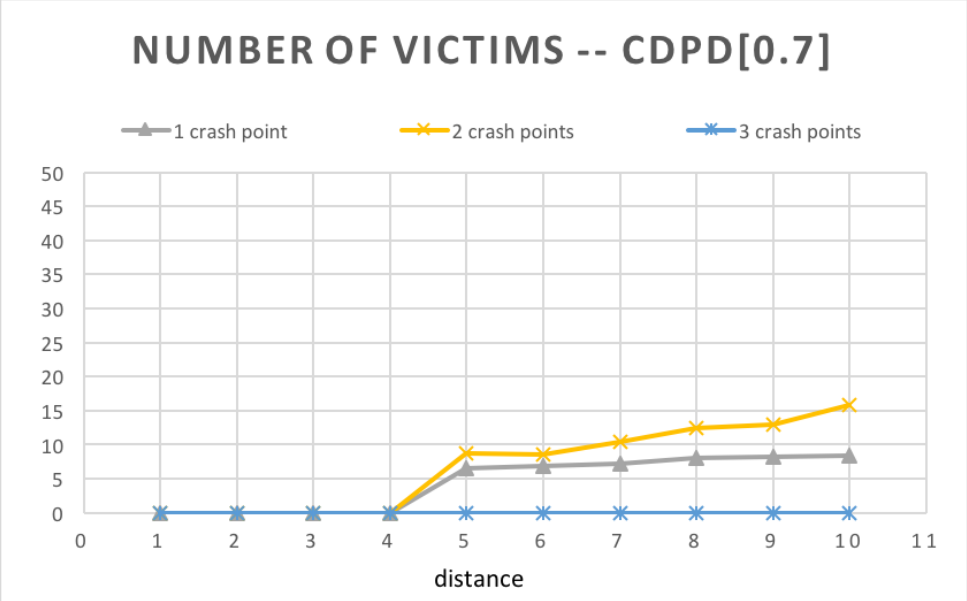
number of victims -- three point in scenario one



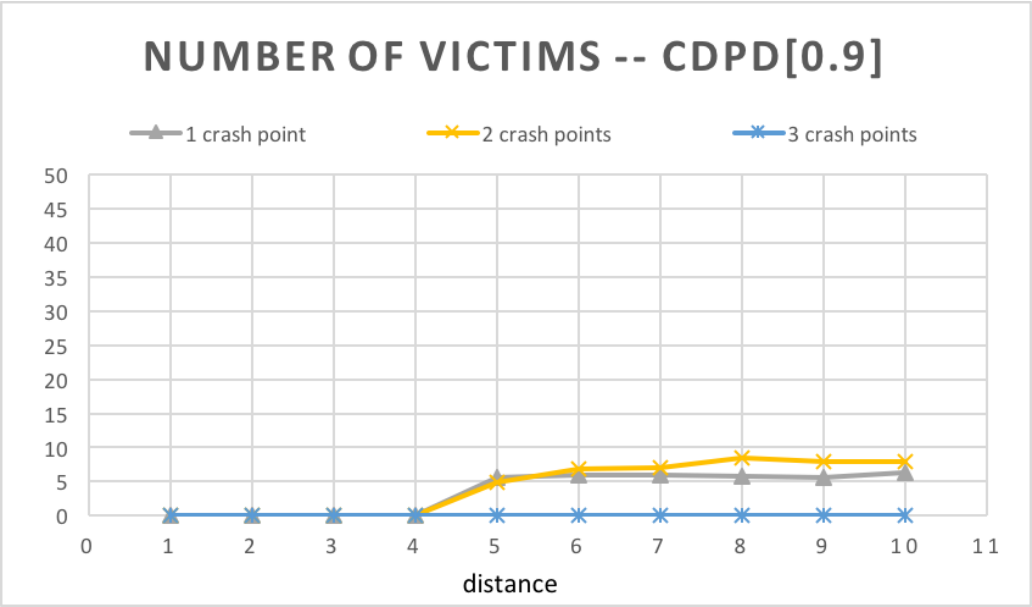
number of victims -- four point in scenario one



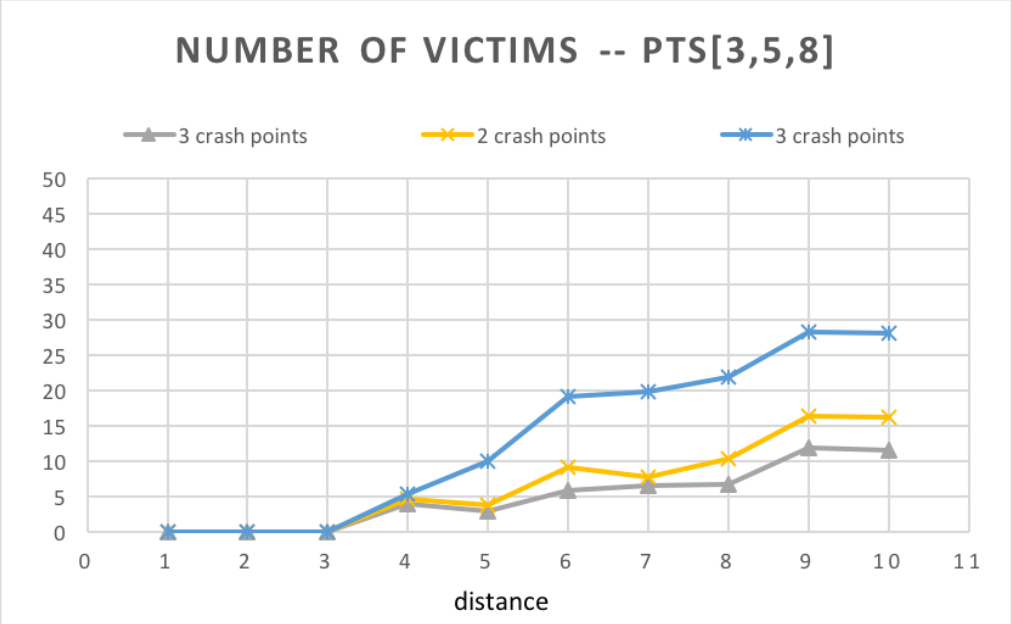
number of victims -- five point in scenario one



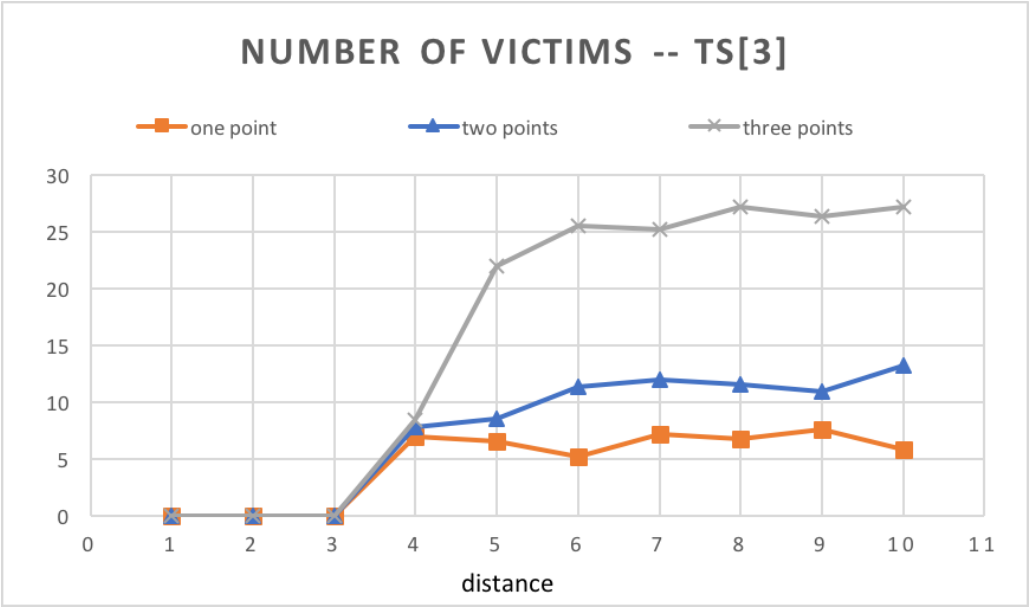
result of CDPD [0.7] in scenario two



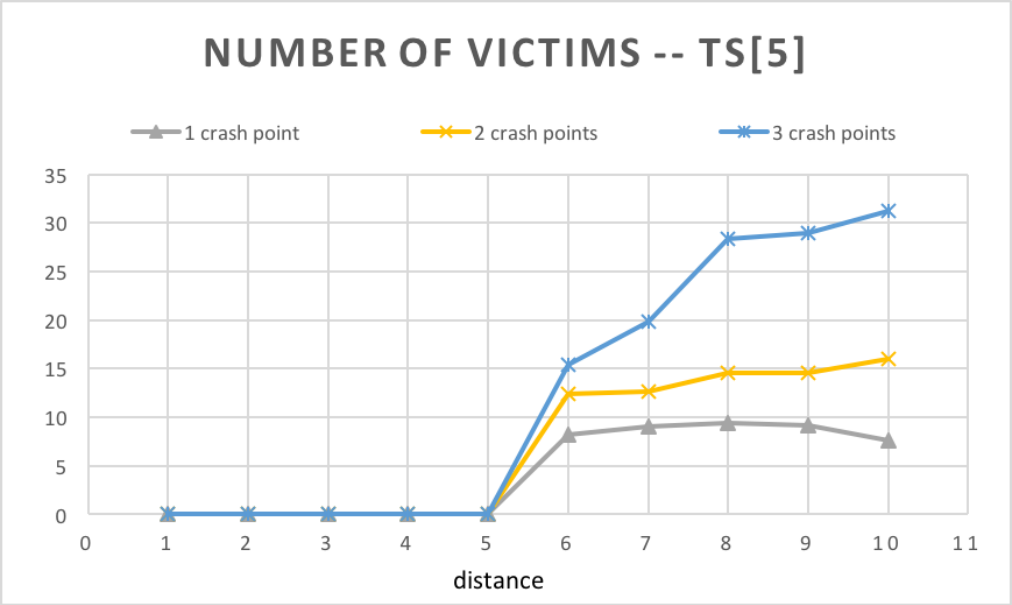
result of CDPD [0.9] in scenario two



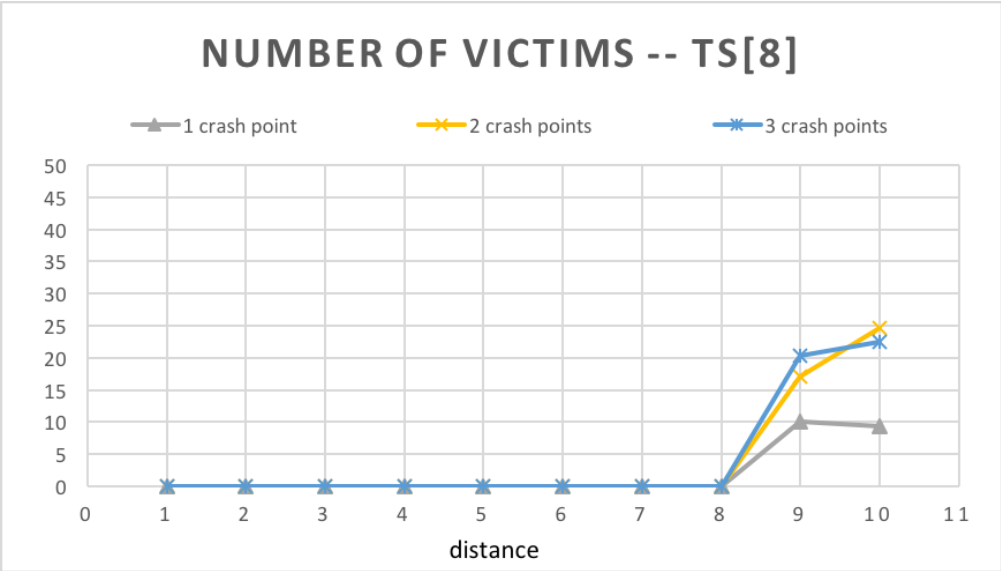
result of PTS [3,5,8] in scenario two



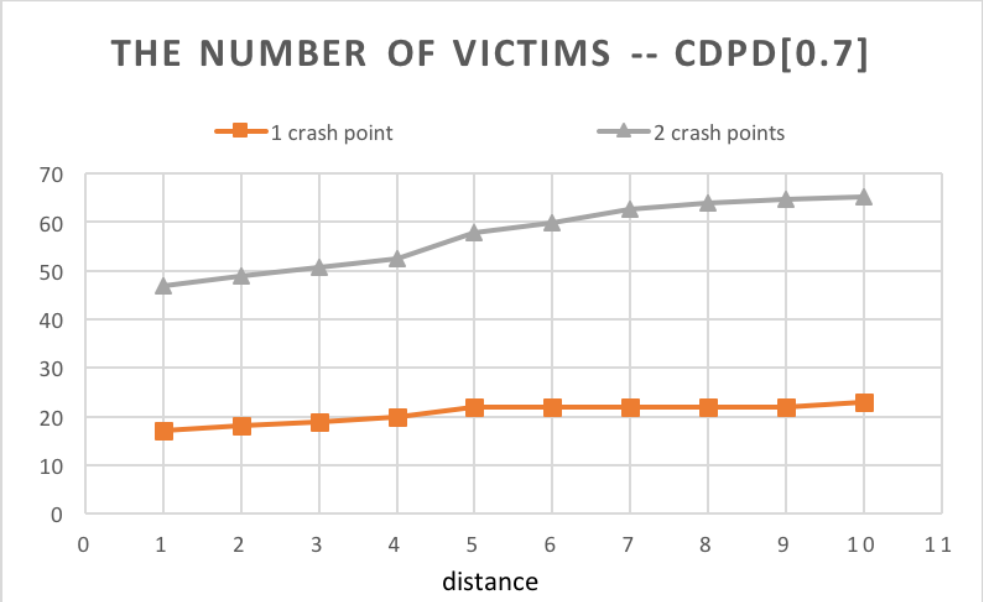
result of TS [3] in scenario two



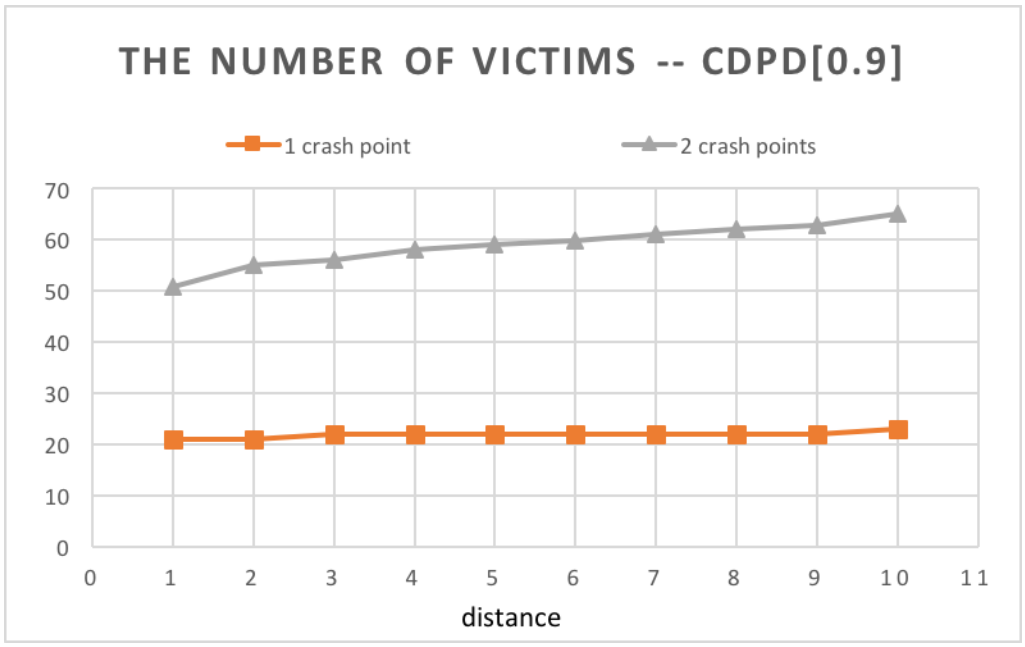
result of TS [5] in scenario two



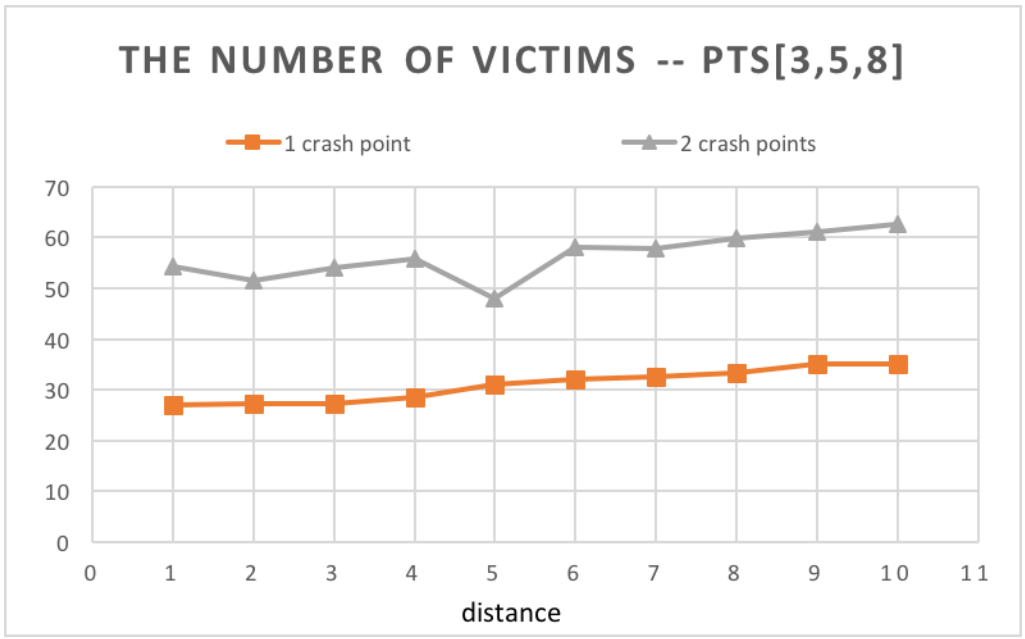
result of TS [8] in scenario two



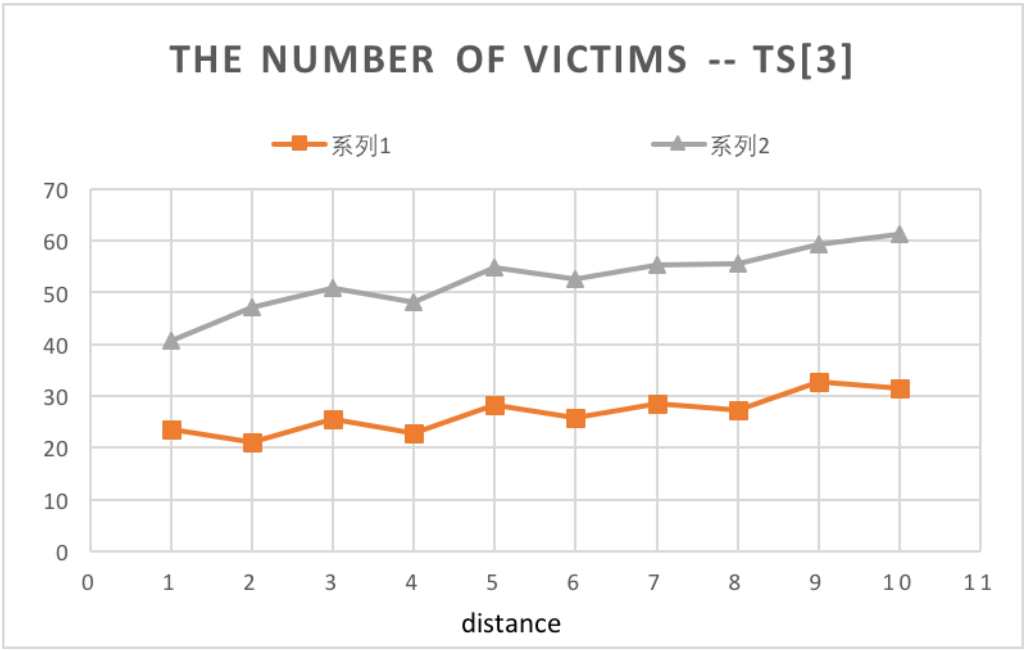
result of CDPD [0.7] in scenario three



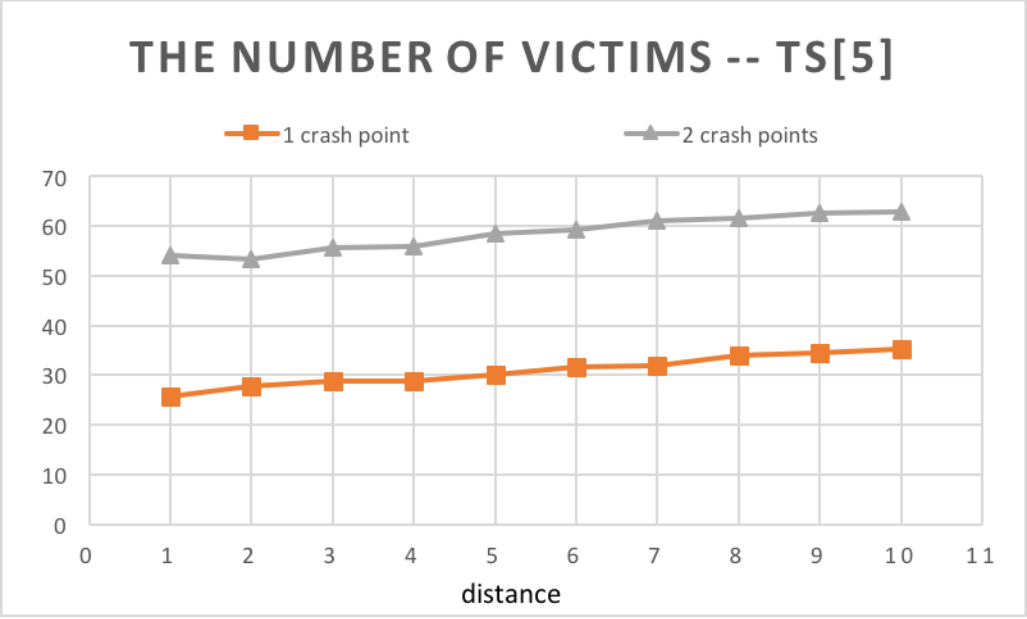
result of CDPD [0.9] in scenario three



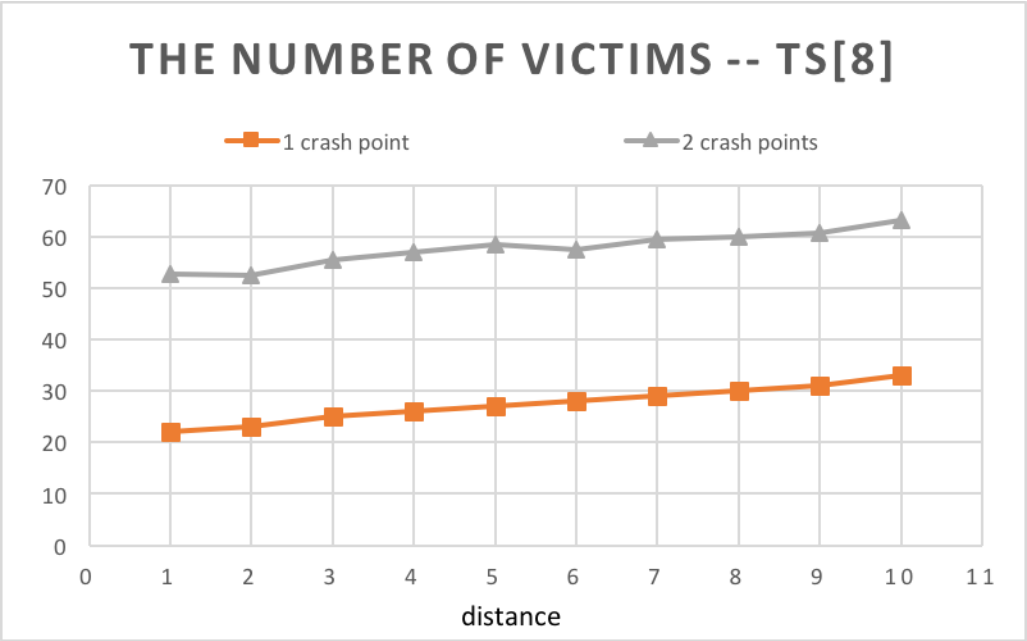
result of PTS [3,5,8] in scenario three



result of TS [3] in scenario three



result of TS [5] in scenario three



result of TS [8] in scenario three

Appendix B (Workload Table):

Milestones	Items	Time	Position
Selection of dissertation topic			
Confirm choosing 'Research on trust level warning system'			
Preparing and studying	In this part I read article in both of Simulation of Urban Mobility and Trust Level System.	**Total: 15**	
Literature review in trust level warning system build up	Article : "Secure, Privacy-preserving, Distributed Motor Vehicle Event Data Recorder"	1	Chapter 2.4
	Article : "Trust and Trusted Computing in VANET "	2	Chapter 2.4
	Article: "A methodology to increase driver trust in rear-obstacle warning systems with imperfect sensing results — Proposal for a warning system using sensor reliability information "	1	Chapter 3
	Article : "Dynamic soft drivers' trust in warning system"	2	Chapter 2.3
	Article:" Distributing Blackbox Data to Multiple Vehicles in a Secure and Privacy-preserving Manner "	2	Chapter 2.1
	Article : " Secure and privacy enhancing communications schemes for VANETs"	2	Chapter 2.4
	Article: " Security and Privacy Issues for Inter-vehicle Communications in VANETs"	2	Chapter 2.1
	Article: " Simulation of VANET Using NS-3 and SUMO "	2	Chapter 2.2
	http://sumo.dlr.de/wiki/TraCI	1	Chapter 2.2
System build up	To build the trust level warning system I proposed in this project. I try different set of operation system with different version of Simulation of Urban Mobility.	**Total: 20**	
Test different set of operation system with variable Simulation of Urban Mobility and find the best set of this project.	Test the combination with different version of Linux and SUMO.	10	Chapter 6.1
	Build up the trust level warning system in mac virtual box.	5	Chapter 6.1
	Finish build the final version of my trust level warning system.	5	Chapter 6.1
Language learning and the working principle	SUMO has it own way to define a traffic map and control the traffic simulation	**Total: 35**	

of Simulation of Urban Mobility	inside. Only C++ and python can do the traffic simulation control. And compare with the two language, I choose python to do this MSc project.		
Learning how to use python language management traffic simulation.	http://www.sumo.dlr.de/	10	Chapter 2.3
	https://www.python.org/	10	Chapter 6
	http://www.runoob.com/python/python-tutorial.html	10	Chapter 6
	https://www.zhihu.com/question/20039623	5	Chapter 6
Do the assumption of the three test scenarios.	To make the traffic simulation more comprehensive, I so the research on the previously traffic investigation. And choose the three high occur situation to be the test scenario in this MSc project. Such as situation without malicious users, situation with fake positive warning message and situation with fake negative warning message.	**Total: 8**	
Do the background research on the present most occurs traffic simulation.	Article: " Early Car Collision Prediction in VANET "	2	Chapter 4
	Article: "Evaluation of Routing Protocols for Vehicular Ad Hoc Networks in Typical Road Traffic Scenarios"	2	Chapter 4
	https://en.wikipedia.org/wiki/Traffic_collision	2	Chapter 4
	https://en.wikipedia.org/wiki/Autonomous_car	2	Chapter 4
Do the test method setting and back ground research on verifying a certain warning message.	To find three methods to verify a certain warning message. This three methods have it own advantage and disadvantage compare with each other.	**Total: 6**	
Literature review and compare with the variable method raise by the previously papers. And finally find out the three most typical one to be the simulation method implement in this project.	Article: " Anonymous Counting Problem in Trust Level Warning System for VANET "	2	Chapter 3.1
	Article: "Efficient Identity-Based Batch Verification Scheme for Vehicular Sensor Networks "	2	Chapter 3.1
	Article:" An Efficient RSU-Aided Message Authentication Scheme in Vehicular Communication Networks"	2	Chapter 3.1
Do the calculation of the selection of the three test threshold in this MSc project.	To select the best threshold value to be the test threshold in this project. Author do the pre-test to find the best threshold to be the simulation threshold in this	**Total: 10**	

	project.		
Selection of threshold	Set up a simple combination of scenario, map and situation.	5	Chapter 6.2
	Do the result data visualization and data mining.	3	Chapter 6.2
	Analysis the figure and data. Find the three best threshold for the following traffic simulation.	2	Chapter 6.2
Simple map part	I built a simple map situation to do the initial traffic simulation and get the test result. By analyzing the most direct result, I can get the most straightforward advantage or disadvantage between three different methods.	**Total: 200**	
	Design the settings of the simple map	5	Chapter 5.1
	Make 5 different random traffic flow into a rou.xml file.	3	Chapter 5.5
	Coding Threshold Method into trust level warning system.	7	Chapter .3 .1.1
	Coding Parallel Threshold Method into trust level warning system.	10	Chapter 3.1.2
	Coding Conditionality distinguishable pseudo identities approach into trust level warning system.	13	Chapter 3.1.3
	Set up the first scenario -- without malicious users.	6	Chapter 4.2
	Set up the second scenario -- with fake positive warning.	7	Chapter 4.3
	Set up the third scenario -- with fake negative warning.	6	Chapter 4.4
	Compose the three method with scenario one. And run the program. (to run this simulation, it causes long time to finish. But while I run maybe 10 hours of the simulation, I can make sure there is no mistake inside, so that I can do the further task)	10	Chapter 6
	Get the test data of scenario one and do the data visualization task. (I fill the result data into a excel file)	9	Chapter 7.1.2
	Do the initial analysis	5	Chapter 8
	Find the mistake inside of this scenario and do the debug task and make the source code more efficiency.	7	Chapter 7.1.2
	Compose the three method with scenario two. Modified the source code to suit the situation with fake positive warning. And run the program. (In this part, it cause almost 20 hours to finish doing the	13	Chapter 7.1.3

	simulation.)		
	Get the test data of scenario two and do the data visualization task.	5	Chapter 8
	Do the initial analysis	3	Chapter 7.1.3
	Find the mistake inside of this scenario and do the debug task and make the source code more efficiency.	9	Chapter 6
	Compose the three method with scenario three. Modified the source code to suit the situation with fake negative warning. And run the program. (In this part, it cause almost 20 hours to finish doing the simulation.)	12	Chapter 7.1.4
	Get the test data of scenario two and do the data visualization task.	5	Chapter 8
	Do the initial analysis	3	Chapter 7.1.4
	Find the mistake inside of this scenario and do the debug task and make the source code more efficiency.	9	Chapter 6
Block map part	I built a block map situation to do the further traffic simulation and get the test result. By analyzing the more complex result, I can get the advantage or disadvantage between three different methods more similar to the real world situation.	**Total: 150**	
	Get a real block map information from “openstreetmap” and save it to be a OSM file	1	Chapter 5.3
	Modify the OSM file and transfer it into a SUMO file	5	Chapter 5.3
	Set up the basic settings with the block map.	3	Chapter 5.3
	Make 5 different random traffic flow into a rou.xml file.	1	Chapter 5.5
	Coding Threshold Method into trust level warning system.	7	Chapter .3 .1.1
	Coding Parallel Threshold Method into trust level warning system.	4	Chapter 3.1.2
	Coding Conditionality distinguishable pseudo identities approach into trust level warning system.	7	Chapter 3.1.3
	Set up the first scenario -- without malicious users.	5	Chapter 4.2
	Set up the second scenario -- with fake positive warning.	3	Chapter 4.3
	Set up the third scenario -- with fake negative warning.	5	Chapter 4.4

	Compose the three method with scenario one. And run the program. (to run this simulation, it causes long time to finish. But while I run maybe 15 hours of the simulation, I can make sure there is no mistake inside, so that I can do the further task)	15	Chapter 6
	Get the test data of scenario one and do the data visualization task.	5	Chapter 7.2.2
	Do the initial analysis	3	Chapter 8
	Find the mistake inside of this scenario and do the debug task and make the source code more efficiency.	11	Chapter 7.2.2
	Compose the three method with scenario two. Modified the source code to suit the situation with fake positive warning. And run the program. (In this part, it cause almost 20 hours to finish doing the simulation.)	19	Chapter 7.2.3
	Get the test data of scenario two and do the data visualization task.	5	Chapter 8
	Do the initial analysis	7	Chapter 7.2.3
	Find the mistake inside of this scenario and do the debug task and make the source code more efficiency.	13	Chapter 6
	Compose the three method with scenario three. Modified the source code to suit the situation with fake negative warning. And run the program.(In this part, it cause almost 40 hours to finish doing the simulation.)	21	Chapter 7.2.4
	Get the test data of scenario two and do the data visualization task.	3	Chapter 8
	Do the initial analysis	1	Chapter 7.2.4
	Find the mistake inside of this scenario and do the debug task and make the source code more efficiency.	6	Chapter 6
Real map part	I built a real map situation to do the further traffic simulation and verify the previously result. By analyzing the more complex result, I can verify the advantage and disadvantage in the efficiency and safety of the three method. And the result almost as same as the real world situation.	**Total: 100**	
	Get a real complex map information from “openstreetmap” and save it to be a OSM file	2	Chapter 5.4

	Modify the OSM file and transfer it into a SUMO file	2	Chapter 5.4
	Set up the basic settings with the real map.	3	Chapter 5.4
	Make 5 different random traffic flow into a rou.xml file.	2	Chapter 5.5
	Coding Threshold Method into trust level warning system.	5	Chapter .3 .1.1
	Coding Parallel Threshold Method into trust level warning system.	4	Chapter 3.1.2
	Coding Conditionality distinguishable pseudo identities approach into trust level warning system.	6	Chapter 3.1.3
	Set up the first scenario -- without malicious users.	5	Chapter 4.2
	Set up the second scenario -- with fake positive warning.	3	Chapter 4.3
	Set up the third scenario -- with fake negative warning.	7	Chapter 4.4
	Compose the three method with scenario one. And run the program.	8	Chapter 6
	Get the test data of scenario one and do the data visualization task.	1	Chapter 7.1.2
	Do the initial analysis	5	Chapter 8
	Find the mistake inside of this scenario and do the debug task and make the source code more efficiency.	3	Chapter 7.1.2
	Compose the three method with scenario two. Modified the source code to suit the situation with fake positive warning. And run the program.	9	Chapter 7.1.3
	Get the test data of scenario two and do the data visualization task.	1	Chapter 8
	Do the initial analysis	3	Chapter 7.1.3
	Find the mistake inside of this scenario and do the debug task and make the source code more efficiency.	7	Chapter 6
	Compose the three method with scenario three. Modified the source code to suit the situation with fake negative warning. And run the program.	10	Chapter 7.1.4
	Get the test data of scenario two and do the data visualization task.	2	Chapter 8
	Do the initial analysis	7	Chapter 7.1.4
	Find the mistake inside of this scenario and do the debug task and make the source code more efficiency.	5	Chapter 6
Experimental summary	In this part I collect all the test data	**Total: 50**	

	during the whole project. And do the summary of all the simulation data.		
	Do the comparison of the three part and do the data visualization task.	15	Appendix
	Compare with simple map part and block part, analysis the conclusion of the three scenario and three different methods	15	Chapter 7
	Use real map part to verify the conclusion author got from the previously two parts.	15	Chapter 7
	Get the final certain conclusion of the whole traffic simulation.	5	Chapter 8
Writing the report, making slides for representing.		**Total: 70**	
	design the poster of dissertation	10	
	writing interim report	10	
	making slides for initial representation	5	
	writing the final report	30	
	making slides for final representation	15	
Total: 718			