Journal of Computer Science

Original Research Paper

A Holistic Approach to Security, Availability and Reliability in Fog Computing

Abdulrahman Alshehri, Hazzaa Alshareef, Samah Alhazmi, Marwah Almasri and Maha Helal

College of Computing and Informatics, Saudi Electronic University, Riyadh 11673, Saudi Arabia

Article history
Received: 19-09-2023
Revised: 18-02-2024
Accepted: 13-03-2024

Corresponding Author:
Hazzaa Alshareef
College of Computing and Informatics, Saudi Electronic University, Riyadh 11673, Saudi Arabia
Email: h.alshareef@seu.edu.sa

Abstract: Cloud computing has become popular in recent years due to the considerable flexibility it provides in terms of its availability and affordability and the reliability of different software and services for remote users. Fog computing has also gained considerable attention in recent years from the research fraternity. Fog computing is an additional layer between the users of the cloud and the cloud infrastructure as a place that stores frequently used data in order to reduce latency, which might occur as a consequence of using cloud computing. It also provides easy access and management mechanisms to the devices located at the edge of the cloud, which leads to better performance when compared with cloud computing. Fog computing does, however, pose certain challenges, related to security, such as data breaches; availability, such as dealing with connectivity interruptions; and the reliability of fog resources and services. This study proposes a lightweight system that adopts the fog computing paradigm and addresses several of its challenges by, for instance, enhancing the security aspects of the whole system by validating nodes that join the fog layer before serving the end user. In addition, the proposed system provides better availability and reliability for fog computing and its associated services by capturing and tracking the progress of tasks and being able to resume once an interruption is detected. Experimental results validate the feasibility of the proposed system in terms of its enhanced security capabilities and time cost. This is achieved by using several security techniques which result in allowing only approved devices to join the fog layer. The results also demonstrate the capability to execute tasks even if an interruption is detected by resuming the remainder of the task through another fog node. The proposed solution is unique in the sense that it provides a simple mechanism for implementation in real-world applications, especially in crowded places or when the mobility of users is high. It can also be enhanced further in several ways to address other predicaments related to fog computing.

Keywords: Cloud Computing, Fog Computing, Internet of Things (IoT), Public Key Infrastructure (PKI)

Introduction

With the rapid growth of network bandwidth and cost-effective, high-speed, interconnected devices, the trend for third-party systems and services has been bolstered as never before in recent years. Cloud computing is a technology that enables servers and electronic services to be hosted online and accessed from anywhere. Among its many advantages, however, there is a major hurdle, in that utilizing this technology may affect latency due to the enormous amount of data transferred between the internet and end users. Having a middle layer between the cloud and end users would greatly enhance the latency and transfer process. This middle layer defines the paradigm of fog computing.

Fog Computing

Fog computing can be defined as an extended model from the cloud that uses the network edge to back up and support the cloud. It also provides data storage and application services. The terms fog computing and edge computing are similar and used interchangeably in the industry. However, the main difference between fog

and edge computing is that fog computing is decentralized, as described by Abdulkareem *et al*. (2019) and the data are stored on local machines and storage systems instead of routing them all to the cloud, as in the case of edge computing.

As part of the fog layer, any of the computing devices, such as web servers, storage servers, or hubs, can act as fog nodes. As stated by De Donno *et al*. (2019), fog computing is a highly virtualized platform that provides services such as computing, storage and networking. In their research, they discuss the foundations of and evaluate modern computing paradigms, including the cloud, Internet of Things (IoT) and edge and fog computing. In earlier work, Yi *et al*. (2015a) refer to fog computing using many different terminologies, such as mobile cloud computing, mobile edge computing, or an extension of cloud computing from the core network.

Adopting fog computing can enhance overall performance when compared to cloud computing in terms of bandwidth utilization, Quality of Service (QoS) assurance, emergency notifications, low latencies and supporting various sensor node types. It also enhances the real-time interaction between the fog and end users (Gia *et al*., 2015).

As well as the several advantages of fog computing, there are a number of associated challenges. According to Alrawais *et al*. (2017), one of the main challenges is the security of fog computing and the availability and reliability of its allied services. Many researchers have, for example, utilized fog computing to address security challenges, such as Denial of Service (DoS) attacks, due to the spread and distribution of fog nodes.

Kunal *et al*. (2019) interpreted fog as an intermediate layer that connects both cloud computing and the end users. They also discuss a five-layered data flow architecture that uses fog computing to provide services to end users. These services include energy lattices, MediFog, UXFog, a connected parking system and FoArgo. Baccarelli *et al*. (2017) refer to fog computing as a model that complements cloud computing through a series of connected nodes, which reduce the latency and improve the QoS level provided to users. Furthermore, there are myriad different security issues associated with fog computing, involving trust, authentication, secure communication, end-user privacy and malicious attacks.

Mukherjee *et al*. (2017) discuss the trust issues associated with fog computing and highlight that all fog nodes should be trusted before serving IoT devices. The authentication issue refers to the servicing process through which no fog node should join the network unless it is authenticated. For this, different researchers have used certificates and Public Key Infrastructure (PKI) for authentication purposes. However, these all have their associated vulnerabilities, which include the need for certificate life-cycle management, proper key management and the computational limitations pertaining to some IoT devices. Secure communication in fog computing is also considered another challenge. Since there are two types of communication, the first between fog nodes and IoT devices and the second between the fog nodes inside the network, the authors recommend using encryption techniques. However, there are a number of challenges associated with asymmetric encryption between IoT devices, such as processing capabilities, computational overhead and key renewal. Furthermore, due to the nature of fog computing and IoT, sensitive user data may be collected, such as time and location, which can invoke privacy issues.

Osanaiye *et al.* (2017) identify other security issues related to fog computing, such as access control and intrusion detection. Wang *et al*. (2015) also demonstrated one of the most critical security issues in fog computing, which is Man in the Middle (MitM) attacks. They were able to compromise a fog node and insert malicious code to prove the possibility of such an attack. In their work, Zhang *et al.* (2018) discuss other issues in detail, such as service availability, application security and technology sharing.

In addition to the above, other challenges associated with hardware and platform standardization are discussed by Zhanikeev (2015). Compatibility and adherence to standards is an issue that eventually violates the availability requirement due to the nature of fog nodes and IoT devices.

A lightweight protocol is proposed by Ni *et al.* (2017) to handle the reliability issue. Complex computing processes might take a long time due to IoT computing capabilities, which means there may be delays that have an eventual effect on reliability. According to Yi *et al*. (2015b), the reliability of fog computing can be improved by rescheduling checkpoints when tasks are failed. Similarly, Hou *et al*. (2020) proposed using a swarm of drones to act as fog nodes to improve reliability and latency issues. In addition, Jonathan *et al*. (2017) stated that reliability could be achieved by incorporating redundancy in task scheduling between computing nodes. Pereira *et al*. (2019) used two systems to assess the reliability of using fog computing: A city traffic anomaly system and bus stop arrival time estimation. The reliability of the two systems is validated through real mobility information.

In addition to the challenges referred to above, heterogeneity and QoS management are also important aspects when dealing with fog computing and are due to the dynamic environment between the nodes, as stated by Mouradian *et al*. (2017); Boonsong *et al*. (2023).

To summarize, the main challenges of fog computing are the following:

- Communication-related challenges, which include malicious attacks, access control and authentication

- Data-related challenges, which include secure communication and end-user privacy
- Hardware and platform standardization challenges, which include compatibility and adherence to standards

Main Contribution of the Research

This study proposes a three-tier architecture-based fog computing system that enhances the security aspect and can thwart illegitimate interaction and access to the system. It also provides a mechanism that ensures the stable availability and reliability of fog computing devices and services.

Materials and Methods

This section discusses some of the underlying system components and concepts that are related to the fog-computing environment in general and to the proposed work in particular.

System Components

Our proposed system consists of three tiers, namely, (i) Cloud layer, (ii) Fog layer and (iii) End-user layer, Fig. 1.

Cloud layer: The first and top layer is the cloud layer, which contains three main components: Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS). This layer has servers (data centers) that provide cloud services, related to both infrastructure and software, to remote users as and when needed.

Fog layer: The second layer in our system is the fog layer, which contains routers, hubs and servers. The main purpose of this layer is to extend the capabilities of cloud computing near the edge of the network, where devices and sensors consume and generate data that are crucial for the broader purpose of the cloud. Among the main tasks of this layer are bandwidth optimization, latency reduction, scalability, resilience, security and privacy, to name a few.

End-user layer: The last layer is the end-user layer, which consists of IoT devices, personal computers and smartphones. This layer serves as the front end and is responsible for communicating with the upper tiers of the fog and the cloud.

System Flow

Figure 1 represents the underlying concept of our proposed methodology. Let us assume that the user requests file A. If the requested file is hosted on one of the fog nodes, then the fog node will send the requested file A to the end user, as shown in links 1 and 2 in Fig. 1. On the other hand, let us assume that the user requests file B, whereby the requested file is not hosted on any of the fog nodes. In this case, fog nodes will connect to the cloud and request the data. Once received and downloaded by the fog node, the requested data (file B here) will be sent directly to the end user. This is depicted in links 3-6 in Fig. 1.

Security Mechanism

According to Yi *et al.* (2015b), one of the dominant security issues in fog computing occurs when an attacker inserts a compromised fog node as a legitimate one into the network, which enables the attacker to manipulate packets or insert a malicious code. To prevent this, our proposed methodology relies on an Authentication Node Table (ANT), which includes the MAC addresses of all nodes as a hash. Only the network administrator can access this table and insert a node's MAC address, generate a hash value and then insert it as a trusted node.

Table 1: Example of an authentication node table

N-name	Hash value	Attribute
Server A	19245bf01466c370dfa6ee674859 d7b29733c25b9a83f7f2cceb72d1fedf0f93	Trusted
Server B	8a4916c51ac43b307d102bd8986db 4ef61ff8b2cafc7951650cb9f63c27ef442	Banned
Server C	b1bc8326661128e448f72209d6c15 c35e7978cc77fb78013fca3e79db81f703c	Waiting

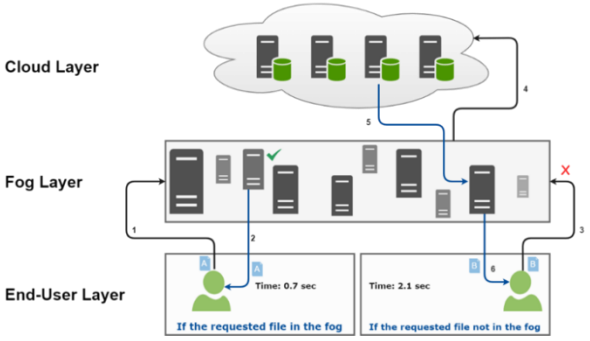


Fig. 1: Generalized architecture of the proposed system

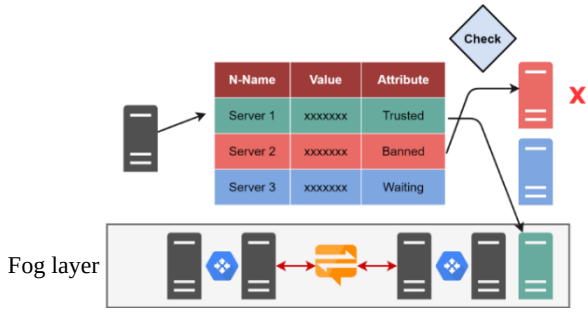


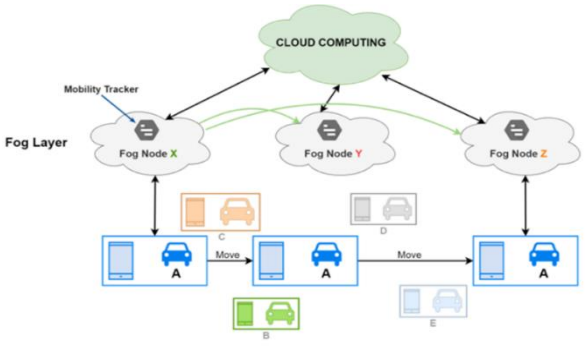
Fig. 2: Security mechanism of the proposed system

The ANT, an example of which is shown in Table 1, is hosted on a standalone server in the fog layer, which can be accessed from inside the network in order to improve security. In addition, synchronization is maintained between the ANT and the routers to make sure that even if the server fails, the routers can block the sending or receiving of requests from untrusted fog nodes.

Using the hashing mechanism, which is a one-way procedure, will make it nearly impossible for a hacker to retrieve the real MAC addresses of nodes, as shown in Fig. 2.

Availability and Reliability Improvement

Ensuring the availability and reliability of data as users are on the move is a crucial task, as this can have a tremendous effect on the overall performance. Users can change their locations while requesting a service from the fog nodes. Therefore, task execution should be maintained, as well as its reliability. To understand the process, let us assume that end-user EU1 is communicating with fog node FN-ID1 and FN-ID1 belongs to the fog network FNW-ID1. During the operation, FN-ID1 keeps sending a progress status through a protocol and, once FN-ID1 notices that EU1 starts to lose the signal, it will push a notification to all other related fog nodes informing them of EU1's state. This information is stored in a routing table. Once EU1 is connected to another fog node, the fog node will broadcast a signal to inform other nodes and fetch the corresponding data from the routing table. By doing so, we improve the availability and reliability of fog computing and do not lose the data that have already been processed. Figure 3 demonstrates an example of this mechanism along with the routing table.



Routing Table

Client	Data	Serve Fog	Node Receiver	Status
Client A	DBO	FN-X	FN-Y	In Progress
Client B	DBO	FN-Y	FN-X	Complete
Client C	DBO	FN-X		Pending

Fig. 3: Mechanism for the availability of data and the corresponding routing table

System Implementation

In our proposed model, Windows Azure is used at the cloud layer. MySQL is also used for the implementation of the database. For the fog layer, a machine with WampServer installed on it is used as a web server with PHP. Both iOS and Android are used at the end-user layer as long as the user uses an internet browser.

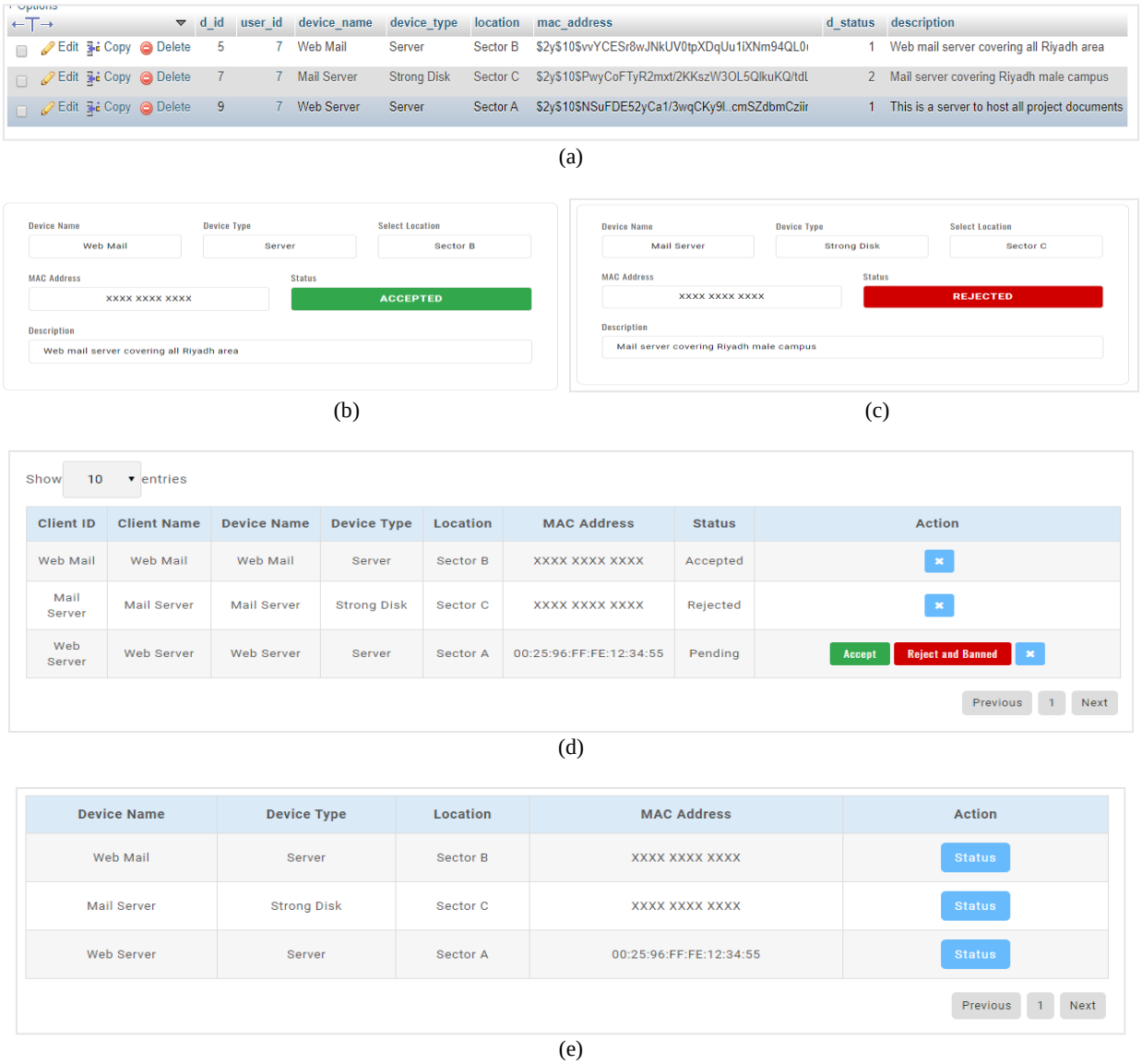
Fog Node Provider System

Any fog node can be registered as a fog node provider. After successful registration, new devices can be added, such as mail servers, web servers, or storage servers. Each fog node must also include the MAC address of each of the added devices. Once accepted by the administrator, all data will be stored in the database and the MAC address of the device will be hashed. The reason for using the hashing function here is to reduce the threat of unauthorized devices joining the fog layer devices. Another reason is that, unlike a hashed MAC, an encrypted MAC can be decrypted and thus there is a chance that an attacker might decrypt it and assign it to a malicious device. Our proposed approach provides several enhanced security and privacy measures, including reduced exposure to spoofing, protection against eavesdropping attacks and mitigating illegitimate device tracking risks.

Figure 4 shows screenshots from the actual implemented system. Figure 4(a) depicts entries of the devices in the database and their associated MAC addresses. Figure 4(b-c) demonstrates the interfaces for adding and rejecting devices, respectively. Moreover, the administrator can manage users who are registered in the fog nodes' providers' system by accepting/rejecting the submitted devices based on the requirements met. An interface is also provided for the admin side of the system for whether to accept a device or reject it. This is shown in Fig. 4(d). Finally, all the devices registered on the fog network, along with their current status, MAC address, name, type and location, are presented in a single list, in Fig. 4(e).

Availability Mechanism

A web application is developed to ensure the availability of the files requested by the user. It contains two pages. The first page is for uploading files and storing them on the server inside a specific folder. These files will be divided into four sections using a PHP function. The second page is for downloading the files with an implemented idea to ensure that the client that requests the file will receive the file in chunks. Therefore, the user can resume the downloading process in the case of any interruption.



	d_id	user_id	device_name	device_type	location	mac_address	d_status	description
Edit Copy Delete	5	7	Web Mail	Server	Sector B	$2y$10$vvYCESr8wJNkUV0tpXDqUu1iXNm94QL0	1	Web mail server covering all Riyadh area
Edit Copy Delete	7	7	Mail Server	Strong Disk	Sector C	$2y$10$PwyCoFTyR2mxt/2KKszW3OL5QlkuKQ/tdl	2	Mail server covering Riyadh male campus
Edit Copy Delete	9	7	Web Server	Server	Sector A	$2y$10$NSuFDE52yCa1/3wqCKy9l..cmSZdbmCziir	1	This is a server to host all project documents

(a)

Device Name: Web Mail
Device Type: Server
Select Location: Sector B
MAC Address: XXXX XXXX XXXX
Status: ACCEPTED
Description: Web mail server covering all Riyadh area

(b)

Device Name: Mail Server
Device Type: Strong Disk
Select Location: Sector C
MAC Address: XXXX XXXX XXXX
Status: REJECTED
Description: Mail server covering Riyadh male campus

(c)

Show 10 entries

Client ID	Client Name	Device Name	Device Type	Location	MAC Address	Status	Action
Web Mail	Web Mail	Web Mail	Server	Sector B	XXXX XXXX XXXX	Accepted	✖
Mail Server	Mail Server	Mail Server	Strong Disk	Sector C	XXXX XXXX XXXX	Rejected	✖
Web Server	Web Server	Web Server	Server	Sector A	00:25:96:FF:FE:12:34:55	Pending	Accept Reject and Banned ✖

Previous 1 Next

(d)

Device Name	Device Type	Location	MAC Address	Action
Web Mail	Server	Sector B	XXXX XXXX XXXX	Status
Mail Server	Strong Disk	Sector C	XXXX XXXX XXXX	Status
Web Server	Server	Sector A	00:25:96:FF:FE:12:34:55	Status

Previous 1 Next

(e)

Fig. 4: Screenshots of different activities and interfaces of the proposed system; (a) Storing data in the database; (b) Device acceptance; (c) Device rejection; (d) Admin interface for the actions against devices and (e) Status and summary of devices

To illustrate this, let us assume that user A sends a request to download a file at instance 1.1, in Fig. 5. The request will be transmitted to the fog layer and the fog layer will then check if the requested file is located in any of the nearby fog nodes. If the file is not available, the fog layer will request the file from the cloud and then the file will be downloaded to the nearest fog node and, finally, directed to the end user. As presented in Fig. 5, the fog node FN-x will start transferring file 1.2 to user A and will, at the same time, synchronize it to process 1.3. During transmission, user A loses the connection with fog node FN-x. Since FN-x can predict that user A is about to lose the connection by measuring the strength of the signal, it will broadcast a signal to all other nodes in the fog layer about the status and store the data. User A returns to the session at a different location. At this time, another fog node, FN-y, picks up the signal. FN-y will resume the transmission from the point at which user A lost the connection, as in instance 2.2 in Fig. 5. Similarly, once FN-y predicts a downgrade in the signal strength and the connection is lost, it will broadcast the status and data to the rest of the fog nodes in the fog layer, as in 2.3. Once user A is back in session 3.1 and requests the data from the fog, FN-z will pick up the signal and request the data, as in 3.2. Again, FN-z will keep updating until the file transmission is completed and user A has no more requests.

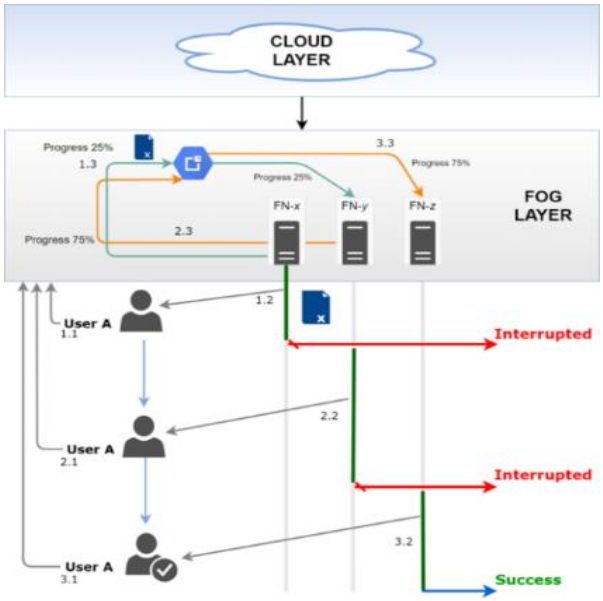


Fig. 5: System flow for the request availability mechanism

Results and Discussion

This section presents the experimental results and some related discussion. The proposed scheme is easy to implement for real-world applications and is simple in its adoption. The implementation details were discussed in the previous section.

Figure 6 shows the time measured in milliseconds (ms) for each part of the divided file from one of the testing cases. The requested file is split into four different parts based on the behavior of the user and the fog network. The image is a screenshot of the actual system. It can be seen from the figure that our proposed mechanism has maintained the service availability in fog computing, whereby the data requested are still available after the connection is interrupted by the fog node and the client. This is done by splitting the file into four parts and sending each part separately. All four parts are stored in the database and, once the user's connection returns, data transmission will resume successfully.

For illustration purposes, Fig. 7 presents the time-based cost in a bar chart representation. The chart clearly shows that all file parts have been downloaded successfully even with several connection interruptions and through different fog nodes after broadcasting the progress of the task (downloading a file). The interruption of the connection and the distance between the user and the fog node, as well as finalizing the task (merging all file parts), demonstrate the time needed to accomplish the whole task. Hence, a high degree of availability was achieved regardless of the interruption of the connection. In other words, there was no need to start the task from the beginning after the user was disconnected as the progress was broadcast to all fog nodes to resume the task.

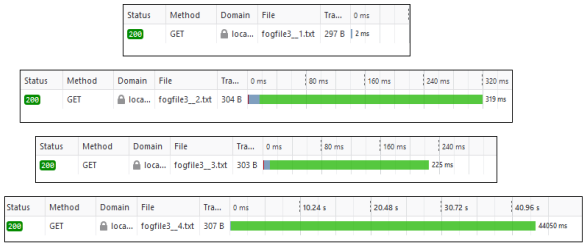


Fig. 6: Screenshot from the implemented system illustrating the time measured in milliseconds of the file access request

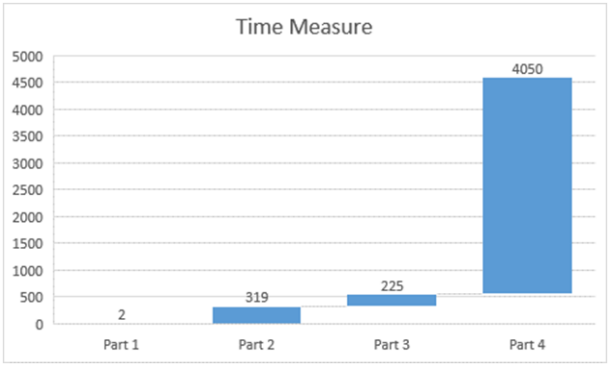


Fig. 7: Bar chart representation of the time-based cost in milliseconds for a file after being divided into four parts

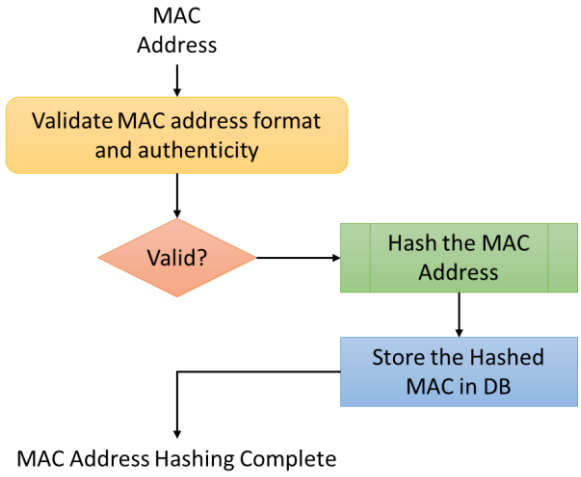


Fig. 8: Hashing mechanism of the MAC address

However, many factors can affect the values shown in Figs. 6-7, such as the file size, which can lead to a higher cost and longer task time; the internet connection, which could badly affect tracking the task progress; and the end-user device location and movement, which might require frequent connection to different fog nodes.

Figure 8 presents the mechanism to incorporate security in the system. In contrast to conventional systems, the proposed work incorporates hashing of the

MAC addresses of newly joined devices, as presented in Fig. 8. This allows the fog node provider to add new devices into the system, where the admin can accept/reject a join request. The MAC address of these devices will be hashed and only accepted devices are allowed to be a part of the fog layer subject to admin approval. By doing so, the system is able to verify the authenticity of the devices that join to ensure it is not spoofed.

Applying the above-mentioned hashing mechanism will tackle the challenges related to access control and preventing malicious attacks by ensuring only validated nodes can join the fog to serve the end user. However, this mechanism requires the accepting and rejecting of requests by an administrator, which could be automated by storing a predefined list of trusted devices in the fog. Then, once a join request is received from one of the trusted devices, acceptance will be granted automatically. Thus, trust challenges will be handled by applying this mechanism.

Conclusion

Fog computing emerged as a middle layer between cloud computing and end users due to the expansion of the use of internet-based services and issues related to cloud computing, such as latency. Fog computing provides many services in the middle tier, such as storage and processing. The advantage of fog computing comes from its location near the end user, which eliminates cloud computing latency. Fog computing has inherent issues, however, such as unauthorized nodes joining the fog layer, vulnerability to spoofing, availability and reliability.

This study presents solutions to these problems in a number of ways. First, the security issue is handled by presenting a system that depends on the hash value of the authorized MAC address. This method only allows devices accepted by the system administrator and depends on technology that adds more reliability and robustness to the fog layer. The second issue on which this study focuses is enhancing the availability of fog computing. The idea is to divide the data into several sections and keep the nodes in the fog layer updated about current clients. In the case of interruption, the status of the process and data will be broadcast to the other nodes. Once any new node is found to be connected to the client, the data will be retrieved from the fog layer and the service will be resumed to the end user.

In the future, we are planning to examine how the proposed system could improve performance when compared to other existing solutions. Moreover, this study could be enhanced by incorporating artificial intelligence in several ways, such as:

- Automatically managing fog nodes joining and leaving, such as by accepting new fog nodes in the fog layer based on predefined criteria or information about the node itself
- Improving the actions that need to be taken in the case of connection interruption, such as to which fog node the task needs to be transferred or which task it is better to terminate

Acknowledgment

The authors of this manuscript would like to express their appreciation and gratitude to their university for supporting this research.

Funding Information

The authors have not received any financial support or have any funding to report.

Author's Contributions

Abdulrahman Alshehri: Acquisition of data, investigation, software, methodology, originally drafted preparation and approved the version to be submitted and any revised version.

Hazzaa Alshareef: Conceptualization, designed, investigation, analysis and interpretation of data, reviewed edited and approved the version to be submitted and any revised version.

Samah Alhazmi: Conceptualization, investigation, methodology, analysis and interpretation of data, reviewed edited and approved the version to be submitted and any revised version.

Marwah Almasri: Conceptualization, methodology, designed, reviewed edited and approved the version to be submitted and any revised version.

Maha Helal: Conceptualization, investigation, designed, reviewed edited and approved the version to be submitted and any revised version.

Ethics

This study is original and innovative and contains unpublished material. There are no ethical issues involved and none of the authors have any conflicts of interest to disclose.

References

Abdulkareem, K. H., Mohammed, M. A., Gunasekaran, S. S., Al-Mhiqani, M. N., Mutlag, A. A., Mostafa, S. A., ... & Ibrahim, D. A. (2019). A review of fog computing and machine learning: Concepts, applications, challenges and open issues. *IEEE Access*, *7*, 153123-153140. https://doi.org/10.1109/ACCESS.2019.2947542

Alrawais, A., Alhothaily, A., Hu, C., & Cheng, X. (2017). Fog computing for the internet of things: Security and privacy issues. *IEEE Internet Computing*, *21*(2), 34-42. https://doi.org/10.1109/MIC.2017.37

Baccarelli, E., Naranjo, P. G. V., Scarpiniti, M., Shojafar, M., & Abawajy, J. H. (2017). Fog of everything: Energy-efficient networked computing architectures, research challenges and a case study. *IEEE Access*, *5*, 9882-9910. https://doi.org/10.1109/ACCESS.2017.2702013

Boonsong, W., Inthasuth, T., & Zulkifli, C. Z. (2023). Proposed Precision Analysis of Water Quality Monitoring Embedded IoT Network. *Przeglad Elektrotechniczny*, *2023*(9). https://doi.org/10.15199/48.2023.09.33

De Donno, M., Tange, K., & Dragoni, N. (2019). Foundations and evolution of modern computing paradigms: Cloud, IOT, edge and fog. *IEEE Access*, *7*, 150936-150948. https://doi.org/10.1109/ACCESS.2019.2947652

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. https://doi.org/10.1109/CIT/IUCC/DASC/PICOM.2015.51

Hou, X., Ren, Z., Wang, J., Zheng, S., Cheng, W., & Zhang, H. (2020). Distributed fog computing for latency and reliability guaranteed swarm of drones. *IEEE Access*, *8*, 7117-7130. https://doi.org/10.1109/ACCESS.2020.2964073

Jonathan, A., Uluyol, M., Chandra, A., & Weissman, J. (2017). Ensuring Reliability in Geo-Distributed Edge Cloud. *Proceedings of the 2017 Resilience Week (RWS) Conference, Sept. 18-22, Chase Center on the Riverfront Wilmington, US*, pp: 127-132. https://doi.org/10.1109/RWEEK.2017.8088660

Kunal, S., Saha, A., & Amin, R. (2019). An overview of cloud-fog computing: Architectures, applications with security challenges. *Security and Privacy*, *2*(4), e72. https://doi.org/10.1002/spy2.72

Mouradian, C., Naboulsi, D., Yangui, S., Glitho, R. H., Morrow, M. J., & Polakos, P. A. (2017). A comprehensive survey on fog computing: State-of-the-art and research challenges. *IEEE Communications Surveys and Tutorials*, *20*(1), 416-464. https://doi.org/10.1109/COMST.2017.2771153

Mukherjee, M., Matam, R., Shu, L., Maglaras, L., Ferrag, M. A., Choudhury, N., & Kumar, V. (2017). Security and privacy in fog computing: Challenges. *IEEE Access*, *5*, 19293-19304. https://doi.org/10.1109/ACCESS.2017.2749422

Ni, J., Zhang, K., Lin, X., & Shen, X. (2017). Securing fog computing for internet of things applications: Challenges and solutions. *IEEE Communications Surveys & Tutorials*, *20*(1), 601-628. https://doi.org/10.1109/COMST.2017.2762345

Osanaiye, O., Chen, S., Yan, Z., Lu, R., Choo, K. K. R., & Dlodlo, M. (2017). From cloud to fog computing: A review and a conceptual live VM migration framework. *IEEE Access*, *5*, 8284-8300. https://doi.org/10.1109/ACCESS.2017.2692960

Pereira, J., Ricardo, L., Luís, M., Senna, C., & Sargento, S. (2019). Assessing the reliability of fog computing for smart mobility applications in VANETs. *Future Generation Computer Systems*, *94*, 317-332. https://doi.org/10.1016/j.future.2018.11.043

Wang, Y., Uehara, T., & Sasaki, R. (2015, July). Fog computing: Issues and challenges in security and forensics. *In 2015 IEEE 39th Annual Computer Software and Applications Conference* (Vol. *3*, pp. 53-59). IEEE. https://doi.org/10.1109/COMPSAC.2015.173

Yi, S., Hao, Z., Qin, Z., & Li, Q. (2015a November). Fog computing: Platform and applications. *In 2015 3rd IEEE Workshop on Hot Topics in Web Systems and Technologies (HotWeb)*, (pp. 73-78). IEEE. https://doi.org/10.1109/hotweb.2015.22

Yi, S., Li, C., & Li, Q. (2015b June). A survey of fog computing: Concepts, applications and issues. *In Proceedings of the 2015 Workshop on Mobile Big Data*, (pp. 37-42). https://doi.org/10.1145/2757384.2757397

Zhang, P., Zhou, M., & Fortino, G. (2018). Security and trust issues in fog computing: A survey. *Future Generation Computer Systems*, *88*, 16-27. https://doi.org/10.1016/j.future.2018.05.008

Zhanikeev, M. (2015). A cloud visitation platform to facilitate cloud federation and fog computing. *Computer*, *48*(05), 80-83. https://doi.org/10.1109/MC.2015.122