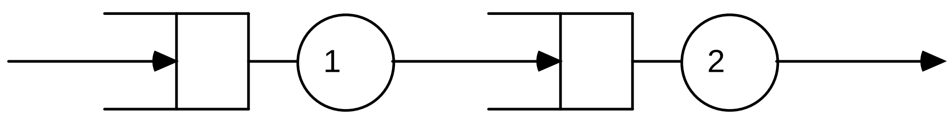
Department of Systems Engineering & Operation Research
George Mason University
OR 635 Discrete System Simulation
Prof. Chun-Hung Chen Fall 2009

Homework Assignment #1

due 4:30 pm, Wednesday, September 16

****** Due time will be strictly enforced. Late HW is subject to at least 25% penalty ******

(35 points) There is only one problem in this assignment. This problem is to make sure that your programming skills are sufficient for this course. The future computer work will be easier than this problem. Consider the following two-node system:



There are three types of events: arrival event (a), departure at node 1 (d_1), and departure at node 2 (d_2). The sizes of the waiting buffer space at both nodes are infinite. Suppose the interarrival times are 1.0 for all customers (So, the arrival times of customers are 1.0, 2.0, 3.0, 4.0, ...). The service times at node 1 are: 2.1, 2.0, 2.0, 2.0, 0.51, 0.51, 0.51, 0.51, 0.51, 0.51, 0.51, 0.51, 0.51, 0.51, ... The service times at node 2 are: 1.0, 2.1, 3.0, 3.0, 3.5, 3.5, 0.1, 0.2, 0.3, 0.43, 0.43, 0.43, 0.43, 0.43, 0.43, 0.43, 0.43, ...

Assumption: For simplicity, if any two or three events have the same event times, we assume that event (a) happens before (d_1), and (d_1) happens before (d_2). This implies (a) happens before (d_2).

Write a program (you can use any computer language: C, FORTRAN, Basic, etc. Excel is fine too.) to simulate this system. In your program, you have to write out the following data step by step (like what I did in class for a single node example): simulation clock, triggering event, # of customers at node 1, # of customers at node 2, and event list. Stop this simulation when the 15th customer leaves this system and calculate the average system time and throughput (average number of customers served per unit time, i.e., $15/\text{Total Simulation Time}$)

You have to hand in the printout of your program, the above simulation results, average system time and throughput.

Here is the result of the first 10 stages.

simul. clock	e'	# of customers at node 1	node 2	gone	Event List
0.00	Initialization	0	0	0	(a, 1.00)
1.00	a	1	0	0	(a, 2.00) (d1, 3.10)
2.00	a	2	0	0	(a, 3.00) (d1, 3.10)
3.00	a	3	0	0	(a, 4.00) (d1, 3.10)
3.10	d1	2	1	0	(a, 4.00) (d1, 5.10) (d2, 4.10)
4.00	a	3	1	0	(a, 5.00) (d1, 5.10) (d2, 4.10)
4.10	d2	3	0	1	(a, 5.00) (d1, 5.10)
5.00	a	4	0	1	(a, 6.00) (d1, 5.10)
5.10	d1	3	1	1	(a, 6.00) (d1, 7.10) (d2, 7.20)
6.00	a	4	1	1	(a, 7.00) (d1, 7.10) (d2, 7.20)
7.00	a	5	1	1	(a, 8.00) (d1, 7.10) (d2, 7.20)

Hints:

1. **The following algorithm is highly helpful.**
2. **To avoid starting from scratch, you can utilize the C code for one-node system which can be downloaded from our class web page.**
3. **Finish the simulation by hand before doing the computer program.**

Algorithm for this problem

```
#include <math.h>
#include <stdio.h>
#include <string.h>
#include <errno.h>

float et[3];
float t;
float min_et, ast;
int x1, x2;
```

```
int arrive, gone;
int e;
int i;
float ia[40]={1, 1, 1, 1, ..... };
float s1[30]={ ....};
float s2[25]={ ...};
int cta, ct1, ct2;
float ar_t[40];
float de_t[15];

main{
INITIALIZATION
while ( gone < 15 )
    {
    e = arg min_{i∈Γ(x)} { et[i] };
    t = et[e];
    if( e==0) new_arrival();
    else if(e==1) departure1();
    esle departure2();
    Print out the simulation results in this iteration
    }
Calculate Ave. ST and TP
}
```

$* \; e = \arg \min_{i \in \Gamma(x)} \{ et[i] \} \; *$

```
    e=0;
    min_et = et[0];
    if(x1 > 0  &&  et[1] < min_et) {e = 1; min_et = et[1];}
    if(x2 > 0  &&  et[2] < min_et) {e = 2; min_et = et[2];}
```

```
* new_arrival() *
    x1 = x1+1;
    update et[0];
    cta = cta+1;
    if(x1==1) {update et[1]; ct1 = ct1+1;}
    update the arrival time statistics

* departure1() *
    x1 = x1-1;
    x2 = x2+1;
    if(x1>=1) {update et[1]; ct1 = ct1+1;}
    if(x2==1) {update et[2]; ct2 = ct2+1;}


* departure2() *
    x2 = x2-1;
    gone = gone+1;
    if(x2>=1) {update et[2]; ct2 = ct2+1;}
    update the departure time statistics

* update et[2] *
    et[2] = t + s2[ct2];
```