AN ENERGY-EFFICIENT APPROACH TOWARDS NETWORK INTELLIGENCE IN COOPERATIVE COMMUNICATION IN VEHICULAR ENVIRONMENT

Progress Report

In fulfillment of the requirements for the

NU 302 R&D Project

At NIIT University

Submitted by

Kumari Renuka, Rishabh Kumar Kandoi, Isha Pali, Sai Praneeth,

Shailesh Mohta

Area

NIIT University

Neemrana

Rajasthan

CERTIFICATE

This is to certify that the present research work entitled **" An Energy-Efficient Approach Towards Network Intelligence In Cooperative Communication In Vehicular Environment"** being submitted to NIIT University, Neemrana, Rajasthan, in the fulfillment of the requirements for the course at NIIT University, Neemrana, embodies authentic and faithful record of original research carried out by ***Kumari Renuka, Rishabh Kumar Kandoi, Isha Pali, Sai Praneeth and Shailesh Mohta*** of B Tech (**Computer Science and Engineering)** at NIIT University, Neemrana,. She /He has worked under our supervision and that the matter embodied in this project work has not been submitted, in part or full, as a project report for any course of NIIT University, Neemrana or any other university.

Mr. Jetendra Joshi

Assistant Professor

NIIT University

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

RATIONAL OF THE WORK

Presently, transport is the real action of the world and basic for human advancement in this way new improvement ought to be done, better security system, greener fills, and so on. Solid data to the driver ought to send like climate, early alerts of up and coming risks. For this reason, another sort of Technology called VANET is being produced. Vehicular impromptu system is a sub substance of MANETS (Mobile specially appointed system). This implies exceptionally hub in the system can move free and remain associated. VANETS give correspondence among vehicles and vehicle to roadside units. It is exceptionally helpful in giving street wellbeing, route and different applications like short range remote correspondence and they are financially savvy.

Message to the neighbors is important in these cases:

1) Path consolidating/path changing at expressways
2) Blind sides of vehicles
3) Concealed carport crash cautioning
4) Versatile journey control and co-agent driving.
5) Roadway condition mindfulness

Hence, VANETS makes several communications possible such as- vehicle-to-vehicle(V2V) communication, vehicle-to-infrastructure(V2I), vehicle-to-user(V2U). Vehicle-To-Vehicle (V2V) communications is a system designed to transmit information between vehicles and other objects on the road in real-time. This information provides warnings to drivers and other vehicles. Instead of cars working independently, vehicles will be able to transmit vital information to nearby vehicles to improve the overall efficiency and safety of the roadways. The ultimate goal of Vehicle-To-Vehicle communication technology is to help prevent automobile crashes before they occur. Vehicle-to-infrastructure (V2I) is a communication model that allows vehicles to share information with the components that support a country's highway system. V2I sensors can capture infrastructure data and provide travelers with real-time advisories about such things as road conditions, traffic congestion, accidents, construction zones and parking availability.

On Board Units **-** Hubs are furnished with on board units and sensors which gather and process the data and send it to the street side units through remote medium. They can associate with the web through the Road side units which are understudy associated with the servers. Then again, they can send communicate messages in the event of changing path and crisis cases. Normal messages and movement related data should be possible through one bounce communicate and crisis messages should be possible through multi jump communicate, every collector will again communicate the message. It enables the drivers to take early activities to evade any sort of harm.

Road Side Units- VANETS will require an assortment of Transmission innovations like DSRC/WAVE, infrared, Cellular Telephone, 5.9 GHz short range correspondence, WiMAX, satellite, Bluetooth, RFID, and so on. IEEE802.p convention is utilized to make a message in short range for long span. Vehicles with GPS are outfitted with on board units which can convey with each other. DSRC/WAVE works in 5.9GHz or 5.8GHz with 75MHz for vehicle correspondence, extend is up to 1km with a vehicle speed.

QUALITIES OF VANETS

High unique topology: vehicles move at a rapid. For assume on the off chance that they move with the speed of 20m/s and the scope of the system is 160 mts then they can be associated just upto 8 seconds.
Visit detachment: association between vehicles is extremely weak.
Portability displaying: versatility is profoundly relying upon the driver, street show, activity.
Correspondence condition: Communication in scanty system and thick system ought to appear as something else.
Communications with the on-board sensors: position of hubs can be gotten to through sensors like GPS.

APPLICATIONS OF VANETS

The main application of the VANET includes the following listed below-

- Co-agent impact cautioning
- Path changing cautioning
- Crossing point impact cautioning
- Moving toward crisis vehicle
- Move over notice
- Work zone cautioning
- Coupling/decoupling
- Between vehicle correspondence
- Electronic toll gathering

SECURITY REQUIREMENT IN VANETS:

VANETS should give an assortment of utilizations from security admonitions to correspondence amongst vehicles and other esteem included administrations. However before executing this sort of utilization a portion of the security issues like verification must be illuminated. This is to ensure the basic data of the client from malignant clients and assaults. A mishap maintaining a strategic distance from framework requires quick and exact arrangements.

Authentication- Response of the vehicles is very reliant on the messages got by them so they should be honest to goodness to accept on them. Along these lines validation of the sender of the message is required.

Data consistency- Sender might be honest to goodness however message can be false so we have to check the message which is gone through the sender.

Location tracking- Area of the vehicle at specific purpose of time and way is considered as private data as it helps in building profile of the vehicle and in turn building profile of the driver.

Availability- It implies that each hub ought to be prepared to send message at a specific purpose of time since it can very influence the activity condition and movement security.

Non-repudiation- Drivers causing mishaps ought to be fundamentally recognized. What's more, sender ought not deny the message to be transmitted as it can be helpful in examination.

Privacy preservation- Security of the clients against the unapproved eyewitnesses ought to be ensured. This ought to be available in Vehicle to Vehicle correspondence not in Infrastructure to Vehicle correspondence. As the Infrastructure does not require any sort of protection.

Ongoing requirements- At a high portability of the hubs in the system time requirement ought to be regarded as the crisis messages ought to be moved rapidly and in time.

Event data recording- Blackbox is a gadget utilized as a part of Airplanes to record the data of the considerable number of sensors and other data. Correspondingly EDR 's (Event information recorders) can be utilized as a part of the vehicle which are valuable in circumstances like mischance.

SECURITY ISSUES AND THREATS

THREATS TO AVAILABILITY:

VANETs put an awesome test in the field of the correspondence security and they get upheaval the fields of the street wellbeing. Messages in the system will exceedingly influence the conduct of the driver and activity. In this manner, security is a central point in VANETs.

1) Dangers to Availability are-

Black Hole Attack: Nodes in the system deny to take an interest or when a set up hub drops out. This causes all the system movement to be diverted to a specific hub. Subsequently, causing disturbance in the VANET.

Malware: Malware assaults like infections can make a genuine impact typical activity. Malware assault will probably be finished by insider than pariah.

Spamming: Spam messages can bring about more transmission idleness. Furthermore, absence of concentrated organization.

Selfish Driver: Some drivers can send wrong messages to get greatest benefit from the system. For assume that he communicates something specific that there is a street blockage ahead other vehicle drivers may have faith in him and alter the course, at that point he can go effortlessly.

Malicious Attacker: They can be anybody utilizing applications accessible in vehicular system and assault it. They may approach the assets of the system. For assume an aggressor can send decelerating cautioning to make street congested and assault.

Denial of service: This is using assets with the end goal that vehicle can't perform it' s essential errand.

2) Dangers to Authentication:

Global positioning Spoofing: An assailant would you be able to GPS satellite test systems to create false readings in GPS gadgets. This will deliver trouble in land steering of the VANETS.

Pranksters: People can hack into the VANET framework and make vehicles to do anything like halting them, making them to back off.

Masquerading: An Intruder into the framework can imagine as a vehicle or crisis vehicle or police vehicle by utilizing false personalities and misrepresentation different vehicles

Message Tampering: Any hub going about as a go between two hubs for correspondence can drop, degenerate, harm, alter message and if there should be an occurrence of crisis messages it can make colossal misfortune the framework.

Dangers to Confidentiality: Mobility in VANETS is higher than MANETS and Security is significant issue than adhoc. Illicit social affair of area of a specific hub by communicate messages. So, protection issue is central point.

ID Disclosure: A worldwide eyewitness can send an infection to the neighbors of the objective hub and when the neighbor vehicles are assaulted then they get the Id of the objective hub thusly the aggressor can get the correct area of the objective hub.

Sybil Attack: Attackers can make a car influx utilizing false personalities and In turn making a rushing about among the vehicle drivers, So that they can pick a backup way to go.

Black Hole Attack- Dark gap in a system is a territory where either there is no hub or hubs in that area decline to take an interest in the system. It brings about information misfortune, and course is likewise lost, along these lines, finding another way may cost more. There are two sorts of Black gap assaults. Right off the bat, Internal Black gap assault which alludes to a hub in the system which tries to fit in the middle of way from source to goal and turns into a dynamic hub. Presently this hub is equipped for controlling information. Besides, External Black gap assault this alludes to a hub which isn't physically present in arrange and deny the entrance to the system. This can likewise turn into an inner dark gap assault when outer hub takes the control over a malevolent hub in the system. Identifying an Internal noxious hub is a troublesome assignment. along these lines, Internal Black gap assault is more helpless. Procedure of Black gap assault, in the first-place vindictive hub finds a dynamic hub and gathers the goal address. At that point it sends RREP (Route answer parcel) to the dynamic hub with mock goal address with jump tally esteems set to low and succession esteem set to high. RREP got by the closest dynamic hub builds up another way from source to goal. Presently pernicious hub will drop the information which was sent by the closest dynamic hub.

COUNTERACTIVE ACTION TECHNIQUES:

Redundant route method: In this strategy we will discover every one of the ways from source to the goal (no less than three). In this way, we have excess ways for steering. Presently source will discover most secure way considering jump tally and subsequently counteracts Black opening.

Unique Sequence number: In this technique an exceptional succession number is given to the bundles. The succession number gets amassed subsequently it's esteem is constantly higher than the past number. Along these lines, we generally store two numbers, one is the last parcel grouping number for the last bundle sent and other is the last parcel got this is refreshed every single time when a parcel is gotten or sent. Thusly we can discover vindictive hub, accordingly, maintaining a strategic distance from dark opening assault.

Time based threshold: In this strategy we consider a clock for gathering the data of time when it gets first demand. At that point we store the principal ask for time and bundle's arrangement number in Collect Route Reply Table (CRRT) and tally the timeout esteem in light of the landing time of the main course ask. Presently we can recognize the noxious hub and can discover safe course to evade dark gap.

React (Resource-Efficient-Accountability method): In this technique sender takes input from the objective hub when the bundle proportion drops then we can perceive vindictive hub and maintain a strategic distance from that course. This has three stages 1. Audit stage, 2. Search stage, 3. Identification stage

DOS ATTACK (DENIAL OF SERVICE)

In this assault assailant tries to keep the correspondence in the system. This completes more regrettable when dos is by vast lump of vehicles coming about circulated disavowal of administration. In VANET is happens in transport layer. Transport layer is in charge of end to end association. Assailant influences himself to some portion of the VANET by mocking the IP address and precluding the entrance from claiming assets of that hub. In V2V correspondence assailant begins sending consistent messages to the casualty hub keeping it occupied and thusly casualty hub can't take part in organize. In V2I correspondence assailant sends consistent messages to Road side unit and Road side unit is caught up with confirming the messages and unfit to answer to alternate hubs.

Counteractive Action Strategies:

Channel Switching: DSRC correspondence show gives 7 Channels having data transfer capacity of 10Mhz with speed of 27Mbps. So, at whatever point there is a dissent of administration we can change the channel.

Technology switching: There are numerous advances which work VANETs for interchanges V2V and V2I. So, this technique proposes that change the innovation at whatever point we see a Denial of administration.

Frequency Hopping Spread Spectrum: In this strategy bundles of messages are sent on various frequencies. At whatever point an assault is seen then Frequency is bounced.

SDN (SOFTWARE-DEFINED NETWORKING)

The physical division of the system control plane from the sending plane, and where a control plane controls a few gadgets. Software Defined Networking (SDN) is a developing design that is dynamic, reasonable, financially savvy, and versatile, making it perfect for the high-transmission capacity, dynamic nature of the present applications. This design decouples the system control and sending capacities. Empowering the system control to wind up straightforwardly programmable and the fundamental foundation to be dreamy for applications and system administrators. The OpenFlow convention is a foundational component for building SDN arrangements.

The SDN Architecture is:

Straightforwardly Programmable- System control is straightforwardly programmable in light of the fact that it is decoupled from sending capacities.

Spry- Abstracting control from sending lets heads progressively modify arrange wide movement stream to address evolving issues.

Midway Managed- System knowledge is (sensibly) incorporated in programming based SDN controllers that keep up a worldwide perspective of the system, which appears to applications and strategy motors as a solitary, legitimate switch.

Automatically Configured- SDN lets arrange administrators design, oversee, secure, and upgrade organize assets rapidly by means of dynamic, computerized SDN programs, which they can think of themselves in light of the fact that the projects don't rely upon restrictive programming.

Open Standards-Based and Vendor-Neutral- At the point when actualized through open benchmarks, SDN disentangles organize plan and activity since guidelines are given by SDN controllers rather than different, seller particular gadgets and conventions.

FOG COMPUTING

Fog computing is a term made by Cisco that alludes to stretching out distributed computing to the edge of an undertaking system. Otherwise called Edge Computing or fogging, fog computing encourages the activity of figure, stockpiling and systems administration benefits between end gadgets and distributed computing server farms. Cisco presented its fog computing vision in January 2014 as a method for conveying distributed computing abilities to the edge of the system and subsequently, nearer to the quickly developing number of associated gadgets and applications that expend cloud benefits and produce progressively huge measures of information.

LITERATURE REVIEW

1. "Vehicular Ad Hoc Networks (VANETS): Status, results and challenges"

- Sherali Zeadally, Ray Hunt, Yuh-Shyan Chen, Angela Irwin, Aamir Hassan

Late Advances In hardware, software, and correspondence innovations are empowering the plan and execution of an entire scope of various kinds of networks that are being conveyed in different conditions. One such network that has received a lot of interest in the last couple of a long time is the Vehicular Ad-Hoc Network (VANET). VANET has turned into a dynamic zone of research, institutionalization, and development because it has tremendous potential to improve vehicle and road wellbeing, traffic efficiency, and accommodation and also solace to the two drivers and travelers. Late research endeavors have set a solid accentuation on novel VANET outline models and usage. A ton of VANET explore work have concentrated on specific zones including routing, broadcasting, Quality of Service(QoS), and security. We study a portion of the current research brings about these areas. We present Review of Wireless access Standards for VANETs and depict a portion of the current VANET trials and arrangements in the US, Japan, and the European Union. In addition, we likewise briefly introduce a portion of the test systems right now accessible to VANET specialists for VANET recreations and we evaluate their benefits and limitations. Finally, we diagram a portion of the VANET inquire about difficulties that still should be addressed to empower the universal sending and across the board adoption of versatile, solid, strong, and secure VANET models, conventions, advancements, and administrations.

The merging of processing, broadcast communications (fixed and versatile), and different sorts of administrations are empowering the arrangement of various types of VANET advances. In the previous decade, numerous VANET extends far and wide have been attempted and a few VANET models have been created to enhance vehicle-to-vehicle or vehicle-to infrastructure correspondences. In this work, we looked into a portion of the principle territories that scientists have concentrated on over the most recent couple of years and these incorporate security, directing, QoS, and broadcasting methods and we featured the most notable outcomes accomplished to date. We displayed an exhaustive examination of different recreation instruments that are accessible for VANET reproductions. We trust this scientific classification on VANET test systems will be useful to future VANET specialists in picking the ideal VANET test system most appropriate for their VANET outline objectives. At long last, we talked about a portion of the difficulties that still should be addressed keeping in mind the end goal to empower the sending of VANET advancements, frameworks, and administrations cost-viably, safely, and dependably

2. "Vehicular Cloud Networking: Architecture and Design Principles"

- Euisin Lee, Eun-Kyu Lee, Soon Y. Oh, and Mario Gerla

Over the past several decades, VANET has been a core networking technology to provide safety and comfort to drivers in vehicular environments. Emerging applications and services, however, require major changes on its underlying computing and networking models, which demands new network planning for VANET.
The article briefs about the emerging VANET applications and gives three noticeable characteristics, which cannot be supported efficiently by the existing VANET technology (1) vehicles produce and consume a great amount of contents having the property of local relevance (time, space, and consumer); (2) vehicles simply seek contents regardless of their providers; (3) vehicles collaborate using their resources to create

value-added services with minimum help from the Internet infrastructure. To accommodate such characteristics, the article introduces a new VANET network planning, consisting of two recent paradigms - Vehicular Cloud Computing and Information Centric Networking. Thus, the article examines how VANET evolves with the two paradigms. As a computing model, VCC enables vehicles to discover and share their resources so as to create a vehicle cloud on which they collaborate to produce value-added services. ICN is leveraged, as a networking model, to disseminate cloud contents efficiently among vehicles. This article envisions a new vehicular networking system, Vehicular Cloud Networking that is built on top of Vehicular Cloud Computing and Information Centric Networking. It then discusses the fundamentals of Vehicular Cloud Computing(VCN), its system operations (how the VCN system operates to establish a virtual computing platform and to enable cloud type collaboration in it), the services VCN has to offer, and the design principles of VCN from System Perspective, Networking Perspective and Service Perspective.

3. “Software Defined Networking-based Vehicular Adhoc Network with Fog Computing”

- Nguyen B.Truong, Gyu Myoung and Yacine Ghamri-Doudan

Vehicular Adhoc Networks (VANETs) have been attracted a lot of research recent years. Although VANETs are deployed in reality offering several services, the current architecture has been facing many difficulties in deployment and management because of poor connectivity, less scalability, less flexibility and less intelligence. In this paper, the authors present a new VANET architecture called FSDN which combines two emergent computing and network paradigm Software Defined Networking (SDN) and Fog Computing as a prospective solution. SDN-based architecture provides flexibility, scalability, programmability and global knowledge while Fog Computing offers delay-sensitive and location-awareness services which could be satisfy the demands of future VANETs scenarios. FSDN VANET architecture is operated in heterogeneous network environment in which forwarding infrastructure use several wireless technologies to communicate. V2V and V2I could be wireless connection or WAVE. They have figured out all the SDN-based VANET components as well as their functionality in the system. They have also considered the system basic operations in which Fog Computing are leveraged to support surveillance services by taking into account resource manager and Fog orchestration models. The proposed architecture could resolve the main challenges in VANETs by augmenting Vehicle-to-Vehicle (V2V), Vehicle-to-Infrastructure (V2I), Vehicle-to-Base Station communications and SDN centralized control while optimizing resources utility and reducing latency by integrating Fog Computing. Streaming applications have been dramatically increased and contribute a significant amount of traffic in network. The key point is how to distribute data streaming traffic from data streaming server to some nodes in a vehicular network which is high mobility and topology changes; and low delay requirements. Conventional approaches for Lane-Change service face many challenges since it requires planning at microscopic traffic information such as road conditions and traffic flow variables. Solution to the challenges faced by the Lane-Change service and data streaming services is presented in this paper. Two use-cases for non-safety service (data streaming) and safety service (Lane-change assistance) are also presented to illustrate the benefits of the proposed architecture.

4. “Network Intelligence Based on Network State Information for Connected Vehicles Utilizing Fog Computing”

- Seongjin Park and Younghwan Yoo

This paper proposes a method to take advantage of fog computing and SDN in the connected vehicle environment, where communication channels are unstable and the topology changes frequently. A controller knows the current state of the network by maintaining the most recent network topology. Of all the information collected by the controller in the mobile environment, node mobility information is particularly important. Thus, they have divided the nodes into three classes according to their mobility types and use the irrelated attributes to efficiently manage the mobile connections. In this paper, they measure the signal strength of the link between the controller and each switch in order to help the controller supervise global network resources. They further propose an intelligent maintenance method utilizing network information and give some practical use cases for connected vehicles. The presented approach utilizes mobility information to reduce control message overhead by adjusting the period of beacon messages and successfully support efficient failure recovery. They have focused on two recovery process. One is to recover the connection failures using only mobility information, and the other is to suggest a real-time scheduling algorithm to recover the services for the vehicles that lost connection in the case of a fog server failure. A real-time scheduling method is first described and then evaluated. Algorithm1 shows the controllers decision process. When the controller forecasts a connection loss, it is classified as either a temporal or a severe failure. The recovery process then proceeds is shown in Algorithm 2. They constructed the campus network and have demonstrated the methodology for the reduction of control overhead and the connection recovery by using network simulator Simulation called SUMO. Various simulation experiments have been conducted based upon several parameters such as Control Message Overhead, Connection Lost Time and Service Deadline Miss Ratio. The simulation results show that control message overhead and failure recovery time are decreased by approximately 55% and 5%, respectively.

5. "A Security and Privacy Review of VANETs"

- Fengzhong Qu, Zhihui Wu, Fei-Yue Wang and Woong Cho, Member

Vehicular ad hoc networks (VANETs) have been quite a hot research area in the last few years. Due to their unique characteristics such as high dynamic topology and predictable mobility, VANETs attract so much attention of both academia and industry. Vehicular ad hoc networks would bring a new driving experience to drivers once it is deployed. However, communicating in an open-access environment makes security and privacy issues a real challenge, which may affect the large-scale deployment of VANETs. Researchers have proposed many solutions to these issues. This paper provides background information of VANETs and classifying security threats that challenge VANETs. After clarifying the requirements that the proposed solutions to security and privacy problems in VANETs should meet, on the one hand, they present the general secure process and point out authentication methods involved in these processes. Detailed survey of these authentication algorithms followed by discussions comes afterward. On the other hand, privacy preserving methods are reviewed, and the tradeoff between security and privacy is discussed. Finally, they provide an outlook on how to detect and revoke malicious nodes more efficiently and challenges that have yet been solved.

6. "A Survey on the Security of Stateful SDN Data Planes"

- Tooska Dargahi, Alberto Caponi, Moreno Ambrosin, Giuseppe Bianchi and Mauro Conti

This paper has a twofold goal: i) to provide the reader with background on the (novel) trend of stateful SDN data planes, and ii) dissect the relevant security issues, also via a concrete analysis of selected use case applications. After a brief background on the basic concepts behind Software-Defined Networking (SDN), OpenFlow, and the relevant evolution occurred after the first OpenFlow standardization, the paper reviews the state of the art in the area of stateful SDN data planes. Having programmable SDN switches enhances the network performance and flexibility in response to real-time network applications, since it allows the switches to manage dynamic applications faster and more efficiently. Stateful SDN data plane, although practically competent, is prone to various security attacks. Furthermore, the paper lists some relevant use cases enabled by stateful SDN such as Stateful Failure Recovery, HULA, Port Knocking, DNS Tunneling, Stateful Detection and UDP Flooding Stateful Mitigation. There are several different schemes proposed in the literature (OpenState, FAST, SDPA, RMT, P4, Domino, SNAP, Event-driven Programming), each of which addressing a specific need in bringing stateful SDN data plane into reality (such as different platforms, programming languages or applications). However, none of them has built-in security measures. Therefore, the paper reviews and analyzes the main features of the stateful data plane schemes in the literature, followed by highlighted major security vulnerabilities that are in place due to the intrinsic properties of the stateful schemes such as Unbounded flow state memory allocation, Triggerable CPU intensive operations, *Lack of authentication mechanisms in the data plane,* Lack of central data plane state management. Furthermore, the paper provides attack examples to the existing stateful SDN schemes, exploiting these vulnerabilities. Attacks being Switch Memory Saturation Attack, State Inconsistency Attack, CPU Exhaustion Attack and to conclude, it provided some basic recommendations to be considered in new stateful SDN proposals to cope with the explained vulnerabilities, as well as possible future research directions.

7. “Securing Software Defined Networks: Taxonomy, Requirements, and Open Issues”

- Adnan Akhunzada, Ejaz Ahmed, Abdullah Gani, Muhammad Khurram Khan, Muhammad Imran, and Sghaier Guizani

The emergence of SDNs promises to dramatically simplify network management and enable innovation through network programmability. Despite all the hype surrounding SDNs, exploiting its full potential is demanding. Security is still the key concern and is an equally striking challenge that reduces the growth of SDNs. Moreover, the deployment of novel entities and the introduction of several architectural components of SDNs pose new security threats and vulnerabilities. Besides, the landscape of digital threats and cyber-attacks is evolving tremendously, considering SDNs as a potential target to have even more devastating effects than using simple networks. Security is not considered as part of the initial SDN design; therefore, it must be raised on the agenda. This article discusses the state-of-the-art security solutions proposed to secure SDNs. Since security is not considered initially as part of SDN design, each layer of an SDN has its own security implications and requirements. Moreover, establishing trust throughout an SDN is even more critical. In this paper, they have presented the classification of the different security solutions, the thematic taxonomy based on SDN layers/interfaces, security measures, simulation environments, and security objectives. Moreover, this article points out the possible attacks and threat vectors targeting different layers/interfaces of SDNs. The potential requirements and their key enablers for securing SDNs are also identified and presented. Also, the article gives great guidance for secure and dependable SDNs. It presents the various requirements and key enablers for SDN security such as securing the SDN controller, protecting the flow paradigm of the SDN, fortifying SDN agents and hardening application programming interfaces and communication channels. Finally, they discuss open issues and challenges of SDN security which includes the issues related to SDN while creating a virtual network such as Denial of Service Attack, Spoofing Attack and Malicious Injection that may be deemed appropriate to be tackled by researchers and professionals in the future.

8. “On the Effectiveness of Changing Pseudonyms to Provide Location Privacy in VANETs”

- Levente Butty´an, Tam´as Holczer, and Istv´an Vajda (2007)

The promise of vehicular communications is to make road traffic safer and more efficient. Vehicular communications will play a central role in this effort, enabling a variety of applications for safety, traffic efficiency, driver assistance, and entertainment. However, besides the expected benefits, vehicular communications also introduce some privacy risk by making it easier to track the physical location of vehicles. In particular, many envisioned safety related applications require that the vehicles continuously broadcast their current position and speed in so called heart beat messages. This allows the vehicles to predict the movement of other nearby vehicles and to warn the drivers if a hazardous situation is about to occur. While this can certainly be advantageous, an undesirable side effect is that it makes it easier to track the physical location of the vehicles just by eavesdropping these heart beat messages. One approach to solve this problem is that the vehicles use pseudonyms that they change with some frequency. In this paper, we study the effectiveness of this approach. We define a model based on the concept of the mix zone, characterize the tracking strategy of the adversary in this model, and introduce a metric to quantify the level of privacy enjoyed by the vehicles. They also report on the results of an extensive simulation where they have used their model to determine the level of privacy achieved in realistic scenarios. In particular, in their simulation, they used a rather complex road map, generated traffic with realistic parameters, and varied the strength of the adversary by varying the number of her monitoring points. Their simulation results provide detailed information about the relationship between the strength of the adversary and the level of privacy achieved by changing pseudonyms. In this paper, they abstracted away the frequency with which the pseudonyms are changed, and they simply assumed that this frequency is high enough so that every vehicle surely changes pseudonym while in the mix zone.

9. "Pseudonym Changing at Social Spots: An Effective Strategy for Location Privacy in VANETs"

- Rongxing Lu, Xiaodong Lin, Tom H. Luan, Xiaohui Liang, and Xuemin (Sherman) Shen

As a prime target of Quality of Privacy (QoP) in vehicular ad hoc networks (VANETs), location privacy is imperative for the full flourish of VANETs. Although frequent pseudonym changing provides a promising solution for location privacy in VANETs, if the pseudonyms are changed in an improper time or location, such a solution may become invalid. To cope with the issue, this paper presents an effective pseudonym changing at social spots (PCS) strategy to achieve the provable location privacy. They, first introduce the social spots where many vehicles may gather, e.g., a road intersection when the traffic light turns red or a free parking lot near a shopping mall. By taking the anonymity set size (ASS) as the location privacy metric, then they develop two anonymity set analytic models to quantitatively investigate the location privacy achieved by the PCS strategy. In addition, they used game theoretic techniques to prove the feasibility of PCS strategy in practice. Extensive performance evaluations were conducted to demonstrate that better location privacy can be achieved when a vehicle changes its pseudonyms at some highly social spots, and the proposed PCS strategy can assist vehicles to intelligently change their pseudonyms at the right moment and place.

10. "Privacy Protection with Dynamic Pseudonym-Based Multiple Mix-Zones Over Road Networks"

- Qasim Ali Arain , Zhongliang Deng, Imran memon , Asma Zubedi, Jichao Jiao, Aisha Ashraf, Muhammad Saad Khan

In this paper they proposed a strategy for location privacy protection which addresses the issues related with existing location privacy protection techniques. Mix-Zones and pseudonyms are considered as the basic building blocks for location privacy; however, continuously changing pseudonyms process at multiple locations can enhance user privacy. It has been revealed that changing pseudonym at improper time and location may threat to user's privacy. Moreover, certain methods related to pseudonym change have been proposed to attain desirable location privacy and most of these solutions are based upon velocity, GPS position and direction of angle. They analyzed existing methods related to location privacy with mix zones, such as RPCLP, EPCS and MODP, where it has been observed that these methods are not adequate to attain desired level of location privacy and suffered from large number of pseudonym changes. By analyzing limitations of existing methods, they proposed Dynamic Pseudonym based multiple mix zone (DPMM) technique, which ensures highest level of accuracy and privacy. they simulated their data by using SUMO application and analysis results has revealed that DPMM outperformed existing pseudonym change techniques and achieved better results in terms of acquiring high privacy with small number of pseudonym change.

11. “A survey of black hole attacks in wireless mobile ad hoc networks”

- Fan-Hsun Tseng, Li-Der Chou and Han-Chieh Chao

The dark gap assault is one of the outstanding security dangers in remote portable ad hoc networks. The interlopers use the escape clause to do their vindictive practices on the grounds that the course revelation process is important and inescapable. Numerous scientists have directed distinctive location procedures to propose diverse sorts of recognition plans. In this paper, we overview the current arrangements and examine the best in class directing strategies. We not just arrange these propositions into single dark gap assault and community dark opening assault yet in addition break down the classifications of these arrangements and give a correlation table. We hope to outfit more scientists with a nitty gritty work in expectation.

Because of the intrinsic plan disadvantages of steering convention in MANETs, numerous analysts have led various procedures to propose diverse kinds of avoidance instruments for dark gap issue. In this paper, we first synopsis the advantages and disadvantages with famous steering convention in remote portable ad hoc networks. At that point, the state-of the-workmanship steering techniques for existing arrangements are ordered and talked about. The recommendations are introduced in a sequential request and isolated into single dark gap and communitarian dark gap assault. As indicated by this work, we watch that both of proactive directing and responsive steering have specific aptitudes. The proactive recognition strategy has the better parcel conveyance proportion and right location likelihood yet experienced the higher directing overhead due to the intermittently broadcast bundles. The responsive identification technique takes out the steering overhead issue from the occasion driven way, however experienced some bundle misfortune in the start of directing methodology. Along these lines, we prescribe that a half and half location technique which joined the advantages of proactive steering with receptive steering is the propensity to future research course. Be that as it may, we additionally find that the assailant's mischief activity is the key factor. The assailants can keep away from the location instrument, regardless of what sorts of directing recognition utilized. Likewise, some key encryption strategies or hash-based techniques are abused to take care of this issue. The dark gap issue is as yet a dynamic research territory. This paper will profit more scientists to understand the present status quickly.

OBJECTIVE

Vehicular Adhoc Networks (VANETs) have been attracted a lot of research recent years. Although VANETs are deployed in reality offering several services, the current architecture has been facing many difficulties in deployment and management because of poor connectivity, less scalability, less flexibility and less intelligence. Software-defined networking is an introduction of software to the traditional networking leading to more control over the networking devices, it centralizes the control to the SDN controller. The switches are programmable hence makes the placement of new rules in switches easier. Fog computing extends the services of the cloud computing to the edge of the enterprise network leading to the provision of real time services in VANET architecture. SDN-based architecture provides flexibility, scalability, programmability and global knowledge while Fog Computing offers delay-sensitive and location-awareness services which could be satisfy the demands of future VANETs scenarios. Integration of the technologies such as fog computing, software-defined networking and 5G services have led to several new opportunities and solves the challenges related to management and deployment of the VANET architecture.

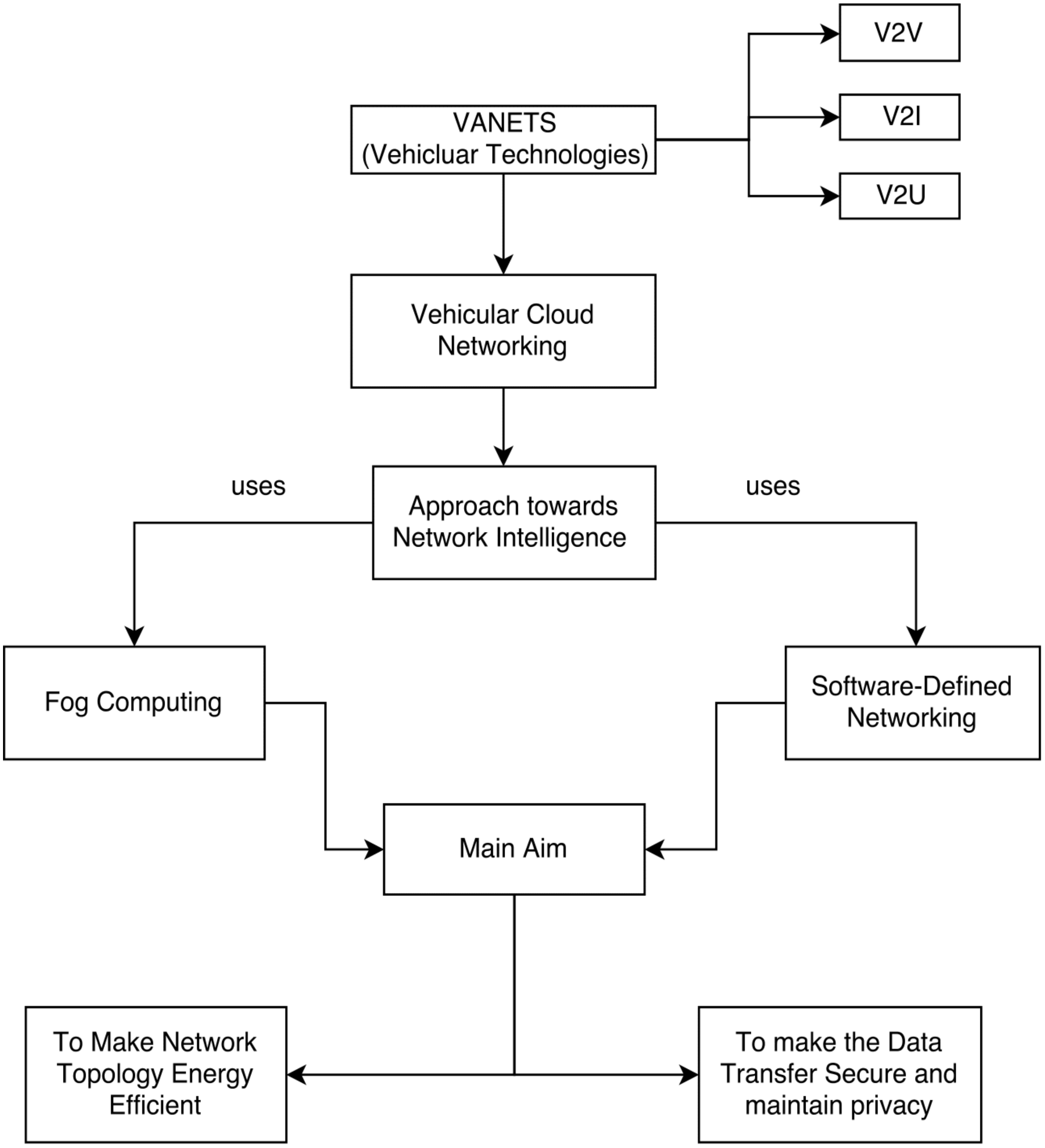
In Connected Vehicle Environment, the communication channels are unstable and the topology changes frequently, Communication between the Server and the mobile devices can use various wireless technologies. Efficient resource and energy management is essential but difficult because of the inflexibility of the current network structure. One switches or router have to perform all the functions including forwarding, routing and management of the network resources. Hence, due to increase in the demand of these services have raised big concern over the amount of energy consumed. Solution must be provided to reduce the amount of energy consumption in the vehicular environment. There is a huge variation in the traffic during the night and the day time, traffic during the night time is less during the day time, this leads to an opportunity to optimize the consumption of energy in VANETS.

Besides the consumption of energy in VANET, another major challenge faced in VANET architecture is the security. The emergence of SDNs promises to dramatically simplify network management and enable innovation through network programmability. As SDN relies on centralized network management, it adds to administrators' worries regarding server (controller) security. If by any means server gets hacked, then whole network becomes more prone to be attacked. Despite all the hype surrounding SDNs, exploiting its full potential is demanding. Security is still the key concern and is an equally striking challenge that reduces the growth of SDNs. However, communicating in an open-access environment makes security and privacy issues a real challenge, which may affect the large-scale deployment of VANETs. This provides us an opportunity to work on the security related issues.

OBJECTIVE

The main objective is to design an efficient network management strategy with guaranteed satisfaction of network traffic demand utilising SDN and fog computing in connected vehicular environment and to ensure secure data communication in VANETs.

FIG 1: FLOW OF NARROWING DOWN OF OBJECTIVE


VANETS
(Vehicluar Technologies)
V2V
V2I
V2U
Vehicular Cloud Networking
uses
Approach towards Network Intelligence
uses
Fog Computing
Software-Defined Networking
Main Aim
To Make Network Topology Energy Efficient
To make the Data Transfer Secure and maintain privacy

METHODOLOGY

SIMULATION TOOLS USED

A. SUMO

ABOUT

"Simulation of Urban Mobility", or "SUMO" for short, is an open source, microscopic, multi-modal traffic simulation. It allows to simulate how a given traffic demand which consists of single vehicles moves through a given road network. The simulation allows to address a large set of traffic management topics. It is purely microscopic: each vehicle is modelled explicitly, has an own route, and moves individually through the network. Simulations are deterministic by default but there are various options for introducing randomness.

FEATURES

- Includes all applications needed to prepare and perform a traffic simulation (network and routes import, DUA, simulation)
- Simulation
- Space-continuous and time-discrete vehicle movement
- Different vehicle types
- Multi-lane streets with lane changing
- Different right-of-way rules, traffic lights
- A fast openGL graphical user interface
- Manages networks with several 10.000 edges (streets)
- Fast execution speed (up to 100.000 vehicle updates/s on a 1GHz machine)
- Interoperability with other application at run-time
- Network-wide, edge-based, vehicle-based, and detector-based outputs
- Supports person-based inter-modal trips
- Network Import
- Imports VISUM, Vissim, Shapefiles, OSM, RoboCup, MATsim, OpenDRIVE, and XML-Descriptions
- Missing values are determined via heuristics
- Routing
- Microscopic routes - each vehicle has an own one
- Different Dynamic User Assignment algorithms

- High portability
- Only standard C++ and portable libraries are used
- Packages for Windows main Linux distributions exist
- High interoperability through usage of XML-data only
- Open source (**EPL**)

USAGE EXAMPLES

Since 2001, the SUMO package has been used in the context of several national and international research projects. The applications included:

- traffic lights evaluation
- route choice and re-routing
- evaluation of traffic surveillance methods
- simulation of vehicular communications
- traffic forecast

INSTALLATION PROCEDURE

Linux

If you run debian or ubuntu, SUMO is part of the regular distribution and can be installed like this:

```
sudo apt-get install sumo sumo-tools sumo-doc
```

If you need a more up-to-date ubuntu version, it may be found in a separate ppa, which is added like this:

```
sudo add-apt-repository ppa:sumo/stable
sudo apt-get update
```

and then again

```
sudo apt-get install sumo sumo-tools sumo-doc
```

Precompiled binaries for different distributions like openSUSE and Fedora can be found at these repositories for binary Linux versions. These repositories contain nightly builds as well. In the case your system is not listed here or you need to modify the sources, you have to build SUMO from sources.

B. SIMULATE TRAFFIC USING OPENSTREETMAP-DATA AND SUMO- SIMULATION FOR URBAN MOBILITY

ABOUT

OpenStreetMap (OSM) is a collaborative project to create a free editable map of the world. The creation and growth of OSM has been motivated by restrictions on use or availability of map information across much of the world, and the advent of inexpensive portable satellite navigation devices. OSM is considered a prominent example of volunteered geographic information. Rather than the map itself, the data generated by the OpenStreetMap project is considered its primary output. The data is then available for use in both traditional applications, like its usage by Craigslist, OSM And, Geocaching, MapQuest Open, JMP statistical software, and Foursquare to replace Google Maps, and more unusual roles like replacing the default data included with GPS receivers. OpenStreetMap data has been favorably compared with proprietary data sources, though data quality in 2009 varied worldwide.

STEPS TO SIMULATE TRAFFIC OF A PARTICULAR REGION

Step 1: Download OSM data from Open Street Maps (<file_name>.osm)

Step 2: Run this command to get the .net file required for simulation. *(Netconvert imports digital road networks from different sources and generates road networks that can be used by other tools from this package.)*

```
netconvert --osm-files <file_name>.osm -o <file_name>.net.xml --output.street-names true --output.original-names true
```

Step 3: From the following link copy the additional polygons structures

http://sumo.dlr.de/wiki/Networks/Import/OpenStreetMap

Step 4: Save the data into a file and name it as 'typemap.xml' then run the following command. *(Polyconvert imports geometrical shapes (polygons or points of interest) from different sources, converts them to a representation that may be visualized using SUMO-GUI)*

```
polyconvert --net-file <file_name>.net.xml --osm-files <file_name>.osm --type-file typemap.xml -o <file_name>.poly.xml
```

```
python /Applications/sumo-0.23.0/tools/trip/randomTrips.py –n <file_name>.net.xml -e 100 -l
```

```
python /Applications/sumo-0.23.0/tools/trip/randomTrips.py -n <file_name>.net.xml -r <file_name>.rou.xml -e 100 -l
```

After the above steps, the routes have been generated, in the xml file you need to configure the file for the SUMO gui.

Step 5: Search for the sumo.cfg file in the sumo folder and copy it to your working folder.

The configuration file should be modified with the following contents:

```
<input>
   <net-file value="<file_name>.net.xml"/>
   <route-files value="<file_name>.rou.xml"/>
   <additional-files value="<file_name>.poly.xml"/>
</input>
<time>
   <begin value="0"/>
   <end value="1000"/>
   <step-length value="0.1"/>
</time>
```

Step 6: To run the simulator

```
sumo-gui <file_name>.sumo.cfg
```

- Helps one understand how each tag that we use impacts the traffic.
- Helps in, in-depth analysis of the road network that's established by the OSM data.
- It by default simulates right-hand driving traffic conventions and has no options for left-hand driving conventions.

C. OMNET++

<u>ABOUT</u>

OMNeT++ is an extensible, modular, component-based C++ simulation library and framework, primarily for building network simulators. "Network" is meant in a broader sense that includes wired and wireless communication networks, on-chip networks, queueing networks, and so on. Domain-specific functionality such as support for sensor networks, wireless ad-hoc networks, Internet protocols, performance modeling, photonic networks, etc., is provided by model frameworks, developed as independent projects. OMNeT++ offers an Eclipse-based IDE, a graphical runtime environment, and a host of other tools. There are extensions for real-time simulation, network emulation, database integration, SystemC integration, and several other functions. Although OMNeT++ is not a network simulator itself, it has gained widespread popularity as a network simulation platform in the scientific community as well as in industrial settings and building up a large user community. OMNeT++ provides a component architecture for models. Components (modules) are programmed in C++, then assembled into larger components and models using a high-level language (NED). Reusability of models comes for free. OMNeT++ has extensive GUI support, and due to its modular architecture, the simulation kernel (and models) can be embedded easily into your applications.

<u>INSTALLATION</u>

Supported Linux Distributions

- Ubuntu 16.04 LTS
- Fedora Core 25
- Red Hat Enterprise Linux Desktop Workstation 7.x
- OpenSUSE 42

Installing the Prerequisite Packages

OMNeT++ requires several packages to be installed on the computer. These packages include the C++ compiler (gcc or clang), the Java runtime, and several other libraries and programs. These packages can be installed from the software repositories of your Linux distribution. Generally, you will need superuser permissions to install packages. Not all packages are available from software repositories; some (optional) ones need to be downloaded separately from their web sites and installed manually.

Downloading and Unpacking

Download OMNeT++ from http://omnetpp.org. Make sure you select to download the generic archive, omnetpp-5.3-src.tgz.
Copy the archive to the directory where you want to install it. This is usually your home directory, /home/<you>. Open a terminal, and extract the archive using the following command:

```
$ tar xvfz omnetpp-5.3-src.tgz
```

This will create an omnetpp-5.3 subdirectory with the OMNeT++ files in it.

Enviroment Variables

OMNeT++ needs its bin/ directory to be in the path. To add bin/ to PATH temporarily (in the current shell only), change into the OMNeT++ directory and source the setenv script:

```
$ cd omnetpp-5.3
$ . setenv
```

The script also adds the lib/ subdirectory to LD_LIBRARY_PATH, which may be necessary on systems that don't support the rpath mechanism.
To set the environment variables permanently, edit .bashrc in your home directory. Use your favourite text editor to edit .bashrc, for example gedit:

```
$ gedit ~/.bashrc
```

Add the following line at the end of the file, then save it:

```
export PATH=$HOME/omnetpp-5.3/bin:$PATH
```

You need to close and re-open the terminal for the changes to take effect.
Alternatively, you can put the above line into ~/.bash_profile, but then you need to log out and log in again for the changes to take effect.

Configuring and Building OMNeT++

In the top-level OMNeT++ directory, type:

```
$ ./configure
```

The configure script detects installed software and configuration of your system. It writes the results into the Makefile.inc file, which will be read by the makefiles during the build process.
Normally, the configure script needs to be running under the graphical environment (X11) in order to test for wish, the Tcl/Tk shell. If you are logged in via an *ssh* session, or there is some other reason why X is not running, the easiest way to work around the problem is to tell OMNeT++ to build without Tcl/Tk and Qtenv. To do that, use the command

$ ./configure WITH_TKENV=no WITH_QTENV=no instead of plain
./configure.
When ./configure has finished, you can compile OMNeT++. Type in the terminal: $ make

Verifying the Installation

You can now verify that the sample simulations run correctly. For example, the dyna simulation is started by entering the following commands:

```
$ cd samples/dyna
$ ./dyna
```

By default, the samples will run using the Tcl/Tk environment. You should see nice gui windows and dialogs.

You can launch the OMNeT++ Simulation IDE by typing the following command in the terminal:

```
$ omnetpp
```

D. MININET

ABOUT

Mininet is a tool used to simulate the Software Defined Networks, allowing a simple and quick approach to create, interact and customize prototypes for Software Defined Networks. Mininet allows network topologies to be specified parametrically. It also allows configuration of a range of performance parameters for every virtual link. This is necessary for simulating real world systems and a requirement to implement most attack scenarios simulated in this thesis. Mininet networks run real code including standard Unix/Linux network applications as well as the real Linux kernel and network stack (including any kernel extensions which you may have available, as long as they are compatible with network namespaces.)
Because of this, the code you develop and test on Mininet, for an OpenFlow controller, modified switch, or host, can move to a real system with minimal changes, for real-world testing, performance evaluation, and deployment. Importantly this means that a design that works in Mininet can usually move directly to hardware switches for line-rate packet forwarding.

INSTALLATION (LINUX):

Install git. Git is the software version control system used by the Mininet project.

```
$ sudo apt-get install git
```

Use git to download the Mininet 2.2.0 source code.

```
$ git clone git://github.com/mininet/mininet
```

The Mininet project provides an install script. Run the script. This will install Mininet.

```
$ ~/mininet/util/install.sh -a
```

After the script stops running, test that the installation was successful. Execute the following command to test the installation. It should run a short Mininet scenario successfully.

```
$ sudo mn --test pingall
```

E. FLOODLIGHT

ABOUT

Floodlight Controller is an SDN Controller offered by Big Switch Networks that works with the OpenFlow protocol to orchestrate traffic flows in a software-defined networking (SDN) environment. OpenFlow is one of the first and most widely used SDN standards; it defines the open communications protocol in an SDN environment that allows the SDN Controller (brains of the network) to speak to the forwarding plane (switches, routers, etc.) to make changes to the network. The Floodlight Controller can be advantageous for developers, because it offers them the ability to easily adapt software and develop applications and is written in Java. Included are representational state transfer application program interfaces (REST APIs) that make it easier to program interface with the product, and the Floodlight website offers coding examples that aid developers in building the product.

INSTALLATION (LINUX):

PRE-REQUISITES:

- Your favorite flavor of Linux
- Java development kit
- JDK 8 for Floodlight master and above
- JDK 7 for Floodlight v1.2 and below
- Ant or Maven to build
- Python development package
- Eclipse IDE (Eclipse Luna Preferred)

To download dependencies for Floodlight v1.2 and below:

```
$ sudo apt-get install build-essential openjdk-7-jdk ant maven python-dev eclipse
```

The "git clone" step below uses the master version of Floodlight.

```
$ git clone git://github.com/floodlight/floodlight.git
$ cd floodlight
$ git submodule init
$ git submodule update
$ ant
$ sudo mkdir /var/lib/floodlight
$ sudo chmod 777 /var/lib/floodlight
```

Assuming java is in your path, you can directly run the floodlight.jar file produced by ant from within the floodlight directory:

```
$ java -jar target/floodlight.jar
```

Floodlight will start running and print log and debug output to your console. If you would like to save your log, you can redirect it to a file.

F. NS2 (NETWORK SIMULATOR)

ABOUT

Network simulators are tools used to simulate discrete events in a network and which helps to predict the behaviors of a computer network. Generally, the simulated networks have entities like links, switches, hubs, applications, etc. Once the simulation model is complete, it is executed to analyze the performance. Administrators can then customize the simulator to suit their needs. Network simulators typically come with support for the most popular protocols and networks in use today, such as WLAN, UDP, TCP, IP, WAN, etc. Simulating the network involves configuring the state elements like links, switches, hubs, terminals, etc. and also the events like packet drop rate, delivery status and so on. The most important output of the simulations are the trace files. Trace files log every packet, every event that occurred in the simulation and are used for analysis. Network simulators can also provide other tools to facilitate visual analysis of trends and potential trouble spots. Most of the simulation is performed in discrete time intervals where events that are in the queue are processed one after the other in an order.

Supporting downloads required:

NAM:

Nam or Network animator is an animator tool for graphical representation of network traces and real-world packet traces. NS and nam can be used together to create a simulated network and analyze it manually or graphically.

XGRAPH:

The xgraph program draws a graph on an X display given data read from either data files or from standard input if no files are specified. xgraph in ns2 is used to plot the network parameter characteristics like throughput, delay, jitter, latency etc.

INSTALLATION (LINUX):

Download NS2 tar file and run the following command:

```
$ tar -xvzf ns-allinone-2.35.tar.gz
```

All the files will be extracted into a folder called "ns-allinone-2.35".

Type following commands on terminal:

```
sudo apt-get update
sudo apt-get dist-upgrade
sudo apt-get update
sudo apt-get gcc
sudo apt-get install build-essential autoconf automake
sudo apt-get install tcl8.5-dev tk8.5-dev
sudo apt-get install perl xgraph libxt-dev libx11-dev libxmu-dev
$ cd ns-allinone-2.35
$ ./install
```

Open bashrc file to Set the Environment Variables.

```
$ sudo gedit ~/.bashrc
```

Lines to be added in bashrc:

```
# LD_LIBRARY_PATH
OTCL_LIB=/home/akshay/ns-allinone-2.35/otcl-1.14
NS2_LIB=/home/akshay/ns-allinone-2.35/lib
X11_LIB=/usr/X11R6/lib
USR_LOCAL_LIB=/usr/local/lib
export
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$OTCL_LIB:$NS2_LIB:$X11_LIB:$USR_LOCAL_LIB
# TCL_LIBRARY
TCL_LIB=/home/akshay/ns-allinone-2.35/tcl8.5.10/library
USR_LIB=/usr/lib
export TCL_LIBRARY=$TCL_LIB:$USR_LIB
# PATH
XGRAPH=/home/akshay/ns-allinone-2.35/bin:/home/akshay/ns-allinone-
2.35/tcl8.5.10/unix:/home/akshay/ns-allinone-2.35/tk8.5.10/unix
#the above two lines beginning from xgraph and ending with unix should come on the same line
NS=/home/akshay/ns-allinone-2.35/ns-2.35/
NAM=/home/akshay/ns-allinone-2.35/nam-1.15/
PATH=$PATH:$XGRAPH:$NS:$NAM
```

RUNNING NS2:

Once the changes have been made, save the file and restart the system. Once the system has restarted, open a terminal and start ns2 by using the following command:

```
$ ns
```

If the percentage symbol is shown, it means ns2 is successfully downloaded.

2. SOFTWARE DEFINED NETWORKING (SDN)

Since networks have been increasing a lot in size and in requirements, moving around hardware switches has become a burden. Even manually setting up individual software switches has become a complicated and error-prone task for companies running strongly virtualized environments along with large networks. This is where Software Defined Networking—SDN for short—enters the game, in the early 2010s. “SDN” is a bit like “cloud computing”: it is a trendy topic in computing, and it is often sold as a “miracle” solution to all network infrastructure problems. But beside marketing, it also corresponds to a new type of network architecture. “Software defined” does not mean that it only uses virtual switches instead of dedicated hardware (even if it mostly does): it refers to the fact that switches can be programmed. Their behavior is defined by a software configuration. Indeed, the main feature of SDN is that the switch control plane is decoupled from the data plane. What does this mean?

- The **control plane** is where the administration of the network takes place: it corresponds to the setting up of the packet processing rules, and from there to the establishment of the whole network switching policy.
- **Data plane** encompasses the application of those rules defined on control planes: this is the actual packet processing. When some packets require some particular, more complex processing, they can be handled to the control plane, where the decision regarding this packet will occur.

As a generic principle, the data plane must be very fast to handle a very high number of packets with simple rules so as to obtain a good bit rate in the network. By contrast, the control plane is slower. For legacy switches, the whole control and data planes would be tightly linked into a same hardware box. They would not be clearly delimited, and the switch would usually present a single setup interface. But virtual switches offer new possibilities: they make it doable to centralize the management of the switches and to remotely program them. The control plane is organized in such a way that switches take orders from a centralized controller in the network. This controller dynamically installs packet processing rules onto the switches and receives and (slowly) handles exception packets when needed. If the controller is to be represented at the center of a diagram, it is attached to network-requiring applications—on top—by so-called northbound APIs, while it is linked to the data plane—below—by what are called southbound APIs, used both to set up the switches configuration and to handle the exceptions packets.

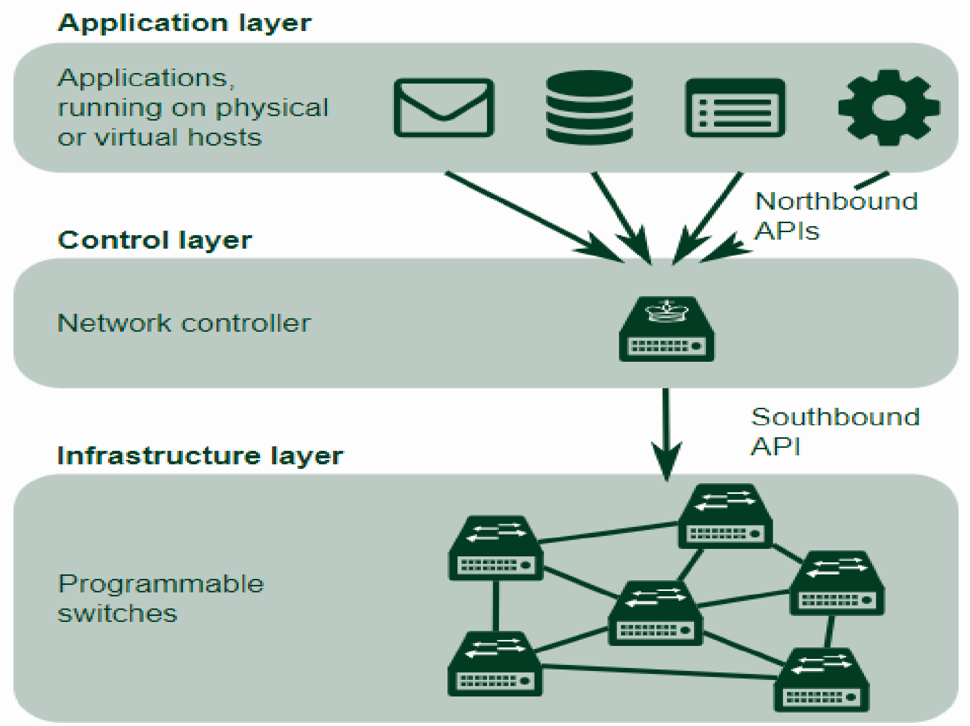


FIG 2: SDN ARCHITECTURE

In fact, the distinction between control and data plane should be considered as an “implementation choice” of SDN, and not as a definition of SDN by itself. And as technology evolves, so does the architecture. Now the switching equipment’s often include separated components, thus creating an architecture with three levels:

- “Control-management” plane is where the administration takes place, this is the controller configuring the remote switches.
- “Control-plane APIs” of a programmable switch can be used to form a component that both receives setup instructions from the controller and is able to somewhat perform simple updates of the data plane without always referring to the controller.

- Data plane, as before, handles fast packet processing.

Some other architectures have also been designed; for example, the Neutron component of the OpenStack suite embeds both the controller and the data plane. They remain logically decoupled, but they run on the same host.

3. FOG COMPUTING

Despite the increasing usage of cloud computing, there are still issues unsolved due to the inherent problem of cloud computing such as unreliable latency, lack of mobility support and location-awareness. Fog computing, also termed edge computing, can address those problems by providing elastic resources and services to end users at the edge of network, while cloud computing is more about providing resources distributed in the core network. Fog computing or fog networking, also known as fogging, is an architecture that uses edge devices to carry out a substantial amount of computation, storage, communication locally and routed over the internet backbone, and most definitively has input and output from the physical world known as transduction. Fog computing consists of Edge nodes directly performing physical input and output often to achieve sensor input, display output, or full closed loop process control, and may also use smaller Edge.

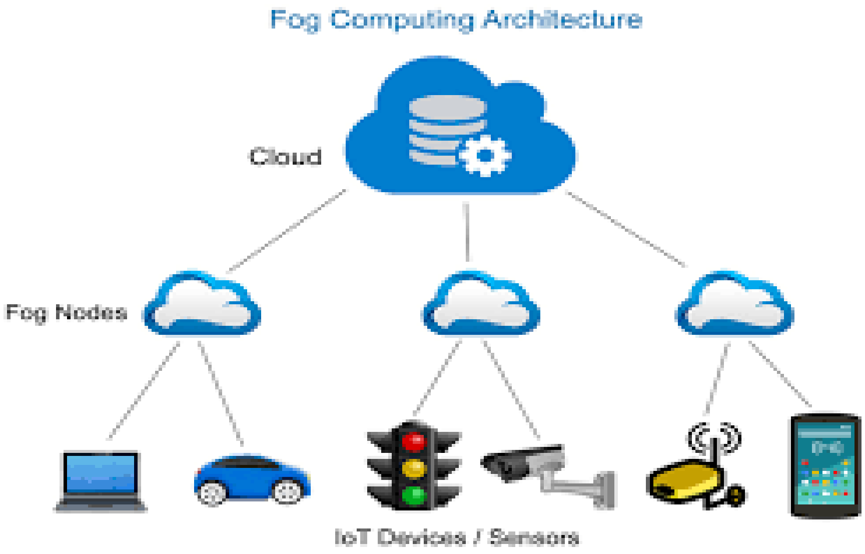


FIG 3: FOG COMPUTING ARCHITECTURE

Clouds often called as Cloudlets at the Edge or nearer to the Edge than centralized Clouds residing in very large data centers. The processing power in advanced Edge Clouds like those that control autonomous vehicles can be considerable compared to more traditional Edge personal devices such as mobile phones and personal computers. Fog computing is considered as an extension of the cloud computing paradigm from the core of network to the edge of the network. It is a highly virtualized platform that provides computation, storage, and networking services between end devices and traditional cloud servers. While in the flavor of work, fog computing is defined as "*a scenario where a huge number of heterogeneous (wireless and sometimes autonomous) ubiquitous and decentralized devices communicate and potentially* *cooperate among them and with the network to perform storage and processing tasks without the intervention of third parties. These tasks can be for supporting basic network functions or new services and applications that run in a sandboxed environment. Users leasing part of their devices to host these services get incentives for doing so.*" *Fog computing* is proposed to enable computing directly at the edge of the network, which can deliver new applications and services especially for the future of Internet [3]. For example, commercial edge routers are advertising processor speed, number of cores and built-in network storage. Those routers have the potential to become new servers. In fog computing, facilities or infrastructures that can provide resources for services at the edge of the network are called *fog nodes*. They can be resource-poor devices such as set-topboxes, access points, routers [52], switches, base stations, and end devices, or resource-rich machines such as Cloudlet and IOx. Fog computing can be perceived both in large cloud systems and big data structures, making reference to the growing difficulties in accessing information objectively. This results in a lack of quality of the obtained content. The effects of fog computing on cloud computing and big data systems may vary; yet, a common aspect that can be extracted is a limitation in accurate content distribution, an issue that has been tackled with the creation of metrics that attempt to improve accuracy. Fog networking consists of a control plane and a data plane. For example, on the data plane, fog computing enables computing services to reside at the edge of the network as opposed to servers in a data-center. Compared to cloud computing, fog computing emphasizes proximity to end-users and client objectives, dense geographical distribution and local resource pooling, latency reduction and backbone bandwidth savings to achieve better quality of service (QoS) and edge analytics/stream mining, resulting in superior user-experience and redundancy in case of failure while it is also able to be used in AAL scenarios. Fog networking supports the Internet of Things (IoT) concept, in which most of the devices used by humans on a daily basis will be connected to each other. Examples include phones, wearable health monitoring devices, connected vehicle and augmented reality using devices such as the Google Glass.

4. FSDN ARCHITECTURE

This section describes the proposed SDN-based architecture, role and operation for each component. Fog Computing framework, which takes advantage of SDN concept, is also discussed to show the benefits in offering services. In this part, we discuss the proposed SDN-based VANET architecture leveraging Fog Computing called FSDN VANET. The below SDN components are needed to incorporate for deploying the system:

SDN Controller: the global intelligence which controls all the network behaviors of the entire SDN-based VANET system. It also plays as Fog Orchestration and Resource Management for the Fog.

SDN Wireless Nodes: the vehicles act as the end-users as well as forwarding element, equipped with OBU and operating OpenFlow. They are Data Plane elements

SDN Road-Side-Unit: RSU running OpenFlow and controlled by the SDN Controller. It is a Fog device.

SDN Road-Side-Unit Controller (RSUC): A cluster of RSUs are connected to a RSUC through broadband connections before accessing to the SDN Controller. RSUC is OpenFlow-enabled and controlled by SDN Controller. Besides the responsibility of forwarding data, RSUCs also store local road system information and perform emergency services. RSUCs are fog devices, under the orchestration of SDN Controller.

Cellular Base Station (BS): In our proposed architecture, BS is not simply carrying voice calls and conveying data, it is more sophisticated. BS is under the control of SDN controller, running OpenFlow, capable of delivering fog services. Similar to RSUC, BS is also local intelligence; a Fog device under the control of SDN Controller. This architecture provides mobile operators opportunities to offer their clients valuable transportation services, motivates them participate in developing the proposed system.

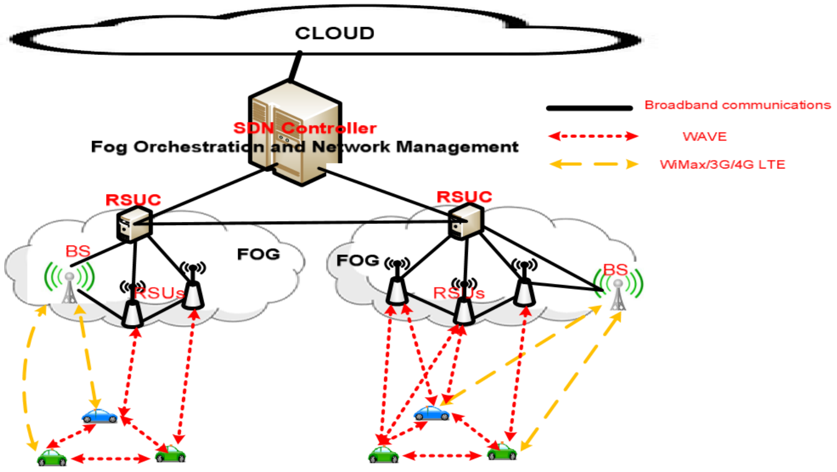


FIG 4: SDN-BASED VANET ARCHITECTURE LEVERAGING FOG COMPUTING

FSDN VANET architecture is operated in heterogeneous network environment in which forwarding infrastructure use several wireless technologies to communicate. V2V and V2I could be wireless connection or WAVE. Vehicle to BS could be long-range wireless connection such as WiMax or 3G/4G LTE. RSU-RSUC, RSUC-RSUC, RSUC-SDN Controller, and BS to SDN Controller are broadband, high-speed connections. The assumption is that each SDN wireless node is equipped with WiMax/3G/4G LTE interface mostly for Control Channel and WiFi/WAVE interface for Data Channel. The SDN wireless nodes are supposed to support fallback mechanism to turn back to conventional operations (i.e., AODV, DSDV and OLSR routing protocols) once SDN Controller connection is lost.

5. SECURITY IN VANETS

Despite the advantages, VANETs come with their own set of challenges, particularly in the aspects of security and privacy. Lack of authenticated information shared in the network may lead to malicious attacks and service abuses, which could pose great threats to drivers. In addition, unlike traditional wired networks which are protected by several lines of defense such as firewalls and gateways, security attacks on such wireless networks could come from various sources and target all nodes. Following are some of the **threats** to VANET architectures:

- ***Bogus information:*** This attack happens when information sent by the adversaries, including certificates, warnings, security messages, and identities, is not true. The adversaries may alter or even fake data, or send data captured earlier in time, to confuse other drivers. For example, a sybil attack, an attack that happens when the adversaries create a large number of pseudonymous, and acts like they are more than a hundred vehicles, may tell other vehicles that there is traffic jam ahead, and force them to take alternate routes, even though there is no traffic jam.

- ***Denial of service:*** This attack happens when adversaries send irrelevant bulk messages in order to jam the communication channel used in VANETs and consume the computational resources of the other nodes. The goal behind this kind of attack is to bring the network down, consequently rendering the VANET unavailable, which could have fatal consequences to drivers if an emergency occurred.

- ***Impersonate:*** This attack happens when the adversaries pretend to be authenticated vehicles or RSUs. The adversaries use the legitimate identities they hacked into to insert malicious information in the network, which would not only fool other vehicles but also make the innocent drivers whose identities were taken be removed from the network and denied service.

- ***Eavesdropping:*** This attack happens when an attacker is located in a vehicle, be it stopped or moving, or in a false RSU. The collection of vehicle-specific information from overheard vehicular communications is easy in a wireless network. The attackers obtain the target vehicles' confidential data, including the driver's real identities, their preferences or even their credit card codes, which seriously violates the privacy of the drivers.

- ***Message suspension:*** This attack happens when adversaries hold onto messages before sending them. An attacker selectively drops packets of messages from the network, which may hold critical information for the intended receiver, and the attacker suppresses these packets and can use them again in the future. One goal of such an attack would be to prevent registration and insurance authorities from learning about collisions involving the attacker's vehicle and/or to avoid delivering collision reports to roadside access points.

- **Hardware tampering:** This attack happens when the sensors, other on-board hardware RSUs are manipulated by adversaries. For example, an adversary can relocate a tampered RSU to launch a malicious attack, such as tampering the traffic lights to always be green when the malicious attack is approaching an intersection.

REQUIREMENTS

Given that drivers should be informed of emergencies as early as possible, since the driver who receives a warning message must have sufficient time to react, time constraints were also suggested as requirement. In the past few years, considerable research effort has been made into VANET security protocols. In summary, the primary requirements for security in VANETs are as follows:

- ***Integrated messages as well as efficiently authenticated sources:*** First, the senders of broadcast messages should be authenticated as legitimate nodes, which could efficiently prevent outsider attacks [34]. Secondly, messages collected should be consistent with the raw data from the road, which would mean that information shared in the VANETs is not maliciously fabricated and is instead unmodified and consistent with similar data generated in close space and time. Furthermore, since authentication needs to be performed before data can be collected and the service delivered, the latency of authentication should be as short as possible.

- ***Confidentiality and non-repudiation:*** Confidentiality in VANETs protects the confidential information of drivers, such as their real identity. All sensitive information should be encrypted and not available to adversaries. However, confidentiality is conditional. For those malicious adversaries, privacy could be revoked and their real identities would be broadcast to all the vehicles in VANETs. Furthermore, a sender should not be able to deny the transmission of a message. One of the applications of VANETs is tracking the responsible car in case of an accident. It may be crucial for an accident investigation to determine the correct sequence and content of messages exchanged before an accident to determine the fault and the cause.

- ***Availability and scalability:*** Communication in VANETs should be supported by alternative means when the communication channel breaks down. During traffic congestion, there may be a large number of authentication requests delivered to the authentication server. The network may then be brought down, and to ensure the ongoing communication between vehicles and vehicles to infrastructures, an alternative channel should be provided.

A. SILENCE AT LOW SPEEDS (SLOW)

Un-Traceability of vehicles is an important requirement in future vehicle communications systems. Unfortunately, heartbeat messages used by many safety applications provide a constant stream of location data, and without any protection measures, they make tracking of vehicles easy even for a passive eavesdropper. One commonly known solution is to transmit heartbeats under pseudonyms that are changed regularly in order to obfuscate the trajectory of vehicles. However, this approach is effective only if some silent period is kept during the pseudonym change and several vehicles change their pseudonyms nearly at the same time and at the same location. In order to address this problem, authors in defined a model based on the concept of the mix zone. They assumed that the adversary has some knowledge about the mix zone, and based on this knowledge, he/she tries to relate the vehicles that exit the mix zone to those that entered it earlier. They also introduced a metric to quantify the level of privacy enjoyed by the vehicles in this model. In addition, they performed extensive simulations to study the behavior of our model in realistic scenarios. In this paper, they abstracted away the frequency with which the pseudonyms are changed, and they simply assumed that this frequency is high enough so that every vehicle surely changes pseudonym while in the mix zone. In paper, the authors have proposed an effective pseudonym changing at social spots (PCS) strategy for location privacy in VANETs. In particular, they have developed two anonymity set analytical models in terms of ASS to formally analyze the achieved location privacy level, and we used game theoretic techniques to prove its feasibility. In addition, they introduced a practical KPSD model to mitigate the hazards caused by vehicle theft. Therefore, the analytical models on location privacy at social spot shed light on this research line.

SLOW TECHNIQUE

The basic idea of SLOW is that vehicles should not transmit heartbeat messages when their speed drops below a given threshold, say 35 km/h, and they should change pseudonym during each such silent period. This ensures that vehicles stopping at traffic lights or moving slowly in a traffic jam will all refrain from transmitting heartbeats and change their pseudonyms nearly at the same time and location. In the first place, all the vehicles are set to the V2V mode and broadcast their heartbeat messages (i.e. the basic safety messages) so that each vehicle is able to identify a list of neighbours.

Possibilities for future extension under this would be: Reducing heartbeat rates as the vehicle’s speed reduces, rather than eliminating them altogether or work on that threshold value of the speed.

Our approach:

Instead of eliminating the heartbeat messages during low speeds, what we propose is a technique of securing the heartbeat messages while transmission.

System Model

It is assumed that a vehicular network to be comprising of on-board units (OBUs) set up on vehicles and road-side units (RSUs) along the roads. Here, the cloud acts as the trusted authority (TA) which holds the identities of vehicles. Over the wireless channel using (DSRC) dedicated short range communication protocol the OBUs and the RSUs communicate with each other. The trusted server communicates using a secure fixed network (e.g. the internet). We consider that vehicles and the (road side units) RSUs along the roads in the vehicular network consisting of (OBUs) on-board units installed on them. We further assume the following:

1)The heartbeat message is sent to the cloud on demand of another node in the cloud network, so that the cloud will not have to bear the burden of the processing the message and sending it to everyone in encrypted form.

2)Every vehicle in the cloud and the cloud itself will have a pair of public and private keys.

3)Every vehicle when enter the cloud will get registered in the network with its public key.

4)Also, there is a proper join and leave mechanism for a vehicle to enter and leave the cloud

SCHEME 1: SLOW SCHEME

In this section we propose a scheme for providing secure message transmission V2I model. Basically, the proposed scheme has 6 steps:

*1. **Preparation:** TA prepares all system parameters and keys.*

*2.**Request sent to cloud:** The vehicle who want to know the presence of other vehicle in the that zone (registered in the same cloud) will send a request to the cloud.*

*3. **Cloud broadcast message:** The cloud will then broadcast a message to all the vehicles in the network to send their heartbeat messages.*

*4. **Vehicle’s Response:** The vehicles present in the cloud will send the encrypted message to the cloud.*

*5. **Serving the request:** The cloud will send the message again in encrypted form to the requester.*

*6.**Decryption of message**: The requester will receive the message and will decrypt and get the plain text.*

B. RSA ENCRYPTION IN VANET

RSA is one of the first practicable public-key cryptosystems and is widely used for secure data transmission. In such a cryptosystem, the encryption key is public and differs from the decryption key which is kept secret. The RSA algorithm involves three steps: **key generation, encryption and decryption**. RSA algorithm is asymmetric cryptography algorithm. RSA involves a public key and a private key. The public key can be known by everyone and is used for encrypting messages. Messages encrypted with the public key can only be decrypted in a reasonable amount of time using the private key.

ALGORITHM 1: RSA KEY GENERATION

INPUT: *Required modulus bit length, k.*
OUTPUT: *An RSA key pair ((N,e), d) where N is the modulus, the product of two primes (N=pq) not exceeding k bits in length; e is the public exponent, a number less than and coprime to (p-1)(q-1); and d is the private exponent such that ed ≡ 1 (mod (p-1)(q-1)).*

```
Select a value of e from $\{3, 5, 17, 257, 65537\}$
repeat
    $p \leftarrow genprime(k/2)$
until $(p \bmod e) \neq 1$
repeat
    $q \leftarrow genprime(k - k/2)$
until $(q \bmod e) \neq 1$
$N \leftarrow pq$
$L \leftarrow (p-1)(q-1)$
$d \leftarrow modinv(e, L)$
return $(N, e, d)$
```

The public key consists of the modulus n and the public (or encryption) exponent e. The private key consists of the modulus n and the private (or decryption) exponent d, which must be kept secret. p, q, and φ(n) must also be kept secret because they can be used to calculate d.

ENCRYPTION

When any RSU want to transmits public key (n, e) to vehicles and keeps the private key d secret. In RSU, first turns M into an integer m, such that $0 \leq m < n$ by using an agreed-upon reversible protocol known as a padding scheme. Then computes the cipher text c corresponding to

$$c \equiv m^e \pmod{n}$$

This can be done efficiently, even for 500-bit numbers, using Modular exponentiation. RSU then transmits *c* to vehicles.

DECRYPTION

Vehicles can recover *m* from *c* by using private key exponent *d* via computing

$$m \equiv c^d \pmod{n}$$

Given *m*, recover the original message *M* by reversing the padding scheme.

C. METHOD TO TACKLE BOGUS MESSAGES

Security mechanisms recommended by VANET standardization bodies are not sufficient to stop authorized vehicles from sending bogus or fake messages. Existing VANETs standards address the security and privacy issues, however, evaluating the behavior of authorized vehicles is not addressed in the standards. For example, a legitimate vehicle could send fake information about the road conditions to a central monitoring system and as a result the system could take a wrong decision. These attacks not only decrease the transportation efficiency, in worst cases, they may cause accidental events that can threaten human life. Our proposal ARS: Anonymous Reputation System for VANET focuses on a centralized reputation system privacy-preserving using pseudonyms and giving reputation level to different behavioral aspects of the vehicle. A reputation server collects feedback from the vehicles through a Road-Side Unit or vehicles with LTE/4G/5G connection and updates the vehicles reputation level by matching the pseudoidentities with its real identity.

ARS SCHEME OPERATIONS

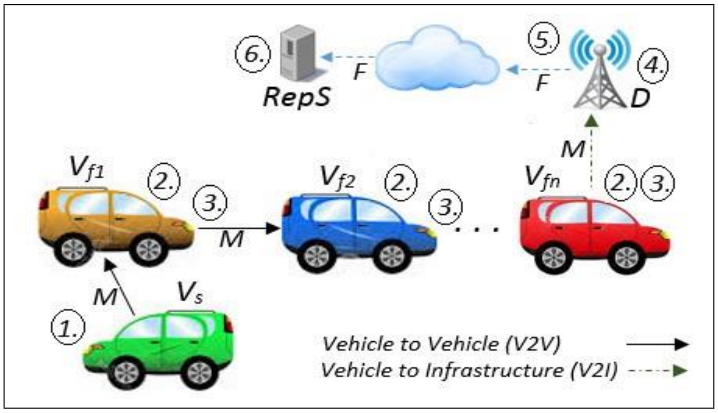


FIG 5: ARS SCHEME

The vehicles participate in the generation and forwarding of messages. The destination of a message generates a feedback that sends to the RepS containing a rating that evaluates the confirmation or not of the event announced in the message. The feedback also contains a list with the pseudo-identities of the vehicles that forwarding the message. RepS uses the list for updating the *RL* of the vehicles involved. Thus, the operation of ARS consists of the following steps, as shown in the figure.

SCHEME 2: METHOD TO TACKLE BOGUS MESSAGE

1) Sending of application messages.
2) Evaluation of the sender V_s
3) Evaluation of the intermediate V_{fi}
4) Evaluation of the message in D
5) Feedback reporting:
6) Updating of the reputation level RL:

SECURITY IN SDN

A. INTRODUCTION

The emergence of the software defined networking (SDN) paradigm has created great potential and hope to overcome the need for flexible, secure, reliable, and well managed next generation networks. Besides, the architecture of SDNs poses new external and internal threats and vulnerabilities. Predominantly, the integrity and security of SDNs remain unproven when it comes to the placement of management functionality in a single centralized virtual server. Subsequently, compromising the whole network through a single point of failure is much easier. Moreover, it becomes the primary potential attack target. The programmability aspect of SDNs also makes them more vulnerable to a number of malicious code exploits and attacks. Furthermore, the abstraction of different available flows and underlying hardware resources at the SDN controller significantly supports harvesting intelligence from the existing resources. Afterward, it can be effortlessly used for further attacks, exploitations, and particularly reprogramming the entire network. Likewise, the southbound interface of an SDN can also easily be targeted with diverse denial of service and side channel attacks. Equally important, configuration errors of SDNs can have more serious consequences than in traditional networks. Besides, SDN agents can also potentially be targeted for injecting false flows. Keeping in view the SDN features and architecture, cyber-attacks launched through SDNs can have even more devastating and larger effects than using simple networks. Since security is not considered initially as part of SDN design, each layer of an SDN has its own security implications and requirements. Moreover, establishing trust throughout an SDN is even more critical. Likewise, the network essentially needs a dynamic forensic remediation and robust policy frameworks ensuring the right direction of the controller. Although security should be built in as part of SDN architecture, it must also be delivered as a service to ensure the privacy and integrity of all the connected resources. Some researchers claim that we are still far away from secure and dependable SDN architecture. On the contrary, it is also complementary to say that SDN can be better used to enhance and implement security; meanwhile, security of the SDN itself becomes a priority. SDN certainly necessitates a simple, cost-effective, scalable, and efficient secure environment.

B. ARP SPOOFING IN SDN

ARP spoofing is a type of attack in which a malicious actor sends falsified ARP (Address Resolution Protocol) messages over a local area network. This results in the linking of an attacker's MAC address with the IP address of a legitimate computer or server on the network. Once the attacker's MAC address is connected to an authentic IP address, the attacker will begin receiving any data that is intended for that IP address. ARP spoofing can enable malicious parties to intercept, modify or even stop data in-transit. ARP spoofing attacks can only occur on local area networks that utilize the Address Resolution Protocol. ARP spoofing attack comes in different forms namely request and response attacks. In request attack, an attacker broadcasts ARP request message with forged source IP-MAC in the ARP header. When victim receives this spoofed ARP message, it updates its ARP cache table with the attacker's forged IP-MAC pair. When the victim sends its subsequent packets destined to the forged IP, the packets will be destined to the MAC of the host specified by the attacker in the forged IP-MAC pair. This way attacker can intercept or deny the traffic sent to a user in the network. The other form of attack is the response attack, in which the attacker will either respond to normal ARP request with forged ARP replies that maps the next hop IP to the attacker MAC address or send spoofed ARP replies without having requests being issued. ARP spoofing may be used to launch either one of the following attacks:

1. ***DoS attacks****:* cyber-attack in which the perpetrator seeks to make a machine or network resource unavailable to its intended users by temporarily or indefinitely disrupting services of a host connected to the Internet. Denial of service is typically accomplished by flooding the targeted machine or resource with superfluous requests in an attempt to overload systems and prevent some or all legitimate requests from being fulfilled.

2. ***Host impersonation attack****:* the attacker will receive packets intended to the victim and can reply to these packets on behalf of the victim.

3. ***Man-In-The-Middle (MITM) attack****:* cyberattack where a malicious actor inserts him/herself into a conversation between two parties, impersonates both parties and gains access to information that the two parties were trying to send to each other. A man-in-the-middle attack allows a malicious actor to intercept, send and receive data meant for someone else, or not meant to be sent at all, without either outside party knowing until it is too late.

They can also be used to execute other data link layer attacks such as MAC flooding attacks, STP attacks, DHCP attacks, and VLAN attacks.

C. DDOS ATTACK ON SDN

One of the possibilities that can cause the controller of SDN to be unreachable is a Distributed Denial of Service (DDoS) attack. In DDoS attacks, a large number of packets are sent to a host or a group of hosts in a network. If the source addresses of the incoming packets are spoofed, which is usually the case, the switch will not find a match and has to forward the packet to the controller. The collection of legitimate and DDoS spoofed packets can bind the resources of the controller into continuous processing up to the point where they are completely exhausted. This will make the controller unreachable for the newly arrived legitimate packets and may bring the controller down causing the loss of the SDN architecture. Even if there is a backup controller, it has to face the same challenge. Since the controller software can be run on a laptop or a powerful server, the term "early" depends on the tolerance of the device and traffic properties. However, if the detection happens in the first few hundred packets, the mitigation can be applied before the controller is completely swamped with the large number of malicious packets. To accomplish this goal, a fast and effective method is needed that works within the controller. At the same time, it must be lightweight to avoid excessive processing power usage, specially, at the peak of an attack.

DDOS ATTACK IMPLEMENTATION IN FLOODLIGHT-VM

Following are the steps to implant a DDOS Attack in floodlight, in SDN Architecture topology.

```
1.   java -jar target/floodlight.jar
2.   sudo mn --controller=remote,ip=127.0.0.1,port=6653 --topo=single,3
3.   sudo ./start.sh
4.   sudo ovs-vsctl -- --id=@sflow create sflow agent=eth0 target=\"127.0.0.1:6343\" sampling=10
    polling=20 -- -- set bridge s1 sflow=@sflow
5.   xterm h1 xterm h2
6.   ping -f 10.0.0.1
```

D. HYBRID SECURITY ENCRYPTION SCHEME FOR COMMUNICATIONS IN VANET

Even though, VANET plays a very crucial role in the ITS applications by offering various attractive features, without proper security assurance communicating the information over VANET can be risky. This is because, vehicular communications in VANET deal with not only information about traffic and road conditions but also involved user specific information such as their route, trip path and time critical information that is intended only for the control centers. Such type of user specific information and time critical information must be kept private, making security as one of crucial requirements for VANET applications. Thus, to make communications in VANET more secure, we propose a Hybrid security framework based on hybrid cryptography concept, which is a combination of public key cryptographic method and private key cryptographic method. Hybrid cryptographic method is developed by combining RSA and AES (Advanced Encryption Standard) cryptographic algorithms.

REASON FOR HYBRID CRYPTOGRAPHIC METHOD

Symmetric key cryptography is one possible security method for VANET. The symmetric cryptography is private key cryptography uses the same key for encrypting the data to cipher text and decrypting it back to plain text. Asymmetric key cryptography or public key cryptography is another possible security method that can be applied for VANET. This method uses two keys; private key and public key. Here private key is used for data encryption and public key is used for data decryption. Even though using two keys in asymmetric key cryptography enhances the security level, it faces the problem of consuming high extent of network resources in terms of memory and computations. In case of symmetric key cryptography, the network resources required are comparatively less but offer less security when compared to asymmetric key cryptography due to the use of same key for encryption and decryption. Therefore, to tackle these limitations, Hybrid cryptographic method is introduced. The security framework that is proposed for the VANET is aimed at providing higher level of security by consuming less number of resources. To satisfy this aim, the combination of RSA and AES algorithms is chosen for the security framework of hybrid cryptography.

PROPOSED METHOD

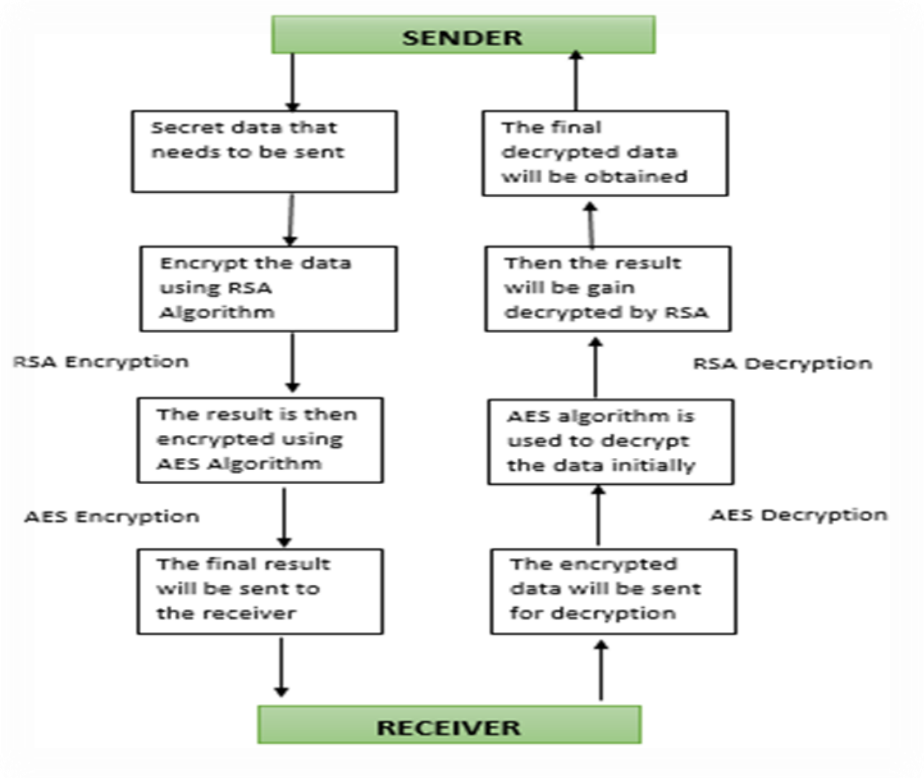

SENDER
Secret data that needs to be sent
Encrypt the data using RSA Algorithm
RSA Encryption
The result is then encrypted using AES Algorithm
AES Encryption
The final result will be sent to the receiver
RECEIVER
The encrypted data will be sent for decryption
AES Decryption
AES algorithm is used to decrypt the data initially
RSA Decryption
Then the result will be gain decrypted by RSA
The final decrypted data will be obtained

FIG 6: HYBRID CRYPTOGRAPHY IN VANET

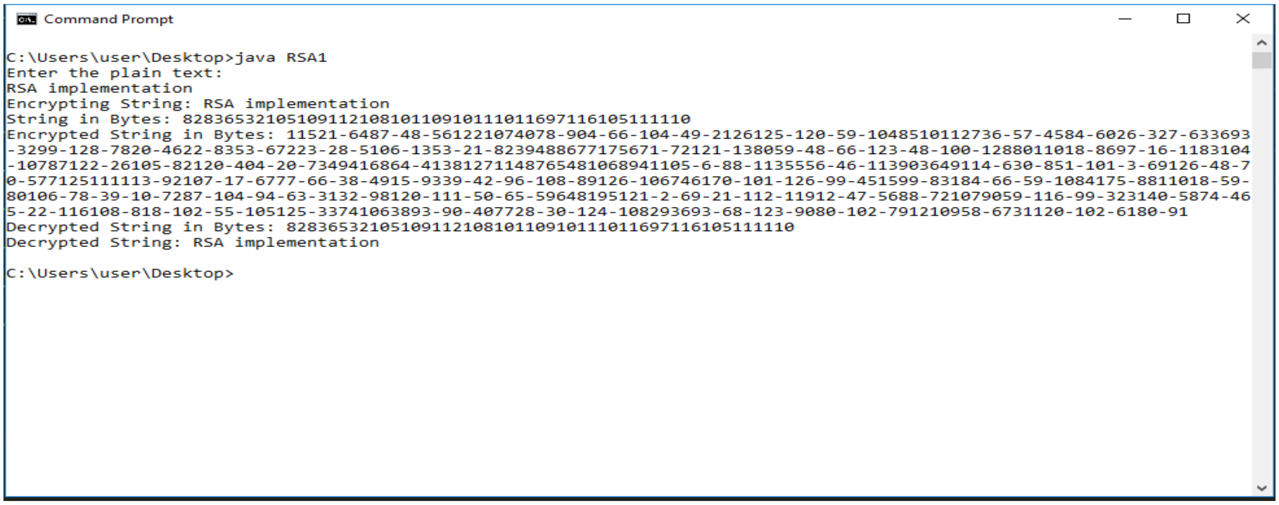
As shown in the figure, using this hybrid encryption method the private data will be initially encrypted using the RSA algorithm. The cipher text that is obtained from the RSA algorithm will be given as the input to the AES algorithms and then the encryption will take place to produce the cipher text. The private data or secret data of the user will be secured by double encrypting it using hybrid cryptography. In the encryption process initially, RSA algorithm is used to encrypt the secret data the output of the RSAS algorithm is given as input to AES algorithm. The output of AES algorithm is the cipher text which is encrypted twice using hybrid cryptography. This cipher text will be transmitted over the wireless channel to the other vehicle user in V ANET. The receiver of this encrypted data can understand the message contents only by decrypting it using hybrid cryptography. The decryption process of this proposed framework is designed as the reverse process of encryption. That is, the received cipher text is decrypted first by AES algorithm and the output of AES algorithm will be given as input to RSA algorithm for further decryption. The resultant output of RSA algorithm will be the original message sent by the sender over the wireless channel of VAN ET. The hybrid cryptography offers comparatively high level of security as it is difficult for the attacker to decrypt the information that is encrypted by two cryptographic algorithms. As the attacker must choose the exactly the same algorithms RSA and AES for decryption process in order to decrypt the data, guessing the algorithms takes time and is also not practically possible. Implementation of this proposed security framework is cost effective as the algorithm chosen for hybrid cryptography requires less number of computational resources. On the other hand, the chosen algorithms RSA and AES increases security level of application. The results of simulation are shown in next section.

RESULTS OF THE WORK COMPLETED

RSA IMPLEMENTATION IN JAVA

We implemented the RSA encryption technique in java. Figure 7 represents the output of the implemented code in which a string is passed, encryption takes place then the data is decrypted.

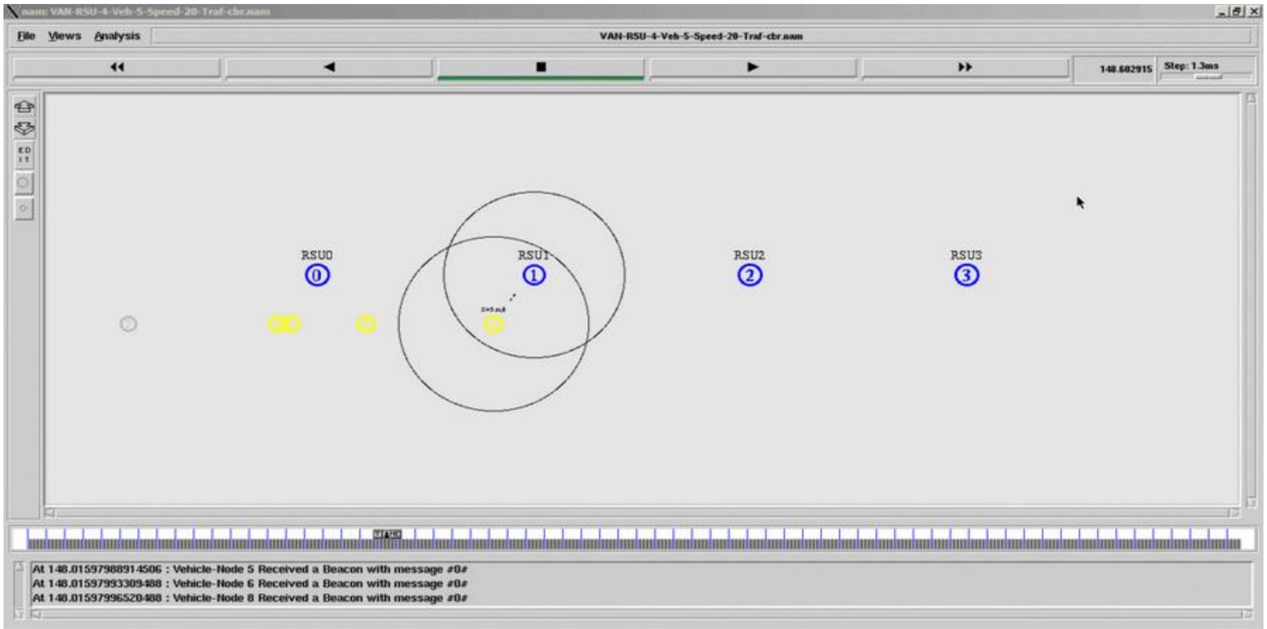
FIG 7: RESULTS OF RSA IMPLEMENTATION IN JAVA



```
C:\Users\user\Desktop>java RSA1
Enter the plain text:
RSA implementation
Encrypting String: RSA implementation
String in Bytes: 8283653210510911210810110910111011697116105111110
Encrypted String in Bytes: 11521-6487-48-561221074078-904-66-104-49-2126125-120-59-1048510112736-57-4584-6026-327-633693
-3299-128-7820-4622-8353-67223-28-5106-1353-21-8239488677175671-72121-138059-48-66-123-48-100-1288011018-8697-16-1183104
-10787122-26105-82120-404-20-7349416864-41381271148765481068941105-6-88-1135556-46-113903649114-630-851-101-3-69126-48-7
0-577125111113-92107-17-6777-66-38-4915-9339-42-96-108-89126-106746170-101-126-99-451599-83184-66-59-1084175-8811018-59-
80106-78-39-10-7287-104-94-63-3132-98120-111-50-65-59648195121-2-69-21-112-11912-47-5688-721079059-116-99-323140-5874-46
5-22-116108-818-102-55-105125-33741063893-90-407728-30-124-108293693-68-123-9080-102-791210958-6731120-102-6180-91
Decrypted String in Bytes: 8283653210510911210810110910111011697116105111110
Decrypted String: RSA implementation

C:\Users\user\Desktop>
```

FIG 8: VANET ARCHITECTURE IMPLEMENTATION IN NS2


nam: VAN-RSU-4-Veh-5-Speed-20-Traf-cbr.nam
File
Views
Analysis
VAN-RSU-4-Veh-5-Speed-20-Traf-cbr.nam
148.602915
Step: 1.3ms
EDIT
RSU0
0
RSU1
1
RSU2
2
RSU3
3

```
At 148.01597988914506 : Vehicle-Node 5 Received a Beacon with message #0#
At 148.01597993309488 : Vehicle-Node 6 Received a Beacon with message #0#
At 148.01597996520488 : Vehicle-Node 8 Received a Beacon with message #0#
```

We implemented RSA algorithm in VANET architecture to securely transmit the data. The TCL script for VANET architecture was written and successfully implemented. Figure 3 represents the output model generated after successful implemented of TCL script. In order to evaluate the performance awk files were generated and values of the different parameter was evaluated as shown in table 1.

TABLE 1: RESULTS OF DIFFERENT PARAMETER GENERATED

NUMBER OF VEHICLES	DELAY	PDR	THROUGHPUT
5	39	50	3.5
10	41	51	3.6
15	42	53	3.7
20	45	56	3.9
25	47	58	3.9

FIG 9: THROUGHPUT

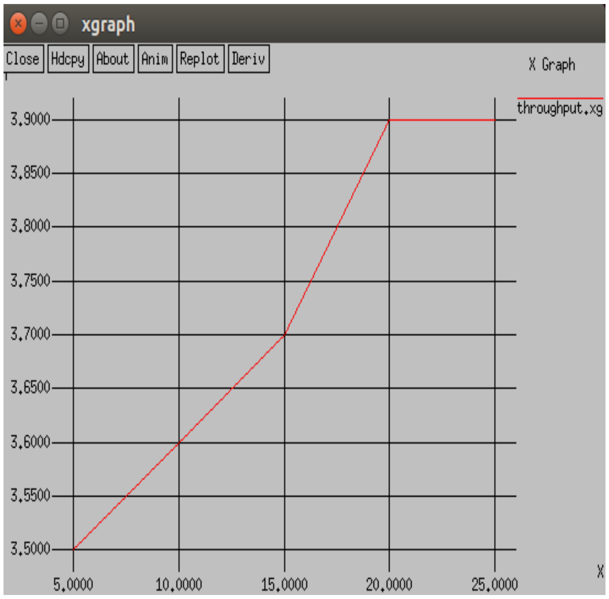
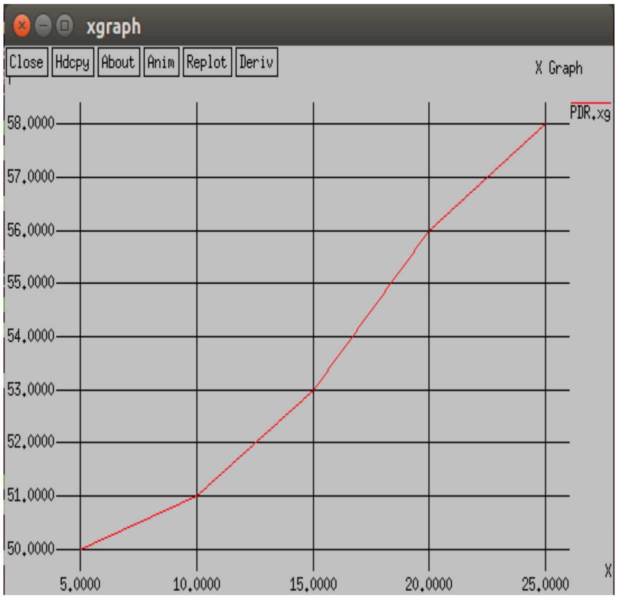
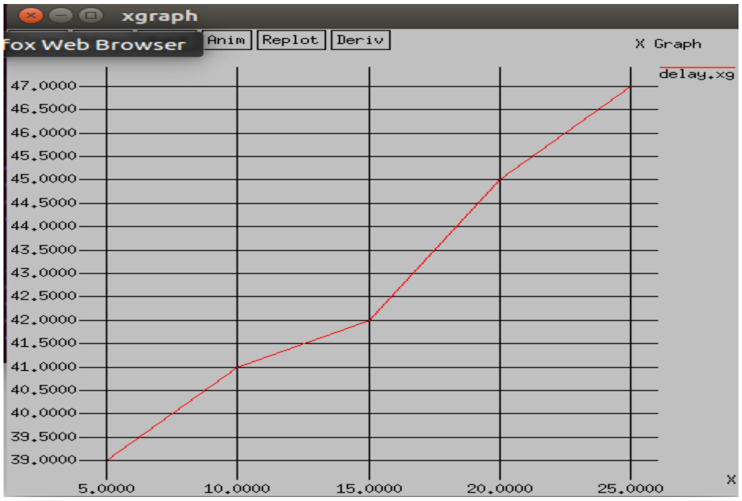


FIG 10: PDR CALCULATION



Above figures 9, 10, 11 represents the xgraph of the generated values by successfully implementation of the AWK file in order to evaluate the performance of the RSA encryption technique.

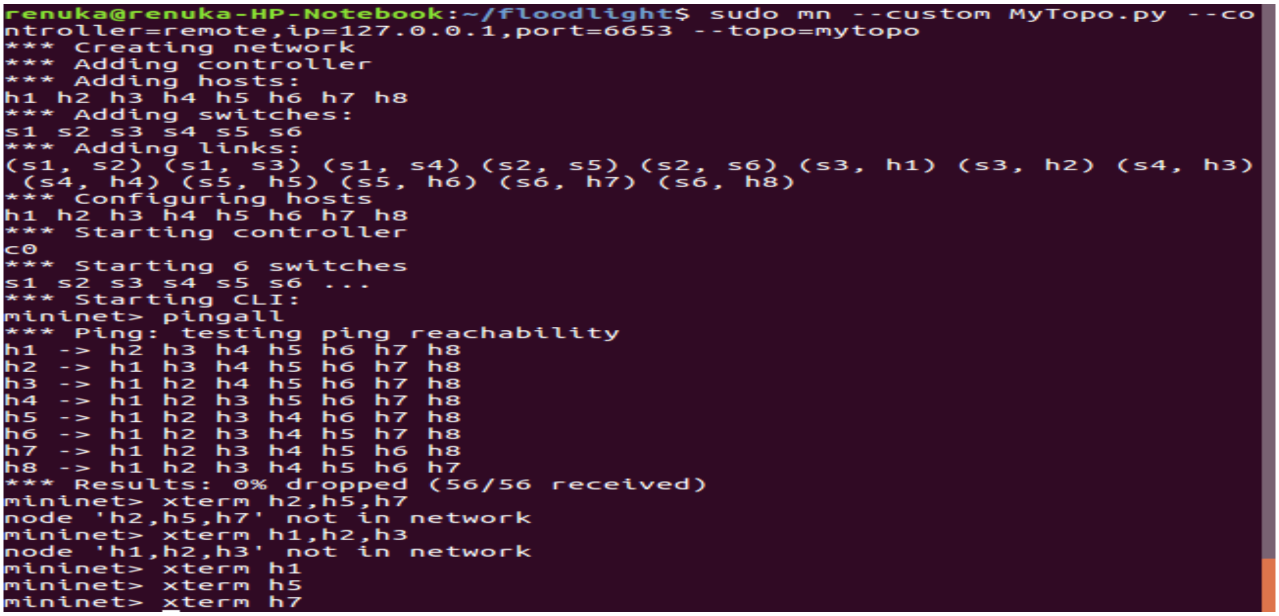
FIG 11: DELAY OUTPUT



ENERGY -EFFICIENT APPROACH IN VANET

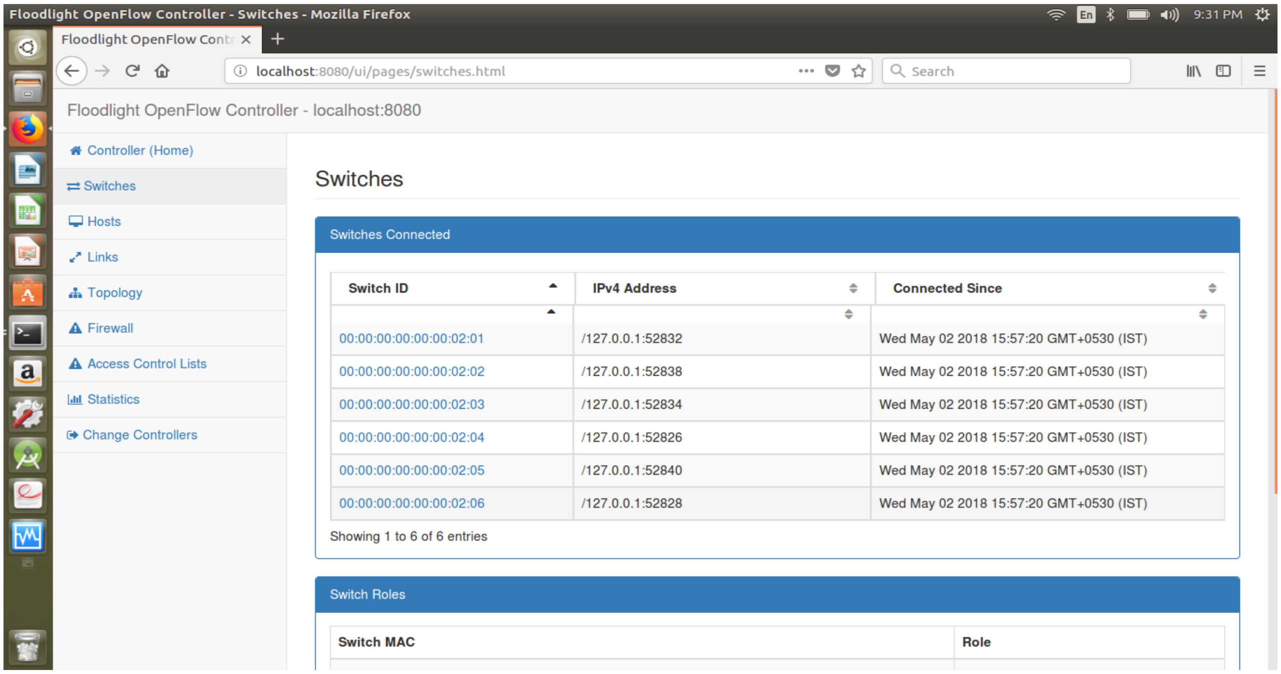
In order to simulate the energy-efficient model we used floodlight and Mininet tool. We created a custom topology by implementing the topology code in python. Figure 12 represents the custom topology generated with the configuration consisting of one controller, 6 switches and 8 hosts. The link between the nodes are also represented. Hence in order to check the reachability of each node from each other node we used pingall command and the output shows that each node is reachable from each other node as the packet dropped is equal to 0%.

FIG 12: SCREENSHOT OF TOPOLOGY GENERATION CODE IN FLOODLIGHT



```
renuka@renuka-HP-Notebook:~/floodlight$ sudo mn --custom MyTopo.py --co
ntroller=remote,ip=127.0.0.1,port=6653 --topo=mytopo
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 h3 h4 h5 h6 h7 h8
*** Adding switches:
s1 s2 s3 s4 s5 s6
*** Adding links:
(s1, s2) (s1, s3) (s1, s4) (s2, s5) (s2, s6) (s3, h1) (s3, h2) (s4, h3)
 (s4, h4) (s5, h5) (s5, h6) (s6, h7) (s6, h8)
*** Configuring hosts
h1 h2 h3 h4 h5 h6 h7 h8
*** Starting controller
c0
*** Starting 6 switches
s1 s2 s3 s4 s5 s6 ...
*** Starting CLI:
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2 h3 h4 h5 h6 h7 h8
h2 -> h1 h3 h4 h5 h6 h7 h8
h3 -> h1 h2 h4 h5 h6 h7 h8
h4 -> h1 h2 h3 h5 h6 h7 h8
h5 -> h1 h2 h3 h4 h6 h7 h8
h6 -> h1 h2 h3 h4 h5 h7 h8
h7 -> h1 h2 h3 h4 h5 h6 h8
h8 -> h1 h2 h3 h4 h5 h6 h7
*** Results: 0% dropped (56/56 received)
mininet> xterm h2,h5,h7
node 'h2,h5,h7' not in network
mininet> xterm h1,h2,h3
node 'h1,h2,h3' not in network
mininet> xterm h1
mininet> xterm h5
mininet> xterm h7
```

FIG 13: SCREENSHOT OF SWITCH CONFIGURATION USED FOR TOPOLOGY GENERATION IN FLOODLIGHT


Floodlight OpenFlow Controller - Switches - Mozilla Firefox
9:31 PM
localhost:8080/ui/pages/switches.html
Search
Floodlight OpenFlow Controller - localhost:8080
Controller (Home)
Switches
Hosts
Links
Topology
Firewall
Access Control Lists
Statistics
Change Controllers

Switches

Switches Connected

Switch ID	IPv4 Address	Connected Since
00:00:00:00:00:00:02:01	/127.0.0.1:52832	Wed May 02 2018 15:57:20 GMT+0530 (IST)
00:00:00:00:00:00:02:02	/127.0.0.1:52838	Wed May 02 2018 15:57:20 GMT+0530 (IST)
00:00:00:00:00:00:02:03	/127.0.0.1:52834	Wed May 02 2018 15:57:20 GMT+0530 (IST)
00:00:00:00:00:00:02:04	/127.0.0.1:52826	Wed May 02 2018 15:57:20 GMT+0530 (IST)
00:00:00:00:00:00:02:05	/127.0.0.1:52840	Wed May 02 2018 15:57:20 GMT+0530 (IST)
00:00:00:00:00:00:02:06	/127.0.0.1:52828	Wed May 02 2018 15:57:20 GMT+0530 (IST)

Showing 1 to 6 of 6 entries

Switch Roles

Switch MAC	Role

FIG 14: SCREENSHOT OF TOPOLOGY GENERATED BY IMPLEMENTATION OF CODE IN FLOODLIGHT

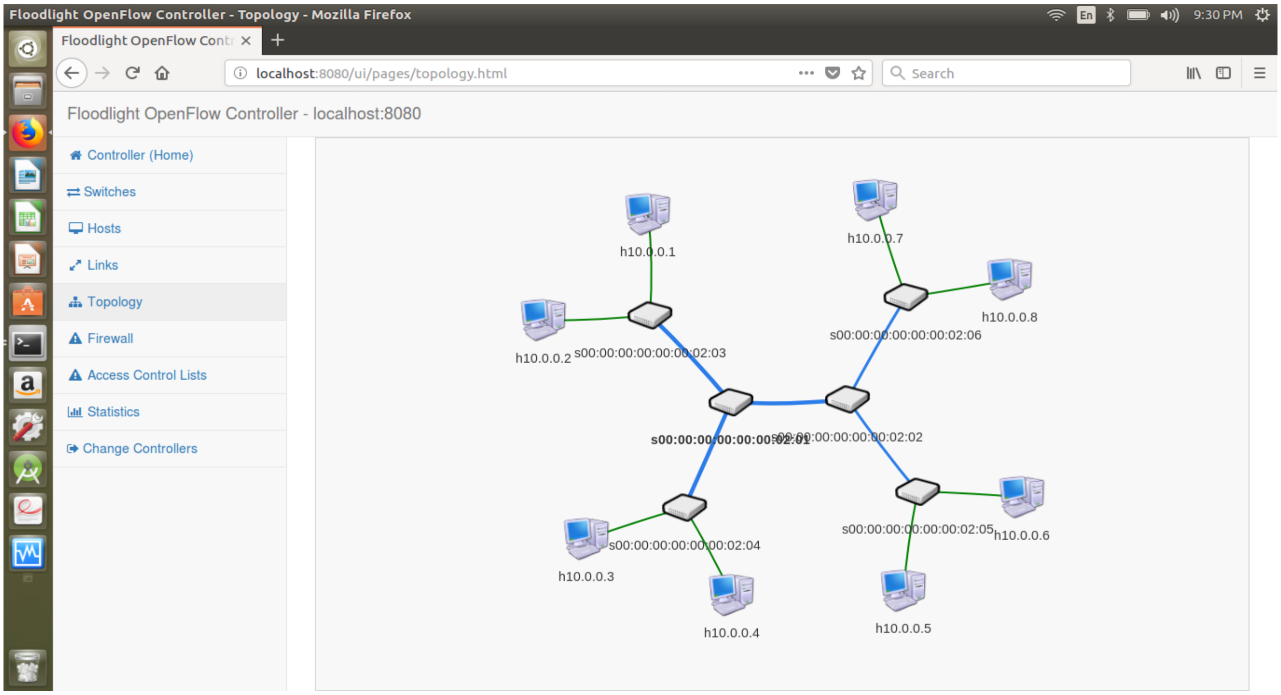


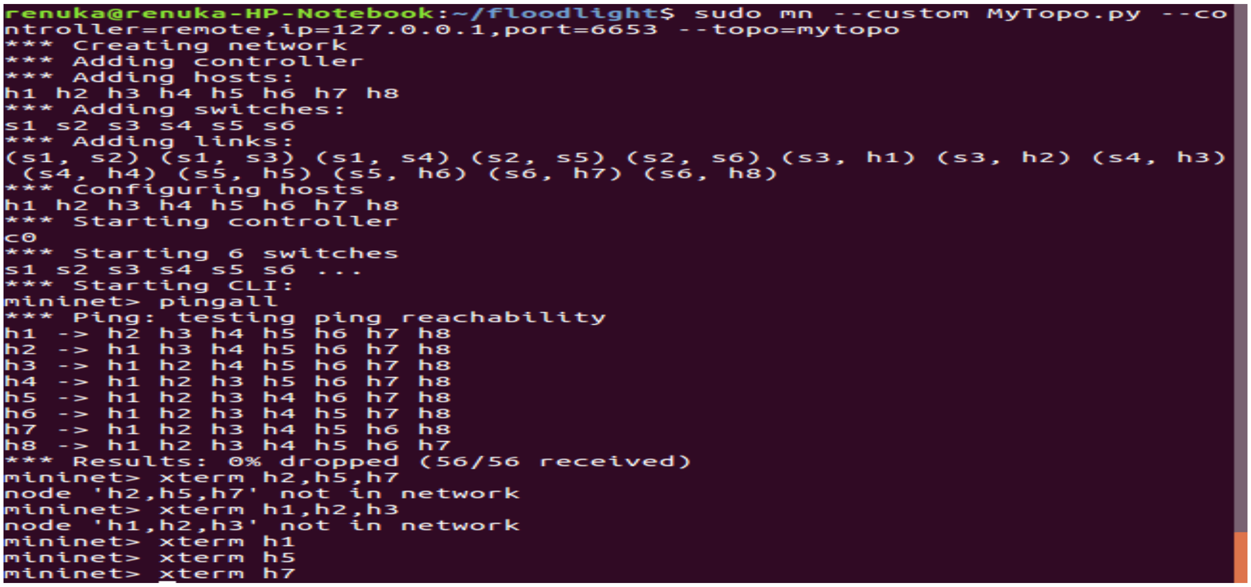
Figure 13 represents the switch configuration used to generate model and Figure 14 represents the screenshot of the custom topology by successfully implementing the python code in floodlight and Mininet.

IMPLEMENTATION OF SECURITY IN SDN

ARP SPOOFING IN FLOODLIGHT

ARP spoofing is a type of attack in which a malicious actor sends falsified ARP (Address Resolution Protocol) messages over a local area network. This results in the linking of an attacker's MAC address with the IP address of a legitimate computer or server on the network. Once the attacker's MAC address is connected to an authentic IP address, the attacker will begin receiving any data that is intended for that IP address. ARP spoofing can enable malicious parties to intercept, modify or even stop data in-transit. ARP spoofing attacks can only occur on local area networks that utilize the Address Resolution Protocol. We have implemented the ARP spoofing in floodlight in order to show the spoofing method in SDN architecture. We considered the same custom topology generated by implementing the topology code in python. Figure 14 represents the custom topology generated on which ARP spoofing is applied. Figure 15 represents the configuration of custom topology consisting of one controller, 6 switches and 8 hosts. The link between the nodes are also represented. Hence in order to check the reachability of each node from each other node we used pingall command and the output shows that each node is reachable from each other node as the packet dropped is equal to 0%. The command xterm h2, h5 and h7 is used to access the given host in order to make changes to its configuration so to successfully implement ARP spoofing.

FIG 15: PING ALL NODES IN CUSTOM TOPOLOGY

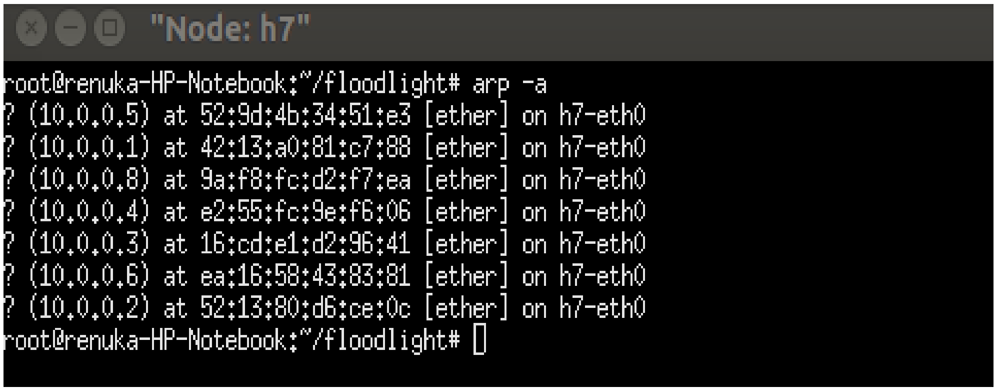


```
renuka@renuka-HP-Notebook:~/floodlight$ sudo mn --custom MyTopo.py --co
ntroller=remote,ip=127.0.0.1,port=6653 --topo=mytopo
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 h3 h4 h5 h6 h7 h8
*** Adding switches:
s1 s2 s3 s4 s5 s6
*** Adding links:
(s1, s2) (s1, s3) (s1, s4) (s2, s5) (s2, s6) (s3, h1) (s3, h2) (s4, h3)
 (s4, h4) (s5, h5) (s5, h6) (s6, h7) (s6, h8)
*** Configuring hosts
h1 h2 h3 h4 h5 h6 h7 h8
*** Starting controller
c0
*** Starting 6 switches
s1 s2 s3 s4 s5 s6 ...
*** Starting CLI:
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2 h3 h4 h5 h6 h7 h8
h2 -> h1 h3 h4 h5 h6 h7 h8
h3 -> h1 h2 h4 h5 h6 h7 h8
h4 -> h1 h2 h3 h5 h6 h7 h8
h5 -> h1 h2 h3 h4 h6 h7 h8
h6 -> h1 h2 h3 h4 h5 h7 h8
h7 -> h1 h2 h3 h4 h5 h6 h8
h8 -> h1 h2 h3 h4 h5 h6 h7
*** Results: 0% dropped (56/56 received)
mininet> xterm h2,h5,h7
node 'h2,h5,h7' not in network
mininet> xterm h1,h2,h3
node 'h1,h2,h3' not in network
mininet> xterm h1
mininet> xterm h5
mininet> xterm h7
```

The command xterm h2, h5 and h7 is used to access the given host in order to make changes to its configuration so to successfully implement ARP spoofing. Figure 16, Figure 17 and Figure 18 represents the output of the xterm command.

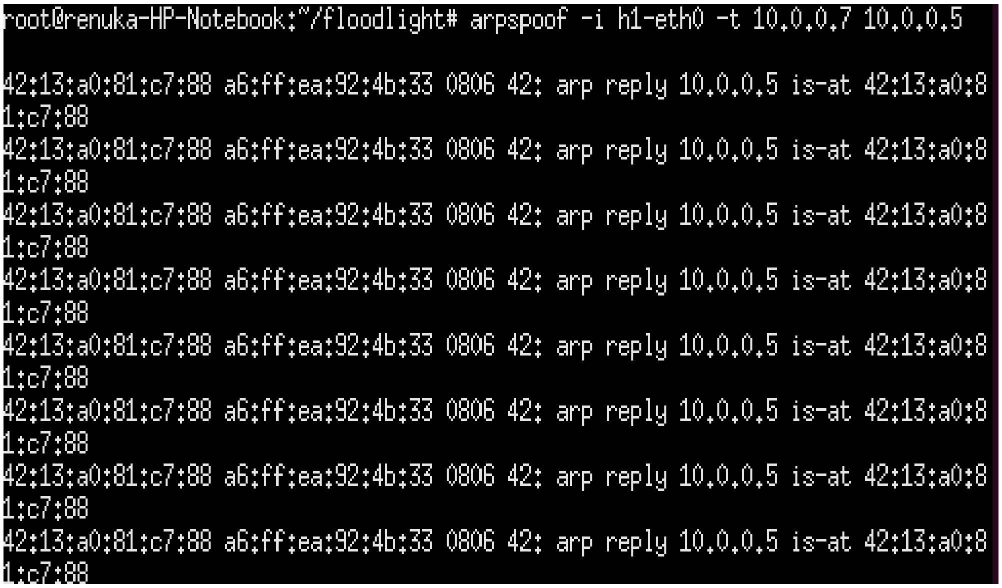
In the individual interface of host nodes h1, h5 and h7 which corresponds to the IP Address 10.0.0.1, 10.0.0.5 and 10.0.0.7, some attack is introduced. The command "arp -a" used in terminal of h7 shows all the connections of that node. We have simulated an attack via h1 between the nodes h5 and h7, and via h5 between nodes h1 and h7. The command used to do so is "arpspoof -i nodeName -t node1 node2", where "nodeName" is the attacker and the "node1" and "node2" are the victims between which the communication will get hindered by the attacker. The result can be verified by Figure 19.

FIG 16: NODE H7 INTERFACE



```
root@renuka-HP-Notebook:~/floodlight# arp -a
? (10.0.0.5) at 52:9d:4b:34:51:e3 [ether] on h7-eth0
? (10.0.0.1) at 42:13:a0:81:c7:88 [ether] on h7-eth0
? (10.0.0.8) at 9a:f8:fc:d2:f7:ea [ether] on h7-eth0
? (10.0.0.4) at e2:55:fc:9e:f6:06 [ether] on h7-eth0
? (10.0.0.3) at 16:cd:e1:d2:96:41 [ether] on h7-eth0
? (10.0.0.6) at ea:16:58:43:83:81 [ether] on h7-eth0
? (10.0.0.2) at 52:13:80:d6:ce:0c [ether] on h7-eth0
root@renuka-HP-Notebook:~/floodlight# []
```

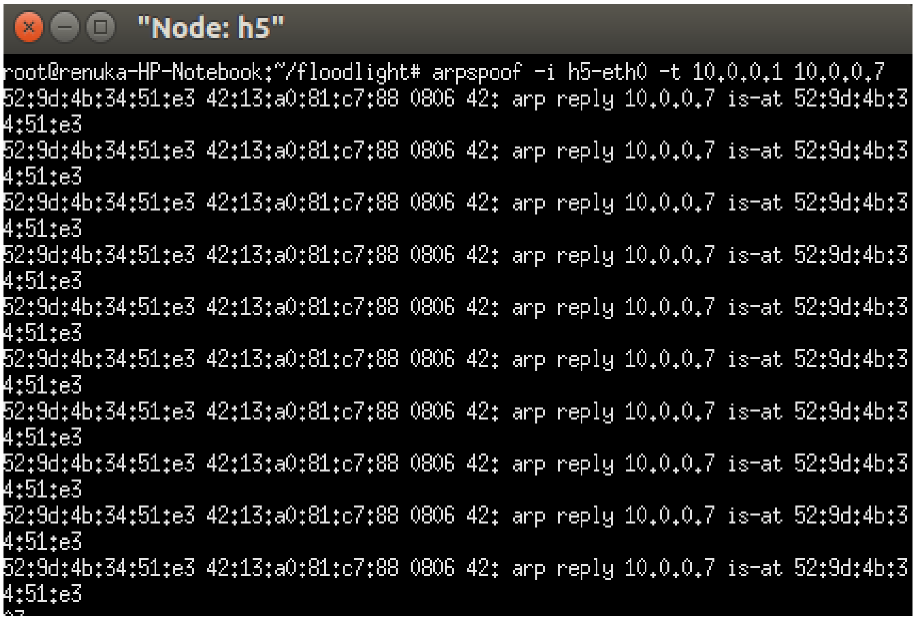
FIG 17: ARP SPOOF GENRATION BY NODE H1



```
root@renuka-HP-Notebook:~/floodlight# arpspoof -i h1-eth0 -t 10.0.0.7 10.0.0.5

42:13:a0:81:c7:88 a6:ff:ea:92:4b:33 0806 42: arp reply 10.0.0.5 is-at 42:13:a0:8
1:c7:88
42:13:a0:81:c7:88 a6:ff:ea:92:4b:33 0806 42: arp reply 10.0.0.5 is-at 42:13:a0:8
1:c7:88
42:13:a0:81:c7:88 a6:ff:ea:92:4b:33 0806 42: arp reply 10.0.0.5 is-at 42:13:a0:8
1:c7:88
42:13:a0:81:c7:88 a6:ff:ea:92:4b:33 0806 42: arp reply 10.0.0.5 is-at 42:13:a0:8
1:c7:88
42:13:a0:81:c7:88 a6:ff:ea:92:4b:33 0806 42: arp reply 10.0.0.5 is-at 42:13:a0:8
1:c7:88
42:13:a0:81:c7:88 a6:ff:ea:92:4b:33 0806 42: arp reply 10.0.0.5 is-at 42:13:a0:8
1:c7:88
42:13:a0:81:c7:88 a6:ff:ea:92:4b:33 0806 42: arp reply 10.0.0.5 is-at 42:13:a0:8
1:c7:88
42:13:a0:81:c7:88 a6:ff:ea:92:4b:33 0806 42: arp reply 10.0.0.5 is-at 42:13:a0:8
1:c7:88
```

FIG 18: ARP SPOOF GENRATION BY NODE H5

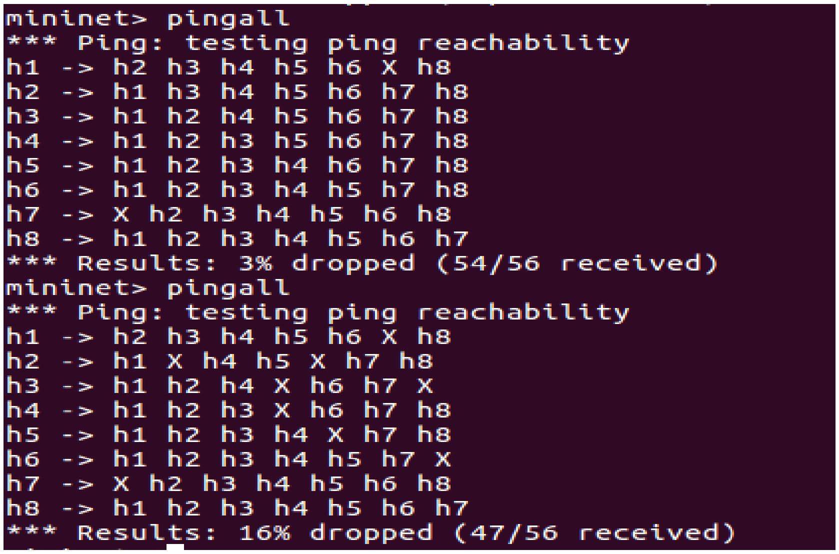


"Node: h5"

```
root@renuka-HP-Notebook:~/floodlight# arpspoof -i h5-eth0 -t 10.0.0.1 10.0.0.7
52:9d:4b:34:51:e3 42:13:a0:81:c7:88 0806 42: arp reply 10.0.0.7 is-at 52:9d:4b:3
4:51:e3
52:9d:4b:34:51:e3 42:13:a0:81:c7:88 0806 42: arp reply 10.0.0.7 is-at 52:9d:4b:3
4:51:e3
52:9d:4b:34:51:e3 42:13:a0:81:c7:88 0806 42: arp reply 10.0.0.7 is-at 52:9d:4b:3
4:51:e3
52:9d:4b:34:51:e3 42:13:a0:81:c7:88 0806 42: arp reply 10.0.0.7 is-at 52:9d:4b:3
4:51:e3
52:9d:4b:34:51:e3 42:13:a0:81:c7:88 0806 42: arp reply 10.0.0.7 is-at 52:9d:4b:3
4:51:e3
52:9d:4b:34:51:e3 42:13:a0:81:c7:88 0806 42: arp reply 10.0.0.7 is-at 52:9d:4b:3
4:51:e3
52:9d:4b:34:51:e3 42:13:a0:81:c7:88 0806 42: arp reply 10.0.0.7 is-at 52:9d:4b:3
4:51:e3
52:9d:4b:34:51:e3 42:13:a0:81:c7:88 0806 42: arp reply 10.0.0.7 is-at 52:9d:4b:3
4:51:e3
52:9d:4b:34:51:e3 42:13:a0:81:c7:88 0806 42: arp reply 10.0.0.7 is-at 52:9d:4b:3
4:51:e3
52:9d:4b:34:51:e3 42:13:a0:81:c7:88 0806 42: arp reply 10.0.0.7 is-at 52:9d:4b:3
4:51:e3
```

The attack being introduces between h1 and h7 created a barrier in communication among them and thus no packet got transferred, which was clearly visible using “pingall”. The packet drop rate got further increased when we increased the number of attacks between several nodes as represented in figure 19. Figure 19 represents the reachability of each node from each other nodes. As we can see that all the nodes are not reachable from every other node and there is 3% initially then 16% of the packet dropped after introduction to the ARP spoofing in floodlight.

FIG 19: CHECKING PACKET DROP (NODE REACHABILITY)



```
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2 h3 h4 h5 h6 X h8
h2 -> h1 h3 h4 h5 h6 h7 h8
h3 -> h1 h2 h4 h5 h6 h7 h8
h4 -> h1 h2 h3 h5 h6 h7 h8
h5 -> h1 h2 h3 h4 h6 h7 h8
h6 -> h1 h2 h3 h4 h5 h7 h8
h7 -> X h2 h3 h4 h5 h6 h8
h8 -> h1 h2 h3 h4 h5 h6 h7
*** Results: 3% dropped (54/56 received)
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2 h3 h4 h5 h6 X h8
h2 -> h1 X h4 h5 X h7 h8
h3 -> h1 h2 h4 X h6 h7 X
h4 -> h1 h2 h3 X h6 h7 h8
h5 -> h1 h2 h3 h4 X h7 h8
h6 -> h1 h2 h3 h4 h5 h7 X
h7 -> X h2 h3 h4 h5 h6 h8
h8 -> h1 h2 h3 h4 h5 h6 h7
*** Results: 16% dropped (47/56 received)
```

DDOS ATTACK IDENTIFICATION AND MITIGATION IN FLOODLIGHT

Simulation of DDOS attack was conducted in floodlight and Mininet by using the same custom topology as generated above.

The simulation of DDOS attack consist of the following steps-

1) Load floodlight module (Figure 20).

2) Create the sflow-rt configuration (Figure 21)

3) Open the terminal for each node and check the connectivity, then perform the flood operation using the command: ping -f 10.0.0.1 (Figure 23)

4) Running a basic ping to verify connectivity, Note the rate doesn’t exceed a few packets per second (Figure 22)

5) Starting a ping flood, the data rate soars to 1.5M pps.

6) At this point the mitigation command is started, the data rate almost falls immediately to zero. (Figure 24)

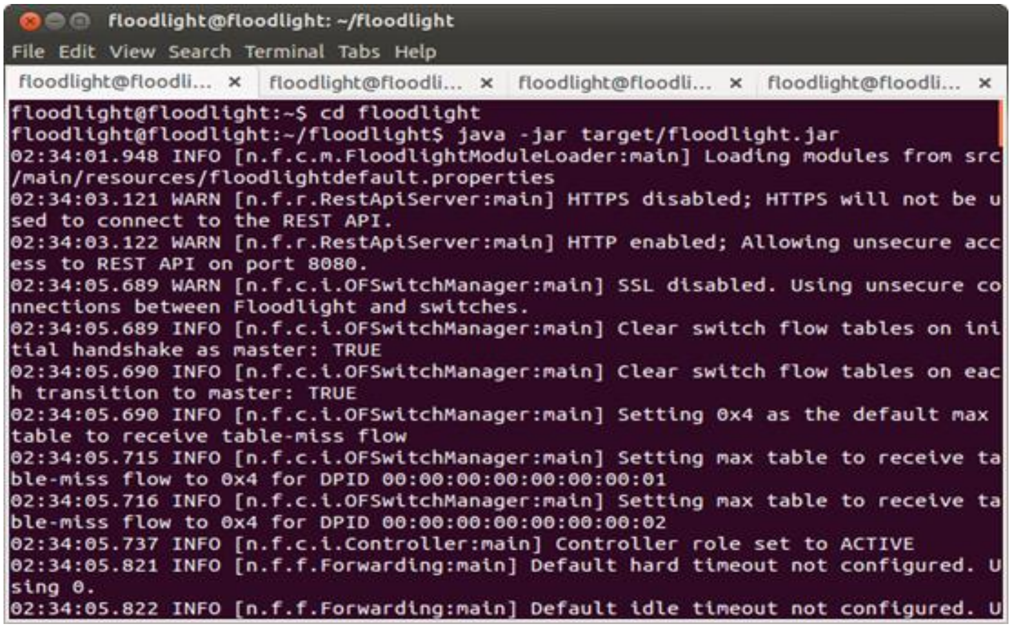
7) Searching for ping operation in the given process IDs. (Figure 25)

8) Mitigating the attack by killing the process. (Figure 26)

9) After mitigation the graph falls down to 0 as the flooding stops. (Figure 27)

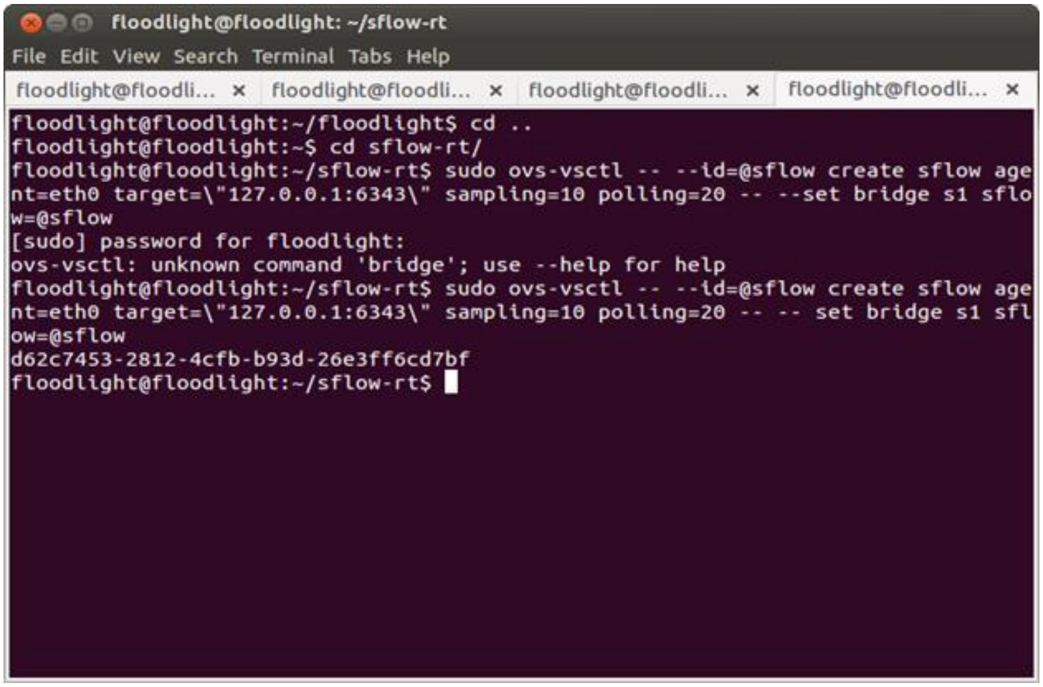
10) The byte transfer came down to zero after flooding stops (Figure 28)

FIG 20: FLOODLIGHT



```
floodlight@floodlight:~$ cd floodlight
floodlight@floodlight:~/floodlight$ java -jar target/floodlight.jar
02:34:01.948 INFO [n.f.c.m.FloodlightModuleLoader:main] Loading modules from src
/main/resources/floodlightdefault.properties
02:34:03.121 WARN [n.f.r.RestApiServer:main] HTTPS disabled; HTTPS will not be u
sed to connect to the REST API.
02:34:03.122 WARN [n.f.r.RestApiServer:main] HTTP enabled; Allowing unsecure acc
ess to REST API on port 8080.
02:34:05.689 WARN [n.f.c.i.OFSwitchManager:main] SSL disabled. Using unsecure co
nnections between Floodlight and switches.
02:34:05.689 INFO [n.f.c.i.OFSwitchManager:main] Clear switch flow tables on ini
tial handshake as master: TRUE
02:34:05.690 INFO [n.f.c.i.OFSwitchManager:main] Clear switch flow tables on eac
h transition to master: TRUE
02:34:05.690 INFO [n.f.c.i.OFSwitchManager:main] Setting 0x4 as the default max
table to receive table-miss flow
02:34:05.715 INFO [n.f.c.i.OFSwitchManager:main] Setting max table to receive ta
ble-miss flow to 0x4 for DPID 00:00:00:00:00:00:00:01
02:34:05.716 INFO [n.f.c.i.OFSwitchManager:main] Setting max table to receive ta
ble-miss flow to 0x4 for DPID 00:00:00:00:00:00:00:02
02:34:05.737 INFO [n.f.c.i.Controller:main] Controller role set to ACTIVE
02:34:05.821 INFO [n.f.f.Forwarding:main] Default hard timeout not configured. U
sing 0.
02:34:05.822 INFO [n.f.f.Forwarding:main] Default idle timeout not configured. U
```

FIG 21: SFLOW-RT CONFIGURATION



```
floodlight@floodlight:~/floodlight$ cd ..
floodlight@floodlight:~$ cd sflow-rt/
floodlight@floodlight:~/sflow-rt$ sudo ovs-vsctl -- --id=@sflow create sflow age
nt=eth0 target=\"127.0.0.1:6343\" sampling=10 polling=20 -- --set bridge s1 sflo
w=@sflow
[sudo] password for floodlight:
ovs-vsctl: unknown command 'bridge'; use --help for help
floodlight@floodlight:~/sflow-rt$ sudo ovs-vsctl -- --id=@sflow create sflow age
nt=eth0 target=\"127.0.0.1:6343\" sampling=10 polling=20 -- -- set bridge s1 sfl
ow=@sflow
d62c7453-2812-4cfb-b93d-26e3ff6cd7bf
floodlight@floodlight:~/sflow-rt$
```

FIG 22: SFLOW-RT CONFIGURATION BEFORE FLOODING

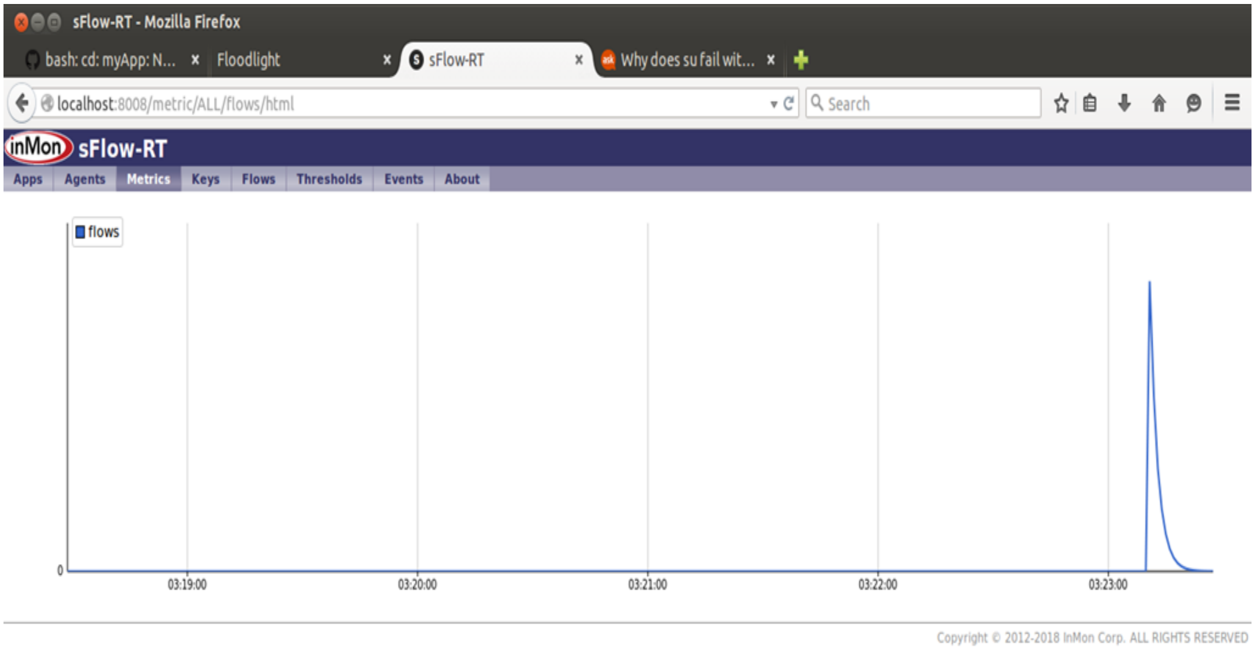
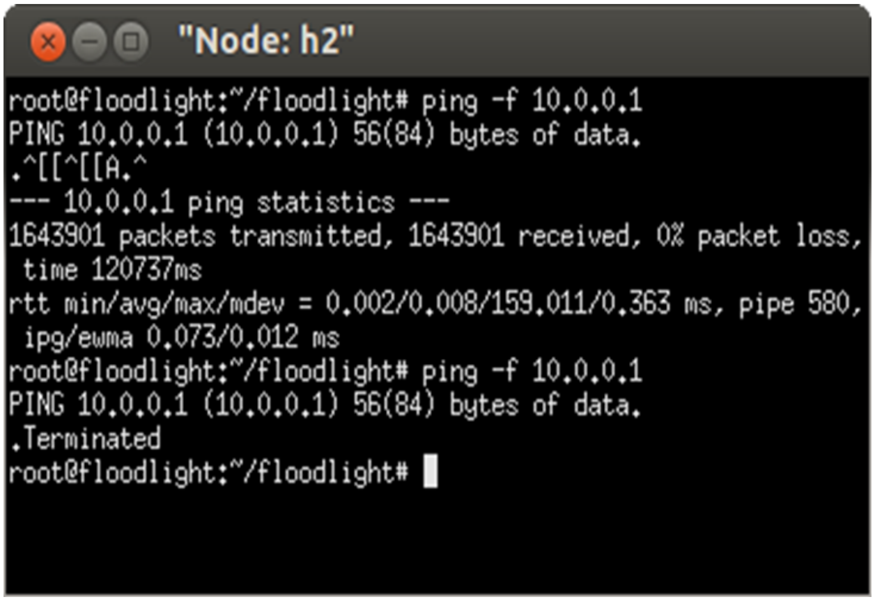


FIG 23: PERFORMING FLOOD OPERATION TO NODE 1



```
root@floodlight:~/floodlight# ping -f 10.0.0.1
PING 10.0.0.1 (10.0.0.1) 56(84) bytes of data.
.^[[^[[A.^
--- 10.0.0.1 ping statistics ---
1643901 packets transmitted, 1643901 received, 0% packet loss,
 time 120737ms
rtt min/avg/max/mdev = 0.002/0.008/159.011/0.363 ms, pipe 580,
 ipg/ewma 0.073/0.012 ms
root@floodlight:~/floodlight# ping -f 10.0.0.1
PING 10.0.0.1 (10.0.0.1) 56(84) bytes of data.
.Terminated
root@floodlight:~/floodlight# █
```

FIG 24: SFLOW-RT CONFIGURATION AFTER FLOODING

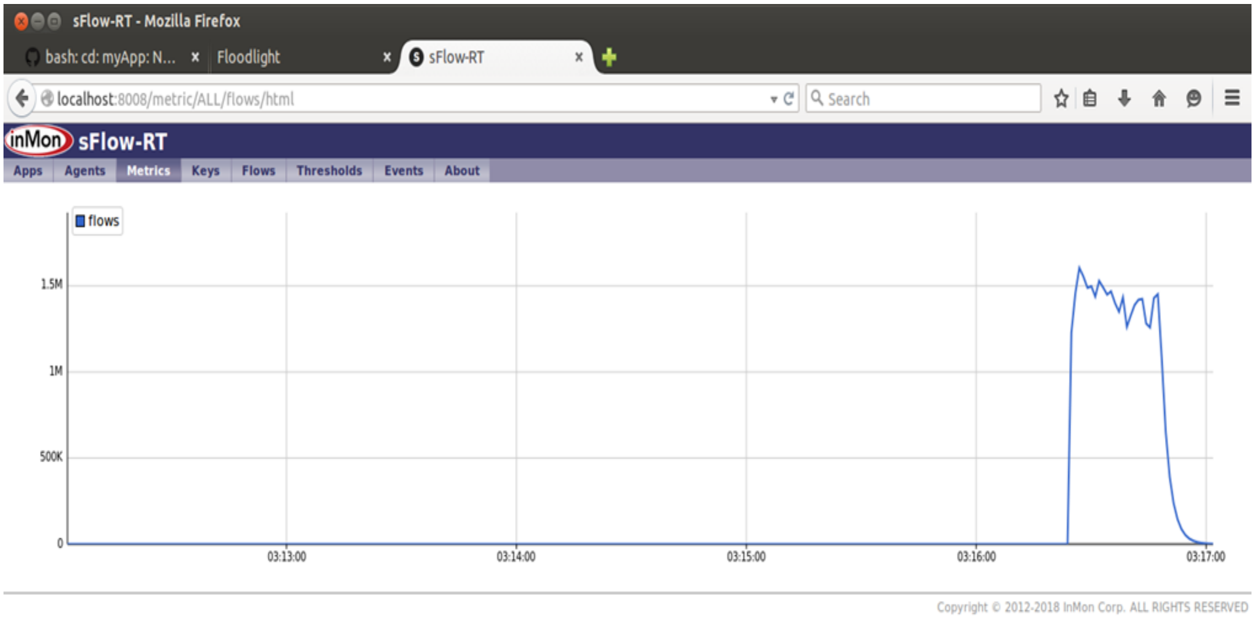
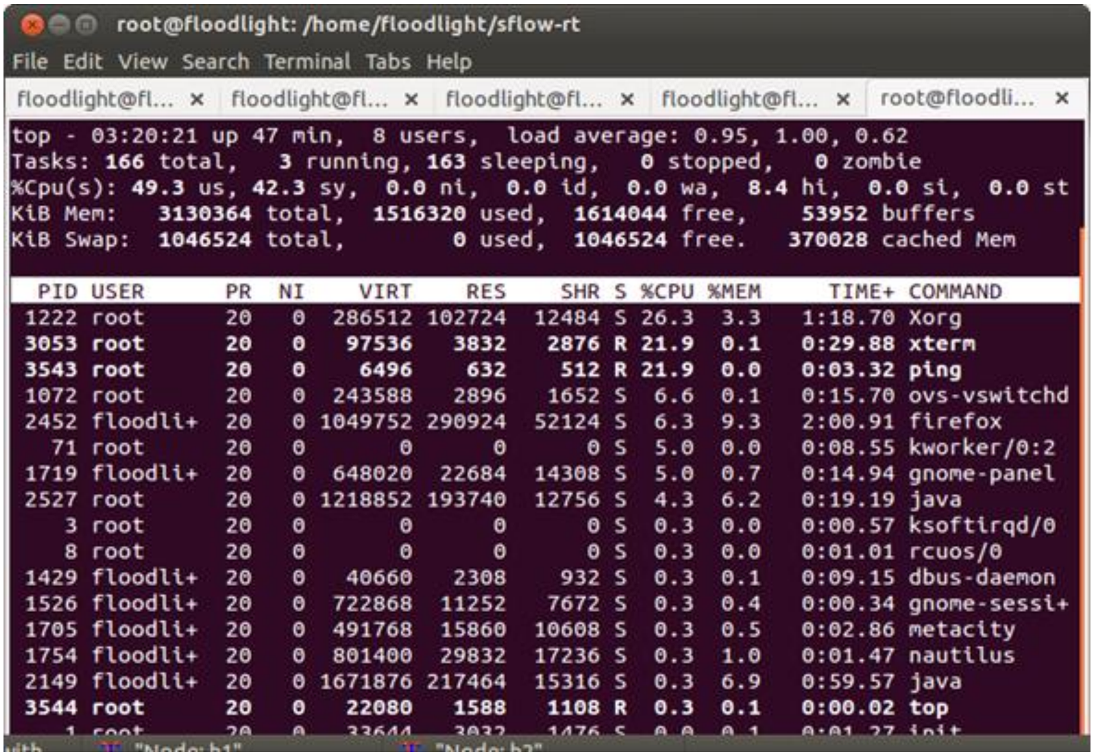


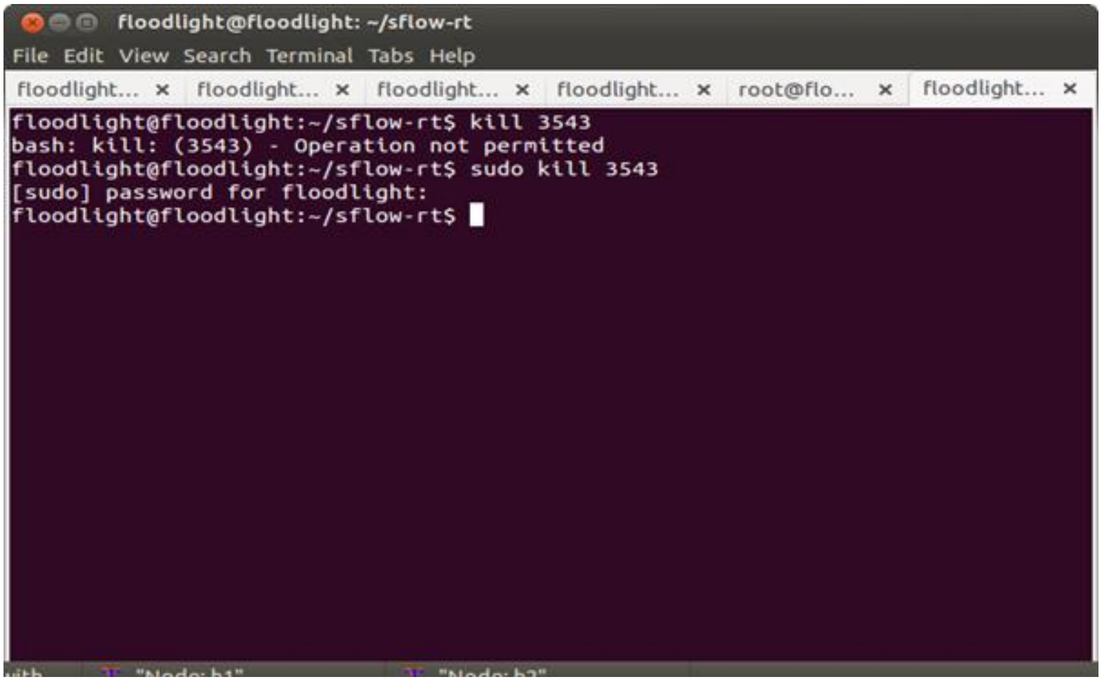
FIG 25: CHECKING PING OPERATION IN LIST OF PROCESS ID s



```
top - 03:20:21 up 47 min,  8 users,  load average: 0.95, 1.00, 0.62
Tasks: 166 total,   3 running, 163 sleeping,   0 stopped,   0 zombie
%Cpu(s): 49.3 us, 42.3 sy,  0.0 ni,  0.0 id,  0.0 wa,  8.4 hi,  0.0 si,  0.0 st
KiB Mem:   3130364 total,  1516320 used,  1614044 free,    53952 buffers
KiB Swap:  1046524 total,        0 used,  1046524 free.   370028 cached Mem

  PID USER      PR  NI    VIRT    RES    SHR S %CPU %MEM     TIME+ COMMAND
 1222 root      20   0  286512 102724  12484 S 26.3  3.3   1:18.70 Xorg
 3053 root      20   0   97536   3832   2876 R 21.9  0.1   0:29.88 xterm
 3543 root      20   0    6496    632    512 R 21.9  0.0   0:03.32 ping
 1072 root      20   0  243588   2896   1652 S  6.6  0.1   0:15.70 ovs-vswitchd
 2452 floodli+  20   0 1049752 290924  52124 S  6.3  9.3   2:00.91 firefox
   71 root      20   0       0      0      0 S  5.0  0.0   0:08.55 kworker/0:2
 1719 floodli+  20   0  648020  22684  14308 S  5.0  0.7   0:14.94 gnome-panel
 2527 root      20   0 1218852 193740  12756 S  4.3  6.2   0:19.19 java
    3 root      20   0       0      0      0 S  0.3  0.0   0:00.57 ksoftirqd/0
    8 root      20   0       0      0      0 S  0.3  0.0   0:01.01 rcuos/0
 1429 floodli+  20   0   40660   2308    932 S  0.3  0.1   0:09.15 dbus-daemon
 1526 floodli+  20   0  722868  11252   7672 S  0.3  0.4   0:00.34 gnome-sessi+
 1705 floodli+  20   0  491768  15860  10608 S  0.3  0.5   0:02.86 metacity
 1754 floodli+  20   0  801400  29832  17236 S  0.3  1.0   0:01.47 nautilus
 2149 floodli+  20   0 1671876 217464  15316 S  0.3  6.9   0:59.57 java
 3544 root      20   0   22080   1588   1108 R  0.3  0.1   0:00.02 top
    1 root      20   0   33644   3032   1476 S  0.0  0.1   0:01.27 init
```

FIG 26: KILLING THE PING OPERATION FOR DDOS MIGITATION



```
floodlight@floodlight:~/sflow-rt$ kill 3543
bash: kill: (3543) - Operation not permitted
floodlight@floodlight:~/sflow-rt$ sudo kill 3543
[sudo] password for floodlight:
floodlight@floodlight:~/sflow-rt$ 
```

FIG 27: SFLOW-RT CONFIGURATION DDOS MITIGATION

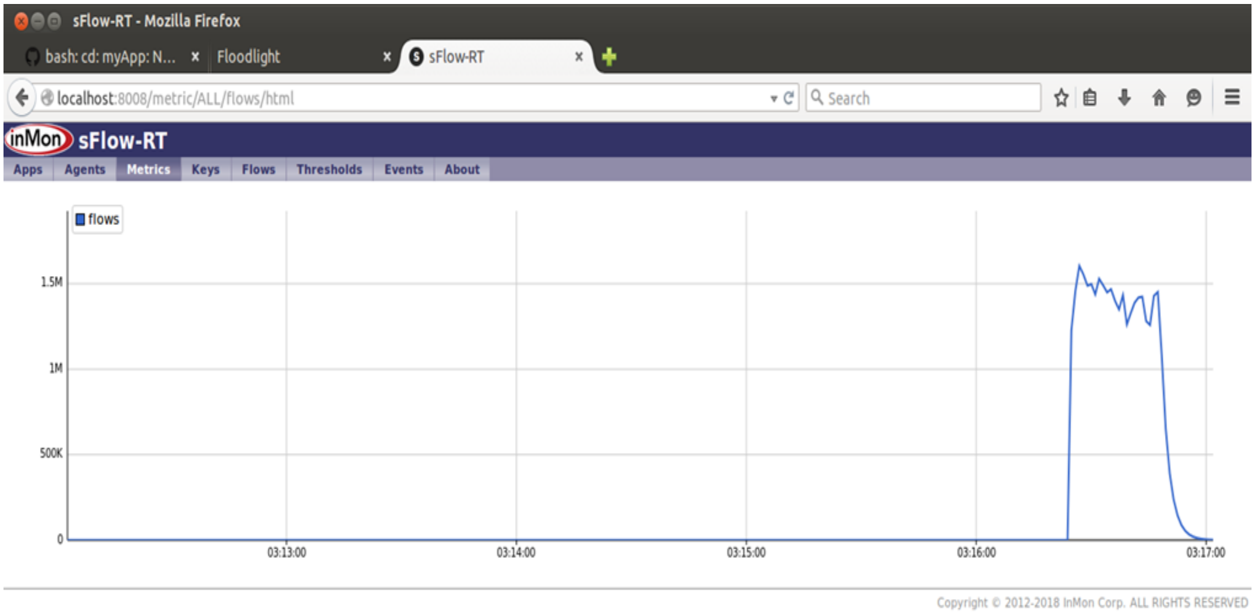
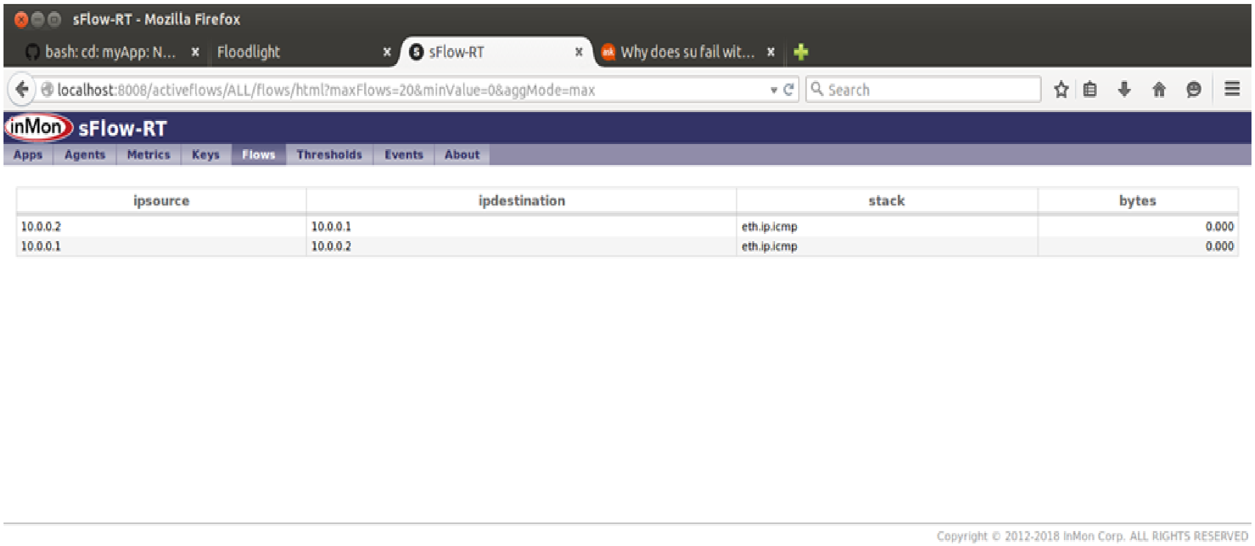


FIG 28: SFLOW-RT CONFIGURATION REPRESENTING BYTE TRANSFER FALLING TO 0


sFlow-RT - Mozilla Firefox
bash: cd: myApp: N...
Floodlight
sFlow-RT
Why does su fail wit...
localhost:8008/activeflows/ALL/flows/html?maxFlows=20&minValue=0&aggMode=max
Search
inMon sFlow-RT
Apps
Agents
Metrics
Keys
Flows
Thresholds
Events
About
Copyright © 2012-2018 InMon Corp. ALL RIGHTS RESERVED

ipsource	ipdestination	stack	bytes
10.0.0.2	10.0.0.1	eth.ip.icmp	0.000
10.0.0.1	10.0.0.2	eth.ip.icmp	0.000

HYBRID CRYPTOGRAPHY IMPLEMENTATION IN VANET

In order to make communications in VANET more secure, we propose a Hybrid security framework based on hybrid cryptography concept, which is a combination of public key cryptographic method and private key cryptographic method. Hybrid cryptographic method is developed by combining RSA and AES (Advanced Encryption Standard) cryptographic algorithms. Several simulation experiments were conducted in NS2 by making changes in the .cc. The proposed method was implemented and the model used for VANET architecture is represented in Figure 3.The Table 2 represents the simulation parameters which was considered to check the effectiveness of the proposed hybrid cryptography method in VANET architecture.

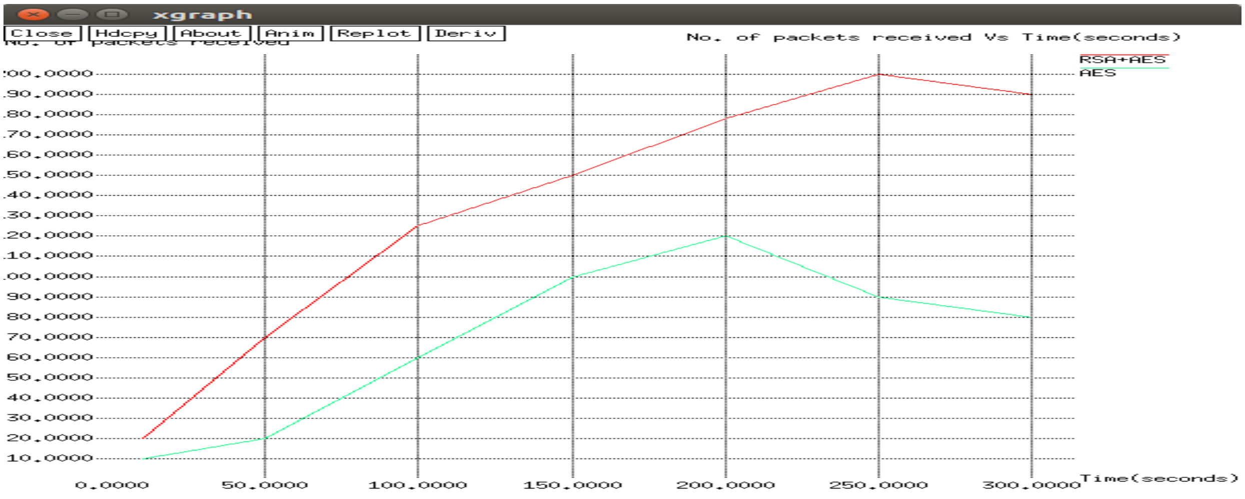
TABLE 2: SIMULATION PARAMETER ENVIRONMENT

SIMULATION PARAMETERS	VALUES
Number of Nodes	100
Type of the packets	Data packets and Control Packets
Packet Size	50 bytes
Type of the traffic	CBR Traffic
Type of the Nodes	Mobile nodes
Simulation Time	300 sec

PERFORMANCE ANALYSIS OF HYBRID CRYPTOGRAPHY

After successful implementation of the hybrid crptography method in NS2, awk files were generated to evaulate the values of different parameters.

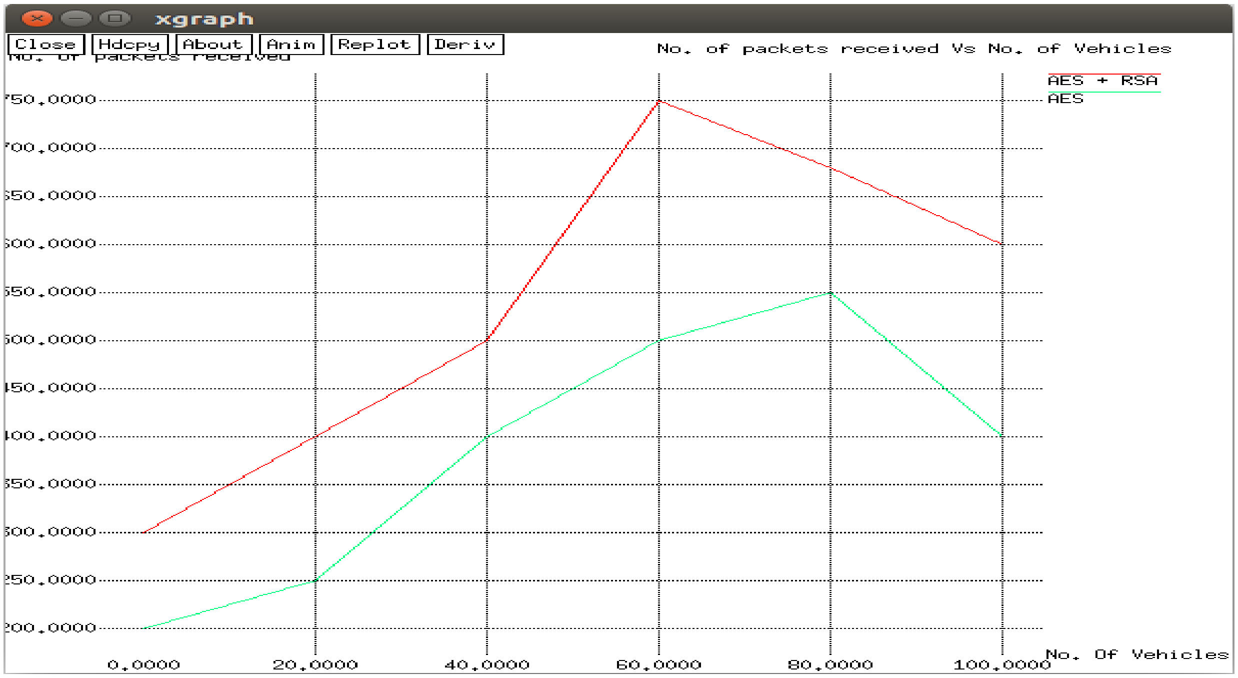
FIG 29: XGRAPH REPRESENTATION OF NO. OF PACKET RECEIVED VS TIME



Graph obtained shows that hybrid cryptography (AES +RSA) method offers higher performance level when compared to AES algorithm. Figure 17 represents the xgraph which describes the behaviour of number of packets received with respect to time.. From the graph, it is evident that the Performance of VANET has increased at a continuous level when hybrid cryptography is implemented. In case of implementing AES algorithm to secure private messages over VANET, performance increased to certain extent and then decreased. Hence in terms of performance, the proposed hybrid cryptography (RSA+AES) algorithm is better than individual cryptographic algorithms.

Figure 18 indicates the success factor level of the hybrid cryptographic method and AES method when the number of nodes in the simulation is increasing. This graph evaluates the ratio of the packets reached the destination correctly as the number of vehicles in the VANET is increasing. From the graph, it is evident that the source factor level of the hybrid cryptography (RSA+AES) is higher when compared to the AES algorithm. The main reason for this is due to the utilisation of two cryptographic methods that results in encrypting the content twice and decrypting it twice, which thereby increases the number of packets that reached destination correctly. Hence in terms of security, the hybrid cryptography (RSA+AES) algorithm is better than individual cryptographic algorithms.

FIG 30: XGRAPH REPRESENTATION OF NO. OF PACKET RECEIVED VS NO. OF VEHICLES



SUMMARY

The topic ***'An Energy-Efficient Approach Towards Network Intelligence in Cooperative communication In Vehicular Environment'*** covers the entire way to achieve our objective. So firstly, the energy efficient approach tells that the main aim is to make the transportation system energy efficient by making the network intelligent so that it can judiciously consume the energy in the cooperative communication environment. Cooperative communication is basically the transfer of messages or any sort of communication with the help of different nodes in between. Here, it is the use of vehicles present in the network to communicate with every other node with hop by hop message transfer. So, over entire objective of the R&D project is to make the secure and intelligent transportation system using new technologies such as fog computing and Software Defined Networking and making the system secure by analysing different security issues and the methods to tackle them. So, this objective has been achieved by first digging the roots of the vanet system. Firstly, we studied the concept of Vehicular Communication System, VCN (Vehicular Cloud Network) and VANET (Vehicular Ad Hoc Network) which is a promising approach for future Intelligent Transportation System (ITS). Vehicular Communication Systems are the networks in which vehicles and roadside units are the communicating nodes, providing each other with information, such as safety warnings and traffic information. They can be effective in avoiding accidents and traffic congestion.

Vehicles are expected to carry relatively more communication systems, on board computing facilities, storage and increased sensing power. Hence, several technologies have been deployed to maintain and promote Intelligent Transportation Systems (ITS). Recently, several solutions were proposed to address the challenges and issues of vehicular networks. Vehicular Cloud Computing (VCC) is one of the solutions. VCC is a new hybrid technology that has a remarkable impact on traffic management and road safety by instantly using vehicular resources, such as computing, storage and internet for decision making. VANETs change the computing and networking models, which leads to a future vehicular networking system. Here comes the introduction of VCN (Vehicular Cloud Networking), VCN = VCC + ICN. The eventual goal of VCN is to create a vehicle cloud and to encourage collaborations amongst cloud members to produce advanced vehicular services that individual alone cannot make.

To address the challenges faced by VANETS such as real time services, we consider two emerging networking paradigms; Software Defined Networking (SDN) and Edge Computing (Fog Computing). The core idea of SDN is the separation between network control (control plane) and forwarding functions (data plane). This novel networking paradigm offers many benefits compared to the traditional distributed approaches. Firstly, it simplifies networking in both development and deployment of new protocols and applications. With software-based controller, network operators are much easier to program, modify, manipulate and configure of a protocol in a centralized way without independently accessing and configuring individual network hardware devices scattering across the whole network. Secondly, SDN-based architectures provide centralized controller to the network with global knowledge of the network state which are capable of controlling network infrastructure in a vendor-independent manner. These network devices just simply accept policies from the controller without understanding and implementing various network protocols standards, resulting in directly control, program, orchestrate, and manage network resources at the SDN controller. This feature, thus, saves a lot of workforce and resources. To deploy SDN architecture, the key requirement is a protocol for communicating between Data Plane and Control Plane (South-bound Interface SBI). This protocol should be standardized and vender-agnostic so as networking companies should follow to develop their products. The SBI protocol facilitates heterogeneous networking switches and routes in the same way, thus, simplifies the operations of the network system. OpenFlow is the typical example of such kind of protocol. OpenFlow-enabled networking switches and routers maintain a forwarding flow-table containing three entries called rules, actions associated with each rule defining how packets should be processed, and statistic counting the number of packets and bytes for the flow.

In Connected Vehicle Environment, the communication channels are unstable and the topology changes frequently, Communication between the Server and the mobile devices can use various wireless technologies. Efficient resource and energy management is essential but difficult because of the inflexibility of the current network structure. One switches or router have to perform all the functions including forwarding, routing and management of the network resources. Hence, due to increase in the demand of these services have raised big concern over the amount of energy consumed. Solution must be provided to reduce the amount of energy consumption in the vehicular environment. There is a huge variation in the traffic during the night and the day time, traffic during the night time is less during the day time, this leads to an opportunity to optimize the consumption of energy in VANETS. So, this was the motivation behind the main objective to design an efficient network management strategy with guaranteed satisfaction of network traffic demand utilizing SDN and fog computing in connected vehicular environment and to ensure secure communication in VANETs. We studied about the implementation of SDN and fog computing in VANET architecture and realized that efficiency of the entire model can still be improved. Thus, we designed an algorithm that improves the energy efficiency of the entire model. We generated our proposed design model using Mininet and Floodlight. The topology generated had the shortest path algorithm implemented in it as per our proposed algorithm.

After, the energy efficiency approach we focused in the security area in Vehicular environment. To start with the security aspects, we first analyzed the different types of security requirements. These include Data Integrity, Data Verification, Non-repudiation, Availability and Privacy. Firstly, Availability is a very important factor for VANETs. It guarantees that the network is functional, and useful information is available at any functioning time. This critical security requirement for VANETs, which main purpose is to ensure the users' lives, is an important target for most of the attackers. Several attacks are in this category, the most famous are the Denial of Service attacks (DoS). Next to ensure authenticity in a vehicular network is to protect the authentic nodes from outside or inside attackers infiltrating the network using a falsified identity. Confidentiality is another important security requirement for VANETs communications, it ensures that data are only read by authorized parties. In the absence of a mechanism to ensure the confidentiality of the exchanged data between nodes in a vehicular network, exchanged messages are particularly vulnerable to attacks such as the improper collection of clear information. Integrity mechanisms help therefore to protect information against modification, deletion or addition attacks. Non-repudiation in computer security means the ability to verify that the sender and the receiver are the entities who claim to have respectively sent or received the message. After studying these security threats in VANET architecture, we narrowed down our objective to three aspects. Privacy, Availability and Data Integrity.

To ensure privacy we worked on the concept of Heartbeat messages. Unlike previous works discussed in review of literature that proposed explicit synchronization between a group of vehicles and/or required pseudonym change in a designated physical area (i.e., a static mix zone), we propose a much simpler approach that does not need any explicit cooperation between vehicles and any infrastructure support. The basic idea that vehicles should not transmit heartbeat messages when their speed drops below a given threshold, say 30 km/h, and they should change pseudonym during each such silent period. This ensures that vehicles stopping at traffic lights or moving slowly in a traffic jam will all refrain from transmitting heartbeats and change their pseudonyms nearly at the same time and location. Since this approach also have some drawbacks, we proposed that instead of eliminating the heartbeat messages during low speeds, we can secure the heartbeat messages while transmission.

After ensuring the privacy, still Data integrity was an issue. So, to securely transmit data in vehicular environment, we decided to go with RSA algorithm for data encryption. Hence, we studied the algorithm and implemented it in JAVA to check its correctness. We simulated the encryption algorithm in ns2 and showed the results in the previous section.

Next purpose was to cover the third aspect i.e. availability. So, we worked on the three types of attacks in VANET architecture. We studied the concept of ARP spoofing and why is it important to mitigate it. ARP spoofing is a type of attack in which a malicious actor sends falsified ARP (Address Resolution Protocol) messages over a local area network. This results in the linking of an attacker's MAC address with the IP address of a legitimate computer or server on the network. Once the attacker's MAC address is connected to an authentic IP address, the attacker will begin receiving any data that is intended for that IP address. ARP spoofing can enable malicious parties to intercept, modify or even stop data in-transit. ARP spoofing attacks can only occur on local area networks that utilize the Address Resolution Protocol. To understand its effect more clearly, we also performed simulation with 8 nodes and 6 switches in NS2.

We then studied the concept of DDOS and why is it important to mitigate it. A Denial-of-Service (DoS) attack is an attack meant to shut down a machine or network, making it inaccessible to its intended users. To understand its effect more clearly, we performed its implementation using Floodlight and sFlow-RT (to analyze the data information collected).

Then the final part was to strengthen the encryption of data using the Hybrid encryption. Earlier, we had implemented RSA algorithm (an asymmetric encryption technique) for data encryption but on further studying about the advantages and disadvantages of symmetric and asymmetric encryption algorithms, we designed a hybrid encryption (RSA + AES) algorithm that incorporates the benefits of both symmetric and asymmetric encryption algorithms and provides an additional level of security for the transmission of messages. We then implemented this algorithm in Java and implemented its simulation in NS2.

This was our approach to make the transportation system energy efficient and secure. Although we faced many difficulties such in understanding and implementation on the software mininet, floodlight and working with ns2 in terms of security implementation, we successfully completed the project with some future scope. In future to extend this work, we can consider other security aspects to make the system more secure and in terms of energy efficiency we can try to achieve this in different scenarios, especially in our country India.

All the simulation results have been shown in the result section. It was a really good experience of working in depth in the field of VANETS and creating new ways to make it feasible for the implementation in real life.

FUTURE WORK

Vehicular Adhoc Networks (VANETs) have been attracted a lot of research recent years. Although VANETs are deployed in reality offering several services, the current architecture has been facing many difficulties in deployment and management because of poor connectivity, less scalability, less flexibility and less intelligence. Software-defined networking is an introduction of software to the traditional networking leading to more control over the networking devices, it centralizes the control to the SDN controller. The switches are programmable hence makes the placement of new rules in switches easier. Fog computing extends the services of the cloud computing to the edge of the enterprise network leading to the provision of real time services in VANET architecture. SDN-based architecture provides flexibility, scalability, programmability and global knowledge while Fog Computing offers delay-sensitive and location-awareness services which could be satisfy the demands of future VANETs scenarios. Integration of the technologies such as fog computing, software-defined networking and 5G services have led to several new opportunities and solves the challenges related to management and deployment of the VANET architecture. Due to increase in the demand of cloud services a large amount of energy is consumed, hence energy optimization is an emergent issue. We designed an energy-efficient network management strategy which reduces the consumption of energy by the networking devices. Various simulation experiment was conducted using tools such as Floodlight and Mininet. Since software-defined networking is a new emerging technology working with it is a major issue. In the given span of time we could implement one of the major part of the project that is the implementation of the shortest path algorithm rest left work such as efficient management introduction in SDN would be carried out further in future.

As SDN relies on centralized network management, it adds to administrators' worries regarding server (controller) security. If by any means server gets hacked, then whole network becomes more prone to be attacked. By keeping this security issues in SDN in mind we have shown several attacks such as implementation of the ARP Spoofing in SDN, DDOS attack and its mitigation. We considered only some aspects of the security in Software-defined networking in the given span of time, but we will be focusing on little more on the security part in future.

REFERENCES

[1] Tseng, Fan-Hsun, Li-Der Chou, and Han-Chieh Chao. "A survey of black hole attacks in wireless mobile ad hoc networks." Human-centric Computing and Information Sciences 1.1 (2011): 4.

[2] Lee, Euisin, Eun-Kyu Lee, Mario Gerla, and Soon Y. Oh. "Vehicular cloud networking: architecture and design principles." *IEEE Communications Magazine* 52, no. 2 (2014): 148-155.

[3] Truong, Nguyen B., Gyu Myoung Lee, and Yacine Ghamri-Doudane. "Software defined networking-based vehicular adhoc network with fog computing." In *Integrated Network Management (IM), 2015 IFIP/IEEE International Symposium on*, pp. 1202-1207. IEEE, 2015.

[4] Park, Seongjin, and Younghwan Yoo. "Network Intelligence Based on Network State Information for Connected Vehicles Utilizing Fog Computing." *Mobile Information Systems* 2017 (2017).

[5] Qu, Fengzhong, Zhihui Wu, Fei-Yue Wang, and Woong Cho. "A security and privacy review of VANETs." *IEEE Transactions on Intelligent Transportation Systems* 16, no. 6 (2015): 2985-2996.

[6] Dargahi, Tooska, Alberto Caponi, Moreno Ambrosin, Giuseppe Bianchi, and Mauro Conti. "A survey on the security of stateful SDN data planes." *IEEE Communications Surveys & Tutorials*19, no. 3 (2017): 1701-1725.

[7] Akhunzada, Adnan, Ejaz Ahmed, Abdullah Gani, Muhammad Khurram Khan, Muhammad Imran, and Sghaier Guizani. "Securing software defined networks: taxonomy, requirements, and open issues." *IEEE Communications Magazine* 53, no. 4 (2015): 36-44.

[8] Buttyán, Levente, Tamás Holczer, and István Vajda. "On the effectiveness of changing pseudonyms to provide location privacy in VANETs." In *European Workshop on Security in Ad-hoc and Sensor Networks*, pp. 129-141. Springer, Berlin, Heidelberg, 2007.

[9] Lu, Rongxing, Xiaodong Lin, Tom H. Luan, Xiaohui Liang, and Xuemin Shen. "Pseudonym changing at social spots: An effective strategy for location privacy in vanets." *IEEE Transactions on Vehicular Technology* 61, no. 1 (2012): 86-96.

[10] Arain, Qasim Ali, Zhongliang Deng, Imran Memon, Asma Zubedi, Jichao Jiao, Aisha Ashraf, and Muhammad Saad Khan. "Privacy protection with dynamic pseudonym-based multiple mix-zones over road networks." *China Communications* 14, no. 4 (2017): 89-100.

[11] Zeadally, S., Hunt, R., Chen, Y. S., Irwin, A., & Hassan, A. (2012). Vehicular ad hoc networks (VANETS): status, results, and challenges. Telecommunication Systems, 50(4), 217-241.