# EVENT-DRIVEN SIMULATION OF ADMISSION CONTROL TO MULTIPLE SERVERS

## APPLICATIONS OF RESEARCH AND INNOVATION – 2025/2026

INSTRUCTOR: CLEQUE–MARLAIN MBOULOU–MOUTOUBI
AND FRANCESCO DE PELLEGRINI

Please read first the General Instructions and be sure you address the work breakdown described there.

**System to be simulated.** The paper(s) you are concerned with are to be found in the corresponding Moodle directory. The main system description is the one presented in paper [1], basically Section II, i.e., System Model. You are *not* required to use the optimal admission control policy and to be aware of all the RL techniques to determine it. Instead, you can use a simple admission control policy with some heuristics we can discuss while you develop the simulator. However, the simulator should be able to plug-in any admission control policy.

*Note:* your implementation needs to be parametric: for instance, the number of areas $M$, the access capacity of a server or a the number of active flows allowed on a given server should be an input parameter. When you refer to variables, use same symbols you find in [1], this avoids any ambiguity.

**Logical Components.** Whatever the implementation you will perform, we expect to see the following components.

**Component 1: Area Traffic Generator.** This is the main engine of your event driven simulator. The events are the arrivals of a new flow to an area. The traffic generator generates the incoming flows generated by each area; the interarrival times per area are exponential random variables. The duration of the flow is an exponential random variable. Flows may have different size (bitrates). You should be able to use general distributions in your simulator.

**Component 2: Area Load Balancer.** Each area is ruled by a load balancer which sends each flow to a specific server. For your simulations, you can implement a randomized load balancer which sends each flow to target servers at random according to a prescribed probability distribution.

**Component 3: Server.** The server hosts the running applications. Each application consumes computing resources, so that a maximum number of active flows per server is possible. There is some quantity of access bandwidth per active flow that is consumed also. In your simulation you should be able to account for the said capacities.

**Component 4: Admission Control.** At each server, based on the charge on the access link, on the occupation of the server and on the utility of an application related to a given flow, the policy of admission decides if the flow routed to the target server will be admitted or rejected.

**Component 5: Application.** Each application is processing active flows, i.e, flows admitted on each server. You need to define a utility function for each application, depending on the source of each flow and the number of other flows from each area. You can propose utilities as you deem convenient and we are discussing them during the design of your simulator.

## References

[1] A. Fox, F. De Pellegrini, F. Faticanti, E. Altman, and F. Bronzino, "Optimal flow admission control in edge computing via safe reinforcement learning," in *Proc. of IEEE International Symposium on Modeling and Optimization in Mobile, Ad hoc, and Wireless Networks (WiOPT)*, Seul, South Korea, 21-24 October 2024, to appear.