1. Two Customer types, Two Server Types

|  |  |
| --- | --- |
| Parameters kj = # servers of type j (j=0,1)  = Service times for customer type i on server of type j (i=0,1; j=0,1) | State Variables Qi = number of customers of type i in queue (i=0,1) – initial value 0  Sj = number of available servers of type j (j=0,1) – initially kj |

## Event Graph Component



## Definition of Events

* Arrivali – arrival of a customer of type i (i=0,1)
* StartServiceij – Start service of a customer of type i on a server if type j (i=0,1; j=0,1)
* EndServicej – end of service of a server of type j (j=0,1)

## Note:

* This formulation does not use arguments on the events.
* Only two EndServicej events are necessary, since the type of customer a server has completed has no impact on the one they will serve next.
* Other formulations are possible, some of which do use arguments on events.

## Listeners

Instantiate two ArrivalProcesses with their respective interarrival times and use adapters to schedule the respective arrivals to the server:



1. Same Situation with Just Three Events (Not Counting Run)

## Server

1-j, j

## Arrivals

### Parameters:

{U} = iid Uniform(0, 1) random variates

p = Pr{arrival is type 0}

## Event Graph



## Alternate Event Graph

* Let X ~ Bernoulli(1-p); that is,



## Listener Diagram

