Malicious fog node management framework for update dissemination in software-define vehicles

By

Nadia Kalsoom

2019-NUST-MS-IT-17 205030

Supervisor

Dr. Asad Waqar Malik

Department of Computing

A thesis submitted in partial fulfillment of the requirements for the degree of Masters of Science in Information Technology (MS IT)

In

School of Electrical Engineering and Computer Science,
National University of Sciences and Technology (NUST),
Islamabad, Pakistan.

(June 2021)

Approval

It is certified that the contents and form of the thesis entitled "**Malicious fog node management framework for update dissemination in software-define vehicles**" submitted by **Nadia Kalsoom** have been found satisfactory for the requirement of the degree.

Advisor: **Dr. Asad Waqar Malik**

Asad

Signature:________________

27-Apr-2021

Date: ________________

Committee Member 1: **Dr. Arsalan Ahmad**

Signature: ________________

04-May-2021

Date: ________________

Committee Member 2: **Dr. Anis UR Rahman**

Signature: ________________

12-May-2021

Date: ________________

Dedication

I would like to dedicate this thesis to my parents who have always gone above and beyond to assure my utmost comfort. It is due of their prayers, love, and support that enabled me to reach where I am today.

Certificate of Originality

I hereby declare that this submission is my own work and to the best of my knowledge it contains no materials previously published or written by another person, nor material which to a substantial extent has been accepted for the award of any degree or diploma at NUST SEECS or at any other educational institute, except where due acknowledgement has been made in the thesis. Any contribution made to the research by others, with whom I have worked at NUST SEECS or elsewhere, is explicitly acknowledged in the thesis.

I also declare that the intellectual content of this thesis is the product of my own work, except for the assistance from others in the project's design and conception or in style, presentation and linguistics which has been acknowledged.

Author Name: **Nadia Kalsoom**

Signature: ________________

THESIS ACCEPTANCE CERTIFICATE

Certified that final copy of MS/MPhil thesis entitled ”Malicious fog node management framework for update dissemination in software-defined vehicles” written by NADIA KALSOOM, (Registration No 00000205030), of SEECS has been vetted by the undersigned, found complete in all respects as per NUST Statutes/Regulations, is free of plagiarism, errors and mistakes and is accepted as partial fulfillment for award of MS/M Phil degree. It is further certified that necessary amendments as pointed out by GEC members of the scholar have also been incorporated in the said thesis.

Signature: Asad

Name of Advisor: **Dr. Asad Waqar Malik**

Date: 27-Apr-2021

Signature (HOD): ____________

Date: ____________

Signature (Dean/Principal): ____________

Date: ____________

Acknowledgment

I would firstly like to pay my gratitude to Allah Almighty for showering upon me His countless blessings at every instance of life. Without His Will I couldn't have imagined completing this task.

I am most thankful to Dr. Asad Waqar Malik for encouraging and guiding me at every step to complete my thesis not only as a wonderful supervisor but as a brilliant mentor as well. His contribution in stimulating suggestions and encouragement, helped me to coordinate my thesis into its final form.

I am also thankful to the School of Electrical Engineering and Computer Sciences department of Computing, and the faculty for invoking in me a strong educational foundation, which enabled me to complete this thesis.

Table of Contents

List of Figures

Abstract

Fog computing contains fog nodes with minimal capabilities that are strategically placed near data-producing sources. These fog nodes are complete small interactive tasks that require low latency. In the cloud data center, bigger, more latency-tolerant activities can be done. Multiple sensors and actuators in smart cars automate various activities such as entertainment, acceleration, and traffic monitoring, which are common use cases for fog computing. Over the air (OTA) is a platform focused on fog computing for safely distributing updates to smart cars. This enables automakers to send directly updates to the vehicles without having to visit the dealership.

A traffic Simulation and Analysis framework based on bottom-up process modeling and simulation using real-scale road structures have been proposed. A dynamic update simulation model has been developed in which the primary goal being an algorithm proposed to devise the most optimal, secure OTA update strategy and tests look at how OTA updates are affected by network threats. The fog nodes and vehicles in the simulation have been modeled as agents enabling us to understand the collective behavior of the agents in the simulated environment. Furthermore, the formulation and evaluation of secure update strategies on congested roads are discussed. The functionality of the algorithm is demonstrated using simplistic examples and then applied using four scenarios against different traffic congestion values. Validation of the proposed algorithm and its performance has also been conducted, the details of which are mentioned later on in the study. The main goal of this research is to assist in developing effective threat management plans for OTA updates.

Keywords: *Road Traffic Simulation, OTA updates, Fog Computing, Process Modelling*

Chapter 1

Introduction

This chapter provides the opening and general information of the research to provide a clear understanding about this thesis. It also covers the problem statement along with solution statement.

It is predicted that we are about to enter a new age in which world will be overrun by smart objects. Specifically, due to the exponential advancement of enabling technologies in smart utensils used in our everyday lives are now equipped with networking capabilities through wired and wireless networks, as well as computing abilities via embedded chips and remote clouds. In the automobile business, the Internet of Vehicles represents a genuine technical breakthrough.

Software development has entered in an era where manufacturers designing their products in such a way that new updates are placed on the cloud, and these updates delivered to the users. However, keeping theses updates up to date is a major challenge. Despite these challenges some other features have been added to this update process like dealing with technical matters, security servicing and fixing errors.The question then becomes how to efficiently, effectively, and securely distribute these various updates over the air to millions of automobiles across the world. As the number of in-car connect systems increases, comparatively, the incoming updates increasing exponentially.

Therefore, in order to further strengthen this updates system which is known as over the air update, it is necessarily important to anticipate the problems that may arise and take possible steps to deal with them. Leading automobile manufacturers and over-the-air technology suppliers are collaborating with Akamai [1] to enhance connected car apps and upgrades, ensuring that they are delivered on time, reliably, and securely.To implement and manage their technology, they provide an extensible network framework and connect manufactures, software teams and more importantly vehicles.

One of the most important aspects of installing IoT technology is the ability to properly deliver updates. They must keep up as more devices connect to one other. Managing a secure world wide network and controlling to effectively distribute efficient and secure updates to the millions of the devices is not only complex but also a costly process. The mission of the IoT deployment team is to decrease the number of supported versions in the field, offer essential security updates rapidly, and deliver new capabilities to improve their products and expand their client relationships. When opposed to manual deployment, over-the-air updates must be carried out in a safe way so that time and money can be saved.

In a typical Vehicular Networks offer cutting-edge services for various modes of transportation and allow users to be more aware to use transportation networks in automatic, secure and cooperative manner. It can refer to all modes of transportation, it is most commonly used in the field of road transportation, including infrastructure, automobiles, and users, as well as traffic and accessibility management and interfaces with other modes of transportation. Forming an intelligent transportation system, vehicles enabled with On-Board Units (OBUs), able to communicate with other vehicles forming a Vehicle to Vehicle (V2V) connection [2] and roadside units (e.g., fog nodes).

Other infrastructures, such as cellular Base Stations and Wi-Fi APs, also provide data services to vehicles. In V2V communication model, the OBU communicates with other vehicles and exchanges data. For example, in foggy conditions, vehicle fitted with OBU may determine the precise distance to its neighboring peers or may share their position, speed and direction information to avoid traffic congestions and accidents. However, technical advances in ITS and VANET have ushered in the philosophy of autonomous and connected vehicular systems, enriching the notion of intelligence in the transportation field in order to improve customer comfort and road safety. Vehicles are being linked to the Internet as the Internet of Things (IoT), evolves with aim of supplying ubiquitous access to the information. This results in a new technical breakthrough known as the Internet of Vehicles (IoV).

1.1 Problem Statement

OTA update mechanism that has been thoroughly tested and verified are needed to be developed in order to improve the security and safety of the fog nodes. There is a need to develop a simulation framework that models the update dynamics of fog nodes and their response/behavior when they are under security threats e.g. interactions in Denial of Service attack. This will lead to the analysis of key factors affecting the complete update process. A simulation framework that incorporates the fog layer coupled with traffic library will enable us to identify the limitations in the current update methods/models deployed in fog computing. This will further provide insights that will fuel the decisions for a better update approach.

1.2 Solution Statement

A Secure OTA Update Algorithm (SUA) is proposed with the help of modelling and simulation framework. The mentioned framework provides decision support and strategies by the modeling and simulation of different scenarios encountered in a traffic OTA update process. These important findings then aid in the fine tuning and development of the proposed algorithm which assists in the complete and quick update of entire traffic network.

1.3 Key Contributions

With the help of the proposed algorithm combined with the process modelling framework, the vehicles will be able to (i) get update in the shortest possible time while (ii) avoiding congestion at fog nodes. SUA able to (iii) identify black hole attack (iv) identify accurate update in case of malicious attack. OEM will be able to (v) utilize our proposed solution for the devising reliable update mechanism (vi) have system consistency (vii) optimal resource utilization.

1.4 Thesis Organization

Thesis organized in the following chapters:

1.4.1 Chapter 2: Background

Chapter 2 provides brief overview of OTA update mechanism. This chapter also explains current facts and figures caused due to malicious update in the traffic history followed by the basic concepts of fog computing and also a brief introduction of AnyLogic Simulation software.

1.4.2 Chapter 3: Literature Review

This chapter explains the work done so far related to fog node security techniques and OTA update forwarding approaches. It also provides a table of the literature used in this study followed by the position of this thesis in the state of the art.

1.4.3 Chapter 4: Methodology

Modeling and simulation of OTA update using fog computing is carried out using AnyLogic simulation Software. Brief introduction to the simulation framework is provided in this section along with detailed explanation of the proposed algorithm and its details. Furthermore, the working and implementation of the proposed algorithm is also explained in detail.

1.4.4 Chapter 5: Simulation and Results

This chapter is dedicated to the demonstration of the functionality and results of our proposed system. Validate proposed system performance with random system. The chapter is further concluded by the obtained simulation results and their discussion.

1.4.5 Chapter 6: Conclusion and Future work

Brief summary of the thesis research work is presented in this section provided with tasks that can be carried out later for further research studies.

Chapter 2

Background

This chapter provides a brief overview of fog computing, proposed architectures use to provide secure updates, followed by an overview of traffic simulation.

Vehicle suppliers send software updates to the cloud, where all linked cars can install latest models of the software, with cloud-based OTA updates. OTA updates can perform while the vehicle is parked or while it is moving. There are some disadvantages of installing updates from the cloud data center [3].Due to its distant location from the vehicle, the cloud cannot have quick response times for time-critical upgrades. When critical updates are sent over worldwide linked channels, they become vulnerable to security attacks. There has been some research into delivering security to OTA systems in vehicles. There has been some work that looks at providing security to OT A updates in vehicles [4] , [5]. Cloud systems are taken closer to data generators, such as humans and computers, using fog computing [6]. In Which fog nodes are at close vicinity to data generating devices such actuator and sensors therefor data proceed locally and get immediate response [7].

Furthermore, since the fog nodes have been distributed in an ad hoc manner, the system has become reliable and more stable that there is no single pint of failure [8]. Mobility is a significant issue for OTA [9]. Because of this, a mobile car can initiate connection in new fog node network and disconnect previously connected fog nodes. It is fascination research direction to examine vehicle movement effects on OTA update mechanism. [10] suggested handover solution to the problem of mobility. The impetus for offering a trusted fog environment for IoT based services rise from the open challenges and problems associated with fog computing, which are still being researched and developed. Bringing computational resources to network edges is a hot topic among academics [11, 12]. This would make data processing at the edge easier for time-sensitive applications and services, allowing for faster responses. Fog nodes are located at the network's edge, and they lack the resources and computing power of cloud [13, 14]. In result, fog nodes will quickly become overburdened with service requests. Another problem with cyber-threats that has been mentioned is hostile/open deployment [15,16,17]. As a result, there are misbehaving fogs that may engage in biased attacks in order to tarnish the credibility of an IoT service [18].

A malicious fog node will interrupt network operations by different attacks. In this research, we consider the following threats that specifically impact on the reliability and collaboration among fog nodes [19].

Tampering:
In this type of attack, attacker change the behavior of fog node by manipulating network updates to mislead vehicles. This fake update circulation is a serious drain on network resources, and it is difficult to identify such malicious nodes.

Denial of Service (DoS:)
This attack makes the fog node inaccessible, consumes network resources and prevent update requests from vehicles. Fog nodes are at high risk to DoS attack as compare to other distrusted approaches.

Fog Security Requirement: The following safety criteria should be met in order to allow a stable fog collaboration model [19] that offers a protected data sharing environment.

Service Integrity: Since the packets of the transmitted service between fog nodes may be modified by malicious fog node during the transmitting time, the packets must be tested so that they fully fit what they originally sent (such as packet authentication from source).

Service Availability: The provision of fog services ensures that, as necessary, the services must be available. The accessibility of services will be greatly impacted by unforeseen circumstances such as service crashes. In addition, the fog should be capable of tolerating DoS attempts aimed at crashing the fog services. It is interesting that the update dissemination among fog node tends to enhance service availability.
In this article, we propose a stable OTA update framework for detecting and addressing black hole and malicious attacks. We also propose an algorithm called: Secure Update Dissemination Algorithm (SUA). The prominent features of the framework are:

1. Reduce the average update time of the whole network.
2. To prevent network congestion on fog nodes, design a pivot selection mechanism.
3. A fog synchronization method is proposed to identify black hole attack.
4. System remains consistent even during malicious attacks.
5. Develop SUA, installed on fog units as well as vehicles
    (a) Identify black hole and malicious attacks and then react accordingly.
6. Simulation data use to determine suggested system performance in term of number of successful, fail and non-updated vehicles and in term of time duration required to update entire network

2.1 Fog computing

Fog computing refers to a network that extends from the point where data is created to the point where it will be stored. Fog is a distributed network environment layer that is intimately linked to cloud computing and the internet of things (IoT). Low-latency network connect devices and analytics via fog computing. As a result, compared to if the data had to be transmitted back to the cloud or data centre for processing, this design decreases the amount of bandwidth required. It can also be utilised where there isn't enough bandwidth to transport data and it needs to be processed locally. A fog network may also include security measures such as split network traffic and virtual firewalls to secure it. Connected cars are increasingly relying on real-time data. Because of the distinctive characteristics of the vehicle environment, the applicable technologies as well as the appropriate security model play a critical role in passenger safety.

Furthermore, fog computing is a viable solution for meeting the needs of VANETs. For example, provides a rapid response to the lower layer i.e, IoT device. It also minimizes the load on the cloud and allows for real-time analysis of data streams in the cloud. It is a good way to enhance traffic management and expand safety services, which both require local data and real-time processing. Because vehicular network relies on event triggers, therefore data precision and trust management are interesting challenges in terms of a security model. It is self-evident that threat possibility degrade data accuracy, resulting in decreased network efficiency. Several security risks, such as update manipulation and black holes, have been presented as security vulnerabilities in this research.

2.2 Traffic Simulation

Traffic simulation include road network, infrastructure planning, traffic light sequencing, placing access lanes, avoiding congestion, forecasting traffic flow and many more.

2.2.1 Anylogic

AnyLogic is a multipurpose modeling and simulation tool that comes with a graphical user interface which allows the modeler to quickly model complex environments. AnyLogic Simulation environment provides an interactive Integrated Development Environment (IDE). AnyLogic IDE not only user friendly but has a well-structured simulation engine which enables the rapid modeling with higher precision for complicated. It also support a variety of modelling methodologies, we can develop complex hybrid models with the combination of discrete event and system dynamics. It provides a Java-based environment as well as a collection of libraries, which together contribute to the robustness of the modelling process.

The Space markup library support to define specific space markup shapes like road tracks and intersections and support agents interactions within the model. Next, logical traffic flow can create using built-in road traffic library. The vehicles are represented as agents with their physical and behavioral characteristics. In this research we are using AnyLogic simulation software for the development of proposed framework. Its rich features help us to build complex simulations robustly. We can design custom agents in AnyLogic using different paradigm whereas other simulation tools [20] don't provide the capability of using Process modeling approach.

2.2.2 Road Traffic Library

The AnyLogic road traffic library is one of the many built-in libraries provided by the AnyLogic developers. This library was designed to incorporate real traffic dynamics in the AnyLogic simulation environment. This library aids modelers by allowing them to visualize, analyze and accurately model the vehicle flow in a physical environment. With this level of support provided by the software, the simulation model can be optimized to a higher resolution, resulting in more realistic and accurate simulations. the library provides tools to analyze the developed model so that results can be easily obtained pertaining to the domain of traffic dynamics. Density maps, vehicle counters, and vehicle flows are some of the many analysis tools offered by this library. The modeling blocks provided by this library enable a smooth and quick model development process. Designated traffic flow blocks offer detailed control of the agents at each stage of the model.

Chapter 3

Literature Review

This chapter documents the contributions and achievements of the latest secure update methodologies provided in the existing literature. Moreover, it contributes to understand development in the area of research.

3.1 Area of research

A detailed literature review has been conducted to identify studies related to existing update methods. Studies show that there are multiple techniques available for the purpose of secure OTA updates. The provided literature review is divided into the following sub-sections as shown in Figure 4. The first section covers those articles in which researchers have developed a secure updater system for fog computing environment while second subsection purely consists of secure OTA update mechanisms. So far, only encryption techniques have been used in both types of research. In which trust management system has been created using secret keys, digital signatures and certificates.

3.2 Fog node security techniques

Trust can assist in identifying and isolating malicious node that use legitimate identities to join a network. Furthermore, trust is critical in protecting the relationship between various fog nodes with regard to keep user secrecy and certainty. Ideally, vehicles are supposed to connect to random node to avail its services. The incorporation of fog computing and trust management would help fog nodes pick the most reliable stable nodes according to their requirements and specifications in their locality. All of the coordinating fog nodes must have a certain level of trust on one another to accomplish this.

However, owing to its decentralised framework, it is tricky to implement trust management system. The key challenge with this decentralised architecture is that it makes it impossible to gather and maintain facts and actions needed for the trustworthiness of distributed fog nodes [19].

There are a variety of models based on trust that have been extensively studied in the literature [21, 22, 23]. For the estimation of trustworthiness, reputation is seen as a significant parameter. Therefore, Several frameworks adopt this approach to determine the trustworthiness of the mobile ad-hoc networks [24] accompanying the vehicular ad-hoc networks (VANET) [25]. K. Hwang et al. [26] introduced the concept of cloud trust. They recommend integrating secure data centers in which reputation mechanism use to operate virtual clusters and data retrieval.

[22] provides a trust system that protects against unwanted intrusion by using a point-based technique. Trust was used in the gateway devices to secure data transfer between two devices. But it will not guarantee the accuracy of sensor data. The authors [27] design efficient distributed trust model (EDTM) to address this limitation. Direct and recommendation trust value calculate on the basis of number of packets collected by sensor nodes. This method is useful for distinguishing various forms of attacks. However EDTM, liable to communication overhead.

[23] combines cloud and edge computing trust assessment mechanisms, resulting in a significant reduction in resource use for trust evaluation and improved IoT-cloud service performance. They used a mean trust value calculation based on detected values from the communicating nodes.This could result in network communication overhead. Offloading may maximises resource productivity while avoiding congestion and overloading [28].

Since there has been relatively small number of documented work available on fog computing trust management mechanism. The authors of [19] conducted a survey to identify existing Internet of Things security problems and challenges, and they recommend a security protocol to enhance the sharing of revocation of certification information among fog nodes. The authors in [29] propose a fog base trust framework for cloud. Trust among sensor and cloud services providers are two distinct attributes proposed in the system. [30] suggested fog middle-ware in which decentralised entropy specification is used to quantify trust among cloud and fog nodes.

Soleymani et al. [31] suggested a fuzzy trust-based model for protecting vehicular networks. Multiple numbers of security tests are conducted to guarantee the accuracy of information obtained from registered nodes. Moreover, fog layer employ to assess precision of the location of event. Based on know attributes, a Markov model is used to predict the tendency of cyber threats. The authors [32] suggest a system for dealing with malicious fog nodes in three stages. A secure load balancer monitors the fog node services and categorize them by using 2-stage Markov model. Virtual honeypot devices add in the framework to improve system adaptability. The suggested approach successfully classifies fog nodes based on the intensity and attempt ratio of threat.

3.3 Secure OTA update forwarding approaches

In [33] software Supplier, the vehicle, original equipment manufacturer all share a set of connection keys in the proposed method. Prior to any updates of software, In order to share symmetric keys, a link key is used to establish a secure connection between the SS and the vehicle. Using the shared symmetric key, the SS encrypts update and passes it to the vehicle.

To improve protection, the authors suggested sending a copy of encrypted software at least twice at random intervals. Vehicle first decrypt and installs one of the two copies of the encrypted program it receives. Propose methodology require significant bandwidth. Authors [34]proposed a stable OTA update protocol. There are vehicles, portal, Internet, and tower entities in a system. Portal is a key unit use to connect vehicles. In proposed approach, binary update partitioned in multiple segments, then applied reverse order hashing on each section of update. Before sending the hash chain to the cars, the portal encrypts each fragment. Because of the car's limited resources, the portal employs cipher-block encryption strategy. The proposed protocol protects against eavesdropping, intercepting and malicious attacks. But the propose methodology has a significant memory overhead and inefficient to DoS attacks.

All participating components in the proposed architecture of [35] work as a cluster. To avoid central management, cluster heads of each cluster linked one another through an overlay network. once the update is ready, cloud validate through signature and upload. Then update BC block is generated which contain update location on cloud and broadcast encrypted update to vehicles with private key signature. According to the findings, proposed framework outperforms the certificate-based architecture. Uptane framework used in [36], in which update structured as metadata format. Metadata includes a variety of details, such as cryptography hashes and length, that can be used to verify the validity of an image and other metadata. This uptane framework unreliable to rollback attack.

According to our comparative analysis, the use of encryption techniques may lead many other problems throughout the network. Which not only affects the update process but also increases many other security risks. Delayed message in the update process loses the usefulness of the received message. Therefore, from the current point of view, different methods have to be adopted so that these messages can be delivered to the vehicles in a timely and safe manner.

Chapter 4

Methodology

In this chapter we first discuss the proposed Traffic Simulation Framework and then propose Algorithm based on process based traffic flow and fog simulation.

Before we going into the specifications of the adopted approach, it's worth disclosing the network domain for fog computing. Figure 4.1 illustrates a

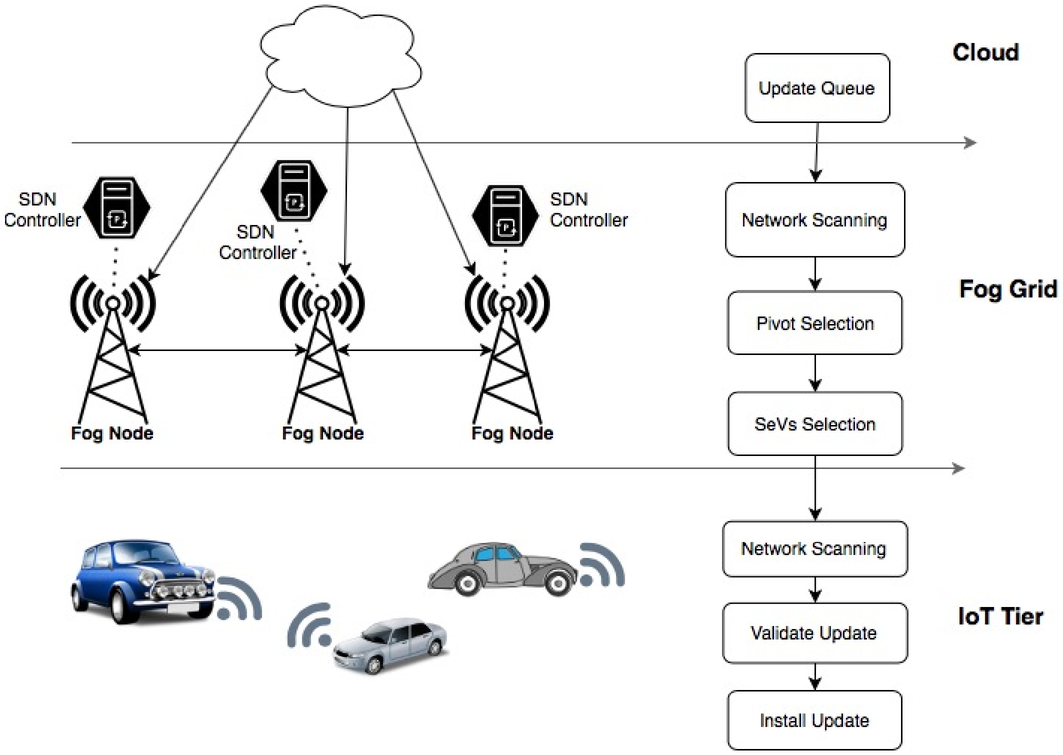


Figure 4.1: proposed layer architecture for normal traffic flow.

typical network of implementing software Define Vehicles (SDV) based on fog computing. As Vehicles receives update from the nearest fog node via existing SDN controller and data flows for fog nodes are also described by the SDN. Intended to include SDN management module to make network salable and resource aware. The purpose of adding each layer and their working principle explained in the next section. We have proposed solutions for malicious and black holes attacks on fog nodes which also ensure the integrity and availability of the fog node.

4.1 Architecture diagram

This section consists on two parts, the first half contains the details of layers while in the reaming half describes an algorithm details. In a global perspective, SDN offers stability and scalability to distributed fog enabled grids. Hence, in this study, the SDN controller provides the rules for data transmitting among the fog nodes. Figure4.1 describes the fog grid architecture based on SDN. In this architecture, forwarding will take place between any

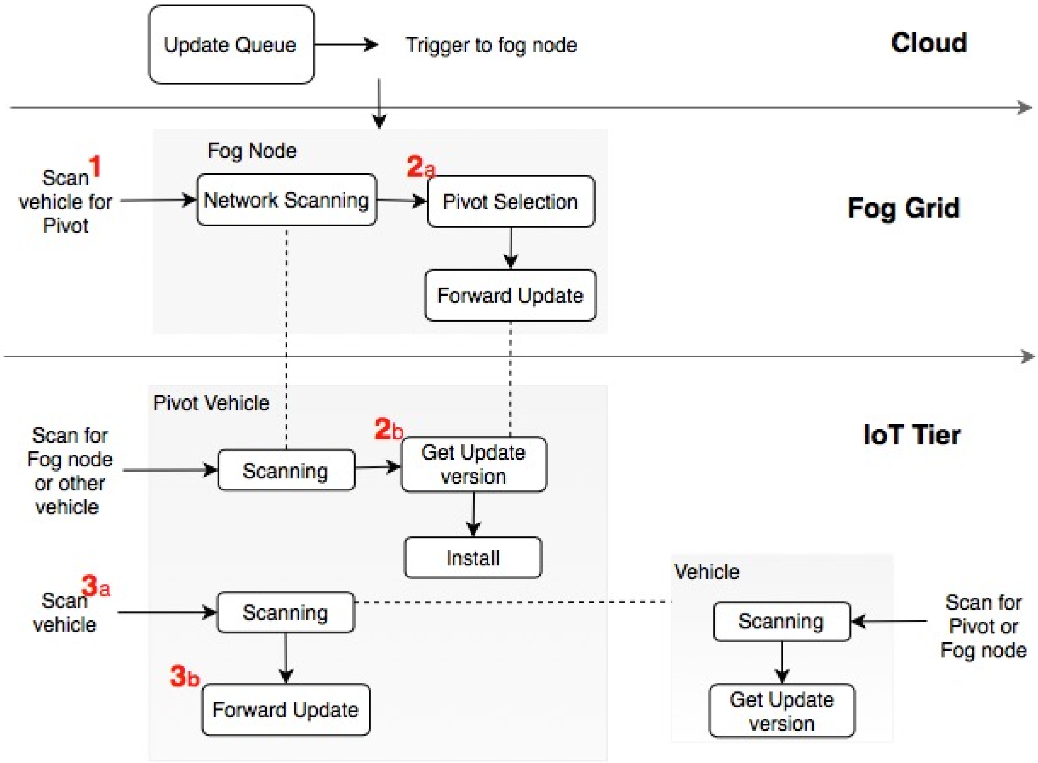


Figure 4.2: proposed layer architecture for normal traffic flow.

two nodes. Fog node data is supposed to be transmitted to vehicles and other fog nodes. Because of autonomous controller, as seen in figure, is positioned on the fog tier, view and manage locally. This fog grid hierarchy has three tiers, as seen in the details of system architecture (Figure 4.2): IoT tier, fog grid tier, and cloud tier. Each tier takes on various roles during data transfer, from permanent storage in the cloud tier to forwarding updates in the IoT tier. The IoT tier positioned, according to this hierarchy, is at the bottom and contains a large number of IoT vehicles. Each vehicle is supposed to be connected to the Fog node or another pivot vehicle. The collection unit, i.e., vehicles that are also responsible for forwarding updates, as pivot vehicles.

4.1.1 Cloud Tier:

The original equipment manufacturer launch updates to the cloud. Tier responsible to upload software updates directly to all nodes in the fog layer for further dissemination.

Update Queue:

Maintain update queue and trigger next layer when fog node receive update notification.

4.1.2 Fog Tier:

Fog tier responsible for multiple task. Because of OTA update response issues fog integrate with cloud base architecture to over come the latency issue. Despite the integration of cloud and fog, some communication barriers remain exists between fog and IoT layer layers. To solve this problem effectively, we have introduced pivots vehicles in our system which minimize the fog processing. Only pivot vehicles communicate with fog nodes and these pivots update all other vehicles existing on same layer also minimize two different layer communication overhead.

Network Scanning Layer:

Under the proposed system, layer of network scanning responsible for various tasks. Establish connection with vehicles in its range upon get new update, if vehicle is not already updated, provide an update when it enters into the rang of fog. The fog exchange update packet with vehicles that manage to establish their connection with fog during scanning. This layer also scan for neighboring fog nodes and create list.

Pivot Selection Layer:

Once the vehicle update process is complete, pivot selection layer selects some of the vehicles in such a way that they can update the other vehicles in their range. The purpose of pivot creation is to reduce burden on the entire system. Complete normal flow with pivot selection.

4.1.3 IoT Tier

Traditionally, this layer only used to produce data. But in proposed system the inclusion of pivot vehicles in the system, some responsibilities have been placed on it. Following is the update dissemination sequence which is graphically presented in 4.2
Step 1: Scanning At fog layer, fog nodes scan for vehicles
Step 2: Pivot Selection After successful pivot scanning, fog nodes forward update version to the pivot vehicle. Pivot install this update.
Step 3: Forwarding update Pivot scan for other vehicles, this scanning done on IoT layer. Pivot forward update to the vehicles after scanning.

4.2 Malicious Fog Node Behaviour Model Mechanism

SUA that comprises on two parts. One part of SUA inside fog layer, mounted on each fog node and the other part mounted on vehicle side. It responsible for providing a stable and trustworthy environment for fogs to share resources and exchange data packets. As a result, It allows the fog to make a quick decision about when it can test. The decision not only includes the best node that can handle the other fog node blackout but also has the best QoS (i.e., lower the convergence time), such as service delivery without interruption and data security assurance. Section 5 delves deeper into this subject. In order for the SUA to choose the trusted node, it must first determine if the fog communicates with its neighbouring fogs within the defined time frame as shown in Figure 4.3. The following are the key protocols and processes that SUA performs:
Fog interactions:
while testing status, SUA is in charge of assigning the closest adjacent fog that can accommodate the blackout fog. End-user demand delays must be kept to a minimum as services are processed at the fog layer. This procedure in our proposed system has manage to reduce this problem to some extent.
Update validation:
As far as the correctness of the update is concerned, we have installed a part of SUA to the vehicle side so that the vehicles themselves can get different update versions and test these update versions by using paxos consensus algorithm [37].

4.3 Black-hole behavior

In case of black hole attack Vehicles will not get update notification until reach to the active fog node range under traditional system. Following is step by step procedure to address this problem.
Step 1: Black-hole identification Active fog node identify other blackout nodes that are currently under attack by the timer base mechanism.
Step 2: Selection of SEVs After the identification, neighboring safe fog node elects successful vehicles which are moving from safe nodes to blackout nodes as SEVs.
Step 3: Update These SEVs after reaching in the range of blackout nodes will update vehicles in the same way that fog node update vehicles.

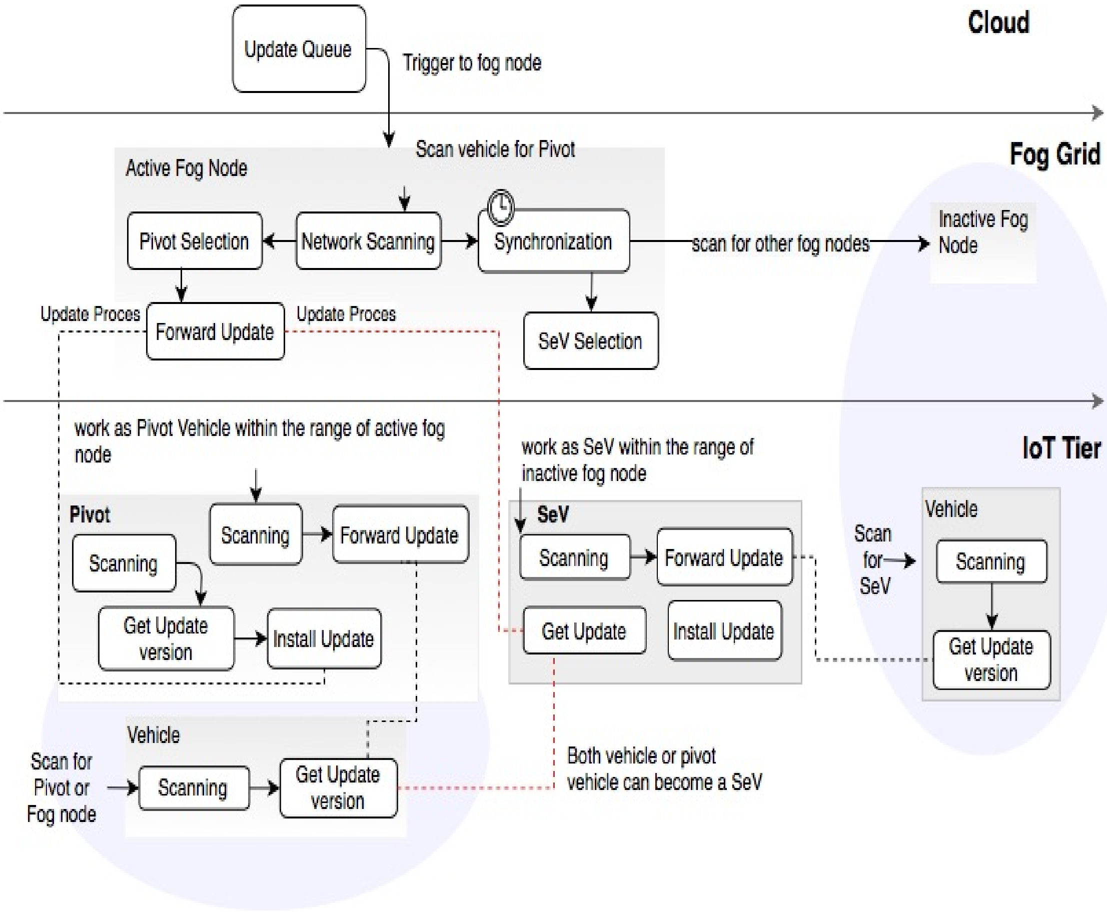


Figure 4.3: Proposed update architecture for black hole problem.

Synchronization Layer:
For black hole solution time clock feature added in the synchronization layer

which allows fog nodes to probe their neighbor fog status within a specified time and establish an implicit bond of trust within the network, thus, each node test periodically to its neighboring nodes, a list maintained by Network scanning.

SeVs Selection Layer:

After establishing trust bond among black out and active fogs nodes, on trusted fog node this layer select SeVs(Service Vehicle) for blackout fog node. Detail SeVs selection process and its function represents in Figure 4.3.

Under the random system, if the fog nodes are attacked, all passing vehicles will not receive timely updates. Vehicles will not get update notification until reach to the next fog node range. In this way, delayed updates reduce the efficiency of the system which later leads to system malfunction.

4.4 Update Validations Process

In the event of an attack of malicious update on fog nodes under the traditional system, all passing vehicles receive an older version of update. The flow of these two different update versions into the system will disrupt the system. We set up a much-sophisticated system to solve this update problem and the steps are:

Step 1: Update Checksum Vehicles are given update checksum instead of update.

Step 2: Checksum verification The vehicle compares this checksum with other fog nodes checksum it receives and install update when accurate one is found as shown in Figure 4.4.

Our proposed SDN-based fog grid architecture ordered hierarchically.Following are the key components and roles of the proposed SDN-architecture:

Control Plane:

Define propagation laws and regulate other elementary layers, such as pivot selection, SeVs selection, network scanning, synchronization and authentication. This is where the part of SUA that was built for the fog node operates.

Data Plane:

Responsible for various tasks, update circulation to neighboring vehicles and in particular extracting update status, location and direction data for further conclusions. Download the update when receive trigger from the update queue.

Application Plane:

After receiving the update, this plane validate update for which a part of SUA runs here

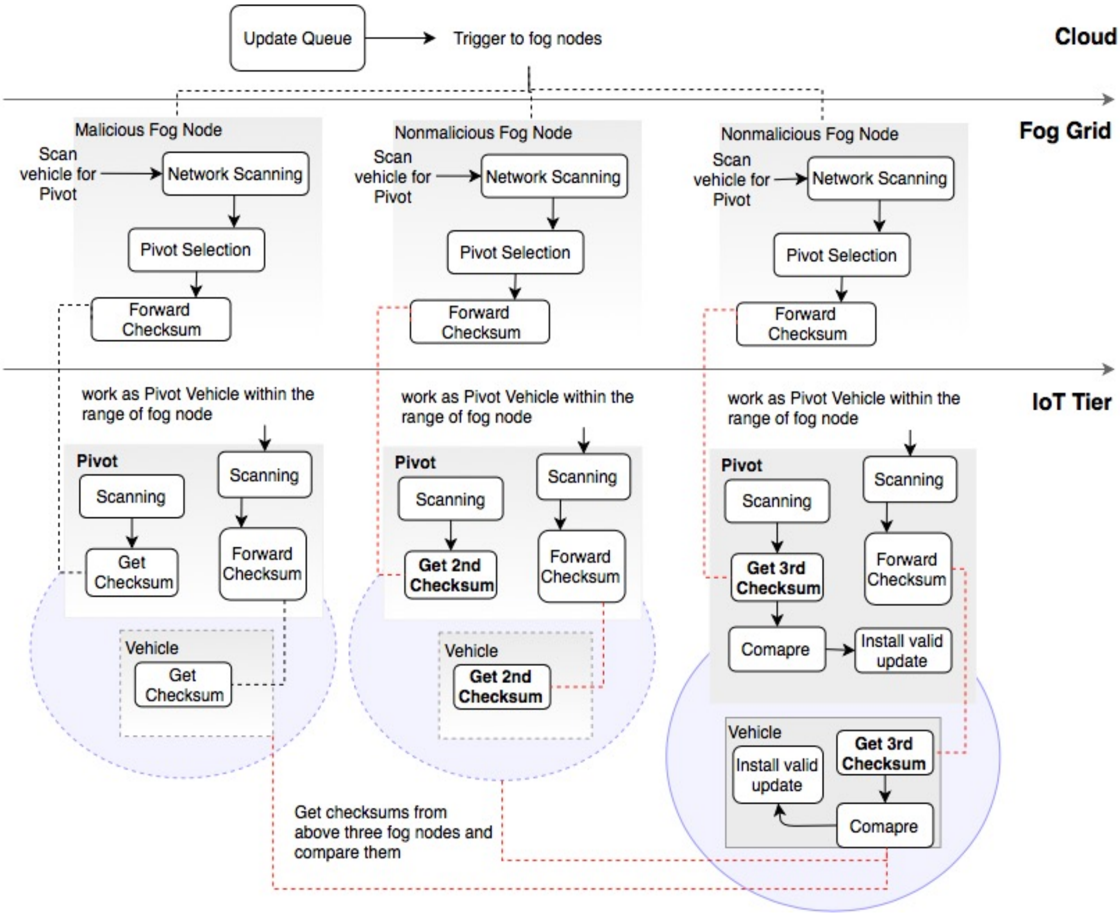


Figure 4.4: Proposed update architecture for malicious attack.

Chapter 5

Simulation and Results

In this chapter we first discuss the proposed Traffic Simulation Framework and then propose Algorithm based on process-based traffic flow and fog simulation and at the end results.

5.1 Traffic Simulation and Analysis Framework

In this section we discuss our proposed update mechanism for traffic among fog nodes, which is capable of: (i) modeling traffic flow using the Road Traffic dynamics; (ii) modeling fog grid (iii) simulating traffic flow among multiple fog nodes using different scenarios; and (iv) evaluating securing fog node strategies. This helps in creating an optimal update plan or evaluating an existing plan for effectiveness. It is composed of three layers, this layered architecture was initially introduced by Asad et al. [38] and is now extended to focus on malicious fog node behavior.

5.1.1 Building the Model Environment

In order to map the real-world environment into the simulation model, to-scale road structure loaded into the AnyLogic program. This is done by going to the pallet section, then clicking on the space markup tab, clicking and dragging the road and intersection block onto the blank work space. This will open up a pop-up window from where you will browse and select the map of the road you want to make the environment for as shown in Figure 5.1 and set properties shown in 5.2 Now the next step is to trace the fog node onto the road map we have just imported into the project work spaces. In order to achieve this, in the palette section, again go to the space markup tab and

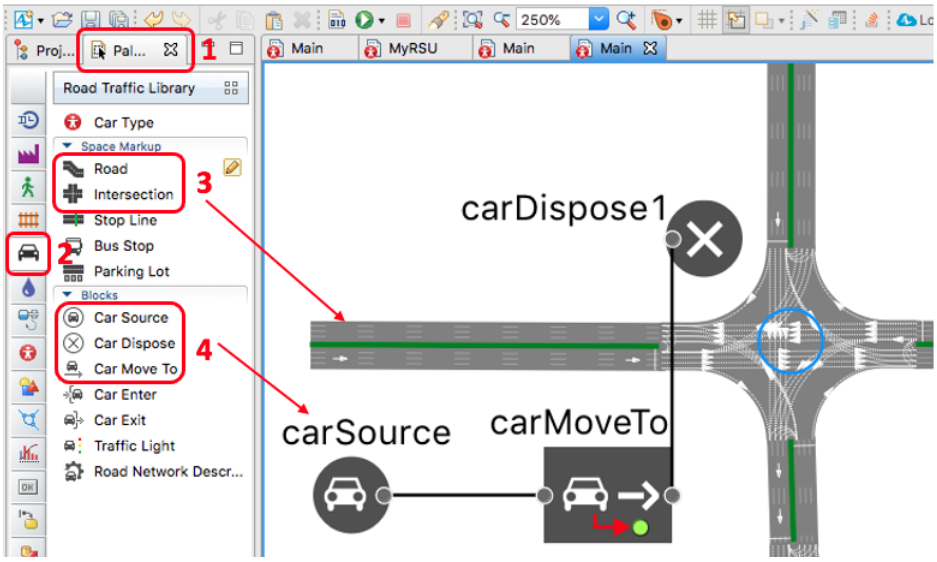
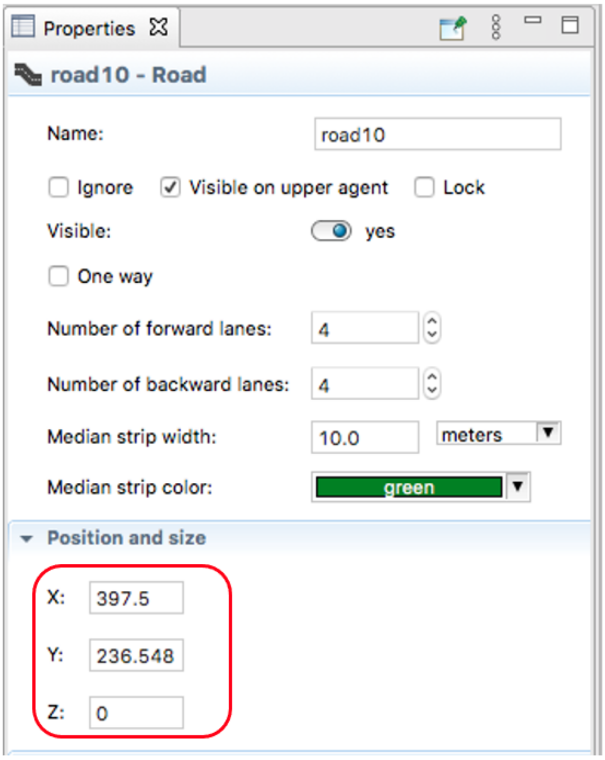


Figure 5.1: Drawing roads and intersections in Anylogic.


Properties

road10 - Road

Name: road10

□ Ignore ☑ Visible on upper agent □ Lock

Visible: yes

□ One way

Number of forward lanes: 4

Number of backward lanes: 4

Median strip width: 10.0 meters

Median strip color: green

Position and size

X: 397.5

Y: 236.548

Z: 0

Figure 5.2: Road and intersection properties in Anylogic.

double click on the point node icon as shown in 5.3. Fog node settings may change according to the from properties tab as in Figure 5.4 node tracing process and its properties tab. This will enable to add the fog node position on road map, we have add nine agent for fog nodes, for their positions import nine point nodes and build their network for communication. Now go to

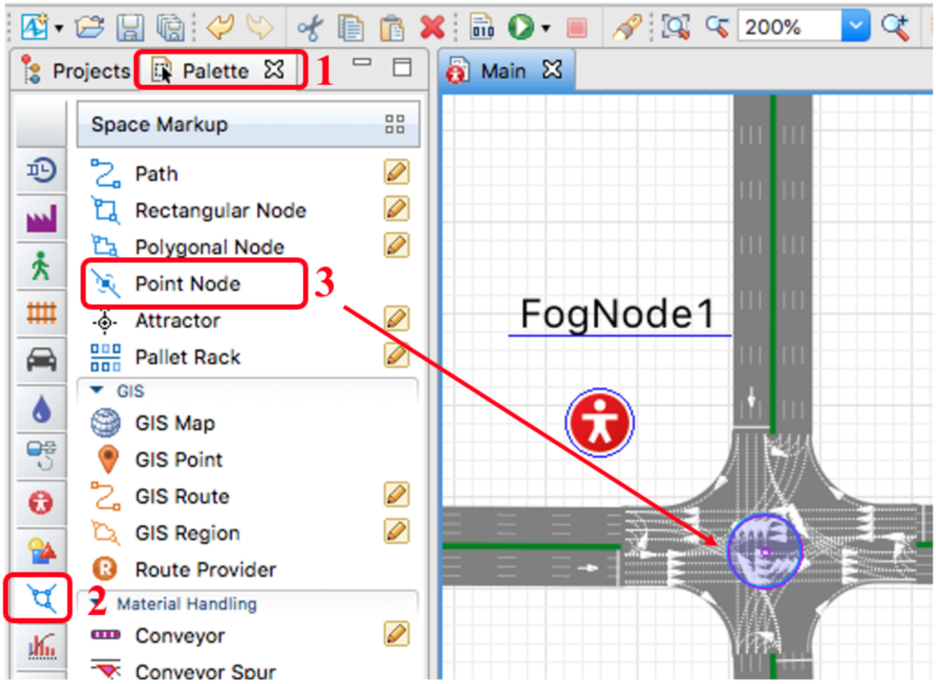
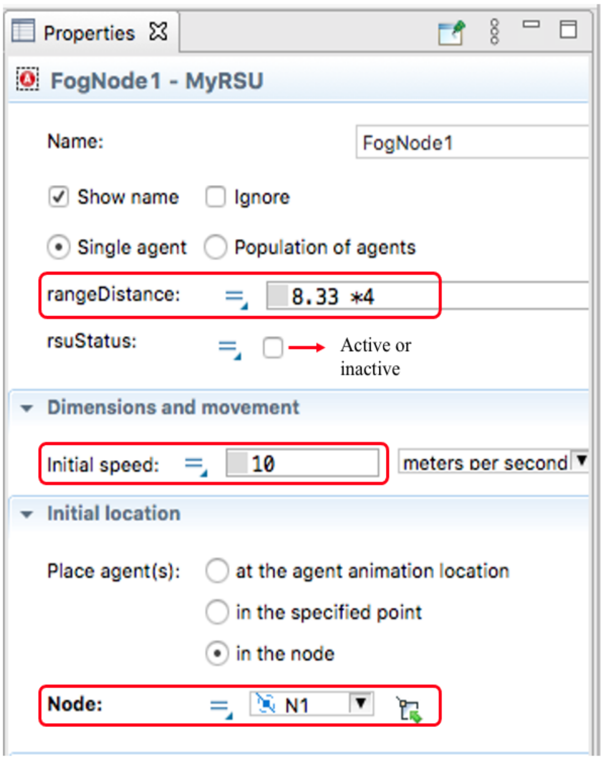


Figure 5.3: Tracing Node structure for Fog node location and its properties in Anylogic.

your Road and select a road to trace. Click on the starting point of the road and move the cursor over the straight path of road and double click at the finishing point of the road where you want to stop. Once you double click, the road will appear with default settings. The properties tab on the right-hand side of the program will open automatically from where you can adjust the settings of the drawn road as per requirements. After tracing all the road, the Fog nodes are also placed on the intersections in a similar manner. The end result will be similar to show in Figure 5.5. The figure shows that there are eight traffic entry points and two way traffic passes on all roads. And there are nine fog node agents and two types of nine corresponding node points: (a) five densely populated nodes named as overloaded node in result section, (b) four had less traffic flow named as under loaded node. The purpose of this distinction of fog nodes is currently to analyze the update pattern of vehicles over the traffic rate, however later it will be used to define the microscopic



Properties

FogNode1 - MyRSU

Name: FogNode1

☑ Show name ☐ Ignore

● Single agent ○ Population of agents

rangeDistance: 8.33 *4

rsuStatus: ☐ → Active or inactive

Dimensions and movement

Initial speed: 10 meters per second

Initial location

Place agent(s): ○ at the agent animation location
○ in the specified point
● in the node

Node: N1

Figure 5.4: Node location and its properties in Anylogic.

behavior of fog agent. In future the aim is to implement the microscopic behavior of different types of agents to address further complexities in the update process. Now the next step is to trace the traffic flow onto the map of the road we have just imported into the project work space. In order to achieve this, in the palette section, go to the Road Traffic library tab and double click on the carSource icon as shown in Figure 5.1.

For vehicle movement modelling, the library provides extensive and highly efficient physical-level assistance. It can be used to simulate highway traffic or some other that includes cars, roads, and lanes. We often use driving behaviour to monitor vehicle direction and speed. The action of the car object and its relationships were thus established using CarSource, CarMoveTo, CarDispose and Select blocks. The CarSource creates cars and places them at road entry points and set its properties in Figure 5.7. Arrivals of cars are determined by their arrival rate, which ranges from 100 to 500 in order to measure the propose system efficiency. CarDispose is used to remove car from the model, and car movements control by CarMoveTo block, from entry to CarDispose block. For each car entering the road and changing lanes, we use the RoadNetworkDescriptor block.

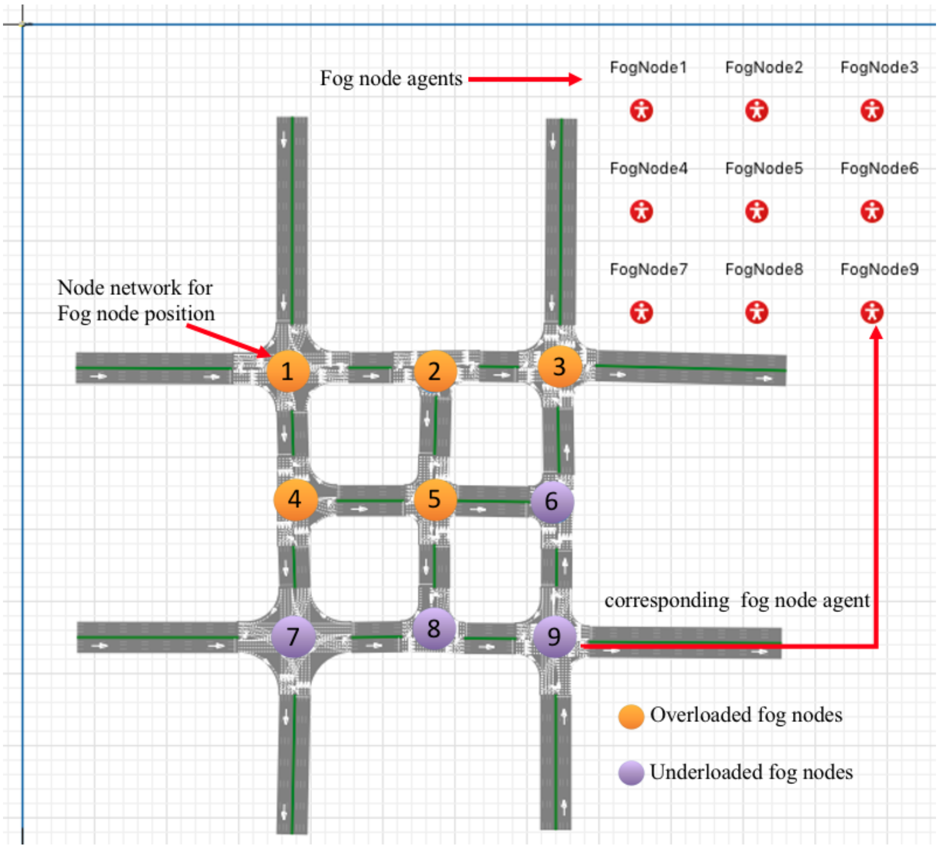


Figure 5.5: Roads and intersection properties in Anylogic.

Figure 5.6 depicts the traffic distributed amongst all the available directions and also represent the probabilities output blocks. The Select block uses a given probability distribution to evaluate car turning directions. This can be achieved by simply assigning a different probability according to the proportion of high and low congestion to all the exits. In our simulation we use three selectOutput blocks one block use for overloaded fog nodes with high traffic probabilities while remain two have half of the probability of first block. Lastly, after completing tasks all vehicles exit through exit block.

5.1.2 Fog Layer

This layer comprises of three main components: (i) Spatial Environment, (ii) Process Modeling and (iii) Road traffic Dynamics. The main responsibility of the fog layer is to ease the process of modeling road network and fog grid. The Space markup library within AnyLogic offers the necessary elements required for recreating a scaled graphical model of the spatial environment

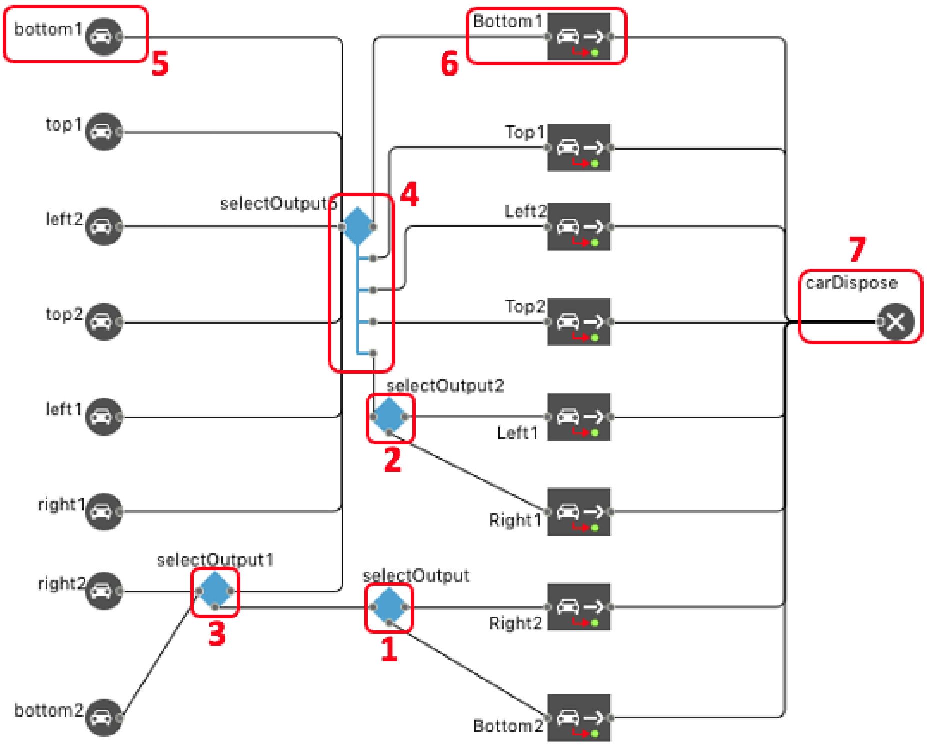


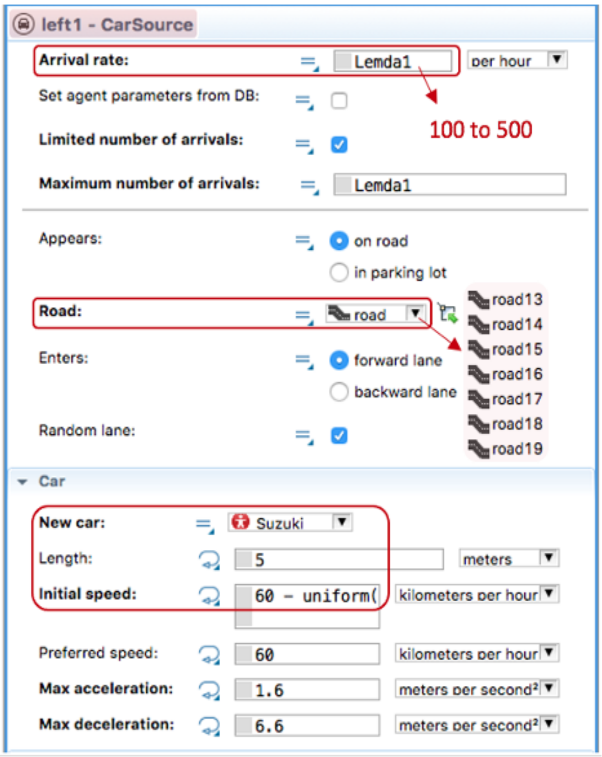
Figure 5.6: Proposed Fog node placement and road model.

in which the agents will reside within the model [39]. In our proposed model, the vehicle range diameter is taken to be 40 meters while for fog use 33 meters. We assume the vehicle speed to be 60 km/hrs .

5.1.3 Simulation Layer

The simulation layer couples the fog layer with the simulation environment. In this layer the modelers can create different simulation scenarios. It uses traffic simulation techniques to model the different patterns of a traffic and simulate it within the modeled traffic environment. It allows the modelers to control the flow of the traffic and direct the flows toward fog node. It also allows to segregate traffic into various directions and manage their update process. Figure 7 shows the basic traffic flow diagram developed in the simulation layer. Later we will discuss how intelligent algorithms can be implemented, that assess the overall situation and make effective decisions.

A traffic control flow using the simulation layer is developed. A total of forty scenarios were run With 100 to 500 vehicle arrival rates to evaluate the performance of our algorithm. Following are the details of each scenario



left1 - CarSource

Arrival rate: Lemda1 per hour

100 to 500

Set agent parameters from DB: □

Limited number of arrivals: ☑

Maximum number of arrivals: Lemda1

Appears: ● on road ○ in parking lot

Road: road

road13
road14
road15
road16
road17
road18
road19

Enters: ● forward lane ○ backward lane

Random lane: ☑

Car

New car: Suzuki

Length: 5 meters

Initial speed: 60 - uniform(kilometers per hour

Preferred speed: 60 kilometers per hour

Max acceleration: 1.6 meters per second^2

Max deceleration: 6.6 meters per second^2

Figure 5.7: Proposed Fog node placement and road model.

and its comparison with Random update dissemination process. In case of black hole attack the next four cases execute for each arrival rate (i)with one overloaded blackout node (ii) with two overloaded blackout nodes (iii) with one under loaded blackout node (iv) with two under loaded blackout nodes. These four cases also execute for malicious fog nodes with each arrival rate.

The Random update dissemination process is considered a worst-case scenario in any OTA update activity because there is no threat/attack mitigation plan and no predefined update validation mechanisms. The vehicles in random update dissemination are randomly get their update (based on uniform dissemination) and may lead to delayed and malicious update.

5.2 Results

We measured the efficiency of our proposed work in the face of a black-hole and malicious threat. The benchmark is run with an increasing vehicle arrival rate; however, 10 to 20 percent of fog nodes undergo a black hole and malicious attacks. In case of black hole attack affected fog nodes do not forward the updates to the vehicles while disseminate older version in case of malicious attack in any simulation run. The primary purpose of blackout nodes is to avoid the distribution of alerts. Because of older and newer update version dissemination make system inconsistent. Proposed approach analyze with network convergence time, connection overhead in terms of failure ratio and efficiency in terms of success and failure. To evaluate the performance of the system, first one and then two fog nodes were attacked. To draw conclusion and understand the nature of both malicious and black hole attack, simulation run in three sections with different conditions. After running these sections, data is produced. All the graphs below are design using this data. There are three lines in the line charts represented the following flows.

Normal Flow: Vehicles and fog nodes work in the same way as we set up the system in Figure 5.8.

Random Flow: worst scenario When fog nodes were attacked in the proposed system without threat mitigation system.

Proposed Flow: It shows the flow when fog nodes were attacked as well as proposed solution run simultaneously. **Average update time –** is a time require to update vehicles entered into the simulation. There are four line graphs use to represent the blackhole scenario, two graphs for overloaded and two for underloaded fog node structures. One of two overloaded/underloaded plot represent the single blackout fog node performance and the other one represents the double blackout fog nodes behavior. Arrival rate from 100 to

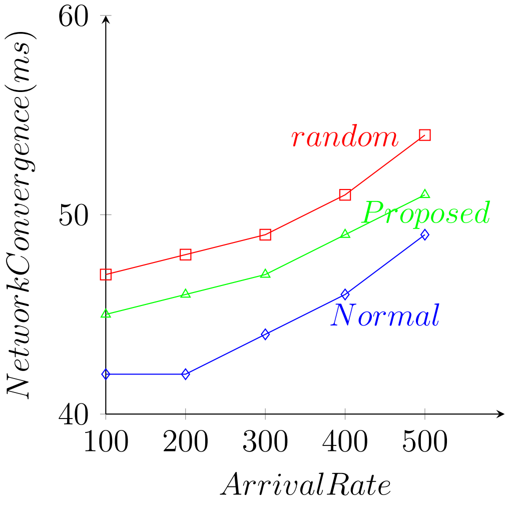


Figure 5.8: single over-loaded Fog Node Average update time trend with normal, random and proposed approaches

500 on x-axis and average update time of entire simulation in milliseconds represents on y-axis.

The 5.8 line graph illustrates the update rates in Normal, Random and Proposed update dissemination schemes from 100 to 500 vehicle arrival rates with one overloaded black out fog node. The update rate of normal, random and proposed showed a steady but significant rise over the arrival rate. At 100 arrival rate the update rates of normal, random and proposed strategies are nearly 42 ms, 45 ms and 47 ms respectively. random flow steadily increases as arrival rate increases. After 200 arrival rate the update time of normal traffic flow increased sharply while proposed varies between normal and random update time, from 100 to 300 arrival rate closer to the normal update time whereas at 400 and 500 near to random.

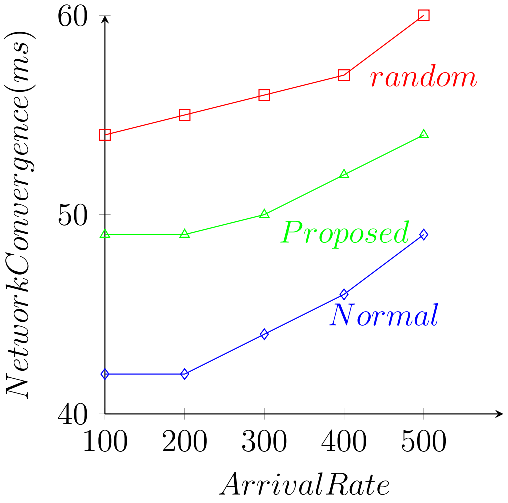


Figure 5.9: double over-loaded Fog Node Average update time trend with normal, random and proposed approaches

Above stated triad update dissemination trends with two overloaded black out fog nodes shown in 5.9 line graph. Green line reveals the stable rise of proposed system, remaining both random and normal showed a rapid but notable rise over the arrival rate. According to the chart at 100 arrival rate, proposed and random update time far higher than normal. This line chart shows that two blackout fog nodes not only affects the random system but the proposed system. Initially at 100 arrival rate proposed line lies below to the random line with 5msec difference as can be seen in the graph because of additional 7msec to the normal point, this reaches above the normal line. After 300 its effect begins to reduce and update time of proposed line becomes closer to the normal line, while the line of the random system continues to move upwards.

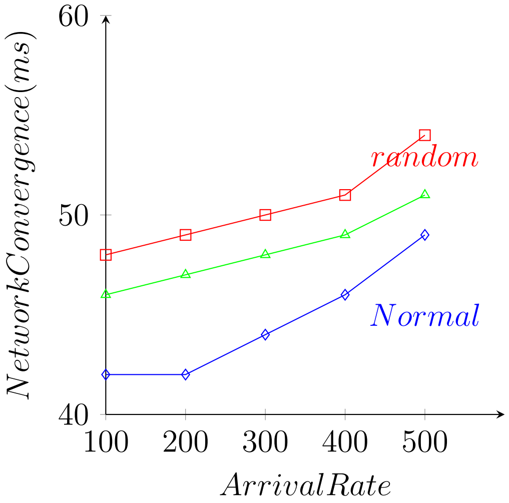


Figure 5.10: Under-loaded single Fog Node Average update time trend with normal, random and proposed approaches

The 5.10 and 5.11 line graphs gives information about the update rates of triad flows from 100 to 500 vehicle arrival rates with single and double underloaded black out fog nodes respectively. In both these plots the update rate of normal, random and proposed experienced a dire rise over the arrival rate.

At the arrival rate 100, the normal lies on 42, proposed on 46 and little higher random lies on 48 in the graph in case of single fog node. Due to the two blackout fog nodes, just as random and proposed trends in the overloaded scenario jump to the higher in the graph. In the same way, these two lines jump at 54 msec and 52 msec respectively. As can be seen clearly proposed avg update time become closer to the normal in plot 5.10 after 400 arrival rate and after 200 arrival rate in plot 5.11. While random moves upward in both scenarios.

To be concluded, it is said that proposed system works well even if the severity of the problem is increased, it does not reduce the efficiency of the system. And all the updates reached to the vehicles on time. The performance of proposed system also showed some changes i.e. the network avg update time of the proposed system changes perpendicularly as the number of blackout nodes increases. This is because the chance of SeVs generation declines as the number of blackout fog nodes increases. The propagation of message is managed by the proposed system, whereas the next fog node requires the vehicles to move in the random network in case of blackout.

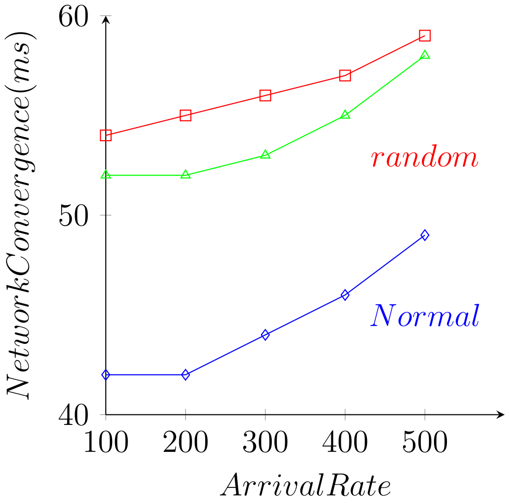


Figure 5.11: Under-loaded double Fog Node Average update time trend with normal, random and proposed approaches

This was a comparative analysis of normal, random and proposed simulation flows in densely populated fog nodes. The less densely populated fog nodes show slightly different trends, especially proposed line trend in the graph. The reason why the proposed line in both underloaded graphs is close to the random line is due to the low traffic congestion on the fog nodes. Due to the low number of vehicles, the count of the SeVs also decreases while the proposed system dependents on SeVs. The fewer SeVs, the more vehicles receive late update and that will greatly increase the avg update time of the entire system. However, the average update time is not long enough to reduce the whole system efficiency. Under the proposed system, each signal waits for three milliseconds for the status of its neighboring fog nodes. This waiting time decreases significantly after the initial wait at the start of the simulation, and the vehicles in the range of blackout nodes begin receiving quick updates, as seen in the graphs above. Like the black hole, this malicious framework also has four graphs, four of which are densely populated while four graphs show the results of non-populated fog nodes. Here too the simulation run first with a single fog node and then with two fog nodes. In order to conclude the graph results from malicious attack, it is necessary to repeat the system proposed for the nature of this attack. In random system malicious fog node forward a malicious update (we assume to forward an older update version), it makes the entire system unpredictable due to the circulation of different update versions.

To mitigate this inconsistency in our proposed model, vehicle get update

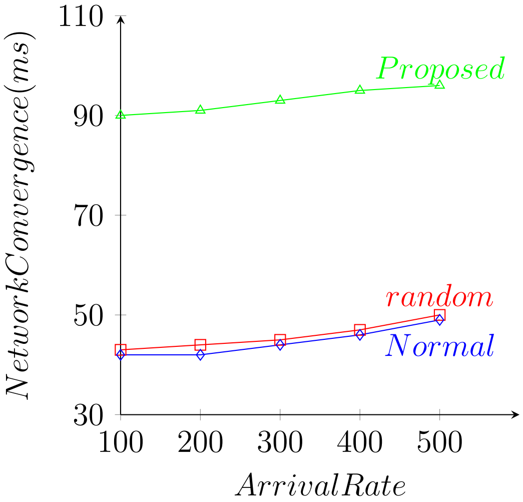


Figure 5.12: over-loaded single Fog Node Average update time trend with normal, random and proposed approaches

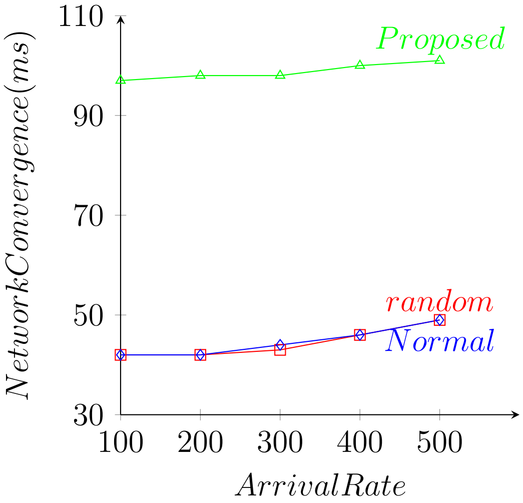


Figure 5.13: Over-loaded double Fog Node Average update time trend with normal, random and proposed approaches

checksum from at least three fog nodes and find the valid update version. Because in this situation, the accuracy of the update is more important than the timely receipt. So, you can see the avg update time of the proposed system is too high, the time recorded in the proposed flow is the time when the vehicle learns about the valid update after passing through at least two or three fog nodes. The proposed line fluctuate between 90 to 95 milliseconds in following eight 5.12 to 5.15 line graphs.

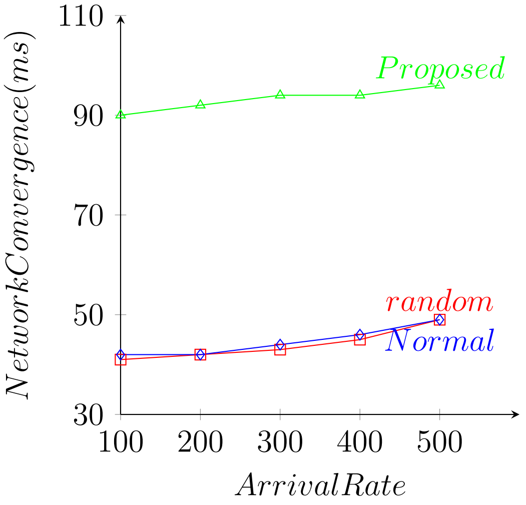


Figure 5.14: Under-loaded single Fog Node Average update time trend with normal, random and proposed approaches

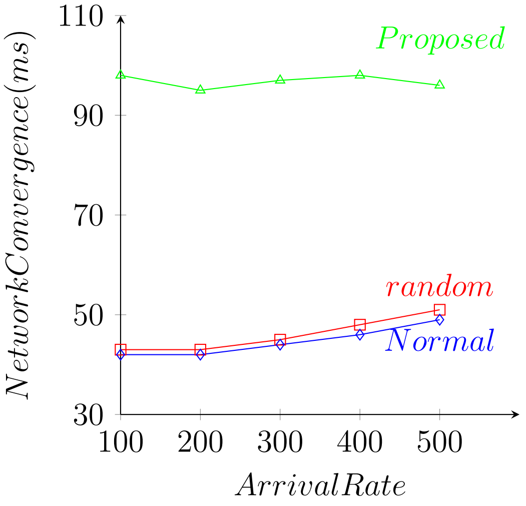


Figure 5.15: Under-loaded double Fog Node Average update time trend with normal, random and proposed approaches

Efficiency - is a system performance measured by the available vehicles to the valid updated vehicles.

The bar chart 5.16 and 5.17 provides information about the efficiency rate in Random system and proposed system with one and two blackout fog nodes

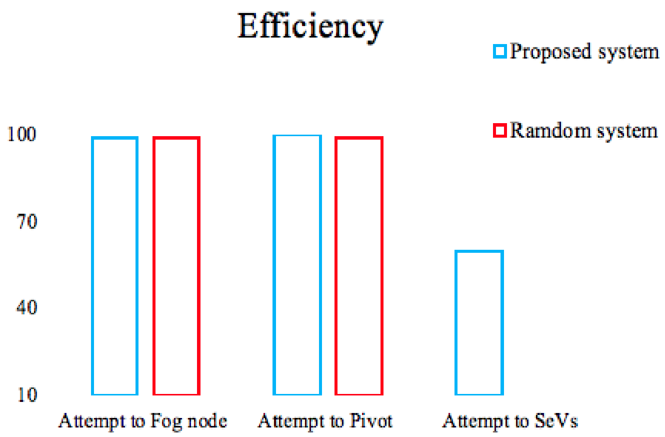


Figure 5.16: Efficiency trend with 1 underloaded blackout fog node

respectively. Overall, both random and proposed system experienced a same trend except for the proportion of SeVs attempts. An attempt efficiency of SeVs as a result of one blackout fog node is better than two blackout fog nodes. Because the random system did not have a clear method to deal with an attack, therefore only the proposed system has a trend representation for SeVs. As a result of the attack on more fog nodes, the number of active fog nodes reduced that provide SeVs for the blackout fog nodes that inactive as a result of the black hole attacks. The 100 percent success rate of pivots and fog nodes attempt in above mentioned bar charts indicate that the failure rate of vehicles in both the systems is too low.

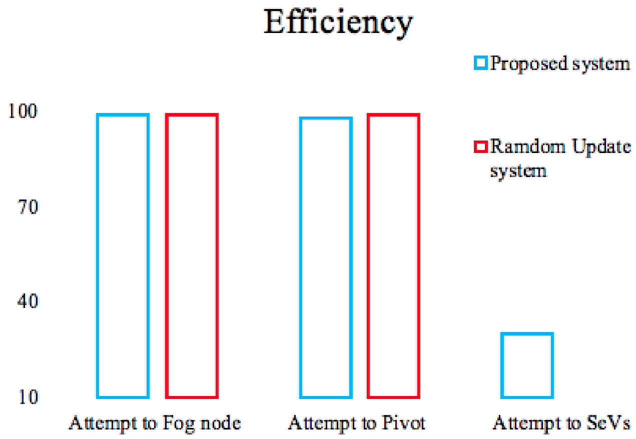


Figure 5.17: Efficiency trend with 2 underloaded blackout fog nodes

The bar 5.18 chart shows about the overall system efficiency rate in Random system and proposed system with one and two blackout fog nodes. On the whole, random system experienced a downward trend, while proposed showed an upward trend throughout the period. Proposed efficiency rate is 90 percent with one blackout fog node, being higher than random rate by approximately 10 percent. Then it remains same with 2 blackout fog nodes. The 5.18 bar chart shows about the system efficiency rate in terns of vehicles attempt with fog node and pivots for Random system and proposed system. Generally, system fluctuation between 70 to 100. For fog node attempt both systems show same trend. In case of pivot attempt efficiency drops 20 percent for random system. In the case of the proposed system, the efficiency is reduced by at least 30 percent. The reason for the 10 percent reduction in the performance of the vehicles attempts to pivots from the random system is that the all vehicles in proposed system request update with at least three time, which increases the connection failure due to repeated connections of all vehicles with other pivot vehicles and This includes the validation process overhead that performed on the side of the vehicle to determine the accurate update.

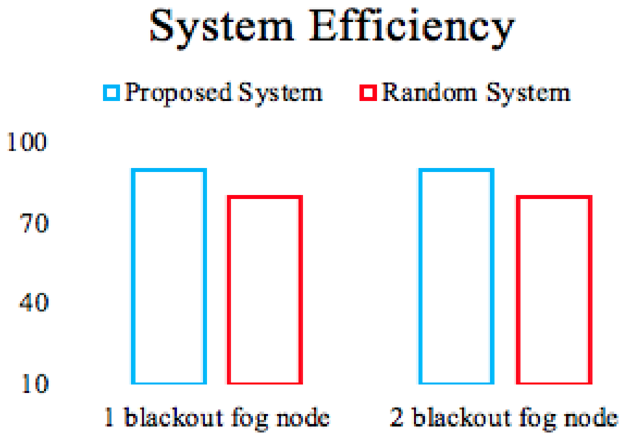


Figure 5.18: Entire System Efficiency comparison bar chart of random and proposed system for black-hole attack

Bar chart 5.18 shows the efficiency rates in random and proposed approaches. Overall, proposed system experienced an upward trend, while random system showed a downward trend. random system's efficiency rate had some fluctuations. Although random system has higher efficiency rate when single fog node experiences a malicious attack, it reduces as two fog node get malicious.

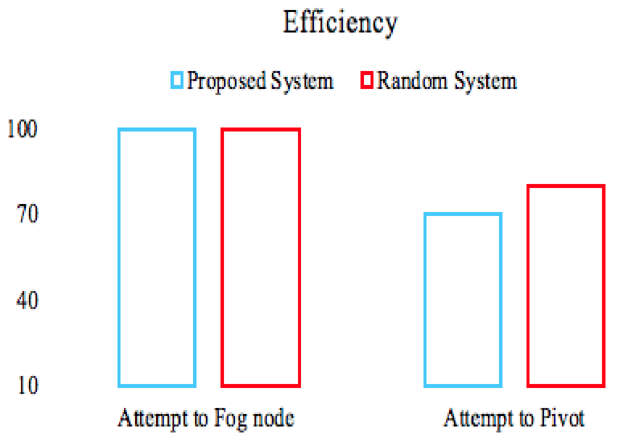


Figure 5.19: Efficiency trend of random and proposed system for malicious attack

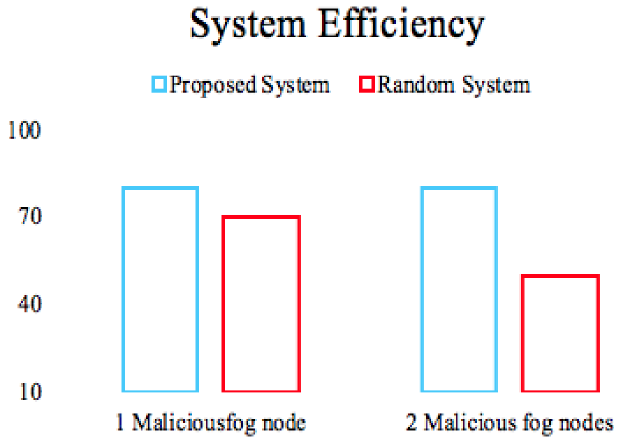


Figure 5.20: Entire System Efficiency comparison bar chart of random and proposed system for malicious attack

The bar 5.20 chart shows about the overall system efficiency rate in Random system and proposed system with one and two malicious fog nodes. Generally, random system experienced a downward trend, while proposed showed an upward trend throughout the simulation run. Proposed efficiency rate is 80 percent with one blackout fog node, being higher than random rate by approximately 10 percent. Proposed system efficiency remain same with 2 malicious fog nodes while random system efficiency further decreases by 20 percent. However, the figure showed a gradual decrease in random efficiency, reaching at 50 percent with 2 malicious nodes. This means that half of the vehicles in random system become malicious. In proposed system the number of malicious vehicles is reduced to only 20 percent, these 20 percent are vehicles that cannot verify the accuracy of update from the second or third fog node and exit from the simulation at first fog node.

Chapter 6

Conclusion and Future Work

This chapter provides the conclusion and future work of the thesis.

6.1 Conclusion

In this research, a traffic simulation and Analysis framework is proposed for the secure OTA update using simulation, visualization, analysis and the optimization of large traffic flow with fog computing architecture. Furthermore, an algorithm is proposed called: Secure Update Dissemination Algorithm (SUA), in order to devise a dissemination strategy, that not only forward valid update to vehicles but also load balances on fog layer by using third layer (IoT Devices layer) based strategies. The functionality of our algorithm is demonstrated using a simplistic example and apply the algorithm using fours scenarios. The algorithm outperforms the random and scales up when the traffic congestion is increased. The focus of this research is to propose a secure update strategy using SUA algorithm and apply it in a real world setting to demonstrate effective OTA update using road traffic simulation framework.

We extensively presented our suggested architecture for identifying malicious fog nodes in a distributed fog computing system.The proposed framework uses the timer base technique for malicious fog node identification. Results of the experiments show the effectiveness of our frame-work with supporting test outcomes. A key point of the proposed framework is the minimal overhead of proposed strategy in which function added to the IoT layer instead of fog layer. Which is why vehicles provide updates to other vehicles and validate themselves. The simulation framework will help develop strategies/policies for effective OTA update dissemination plans. It can also be used to analyze existing plans and identify their performance. The proposed algorithm can further be extended for complex scenarios where priority base update dissemination is needed.

6.2 Future Work

There are several potential directions for our future study. For example, how might various threats be carried out collaboratively, include other security threats to ensure our system is robust and resilient enough against attacks and may also integrate with large scale network.

Bibliography

[1] Akamai, “Space markup palette,” *[Online] Available:https://www.akamai.com*, 2019.

[2] Z.El-Rewini, K. Sadatsharan, D. Selvaraj, S. Plathottam, and P.Ranganathan, ““cybersecurity challenges in vehicular communications”,” *Vehicular Communications Volume 23*, p. 100214, 2020.

[3] M. Steger, A. Dorri, S. S. Kanhere, K. Römer, R. Jurdak, and M. Karner, ““secure wireless automotive software updates using blockchains: A proof of concept”,” *Advanced Microsystems for Automotive Applications 2017, Springer*, p. 137–149, 2018.

[4] M. Khurram, H. Kumar, A. Chandak, V. Sarwade, N. Arora, and T. Quach, ““enhancing connected car adoption: Security and over the air update framework”,” *IEEE 3rd World Forum on Internet of Things (WF-IoT)*, p. 194–198, 2016.

[5] Z. E. Ahmed, R. Saeed, and A. Mukherjee, “Challenges and opportunities in vehicular cloud computing, in cloud security: Concepts, methodologies, tools, and applications,” *IGI Global*, p. 2168–2185, 2019.

[6] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, ““fog computing and its role in the internet of things”,” *MCC workshop on Mobile cloud computing, ACM*, pp. 13–16, 2012.

[7] S. Yi, C. Li, and Q. Li, ““a survey of fog computing: concepts, applications and issues”,” *workshop on mobile big data, ACM*, pp. 37–42, 2015.

[8] K. Fizza, N. Auluck, O. Rana, and L. Bittencourt, ““pashe: Privacy aware scheduling in a heterogeneous fog environment”,” *IEEE 6th International Conference Future Internet of Things and Cloud (FiCloud)*, pp. 333–340, 2018.

[9] R. Roman, J. Lopez, and M. Mambo, ““mobile edge computing fog et al.: A survey and analysis of security threats and challenges”,” *Future Generation Computer Systems 78*, pp. 680–698, 2018.

[10] M. R. Palattella, R. Soua, A. majid Khelil, and T. Engel, ““fog computing as the key for seamless connectivity handover in future vehicular networks”,” *34th ACM/SIGAPP Symposium on Applied Computing, ACM*, p. 1996–2000, 2019.

[11] M. Al-khafajiy, T. Baker, H. Libawy, Z. Maamar, M. Aloqaily, and Y. Jararweh, ““improving fog computing performance via fog-2-fog collaboration”,” *Future Gener. Comput. Syst. 100*, p. 266–280, 2019.

[12] D. Puthal, R. Ranjan, A. Nanda, P. Nanda, P. Jayaraman, and A. Zomaya, ““secure authentication and load balancing of distributed edge datacenters”,” *J. Parallel Distrib. Comput. 124*, p. 60–69, 2019.

[13] M. Al-khafajiy, T. Baker, A. Waraich, D. Al-Jumeily, and A. Hussain, ““iot-fog optimal workload via fog offloading,” *IEEE/ACM International Conference on Utility and Cloud Computing Companion”*, p. 359–364, 2018.

[14] X. Wang, L. Yang, X. Xie, J. Jin, and M. Deen, ““a cloud-edge computing framework for cyber-physical-social services”,” *IEEE Commun. Mag. 55 (11)*, p. 80–85, 2017.

[15] D. Puthal, S. Mohanty, S. Bhavake, G. Morgan, and R. Ranjan, ““fog computing security challenges and future directions [energy and security]”,” *IEEE Consum. Electron. Mag. 8 (3)*, p. 92–96, 2019.

[16] D. Puthal, S. Nepal, R. Ranjan, and J. Chen, ““threats to networking cloud and edge datacenters in the internet of things”,” *IEEE Cloud Comput. 3 (3)*, p. 64–71, 2016.

[17] D. Puthal, R. Ranjan, A. Nanda, P. Nanda, P. Jayaraman, and A. Zomaya, ““ secure authentication and load balancing of distributed edge datacenters”,” *J. Parallel Distrib. Comput. 124*, pp. 60–69, 2019.

[18] R. Chen, J. Guo, and F. Bao, ““trust management for soa-based iot and its application to service composition”,” *IEEE Trans. Serv. Comput. 30 (9)*, pp. 482 – 495, 2014.

[19] J. Ni, K. Zhang, and S. S. X. Lin, ““ securing fog computing for internet of things applications: Challenges and solutions”,” *IEEE Commun. Surv. Tutor. 20 (1)*, p. 601–628, 2018.

[20] B. Ahmed, A. W. Malik, T. Hafeez, and N. Ahmed, ““services and simulation frameworks for vehicular cloud computing: a contemporary survey”,” *EURASIP Journal on Wireless Communications and Networking (2019)*, 2019.

[21] L. Galluccio, S. Milardo, and G. Morabito, ““ sdn-wise: Design, prototyping and experimentation of a stateful sdn solution for wireless sensor networks”,” *IEEE Conference on Computer Communications, INFOCOM.*, p. 513–521, 2015.

[22] M. Henze, R. Hummen, R. Matzutt, and K. Wehrle, ““a trust point-based security architecture for sensor data in the cloud”,” *Trusted Cloud Computing*, p. 77–106, 2014.

[23] T. Wang, G. Zhang, A. Liu, M. Bhuiyan, and Q. Jin, ““a secure iot service architecture with an efficient balance dynamics based on cloud and edge computing “,” *IEEE Internet of Things J.*, pp. 4831–4843, 2019.

[24] J. Cho, A. Swami, and R. Chen, ““a survey on trust management for mobile ad hoc networks”,” *IEEE Commun. Surv. Tutor. 13 (4)*, p. 562–583, 2010.

[25] Q. Li, A. Malip, K. Martin, S.-L. Ng, and J. Zhang, ““a reputation-based an- nouncement scheme for vanets”,” *IEEE Trans. Veh. Technol. 61 (9)*, p. 4095–4108, 2012.

[26] K. Hwang, S. Kulkareni, and Y. Hu, ““ cloud security with virtualized defense and reputation-based trust mangement”,” *International Conference on Dependable, Autonomic and Secure Computing, IEEE*, p. 717–722, 2009.

[27] J. Jiang, G. Han, F. Wang, L. Shu, and M. Guizani, ““ an efficient distributed trust model for wireless sensor networks”,” *IEEE Trans. Parallel Distrib. Syst. 26 (5)*, p. 1228–1237, 2015.

[28] Q. Fan and N. Ansari, “Towards workload balancing in fog computing empowered iot,” *IEEE Transactions on Network Science and Engineering*, pp. 253 – 262, 2020.

[29] T. Wang, G. Zhang, M. Bhuiyan, A. Liu, W. Jia, and M. Xie, ““ a novel trust mechanism based on fog computing in sensor–cloud system”,” *Future Generation Computer Systems 109*, p. 573–582, 2020.

[30] A. Elmisery, S. Rho, and D. Botvich, "" a fog based middleware for automated compliance with oecd privacy principles in internet of healthcare things"," *IEEE Access 4*, p. 8418–8441, 2016.

[31] S. Soleymani, A. Abdullah, M. Zareei, M. Anisi, C. Vargas-Rosales, M. Khan, and S. Goudarzi, "" a secure trust model based on fuzzy logic in vehicular ad hoc networks with fog computing"," *IEEE Access 5*, p. 15619–15629, 2017.

[32] A. Singh, R. Sandhu, S. K. Sood, and V. Chang, ""a cybersecurity framework to identify malicious edge device in fog computing and cloud-of-things environments"," *computers security 74*, pp. 340–352, 2018.

[33] S. Mahmud, S. Shanker, and I. Hossain, "" secure software upload in an intelligent vehicle via wireless communication links "," *IEEE Intelligent Vehicles Sym posium*, p. 588–593, 2005.

[34] D. Nilsson and U. Larson, ""secure firmware updates over the air in intelligent vehicles"," *International Conference on Communications Workshops - IEEE*, p. 380–384, 2008.

[35] M. Steger, A. Dorri, S. Kanhere, K. Romer, R. Jurdak, and M. Karner, ""secure wireless automotive software updates using blockchains: a proof of concept"," *22nd International Forum on Advanced Microsystems for Automotive Applications, Springer*, p. 137–149, 2017.

[36] T. Kuppusamy, A. Brown, S. Awwad, D. McCoy, R. Bielawski, and J. C. C. Mott, A. Lauzon S.and Weimerskirch, ""uptane :securing software updates for automobiles"," *14th International Conference on Embedded Security in Car Europe*, pp. 1–11, 2016.

[37] L. Lamport, ""paxos made simple"," *ACM SIGACT News*, pp. 51–58, 2001.

[38] A. W. Malik, A. U. Rahmana, A. Ahmad, and M. M. Santos, ""over-the-air software-defined vehicle updates using federated fog environment"," *Vehicular Communications*, 2021.

[39] Anylogic, ""space markup palette"," *[Online]. Available: https://help.anylogic.com*, 2019.