An Empirical Investigation of Performance Challenges within Context-aware Content Sharing for VANETs

M. Najmul Islam Farooqui[a], Muhammad Mubashir Khan[a,*], Junaid Arshad[b], Omair Shafiq[c]

[a]*Department of Computer Science and IT, NED University of Engineering and Technology, Karachi, Pakistan*
[b]*School of Computing and Digital Technology, Birmingham City University, Birmingham, UK*
[c]*School of Information Technology, Carleton University, Ottawa, ON, Canada*

Abstract

Connected vehicles is a leading use-case within the Industrial Internet of Things (IIoT), which is aimed at automating a range of driving tasks such as navigation, accident avoidance, content sharing and auto-driving. Such systems leverage Vehicular Ad-hoc Networks (VANETs) and include vehicle to vehicle (V2V) and vehicle to roadside infrastructure (V2I) communication along with remote systems such as traffic alerts and weather reports. However, the device endpoints in such networks are typically resource-constrained and, therefore, leverage edge computing, wireless communications and data analytics to improve the overall driving experience, influencing factors such as safety, reliability, comfort, response and economic efficiency. Our focus in this paper is to identify and highlight open challenges to achieve a secure and efficient convergence between the constrained IoT devices and the high-performance capabilities offered by the clouds. Therein, we present a context-aware content sharing scenario for VANETs and identify specific requirements for its achievement. We also conduct a comparative study of simulation software for edge computing paradigm to identify their strengths and weaknesses, especially within the context of VANETs. We use FogNetSim++ to simulate diverse settings within VANETs with respect to latency and data rate highlighting challenges and opportunities for future research.

Keywords: VANETs, Internet of Things, Connected Vehicles, Edge Computing, Cyber-physical systems

1. Introduction

Internet of Things (IoT) is a disruptive technology with applications across diverse domains such as healthcare, manufacturing, business and security. Gartner forecasted that the number of IoT devices would exceed more than 26 billion by 2020, generating revenue of more than $300 billion [1]. The emergence of IoT has witnessed an extraordinary proliferation of such devices with recent studies such as [1] by Gartner estimating the growth of IoT devices to surpass 26 billion by 2020, generating revenue of more than $300 billion. A number of industries, for instance, manufacturing, surveillance, automotive, smart buildings, have adopted IoT devices to automate a variety of complex workflows. According to Forbes [2], owing to the widespread proliferation of IoT in the logistics, transportation and manufacturing industries, there has been approximately 40 billion dollars investment to acquire smart and efficient functioning in these sectors. Smart communication systems, efficient vehicle tracking and many other similar applications have attracted huge investment in the utilisation of IoT in industrial sectors. IIoT is aimed to enable the convergence of IT with operation technology (OT) by leveraging IoT devices (things), cutting edge communication (5G and beyond) and data analytics technologies, and the open Internet achieving cost and performance benefits.

However, the goal of convergence between IT and OT introduces novel interdisciplinary requirements. For instance, consider a thermostat device (designed to measure the temperature of a device) expected to make intelligent decisions aligned with changes in the monitored temperature in a cost-effective and efficient manner. In order to achieve this task, the thermostat device is envisaged to learn and analyze the expected and monitored behavior requiring significant resources such as power, memory and processing. The embedded devices deployed in IIoT systems are typically resource-constrained and therefore require seamless, efficient access to high-performance resources to achieve the intelligence required to accomplish complex operational tasks such as that explained here.

Cloud computing facilitates on-demand and cost-effective access to high-performance resources. However, it is commonly deployed at the backend of an infrastructure. Although cloud computing presents exciting opportunities to achieve resource-hungry computations, it also highlights a disparity between the

*Corresponding author: Muhammad Mubashir Khan, Department of Computer Science and IT, NED University of Engineering and Technology, Karachi 75270, Pakistan. Email: mmkhan@neduet.edu.pk

high-performance backend infrastructure and the resource-constrained devices. In this respect, a number of research efforts have been made to investigate cloud architectures to support *connected vehicles* and VANETs such as [3]. Edge computing, formally defined as *cloud computing systems that perform data processing at the edge of the network, near the source of the data* presents a potential technological solution by addressing the disparity between high-performance infrastructures and constrained devices. Within this context, a number of edge computing technologies (multi-access edge, fog and cloudlets) have emerged to facilitate intelligent decision making required by constrained devices in an efficient manner. Similarly, there have been recent efforts to adopt edge computing within VANETs [4] to leverage its benefits to deliver emerging time-sensitive applications.

However, an efficient and seamless convergence of IT and OT requires resultant system to address challenges such as time sensitivity (typical response time can be in milliseconds), communication/ connectivity (mobile/wireless communication is commonly used), context-awareness (locality of data and processing) and governance (including bespoke security and privacy requirements for individual scenarios) as highlighted by [5]. *Connected vehicles* is a leading use-case within IIoT that leverages edge computing, wireless communications and data analytics to enhance transportation by automating tasks such as navigation, accident avoidance, content sharing and auto-driving. It therefore, aims to improve the overall driving experience influencing factors such as safety, reliability, comfort, response and economic efficiency. In recent years, new applications for VANETs have been proposed including content dissemination [6] and social services [7]. Furthermore, researchers have proposed solutions to the challenges faced by above mentioned VANETs applications. Sedjermaci et al. [8] proposed the use of UAVs to improve connectivity and reduce communication overhead in already deployed VANETs. While a study in [9] introduced an efficient hierarchical clustering protocol (EHCP) for better utilisation of resources and increased network lifetime using multipath communication.

Our main contributions in this paper include presenting a futuristic application scenario for *connected vehicles* (secure, context-aware content sharing) which leverages VANETs to achieve inter-vehicle connectivity. We used this scenario to identify specific requirements, highlight emerging and established solutions and describe open challenges to achieve a seamless and cost-effective convergence between IT and OT. In particular, we identify specific requirements that are fundamental to realising the content sharing application within VANETs such as security, connectivity and governance. In view of these requirements, we conducted a comparative analysis of simulation software for edge computing paradigms to highlight their feasibility to investigate the content sharing scenario. Using the open-source simulator FogNetSim++ [10], we conducted empirical analysis highlighting the significance of performance attributes such as latency and data rate.

The remaining paper is organised as follows. Section 2 describes the emergence of connected vehicular technology in recent years as our motivation and background information about enabling technologies such as cloud, edge and fog computing. Section 3 presents a critical overview of state of the art within secure and trustworthy communication for VANETs followed by Section 4 which describes the application scenario and includes a description a high-level architecture to support this scenario. Section 5 includes a comparative study of existing simulation software to aid empirical evaluation of research within VANETs highlighting their limitations followed by experimentation and analysis in Section 6. Section 7 concludes the paper and highlights avenues for future work.

2. Motivation and background

In this section we provide fundamental knowledge about *connected vehicles* paradigm and enabling technologies such as cloud, fog, and edge computing which underpin this emerging paradigm.

2.1. Connected autonomous vehicles

Connected vehicular technology is a prominent strand of IIoT leveraging cutting edge computing technologies such as wireless communication to enhance vehicles and vehicular networks thereby automating range of driving tasks such as navigation, accident avoidance, content sharing and auto-driving. As reported by [11], the impact of autonomous connected vehicles with respect to fuel efficiency, travel time reduction, crash savings, and parking benefits are estimated as approximately $2000 per year per vehicle. This potentially approaches to $4000 per vehicle taking into account the comprehensive crash costs.

A typical connected vehicular paradigm facilitates gathering information from a range of sources including real-time traffic conditions and roadside cameras to help with travel speed, traffic management control center to help with hazards, and other vehicles to help steer and avoid collision. Such information is envisaged to improve overall driving experience influencing factors such as safety, comfort, response and economic efficiency. Additionally, connected vehicular paradigm is envisaged to play fundamental role in improving reliability of vehicles through on-board diagnostics and alert systems as well as inhibiting autonomous behavior to ensure safety and security [12].

Although *connected vehicles* introduce a number of advantages highlighted above, the connected paradigm leads to proliferation of sensor devices generating significant volume of data. However, realizing the benefits of *connected vehicles* requires intelligent processing of data collected from heterogeneous sources in potentially diverse data types and formats. Furthermore, due to the intrinsic characteristics of a connected vehicular environment, the response time and latency in decision making is critical with significant implications with respect to safety and reliability [13]. Although contemporary sensor devices are improving continuously in their computational capacity, these are still considered inefficient to achieve the latency and response time targets especially when tasked with processing large volumes of data in a time-sensitive manner. These characteristics highlight the need for additional resource capabilities to ensure intelligent processing of data streams collected through different interfaces in a timely manner.

2.2. Cloud and edge computing paradigms

Multi-access edge and other edge computing paradigms including fog, cloudlets and mobile cloud computing have been proposed to bring data and computations close to the vehicles thereby extending intelligence to the edge of the network. Cloud computing has profound role in this. As cloud computing requires data to be transported to and from the end nodes, it incurs significant cost especially in terms of time and bandwidth, limiting the impact of cloud computing to address data processing and analysis. Furthermore, emerging IoT applications such as *connected vehicles*, smart metering and fleet management introduce requirements such as geo-distribution, low latency, mobility and heterogeneity which are difficult to be fulfilled by contemporary cloud infrastructures [14]. Edge computing attempts to address this balance by bringing computation closer to data by establishing edge nodes which have enhanced computational and storage resources. These edge nodes are therefore able to bridge resource-constrained devices within a typical IoT infrastructures with high performance computational resources within a cloud. Figure 1 illustrates how edge computing leverages technologies such as IoT and clouds and its role in delivering intelligent IIoT applications. As explained in this figure, by bridging this connection, edge computing facilitates novel use-cases such as smart manufacturing, smart cities and industry 4.0, advancing innovative applications such as that explained in the next section.

- **Multiacess Edge Computing (MEC)** allows radio access network for delivering cloud services at the edge of the network[15]. MEC uses virtualization technology to deploy application servers at the 5G/LTE base stations[16], 3G radio network controllers or cell aggregation sites. It was first introduced by IBM and Nokia, when they implemented a platform for running applications within a mobile base station. Benefit include location awareness, low latency and high bandwidth. After the integration with 5G, MEC ensures that performance of applications such as driverless cars, virtual/augmented reality and Internet of Things can be vastly improved.
- **Fog Computing** brings cloud services closer to the network edge[17]. IoT devices can offload operations such as data storage, processing and networking to the nearby fog gateways which can help increase battery life of these resource constrained devices. Fog computing can be used in applications where low latency is of utmost importance such as patient monitoring in healthcare application, *connected vehicles* and wireless sensors and actuator networks. Data processing such as compression and filtering can also be done at the fog node to minimize the bandwidth use. Furthermore, additional characteristics of fog computing include mobility, location awareness and heterogeneity.
- **Cloudlets** were introduced by Satyanarayanan et al[18]. in 2009 and defined as a small cloud consisting of a cluster of computers near the mobile devices. Cloudlets can store the copy of data and code cache available somewhere else like a remote cloud. The idea was to provide low-latency and high-bandwidth access to the cloud resources in order to support real-time and resource intensive applications. Both cloud service providers and network infrastructure owners can deploy cloudlets and can offer value-added services. It is necessary to simplify the management so that cloudlets can be deployed within local businesses such as restaurants, shops and clinics. Authors proposed the use of VM migration and dynamic VM synthesis to allow the cloudlet to be reused easily.
- **Mobile Cloud Computing (MCC)** allows mobile devices to use cloud for both data storage and processing[19]. It increases the data storage capacity as well as the battery life of mobile devices. Applications include but are not restricted to mobile banking, mobile medical and healthcare, mobile education, mobile surveillance and mobile gaming. Challenges which can be faced in MCC can be related to communication because of low bandwidth, computation offloading and security.

3. Related works

Disseminating content between the devices or vehicles in a typical VANET requires an efficient approach which incurs minimum network overhead, minimum delay and high throughput [20, 21]. Zhang et al.[22] proposed a peer to peer (P2P) based content sharing model to exchange content between vehicles on the road. The

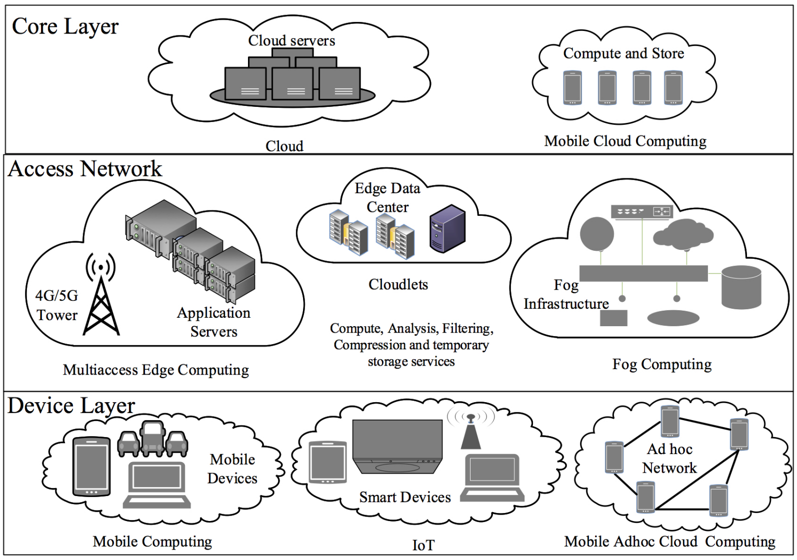


Figure 1: A layered view of cloud and edge computing paradigms

proposed model i.e. "Roadcast" is based on the assumption that most popular data is likely to be shared among the users on the road thus requires overall query delay. The approach requires a model for finding popular content from the large variety of available content, based on the popularity aware content retrieval method. However, this approach incurs some overhead to store the popular content in some centralized buffer so as to minimize the query time. Li et al. [23], evaluated how the distribution and propagation of of popular content effects the performance of underlying content delivery network in the vehicular network. Specifically, the authors evaluated the performance in the context of traffic overload the network could support for different capacity scaling laws.

Luigi et al. [24] addressed the problem of traffic offloading through the use vehicular clouds considering the user quality of experience along with some realistic assumptions (e.g., content of popularity, similarity in content, same size etc.). Ding et al. [25] improves the dissemination of content and achieve high data throughput by deploying static nodes at road intersections for relying the data between communicating nodes. Zhu et al. [26] used a content-centric unit based on the naming information for exchanging the data or content between vehicles in the network. The model operates in two phases, in first phase, vehicles can broadcast their requests for the required content in a conventional information request way, and in the second phase the vehicles deliver the content with the help of content-centric unites. Malandrin et al. [27] investigate the impact of different factors such as placement and location of road side units, penetration of vehicles on the data delivery between the vehicles.

A context-aware system needs to consider context attributes such as location, energy requirement of the node, energy requirement of data transfer, bandwidth requirements for the dissemination of content between vehicles in the vehicular network. The context-awareness become focal point in the IIoT because of resource constrained devices and variety of tasks performed by the nodes in the network. Valerio et al. [28] proposed a context-aware model that collects data from the uses or mobile devices and build a reasoning model in the multidimensional context using data and information retrieved from the mobile devices and their users. Madhukalya et al. [29] proposed a context-aware model for for dynamic participatory environments based on the a publish-subscribe architecture. Further, Jafar et al. [30] propose a content delivery system for VANET-based software defined edge computing framework. The approach incorporate mobile edge computing services in the network base station to ensure reduced latency for data delivery. It also uses vehicle-level caching techniques to provide vehicle-to-vehicle services instantaneously from neighboring vehicles on the road.

In [31], the authors proposed emergency message broadcasting schemes for VANET and vehicular fog computing based on congestion avoidance.They present a Fog-assisted VANET architecture in order to implement the message congestion scenarios in an efficient manner. Furthermore, a taxonomy of schemes is presented to describe the message congestion avoidance.

Rahmani et al. [32] used fog computing to implement an intelligent intermediate layer at the network edge to provide services such as local storage, real-time processing and data mining in IoT based healthcare systems. Authors claim to minimize the processing load on the sensor network and tackle challenges such as power management, mobility, reliability and scalability in e-health applications.

Cech et al. [33] developed fog computing-based solution which uses blockchain technology to store and share sensor data. They designed a network of fog nodes consisting of Raspberry Pi single board computers, docker container system and multichain permissioned blockchain. Authors proposed the use of the architecture for healthcare applications where fog computing integrates with immutable local storage to store and transfer data securely.

4. Secure context-aware content sharing for intelligent IIoT

Connected vehicular paradigm is a broad spectrum encompassing diverse dimensions such as vehicle-to-sensor (V2S) (inter-vehicular), vehicle-to-vehicle (V2V) (intra-vehicular), vehicle-to-road (V2R), and vehicle-to-internet (V2I) [6]. These different dimensions of connected vehicular paradigm are illustrated in Figure 2 where each of these introduces specific challenges with respect to connectivity, response time, reliability and security. This section focuses on the VANETs and presents a futuristic content-sharing scenario along with a detailed description of specific challenges it introduces with respect to time sensitivity, connectivity, context-awareness and governance including security and privacy.

4.1. A content hub for connected vehicles

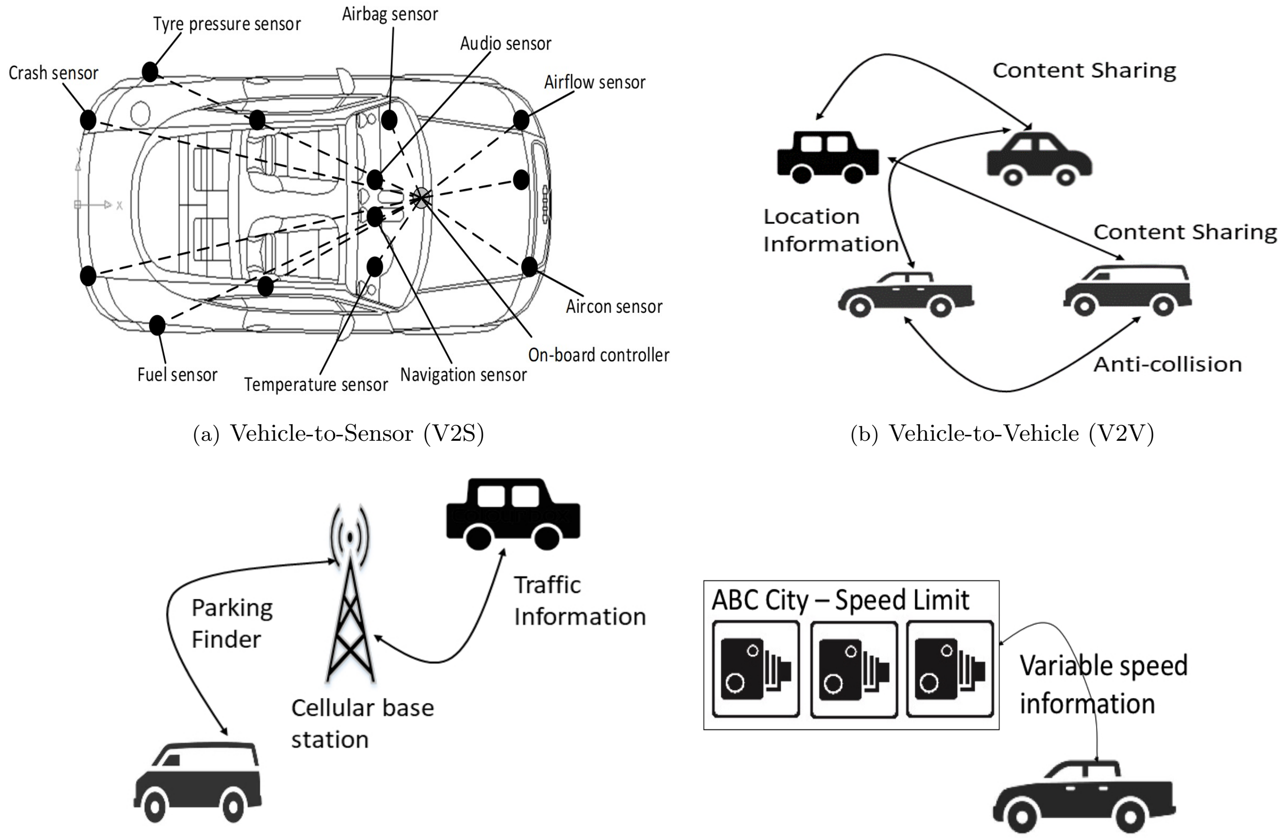
In this section we consider a scenario to simulate ad-hoc content sharing among vehicles within a connected environment. For instance, Deborah is travelling on a motorway with her family on a long trip. Deborah envisages considerable traffic during her journey and therefore is keen to keep things refreshed to avoid boredom through the journey. Deborah's car is an intelligent, connected vehicle which is able to plan the journey ahead and estimate travel time based on the traffic information it receives from the traffic control center(s) for the route. This information (the route and travel time estimates) is updated in real-time through interaction with motorway traffic management control systems installed at the roadside units on the smart motorway. These units are able to feed live traffic information such as incidents, speed limit and route information to the vehicles in their vicinity to enable real-time updates and traffic management. Deborah's car is also fitted with personalized entertainment units for each passenger which can support variety of content. It enables Deborah to keep her kids occupied through a long journey whilst allowing her to concentrate on driving and the road ahead. In this scenario, the personalized entertainment units can get content through service providers such as the cellular operators which use their intelligent infrastructure to deliver the required content. Furthermore, these units are also able to fetch content from other similar (autonomous, connected) vehicles on the road in the vicinity of Deborah's car therefore achieving context-awareness. We envisage it is achieved through implementation of an ad-hoc content hub which serves as a content repository for each vehicle. The content hub enables ad-hoc content sharing among vehicles within a V2V network through the following features:

- Store content for each vehicle that can be viewed through personalized entertainment units.
- Ability for a user to set and reset content as available for sharing making it visible to other vehicles in the vicinity.
- User's ability to download content of interest from this repository onto their local repository.
- Real-time viewing and interaction with the content.
- The content repository as a hot storage and enabling a user to maintain backup at a stable storage location.

The above scenario is illustrated in Figure 3 and highlights a number of interactions within different aspects of connected vehicles paradigm. For instance, the real-time route planning is achieved through vehicle's interaction with the road-side traffic management service. The information collected through this service can be utilized by an edge server to identify traffic patterns throughout Deborah's route and identify an optimal route for the remaining journey. The ad-hoc content hub service is envisioned to be empowered by an edge service which ensures real-time content delivery to the personalized entertainment units installed in Deborah's vehicle. Although a user's content is envisaged to be hosted at a back-end cloud infrastructure, cutting edge caching services are envisaged to be used at the edge to make content of interest available to its users. Furthermore, an intelligent content management service, hosted at the edge, is expected to add and remove content from within a user's content repository as well as setting and resetting of content properties such as sharing. Additionally, it is envisaged to leverage high-end ad-hoc connectivity with neighboring vehicles to achieve seamless content sharing with them.

In order to explore solutions to achieve this scenario, we identify and elaborate specific requirements below.

Time sensitivity: One of the fundamental requirements to realize this scenario is the time required to access and download the content to a user's personalized entertainment unit as this will have a direct impact on the user's experience. Typical response time requirement is in milliseconds as it is envisioned to avoid jitter and guarantee the quality of content available to the end user. We elaborate this requirement further in



(a) Vehicle-to-Sensor (V2S)

(b) Vehicle-to-Vehicle (V2V)

(c) Vehicle-to-Internet (V2I)

(d) Vehicle-to-Road (V2R)

Figure 2: Autonomous connected vehicle paradigm

Section 6 through experimentation with a testbed to evaluate the delay in accessing content.

Connectivity: The communication media used within a network depends on the application under consideration. From Figure 3, the communication media can vary across the connectivity such as V2V, V2R and V2I. For instance, V2I connectivity can be achieved through cellular service provider using Wifi or emerging technologies such as 5G. Furthermore, V2V communication has benefited from advancements in ad-hoc networks with special emphasis on Vehicular ad-hoc networks (VANETs) which is envisaged having significant impact on functions such as collision avoidance, fleet management, and road safety applications. Technologies such as millimeter wave have emerged to support communication within VANETs due to their ability to; achieve minimal response time and support dynamic nature of these infrastructures.

Context-awareness: Context-awareness refers to the property of an entity being informed of its surroundings and is envisaged to achieve intelligent decision making utilizing information gathered through local sources. For the content sharing scenario presented in Figure 3, there are two distinct dimensions which mandate context-awareness i.e. interactions between vehicles (VANET) and between a vehicle and a base station or road side unit to enable localized data storage and processing. In this context, the interactions between vehicles enable sharing content of interest across these vehicles whereas the interactions between vehicle and a road side unit enables intelligent decision making for applications such as real-time route planning.

Governance: Governance generally includes variety of issues such as usability, data quality, security and privacy. The content sharing scenario introduces a number of security challenges such as; secure storage of content at the edge network, robust access control policies to enable fine-grained content sharing policies, an effective identity management system to design authentication and authorization solutions, dynamic trust establishment to achieve trustworthy data sharing across vehicles. Although number of these issues are synonymous to existing systems, the dynamic and ad-hoc characteristics of such networks makes them non-trivial and therefore require explicit efforts to investigate and address them. For instance, in order to facilitate innovative applications such as ad-hoc content sharing across vehicles, the trustworthiness of vehicles within a

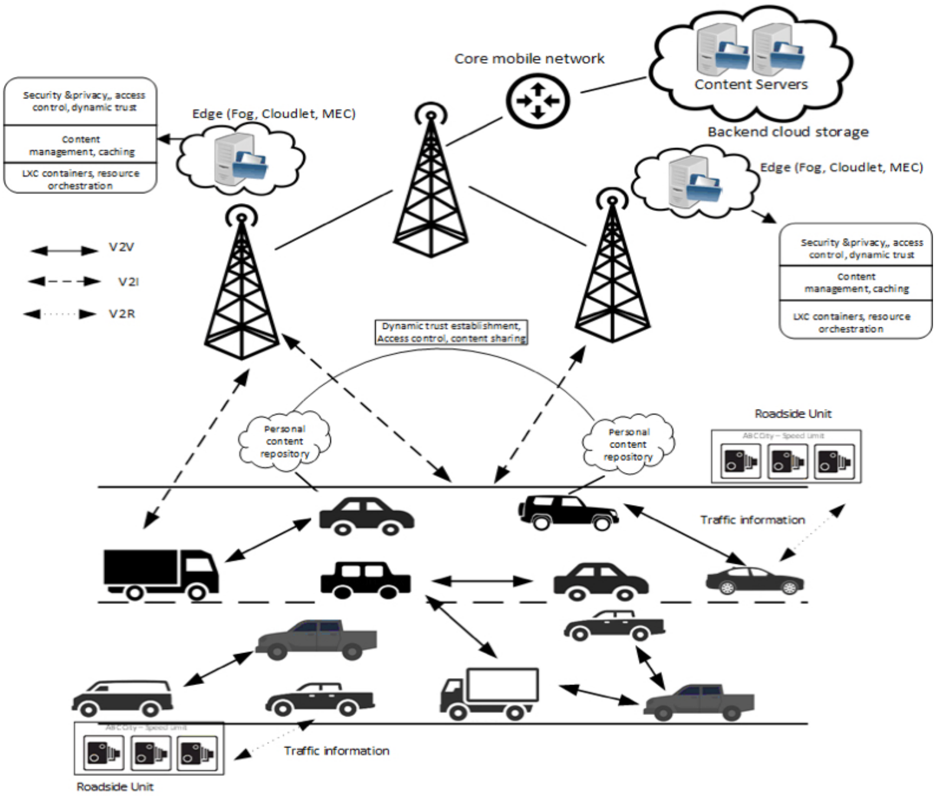


Figure 3: A content hub for connected vehicles

VANET is paramount as absence of an effective mechanism can lead to malware distribution across such network compromising the network and endpoint integrity. Furthermore, a connected vehicle will rely on a number of data sources to operate in an expected manner such as GPS to help navigation, roadside cameras to help with travel speed, road control center to help with hazards and accidents and other vehicles to help steer and avoid collision. Therefore, all such information is required to be trustworthy and transmitted in a secure manner to avoid tampering. In addition to these, there are privacy challenges with regards to storing and utilizing vehicle specific information such as location, travel patterns as well as interactions with different services through a route. Such personal information is generally stored within a vehicle and is vulnerable to hacking and burglary.

5. Comparative study of simulation software for edge computing

In order to implement the application scenario presented in Section 4.1, a thorough study of the existing fog and edge computing simulators and emulators was conducted. We particularly focus on iFogSim [34], EdgeCloudSim [35], Yet Another Fog Simulator (YAFS) [36], FogNetSim++ [10], EmuFog [37] and FogBed [38] due to their suitability for our application scenario. We present a description of this comparison below as summarize the results in Table 1. The following criteria has been used for this comparative study.

- Programming language used to develop the simulator which can impact its adoption and extensibility.
- Underlying edge computing model such as fog, cloudlets or multi-edge access.
- Topology structure supported for nodes which can include graph or tree etc.
- Mobility indicating if the nodes can exhibit mobility patterns.
- Security such as communication security, privacy, trust establishment and management.
- Scalability to highlight the number of nodes supported within a simulation and if these can be scaled dynamically.

5.1. iFogSim

iFogSim is a popular fog simulator used to model and simulate IoT, Fog, and Edge computing scenarios [34]. It is based on CloudSim and written in JAVA programming language. iFogSim can be used to measure different performance metrics by simulating cloud data centers,

Simulators	Underpinning Technology	Paradigm	Platform	Cloud integration	Topology Support	Security Support	Mobility Support	Scalability Support	Performance monitoring
iFogSim	CloudSim	Fog	Java	Yes	Tree	No	No	Yes	Yes
EdgeCloudSim	Cloudsim	Edge	Java	Yes	Tree	No	Yes	Yes	Yes
YAFS	Nil	Fog	Python	Yes	Graph	No	Yes	Yes	Yes
FogNetSim++	Omnet++	Fog	C++	Yes	Graph	No	Yes	Yes	Yes
emuFog	Mininet	Fog	Java	Yes	Graph	No	No	No	Yes
FogBed	Mininet	Fog	Python	Yes	Graph	No	No	Yes	Yes

Table 1: Comparison of edge computing simulators and emulators

edge devices and network connections. iFogSim implements a model called as *sense-process-actuate* to measure average latency, energy consumption and network usage during a simulation. With respect to placement of application modules iFogSim provides two strategies i.e. cloud-only placement and edge-ward placement. Cloud-only placement implements conventional cloud-based applications where all the modules are executed in the data centers. Whereas, edge-ward placement strategy deploys application modules closer to network edge. The drawbacks include lack of mobility, security and scalability support.

5.2. EdgeCloudSim

EdgeCloudSim is another simulator based on CloudSim which is written in JAVA and focuses on edge computing[35]. It can be used to model network, computation and fog specific parameters. EdgeCloudSim allows hierarchical structure of network topology and can be used to model Wireless LAN, LAN and WAN. It uses CloudSim to provide cloud services which include managing VM allocation on data centers, handling task execution on VMs, providing the cloud resources, and modelling consumption of power of the data centers. Furthermore, five unique modules are provided to handle the edge computing scenarios, such as core simulation, load generator, edge orchestrator, networking, and Mobility. Simulations generate Comma Separated Values (CSV) format files to store results including delay, average task failures because of mobility, service time, VM utilization, and cost.

5.3. Yet Another Fog Simulator (YAFS)

YAFS is written in Python and uses Simpy to generate discrete event simulation scenarios[36]. Almost all the simulators in this study extend previous simulators, but YAFS is written from scratch and is not based on any other software. As compared to the tree topology structure provided in iFogSim and EdgeCloudSim, YAFS supports graph topology. Simulation results can be stored in CSV format which can then be analyzed using third-party libraries Panda or R. These output parameters include network delay, response time and network utilization. YAFS authors claim to model three scenarios which can not be implemented with any of the current simulators which include, allocating application module dynamically, dynamic node failure and user mobility.

5.4. FogNetSim++

FogNetSim++ is written in C++ programming language and extends OMNet++ simulator[10]. It is a discrete-event simulator in which users can simulate diverse fog computing scenarios. It uses graph to represent network topology and allows static and dynamic nodes addition. End devices, Fog nodes and Brokers are the core modules in FogNetSim++ which communicate using different protocols like Advanced Message Queue Protocol (AMQP), Constrained Application Protocol (CAP), and Message Queue Telemetry Transport (MQTT). It also allows simulations to perform node scheduling, incorporate customized mobility models, and manage handover mechanisms. FogNetSim++ supports entity and models for group mobility. In the entity model, nodes move independent to other nodes. Whereas, in group model, node mobility is dependent on other nodes. New entity mobility models can also be added to the already available Random Waypoint model, mass mobility, deterministic motion model and Guass-Markov model. FogNetSim++ simulation records parameters such as delay, latency, data rate, handovers, packet error rate and bit error rate. Although it supports mobility but doesn't yet support VM migration between the fog nodes.

5.5. EmuFog

EmuFog is a very useful tool for emulating workloads in real applications [37]. It is an emulation framework that is designed for fog computing infrastructures. It is claimed to be an extensible and scalable framework to simulate different types of workloads for different scenarios, and large network topologies. It extends Maxinet which is multimode extension of Mininet. As compared to real-world test-beds and simulation, emulation uses real applications to conduct experiments that are both controllable and repeatable. These EmuFog emulations of real applications and workload, implement design objectives such as extensibility and scalability for large-scale topologies. The workflow of fog computing emulations in EmuFog consists of 4 steps which include topology generation, topology transformation, topology enhancement and deployment and execution. After designing a network topology, fog nodes are connected in the topology, which in then run Docker applications.

5.6. FogBed

FogBed is a Mininet based emulator which is written in Python and uses Docker containers for rapid prototyping of fog nodes[38]. It emulates the fog layer and provides low cost deployment, flexible setup and real-world protocols and services. Fog nodes can be deployed in several network configurations as docker containers. FogBed allows preconfigured container images to be used as virtual cloud, fog and edge nodes in an emulation. In order to emulate any fog scenario, container image and topology need to be defined. Configuration settings, code and service description is included in a container image. If required, SDN controller can also be started. FogBed also provides standard interfaces to test third party systems such as virtualization, resource management and orchestration. FogBed does not provide support to test mobility, security and reliability of a fog service.

5.7. Comparative analysis of simulation software

From the above-mentioned 6 simulation software, 4 are simulators i.e. iFogSim, EdgeCloudSim, YAFS and FogNetSim++ and 2 are emulators i.e. emuFog and FogBed. iFogSim and EdgeCloudSim are extensions of CloudSim while FogNetSim++ extends OMNet++ simulator. Similarly, EmuFog and FogBed are extensions of Mininet emulator and use docker containers to emulate fog nodes. Three of the simulators (EdgeCloudSim, YAFS, and FogNetSim++) provide either user or App/ VM mobility support while in EmuFog the mobility capabilities will be added as part of the future work. Mobility support is an important requirement for Edge/Fog applications as users can be moving and location-aware services are required. An interesting observation identified as part of our study is that none of the above mentioned simulators provide security features which is one of the emerging challenges within 5G and associated technologies. Embedding security functionalities is mentioned as future work by the authors of FogBed emulator. Furthermore, FogNetSim++ also does not directly support security features. However, keeping in view of the fact that FogNetSim++ extends OMNET++ that is why additional modules may be integrated in order to support security analysis. Furthermore, as scalability support is significant to evaluate methods and techniques with respect to varying volume of traffic, all emulators except EmuFog have support for scalability in terms of number of nodes. As can be seen in the Table 1, most of the simulators are developed in JAVA programming language with limited support for Python.

Through these findings, we concluded to choose FogNetSim++ to implement and evaluate the application scenarion presented in Section 4.1 as it provides the mobility support required in the scenario as well as allow additional OMNet++ modules to be integrated in the simulation. However, as with all other simulators, FogNetSim++ does not support implementation and evaluation with respect to security and we consider this as a future avenue of work for us.

6. Evaluation and analysis

In order to understand the requirements of the content sharing use-case presented in Section 4.1, an in-depth evaluation was conducted using different settings with respect to mobility pattern, number of users, and data rates. Details of experimentation along with results and analysis is presented below.

6.1. Experimentation setup

This section presents the simulation setup, experimentation scenarios, results and analysis of the the VANET use-case presented in 4.1. In order to conduct experimentation, a comparative analysis of existing fog/edge simulation software was carried out and presented in Section 5. The comparison showed different characteristics of simulators and support for features such as mobility and security. In case of VANETs in general and the content-sharing use-case in particular, mobility and security are two of the most important features. As such, security is a composite attribute which includes secure content sharing, communication, and authentication access control. Our comparative study of existing simulators highlighted that security features are not available as part of any of the existing simulators. On one hand it limits experimentation with respect to security analysis but at the same time, it presents opportunities for further work to extend capabilities of existing simulation software. We envisage exploring opportunities to achieve this as part of our future work.

Mobility is another important feature of a VANET content sharing scenario as vehicles within a typical VANET as expected to be mobile. The comparison of existing simulation software highlight that mobility feature is available in two simulators i.e. EdgeCloudSim and FogNetSim++. We have chosen FogNetSim++ as the simulation software to conduct experimentation as it provides multiple mobility models (circle and linear) and allows to record and analyze numerous simulation parameters such as latency, packet error rate, bit error rate and task completion rate.

The simulation testbed as modelled in FogNetSim++ is presented in Figure 4 whereas a refined model is illustrated in Figure 5 which consists of four vehicular nodes which are connected to multiple base stations which also host the edge computing infrastructure. Each node is envisaged to have a content repository where the content is stored in the cloud storage at the backend cloud servers. This content is delivered to the vehicular nodes through edge devices via a BaseBroker. Data published by the registered vehicular node is sent to the

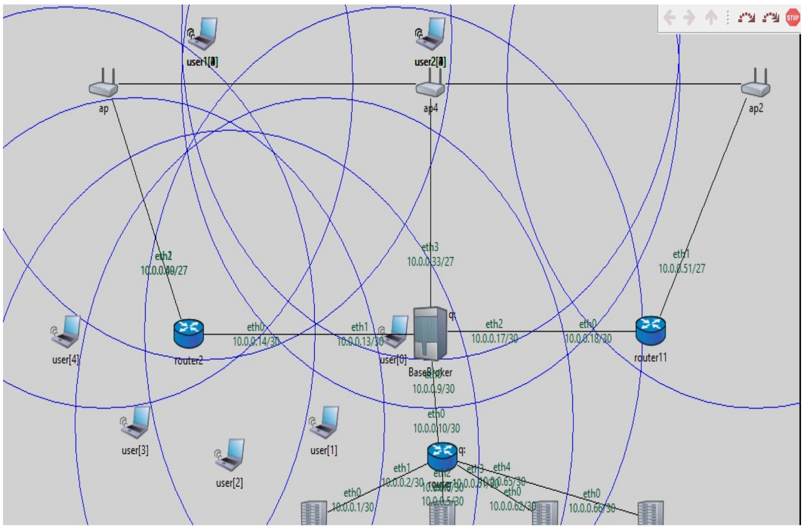


Figure 4: Simulation testbed in FogNetSim++

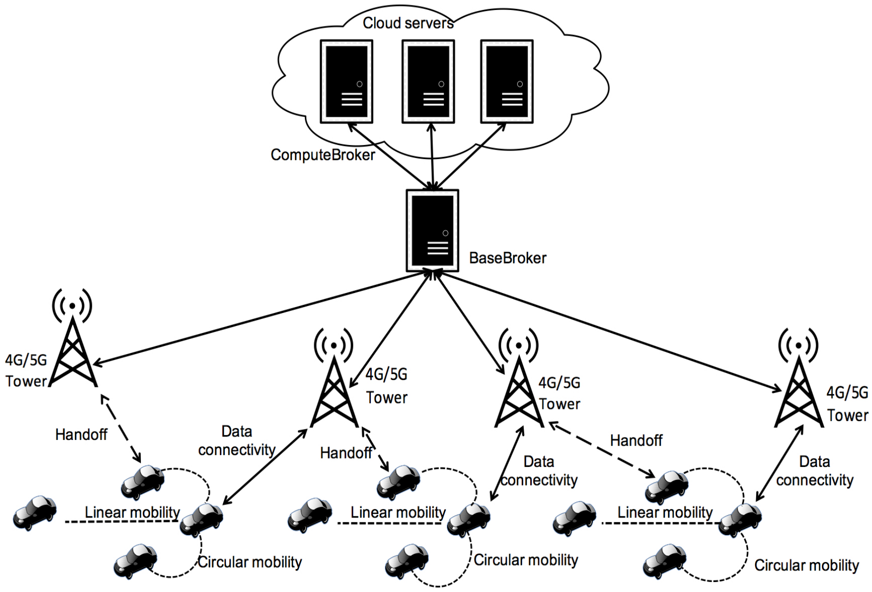


Figure 5: Simulation testbed used for evaluation

broker node. Moreover, all nodes are IP enabled and can be dynamically configured through DHCP to connect with data services. Apart from MQTT protocol several other TCP/IP protocols including TCP, UDP, HTTP, FTP and SNMP are also supported by the devices.

FogNetSim++ allows users to configure various parameters for example, the number of mobile users and their types, the number of applications and brokers at each node. In the case of mobile user, mobility type, speed and angle can be defined. Wireless network data rate and the shared content size can also be changed for different simulation scenarios which can help to calculate the time it takes to complete a task. Using these capabilities, multiple mobile users have been considered which generate messages to the neighboring nodes. The simulation was run on a core i5-4200 2.5 GHz system with 8 GB RAM.

6.2. Experimentation scenarios

In order to aid the understanding of the scenario as well as to evaluate the ability of simulator to support such use-cases, we consider a smart content delivery system where mobile vehicles form ad-hoc network and can offer and use content streaming services. A moving car can register its content sharing service to the broker running at the fog node and whenever another vehicle comes in the vicinity it can steam and download the shared content after being notified by the fog node. The quality of service with respect to content streaming and delivery depends on multiple parameters such as the mobility model, speed of the vehicles and the size of the content. In addition, network parameters such as

Scenario No.	Number of users	Mobility pattern	Data	Vehicle speed
1	4	circle mobility	100KB	20mps
2	4	circle mobility	100MB	20mps
3	8	circle mobility	100KB	20mps
4	8	circle mobility	100MB	20mps
5	4	linear mobility	100KB	40mps
6	4	linear mobility	100MB	40mps
7	8	linear mobility	100KB	40mps
8	8	linear mobility	100MB	40mps

Table 2: Scenario used in experimentation

bandwidth, user speed and latency play important role in this communication.

The experimentation consisted of eight scenarios which have been described in Table 2. These scenarios use settings varying across mobility pattern, data size, vehicle speed and number of users and therefore provide useful insight into different factors affecting content sharing within VANETs. Specifically, different mobility models such as *Circle Mobility* and *Linear Mobility* were used to simulate the model. For both models, speed was modified from 20mps to 40mps and different data sizes i.e. 100KB and 100MB were used with varying number of users.

6.3. Results and discussion

We have conducted experiments using varying settings to analyse the content sharing scenario with respect to latency, task completion rate and throughput. These parameters enable us to understand the performance implications of use-case considered in this paper especially at the edge computing layer. Specific results and analysis is presented below.

6.3.1. Latency

As illustrated in Table 2, we analysed latency for different scenarios varying with respect to mobility pattern, vehicle speed and number of users. Figure 6 to Figure 9 presents the results of the experiments to analyse latency across different settings. Figure 6 shows the latency for circle mobility model with 100KB data while Figure 7 shows the latency for the same model with 100MB data. Similarly, Figure 8 shows the latency for linear mobility model with 100KB data while Figure 9 shows the latency for the same model with 100MB data.

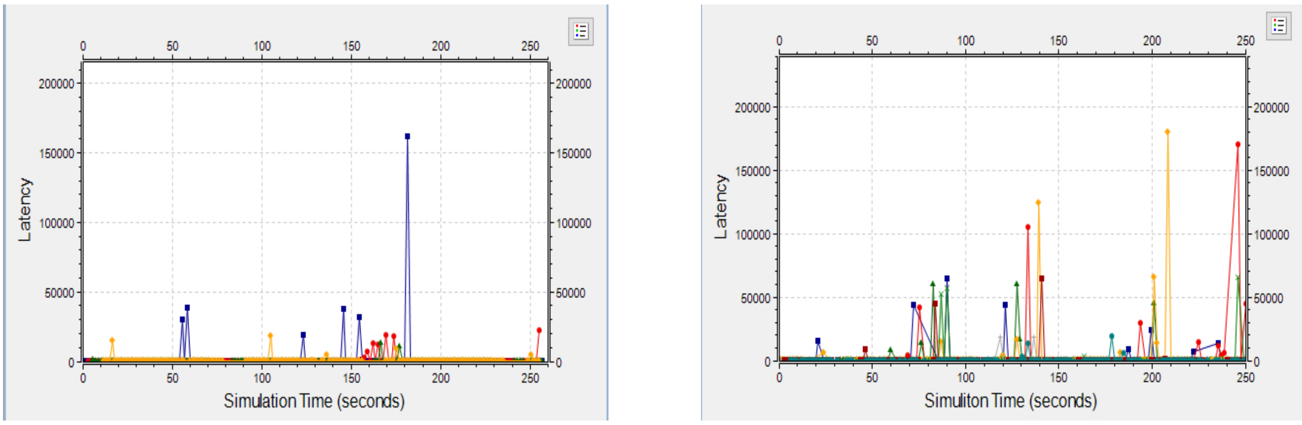
The first set of experiments were focused on analyzing latency for varying settings (number of users and data size) while preserving with circle mobility model. As is evident from Figure 6, with data size fixed at 100KB, if the number of users is increased from 4 to 8, the overall latency increases. This is envisioned due to increased workload on the network due to the increase in number of users. Similarly, as illustrated by Figure 7, when the data size is increased from 100KB to 100MB, the average latency is increased due to change in data size. Although the change in data size is significant, the increase in average latency is observed as double compared to the latency measurements for 100KB data. However, the change in latency for individual users does not necessarily follow the same pattern. For instance, for experiments with circle mobility, mean latency increased for user 2 increased from 279.5 to 1057.9 while the data size was increased from 100KB to 100MB and keeping the number of users to 8. Similarly, for experiments with linear latency model, mean latency for user 2 in case of 100KB data is 984.35 ms whereas in case of 100MB data, latency for same user was 3455.7 ms.

The above observations highlighted a linear relationship between latency, data size, and number of users. We observed that when simulation was run with the number of users equal to 4 and data size 100MB, it was observed that the latency decreased as compared to the value observed while the number of users was set to 8. We believe this is due to the increased workload for the infrastructure due to increased traffic generated by greater number of nodes.

6.4. Data rate

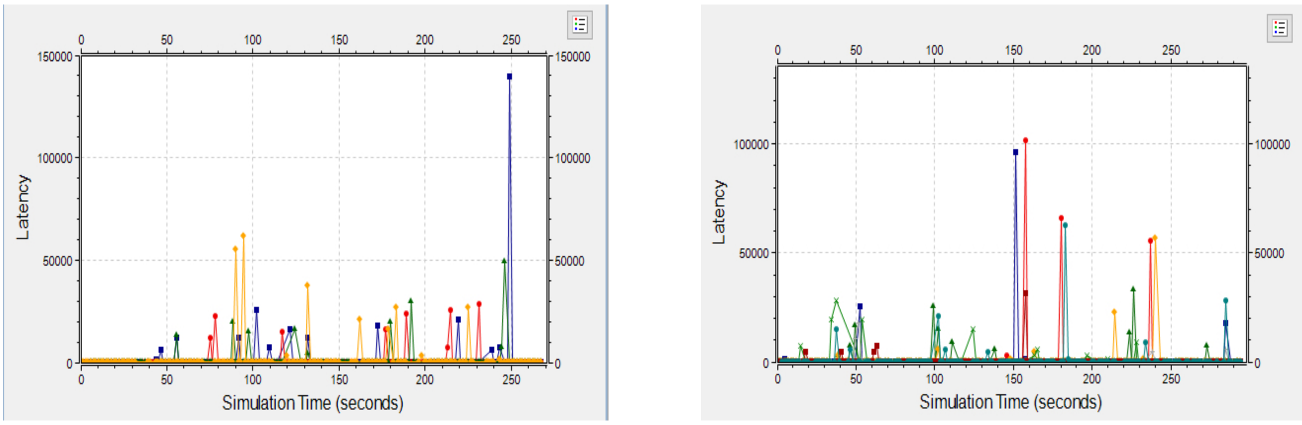
The underpinning technology of FogNetSim++ is OMNeT++ that provides a comprehensive platform for simulating variety of fog computing applications. A user can simulate realistic network scenario by setting different parameters such as the data rate, the number of brokers, fog nodes, end nodes, and the mobility models for individual users and network nodes. In order to facilitate the modelling of communication networks, physical links are modelled using connections which support various parameters including bit error rate, data rate and the packet error rate. These parameters can be assigned in either the NED file or the INI configuration file. During the simulation, when data rates are in use, a packet is delivered to the target module that corresponds to the end of packet reception. Data rate is used for calculating the transmission duration of packets and is represented by bits per second. In our simulation we have observed the actual data rate achieved by each network device.

The results with respect to the data rate have been presented in Figure 10 and Figure 11. Four graphs have been shown where data rates for different nodes have been compared varying data size and mobility models keeping



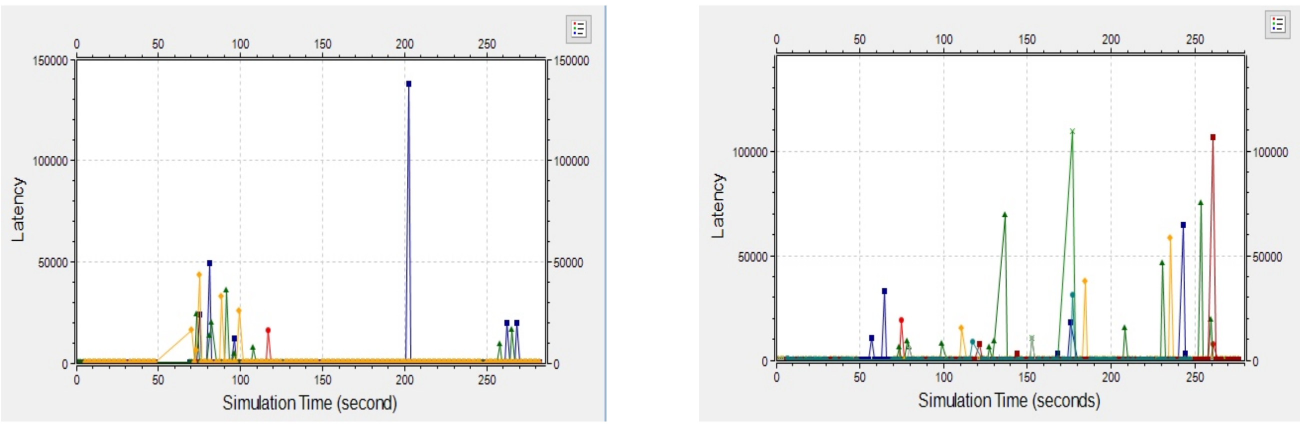
(a) Latency with 4 users (b) Latency with 8 users

Figure 6: Latency for circle mobility model with 100KB data



(a) Latency with 4 users (b) Latency with 8 users

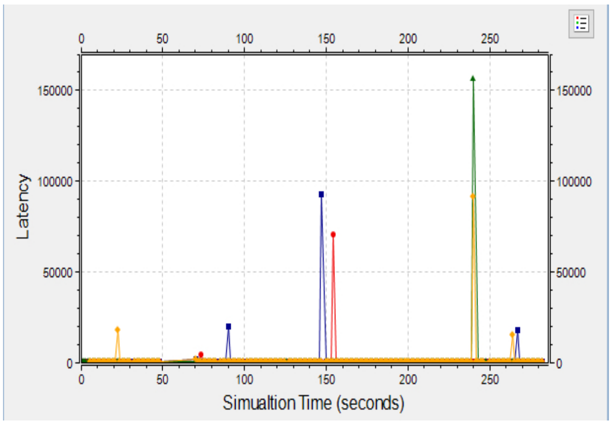
Figure 7: Latency for circle mobility model with 100MB data



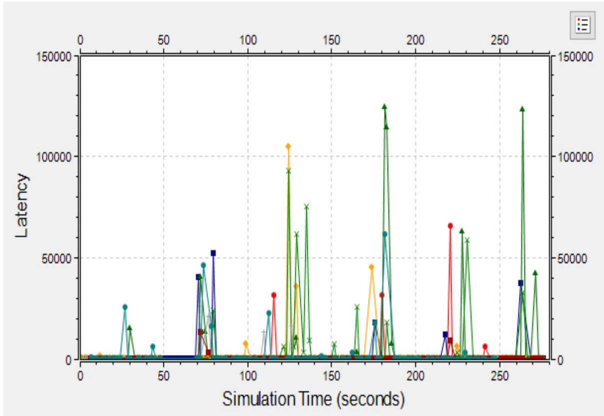
(a) (b)

Figure 8: Latency for linear mobility model with 100KB data

the number of users same. First graph gives comparison between linear and circle mobility models and keeping the data size 100k. As illustrated in this figure, data rates achieved in case of LinearMobility are higher as compared to CircleMobility with data sizes of 100kB and 100MB. Only in some devices this pattern is not followed which can be the result of distance of nodes from the network devices. Two more graphs have been shown comparing the

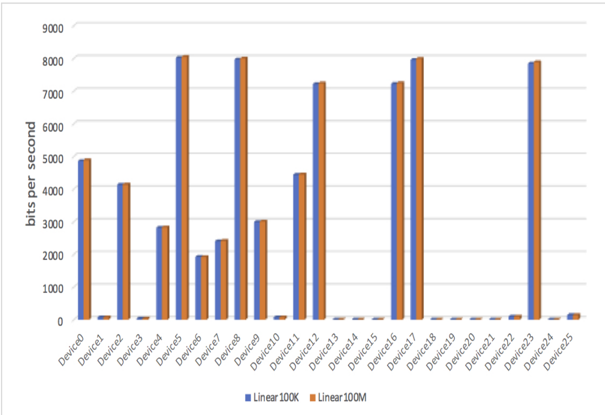


(a)

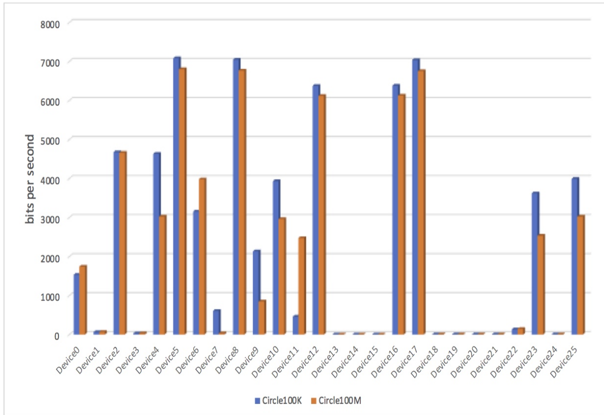


(b)

Figure 9: Latency for linear mobility model with 100MB data

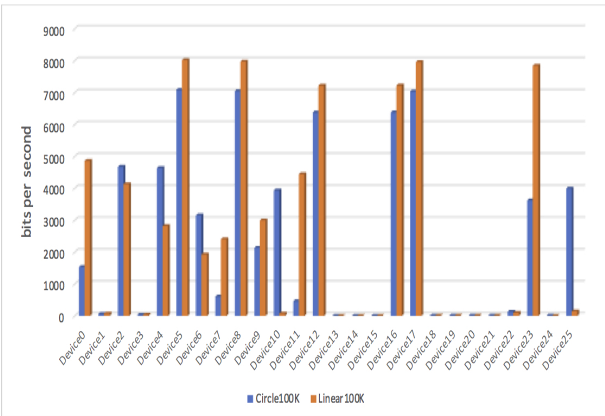


(a) Data rate for linear mobility for different data sizes

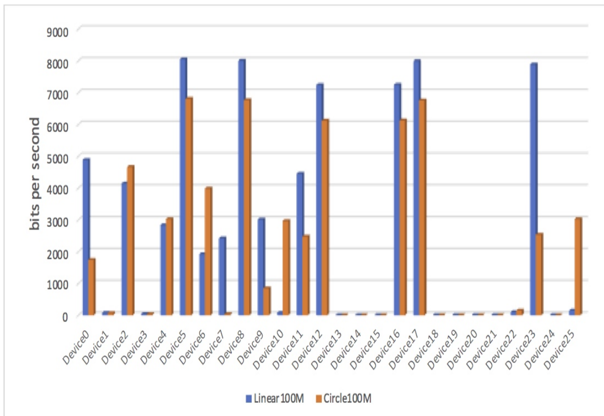


(b) Data rate for circle mobility for different data sizes

Figure 10: Analysis of data rate for linear and circle mobility for different data sizes



(a) Data rate for linear and circle mobility for 100KB data



(b) Data rate for linear and circle mobility for 100MB data

Figure 11: Comparison of data rates between linear and circle mobility for different data rates

data rates achieved with varying data size but keeping the mobility model constant. In case of LinearMobility, there is not much difference observed in the data rates even with different data sizes. But in case of CircleMobility, there is difference in data rates when we changed the data size from 100KB to 100MB which can be caused by the difference in the mobility pattern in terms of circle radius and node direction. In addition to latency, task completion time and throughput, we also measured the memory usage and end-to-end delay for the simulation runs. End-to-end delay in this case represents the time taken fro a request from user to the base broker. After analyzing the simulation results it was observed that as the number of users nodes increase end-to-end delay also increases. This is because an increase in number of users results in larger number of requests made in regular intervals which in turn causes delay in completing these requests.

7. Conclusions and future work

Internet of Things (IoT) are increasingly used within diverse applications to facilitate cutting-edge use-cases such as *smartX systems*. *Connected vehicles* is a prominent use-case of smartX systems that leverage VANETs to achieve automation across a range of driving tasks such as navigation, accident avoidance, content sharing and auto-driving. In this context, edge computing paradigms such as fog, cloudlets and multi-access edge computing have a profound role in enabling seamless integration between resourceful cloud back-end and resource-constrained sensor devices within smart vehicles. In this paper, we have focused at the edge computing layer to investigate challenges and opportunities to achieve secure context-aware content sharing within VANETs. We have achieved a comprehensive comparative analysis of simulation software to facilitate such research highlighting strengths and weaknesses. Further, using FogNetSim++, we presented experimentation to identify and evaluate specific requirements of a content-sharing application, especially with respect to latency and data rate. Through our research, we have identified the lack of support in existing simulation software to investigate security challenges at edge computing layer. We envisage pursuing this direction in our future work. Also, we are interested in the practical implementation of the scenario using more simulators limited to theoretical analysis in this study. Finally, new scenarios are expected to be simulated using these simulators.

8. Data availability statement

The data that support the findings of this study are available from the corresponding author upon reasonable request.

References

[1] L. Algar, 2020. [Online]. Available: http://www.devopsonline.co.uk/iot-installed-base-grow-26-billion-units-2020/

[2] L. Columbus, 2020. [Online]. Available: https://www.forbes.com/sites/louiscolumbus/2017/12/10/2017-roundup-of-internet-of-things-forecasts/2fb0c4e21480

[3] F. Ahmad, M. Kazim, A. Adnane, and A. Awad, "Vehicular cloud networks: Architecture, applications and security issues," in *2015 IEEE/ACM 8th International Conference on Utility and Cloud Computing (UCC)*, 2015, pp. 571–576.

[4] A. Muthanna, R. Shamilova, A. A. Ateya, A. Paramonov, and M. Hammoudeh, "A mobile edge computing/software-defined networking-enabled architecture for vehicular networks," *Internet Technology Letters*, vol. n/a, no. n/a, p. e109. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/itl2.109

[5] E. C. T. Group, "Introduction to edge computing in iiot," Technical Report. [Online]. Available: https://tinyurl.com/y9npszw9

[6] E. Benalia, S. Bitam, and A. Mellouk, "Data dissemination for internet of vehicle based on 5g communications: A survey," *Transactions on Emerging Telecommunications Technologies*, vol. 31, no. 5, p. e3881, 2020. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/ett.3881

[7] R. Esmaeilyfard, N. H. Khiadani, F. Hendessi, and S. M. Safavi Hemami, "A vehicular network for social services using data dissemination," *Transactions on Emerging Telecommunications Technologies*, vol. 28, no. 1, p. e2911, 2017. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/ett.2911

[8] H. Sedjelmaci, M. A. Messous, S. M. Senouci, and I. H. Brahmi, "Toward a lightweight and efficient uav-aided vanet," *Transactions on Emerging Telecommunications Technologies*, vol. 30, no. 8, p. e3520, 2019. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/ett.3520

[9] A. K. Dutta, M. Elhoseny, V. Dahiya, and K. Shankar, "An efficient hierarchical clustering protocol for multihop internet of vehicles communication," *Transactions on Emerging Telecommunications Technologies*, vol. 31, no. 5, p. e3690, 2020. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/ett.3690

[10] T. Qayyum, A. W. Malik, M. A. Khan Khattak, O. Khalid, and S. U. Khan, "FogNetSim++: A toolkit for modeling and simulation of distributed fog environment," *IEEE Access*, vol. 6, pp. 63 570–63 583, 2018.

[11] D. J. Fagnant and K. Kockelman, "Preparing a nation for autonomous vehicles: opportunities, barriers and policy recommendations," *Transportation Research Part A: Policy and Practice*, vol. 77, pp. 167 – 181, 2015. [Online]. Available: https://tinyurl.com/y29pk94b

[12] M. S. Rahman, M. Abdel-Aty, L. Wang, and J. Lee, "Understanding the highway safety benefits of different approaches of connected vehicles in reduced visibility conditions," *Transportation Research Record*, vol. 2672, no. 19, pp. 91–101, 2018. [Online]. Available: https://doi.org/10.1177/0361198118776113

[13] E. Uhlemann, "Introducing connected vehicles [connected vehicles]," *IEEE Vehicular Technology Magazine*, vol. 10, no. 1, pp. 23–31, 2015.

[14] J. Moura and D. Hutchison, "Review and analysis of networking challenges in cloud computing," *Journal of Network and Computer Applications*, vol. 60, pp. 113 – 129, 2016. [Online]. Available: https://tinyurl.com/y5jpg2pk

[15] F. of MEC Industry Initiative, "Mobile-edge computing - introductory technical white paper," Technical Report. [Online]. Available: https://tinyurl.com/k37ghd2

[16] I. Al Ridhawi, M. Aloqaily, Y. Kotb, Y. Al Ridhawi, and Y. Jararweh, "A collaborative mobile edge computing and user solution for service composition in 5g systems," *Transactions on Emerging Telecommunications Technologies*, vol. 29, no. 11, p. e3446, 2018, e3446 ETT-18-0080.R1. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/ett.3446

[17] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things," in *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*, 2012, pp. 13–16.

[18] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies,

"The case for vm-based cloudlets in mobile computing," *IEEE pervasive Computing*, vol. 8, no. 4, pp. 14–23, 2009.

[19] A. u. R. Khan, M. Othman, S. A. Madani, and S. U. Khan, "A survey of mobile cloud computing application models," *IEEE Communications Surveys Tutorials*, vol. 16, no. 1, pp. 393–413, 2014.

[20] S. Ilarri, T. Delot, and R. Trillo-Lado, "A data management perspective on vehicular networks," *IEEE Communications Surveys Tutorials*, vol. 17, no. 4, pp. 2420–2460, 2015.

[21] K. A. Hafeez, L. Zhao, B. Ma, and J. W. Mark, "Performance analysis and enhancement of the dsrc for vanet's safety applications," *IEEE Transactions on Vehicular Technology*, vol. 62, no. 7, pp. 3069–3083, 2013.

[22] Y. Zhang, J. Zhao, and G. Cao, "Roadcast: A popularity aware content sharing scheme in vanets," in *2009 29th IEEE International Conference on Distributed Computing Systems*, 2009, pp. 223–230.

[23] L. Qiu and G. Cao, "Popularity-aware caching increases the capacity of wireless networks," *IEEE Transactions on Mobile Computing*, vol. 19, no. 1, pp. 173–187, 2020.

[24] L. Vigneri, T. Spyropoulos, and C. Barakat, "Quality of experience-aware mobile edge caching through a vehicular cloud," *IEEE Transactions on Mobile Computing*, pp. 1–1, 2019.

[25] Y. Ding and L. Xiao, "Sadv: Static-node-assisted adaptive data dissemination in vehicular networks," *IEEE Transactions on Vehicular Technology*, vol. 59, no. 5, pp. 2445–2455, 2010.

[26] Z. Su, Y. Hui, and Q. Yang, "The next generation vehicular networks: A content-centric framework," *IEEE Wireless Communications*, vol. 24, no. 1, pp. 60–66, 2017.

[27] F. Malandrino, C. Casetti, C. Chiasserini, and M. Fiore, "Optimal content downloading in vehicular networks," *IEEE Transactions on Mobile Computing*, vol. 12, no. 7, pp. 1377–1391, 2013.

[28] V. Arnaboldi, M. Conti, and F. Delmastro, "Cameo: A novel context-aware middleware for opportunistic mobile social networks," *Pervasive and Mobile Computing*, vol. 11, pp. 148 – 167, 2014. [Online]. Available: https://tinyurl.com/y2rgv7gd

[29] M. Madhukalya and M. Kumar, "Concon: Context-aware middleware for content sharing in dynamic participating environments," in *2014 IEEE 15th International Conference on Mobile Data Management*, vol. 1, 2014, pp. 156–161.

[30] J. Al-Badarneh, Y. Jararweh, M. Al-Ayyoub, R. Fontes, M. Al-Smadi, and C. Rothenberg, "Cooperative mobile edge computing system for vanet-based software-defined content delivery," *Computers & Electrical Engineering*, vol. 71, pp. 388–397, 2018.

[31] A. Ullah, S. Yaqoob, M. Imran, and H. Ning, "Emergency message dissemination schemes based on congestion avoidance in vanet and vehicular fog computing," *IEEE Access*, vol. 7, pp. 1570–1585, 2018.

[32] A. M. Rahmani, T. N. Gia, B. Negash, A. Anzanpour, I. Azimi, M. Jiang, and P. Liljeberg, "Exploiting smart e-health gateways at the edge of healthcare internet-of-things: A fog computing approach," *Future Generation Computer Systems*, vol. 78, pp. 641–658, 2018.

[33] H. L. Cech, M. Großmann, and U. R. Krieger, "A fog computing architecture to share sensor data by means of blockchain functionality," in *2019 IEEE International Conference on Fog Computing (ICFC)*. IEEE, 2019, pp. 31–40.

[34] H. Gupta, A. Vahid Dastjerdi, S. K. Ghosh, and R. Buyya, "ifogsim: A toolkit for modeling and simulation of resource management techniques in the internet of things, edge and fog computing environments," *Software: Practice and Experience*, vol. 47, no. 9, pp. 1275–1296, 2017. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/spe.2509

[35] C. Sonmez, A. Ozgovde, and C. Ersoy, "Edgecloudsim: An environment for performance evaluation of edge computing systems," *Transactions on Emerging Telecommunications Technologies*, vol. 29, no. 11, p. e3493, 2018, e3493 ett.3493. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/ett.3493

[36] I. Lera, C. Guerrero, and C. Juiz, "Yafs: A simulator for iot scenarios in fog computing," *IEEE Access*, vol. 7, pp. 91 745–91 758, 2019.

[37] R. Mayer, L. Graser, H. Gupta, E. Saurez, and U. Ramachandran, "Emufog: Extensible and scalable emulation of large-scale fog computing infrastructures," in *2017 IEEE Fog World Congress (FWC)*, 2017, pp. 1–6.

[38] A. Coutinho, F. Greve, C. Prazeres, and J. Cardoso, "Fogbed: A rapid-prototyping emulation environment for fog computing," in *2018 IEEE International Conference on Communications (ICC)*, 2018, pp. 1–7.