# SIMULTATION PROJECT -2

# Simulation & Modeling of Token Bucket Traffic Shaper

**Communication Networks**
**June – October 2022**

***Objective – To get hands-on training and experience in developing and running simulation models in Omnet++ and able to visualize and analyze system performance. Bonus tasks have bonus points. Each task result output should be shown as a graph supported by a short analysis report of not more than 5–10 lines.***

**Task 2 (Submission Deadline: 30 September by 23:59 CET)**

Abstract the topology shown in Figure below. Station-A is sending packets towards Station-B via the Traffic Shaper. The traffic from Station-A is randomly generated following Poisson distribution (or any other random function). That is, the inter-arrival time between packets is Poisson distributed and hence is of variable packet rate. The traffic shaper will queue the incoming packets from A and then "schedule" them at a constant packet rate towards Station-B. Thus, the traffic between Station A and the Traffic Shaper is VBR, while between the Traffic Shaper and Station-B is CBR. The traffic shaper MUST be based on Token Bucket algorithm (see lecture slides)

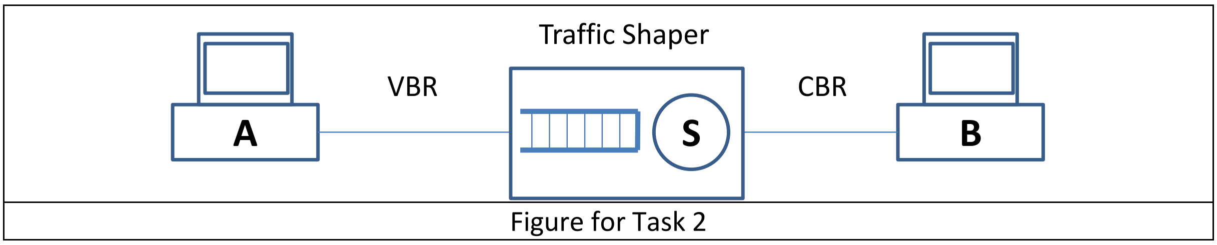

Figure for Task 2

The ingress queue inside the Traffic Shaper should be a FIFO queue with a fixed size (K) and constant service time. Set the output rate of the Traffic Shaper to some constant value (***ρ*** *pps* (packets per second)) while experiment with

1. Different mean inter-arrival times from Station-A towards the Traffic Shaper.
2. Different size of the token bucket (β) to control maximum burst size
3. Different token rates (*p*) to control the shape/data rate of the outgoing traffic
4. Etc.

Graphically show the input and output rates on a graph and also show the number of packets dropped and the burst size when the inter-arrival time exceeds the service rate of the queuing system. Make a short report as in Simulation Project-1.

**HINT-1**: Check the Aloha sample example to understand the implementation logic of generating packets with Poisson distributed inter-arrival time. That is, use the omnet function *exponential(λ)*, where (*λ*) is the mean inter-arrival time between packets.

**HINT-2**: For implementing the Traffic Shaper, refer to the *queuenet* example and run the oneFifo or *TandemFifos* configuration. Note that the implementation of the source, sink and the queue is under the *queuinglib* file folder.

**HINT-3:** Check out the Dyna sample example with one client, and see if the switch module can be tweaked/modified to implement the traffic shaper (TS) module.

A token bucket can be modeled in many ways. One way is to define a queue that will store tokens (e.g., cQueue tQueue (“tokenBucket”) ). Or implement a integer counter that keeps track of the tokens (i.e., int tokenBucketCounter). Each time a token arrives increment the token counter (i.e., tokenBucketCounter ++) and each time a packet leaves the queue decrement the token counter (i.e., tokenBucketCounter --). The token and its generation can itself be modeled using a self message that is triggered using the timer mechanism explained in the Tic Toc Tutorial.