EEE413 Data Communication and Communication Networks: **Token Bucket Filter Simulation**

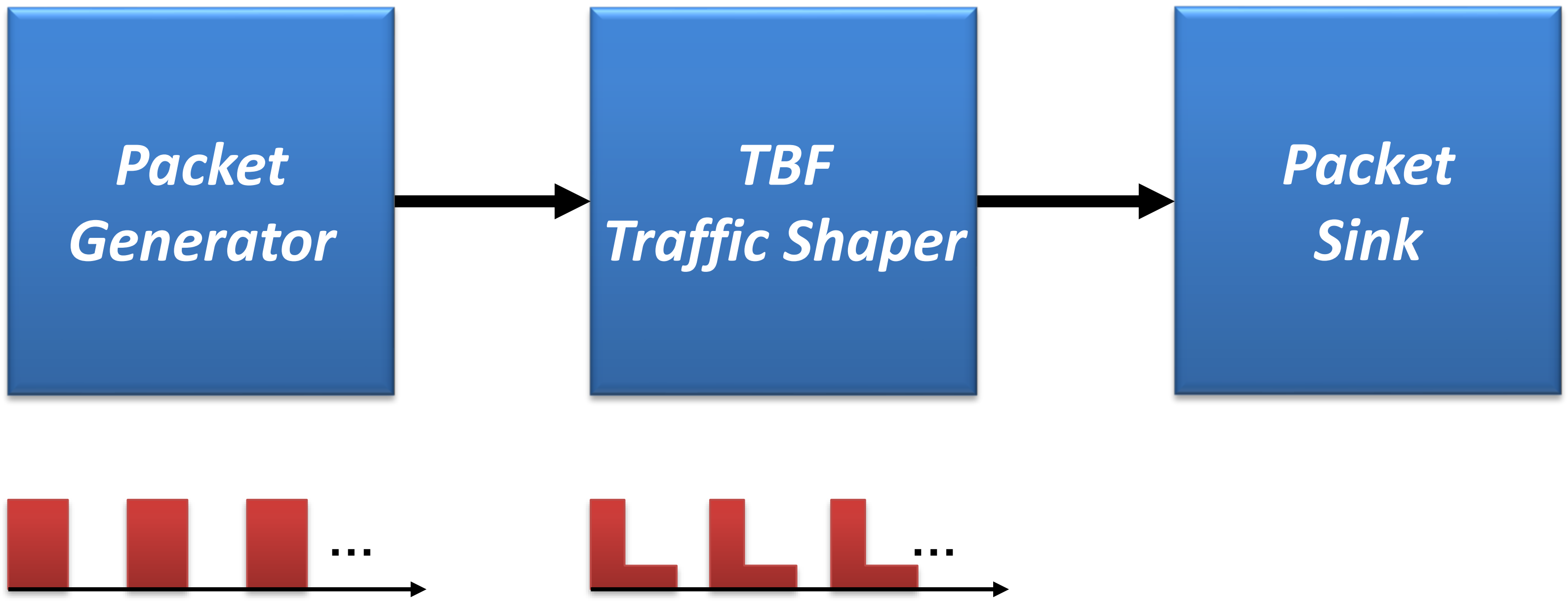
Dr Kyeong Soo (Joseph) Kim

Department of Electrical and Electronic Engineering

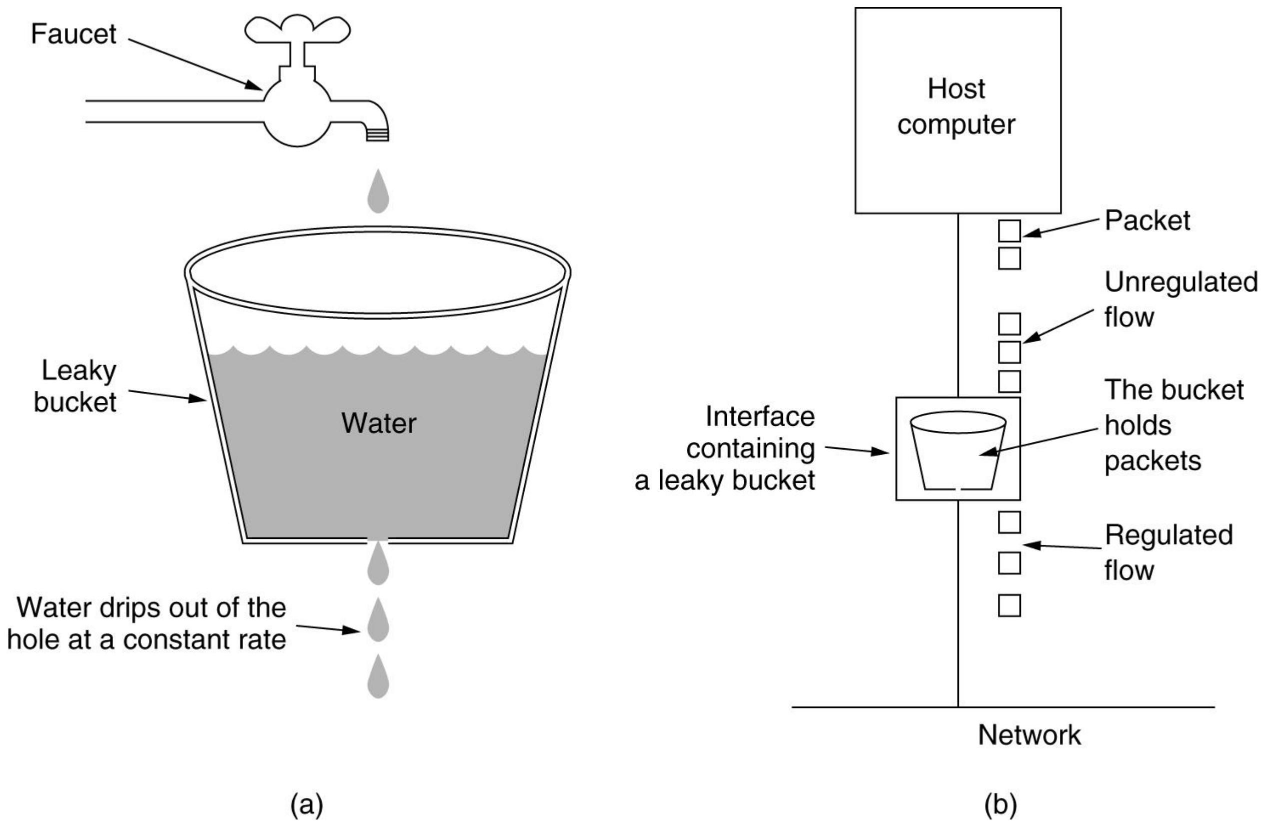
26 November 2019

Xi'an Jiaotong-Liverpool University
西交利物浦大学

Overview of TBF Simulation

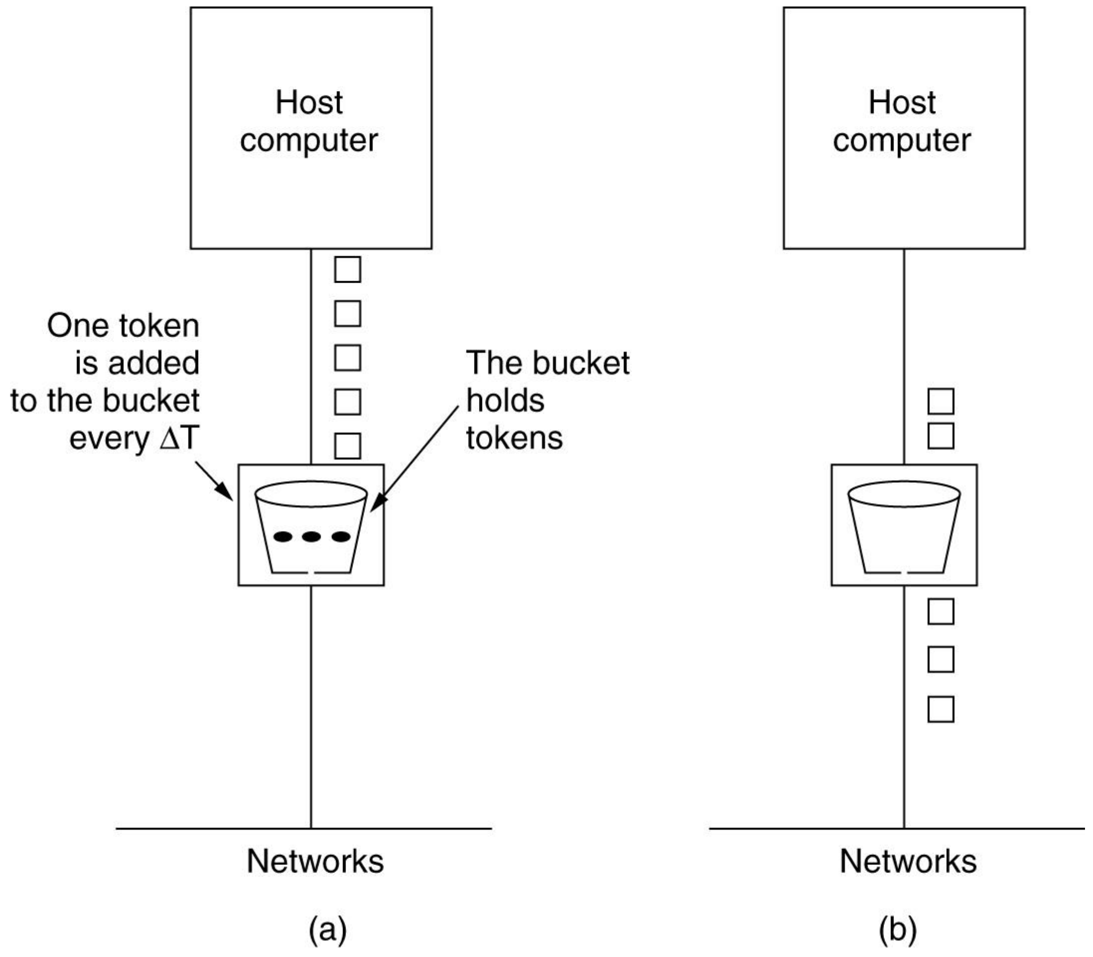


QoS Techniques: Leaky Bucket



(a) A leaky bucket with water. (b) A leaky bucket with packets.

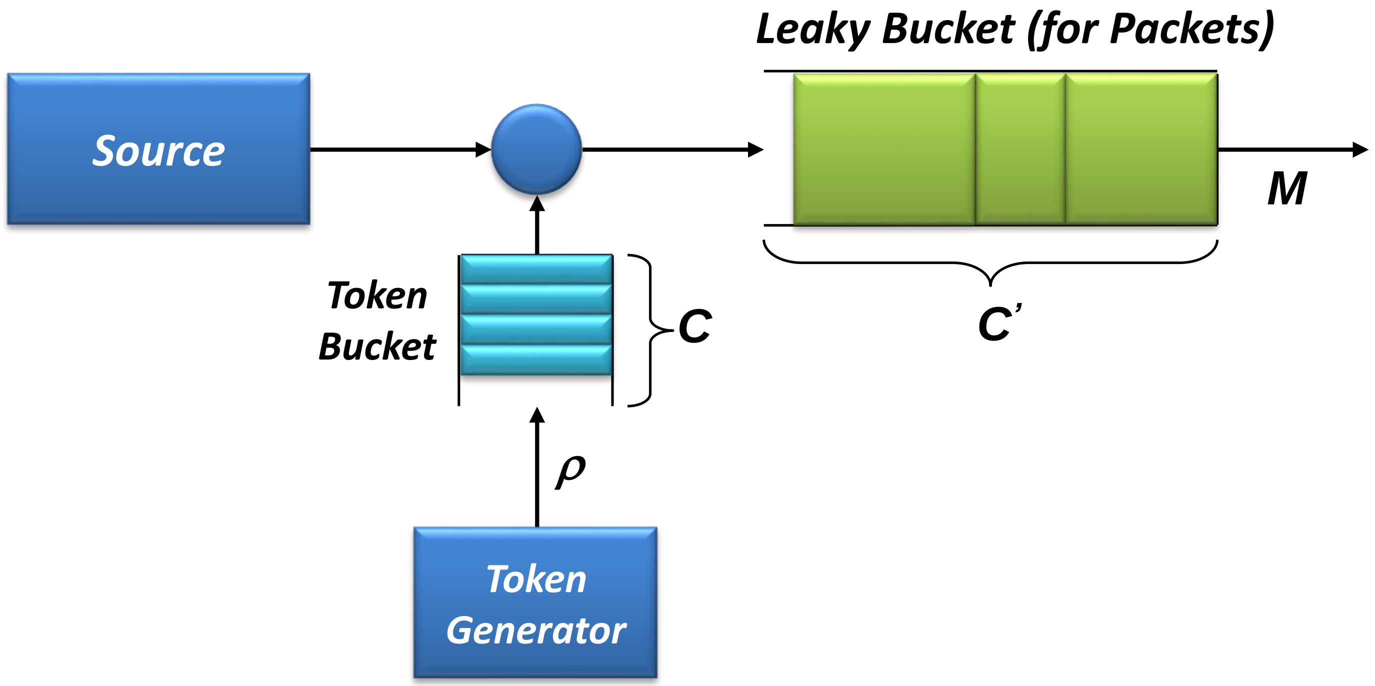
QoS Techniques: Token Bucket



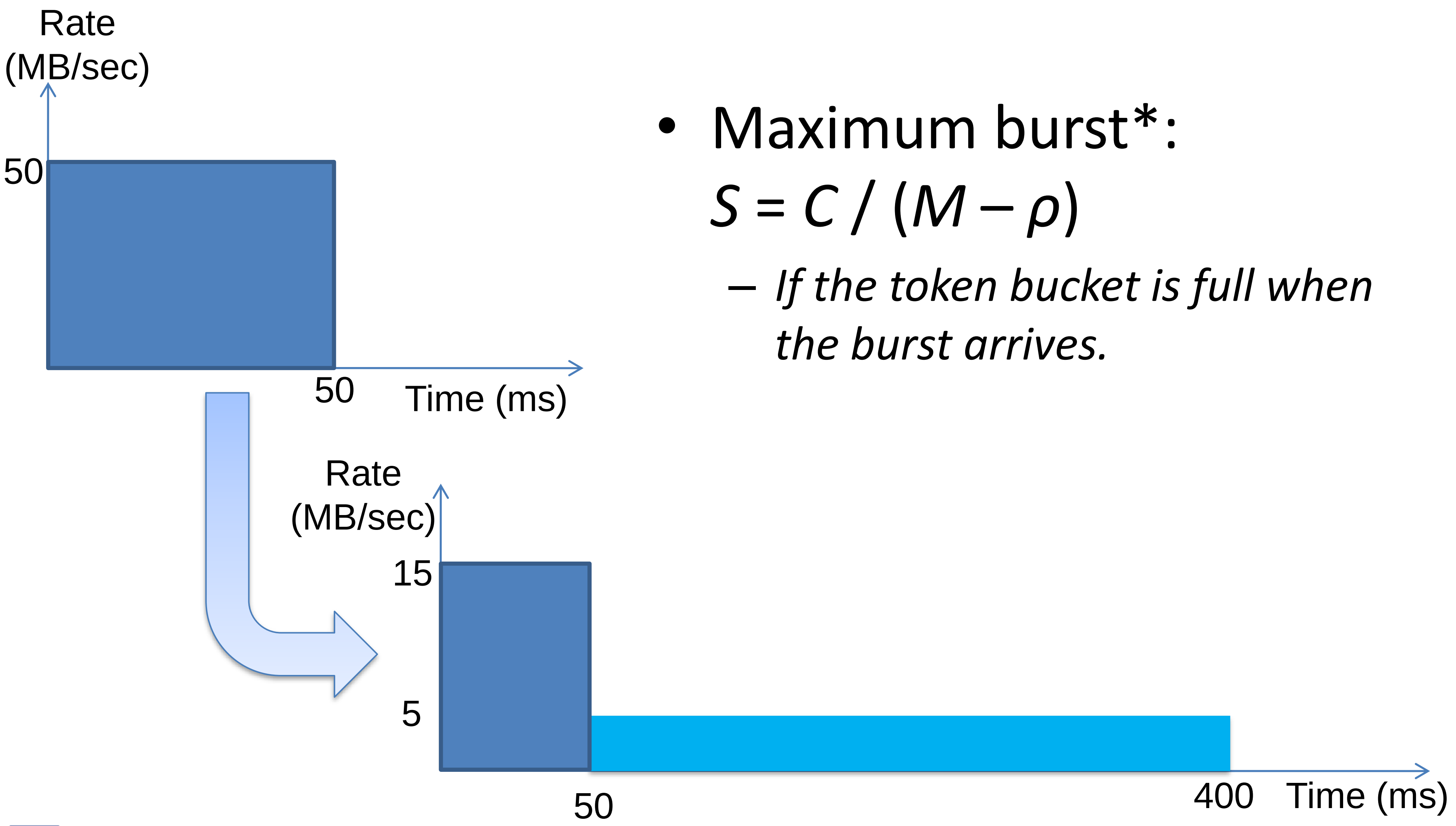
(a) Before. (b) After.

Combined Use of Leaky Bucket and Token Bucket

- ***Token bucket*** controls the ***burst size***.
- ***Leakey bucket*** controls the ***peak rate***.



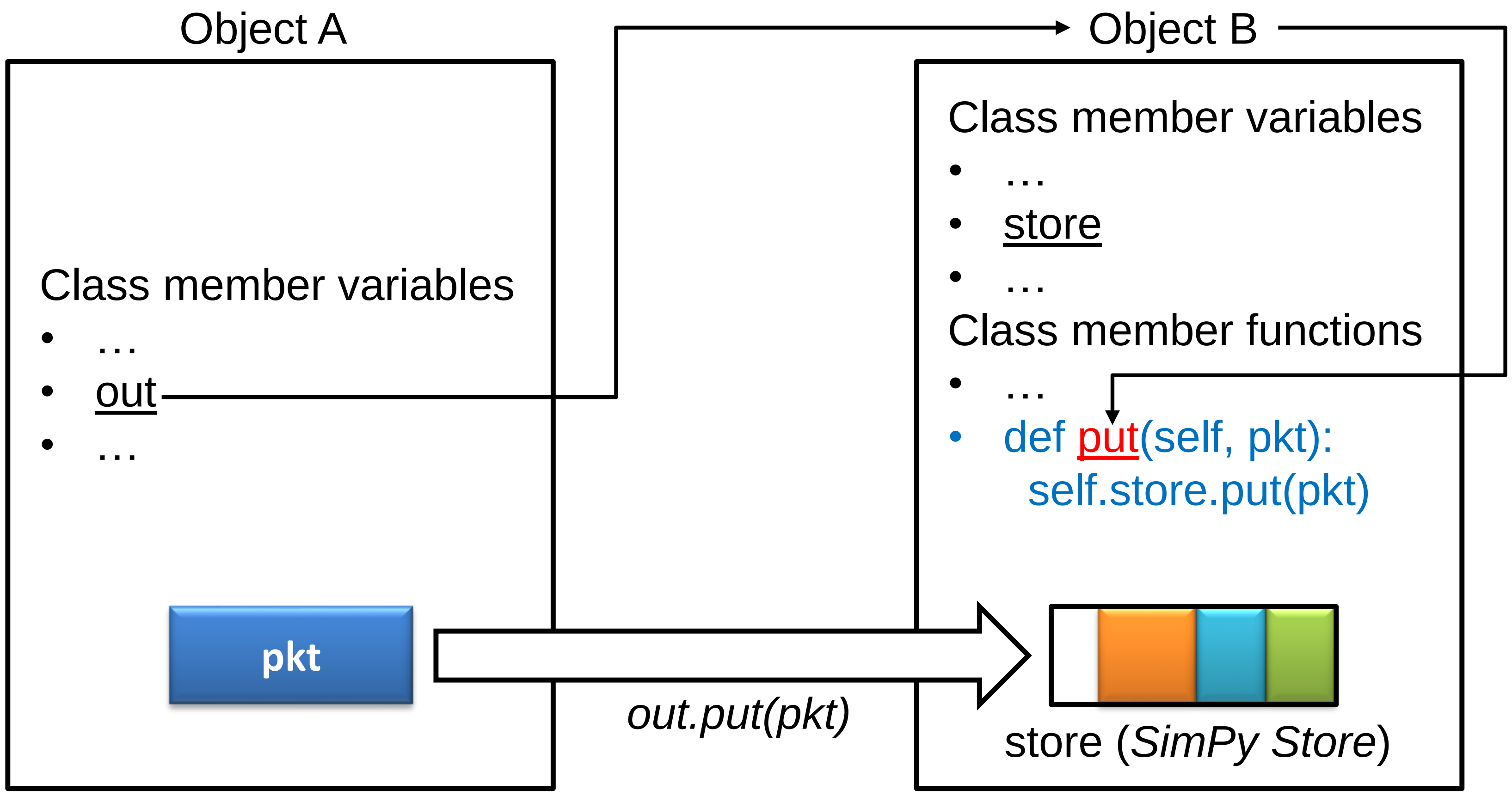
Combined Use of Leaky Bucket and Token Bucket (cont)



- Maximum burst*:
  $S = C / (M - \rho)$
  - *If the token bucket is full when the burst arrives.*

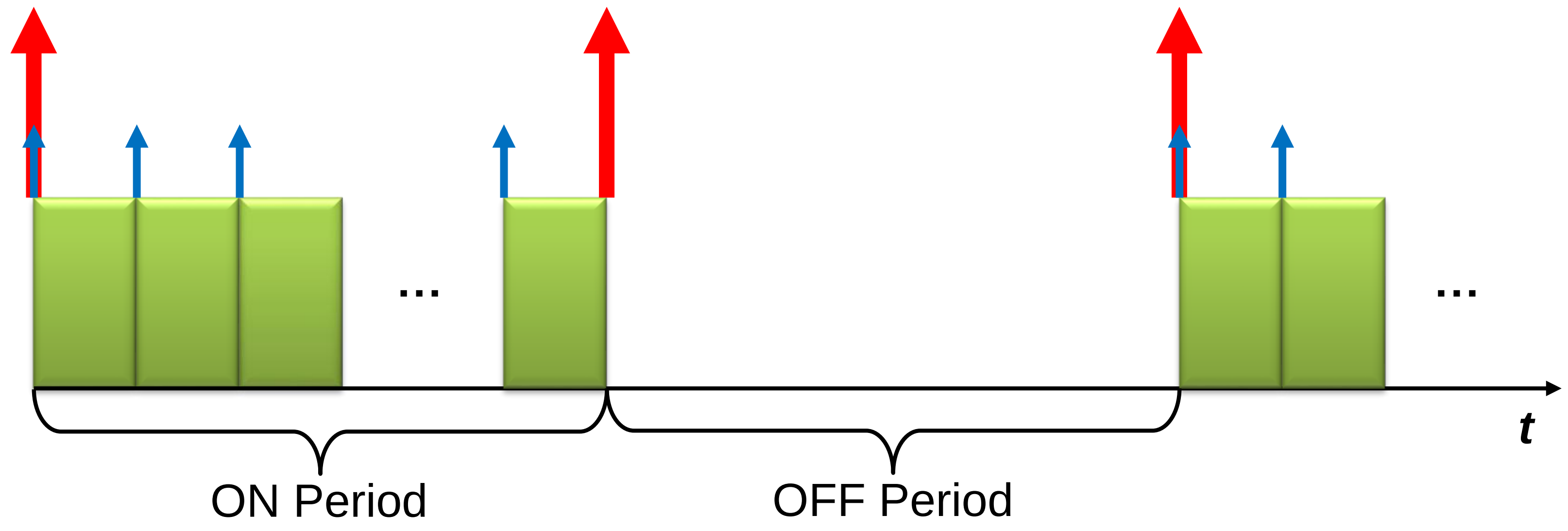
Based on the fluid model.

Modelling Packet Transmission Between Two Objects

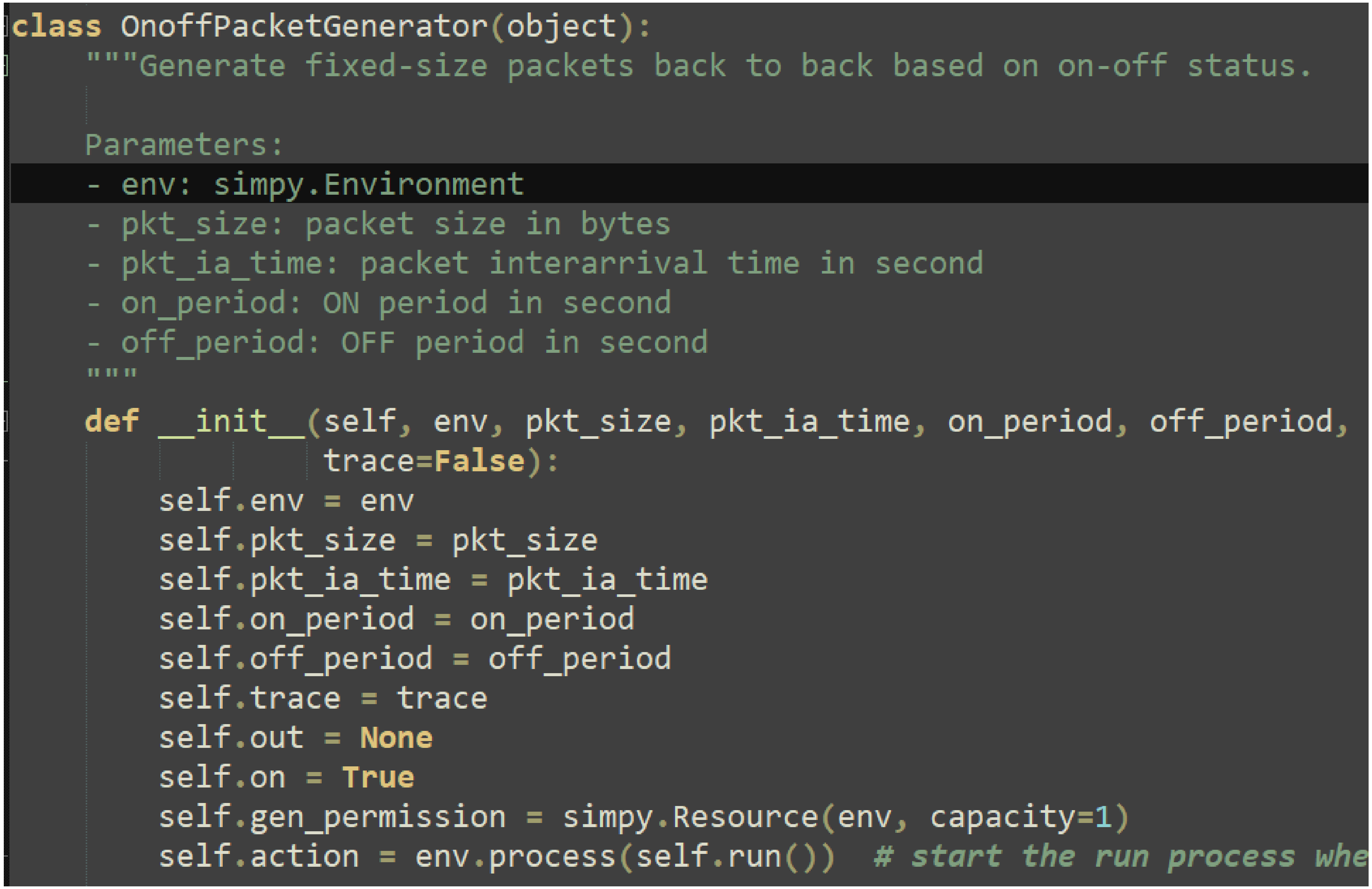


On-Off Packet Generator: Overview

- Events
  - Status change (ON↔OFF): 🡅
  - Packet generation: 🡅



On-Off Packet Generator: Initialization

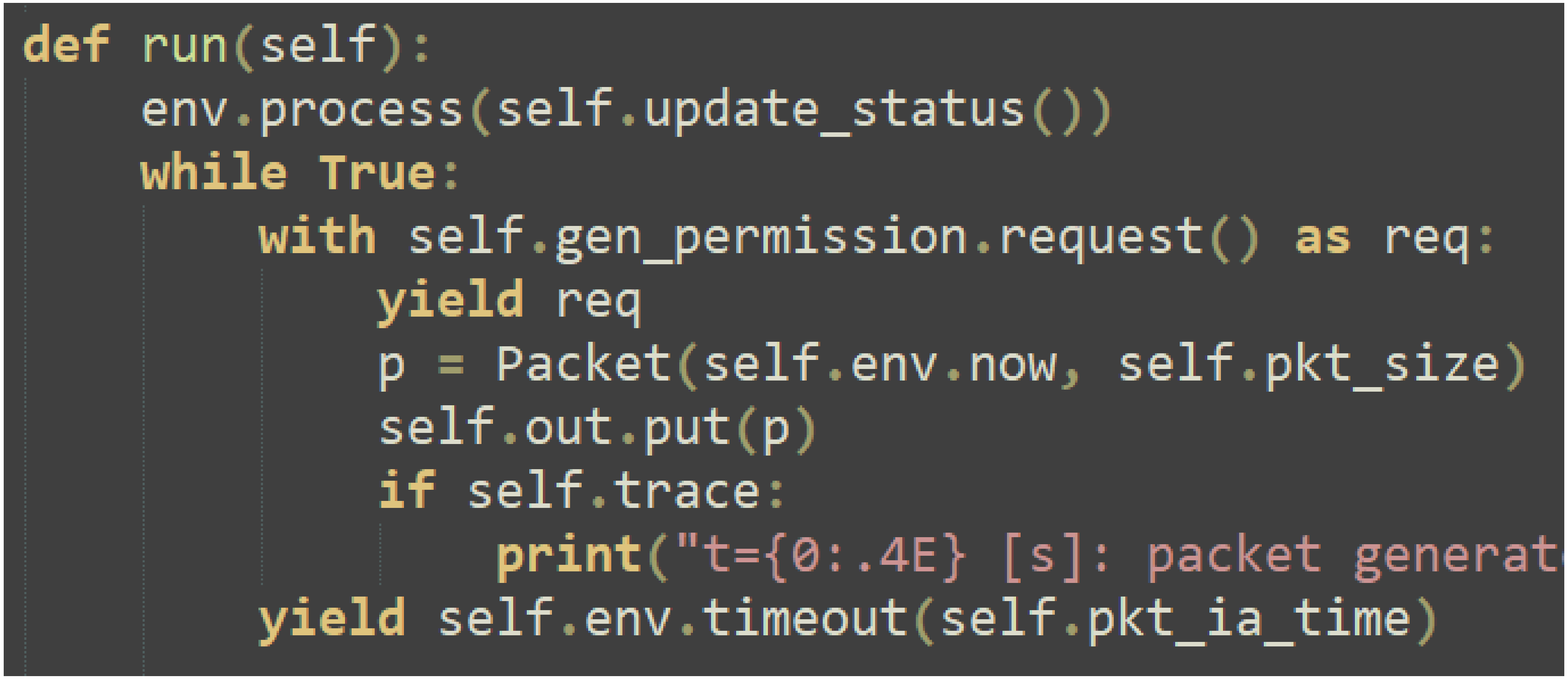


```
class OnoffPacketGenerator(object):
    """Generate fixed-size packets back to back based on on-off status.

    Parameters:
    - env: simpy.Environment
    - pkt_size: packet size in bytes
    - pkt_ia_time: packet interarrival time in second
    - on_period: ON period in second
    - off_period: OFF period in second
    """
    def __init__(self, env, pkt_size, pkt_ia_time, on_period, off_period,
                 trace=False):
        self.env = env
        self.pkt_size = pkt_size
        self.pkt_ia_time = pkt_ia_time
        self.on_period = on_period
        self.off_period = off_period
        self.trace = trace
        self.out = None
        self.on = True
        self.gen_permission = simpy.Resource(env, capacity=1)
        self.action = env.process(self.run())  # start the run process whe
```

Xi'an Jiaotong-Liverpool University
西交利物浦大學

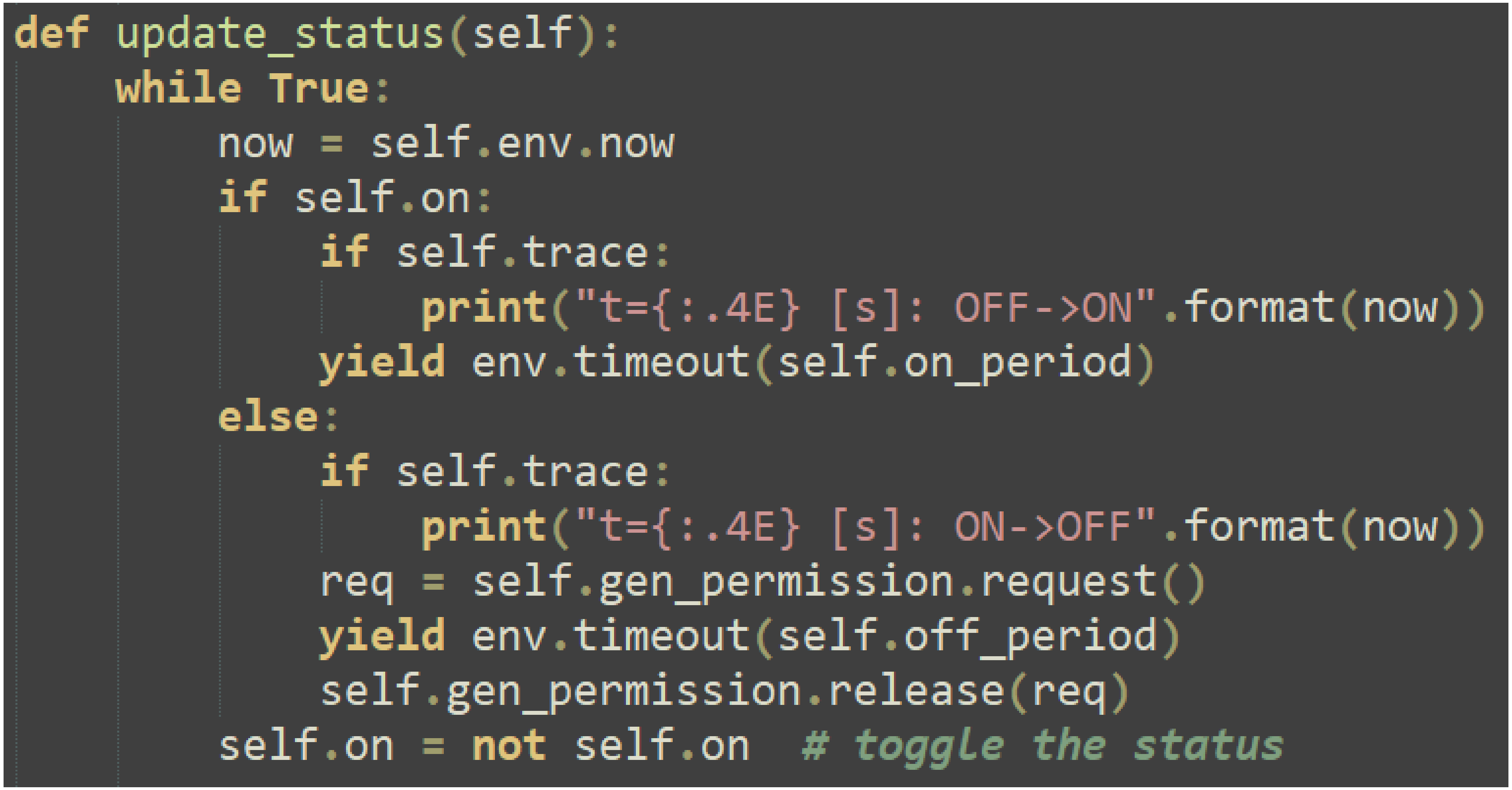
On-Off Packet Generator: Packet Generation



```
def run(self):
    env.process(self.update_status())
    while True:
        with self.gen_permission.request() as req:
            yield req
            p = Packet(self.env.now, self.pkt_size)
            self.out.put(p)
            if self.trace:
                print("t={0:.4E} [s]: packet generat
        yield self.env.timeout(self.pkt_ia_time)
```

Xi'an Jiaotong-Liverpool University
西交利物浦大学

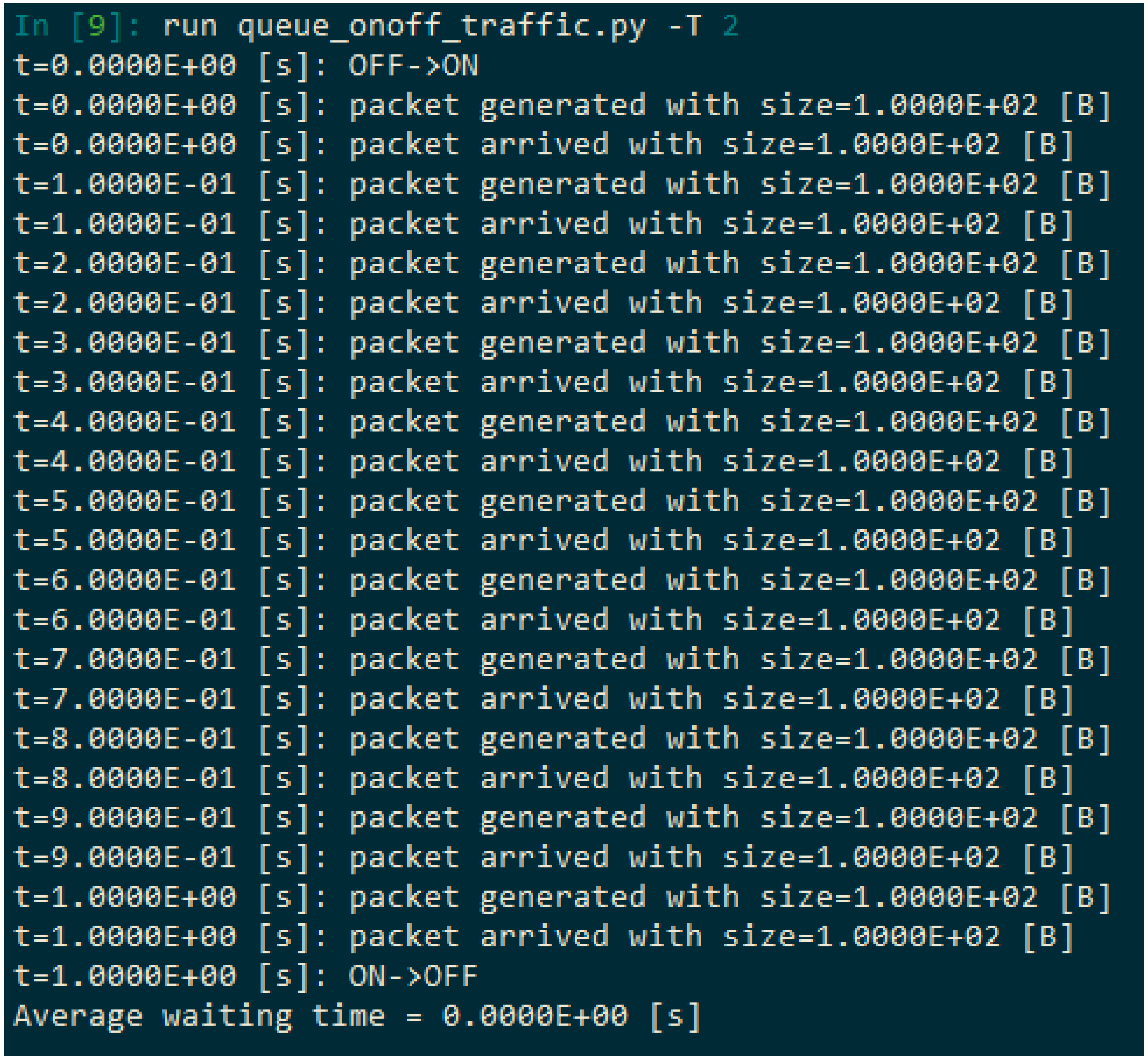
On-Off Packet Generator: Status Update



```
def update_status(self):
    while True:
        now = self.env.now
        if self.on:
            if self.trace:
                print("t={:.4E} [s]: OFF->ON".format(now))
            yield env.timeout(self.on_period)
        else:
            if self.trace:
                print("t={:.4E} [s]: ON->OFF".format(now))
            req = self.gen_permission.request()
            yield env.timeout(self.off_period)
            self.gen_permission.release(req)
        self.on = not self.on  # toggle the status
```

Xi'an Jiaotong-Liverpool University
西交利物浦大学

Sample Run*



```
In [9]: run queue_onoff_traffic.py -T 2
t=0.0000E+00 [s]: OFF->ON
t=0.0000E+00 [s]: packet generated with size=1.0000E+02 [B]
t=0.0000E+00 [s]: packet arrived with size=1.0000E+02 [B]
t=1.0000E-01 [s]: packet generated with size=1.0000E+02 [B]
t=1.0000E-01 [s]: packet arrived with size=1.0000E+02 [B]
t=2.0000E-01 [s]: packet generated with size=1.0000E+02 [B]
t=2.0000E-01 [s]: packet arrived with size=1.0000E+02 [B]
t=3.0000E-01 [s]: packet generated with size=1.0000E+02 [B]
t=3.0000E-01 [s]: packet arrived with size=1.0000E+02 [B]
t=4.0000E-01 [s]: packet generated with size=1.0000E+02 [B]
t=4.0000E-01 [s]: packet arrived with size=1.0000E+02 [B]
t=5.0000E-01 [s]: packet generated with size=1.0000E+02 [B]
t=5.0000E-01 [s]: packet arrived with size=1.0000E+02 [B]
t=6.0000E-01 [s]: packet generated with size=1.0000E+02 [B]
t=6.0000E-01 [s]: packet arrived with size=1.0000E+02 [B]
t=7.0000E-01 [s]: packet generated with size=1.0000E+02 [B]
t=7.0000E-01 [s]: packet arrived with size=1.0000E+02 [B]
t=8.0000E-01 [s]: packet generated with size=1.0000E+02 [B]
t=8.0000E-01 [s]: packet arrived with size=1.0000E+02 [B]
t=9.0000E-01 [s]: packet generated with size=1.0000E+02 [B]
t=9.0000E-01 [s]: packet arrived with size=1.0000E+02 [B]
t=1.0000E+00 [s]: packet generated with size=1.0000E+02 [B]
t=1.0000E+00 [s]: packet arrived with size=1.0000E+02 [B]
t=1.0000E+00 [s]: ON->OFF
Average waiting time = 0.0000E+00 [s]
```

Simulation code available at https://ice.xjtlu.edu.cn/mod/resource/view.php?id=47153

TBF: To Do on A Message Arrival

1. Update the amount of tokens based on
   - Amount of time passed since the last update (***see #3 below***)
   - Token generation rate and bucket size
2. Compare the message size and the token amount.
   - If the message size is larger than the token amount:
     - Wait until enough tokens are generated for the message.
       - Using `yield self.env.timeout(…)`
     - Then, set the token amount to zero.
   - Otherwise, reduce the token amount by the message size.
3. Store the current time in ***a member variable***.
4. Delay packet transmission time.
5. Send the message through the output port.