PROJECT REPORT

Submitted by:

Tayyib Ul Hassan (369648)

Aamina Binte Khurram (393028)

Abstract

Building on the work on optimal BBU placement carried out in [1], we have implemented the heuristic presented in [1], albeit with some modifications in order to model online traffic. The cell sites generating requests, and the number of requests generated were made to change dynamically and the behaviour of the network was observed. Each cell site was assigned a functional split out of the four split options discussed in [2], and the corresponding split parameters for each cell site were kept in account when determining the behaviour of the rest of the network. The Python library, NetworkX, was used for carrying out the simulation of the basic network implemented.

Algorithm

The algorithm implemented is built upon the heuristic presented in [1]. The changes made involve the adaptation of the heuristic to cater to online traffic scenarios, which are basically changes in the creation and destruction of lightpaths. Lightpaths are created as virtual links, utilizing the physical link’s resources. After the creation of lightpaths, they are also teared down, i.e., the resources used are returned to the link. In this way, lightpaths are constantly built and destroyed so that the resources in the network are used efficiently and a maximum number of requests can be catered. The requests are generated in a random fashion from multiple cell sites, with every new burst having a random number of total requests.

Topology

The topology simulated consists of one fixed core central office, a random number of pre-assigned BBU pools, a random number of cell sites, and intermediate nodes. The weights for node types were set during node creation such that a much larger number of cell sites than BBU pools were created.

Online Traffic Generation

As mentioned above, this work explored an implementation of online traffic, i.e., the continuous generation of a random number of requests from a random number of cell sites. Thus, every time the program is run, requests are generated on the fly and multiple requests may end up using resources from the same physical link. In other words, multiple lightpaths may need to be created on the same physical link. In the case of offline traffic, this would lead to much quicker exhaustion of link resources, as resources are not returned to the network. Whereas in the case of online traffic, since resources are returned, complete resource exhaustion occurs much later.

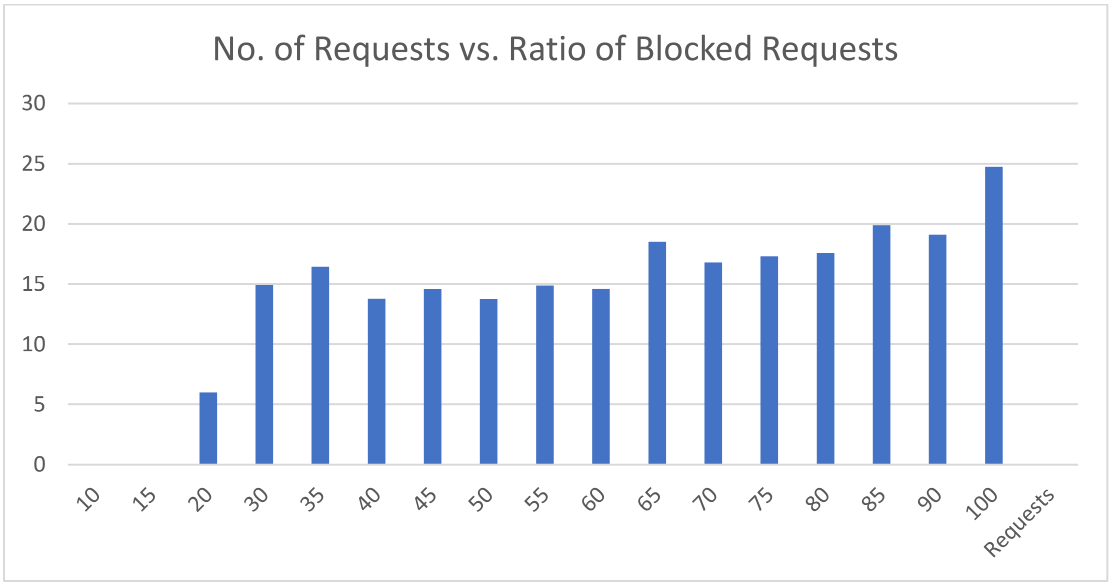
Implementation in code: Online traffic has been simulated via threading in Python. The generation of requests takes place in multiple threads which are executed together to create the effect of simultaneous request generation. The number of requests created in each thread differs randomly, to mimic the randomness in real-life online traffic. Global data(such as the resources allocated to each link) are defined on top of the code file, and each individual thread updates this data. As a result, we have multiple instances of the same code running concurrently, altering the same resources of topology.

Results

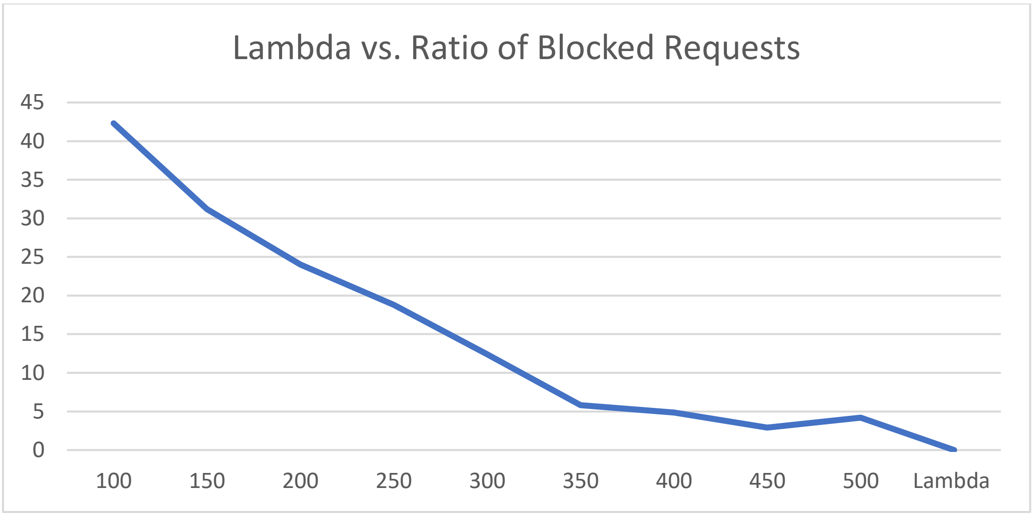
a) No. of requests vs. ratio of blocked requests
The total number of requests(distributed across all the cell sites) were plotted against the ratio of blocked requests to total requests. As was expected, an increase in the total number of requests sent corresponded to an increase in the number (and thus ratio) of requests blocked.

b) Wavelength per channel vs. ratio of blocked requests

The affect of using links of varying capacities on the total number of blocked requests was observed. Increasing the wavelength allocated to each channel in a wavelength-division-multiplexed link resulted in a decrease in the ratio of blocked requests to total requests.



Note: Capacity of lambda was fixed to 300 for each path while generating this graph.



Future Directions

There are multiple additions and enhancements possible in this work. Some of these include the following.

Using this initial implementation as a starting point, in the future, simulations can be carried out to determine the differences in performance of different classes of 5G traffic (namely eMBB, uRLLC, mMTC).

The creation of the same topology in different geographical locations can be modelled, and any deficiencies of one kind of topology vs another can be studied.

References

[1] M. Ahsan, A. Ahmed, A. Al-Dweik and A. Ahmad, "Functional Split-Aware Optimal BBU Placement for 5G Cloud-RAN Over WDM Access/Aggregation Network," in IEEE Systems Journal, doi: 10.1109/JSYST.2022.3150468.

[2] “Small cell virtualization: Functional splits and use cases,” Tech. Rep. Release 5.1, 2016. [Online]. Available: https://scf.io/en/documents/159_-_Small_cell_virtualization_functional_splits_and_use_cases.php