DESEMPENHO E DIMENSIONAMENTO DE REDES

NETWORK SIMULATION

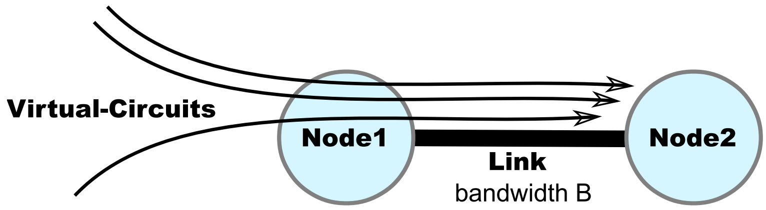
Objectives

- Virtual-circuit switched network simulation,
- Packet switched network simulation,
  - Single-queuing simulation,
  - Multi-queuing simulation,
  - Network capacity simulation.

Python references: SimPy - https://simpy.readthedocs.org/en/latest/

Basic Network Simulations and Beyond in Python - https://www.grotto-networking.com/DiscreteEventPython.html

Virtual-Circuit Switched Network



1. Consider a virtual-circuit network where nodes receive requests to establish virtual-circuits (VCs) using a limited capacity link. Node1 is connected to Node2 through a link with bandwidth *B*. Assume that: (i) all VCs require a bandwidth of b, implying that the link as a capacity for *C* VCs, with *C=B/b* (ii) the time between VC establishment requests is an exponential distributed random variable with average $1/\lambda$, and (iii) each VC is active during a time that is an exponential distributed random variable with average duration $1/\mu$.

Based on the code provided in the python script **VCSim1.py**, infer by simulation Node1 performance in terms of (i) VC blocking probability, and (ii) average link load, for the following traffic values:

- $\boldsymbol{\lambda}$ (average VC requests per minute): 1,1.5, 2, 2.5, and 3;
- **1/$\boldsymbol{\mu}$** (average VC duration – minutes): 2, 4, 6, 8, and 10;
- **b** (Mbits/sec): 2;
- **B** (Mbits/sec): 16, 24 and 32.

Plot graphical representations of the performance results, e.g., for each graph fix B, plot performance metric versus $\boldsymbol{\lambda}$, one curve for each $1/\mu$ value.

2. Compare (numerically and graphically) the simulation results with the theoretical expected performance metrics.

Blocking probability: $$\frac{\frac{\rho^C}{C!}}{\sum_{i=0}^{C}\frac{\rho^i}{i!}}, \rho = \frac{\lambda}{\mu}$$

Average link load (percentage): $$\frac{1}{C}\frac{\sum_{i=1}^{C}\frac{\rho^i}{(i-1)!}}{\sum_{i=0}^{C}\frac{\rho^i}{i!}}, \rho = \frac{\lambda}{\mu}$$

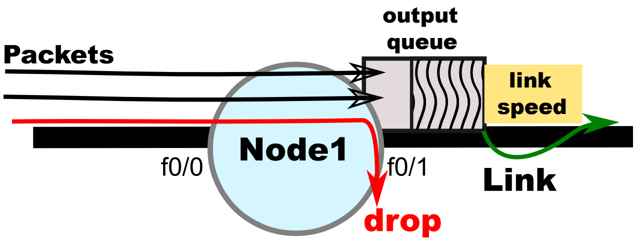
3. Consider now that VCs may be from two different types: standard and special. Standard VCs require 2 Mbits/sec and special VCs require 4Mbits/sec. Moreover, nodes may have *R* Mbits/sec reserved for the special VCs. **Create** a simulation python script **VCSim2.py**, to infer by simulation Node1 performance in terms of (i) standard VCs blocking probability, (ii) special VCs blocking probability, and (iii) average link load, for the following traffic values:

- $\boldsymbol{\lambda}$ ***standard*** (average VC requests per minute): 1, 2, 3, 4, and 5;
- $\boldsymbol{\lambda}$ ***special*** (average VC requests per minute): 1,1.5, 2, 2.5, and 3;
- **1/$\boldsymbol{\mu}$** (average VC duration – minutes): 2, 4, 6, 8, and 10;
- **B** (Mbits/sec): 16, 24, 32.
- **R** (Mbits/sec): 0, 4, 8, and 16.

Take conclusions about the influence of the reservation value on the blocking probabilities of the two types of VCs and also on the average link load.

Packet Switched Network

Single-queue simulation



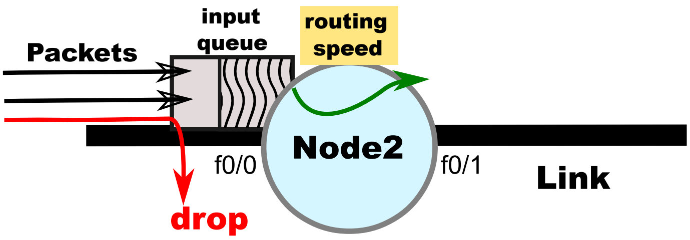
4. Assuming that a network node (Node1) is a Router with a very high routing speed connected to a 2 Mbps link. Both input and output interfaces have a limited size input and output FIFO queues, respectively. Node1 receives packets in interface f0/0 and routes them to the 2 Mbps link using the interface f0/1. Consider that the packet flows received by Node1 are characterized by: (i) the time between packet arrivals is an exponentially distributed random variable with average $1/\lambda$, and (ii) the packet size is a random integer discrete variable with the following probabilities: 50% for 64 bytes, 50% for 1500 bytes. Note that, the link transmission speed depends of the packet size.

Based on the code provided in the python script **PktSim1.py**, infer by simulation Node1 performance in terms of (i) packet loss (percentage of packets dropped due to queue overflow), (ii) average packet delay, and (iii) transmitted bandwidth, for the following traffic and queue size values:

λ (pkts/sec): 150, 300, and 450;

Queue *size* (packets): 64, 96, 128, and 10000.

5. With the performance simulation results, draw some conclusions concerning the performance of the Node1 for the different input values of packet rate (λ) and queue size. Compare the performance results obtained by simulation with theoretical values of average delay, average queue occupation and packet loss for the queuing models M/M/1, M/D/1, M/G/1 and M/M/1/K.



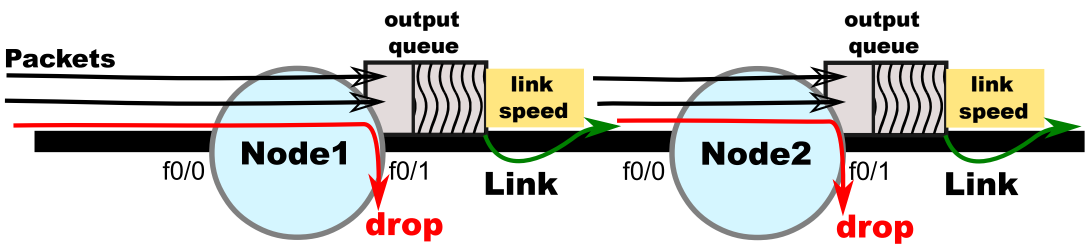
6. Assuming now a network with a node (Node2) that is a Router with a low routing speed (100 pkts/sec) connected to a 10 Gbps link. Both input and output interfaces have a limited size input and output FIFO queues, respectively. Node2 receives packets in interface f0/0 and routes them to interface f0/1. Consider that the packet flows received by Node2 are characterized by: (i) the time between packet arrivals is an exponentially distributed random variable with average $1/\lambda$, and (ii) the packet size is a random integer discrete variable with the following probabilities: 50% for 64 bytes, 50% for 1500 bytes. Note that now, the routing speed is independent of packet size.

Create a python script named **PktSim2.py**, to infer by simulation Node2 performance in terms of (i) packet loss (percentage of packets dropped due to queue overflow), (ii) average packet delay, and (iii) transmitted bandwidth, for the the same traffic values used previously.

7. With the performance simulation results, draw some conclusions concerning the performance of the Node2 for the different input values of packet rate (λ) and queue size. Compare the performance results obtained by simulation with theoretical values of average delay, average queue occupation and packet loss for the queuing models M/M/1, M/D/1, M/G/1 and M/M/1/K.

8. (EXTRA) Test Node1 and Node2 performance, now using traffic recorded on a text file and generated using one of the models inferred in the previous assignment.

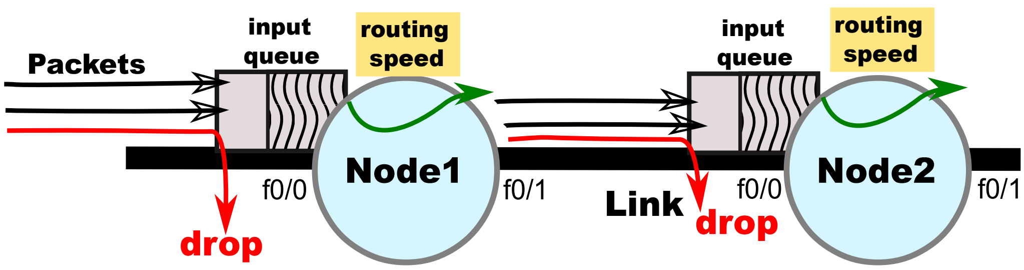
Multi-queue simulation



9. Assuming now a network with two nodes (Node1 and Node2) that are Routers with a very high routing speed interconnected and connected to 2 Mbps links. Both input and output interfaces have a limited size input and output FIFO queues, respectively. Node1 receives packets in interface f0/0 and routes them to Node2 using interface f0/1, Node2 receives the packets in interface f0/0 and routes them to the 2 Mbps link using the interface f0/1. Consider that the packet flows received by Node1 and queues have the same characteristics as before.

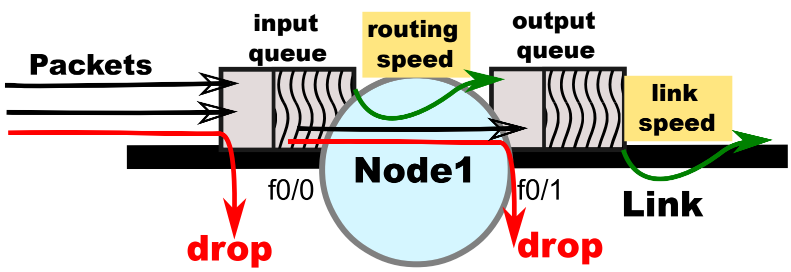
Create a python script named **PktSim3.py**, to infer by simulation Node1 and Node2 (joint) performance in terms of (i) packet loss (percentage of packets dropped due to queue overflow), and (ii) average packet delay.

10. Compare the performance results obtained by simulation with theoretical values obtained using the Kleinrock approximation, assuming that the system obeys all requirements for the approximation.



11. (EXTRA) Assuming now a network with two nodes (Node1 and Node2) that are Routers with a low routing speed (300 pkts/sec) interconnected and connected to 10 Gbps links. Both input and output interfaces have a limited size input and output FIFO queues, respectively. Node1 receives packets in interface f0/0 and routes them to Node2 using interface f0/1, Node2 receives the packets in interface f0/0 and routes them to the 10 Gbps link using the interface f0/1. Consider that the packet flows received by Node1 and queues have the same characteristics as before.

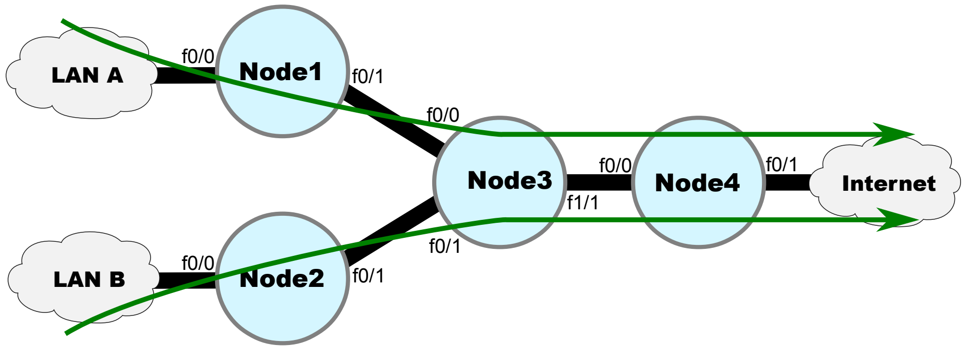
Create a python script named **PktSim4.py**, to infer by simulation Node1 and Node2 (joint) performance in terms of (i) packet loss (percentage of packets dropped due to queue overflow), and (ii) average packet delay. Compare the performance results obtained by simulation with theoretical values obtained using the Kleinrock approximation, assuming that the system obeys all requirements for the approximation.



12. Assuming now a network with a single node (Node1) that is a Router with a routing speed (300 pkts/sec) interconnected by, and connected to 2 Mbps links. Both input and output interfaces have a limited size input and output FIFO queues, respectively. Node1 receives packets in interface f0/0 and routes them to the 2 Mpbs link using interface f0/1. Consider that the packet flows received by Node1 and queues have the same characteristics as before. Note that now, both input and output queues must be considered.

Create a python script named **PktSim5.py**, to infer by simulation Node1 performance in terms of (i) packet loss (percentage of packets dropped due to queue overflow), and (ii) average packet delay. Compare the performance results obtained by simulation with theoretical values obtained using the Kleinrock approximation. Compare the performance results obtained by simulation with theoretical values obtained using the Kleinrock approximation, assuming that the system obeys all requirements for the approximation.

Multi-queue simulation with routing

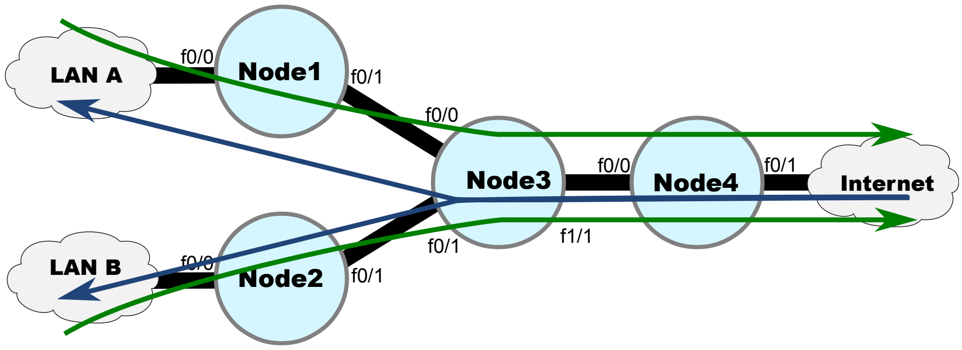


13. Assuming now a network with 4 nodes (Node1, Node2, Node3, and Node4) that are Router with a known routing speed (R pkts/sec) interconnected by 10 Mbps links. Both input and output interfaces have a limited size input and output FIFO queues, respectively. This network allows the transmission of packet flows from two LANs (LAN A and LAN B) to Internet; LANs generate packet flow where (i) the time between packet arrivals is an exponentially distributed random variable with average $1/\lambda$, and (ii) the packet size is a random integer discrete variable with the following probabilities: 50% for 64 bytes, 50% for 1500 bytes.

Note that now, both input and output queues of the nodes must be considered.

Create a python script named PktSim6.py, to infer by simulation the overall network performance in terms of (i) packet loss (percentage of packets dropped due to queue overflow), and (ii) average packet delay, for the following traffic, routing capabilities and queue characteristics:

- **λA** (pkts/sec): 150, 300, 450, and 600;
- **λB** (pkts/sec): 150, 300, 450, and 600;
- ***R*** (pkts/sec): 500, 750, and 1000.
- **Queue *size*** (packets): 64, 96, 128, 192, and 256.



14. Assuming now a network with 4 nodes (Node1, Node2, Node3, and Node4) that are Router with a known routing speed (R pkts/sec) interconnected by 10 Mbps links. Both input and output interfaces have a limited size input and output FIFO queues, respectively. This network allows the transmission of packet flows from two LANs (LAN A and LAN B) to Internet, and vice-versa; LANs/Internet generate packet flows where (i) the time between packet arrivals is an exponentially distributed random variable with average $1/\lambda$, and (ii) the packet size is a random integer discrete variable with the following probabilities: 50% for 64 bytes, 50% for 1500 bytes.

Note that now, both input and output queues of the nodes must be considered.

Create a python script named PktSim7.py, to infer by simulation the overall network performance in terms of (i) packet loss (percentage of packets dropped due to queue overflow), and (ii) average packet delay, for the following traffic, routing capabilities and queue characteristics:

- **λA→Int** (pkts/sec): 150, 300, 450, and 600;
- **λB→Int** (pkts/sec): 150, 300, 450, and 600;
- **λInt→A** (pkts/sec): 150, 300, 450, and 600;
- **λInt→B** (pkts/sec): 150, 300, 450, and 600;
- ***R*** (pkts/sec): 500, 750, and 1000.
- **Queues *sizes*** (packets): 64, 96, 128, 192, and 256.