Dynamic allocation of microservices for virtual reality content delivery to provide quality of experience support in a fog computing architecture

Derian Alencar [1], Helder Oliveira (Co-Advisor)[2], Denis Rosário (Advisor) [1]

[1]Federal University of Pará (UFPA) - Belém, PA, Brazil

[2]Federal University of ABC (UFABC) – Santo André, SP, Brazil

derian.alencar@itec.ufpa.br, helder.oliveira@ufabc.edu.br,

denis@ufpa.br

Abstract. *Virtual Reality (VR) content is gaining popularity and allowing users to immerse themselves in a new world over the Internet. However, the high-demand for resources and the low latency requirements of VR services require changes in the current 5G networks to deliver VR with quality assurance. Microservices present a suitable model for deploying services at different levels of a 5G fog computing architecture for managing traffic and providing Quality of Experience (QoE) guarantees to VR clients. However, finding the most suitable fog node to allocate microservices for VR clients in QoE-aware 5G scenarios is a difficult task. This article proposes a QoE VR-based mechanism for allocating microservice dynamically in 5G architectures, called Fog4VR. Fog4VR determines the optimal fog node to allocate the VR microservice based on delay, migration time, and resource utilization rate. This article also presents the INFORMER, an integer linear programming model aiming to find the optimal global solution for microservice allocation. Results obtained with INFORMER serve as a baseline to evaluate Fog4VR in different scenarios using a simulation environment. Results demonstrate the capabilities of Fog4VR compared to existing mechanisms in QoE, migration time, fairness index, and terms of cost.*

1. Introduction

The usage of Virtual Reality (VR) technologies is gaining popularity. Youtube, for example, has begun supporting for 360° content for VR playback, which uses Video-on-Demand (VoD) to partition a VR video into spatially related tiled videos. However, providing VR services with Quality of Experience (QoE) support is challenging due to its panoramic nature and high video resolution, which necessitates latency below *20 ms* and a bitrate greater than that of traditional videos. As a result, delivering VR video streams with QoE support over the current communication infrastructure is a difficult task [Li et al. 2018].

In this particular situation, fog computing allows for the provision of VR services in closer proximity to mobile users, effectively meeting the requirements for delay and bandwidth in content distribution [Rosário et al. 2018]. The fog infrastructure has the ability to dynamically expand based on the demand for VR video streams, making it suitable for deploying the necessary components for VR video distribution using a microservices architecture. Specifically, the implementation of a microservice architecture can yield advantages in content delivery by reducing the expenses associated with creating and relocating cache instances in response to changes in user demand [Tian et al. 2018]. As a result, the content distribution system can promptly adapt to fluctuations in user demand and decrease the wasteful utilization of resources on unnecessary service instances.

To ensure Quality of Experience (QoE) support for VR video streams is necessary to allocate the microservices in a fog infrastructure. This involves two steps: firstly, deciding where to allocate the microservice within a heterogeneous fog computing architecture, and secondly, migrating content. The decision step is crucial as the allocation of microservices directly impacts VR service performance. Therefore, the decision-making process must consider various metrics to assess fog nodes, networks, and users, to make efficient decisions. Content migration involves transferring content and forwarding user requests to the allocated fog node. An efficient design for these steps is essential for optimal performance.

This paper presents the contributions in the master thesis [Alencar 2022], which tackles the challenge of allocation of microservices for VR content delivery in a fog computing architecture. We also take into account the user side using QoE metrics as a way to evaluate our model and corroborate our proposal. The research conducted and presented in this thesis advances the state-of-the-art in the following ways: i) First, we design a controller to allocate and migrate microservices in a heterogeneous fog computing architecture called Fog4MS. ii) We provide an optimization model for microservices positioning called INFORMER, which considers transmission delay, content migration time, and resource utilization rate of a fog node to determine the optimal position for allocation. Next, based on the insights from INFORMER, we introduces Fog4VR to distribute VR content with QoE support using the concepts of microservices and heterogeneous fog architectures. It uses the same parameters as INFORMER to identify suitable fog nodes to allocate microservices, improving the QoE of VR contents. iii) Simulated experiments shows the proximity of Fog4VR to the optimal results obtained with INFORMER. For instance, Fog4VR reduces cost in 7% and migration time in 12%, while delivering VR video stream with QoE 50% better than compared to existing mechanisms.

2. Related Works

In this section, we provide a concise overview of the current advancements in the dynamic allocation of microservices for VR video streaming. Numerous studies have attempted to address the challenges within this domain; however, many of these works fail to consider all the characteristics inherent in the problem. For instance, certain works solely focus on the QoS aspects, disregarding the overall experience perceived by end-users. Alternatively, some studies only partially account for QoE by considering a single metric and overlooking the intricacies associated with the video streaming characteristics specific to VR applications.

Table 1 provides a summary of the key attributes found in the reviewed studies that focus on content allocation concerning QoE and QoS awareness, Video on Demand (VoD) capability, VR aspects, and the allocation mechanism technique. These characteristics aim to prevent users from abandoning the service due to interruptions (stalls), duration of interruptions, and the time it takes for playback to start. A dynamic content allocation mechanism must effectively handle each request by identifying the fog node capable of meeting resource requirements based on the QoS and QoE characteristics of VR streaming services, ultimately maximizing QoE. However, most studies tend to consider only a single aspect. The absence of VoD support can lead to issues related to the unique nature of this application, particularly concerning the perception of video playback by users (QoE). Ultimately, it is preferable to employ a heuristic technique due to its lower computation time and complexity. As far as we know, only Fog4VR takes into account all the critical attributes for microservice allocation in a fog computing environment for VR distribution with QoE support. Further details about the related works can be found in the Thesis [Alencar 2022].

3. Allocation Architecture and Mechanism

This section describes the fog computing architecture and mechanism for the dynamic allocation of VR microservices.

Table 1. Summary of Characteristics of Related Works

Works	Characteristics				
	QoE	QoS	VoD	VR	Approach
[Rigazzi et al. 2019]			✓	✓	Mathematical Modeling
[Mehrabi et al. 2021]		✓		✓	Heuristic Model
[Apostolopoulos et al. 2020]		✓			Game Theory (PNE)
[Ni et al. 2017]		✓	✓		Petri Network
[Mishra et al. 2020]		✓			AHP-EV
[Yousefpour et al. 2019]		✓			ILP+Greedy
[Mahmud et al. 2019]	✓				Fuzzy
[Lai et al. 2020]	✓	✓			ILP+Heuristic Model
Fog4VR (Proposal)	✓	✓	✓	✓	AHP

3.1. Architecture

We consider fog nodes deployed anywhere in the network, such as microdata centers in mobile network base stations, Base Band Unit (BBU), Internet Service Provider (ISP), etc. In this sense, we consider fog nodes deployed anywhere in a network organized into tiers between mobile devices (at the bottom) and cloud (at the top) [Rosário et al. 2018]. The cloud keeps the original copy of all VR content, and also distributes VR content for each user request and maintains an overview of each service and node status. It is important to mention that we consider the VR service running on a microservice architecture to guarantee availability and scalability. Moreover, the cloud runs an allocation mechanism, such as Fog4VR, to select the fog node, allocate the microservice, and distribute VR content adaptively and proactively. Hence, the Cloud layer maintains all control components and available VR content.

A heterogeneous organization of fog nodes consists of a computational infrastructure with various characteristics to allocate content as closely as possible to the user. Each fog node is represented by $f_i \in F$, which has a unique identity $i \in [1, n]$. For instance, a microservice for VR streaming could be deployed in a fog node f_i to speed up content distribution while improving the QoE. As a result, a fog node f_i could have one or more instances of microservices, which deliver the requested content to the user. Finally, the Client application requests and displays VR content to users.

It is important to mention that each module runs on a microservice architecture to guarantee availability and scalability. In this scenario, the user (*i.e.*, mobile devices, desktop computers, etc.) using any communication network (*i.e.*, 5G, LTE, and WiFi) requests the VR content to the service controller deployed the cloud computing. Hence, the content distribution service has access to these infrastructures to allocate a portion of their resources to instantiate a microservice for VR content delivery. As a result, each fog node f_i has one or more instances of allocated microservice, which delivers the requested VR content to users. The microservice instances are controlled on each fog computing infrastructure by Infrastructure Controller, which the fog computing provider manages. Finally, the Clients layer encompasses all users of the distribution service who make requests for available content.

3.2. Mechanism Operation

The Fog4VR mechanism is found in the *Service Controller* module, which manages the decision steps (*i.e.*, positioning of microservices in computing at a given fog node f_i) and content migration (*i.e.*, transferring content and directing requests to that node). Fog4VR receives the microservice request $m \in M$, which is a 3-tuple with content id id, content size s, and location l of the microservice requisition. Based on such information, the mechanism verifies which fog f_i is suitable to allocate the microservice m. To this end, it computes the resource utilization rate u_i based on microservice size m_s, allocated memory Am_i, and total storage available Ts_i in a given fog f_i.

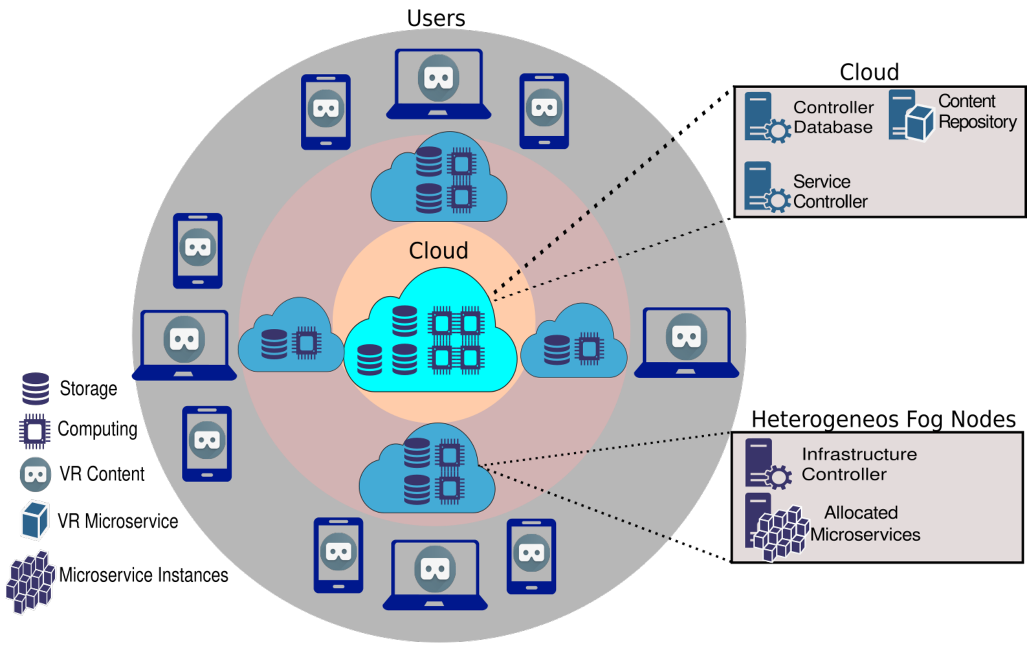


Figure 1. Architecture to deploy virtual reality VoD with microservices

If the fog node f_i has enough storage resources to allocate the microservice m, then Fog4VR must compute the migration time Tm_i to transfer the microservice from the cloud to a given fog node f_i, where W is the TCP window size, and $d_{i,r}$ is the packet transmission time. Lastly, it updates the vector L_i with the values of delay d_{f_i,m_l}, migration time Tm_i, and resource utilization rate u_i.

Fog4VR mechanism computes the cost C_i for allocating the microservice m in given fog node f_i based on Eq. 1. The cost C_i takes into account different metrics (*i.e.*, delay d_{f_i,m_l}, content migration time Tm_i, and resource utilization rate u_i), which have a varying degree of importance.

$$C_i = w_1 \cdot d_{f_i,m_l} + w_2 \cdot Tm_i + w_3 \cdots u_i \tag{1}$$

Fog4VR considers a multi-criteria decision-making method to balance inputs with different degrees of importance, where we argue AHP to compute the best response according to the significance of each parameter to another. Specifically, AHP decomposes a complex problem into a hierarchy of simpler sub-problems, combining qualitative and quantitative factors for analysis. Fog4VR mechanism builds a comparison matrix $V_{j,k}$ for each fog node f_i to compare all pairs of criteria based on Eq. 2.

$$V_{j,k} = \begin{array}{c} \\ d_{f_i,m_l} \\ Tm_i \\ u_i \end{array} \begin{array}{c} \begin{array}{ccc} d_{f_i,m_l} & Tm_i & u_i \end{array} \\ \begin{pmatrix} 1 & 4 & 8 \\ 1/4 & 1 & 2 \\ 1/8 & 1/2 & 1 \end{pmatrix} \end{array} \rightarrow [0.72\ 0.18\ 0.10] \tag{2}$$

As a result, we obtain the eigenvector $P = [0.72{,}0.18{,}0.10]$, indicating the weights of metrics, such as 0.72 for delay (d_{f_i,m_l}), 0.18 for migration time (Tm_i), and 0.10 for resource utilization rate (u_i). These weights are used to compute the cost Eq.1 for allocating a cache microservice m in a given fog node f_i. At the end of the process, the fog node with the highest score is chosen, and the microservice is allocated to the node.

3.3. Fog4VR Computational Complexity

We analyzed the complexity of the Fog4VR algorithm, which depends on three sets of operations: (i) deriving the eigenvalue, (ii) solving equations to find weights, and (iii) computing values for composite value. These operations rely on two parameters: p (number of fogs) and n (number of criteria), in our case. Thus, the complexity is $O(min[pn^2,p^2n])$ [Mamat and Daniel 2007].

In our case, (n) is fixed, but (p) can vary. We tested Fog4VR with varying numbers of fog nodes and found that its complexity is linear with a fixed n value of 3. Futhermore, we shown in Figure 2, the complexity of Fog4VR is linear, with the fixed number of criteria (n=3) in our testing. This means that Fog4VR can run in real-time with a low impact and a high response time, making it ideal for microservice requests.

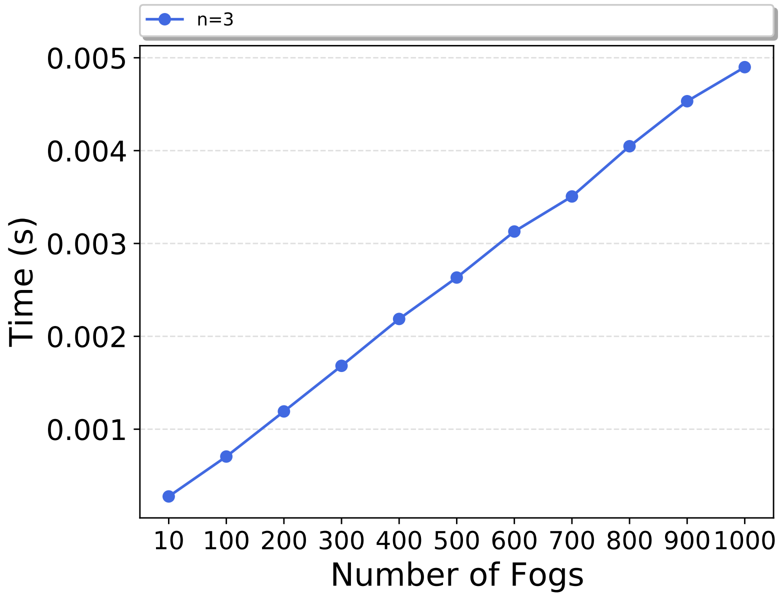


Figure 2. Representation of Fog4VR time complexity

4. INFORMER optimization model

Our proposed optimization model, INFORMER, is designed to facilitate the dynamic allocation of VR microservices within heterogeneous fog computing environments. The primary objective of INFORMER is to minimize latency, thereby maximizing the quality of experience (QoE). The results obtained through INFORMER can serve as a benchmark for comparing the performance of other heuristics, as it represents the optimal solution within the same scenario.

The primary objective of INFORMER is to enhance the quality of experience (QoE) by maximizing it according to Equation 3. This equation is specifically designed to minimize both delay and migration time, as these factors significantly impact QoE in VR streaming. Lower delay results in fewer and shorter stall events, ultimately improving the overall QoE of VR services.

To achieve this, the INFORMER model selects the fog node with the lowest delay and minimum migration time for microservice deployment. Furthermore, INFORMER adheres to the constraint outlined in Eq. 4, ensuring that each microservice M is allocated to a fog node accordingly. Additionally, the maximum storage capacity Ts_i, as specified in Eq. 5, must not be exceeded, and the bandwidth limit B_i of each fog node f_i must not be surpassed, as demonstrated in Eq. 6. Consequently, INFORMER returns a binary variable $\vartheta_{m,f}$ that indicates the fog index to which each microservice should be allocated.

$$Min \quad D = \sum_{m=0 \,\in\, M} \sum_{f_i=0 \,\in\, F} (d_{f_i,M_{l_m}} + d_{z,M_{l_m}}) \times \vartheta_{m,f_i} \tag{3}$$

Subject to:

$$\sum_{f_i=0 \,\in\, F} \vartheta_{m,f_i} = 1 \; \forall m \in M \tag{4}$$

$$\sum_{m=0 \,\in\, M} \vartheta_{m,f_i} \leq Ts_{f_i} \; \forall f_i \in F \tag{5}$$

$$\sum_{m=0 \,\in\, M} \vartheta_{m,f_i} \leq B_{f_i} \; \forall f_i \in F \tag{6}$$

5. Experimental Results

The experimental results were conducted using NS3, and the source code can be accessed on GitHub[1]. Each simulation was executed 33 times with different random seed values, and the obtained results are presented with a 95% confidence interval. The scenario encompasses varying numbers of microservice requests (*i.e.*, 20, 40, 60, 80, and 100), modeled according to a Poisson distribution. The microservices represent video streaming for VR services based on an MPEG-DASH application. Users have the option to choose from a catalog of 100 different VR content, with the selection being determined by a Zipf distribution with α = 0.7 to ensure a more evenly distributed content preference. The videos have a fixed duration of 30 minutes and are encoded at 25Mbps for a 4K VR stream. The scenario incorporates the virtual topology of the FIBRE project network to assign delays. For more details on simulation parameters and evaluation metrics, please refer to the Master thesis [Alencar 2022].

Our proposal is evaluated through two use cases: (i) *Fog4MS* focuses on allocating VoD microservices within a fog computing architecture using the AHP decision-making method described in Section 3.2. It aims to achieve a load balance across the network while minimizing migration time. In summary, Fog4MS prioritizes service providers' perspectives rather than prioritizing elements that yield better QoE for users. It does not consider QoE in its preference. (ii) *Fog4VR*, on the other hand, aims to allocate VR microservices within a fog computing architecture with QoE support. Fog4VR takes into account all the steps introduced in Section 3.2.

5.1. Fog4MS Results

Figures 3(a) and 3(b) display the mean duration of content migration and video buffering, respectively, during the microservice allocation process in the fog. The *buffering* time directly influences the quality of experience (QoE) perceived by Video on Demand (VoD) users, as a shorter buffering time leads to higher user QoE and a reduced rate of video abandonment. Therefore, it is crucial to consider metrics such as service migration time and buffering time when analyzing QoE.

In Figure 3(a), it is evident that the cloud exhibits no migration time since the content is already stored, requiring only service instantiation. However, Figure 3(b) reveals that the initial buffering time for videos increases as the number of microservices

[1]https://github.com/D3F3R4L/Fog4VR

grows. This phenomenon arises because the cloud receives all requests, leading to an increased workload and degraded performance. On the other hand, the greedy mechanism determines allocations based on latency between the client and fog node, resulting in consistently low buffering time. However, this mechanism performs poorly in terms of content migration time as it often assigns videos to remote network points. The random mechanism exhibits highly variable and consistently unsatisfactory performance. In contrast, Fog4MS employs intelligent decision-making that takes into account both latency and migration time, achieving the lowest content migration time even with a slight increase in buffering. Therefore, the cost/benefit ratio of the Fog4MS mechanism is entirely justifiable. Notably, it is crucial to emphasize that Fog4MS is the only method capable of reducing content migration time as the number of microservices increases. It is essential to highlight that Fog4MS is the only method to reduce content migration time with the increase in the number of microservices.

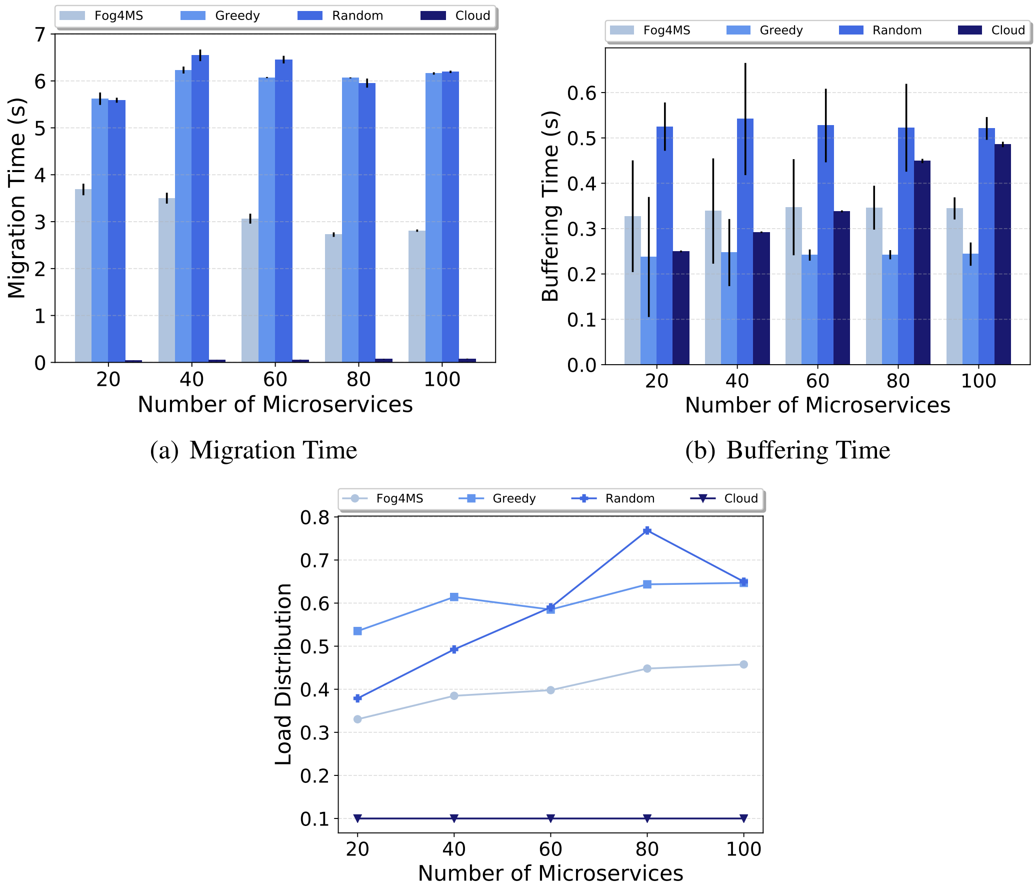
In Figure 3(c), the fairness index for the utilization of fog resources in the scenario simulation is depicted. This index represents the extent to which the workload is evenly distributed across the network. Allocating microservices in a single location, i.e., using only the cloud, represents the worst-case scenario in terms of fairness. Both the greedy and random mechanisms exhibit similar distribution patterns. In the random mechanism, the probability of allocation to any fog node is the same across all possible fog nodes. The fairness index of the Fog4MS mechanism is 33% and 30% lower than the indexes of the greedy and random mechanisms, respectively. Additionally, Fog4MS demonstrates the ability to distribute the workload across the network when necessary, as evidenced by the increase in the number of microservices. Therefore, Fog4MS offers enhanced efficiency in content distribution, as its decision-making prioritizes efficiency while maintaining network balance. The results of Fog4MS is better described on [de Alencar et al. 2020].

5.2. Fog4VR Results

Given its focus on QoE support, Fog4VR performance in terms of stall durations, buffering time, and percentage of unserved users for different numbers of microservice requests is presented in Figure 4. Figure 4(a) shows that Fog4VR yields the lowest number of stalls with the shortest durations, which is crucial as high values can lead users to abandon the VR service altogether. This superior performance can be attributed to Fog4VR's prioritization of VR microservice allocation on fog nodes with lower delay, considering migration time and resource utilization rate. This enables Fog4VR to select optimal locations for VR microservices, thereby minimizing the occurrence and duration of stalls.

The Fog4VR heuristic achieves results that closely resemble the best available solution, namely the INFORMER optimization model. In the worst case, Fog4VR increases the number of stalls by 50% and the stall duration by 5% to 40% compared to INFORMER. Furthermore, Fog4VR significantly reduces the number of stall events by 45% to 75% and the duration of stall events by 54% to 74% compared to the AHP-EV mechanism, depending on the number of microservice requests. When compared to QoS-*Greedy*, Fog4VR lowers the number of stall events by 33% to 45% and the stall duration by 9% to 45%. This improvement is attributed to QoS-*Greedy*'s focus on allocating microservices to fog nodes with shorter delays while overlooking other pertinent metrics necessary for optimal decision-making. Consequently, QoS-*Greedy* tends to make suboptimal decisions and overload fog nodes. This deficiency becomes more pronounced in demanding scenarios where additional metrics such as utilization rate play a vital role in achieving a high QoE.

Figure 4(b) shows that INFORMER and Fog4VR have almost the same buffering time (*i.e.*, initial buffering time). This behavior is because Fog4VR can efficiently determine the fog node to allocate microservice based on multi-criteria metrics combined with different degrees of importance for each metric, leading them to better allocation decisions



(a) Migration Time

(b) Buffering Time

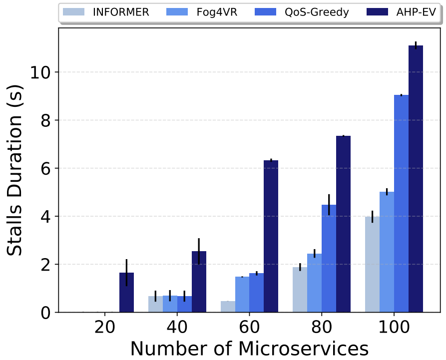
(c) Load Distribution of Microservices

Figure 3. Simulation results for Fog4MS

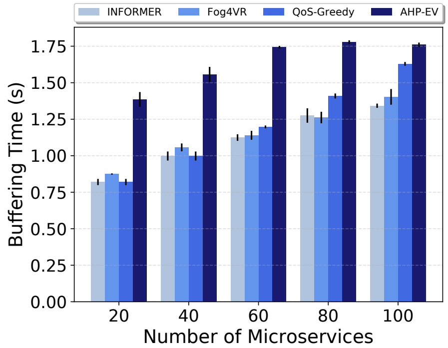
for users with low delay without overloading the fog nodes. Also, Fog4VR provides similar performance in terms of buffering compared to the INFORMER optimization model. QoS-*Greedy* have similar performance compared to INFORMER for scenarios with up to 60 microservice requests, *i.e.*, low demand. Simultaneously, the buffering time is 20% worst than INFORMER for a scenario with more than 80 microservice requests. The AHP-EV mechanism tends to assign users to more distant locations like the cloud, which gives more delay to users and, consequently, a higher buffering time.

Figure 4(c) shows the ratio of users who will probably be unsatisfied with the VR experience obtained. As discussed before, this value is tied to stalls and the duration of stalls that users experienced in the simulation. Results indicate that Fog4VR have similar performance compared to INFORMER since Fog4VR provides a lower number of stalls with a short duration, leading to a lower number of unsatisfied users. In turn, AHP-EV presented the worse ratio in all scenario cases. QoS-*Greedy* have a similar ratio of unsatisfied users compared to INFORMER and Fog4VR in low demand scenarios. Fog4VR have 11% to 22% less unsatisfied users than QoS-*Greedy*, 27% to 56% less unserved users than AHP-EV, and only 20% to 28% more unsatisfied users than INFORMER.

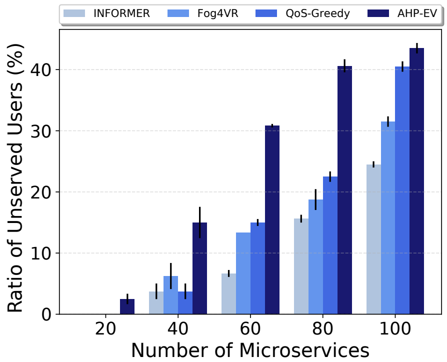
Through our performance evaluation analysis, we have determined that Fog4VR outperforms other allocation mechanisms in terms of both QoE and service provider perspectives. Fog4VR incorporates a multi-criteria approach that takes into account factors such as delay, migration time, and resource utilization, making it highly desirable for microservice allocation in VR applications. As a result, Fog4VR achieves higher efficiency



(a) Stalls Duration



(b) Buffering time



(c) Percentage of Unserved User

Figure 4. Simulation results for Fog4VR

in content distribution by prioritizing network balancing and resource optimization while delivering an enhanced QoE to users. For a more detailed understanding of the results obtained by Fog4VR.

6. Conclusion and Thesis Impact

The challenge of ensuring a high-quality user experience in VoD services, particularly with VR applications, has prompted both researchers and service providers to seek effective solutions. In response to this, we present the Fog4VR mechanism, which considers factors such as delay, content migration time, and fog utilization rate for allocating VR microservices within the fog computing infrastructure.

The results obtained from our evaluation demonstrate that the Fog4VR mechanism significantly reduces both the number and duration of stalls, surpassing the AHP-EV and Greedy mechanisms, while closely approaching the performance of the INFORMER optimal solution. For future research, we envision expanding the capabilities of Fog4VR to encompass the management of service allocation in mobile edge computing, potentially incorporating unmanned aerial vehicles (UAVs) to enhance the QoS of applications in challenging scenarios.

7. Publications

Table 2 sumarizes the papers published as result of this Master Theses.

References

Alencar, D. (2022). Dynamic allocation of microservices for virtual reality video delivery to provide quality of experience support in a fog computing architecture.

Table 2. Summary of Results Published

Works	Qualis	Local	Google Scholar Citations
[Alencar et al. 2022]	A4	SBRC	-
[de Alencar et al. 2020]	A2	TNSM	11
[Santos et al. 2020]	A1	Computer Networks	7

Alencar, D., Both, C., Antunes, R., Oliveira, H., Cerqueira, E., and Rosário, D. (2022). Dynamic microservice allocation for virtual reality distribution with qoe support. *IEEE Transactions on Network and Service Management*, 19(1):729–740.

Apostolopoulos, P. A., Tsiropoulou, E. E., and Papavassiliou, S. (2020). Risk-aware data offloading in multi-server multi-access edge computing environment. *IEEE/ACM Transactions on Networking*, 28(3):1405–1418.

de Alencar, D., Rosário, D., Cerqueira, E., Both, C., and Antunes, R. (2020). Distribuição de conteúdo sob demanda através da alocação de microserviços dinâmicos na borda e núcleo da rede. In *Anais do XXXVIII SBRC*, pages 575–588, Porto Alegre, RS, Brasil.

Lai, P. et al. (2020). QoE-aware user allocation in edge computing systems with dynamic QoS. *Future Generation Computer Systems*, 112:684–694.

Li, J. et al. (2018). Modeling QoE of Virtual Reality Video Transmission over Wireless Networks. In *IEEE Global Communications Conference (GLOBECOM)*, pages 1–7.

Mahmud, R. et al. (2019). Quality of Experience (QoE)-aware placement of applications in Fog computing environments. *JPDC*, 132:190–203.

Mamat, N. J. Z. and Daniel, J. K. (2007). Statistical analyses on time complexity and rank consistency between singular value decomposition and the duality approach in AHP: A case study of faculty member selection. *Mathematical and Computer Modelling*, 46(7-8):1099–1106.

Mehrabi, A., Siekkinen, M., Kämäräinen, T., and yl Jski, A. (2021). Multi-tier cloudvr: Leveraging edge computing in remote rendered virtual reality. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 17(2):1–24.

Mishra, S. et al. (2020). Dynamic Resource Allocation in Fog-Cloud Hybrid Systems using Multi-criteria AHP Techniques. *IEEE Internet of Things Journal*.

Ni, L. et al. (2017). Resource allocation strategy in fog computing based on priced timed petrinets. *IEEE Internet of Things Journal*, 4(5):1216–1228.

Rigazzi, G. et al. (2019). An Edge and Fog Computing Platform for Effective Deployment of 360 Video Applications. In *2019 IEEE WCNCW*, pages 1–6. IEEE.

Rosário, D. et al. (2018). Service Migration from Cloud to Multi-tier Fog Nodes for Multimedia Dissemination with QoE Support. *Sensors*, 18(2).

Santos, H., Alencar, D., Meneguette, R., Rosário, D., Nobre, J., Both, C., Cerqueira, E., and Braun, T. (2020). A multi-tier fog content orchestrator mechanism with quality of experience support. *Computer Networks*, 177:107288.

Tian, Y. et al. (2018). A new live video streaming approach based on Amazon S3 pricing model. In *IEEE 8th CCWC*, pages 321–328.

Yousefpour, A. et al. (2019). FogPlan: a lightweight QoS-aware dynamic fog service provisioning framework. *IEEE Internet of Things Journal*, 6(3):5080–5096.