TP Performance - Introduction

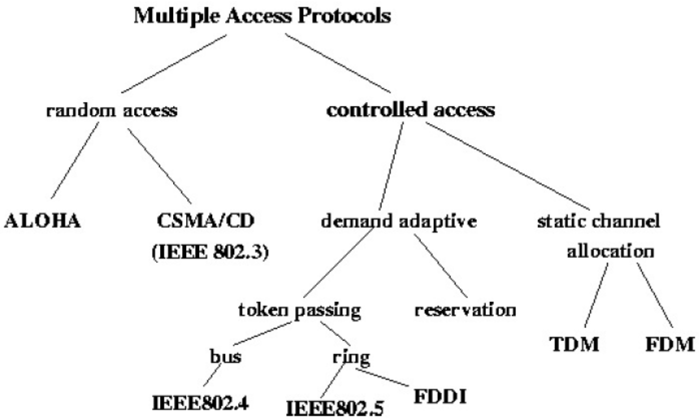
mouhcine.mendil@inria.fr

Wednesday 14th November, 2018

Introduction

Multiple access to the channel

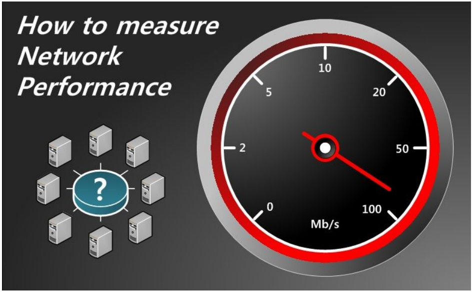
Introduction



Introduction

Performance metrics:

- Spectrum efficiency
- Latency
- Throughput
- ...



Simpy

- SimPy (GPL) is a discrete-event simulation library

```
import simpy
```

- The simulation environment manages the simulation time as well as the scheduling and processing of events

```
env = simpy.Environment() # create environement
env.process(event_generator())
env.run(until=20) # Run simulation until
# the 20th time step
```

- Simpy is event-process based:
    1. Events are wrapped in process functions (or methods)
    2. A process function creates events, yield them to be scheduled, and updates the system state

```
def packet_generator(env):
        while True:
                yield env.timeout(1) # Wait for 1 time step
                packet_count+=1
```

In practice

SimPy implementation requires:

1. A process function (or method) for each event that yields an action and updates the system state
2. An environment (scheduler) that determines the order the event processes should be called

Exercice - Simulation 1

Problem

A router is sending packets through a 64 kbps link. The length of the packets is an exponential random variable with mean 400 bytes. The interval between arrival of packets is an exponential random variable with mean 15 packets per second. During a packet transmission, the router bufferizes the arriving packets in a FIFO queue.

1. Using SimPy, simulate the corresponding Poisson process packet generator (use random.expovariate)
2. What is the average number of generated packets per time unit (use env.now) ?

SimPy ressources

- Stores are object containers (pair of lists) with a limited or unlimited capacity

```
store = simpy.Store(env, capacity=10)
```

SimPy ressources

- Stores are object containers (pair of lists) with a limited or unlimited capacity

```
store = simpy.Store(env, capacity=10)
```

- Putting elements in a store is an event

```
def producer(env, store, packet):
        """ Put a packet in the store
                each two time steps
        """
        while True:
                yield env.timeout(2)
                # Put a packet in the store
                yield store.put(packet)
```

SimPy ressources

- Stores are object containers (pair of lists) with a limited or unlimited capacity

```
store = simpy.Store(env, capacity=10)
```

- Getting elements from a store is an event

```
def consumer(env, store):
        """ Processes a packet from the store
                each time step
        """
        while True:
                yield env.timeout(1)
                # Get the packet from the store
                packet = yield store.get()
                # ... process packet
```

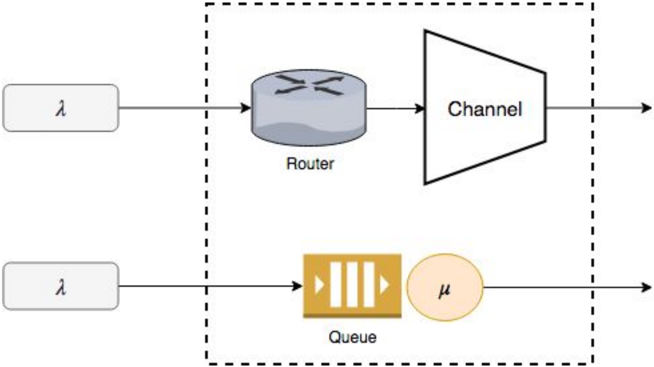
Exercice - Simulation 2

Problem

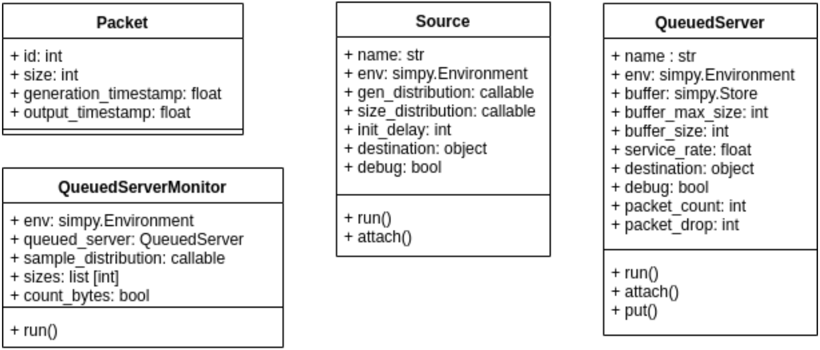
A router is sending packets through a 64 kbps link. The length of the packets is an exponential random variable with mean 400 bytes. The interval between arrival of packets is an exponential random variable with mean 15 packets per second. During a packet transmission, the router bufferizes the arriving packets in a FIFO queue.

1. Using SimPy, model the queue/server system corresponding to the router/link
2. Using SimPy, simulate the M/M/1 queue
3. Compute the following in the stationary regime:
    - Average number of packets in the system
    - Average latency of the system

Model



API



Packet

+ id: int
+ size: int
+ generation_timestamp: float
+ output_timestamp: float

QueuedServerMonitor

+ env: simpy.Environment
+ queued_server: QueuedServer
+ sample_distribution: callable
+ sizes: list [int]
+ count_bytes: bool

+ run()

Source

+ name: str
+ env: simpy.Environment
+ gen_distribution: callable
+ size_distribution: callable
+ init_delay: int
+ destination: object
+ debug: bool

+ run()
+ attach()

QueuedServer

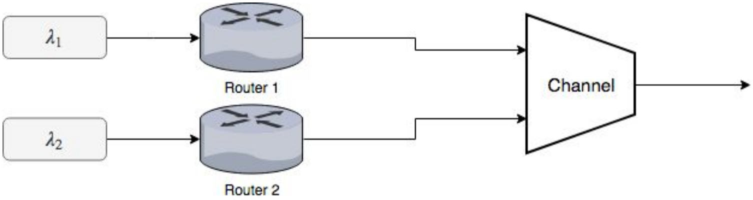
+ name : str
+ env: simpy.Environment
+ buffer: simpy.Store
+ buffer_max_size: int
+ buffer_size: int
+ service_rate: float
+ destination: object
+ debug: bool
+ packet_count: int
+ packet_drop: int

+ run()
+ attach()
+ put()

Two Routers

Problem

Two routers are sending packets through a 64 kbps link following a Poisson process of intensity λ_1 and λ_2 respectively. The length of the packets is an exponential random variable with mean 400 bytes. We suppose that there is no collision and packets can be transmitted at the same time.

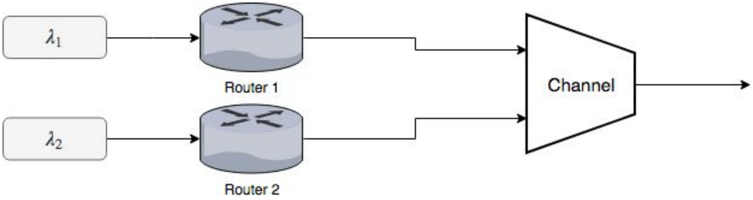


- Simulate the system using simpy for $\lambda_1 = \lambda_2 = 7.5$ packets/sec
- Average number of packet and latency?

Two Concurrent Routers

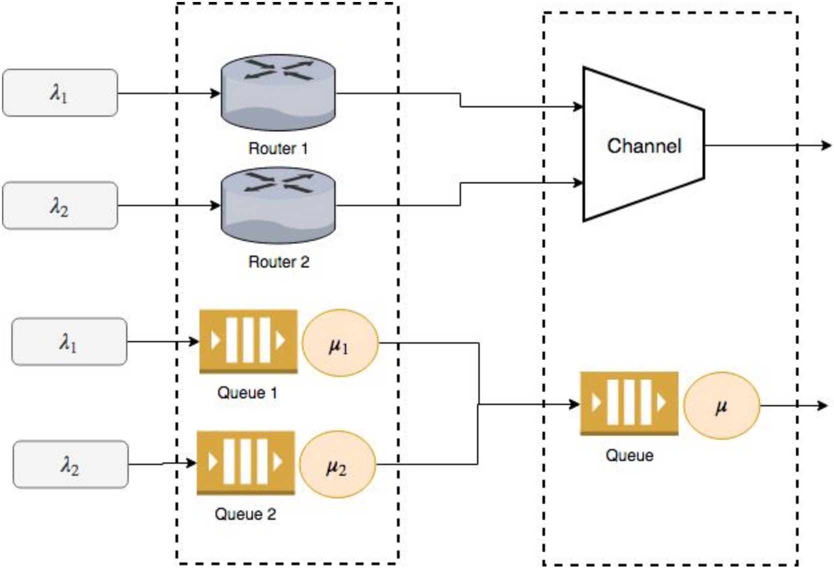
Problem

Two routers are sending packets through a 64 kbps link following independent Poisson process of intensity λ_1 and λ_2, respectively. The length of the packets is an exponential random variable with mean 400 bytes. We suppose that routers transmit their packets even when the channel is busy. The packets being transmitted concurrently collide and are discarded **at the end of their respective transmission**.



- How to model this system ?

Concurrent Network Model



Homework

1. **Friday 23rd of november** before midnight
2. Check the git repo for details