Lecture 11 – part 2
CS/EE 5516 - Spring 1994
Performance Modeling

References:
[BG] Chapter 3: 3.1, 3.2.1, 3.2.3, 3.3.1, 3.3.2, ...
C. H. Sauer and K. M. Chandy, *Computer Systems Performance Modeling*, 2.4.2, 3.2.1-3.2.2, 5

Topics:
1. Performance measures of interest
2. Little's Law and its application
3. Poisson process and the exponential distribution
4. Memoryless property
5. Markovian queue
6. Non-Markovian Queues
7. Networks of Markovian queues

Topic 6: Non-Markovian Queues ([BG] 3.1)

Notation:
- *M/M/1* Exponential interarrival and interservice times; 1 server
- *M/M/n* Not 1 but n servers
- *M/M/1/k* Queue only has room for k customers; others lost
- *M/G/1* General interservice time distribution

  Solved by using Markov chain that exists at the instance a service departure occurs; this is called an *embedded Markov chain*

- *G/M/1* General interarrival time distribution
- *G/G/1* General interarrival and interservice distributions

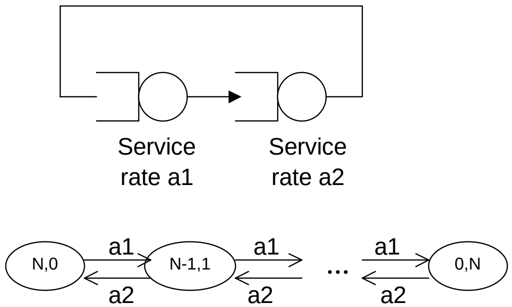
Topic 7: Networks of Markovian Queues

General comments:
- Queueing discipline at each queue must be specified (e.g., First-come, first-served)
- Forks in the network are usually assigned probabilities

- There may be multiple "classes" of customers
- "Flows" must match: flow out of a server must equal flow into all downstream servers that the output is connected to.

What is a "solution" to a queueing network?
- Finding number of customers at queue *i*

Two type of networks:
- Open
  - Good to model datagram network
  - One or more external sources of customers
  - One or more external sinks to which customers depart
  - Queues have unbounded length
  - Solve by computing "relative throughputs" starting at source nodes.
- Closed
  - Good to model sliding window
  - Population (number of customers) is fixed
  - No sources or sinks
  - Max customers at any queue is bounded above by network population
  - Solve by drawing a chain. State in each node is an k-tuple, where k is number of servers in network: $(n_0, n_1, \ldots, n_k)$
  - Key result: product form solution:
    Probability(n_0 customers at queue 0, n_1 at queue 1, …) =
    Pr(n_0 at queue 0) * Pr(n_1 at queue 1) * …


Service
rate a1
Service
rate a2
N,0
a1
a2
N-1,1
a1
a2
...
a1
a2
0,N