
UNIVERSITÀ DI PISA

DEPARTMENT OF COMPUTER ENGINEERING

# Edge Computing
# Project Documentation

?? Dicembre 2024

Team:

**Cavedoni F.**
**Monaci M.**
**Pinna F.**

---

ACADEMIC YEAR 2024/2025

# Contents

# 1 Introduction

A cellular network is composed of M base stations, placed within a 2D floorplan of size L $\times$ H according to a regular grid. Each base station also has edge computing capabilities, i.e., it can receive computing tasks from cellular network's users and serve them at a rate equal to S instructions per seconds following a First Come First Served (FCFS) policy. Assume that all base stations are interconnected between each other via mesh topology.

## 1.1 Problem description

We consider N users placed at random locations `(x, y)` within the same 2D floorplan, where coordinates `x` and `y` are random variables to be defined later. Each user generates a new computing task request every T seconds, and each request consists of I instructions to be executed. T and I are exponentially distributed RVs. In particular, a user sends each new task request to its serving base station (i.e., the closest one), which in turn can follow one of methods below:

**a)** serve the request locally

**b)** forward the request to the less-loaded base station

## 1.2 Objectives

This project aims to fulfill the following:

- Evaluate the time required to complete a computing task for various values of N comparing method a) against method b)
- Evaluate the following scenarios:

  `x` and `y` are *uniform distributed* random variables in range

  $$x \in [0, width] \quad y \in [0, height]$$

  `x` and `y` are distributed as a *lognormal distribution* with parameters defined in subsection 2.2
- `TODO...`

## 1.3 Performance indexes

- Response time of the system (when a packet leaves the system)
- Response time of each queue
- Packet loss
- `TODO...`

# 2 Modeling

Our system is described by the following scheme

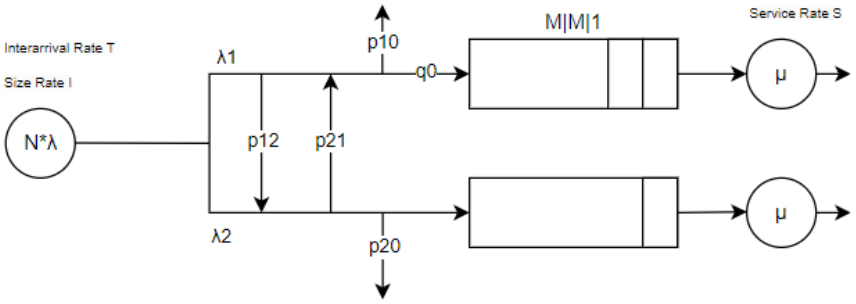


Figure 1: Scheme

As shown in Figure 1, each base station is modeled as an `M/M/1/k` system, as rates $\lambda_i$ and $\mu_i$ are exponentially distributed random variables $\forall\ i \in [1, M]$.

## 2.1 Assumptions

We make some assumption in order to simplify the implementation of the system.

1. Rerouting propagation delay is constant for each base station.
2. Every job entering a queue is destined to be served and cannot exit the queue in any other way.
3. Each queue length is *finite* with `k` slots.
4. *(work in progress)* Width and height of the grid are such that it results in a square area.

## 2.2 Factors description

- `N`: number of users
- `k`: length of each queue
- $\lambda_i$: Interarrival rate for base station $i$
- $\mu$: Service rate for each base station
- $\mu_{log}$: Average of lognormal distribution
- $\sigma_{log}$: Standard deviation of lognormal distribution

# 3 Implementation

## 3.1 Modules

The following modules have been defined:

- **EdgeNetwork**: compound module which represents the system and hosts the following simple modules
  - **BaseStation**: simple module which receives, precesses and (in case of scenario b)) forwards packets sent by users according to the specified distribution.
  - **User**: simple module which generates and sends packets (with length dependent by the specified distribution) to the nearest base station.

## 3.2 Modules' behaviour

### 3.2.1 EdgeComputingNetwork

- Hosts base stations and users physically and memorizes their parameters in order to retreive some of them during the simulation via parent pointers.

### 3.2.2 BaseStation

- Receives packets from users in form of `cMessage` objects and depending on the current scenario performs one of the following actions:

**Locally managed:** if the queue has enough free slots the base station enqueues the packet to be processed, ignoring all other base stations.

**Forwarding:** when a packet is received the base station searches for the base statio with the lowest load on their queue and forwards the packet to it. If and only if every other base station has a higher load, the packet is served locally.

- It also records statistics every time a packet is *dropped* or *forwarded* and about *queue length* and *response time*.

## 3.3 User

- Generates packets with length and rate which are exponentially distributed and sends them to the *nearest*[1] `BaseStation`.

[1] According to the euclidean distance

# 4 Verification