Hardware Testbed based Analytical Performance Modelling for Mobile Task Offloading in UAV Edge Cloudlets

Gaurang Bansal[1], Abhishek Tyagi[2], Vishnu Narayanan[2],
[1]Department of Electrical and Computer Engineering, National University of Singapore, Singapore
[2]Department of Electrical and Electronics Engineering, BITS Pilani, Pilani Campus, India

***Abstract*—In recent times, there is a paradigm shift to cloud services that offer on-demand computer system resources, especially data storage and computing power. The main reason for the shift is that it removes the user's active participation to perform computationally intensive tasks. However, current cloud-based services incur high user latency as being deployed very far from the user. One alternative solution to the traditional cloud-based paradigm is drone-based edge computing. In drone edge computing, drones are located near the user and deployed to provide data offload services. There have been many works that have addressed the issue of efficient task assignment in edge devices. This paper presents a concrete analytical performance model for drone cloudlet networks and factors that influence the service response time to the user. The results can be helpful for network administrators to make the current edge computing paradigm faster, more robust and, cost-effective.**

***Index Terms*—UAVs, Security, cloudlets.**

I. INTRODUCTION

With the surge of innovative technology solutions, mobile devices have no longer remained just communication devices. Instead, they have evolved to become much more innovative, intelligent, dynamic, and functional. Mobile devices are being viewed as a viable replacement for computers [1]. Whatever task could be possible on the computer is being done by smartphones. Smartphones have provided ease of usage, compactness, effectiveness to end-users. It is expected that they will accomplish any task which a computer can do, whether computation, imaging, music player, health monitoring, and energy analytics [2–4].

Each coin has a flip side. Although smartphones are much easier to handle, their functionality is constrained by the computational resources available. Mobile phones generally have small battery backup, small computational processors. So performing a high-end task on the phone can cause increased power consumption, thus reducing the battery backup capacity [5]. With internet technology merging rapidly, many service providers are looking to leverage software to use cloud services. Cloud computing has changed the world tremendously. Since its inception in 2005, cloud computing has had a significant impact on the way we live, work, and learn [6]. Cloud services are a connection of hardware resource centers that have high computational devices connected. Mobile devices can offload their task to cloud servers. Cloud servers compute the tasks and return the answer to the end-user [7].

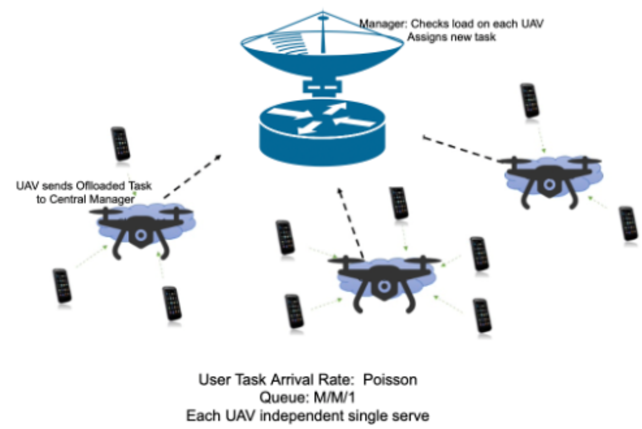

Fig. 1: System Model

However, these cloud servers face an issue of significant user latency. Cloud stations are located very far from the end-user. Since most mobile applications have a hard timing deadline, higher communication latency with the cloud makes the result useless. Many researchers have been working on edge computing solutions [8–10]. The edge computing solutions have small computation devices located near to the user. These edge devices are interconnected to each other to distribute offloading tasks from users adequately. Edge devices provide dual benefits, as they are closer to users with minimal propagation delay than cloud-based services. Edge-based services are much cheaper and do not require extra infrastructure, thus providing win-win solutions to stakeholders and users.

With the coming of unmanned aerial vehicles (UAVs) or drones, the edge-based computing paradigm has further evolved, where the edge devices are located near the user and mobile. This allows the stakeholder to distribute its resources to cater to the need of end-users effectively. In the coming days, unmanned aerial vehicles and drone-based solutions can also eliminate the need for the internet. All the devices are gradually becoming smarter, whether smartwatches [11], smart washing machines [12], or smart TV. All these have one thing in common, which is computation and

Notation	Meaning
R	Response Time
S_c	Service Rate of UAV in zone c
S_k	Service Rate of UAV in zone k
λ_c	Arrival rate of requests in zone c
λ_k	Arrival rate of requests in zone k
Q_c	Queue length with class c requests
Q_k	Queue length with class k requests
ρ_c	Load of task in zone c
ρ_k	Load of task in zone k
ρ	Sum of all ρ_c
S^{max}	Maximum Service rate of UAV
dist(x)	Distance of mobile device from UAV
$\gamma(x)$	File size of mobile device at distance x
α,β	Constants

TABLE I: Notation Table

communication. Since the feasibility of the internet is not possible all the time. Researchers are visioning UAV-based edge servers as a solution to this issue. UAV-based edge servers can be located near IoT devices, where devices can offload the task to edge devices using near communication technologies such as Bluetooth and wifi.

In this paper, we perform a performance analysis of task offloading among UAV edge servers. We design a testbed using mobile devices and edge servers and evaluate the user latency experience by varying factors such as offloading file size, the distance of the user from the server, current number of requests, etc. To the best of our knowledge, this work has not been previously analyzed. Understanding these parameters will have network administrators improve the edge computing services' performance and provide users with a better service experience.

The overview of the paper is as follows. Section II is the system description. Section III discusses the proposed load balancing and task assignment in the UAV edge network. In section IV, we discuss our hardware testbed model. Observations and performance analysis are presented in Section V. Section VI summarises the cumulative factors that affect the user experience. Finally, the Conclusion is presented in Section VII.

II. System Description

Fig. 1 depicts the system model that is being considered. In a region, a cluster of UAVs is deployed to serve the end mobile users. All the UAVs are connected via a wireless network. There exists a central UAV manager in the network. Different mobile devices connect to different UAV edge servers which are closest to it. The connected UAV sends the task to the UAV manager. UAV manager maintains the status of all the UAVs in terms of the number of resources available, number of requests pending on UAV, network congestion, request file size, and user distance. Based on all the parameters, the UAV manager allocates the task to the UAV, resulting in minimum response time to the UAV. All UAVs are considered to have the same computational resources. The notations used in this paper are presented in Table 1.

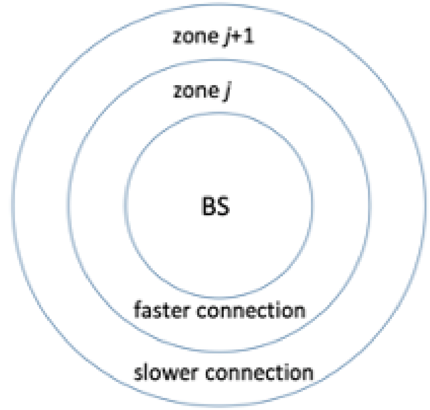

Fig. 2: Zone formation based on service rate variation

III. Proposed Load Balancing

This section discusses the strategy used by the UAV manager to offload the task to the UAV, which results in minimum response time. For formulating the response time, we consider a scenario where task requests arrive for UAV following Poisson distribution. Poisson distribution is a very close approximation to the distribution seen in a real-life scenario. UAV servers follow an M/M/1-PS(processor sharing) queue, model. For each UAV, we consider it as the base server (BS). Each of the UAVs can offer up to S^{max} service rate, which is determined by its hardware setup. It is to be noted that the effective service rate reduces as the distance between the UAV and the mobile device increases. The service rate perceived by the user at a distance x from the UAV server is given as:

$$s(x) = \frac{S^{max}}{1+\beta(dis(x))^{\alpha}} \tag{1}$$

The distance between the mobile device at location x and the UAV is $dis(x)$. α and β parameters enable the service rate to be adjusted to meet a wide range of network scenarios. So, for a single UAV, we can divide the service rate experience by user as a function of distance by dividing the region into zones, as shown in Fig. 2. The service rate in a zone remains almost constant, which is proved part of observation 1 in our hardware testbed. We consider each zone a separate class of requests, as users' same requests in different zones are treated differently. Using queuing theory [13], we can write response time for task in k^{th} zone as:

$$R_k = \sum_c A_c S_c + S_k$$

Using PASTA property of Poisson Arrival [14]

$$R_k = \sum_c Q_c S_c + S_k$$

Using Little's Law [15]

$$R_k = \sum_c \lambda_c R_c S_c + S_k$$
$$\lambda_k R_k = \lambda_k \sum_c \lambda_c R_c S_c + \lambda_k S_k$$

$$\begin{aligned} Q_k &= \lambda_k \sum_c R_c \rho_c + \rho_k \\ &= \lambda_k R_k \sum_c \frac{R_c}{R_k} \rho_c + \rho_k \\ &= Q_k \sum_c \frac{R_c}{R_k} \rho_c + \rho_k \end{aligned}$$

We use the fact that response time is almost similar in all the zones as proved in [16].

$$\frac{R_c}{R_k} \approx 1 \tag{2}$$

$$\begin{aligned} Q_k &\approx Q_k \sum_c \rho_c + \rho_k. \\ &= Q_k \rho + \rho_k \\ Q_k(1-\rho) &\approx \rho_k \\ Q_k &\approx \frac{\rho_k}{1-\rho} \end{aligned}$$

So the latency experienced by UAV is sum of all the Queues of different classes. We term it as latency indicator of the UAV, $\mathscr{L}$, which is given by:

$$\begin{aligned} \mathscr{L} &= \sum_k Q_k \\ &= \frac{\rho}{1-\rho}. \end{aligned}$$

Here, the load at the UAV serving its request (which represents the queuing delay experienced by the mobile device request) is given by ρ, which is UAV server utilization. It depends on file size, arrival rate, service rate, etc.

$$\rho = \int_{\mathscr{R}} \frac{\gamma(x)}{S_x} dx, \ \ 0 \le \rho \le 1 \tag{3}$$

In here, $\gamma(x)$ is file size by mobile device located at location x and S_x is service rate at location x. The UAV manager assigns the task to the UAV with least $\mathscr{L}$. Whenever a request comes, it evaluates $\mathscr{L}$ for each UAV server and allocates the task where $\mathscr{L}$ is minimum. This takes into account file size, the distance of user, current load on the server.

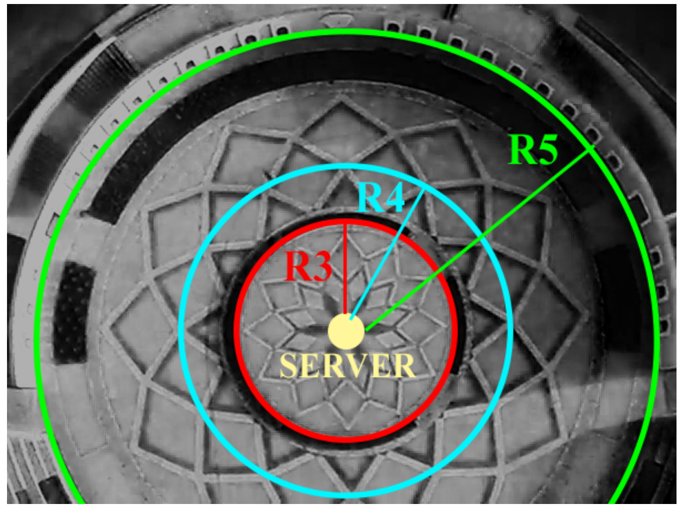

Fig. 3: Concentric zones of service formed during testbed. Each zone has almost same service rate

IV. Hardware Testbed

In this section, we describe our testbed model designed to analyze the performance metric of the parameters. Raspberry pi3 was used as onboard computers in UAVs. Mobile devices could connect to UAV via a wireless connection. Apple Macbook Air was used as the central UAV manager that is connected to the UAV network. All the requests for task offloading arrive from the UAV to the central manager, who then decides where to offload the task. The system's parameters were measured by sending files from a server to a client and back. The server was placed in the middle of a circular park while the client's location was varied in circular peripheries of varying radii.

We considered two different UAV topologies with 10 and 16 UAVs, respectively, to evaluate the proposed scheme's performance. The primary aim of having two different topologies is to test the performance of similar loads on networks of different sizes. Two types of UAVs are taken into consideration, one with a maximum service rate of 6MBps and the other with a maximum service rate of 12MBps. The latency indicator will thus be a qualitative measure, expressing a relative value of the latency. Latency indicator hides the complexity of inference which is experienced in a real-life scenario. However, at the same time gives a good idea of how the actual latency function looks like.

V. Observations and Discussions

In this section, we discuss vital observations from the execution of the hardware testbed.

OBSERVATION 1: The assumption that service rate variation can be modeled as different rings is validated herein.

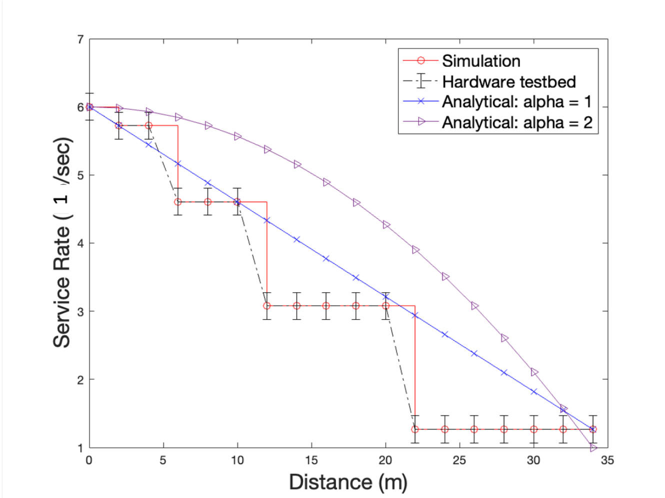

Fig. 4: Variation of Service rate vs Distance of User

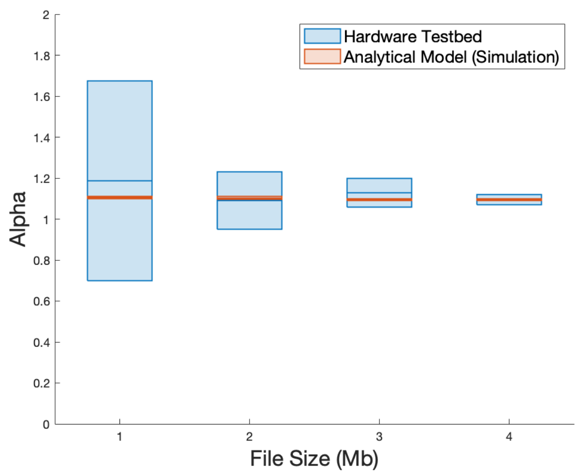

Fig. 5: Variation of Alpha for different file sizes

The service rate rather than linear or quadratic function follows a staircase pattern. Hence the service rate, rather than changing with distance, can be seen as multiple rings where the service rate remains almost the same while the user moves inside the ring. As depicted in Fig. 3, the time for completing the interaction between the user and the client was noted. We generated five rings. The closest distance between the server and the client was 2 meters, while the farthest distance was 34 meters.

OBSERVATION 2: The difference of radius of rings is not constant, as is the assumption used in most research works. Instead, the radius difference increases as we move away from the server, as shown in Fig. 3 and Fig. 4. In Fig. 4, for hardware values, we found the values matched simulation values within the error bound of 0.2.

OBSERVATION 3: Figure 3 shows the difference in service rate generated by simulation by varying alpha values and

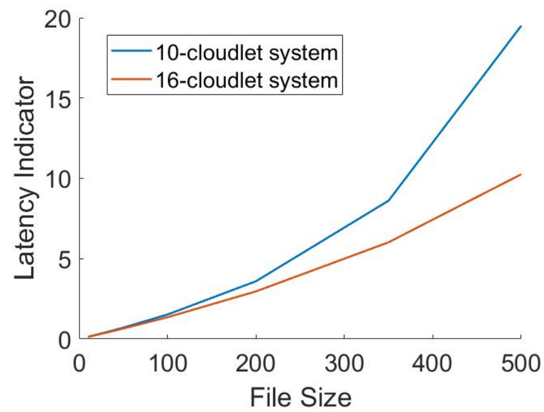

Fig. 6: Variation of Latency Indicator vs file size

our hardware testbed benchmark. For the hardware testbed, we created a 95% confidence interval of variation of service rate with distance. The red line shows the actual output, whereas the black line shows possible variation in service rate due to changes in external factors. We can observe from Fig. 4 that hardware testbed/hardware simulation is closer to the analytical model with $\alpha = 1$. So it is expected that the model to have service rate variation linear rather than quadratic.

OBSERVATION 4: Alpha and beta decide how is the impact of distance on the service rate received by the user. In fig. 5, we see that it is challenging to estimate alpha using experimentation for small file sizes. Alpha values oscillate as the distance is varied. But gradually, when we increase the file size, alpha values converge. A small size file is quite quickly downloaded, and the impact of distance cannot be correctly observed. As we increase the file size, it increases the total response time for the user.

OBSERVATION 5: The network's quality in terms of the latency associated with the distance of the requests from the UAV is amassed within the parameter α. A lower effective service rate at the request's location provided by the UAV would signify that the value of α is high. With experimentation conducted in an open environment, we found the alpha values to converge closer to 1.1 (as shown in fig. 5). This matches our observation in Fig. 4, where we expected alpha to be closer to 1 rather than 2.

OBSERVATION 6: The latency indicator sees a quadratic increase with an increase in the file size (in fig. 6). This is due to a higher load on the UAVs. The difference between the 10 and 16 UAV networks is a difference of transformation. However, in both cases, curvature remains the same. This sudden increment in latency indicator is due to the crowding of requests at each UAV which are not being catered because of the low service capability.

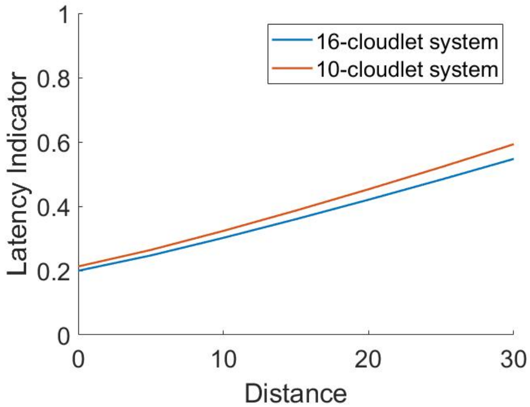

Fig. 7: Variation of Latency Indicator vs Distance

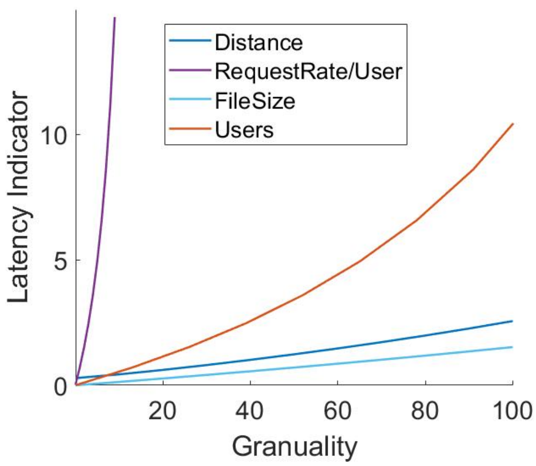

Fig. 8: Observing the effect of all parameters simultaneously.

OBSERVATION 7: Contrary to expectation, the latency indicator sees a linear and slow increase with the distance (fig. 8). This is due to users being uniformly distributed over an area. Changing the location of a single mobile device by a distance does not drastically affect UAV's latency indicator.

VI. Optimising parameters

As observed from Fig. 8, the Latency indicator increases at the maximum rate, increasing the request rate per user. An increase in the number of users on the server also increases the latency sharply. However, file sizes per request and distance between the cloudlet and the user have a more negligible impact on the latency. While distance has a higher latency value in the early part of the plot, it is quickly overtaken by other parameters. Thus, it is advised to keep the request rate to a low value for a network designer, which larger size of request packets can compensate. This will keep the overall latency generally low.

VII. Conclusion

This paper presents an analytical performance model for UAV edge cloudlet networks. Edge devices are located near to the user and are deployed to provide data offload services to users. The paper presents a way through which the latency is calculated using experimental results and theoretical formulae. Simulations were performed, where the distance, file size, number of cloudlets, number of users, and request per user was varied, and its effects on the latency were observed. The results can be helpful for network administrators to make the current edge computing paradigm faster, more robust and, cost-effective.

References

[1] V. Chamola, C.-K. Tham, and G. S. Chalapathi, "Latency aware mobile task assignment and load balancing for edge cloudlets," in *2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*. IEEE, 2017, pp. 587–592.

[2] M. T. R. Khan, Y. Z. Jembre, S. H. Ahmed, J. Seo, and D. Kim, "Data freshness based auv path planning for uwsn in the internet of underwater things," in *2019 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2019, pp. 1–6.

[3] M. T. R. Khan, S. H. Ahmed, Y. Z. Jembre, and D. Kim, "An energy-efficient data collection protocol with auv path planning in the internet of underwater things," *Journal of Network and Computer Applications*, vol. 135, pp. 20–31, 2019.

[4] M. T. R. Khan, Y. Z. Jembre, S. H. Ahmed, and D. Kim, "Remedy: Receiver-initiated mac based on energy-efficient duty-cycling in the iout," *IEEE Access*, vol. 7, pp. 105 202–105 211, 2019.

[5] V. Chamola, C.-K. Tham, S. Gurunarayanan, N. Ansari *et al.*, "An optimal delay aware task assignment scheme for wireless sdn networked edge cloudlets," *Future Generation Computer Systems*, vol. 102, pp. 862–875, 2020.

[6] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica *et al.*, "A view of cloud computing," *Communications of the ACM*, vol. 53, no. 4, pp. 50–58, 2010.

[7] C. Gong, J. Liu, Q. Zhang, H. Chen, and Z. Gong, "The characteristics of cloud computing," in *2010 39th International Conference on Parallel Processing Workshops*. IEEE, 2010, pp. 275–279.

[8] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE internet of things journal*, vol. 3, no. 5, pp. 637–646, 2016.

[9] W. Shi and S. Dustdar, "The promise of edge computing," *Computer*, vol. 49, no. 5, pp. 78–81, 2016.

[10] M. Satyanarayanan, "The emergence of edge computing," *Computer*, vol. 50, no. 1, pp. 30–39, 2017.

[11] M. E. Cecchinato, A. L. Cox, and J. Bird, "Smartwatches: the good, the bad and the ugly?" in *Proceedings of the 33rd Annual ACM Conference extended abstracts on human factors in computing systems*, 2015, pp. 2133–2138.

[12] F. Mattern, "From smart devices to smart everyday objects," in *Proceedings of smart objects conference*, 2003, pp. 15–16.

[13] S. Fomundam and J. W. Herrmann, "A survey of queuing theory applications in healthcare," 2007.

[14] R. M. Loynes, "The stability of a queue with non-independent inter-arrival and service times," in *Mathematical Proceedings of the Cambridge Philosophical Society*, vol. 58, no. 3. Cambridge University Press, 1962, pp. 497–520.

[15] J. D. Little and S. C. Graves, "Little's law," in *Building intuition*. Springer, 2008, pp. 81–100.

[16] B. Baynat and N. Nya, "Performance model for 4g/5g networks taking into account intra-and inter-cell mobility of users," in *2016 IEEE 41st Conference on Local Computer Networks (LCN)*. IEEE, 2016, pp. 212–215.