# Performance Modeling of Computer Systems and Networks

Prof. V. de Nitto Personè

AA 2018/2019

*Project*

Mobile devices are still limited in computing capabilities and battery lifetime. The problem of enhancing user experience may find a solution in cloud computing.
Consider a "two-layer" cloud system, consisting of an edge cloud server (cloudlet) and a remote cloud server, where the cloudlet is at "one-hop" distance from a set of mobile devices. Applications running on these mobile devices autonomously select some of their tasks for offloading to an external server (e.g., because of performance or energy saving reasons), and send an **offloading request** to a **controller** located on the cloudlet. Upon receiving a request, the controller takes a decision about whether the task should be sent to the cloudlet or the cloud, with the goal of minimizing the mean response time.
Typically, tasks hosted by the cloud server benefit of a higher execution rate, but suffer for greater network delay. We assume that the remote cloud server has virtually unlimited resources, hence it is always able to guarantee absence of interference among any number of tasks allocated to it. On the other hand, the cloudlet has limited resources, so that it is able to guarantee absence of interferences among tasks allocated to it as long as their number does not exceed a given threshold N.
Let us assume users belong to two classes and denote with $n_i$ the number of class i tasks in execution on the cloudlet. The controller can take the following simple decision upon task arrival events:

**Algorithm 1**

- arrival:

  **if** n1+ n2=N → send on the cloud
  **else** accept the task on the cloudlet

By considering that execution on the cloudlet of a class 1 task is more convenient than execution of a class 2 task, the controller can set a given threshold $S \leq N$ and can use the following access control algorithm:

**Algorithm 2**

- class 1arrival:

  **if** n1=N → send on the cloud
  **else if** n1+n2<S → accept
  **else if** n2 > 0 → accept the task on the cloudlet and send a class 2 task on the cloud
  **else** accept the task on the cloudlet

- class 2 arrival:

  **if** n1+n2≥S → send on the cloud
  **else** accept the task on the cloudlet

When a class 2 task is *interrupted* and sent on the cloud, a setup time $s_{setup}$ has to be considered to restart the task on the cloud.

Consider the following parameters:

- the mean arrival rates for the two classes are:
  $\lambda_1$= 6 task/s, $\lambda_2$= 6.25 task/s
- services both on cloudlet and on cloud can be assumed exponential, with the following mean rates:
  $\mu_{1clet}$= 0.45 task/s, $\mu_{1cloud}$= 0.25 task/s
  $\mu_{2clet}$= 0.27 task/s, $\mu_{2cloud}$= 0.22 task/s

Note that, as we stated above, the service time includes the transmission time. For this reason, we assume $\mu_{iclet} > \mu_{icloud}$ for both classes. Moreover, execution on the cloudlet of a class 1 task is more convenient than the execution of a class 2 task.
For Algorithm 2, we assume an exponential time with mean $E[s_{setup}]$= 0.8 s.

a.1. Design a next-event simulator for the system above with access control *Algorithm 1* and N=20.
a.2. Determine if the system is stationary or not and design the experiments accordingly.
a.3. Evaluate:
    1. the *system* response time and throughput both global and per-class;
    2. the per-class effective[1] cloudlet throughput;
    3. the per-class cloud throughput;
    4. the *class* response time and mean population, both for the cloudlet and for the cloud;
    5. Analyze the transient system statistics and in case of existence the steady-state statistics with the appropriate approach respectively.
b.1. Define a queueing model for the whole system.
b.2. Derive an analytical solution and validate the simulation results.

c.1. Implement *Algorithm 2* with threshold S=N.
c.2. Evaluate indices 1., 2., 3., 4., in a.3 and
    6. the percentage of class 2 *interrupted* tasks and their response time.
    Please explain how you manage with class 2 interrupted tasks, their service and response time.
c.3. Compare all results with those obtained with *Algorithm 1*.
c.4. Modify queueing model in step b.1. to include *Algorithm 2*.
c.5. Derive an analytical solution and validate the simulation results.

The steady state or transient statistics should be computed with a 95% confidence level. Please, consider as a guideline the paper "MANET Simulation Studies: The Incredibles", S. Kurkowski, T. Camp, M. Colagrosso, *Mobile Computing and Communications Review*, Volume 9, Number 4.

The project must be delivered at least one week before the oral test.
The students should prepare a presentation of the obtained results. The time should not exceed 10 minutes per student in a group (20 minutes for a student alone).

Documents to be delivered:
- The written report should include: conceptual and analytical models, the methodology used for estimating the statistics, the results, both graphical and numerical form, with the relative comments
- the listing code.

Evaluation grid
- clarity
- pertinence
- completeness.

[1] For "effective" throughput we mean the completed tasks rate.