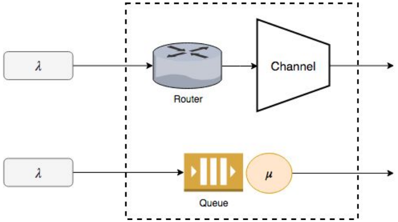
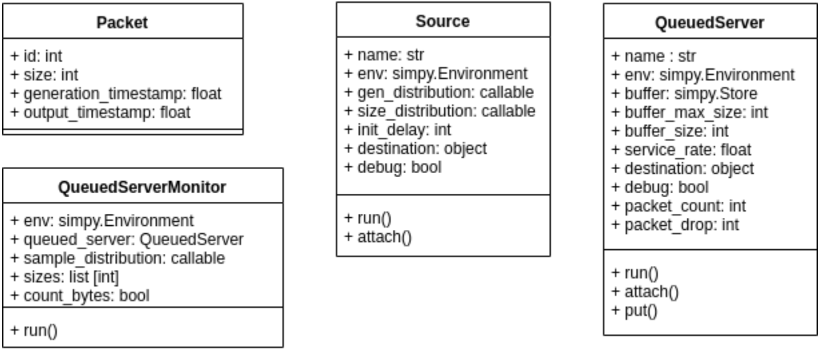
# One router model

## 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.



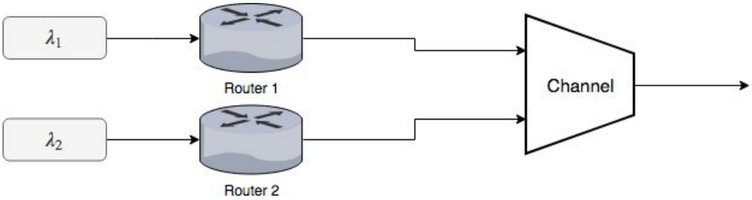
# API


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

## Problem

Two routers are sending packets through a 64 kbps link following a Poisson process of intensity $\lambda_1$ and $\lambda_2$ 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.

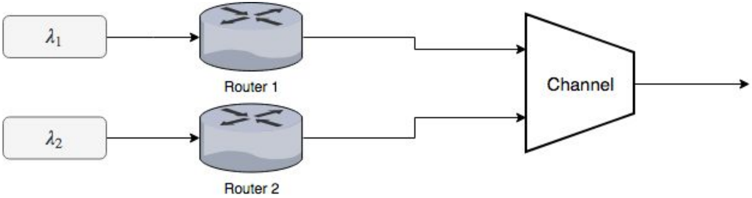



- $\lambda_1 = \lambda_2 = 7.5$ packets/sec

# Two Concurrent Routers Problem

## Problem

Two routers are sending packets through a 64 kbps link following independent Poisson process of intensity $\lambda_1$ and $\lambda_2$, 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**.




- $\lambda_1 = \lambda_2 = 7.5$ packets/sec