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

Presenter: Gaurang Bansal

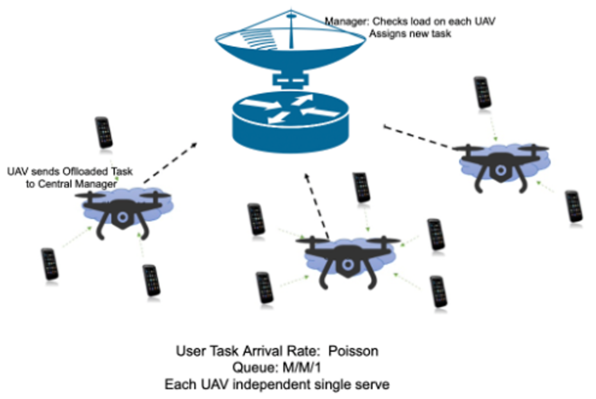
OBJECTIVE

- Cloud services offer on-demand computer system resources, especially data storage and computing power.
- Current cloud-based services incur high user latency as being deployed very far from the user.
- One alternative solution is drone-based edge computing. In drone edge computing, drones are located near the user and deployed to provide data offload services.
- This paper presents an analytical performance model for drone cloudlet networks and factors that influence the service response time to the user.
- Helpful to make the current edge computing paradigm faster, more robust and, cost-effective.

Contributions

- Performance analysis of task offloading among UAV edge servers.
- Testbed designed using mobile devices and edge servers
  Evaluated 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.
- Understanding these parameters will have network administrators improve the edge computing services’ performance and provide users with a better service experience.

System Description



- Cluster of UAVs serve the end mobile users (in a region).
- UAVs are connected via a wireless network.
- Mobile devices connect to UAV edge servers closest to it.
- Connected UAVs 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.
- UAV manager allocates the task to the UAV, resulting in minimum response time to the UAV.

- For formulating the response time, we consider a scenario where task requests arrive for UAV following Poisson distribution.

- UAV servers follow an M/M/1-PS(processor sharing) queue, model.

- UAVs can offer up to ***Smax*** service rate. Effective service rate is ***s(x)***.

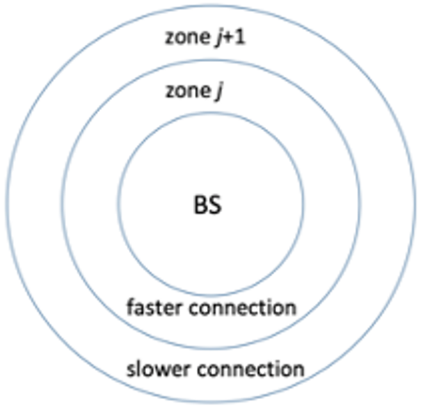
$$s(x) = \frac{S^{max}}{1+\beta(dis(x))^{\alpha}}$$

- Distance between the mobile device at location $\boldsymbol{x}$ and the UAV is ***dis(x)***. $\boldsymbol{\alpha}$ and **β** parameters enable the service rate to be adjusted to meet a wide range of network scenarios.

- Service rate experience by user as a function of distance by dividing the region into zones.

- The service rate in a zone remains almost constant.

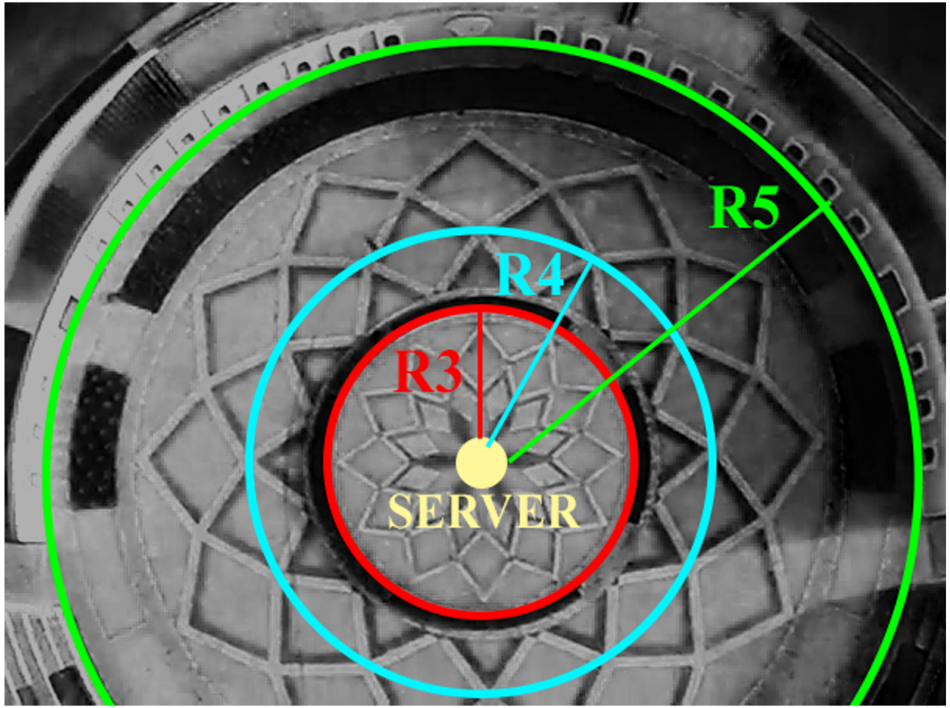
- We consider each zone a separate class of requests, as users' same requests in different zones are treated differently.



Latency experienced by UAV is sum of all the Queues of different classes.

Latency indicator of the UAV, “L”,

$$\mathscr{L} = \sum_k Q_k = \frac{\rho}{1-\rho}.$$



Concentric zones of service formed during testbed.

Hardware testbed

- 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
- The server was placed in the middle of a circular park while the client’s location was varied in circular peripheries of varying radii.

Observations

- The service rate rather than linear or quadratic function follows a staircase pattern
- 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
- The difference of radius of rings is not constant, as is the assumption used in most research works.
- Alpha and beta decide how is the impact of distance on the service rate received by the user. Alpha values oscillate as the distance is varied. But gradually, when we increase the file size, alpha values converge.

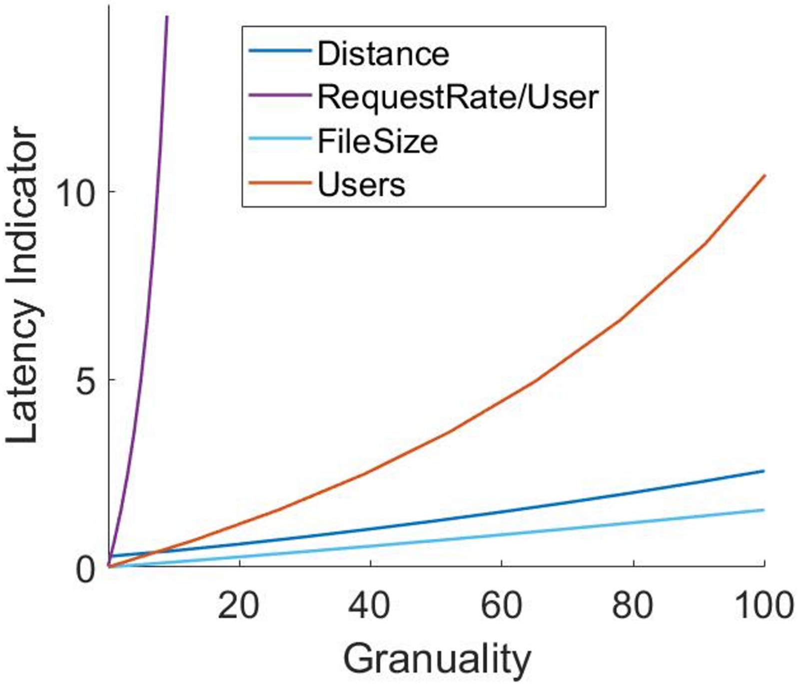
- The network’s quality in terms of the latency associated with the distance of the requests from the UAV is amassed within the parameter α

- 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

Summary

- Contrary to expectation, the latency indicator sees a linear and slow increase with the distance
- The Latency indicator increases at the maximum rate, when the request rate per user is increased.
- Increase in the number of users on the server also increases the latency sharply
- File sizes per request and distance between the cloudlet and the user have a negligible impact on the latency



Thank You.