FOGO - an Optimized Fog and Edge Computing Method for VANETs

Renato Ćorluka*, Ivona Matoš†, Davor Damjanović‡ and Josip Balen§
Faculty of Electrical Engineering, Computer Science and Information Technology Osijek
Josip Juraj Strossmayer University of Osijek
Kneza Trpimira 2b, 31000 Osijek, Croatia
*rcorluka@etfos.hr, †imatos@etfos.hr, ‡davor.damjanovic@ferit.hr, §josip.balen@ferit.hr

***Abstract*—Vehicular Ad-hoc Networks (VANETs) have many specific issues and challenges due to a heterogeneous and highly dynamic environment, and therefore traditional solutions need to be improved and adopted in order to satisfy the networking and processing requirements. Fog and edge computing principles enable VANETs to achieve a more realistic and dependable architecture by using several layers for information processing. In this paper, the developed FOGO method is improved and customized for VANETs. The proposed solution is using an enhanced infrastructure in which additional mobile fog nodes are added to the network together with the existing stationary nodes. Small and medium sized messages are processed by mobile fog nodes, which then disseminate the results across the network. Roadside units (RSUs) are fixed, highly-capable devices positioned throughout the network to assist the processing of increasing volumes of data. Additionally, the cloud is taken into account, but only when processing large amounts of data. Along with the presented method flowcharts, operating algorithms, and message structure, the proposed system architecture is described. Furthermore, potential use cases are suggested together with the metrics for the system performance evaluation. The LuST scenario was used to evaluate the proposed architecture and the results showed improved message processing efficiency.**

***Index Terms*—Fog and Edge Computing, VANET optimization, LuST scenario, On-Board Units (OBU), Traffic Management**

I. INTRODUCTION

Large-scale improvements in computation and communication technologies are leading the automotive industry towards connected vehicles in order to improve traffic safety, efficiency, and user experience. Based on short-range radio, vehicles are forming a special class of mobile networks (MANET) called VANET networks. Due to the highly dynamic topology of VANET, there are lots of challenges to overcome. One of the solutions for that problem and integration of new technology such as 5G in VANET is software-defined vehicular networks. These networks aim to improve security, QoS, routing reliability, and delay in VANET networks [1], [2].

To unlock the full potential of these networks researchers are making models in which VANETs cooperate with today's cutting-edge technology like cloud computing which is used to compute huge amounts of data. Due to VANETs high mobility nodes, scalability, and frequent topology changes, it is difficult to meet real-time traffic requirements using only cloud computing. Fog computing brings required computing power near VANETs nodes, it is strong enough for computing time-sensitive applications and it has enough storage capacity to gather local traffic information to send it to the cloud. There are different ways to improve intelligent transportation systems (ITS) with fog computing, as well as a variety of cloud computing capabilities and applications are available when VANET and fog computing are combined [3]. As it is described in [2], one of the most promising ways is to use fog computing for a dynamic traffic light system to optimize traffic light junctions. As vehicles are becoming more and more interconnected and generating huge amounts of data, there is a need for fog computing in order to meet time-critical requirements. However, it is not enough to just introduce processing at the edge of the network, it is important to optimize it.

Fog and edge computing reduces response time and brings computing closer to edge nodes compared to cloud computing, but also uses cloud computing when it is needed for big data computation which fog nodes can not compute in a predefined time window. In most methods, fog nodes are stationary nodes that bring the additional cost of maintaining infrastructure. Fog nodes can be deployed to collect trust evaluations from vehicles, allowing them to rely on local vehicles to perform certain tasks. Furthermore, fog nodes can be used to keep track of its local vehicles, reducing the need for cloud usage [4]. The proposed method in this paper includes mobile fog nodes, which have reduced maintenance costs and can cover a wider area because of their mobility but with added reliability and connectivity issues because there will be a time when some city districts will not be covered with mobile fog nodes. The performance of the proposed method is evaluated in the simulation of the LuST scenario. The rest of the paper is organized as follows: Section II describes fog and edge computing architecture. Section III presents our innovative method for applying fog computing in VANETs, whereas in Section IV the simulation results are discussed. Finally, conclusions and future work are given in Section V.

II. FOG AND EDGE COMPUTING ARCHITECTURE

Nowadays vehicles are becoming smarter as they are equipped with various sensors that can provide driving assistance. We are reaching peak private car years and entering the era of new market mobility-as-a-service (MaaS) because the total cost of ownership (TCO) is high. As conventional transport develops into ITS, cars have to be equipped with numerous sensors that generate an immense amount of data that need to be processed and transmitted to other vehicles and infrastructure. At this stage, computational power is too expensive for regular cars and high computational capabilities are reserved for high-end cars, so at this stage, a transitional solution before the era of fully autonomous vehicles is needed. Cloud computing is a well-proven and reliable technology that can provide processing and storage power to vehicles. However, vehicles are highly dynamic nodes and require low latency, and clouds are centralized centers, far away from vehicles. Considering that, to achieve ITS goals, a mediator in our architecture between vehicles and the cloud is needed. Fog computing is an alternative to overcome these problems and challenges. It brings resources on the network edge closer to the vehicles so not all the data needs to be sent to the cloud. Most of the data can be processed on the edge and only the necessary data is forwarded to the cloud. As described in [5], fog computing can perform processing and storage tasks on the most logical resources and reduce the overall impact on the network and processing resources. There are three possible fog connections in VANETs: vehicle to fog, fog to fog, and fog to cloud. In VANETs, fog nodes could be RSU, public transportation, or taxis. As described in [2], [6], and [7], three-level architecture is proposed where the bottom level is composed of end-user vehicles, the middle of edge computing, and cloud computing on the top. Grover et-al. in [2] proposed this architecture for four different applications in VANET: smart traffic lights, parking systems, content distribution, and decision support system. Dolui et-al. in [7] compared three different implementations of edge computing: mobile edge computing (MEC), fog computing (FC), and cloudlet. Every application has its requirements and there are always trade-offs, therefore every implementation has advantages and disadvantages in the considered use case. Edge computing has an N-tier architecture where end devices, smartphones, cameras, and vehicles, are in the lower tier; intermediate tiers consist of FC, MEC, or cloudlet nodes; and the cloud is located at the highest tier. FC can use Bluetooth, Wi-Fi, and mobile networks as an access mechanism. On the other hand, MEC uses only mobile networks, and cloudlet uses only Wi-Fi. They proposed a decision tree for finding the best implementations based on six parameters such as physical proximity, power consumption, computation time, context awareness, logical proximity, and non-IP support. The decision tree is given in Table I.

TABLE I: Edge computing decision tree

	FC	**MEC**	**CC**
Physical Proximity	HIGH	LOW	HIGH
Power Consumption	LOW	HIGH	LOW
Computation Time	HIGH	LOW	LOW
Context Awareness	LOW	HIGH	LOW
Logical Proximity	MAYBE	MAYBE	ENSURED
Non-IP Support	YES	NO	NO

Depending on the proposed intra-fog architecture all the vehicles or some of them could be connected to the cloud. RSUs are the most expensive fog nodes and it is too expensive to use only them. Furthermore, finding the ideal number, locations, and computational capabilities of the RSUs is one of the main issues, especially in urban areas where there are many obstacles inside the RSUs' coverage area [8]. Public transportation like buses and trams can have large computational and storage capabilities because they are spacious, but their problem is that they have fixed routes. To cover other regions taxis can be utilized as fog nodes. They do not have huge computational and storage power, but they go on many streets that do not have considerable traffic flow and their fleet is a few times bigger than public transportation. As described in [9], the cloud paradigm is shifting from locally managed software on physical hardware toward virtual services. The cloud brings all different virtualized services which we can distinguish by cloud orchestration into two approaches: orchestration of hardware-as-a-service (HaaS) and orchestration of software-as-a-service (SaaS). The orchestration enables enterprises to use multiple clouds for numerous services and syncs them into a single workflow. Orchestration in ITS will play a huge role because services will probably be deployed in multiple cloud vendors and they will have to be represented as a single workflow to the end-user.

III. DESCRIPTION OF PROPOSED FOGO METHOD

In this paper, a decentralized system distributed throughout the network with stationary and dynamic fog nodes is presented. In subsection A, a proposed system architecture along with the messaging structure is described. Subsection B describes the workflow of the proposed system. In subsection C, possible use cases are presented. Finally, in subsection D, evaluation metrics are proposed.

A. System Architecture

In this particular case, stationary nodes, i.e. fRSUs, would be placed in an area close to bus stops as they cover most of the city road traffic. In addition, usage of dynamically distributed nodes is proposed, which would be set up in buses and taxi vehicles as those are high-performance vehicles with more computing power and storage. Furthermore, it is worth mentioning that buses have been quite thoroughly researched and simulated, as in [6], [10], [11], in order to provide the best solution for fog message distribution. Therefore, these vehicles would be suitable for running real-time, complex applications that edge-level vehicles could not perform due to the lack of resources. The complete architecture of the proposed FOGO system is shown in Figure 1.

There are two types of nodes in FOGO architecture: the first type consists of edge nodes that represent vehicles with low computational power OBUs; the other type of nodes are fog nodes that represent buses and taxi vehicles along with RSUs which are placed at the public transportation stations. This category is further subdivided by the mobility of these

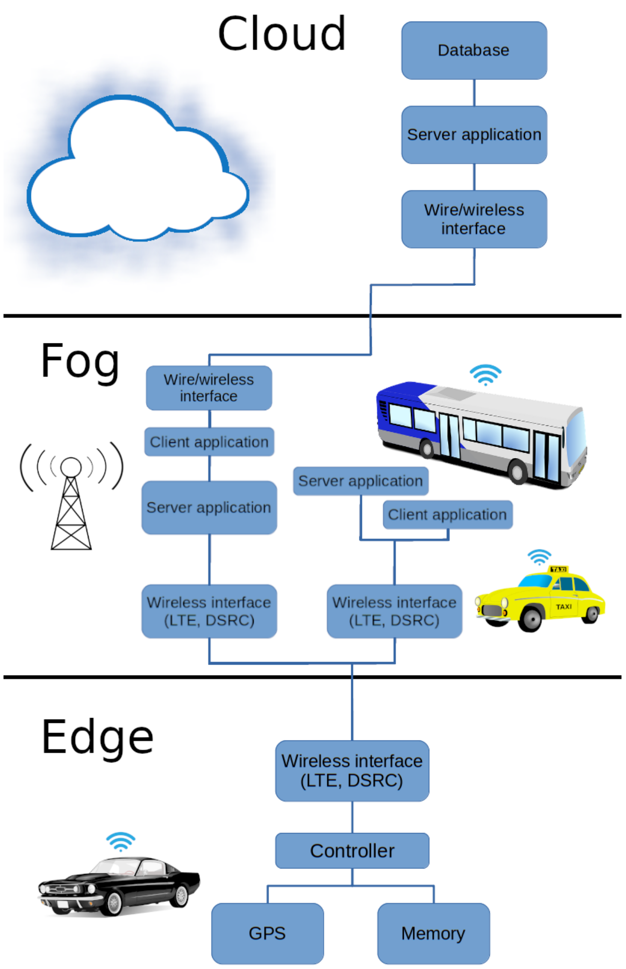

Fig. 1: Proposed FOGO architecture

nodes, taxis, and buses are moving nodes, and RSUs are stationary nodes. Arrows are representing possible types of communication in FOGO architecture. vehicle-to-vehicle (V2V) communication is performed in the DSRC band and it is represented with the white arrow. The blue arrow represents infrastructure-to-cloud (I2C) communication RSUs and cloud, and in FOGO architecture that can be performed wirelessly with a 5G network or by optical fiber infrastructure. The last type of communication is vehicle-to-infrastructure (V2I) in which mobile fog nodes send data to the RSU and RSU sends back results to the mobile fog nodes, which is represented by the green arrow.

In addition, four types of messages are proposed: urgent, essential, application, and media message, each identified by its special header character. Urgent messages refer to messages connected to sudden traffic situations such as congestion, collision, accidents, roadwork, etc. Those messages should be distributed in the network with high priority by using appropriate flags and by sending them to all vehicles on the map of the fog node. Furthermore, these messages contain the following information: message ID, sender ID, GPS coordinates of the location where the traffic situation has occurred and the type of the traffic problem. Moreover, these are the only messages that are being sent without requests from the edge nodes.

Furthermore, essential messages refer to the request messages in which a certain vehicle requests information from the node. The essential message consists of message ID, node ID, GPS location, and type of the information needed (for instance weather or traffic report) or response to the request.

Moreover, there are application messages which refer to edge requests for additional resources for time-critical applications. They consist of message ID, node ID, application ID, resources needed to perform a task, and a short description of a task.

Lastly, some media messages are referred to as audio and video content for passengers and they consist of message ID, node ID, type of media content, requested resources, and media content for processing.

Considering the size of the described messages, we could easily divide them into three classes: small-sized, medium-sized, and large-sized messages. Urgent and essential messages would be considered small-sized messages as they contain basic ID information and short notice about the surrounding traffic situation. Further, application messages and some of the media messages (that are smaller than 20 MB) could be classified as medium-sized messages as they usually consist of some detailed information about the application that needs to be executed such as ID, needed resources, and description. Moreover, media messages greater than 20 MB are considered large-sized messages. That includes transferring large files with video and/or audio content.

B. Proposed System Workflow

In the following section workflow of the proposed architecture will be explained. The proposed FOGO architecture consists of the following components:

a) Composing the vehicle map
b) Scan the vehicle map for the available fog resources (edge vehicles)
c) Processing requests
d) Handling the idle areas
e) Distribution of urgent messages (fRSU)

The algorithm used to simulate the fog message manipulation is presented in Algorithm 1.

Algorithm 1: FOG RSU/TAXI/BUS aplication

```
OnMsgRcvd();
if Msg.GetType() == SendBeaconMsg then
    SendToAll(Msg);
    SendToCloud(Msg);
else
    ProcessMsg();
    SendResultBack();
end
```

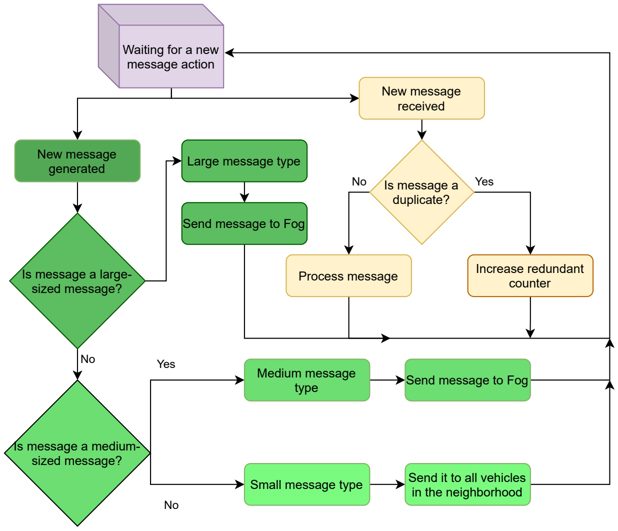

Fig. 2: Flowchart for handling messages on the edge vehicle

It is necessary to distinguish messages by their size to process them on the appropriate system. The flowchart for handling messages on the edge vehicle is presented in Figure 2. The algorithm for generating messages on edge vehicles is shown in Algorithm 2. The algorithm for processing messages on edge vehicles is presented in Algorithm 3.

Algorithm 2: EDGE CLIENT APPLICATION

```
OnGenMsg();
DiscNeighb();
if $Msg.GetSize() > 20MB$ then
    Msg.SetType(TypeLarge);
    SendToFog(Msg);
else if $Msg.GetSize() > 2MB$ then
    Msg.SetType(TypeMedium);
    if $RSUinRange()$ and $ProcessingTimeOnEdge < TransferProcessingTimeFog$ then
        SendToFog(Msg);
    else
        ProcessMsg();
    end
else
    Msg.SetType(TypeSmall);
    if $Msg.GetPurpose == request$ then
        SendToFog(Msg)
    else
        SendToAllNeighbours(Msg);
    end
end
WaitForResult();
```

Algorithm 3: EDGE CLIENT APPLICATION

```
OnMsgRcvd();
MesageStatus = OK;
RedundantMsgCounter = 0;
for each item $i$ in $sizeof(KnownIDs)$ do
    if $Msg.GetID() == KnownIDs[i]$ then
        RedundantMsgCounter++;
        break;
    end
end
if $Msg.GetType() == TypeSmall$ then
    ProcessMsg();
    if $distanceVehicle >= (maxDistanceNeighbourVehicle / 2)$ then
        Forward(Msg);
    if $Msg.GetPurpose == response$ then
        MeasureMsgTransferTime();
    end
end
```

C. Proposed Metrics

In order to precisely determine the advantages of using FOGO, the following metrics will be taken into consideration: an average number of successfully sent messages (aV), edge and processing ratio, fog processing efficiency, and cloud processing efficiency. To acquire metrics, it is necessary to measure the following parameters during the simulation: data network traffic overhead ($dNtO$), which stands for the total number of messages sent during the simulation; the number of received messages (nRm); and the average time for processing messages ($avTime$). The number of received messages will be calculated considering message ID. Redundant messages will not be taken into the calculation.

In order to provide generally useful and realistic data, both urgent and essential messages will be consolidated to determine the coverage of small-sized messages. Coverage by small-sized messages will be described using the following equation:

$$aV_{small} = \frac{nRm_{small}}{dNto_{small}} \tag{1}$$

Furthermore, when talking about medium-sized messages, application requests will be counted to determine the edge and fog processing ratio, PRmedium. It is necessary to count medium-sized messages processed in the fog, as well as all the medium-sized messages generated during simulation.

$$PR_{medium} = \frac{nRm_{medium}}{dNto_{medium}} \tag{2}$$

Moreover, when talking about large-sized messages, media requests will be counted to determine the coverage by large-sized messages. Cloud delivery and processing efficiency by large-sized messages will be described using the following expression:

$$aV_{large} = \frac{nRm_{large}}{dNto_{large}} \tag{3}$$

As was stated above, the influence of clustering on the number of redundant messages will be considered along with message ID which will be used to leave out redundant messages. In that manner, the overall network coverage will be measured. This parameter is described with the following equation:

$$aVrM = \frac{rM}{dNto_{total}} \tag{4}$$

Furthermore, fog processing efficiency Ef (fog) stands for the quotient of average time needed for processing medium-sized messages ($avTime_{med}$) on fog and edge, where time for processing on fog also includes transfer time to fog.

$$E_f(FOG) = \frac{avTime_{med}(EDGE)}{avTime_{med}(FOG)} \tag{5}$$

Simulation results are expected to show that the proposed algorithm efficiently reduces the network congestion by processing messages with a unique ID, consequently decreasing the load on the core network as well as improving the efficiency of data distribution in the fog layer.

IV. Performance Anaysis and Discussion

In simulation conducted via Veins, SUMO, and OMNET++ we compared stated metrics in case of using the aforementioned FOGO architecture. In this section, the results of a simulation performed on the Luxembourg map are presented. Simulations are performed in Veins, an open-source simulation framework, and are based on the LuST scenario which represents realistic 24h traffic on the Luxembourg map [12], [13]. Furthermore, by analyzing the LuST vehicle movement, we deployed 35 RSU fog nodes across different parts of the city, and 70 dynamic fog nodes which are representing taxi and bus vehicles.

There were six conducted simulations. Every simulation lasted for one hour. Every 10 minutes of the simulation, proper statistics were generated to determine an average of all measurements.

a) Network coverage In Figure 3, an increase in the total number of messages generated during simulation is shown. Every 10 minutes several thousand messages were generated. A large proportion of those messages were successfully processed in every step of the simulation which is shown in Figure 4. Furthermore, it is important to say that an average of 35,057 messages were generated in each simulation and an average of 29,132 were successfully processed which resulted in network coverage of 83.1 %. Moreover, for small-sized messages, it was measured that an average of 27,598 small-sized messages were generated, and 20,153 were successfully processed, which leads to 73 % of successfully processed small-sized messages.

b) Edge and fog processing ratio During the simulation, it was measured that an average of 4242 generated medium-sized messages were sent to fog, whereas an average of 10122 medium-sized messages were generated in total. Therefore, it is calculated that the edge and fog processing ratio is around 41.91 %, which leads us to the conclusion that fog is the practical solution for message management, and also, it is very beneficial in a financial manner since it can process more than 40 % of all messages in the network which will certainly reduce the load of the cloud.

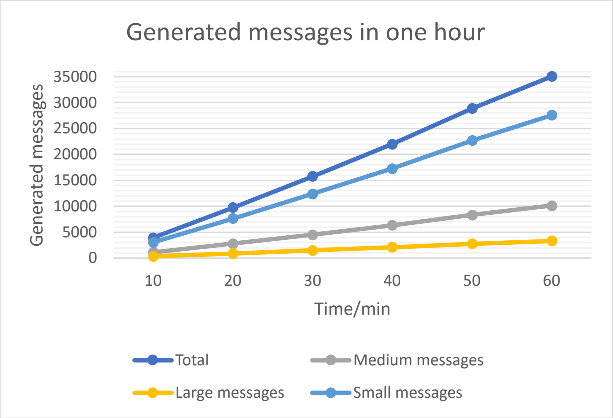

Fig. 3: Trend of generated messages during simulation

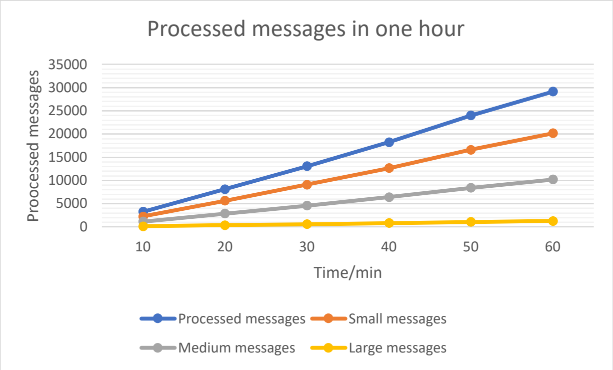

Fig. 4: Trend of processed messages during simulation

c) Fog processing efficiency Fog processing efficiency is a very important parameter since it tells how much faster a certain fog system can process messages that demand high computing power and could easily overload the edge vehicle. In our research, it is measured that the average time for edge vehicles to process a message is 46.18 s whereas the average time for the fog to process a message is 14.482 s. This measurement did not include any of the mechanisms for deleting messages that were not processed in a certain time, all medium-sized messages that did not get lost in the transfer were processed. It was established that fog is close to 320 % more efficient compared to edge vehicles. For example, if a fog can process a video sequence of 1 minute in 20 seconds, it would take more than a minute for an edge vehicle to do the same.

d) Cloud delivery and processing efficiency Cloud delivery and processing efficiency are counted based on successfully processed large-sized messages on the cloud compared to all generated large-sized messages in simulation. In our simulations, there is an average of 3348 large-sized messages created and 1273 messages that were successfully processed. That leads to the final result of

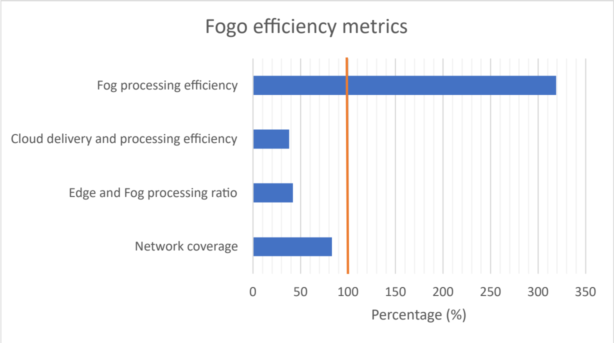

Fig. 5: Overall look on measured parameters

38.02 % of successfully processed large-sized messages. The rest of the messages were lost in the transfer or deleted due to incompetence in finding an RSU. Simulation results can be seen in Figure 5.

e) Discussion When we consider all measured and discussed parameters that are presented in this paper, we managed to prove that edge and fog computing systems are a great way to improve VANETs. Furthermore, we proved that these systems are very efficient, and considering our results in fog processing efficiency and network coverage, this area of research is truly an interesting field that will be one of the key elements toward completely autonomous vehicles. Nevertheless, considering the finances needed to develop VANETs, it is a relatively cheap solution compared to the price of the cloud. Our proposed solution is useful in various road traffic situations such as collisions, congestions, as well as everyday big data distribution in VANETs which appears to be one of the biggest problems that needs to be tackled with in process of developing autonomous driving.

V. CONCLUSION

In this paper, we concluded that fog and edge computing can be efficient in spreading messages across the city. As infrastructure is expensive, and most of the cities have public transportation that covers predetermined routes and taxi vehicles which can cover non-predetermined routes, with a good placing strategy of RSUs and using public transportation and taxi vehicles, most parts of the city could be covered cost-efficiently by using our proposed method.

Since there are a lot of messages being generated in the network with each minute and some of them are waiting in queue to be processed for some time, the time needed for medium-sized messages to be processed both on fog and edge is increasing drastically. As a solution to this problem, there is a possibility of implementing a mechanism that will delete messages that are waiting to be processed more than a certain, realistically set time. In such a manner, loss of important information will not happen due to the number of neighboring vehicles that will send similar requests to the fog and forward the response to other edge vehicles in the essential message.

Even though most of the medium-sized messages are still being processed on edge vehicles, by further improvement of our algorithm, as well as increasing the number of RSUs in simulation, it is possible to increase the efficiency of fog. Furthermore, adding more vehicles and infrastructure as fog nodes, such as trams and tram stations, and changing the parameters of message sizes could be fine-tuned to improve results. The drawback of this method is that most taxi vehicles and public vehicles could be on the same route, over-covering some parts of the city and not covering other parts. But looking at the economic benefits of this solution this could be a necessary trade-off between fog quality of service and infrastructure price.

REFERENCES

[1] W. B. Jaballah, M. Conti, and C. Lal, “A Survey on Software-Defined VANETs: Benefits, Challenges, and Future Directions,” *arXiv:1904.04577 [cs]*, May 2019.

[2] J. Grover, A. Jain, S. Singhal, and A. Yadav, “Real-Time VANET Applications Using Fog Computing,” in *Proceedings of First International Conference on Smart System, Innovations and Computing*, ser. Smart Innovation, Systems and Technologies, A. K. Somani, S. Srivastava, A. Mundra, and S. Rawat, Eds. Singapore: Springer, 2018, pp. 683–691.

[3] H. Singh, V. Laxmi, A. Malik, and Isha, “Fog Computing as Future Perspective in Vehicular Ad hoc Networks,” in *Fog, Edge, and Pervasive Computing in Intelligent IoT Driven Applications*. IEEE, 2021, pp. 177–192.

[4] R. J. Atwa, P. Flocchini, and A. Nayak, “A Fog-based Reputation Evaluation Model for VANETs,” in *2021 International Symposium on Networks, Computers and Communications (ISNCC)*, Oct. 2021, pp. 1–7.

[5] J. C. Nobre, A. M. de Souza, D. Rosário, C. Both, L. A. Villas, E. Cerqueira, T. Braun, and M. Gerla, “Vehicular software-defined networking and fog computing: Integration and design principles,” *Ad Hoc Networks*, vol. 82, pp. 172–181, Jan. 2019.

[6] G. Sun, Y. Zhang, D. Liao, H. Yu, X. Du, and M. Guizani, “Bus-Trajectory-Based Street-Centric Routing for Message Delivery in Urban Vehicular Ad Hoc Networks,” *IEEE Transactions on Vehicular Technology*, vol. 67, no. 8, pp. 7550–7563, Aug. 2018.

[7] K. Dolui and S. K. Datta, “Comparison of edge computing implementations: Fog computing, cloudlet and mobile edge computing,” in *2017 Global Internet of Things Summit (GIoTS)*, Jun. 2017, pp. 1–6.

[8] R. Shahin, A. A. El-Moursy, S. M. Saif, H. M. Abbas, and S. M. Nassar, “Fog Node Optimum Placement and Configuration Technique for VANETs,” in *2020 International Conference on Communications, Signal Processing, and Their Applications (ICCSPA)*, Mar. 2021, pp. 1–6.

[9] K. Bousselmi, Z. Brahmi, and M. M. Gammoudi, “Cloud Services Orchestration: A Comparative Study of Existing Approaches,” in *2014 28th International Conference on Advanced Information Networking and Applications Workshops*, May 2014, pp. 410–416.

[10] F. Zhang, H. Liu, Y.-W. Leung, X. Chu, and B. Jin, “CBS: Community-Based Bus System as Routing Backbone for Vehicular Ad Hoc Networks,” *IEEE Transactions on Mobile Computing*, vol. 16, no. 8, pp. 2132–2146, Aug. 2017.

[11] P. H. Cruz Caminha, F. Ferreira da Silva, R. Gonçalves Pacheco, R. de Souza Couto, P. Braconnot Velloso, M. E. Mitre Campista, and L. H. M. K. Maciel Kosmalski Costa, “SensingBus: Using Bus Lines and Fog Computing for Smart Sensing the City,” *IEEE Cloud Computing*, vol. 5, no. 5, pp. 58–69, Sep. 2018.

[12] C. A. R. L. Brennand, A. Boukerche, R. Meneguette, and L. A. Villas, “A novel urban traffic management mechanism based on FOG,” in *2017 IEEE Symposium on Computers and Communications (ISCC)*, Jul. 2017, pp. 377–382.

[13] L. Codeca, R. Frank, and T. Engel, “Luxembourg SUMO Traffic (LuST) Scenario: 24 hours of mobility for vehicular networking research,” in *2015 IEEE Vehicular Networking Conference (VNC)*, Dec. 2015, pp. 1–8.