University of Arkansas, Fayetteville
**ScholarWorks@UARK**

Computer Science and Computer Engineering Undergraduate Honors Theses

Computer Science and Computer Engineering

5-2019

# Applications of Fog Computing in Video Streaming

Kyle Smith

## Recommended Citation

Smith, Kyle, "Applications of Fog Computing in Video Streaming" (2019). *Computer Science and Computer Engineering Undergraduate Honors Theses*. 68.
https://scholarworks.uark.edu/csceuht/68

# Applications of Fog Computing in Video Streaming

An Undergraduate Honors College Thesis

in the

Department of Computer Science and Computer Engineering
College of Engineering
University of Arkansas
Fayetteville, AR
May 2019

by

Kyle S. Smith

## Abstract

The purpose of this paper is to show the viability of fog computing in the area of video streaming in vehicles. With the rise of autonomous vehicles, there needs to be a viable entertainment option for users. The cloud fails to address these options due to latency problems experienced during high internet traffic. To improve video streaming speeds, fog computing seems to be the best option. Fog computing brings the cloud closer to the user through the use of intermediary devices known as fog nodes. It does not attempt to replace the cloud but improve the cloud by allowing faster upload and download of information. This paper explores two algorithms that would work well with vehicles and video streaming. This is simulated using a Java application, and then graphically represented. The results showed that the simulation was an accurate model and that the best algorithm for request history maintenance was the variable model.

## Table of Contents

# 1. INTRODUCTION

## 1.1 Problem

Companies like Tesla, Waymo, and Uber have made recent advancements in the area of self-autonomous vehicles. Self-driving transportation seems to be the future of the transportation industry. While it is difficult to know how soon this will become a norm, the problems need to be analyzed beforehand. Video streaming is overlooked in the area of vehicular entertainment, due to its current dangerous nature. However, with the rise of the autonomous vehicles, entertainment will be necessary for passengers in the near future. With increasing internet traffic and increasing video resolutions, there is not a straightforward solution that provides fast entertainment to passengers. The current option is the cloud, but it fails to address these options due to latency problems experienced during high internet traffic.

## 1.2 Objective

The objective of this paper is to show that fog computing is a viable solution to the issues of video streaming in vehicles. Fog computing seeks to come aid the cloud and improve speed problems. There are a number of variables with this scenario. This paper analyzes how changing variables such as number of cars and number of movies affect the experience of video streaming.

## 1.3 Approach

First, two algorithms were proposed for testing. Second, a simulation was developed using a Java application. This simulation allows for different inputs and shows the outcomes. Third, the outcomes of the simulation were plotted and analyzed in graphical form. Finally, the algorithms are assessed, and the best solution is determined.

## 2. BACKGROUND

While cloud computing has been useful for small scale problems, larger applications have suffered from latency problems due to the cloud. As the amount of data points increase from IOT (Internet of Things) devices, the original cloud paradigm crumbles under the pressure. Cisco’s explanation says it well, "Today’s cloud models are not designed for the volume, variety, and velocity of data that the IoT generates.” [1] The idea of bringing the cloud closer to the user has been theorized in academia and made popular by Cisco in the last decade. This is formally known as fog or edge computing.

Fog computing has several reasons for gaining popularity. Three of the major reasons are the advent of IOT devices, latency problems with the cloud, and increased mobility. We will analyze each here. However, while fog computing remains a viable option, there are problems of power consumption, security, and proper usage of resources. The number of possible applications of fog computing continues to expand. After analyzing the positives and negatives of fog computing, we will look at more specific applications.

IOT devices have been one of the largest reasons for the invention of fog computing. The main requirements for IoT are to minimize latency, conserve network bandwidth, address security concerns, operate reliably, collect and secure data across a wide geographic area with different environmental conditions, and move data to the best place for processing. Cloud computing fail nearly all of these requirements. Fog computing happens when data is collected at the extreme edge; and with IOT devices, thousands or millions of things across a large geographic area are generating data. It is necessary to analyze and act on the data in less than a second which makes fog computing a viable option. [1]

Another push for fog computing has come predominately from its ease of mobility. Some possible applications theorized include smart grids, smart traffic lights with smart cars, wireless sensors with actuator networks, decentralized smart buildings control, and software-defined networks. There will be several security issues in fog computing that do not exist in a good cloud model. The main security issues are authentication at different levels of gateways. For the smart grid, a user could damage the readings that are gathered by the smart meters. There are some solutions to authentication such as public key infrastructure, Diffie-Hellman key exchange, encryption, and intrusion detection algorithms. Furthermore, the man-in-the-middle attack could occur in the fog computing paradigm. [2]

Two models have been proposed for fog computing. The first is independent fog computing where the device communicates with the cloud server. The second model is interconnected fog computing where the devices consult with one another. [2] This second model has been proposed as a way to increase security for the users. Instead of sending information between servers or using an encrypted cloud service, the information is kept secure in a fog network which increases privacy for the user. [3]

With the advent of fog computing, there needs to be an architecture and load balancing algorithm to handle user requests. Two of the biggest restraints that we run into are with latency and power. The main concern is allocating radio and computational resources that both satisfy the greatest number of user requests possible, as well as, keeping fog node power usage and process complexity minimal. To offset this problem, there has been the idea of small cell computing to satisfy the demands. Several algorithms have been proposed with the small cell model. The best algorithm gave priority to tasks that had the lowest latency gains and formed clusters to address the power problems. [4]

Further work has been done with load balancing from the angle of resource allocation. Resource allocation has been a common topic in the area of hardware and especially with operating systems. With this comes a number of algorithms that have been proposed, some writers believe that proper resource allocation is the key to efficiency in the area of cloud and fog computing. The ERA or Efficient Resource Algorithms has been proposed. Every request is added to a list that is managed by a Fog Server Manager or FSM. Tasks are given to fog node processors to run to competition or split into smaller tasks based on availability. Finally, if none of the fog processors are available, the request is sent to the cloud and handled there. [5]

Security is a common problem in the area of fog computing. Several models have been proposed to combat this. One approach involves using offensive decoy technology, which sends misinformation to possible malicious users. [6] Other solutions include a multi-layered approach with encryption and decryption keys. This seems to be one of the most extensive articles on security in the area of fog computing. This approach even incorporates a breadth-first search algorithm to help with the issues of load balancing. [7]

The number of possible applications continue to increase with fog computing. Several have been proposed in articles such as augmented reality, improved website performance, and big data. [8] There have even been some more theoretical applications such as in smart cities. The smart cities could benefit from pipeline analyzation technology, smart traffic, and even smart buildings. [9]

One proposed application of fog computing is with ECG's in medical care. The fog computing acts as a middle computing layer that has gateways and distributed databases in the healthcare industry. In this case, fog computing is not working to replace cloud computing but rather come along side it to help and aid it. The distributed databases contain a static look-up storage, a general-purpose storage, and a synchronized storage. The static database contains static and essential data and is not kept intact for the system administrators. The general-purpose database is used for fog computing and GUI. The synchronized storage is an inventory of temporarily environmental data. [10] This staggered approach has been popular in the area of fog computing. Fog computing is designed to move types of data to the optimal place for analysis. Most time-sensitive information is closest to the fog node. Moderate time information is in the aggregation node. Less time sensitive is in the cloud. [1]

The application that we are most concerned about is fog computing in the paper is the area of video streaming. The predominant proposed applications with fog computing and videos have been in the realm of image processing. Some examples include the use of camera networks and connected vehicles. This is helpful for the use of vehicle tracking through traffic cameras and traffic monitoring. This is important because it helps crime agencies and real-time traffic tracking. [11] This has been further studied in the area of smart cities. [12] Several surveys of fog computing have proposed video streaming enhancement as a possible improvement. One survey ambitiously says, "With the help of fog, we can achieve real-time processing and feedback of high-volume video streaming and scalability of service on low-bandwidth output data." [9] Further research has looked at power usage as a problem of our carbon footprint. It was proposed that fog computing could help to minimize our carbon footprint, specifically in the area of video streaming, through the use of joint resource allocation. [13]

Video streaming on highways can be inconsistent. With the increased number of video streaming, there needs to be a feasible solution to the latency problems encountered, especially on the highway. One possible solution includes the idea of video streaming and fog computing from Greyhound buses to provide an entertainment system. This would include the abilities to stream videos, play games, and view social networking sites. [14] Another possible solution has been in the area of video caching. Fog computing would be utilized to function as a cache for popular videos. This would help reduce user traffic and latency problems because the most popular content is closest to the user. [15] Edge caching has been further developed in other papers. It has predominately become a need due to the coming of 5G wireless technologies. [16] This idea has been further developed in Information-Centric Networks (ICN) specifically. [17]. However, the idea could be merged well with video streaming. This idea of edge caching is specifically important in our area of study because it seems to be one of the most effective solutions to delivering fast video streaming to people on the road.

# 3. MODEL

There will be two models proposed in this paper and two respective algorithms. The first model proposed is referred to as the fixed coordinator model. The second model proposed is referred to as the dynamic coordinator model. Both of these algorithms come with a number of assumptions, which are laid out below.

Fog nodes are nodes that act as an intermediary between the user and the cloud. Fog nodes have a circular range that they cover around them. Users within this range can communicate directly with that fog node. However, if the user is outside the range of the node, the node is unbale to communicate directly with the user. The fog node that the user regularly communicates with is known as the coordinator. This is because the fog node is responsible for storing personal information, keeping up with the token for every user, and storing the request history of the user. This coordinator is able to identify the user based off the token. For example, if a user identified as 43 was going down the road and contacts fog node number 3, then the token could be identified as $T_{43,3}$. In this case, the token has a double function of both identification and verification.

Fog nodes are placed along a stretch of road, so every car travelling is able to contact a fog node seamlessly. These fog nodes are not always equidistant due to the varying nature of the road. Due to curvy roads, it is possible that one fog node covers more stretch of road than another. For example, suppose there are three nodes along a stretch of road that is 400 miles long. Each fog node covers a 100-mile radius. Node 3 in this picture is able to cover 200 miles of road due to the curvy nature of the road. Therefore, not every node is set to be equidistant apart. This can be seen in the following figure (*Figure 1*):

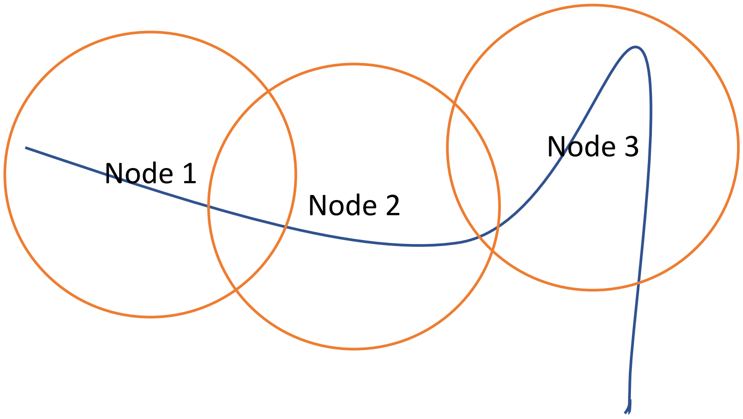

*Figure 1. Road and Fog Nodes to show that nodes have overlap and are made to fit the curvature of the road.*

In order to determine the best fog node, there needs to be an algorithm that determines a way for the user to communicate with a fog node that provides the most continuous amount of uninterrupted service. Without this, the idea of a fog node is meaningless because latency problems will continue if the fog nodes are not optimally placed on roads. In order to calculate the optimal fog node, the function:

$$getOptimalNode(carLocation, neighboringNodes)$$

is called. This function takes in two parameters $carLocation$ and $neghboringNodes$. The car location is kept up with through the user's navigation system within the device. The neighboring nodes indicate the nodes within a reachable geographic range of the user. For example, if a car is travelling along a curvy road, there may be a node farther ahead that is within range of the user, but this node may not provide continuous service to the user. Consider the following figure (*Figure 2*):

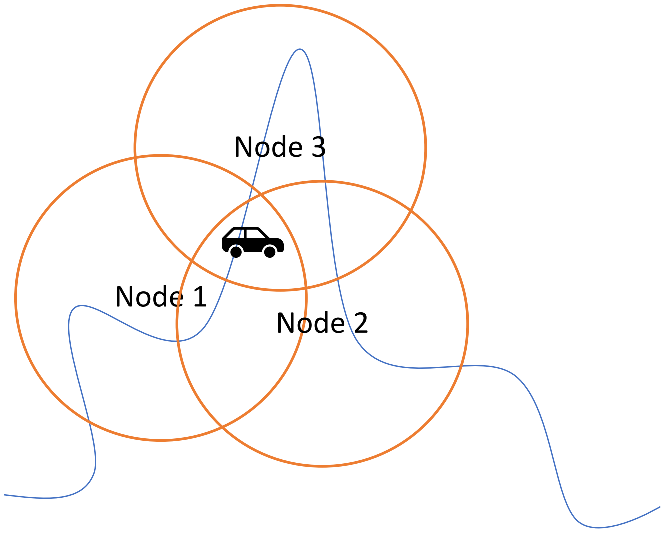

*Figure 2. Car and Fog Nodes on road to show that the optimal fog node provides best uninterrupted connection.*

As can be seen in the figure, the car is in range of both Node 1, 2, and 3. While Node 2 is farther ahead on the path. The $getOptimalFogNode()$ function would return Node 3 because it provides the best uninterrupted connection from the location of the car to the location of the fog node. This is to help minimize the number of requests that are made between the fog node and the users and maintain longer sessions.

## 3.1 User Authentication

The security algorithm is very similar between the two models and will be discussed before the individual algorithms are placed forward. First, the user credentials are sent to the fog node to send for validation in the cloud. During the cloud validation, a temporary token is established. A temporary token helps to ensure that the communication between the fog node and the car are not severed. Also, a temporary token helps to keep up with the location of the user while being authenticated. By having the temporary token, it will allow other fog nodes to know that the user is being validated. The temporary token helps to ensure the communication can be trusted between the user and the fog node. For example, if the user moves out of range of one fog node, the user can still communicate with the next nearest in-range fog node. If the user is not able to be validated, then the temporary token is revoked, and the car is unable to communicate further with the node. If the user is successfully validated, then the transaction is continued, and an official session token is granted. This token method is useful because it helps to establish a strong and secure session between the user and the fog node. The token includes the user's information and the issuer's information. Because if something is wrong, the issuer can be contacted. If there are malicious problems with a certain user, token access can easily be revoked. The token allows for access to all fog nodes in a geographic region. Each token is unique to the user, so tokens cannot be borrowed between users. This helps to establish a steady connection between the user and the fog node. As the car continues to travel, the token is passed from one fog node to another. The token is encrypted using a common key shared by all fog nodes. Thus, any fog node can decrypt the token and verify the user information. Tokens need to be renewed during new travel sessions. This can be set on a certain time limit. For example, after 24 hours, the token is no longer valid, and the user must reauthenticate with the cloud. This will be important in protecting the nodes from being tampered with from trusted users.

The next important step in the process is encryption and keys. User information between the user, fog nodes, and cloud are encrypted to prevent man-in-the-middle attacks. Encryption keys are used to help decrypt the message, so the message is readable to the recipient. There will be two separate keys used in the process.  The first key is established between the individual user and the fog nodes. The second key is established between the neighboring fog nodes. This separation of keys prevents users from intercepting other fog node messages from other users. This also helps to keep user data private during communications. Video content is not encrypted due to the increased computing power and slowed down speeds of encryption and decryption. This will allow for quicker recovery from the fog node. The token access to the fog nodes should prevent unwarranted access to the video content.

## 3.2 Fixed Coordinator Model

The fixed coordinator model derives its name because only one fog node acts as the coordinator for the user throughout a session. This coordinator does not always function to deliver content to the user but works to coordinate history and authentication with the cloud. The algorithm can be stated as follows:

1) A user ($u_i$) makes request for video content ($v$).
2) $getOptimalNode(carLocation,\ neighboringNodes)$ is called to return $n_j$, which is deemed the coordinator ($c_k$).
3) $c_k$ issues temporary token ($t_{i,k}$) for $u_i$ and passes to $u_i$.
4) $c_k$ sends $u_i$'s credentials to the cloud for verification.
5) If $u_i$ is authenticated, $c_k$ creates a new session token ($T_{i,k}$) for $u_i$ and passes to $u_i$. Else, revoke $t_{i,k}$ and end session.
6) $u_i$ presents $T_{i,k}$ to nearest fog node ($n_p$) in direction of travel.
7) $n_p$ verifies $T_{i,k}$ and gets request history from $c_k$.
8) $n_p$ checks buffer for $v_j$ (this is a part of the whole content $v$), which is the next block of content to be given based off of the request history.

9) If $v_j$ is not in buffer, broadcast request for $v_j$ to nearest fog nodes and cloud.

10) If a fog node has $v_j$, it delivers $v_j$ to $n_p$ and informs other fog nodes that received the request about this delivery. $n_p$ upon receiving $v_j$returns to $u_i$.

11) Inform $c_k$ of $v_j$ being provided and $c_k$ updates request history.

12) As $u_i$ nears end of $v_j$, REPEAT Steps 6 – 12.

### 3.3 Dynamic Coordinator Model

Similar to the fixed coordinator model, the dynamic coordinator model makes use of a fog node known as a coordinator. However, in the case of the dynamic coordinator, the coordinator has the ability to change every time the user moves along the path. At a given time, the optimal fog node acts as the coordinator. As the vehicle moves, the coordinator may change if the user moves out of range. This differs from the fixed coordinator model because multiple nodes can be the coordinator.  The algorithm can be stated as follows:

1) A user ($u_i$) makes request for video content ($v$).

2) $getOptimalNode(carLocation, neighboringNodes)$ is called to return $n_j$, which is deemed the coordinator ($c_k$).

3) $c_k$ issues temporary token ($t_{i,k}$) for $u_i$ and passes to $u_i$.

4) $c_k$ sends $u_i$'s credentials to the cloud for verification.

5) If $u_i$ is authenticated, $c_k$ creates a new session token ($T_{i,k}$) for $u_i$ and passes to $u_i$. Else, revoke $t_{i,k}$ and end session.

6) $u_i$ presents $T_{i,k}$ to nearest fog node ($n_p$) in direction of travel. $n_p$ becomes $c_k$. $n_p$ may not be different from $n_j$.

7) $n_p$ verifies $T_{i,k}$ and gets request history from $c_k$.

8) $n_p$ checks buffer for $v_j$ (this is a part of the whole content $v$), which is the next block of content to be given based off of the request history.

9) If $v_j$ is not in buffer, broadcast request for $v_j$ to nearest fog nodes and cloud.

10) If a fog node has $v_j$, it delivers $v_j$ to $n_p$ and informs other fog nodes that received the request about this delivery. $n_p$ upon receiving $v_j$returns to $u_i$.

11) Inform $c_k$ of $v_j$ being provided and $c_k$ updates request history and sends to $n_p$.

12) $n_p$ modifies token $T_{i,k}$ to represent correct $c_k$ and sends to $u_i$.

13) As $u_i$ nears end of $v_j$, REPEAT Steps 6 – 13.

## 3.4 Base Model

For the simulation, a base model was used because it simplified the use of the coordinator. This only affects the node interactions and should not affect the number of cloud calls or the node successes because the car maintains the request history. The base model uses a similar algorithm as the two above, besides the removal of the coordinator. The base model is not proposed as a good algorithm because it increases the necessary memory for the user and complicates the user's device.

## 3.5 Explanation

In this case, the scenario involves a user requesting video content in a smart car. The user is headed to a certain destination. This is handled through the user's navigation system in the smart car. Based off the user's destination and speed, the optimal fog node is chosen based off of the $getOptimalNode(carLocation,\ neighboringNodes)$ discussed above. The first time this is deemed as the coordinator. This coordinator remains the primary source of authentication and request history. After the user is authenticated, the session token is created. The token works to identify who the user is to the fog node and who the coordinator is. The token only lasts for the duration of the session and helps to make the authentication seamless across the different fog nodes.

The primary difference between the dynamic and fixed coordinator is the use of the coordinator. With the dynamic model, the coordinator changes every time the user makes a request. This is determined by the token used before. The token helps to identify the previous coordinator and is changed when the token is presented to the new coordinator. On the other hand, the fixed coordinator has only one fog node that acts as the coordinator.

The coordinator stores the request history, which ensures that the user is getting the correct video content. Video content can be large, especially with the increasing screen resolutions. This means that only small segments of video may be provided to the user at a given time. For example, if a movie is 90 minutes long, then the fog node may only provide the first 15 minutes. In order for the next request to provide continuous streaming, the next fog node must know to get the next 15 minutes of movie. Through the user of the coordinator, the request history ensures that the correct video content is being provided to the user.

A further concern may be with the request for video content to all of the fog nodes and cloud in Step 9. There may be multiple nodes that try to respond with appropriate video content. This is addressed by storing the nodes within the geographic area within the request itself. The first fog node within the geographic area to fulfill the request is able to tell the others to stop trying to fulfill the request, so there are not multiple nodes sending the same data.

The fog nodes work to keep the most popular video content in the fog nodes because these are the most frequently watched videos and movies. This helps to expand on the idea of edge caching discussed in the background session. If the most popular content is kept in the nodes, the response time will be quicker because the user will not have to contact the cloud for content every time. If the fog nodes’ buffer fills, then the least popular content will be deleted to allow room for new content.

# 4. EVALUATION

## 4.1 Approach

The approach used in this experiment was a simulation. This simulation was made using a Java application to run numbers. The model mimics the above model and allows for changing inputs to test the total node hops, cloud calls, and successful node calls. The code for this simulation can be found at https://github.com/Ksmith30/Thesis-Simulation. Because the testing of the variables does not require a difference in the models, a base model is used where the car keeps up with the request history of the car. For each of the graphs below, a total of 5 trials were taken and averaged together to have a fair representation of the data.

## 4.2 Variables

There was a total of 8 variables used in the simulation. Four of the variables will be discussed. There are two outputs for each simulation –cloud calls and node successes. A cloud call is when the requested video content is not found within the requested or neighboring fog nodes. A node success is when the requested video content is found within the requested or neighboring fog nodes. Before the graphs are shown, the variables need to be defined:

- **Miles Travelled** – the number of miles that each of the cars travel along the road until the destination is reached
- **Number of Movies –** the selection of movies that are available to a user when making a request
- **Length of Road (Miles) –** the length of road that the fog nodes cover

- **Number of Cars -** the number of vehicles that are on a road at a given time heading to a certain destination
- **Node Coverage (Miles) –** the amount of space that a node covers within a given area
- **Length of Video Segments (Minutes) –** the maximum length of each video segment retrieved from the fog nodes and cloud
- **Node Hops Allowed –** when content is missing from the fog nodes, this is the number of neighboring nodes that are searched on either side of the fog node for the video content
- **Max Storage (Minutes) –** the number of minutes that each node can store before the least popular movie is removed from the fog node.

Unless shown as a different value on the graphs, the following variables are constant:

| Miles Travelled | Number of Movies | Length of Road | Number of Cars | Node Coverage | Length of Video Segments | Node Hops Allowed | Max Storage |
|---|---|---|---|---|---|---|---|
| **400** | **50** | **500** | **100** | **50** | **5** | **2** | **500** |

*Table 1. Table of variables to show default values.*

## 4.3 Results

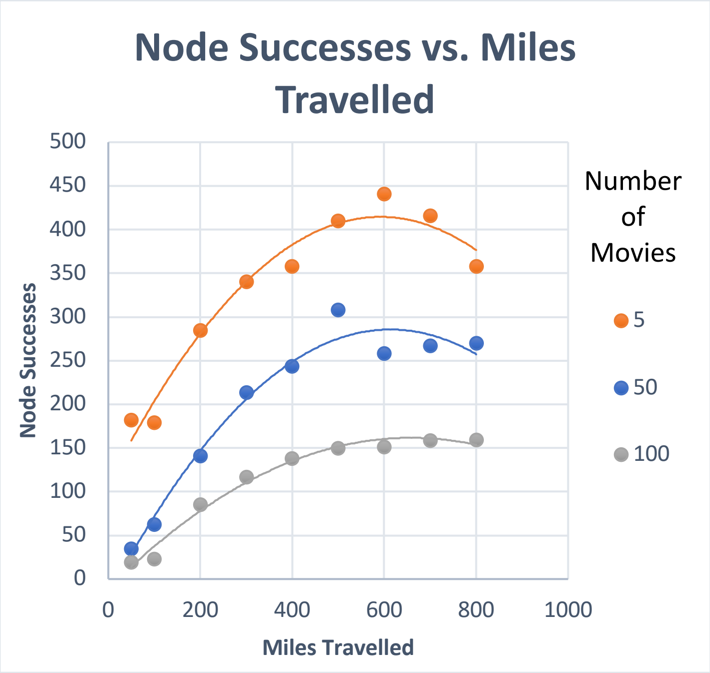

*Figure 3*

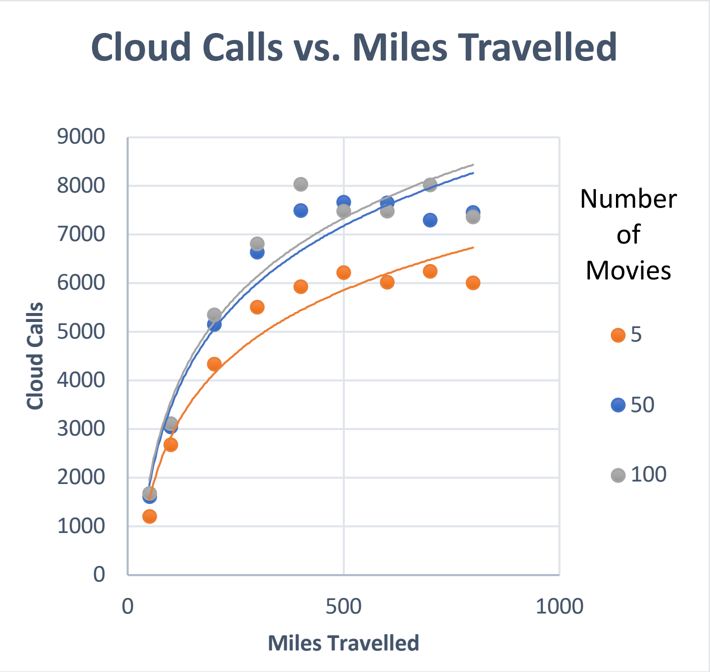

*Figure 4*

### Miles Travelled

*Figure 3* above shows that as the number of miles travelled increase the node successes tend to also increase. However, there reaches a point where it begins tapering off. The number of miles travelled increase the likelihood that a car will pass by a fog node, so the data is coherent. Looking at the different lines, the number of movies has a negative correlation with node successes. As the number of movies increases, the node successes decrease. Because the options have increased, this makes it more difficult for the node to serve the correct video to the user.

Next, the cloud calls are analyzed in *Figure 4*. These graphs are similar in the fact that as the number of miles travelled increased so do the node successes and cloud calls. However, as the number of movies increase so do the number of cloud calls. This is because the number of selections increase, and the chance of node failure also decreases.

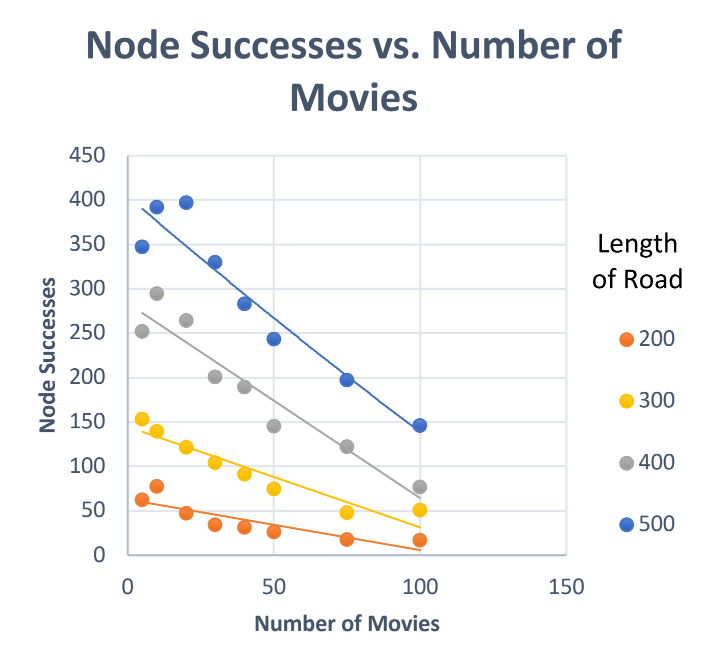

*Figure 5*

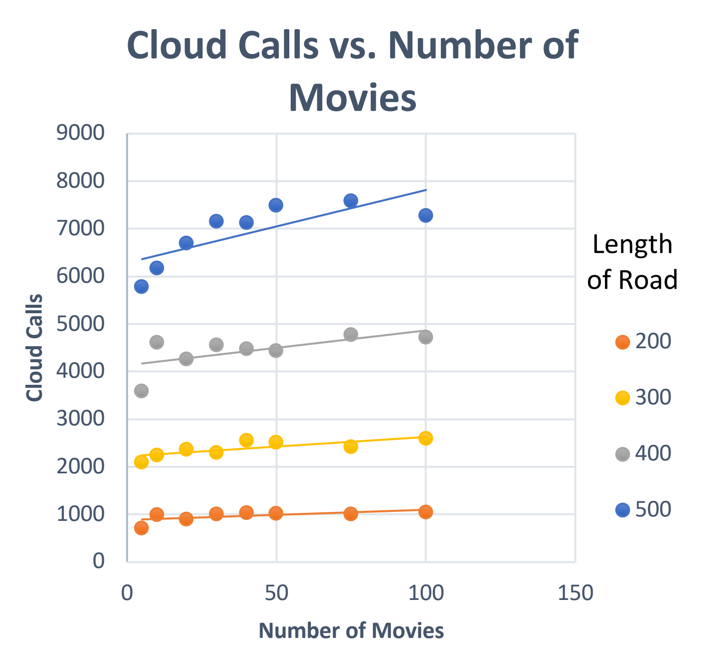

*Figure 6*

## Number of Movies

In *Figure 5,* the node successes decreased as the number of movies increased. While, the cloud calls increased slightly as the number of movies increased. The number of selections makes it difficult to fulfill the user's request more readily due to the increased variability. The length of the road, also, had an effect on how the node successes were affected. The node successes were higher the longer the road was. However, in *Figure 6*, the number of cloud calls was also higher the longer the road was. This is consistent because there are more nodes the longer the road is.  This means that both the node successes and cloud calls would increase because there are more opportunities for success.

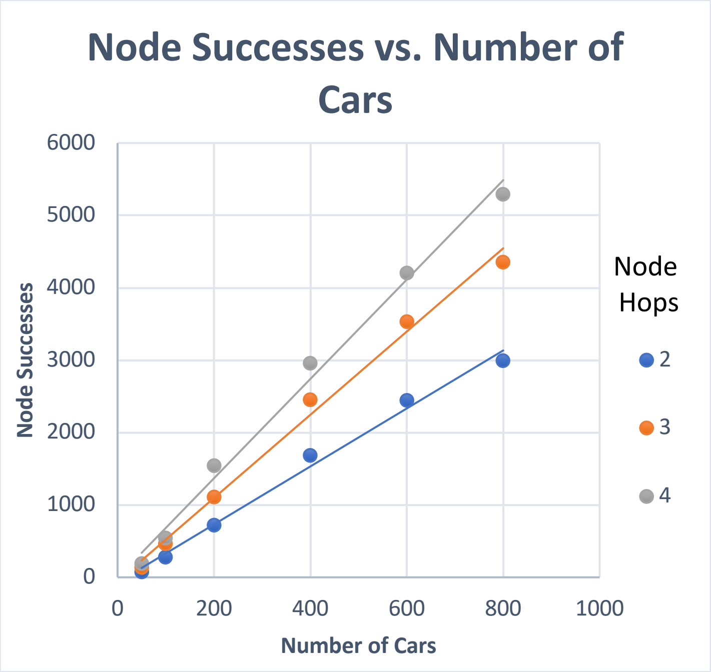

*Figure 7*

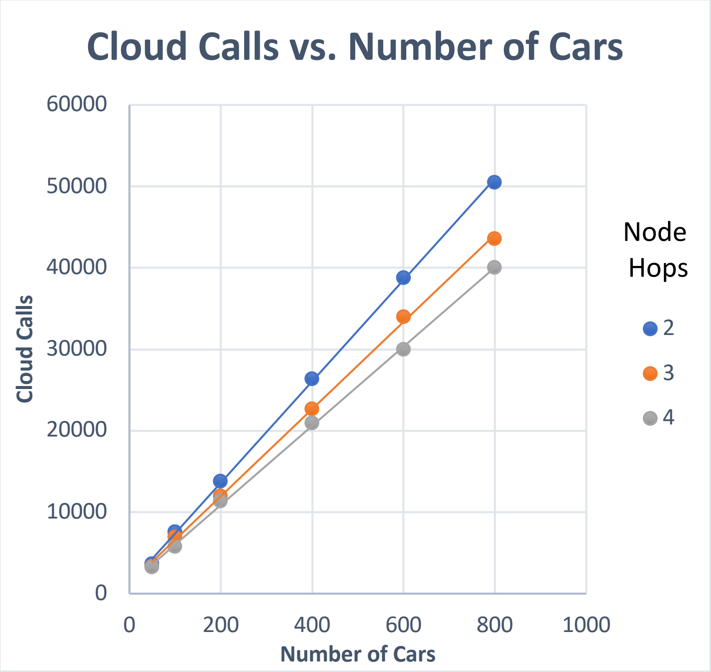

*Figure 8*

### Number of Cars

In *Figure 7,* node successes and the number of cars have a positive relationship. As the number of cars increase, the number of node successes do as well. This is because an increased number of cars allow for higher opportunity for a node to be successful. Node hops allowed also has a positive effect on node successes. As the node hops allowed increases, the number of node successes does so as well. Because there are more fog nodes checked in an area, the likelihood that a fog node has the requested video content increases.

In *Figure* 8, cloud calls have a positive correlation with the number of cars. As the number of cars increase, the number of cloud calls does so as well. Because there is a higher number of users, there will be a higher number of calls to the cloud. The node hops allowed has a negative effect with the number of cloud calls. Because the chances of finding the requested video content is higher, the number of cloud calls will decrease as well.

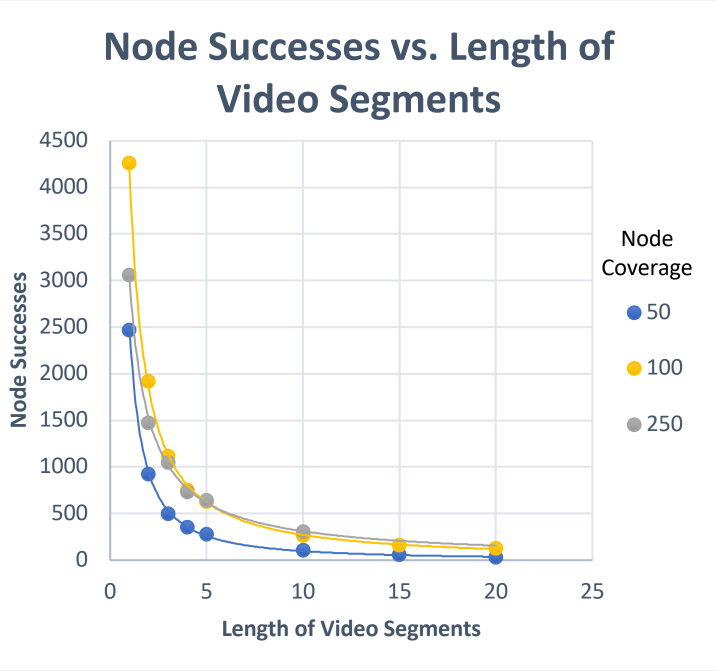

*Figure 9*

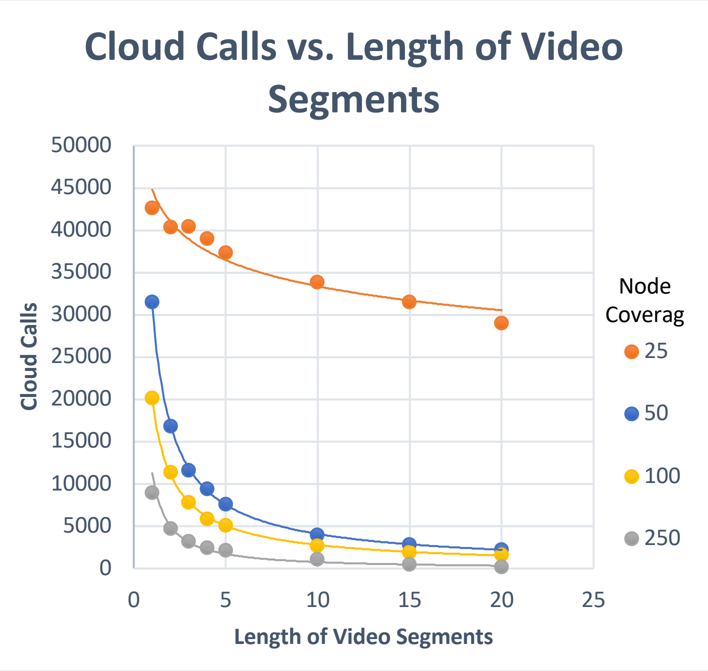

*Figure 10*

## Length of Video Segments

In *Figure 9,* the length of video segments and the number of node successes has a negative correlation. As the length of video segments increase, the number of node successes decrease. This is because the likelihood of finding the requested large segment of video is less than if the content is in smaller segments. Node coverage has an interesting effect on the number of node successes. The optimal node coverage is when 100 miles are covered. This means that there will be 4 nodes along the 400-mile stretch. Either increasing or decreasing the coverage seems to have negative impact on the number of node successes.

In *Figure 10*, cloud calls and the length of video segments have a negative correlation. As the length of the video segments increase, the number of cloud calls decrease as well. Because the user is occupied for longer intervals, the number of calls to the cloud is decreased. Node coverage has a negative correlation on cloud calls. As the node coverage increases, the number of cloud calls decrease. This is because the user is more likely to find the correct content in fewer nodes.

## 4.4 Comparing the Models

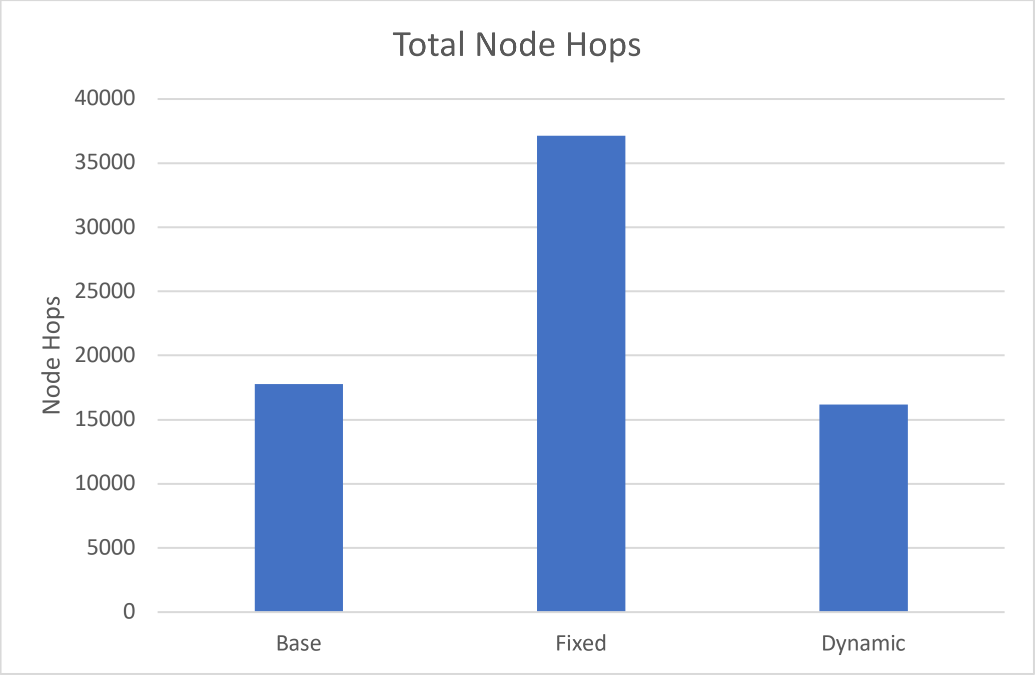

*Figure 11*

Finally, in *Figure 11,* the three models will be compared. The base model was used for testing and the user keeps up their own request history. In the case of the fixed and dynamic models, the fog nodes keep up with the request history of the user. The fixed model has the worst performance and the dynamic the best. Because the dynamic reduces the amount of traffic between the fog nodes. This will be the proposed model of choice. The main disadvantage of the dynamic model is the complexity of the fog nodes. Because the coordinator can change often, the nodes will have more communication overhead than the fixed or base model.

# 5. CONCLUSION

## 5.1 Summary

Autonomous vehicles are on the rise with no viable entertainment options. Video streaming is becoming more difficult to do efficiently with ever increasing video resolutions. With increased internet traffic and the move to the cloud, the cloud does not seem to be the effective long-term solution. Fog computing comes alongside the cloud and addresses the latency problems. Fog computing seems to be the best option for a reliable future.

Two algorithms were proposed to address these problems. These include a fixed and variable model. Both of these models make use of fog nodes that store video content along a road and communicate with the cloud. However, to simplify the user's experience, the video request history is stored in the coordinator. The coordinator stores video history to ensure a seamless experience for the user as they travel between fog nodes. In the fixed model, the coordinator is the first fog node that is connected to. However, in the variable model, the coordinator changes as the user travels along the road.

A simulation was developed in Java, and the variables were then analyzed finding that factors like number of cars, movies, and node distance can have a unique effect on the number of cloud calls and node successes. The above models were compared and then found that the variable option seems to be the best option because it has lowest number of node hops.

## 5.2 Contributions

This work shows the plausibility of fog computing in the realm of video streaming. Two algorithms and a simulation contribute to the areas of video streaming and fog computing.

These contributions do not only affect the world of autonomous vehicles but also improvements of video streaming in general.

## 5.3 Future Work

There is much work left to do in the areas of video streaming and fog computing. Further work would include expanding and improving the proposed algorithms and simulations. More practically, entertainment options need to be further tested in autonomous vehicles. This could be further developed with companies like Waymo, Tesla, Uber, and Netflix as they look to improve the user experience of video streaming.

# References

[1] https://www.cisco.com/c/dam/en_us/solutions/trends/iot/docs/computing-overview.pdf Last Accessed Date: 04/15/19.

[2] Stojmenovic, I., & Wen, S. (2014, September). The fog computing paradigm: Scenarios and security issues. In *2014 Federated Conference on Computer Science and Information Systems* (pp. 1-8). IEEE.

[3] Vaquero, L. M., & Rodero-Merino, L. (2014). Finding your way in the fog: Towards a comprehensive definition of fog computing. *ACM SIGCOMM Computer Communication Review*, *44*(5), 27-32.

[4] Oueis, J., Strinati, E. C., & Barbarossa, S. (2015, May). The fog balancing: Load distribution for small cell cloud computing. In *2015 IEEE 81st Vehicular Technology Conference (VTC Spring)* (pp. 1-6). IEEE.

[5] Agarwal, S., Yadav, S., & Yadav, A. K. (2016). An efficient architecture and algorithm for resource provisioning in fog computing. *International Journal of Information Engineering and Electronic Business*, *8*(1), 48.

[6] Stolfo, S. J., Salem, M. B., & Keromytis, A. D. (2012, May). Fog computing: Mitigating insider data theft attacks in the cloud. In *2012 IEEE symposium on security and privacy workshops* (pp. 125-128). IEEE.

[7] Puthal, D., Obaidat, M. S., Nanda, P., Prasad, M., Mohanty, S. P., & Zomaya, A. Y. (2018). Secure and sustainable load balancing of edge data centers in fog computing. *IEEE Communications Magazine*, *56*(5), 60-65.

[8] Yi, S., Li, C., & Li, Q. (2015, June). A survey of fog computing: concepts, applications and issues. In *Proceedings of the 2015 workshop on mobile big data* (pp. 37-42). ACM.

[9] Tang, B., Chen, Z., Hefferman, G., Wei, T., He, H., & Yang, Q. (2015, October). A hierarchical distributed fog computing architecture for big data analysis in smart cities. In *Proceedings of the ASE BigData & SocialInformatics 2015* (p. 28). ACM.

[10] Gia, T. N., Jiang, M., Rahmani, A. M., Westerlund, T., Liljeberg, P., & Tenhunen, H. (2015, October). Fog computing in healthcare internet of things: A case study on ecg feature extraction. In *2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing* (pp. 356-363). IEEE.

[11] Hong, K., Lillethun, D., Ramachandran, U., Ottenwälder, B., & Koldehofe, B. (2013, August). Mobile fog: A programming model for large-scale applications on the internet of things. In *Proceedings of the second ACM SIGCOMM workshop on Mobile cloud computing* (pp. 15-20). ACM.

[12] Chen, N., Chen, Y., You, Y., Ling, H., Liang, P., & Zimmermann, R. (2016, April). Dynamic urban surveillance video stream processing using fog computing. In *2016 IEEE second international conference on multimedia big data (BigMM)* (pp. 105-112). IEEE.

[13] Do, C. T., Tran, N. H., Pham, C., Alam, M. G. R., Son, J. H., & Hong, C. S. (2015, January). A proximal algorithm for joint resource allocation and minimizing carbon footprint in geo-distributed fog computing. In *2015 International Conference on Information Networking (ICOIN)* (pp. 324-329). IEEE.

[14] Luan, T. H., Gao, L., Li, Z., Xiang, Y., Wei, G., & Sun, L. (2015). Fog computing: Focusing on mobile users at the edge. *arXiv preprint arXiv:1502.01815*.

[15] Tran, T. X., Hajisami, A., Pandey, P., & Pompili, D. (2016). Collaborative mobile edge computing in 5G networks: New paradigms, scenarios, and challenges. *arXiv preprint arXiv:1612.03184*.

[16] Liu, D., Chen, B., Yang, C., & Molisch, A. F. (2016). Caching at the wireless edge: design aspects, challenges, and future directions. *IEEE Communications Magazine*, *54*(9), 22-28.

[17] Dabirmoghaddam, A., Barijough, M. M., & Garcia-Luna-Aceves, J. J. (2014, September). Understanding optimal caching and opportunistic caching at the edge of information-centric networks. In *Proceedings of the 1st ACM conference on information-centric networking* (pp. 47-56). ACM.