Single-Server Queue

Hui Chen, Ph.D.

Department of Engineering & Computer Science

Virginia State University

Outline

- Discussion on project 0
- Single-server queue
  - Concept model
  - Specification model
  - Simulation model and program
  - Numerical examples (Test cases for simulation program)
  - Job-averaged statistics
  - Time-averaged statistics
  - Applications

Single-Server Queue

- A single-server service node consists of a server plus its queue
- Example Applications
  - Switches & routers
    - Telephony switching
    - Frame/packet forwarding (switching & routing)
  - Blanket paging in PCS
  - Single-CPU server
  - Single elevator building
  - Drive-by restaurant with a single waiter
  - ……

Building DES Model

- Algorithm 1.1: How to develop a model?
  - Determine goals and objectives
  - Build a conceptual model
  - Convert into a specification model
  - Convert into a computational model
  - Verify: do we build the model right (do we meet the specification)?
  - Validate: do we build the right model (do we analyze the system to be analyzed)?
- An iterative process

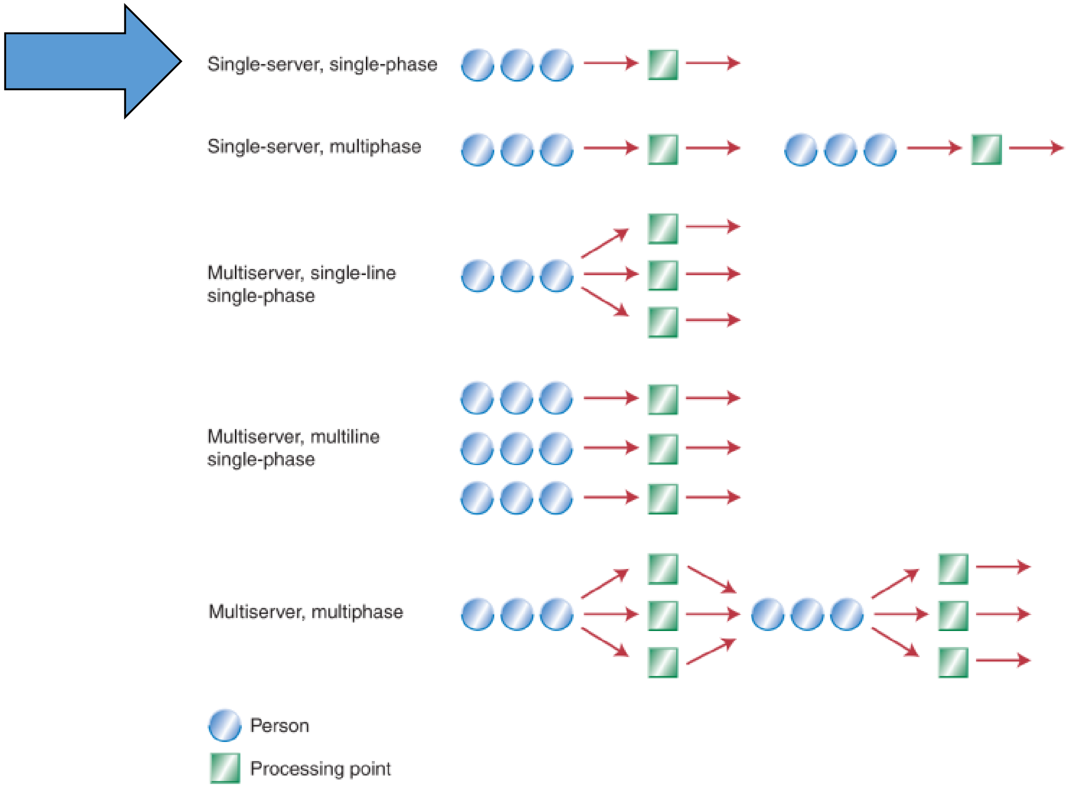
Building DES Model: Three Levels

- Conceptual
  - How comprehensive should the model be?
  - What are the state variables, which are dynamic, which are stochastic, which are important?
  - System diagrams
- Specification
  - On paper
  - May involve equations, pseudo-code, algorithms, etc
  - How will the model receive input, what the output are
- Computational
  - A computer program
  - General purpose or simulation programming language?

Verification vs. Validation

- Verification
  - Did we build the model right?
    - Computational model should be consistent with specification
- Validation
  - Did we building the right model?
    - Computational model should be consistent with the system analyzed
    - Can an expert distinguish simulation output from system output?

Single-Server Queue

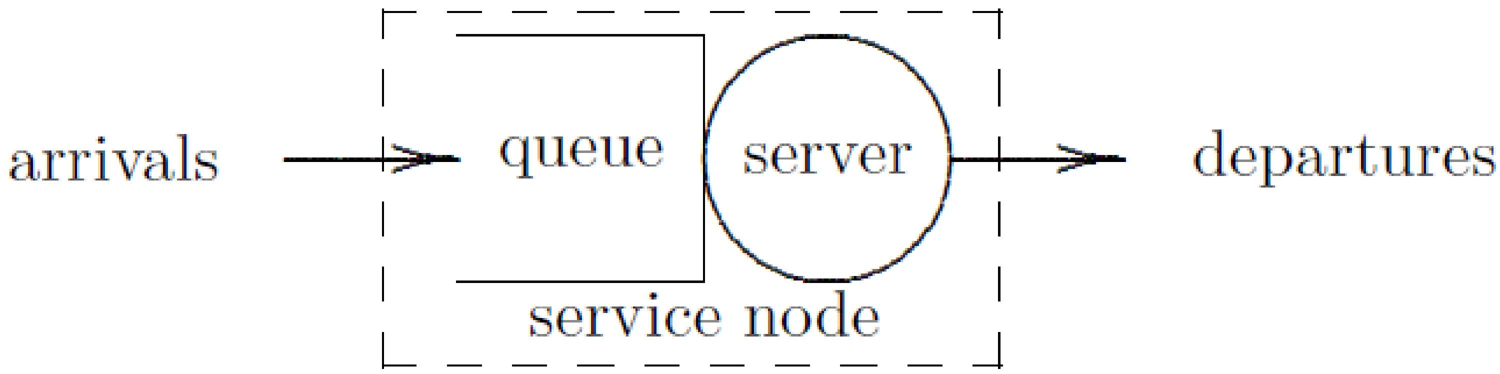


- From "[Dear Mona, Which Is The Fastest Check-Out Lane At The Grocery Store?]" by Mona Chalabi, originally appears in Operations Management, 5th Edition by "R. Dan Reid, Nada R. Sanders", 2010

Let's Answer a Few Questions

- What should the goals and objectives be?
- What should the conceptual model be?
  - How comprehensive should the model be?
  - What are the state variables, which are dynamic, which are stochastic, which are important?
  - Can we illustrate the conceptual model in a diagram?

System Diagram



Queue and Service Model

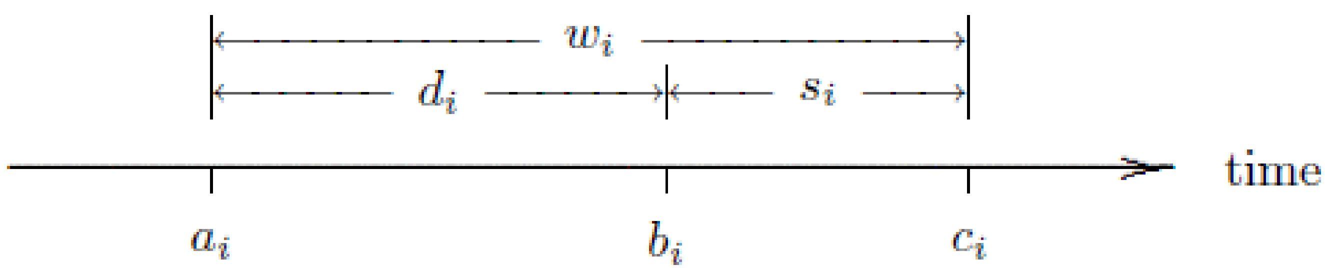
- Queue
  - Queuing discipline: how to select a job from the queue
    - FIFO/FCFS: first in, first out/first come, first serve
    - LIFO: last in, first out
    - SIRO: serve in random order
    - Priority: *e.g.*, shortest job first (SJF)
  - Capacity
  - Unless otherwise noted, assume FIFO with *infinite* queue capacity
- Service model
  - Non-preemptive
    - Once initiated, service of job will continue until completed
  - Conservative
    - Server will never remain idle if there is any job in the service node

Let's Answer a Few More Questions

- How do we specify the model?
  - How will the model receive input, what the output are?
  - How will the input affect the output?
  - How will we meet the goals and objectives?
- We need to specify those without ambiguity.

Specification

- *Arrival* time: a_i
- *Delay* in queue (queuing delay): d_i
- Time that service begins: $b_i = a_i + d_i$
- *Service* time: s_i
- *Wait* in the node (total delay): $w_i = d_i + s_i$
- *Departure* time: $c_i = a_i + w_i$



Arrivals

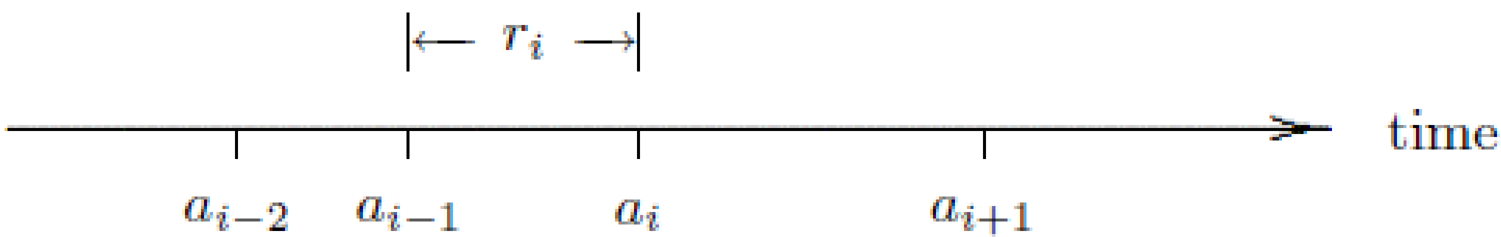
- Inter-arrival time between jobs *i-1* and *i*

  $r_i = a_i - a_{i-1}$

  where $r_1 = 0$

- Note

  $a_i = a_{i-1} + r_i = r_1 + r_2 + \ldots + r_i$

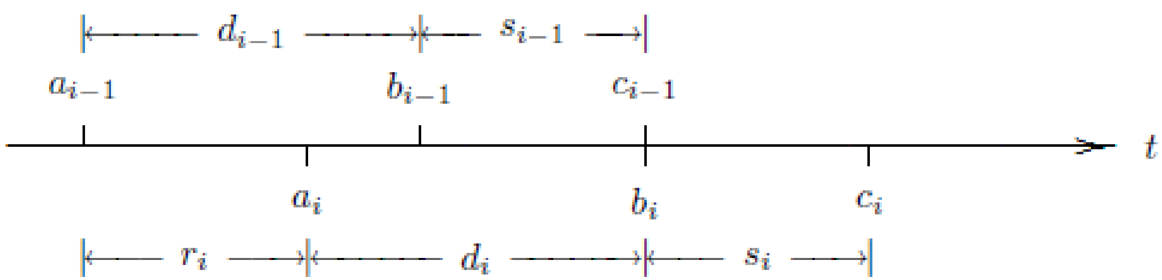


Let's Answer a Few More Questions

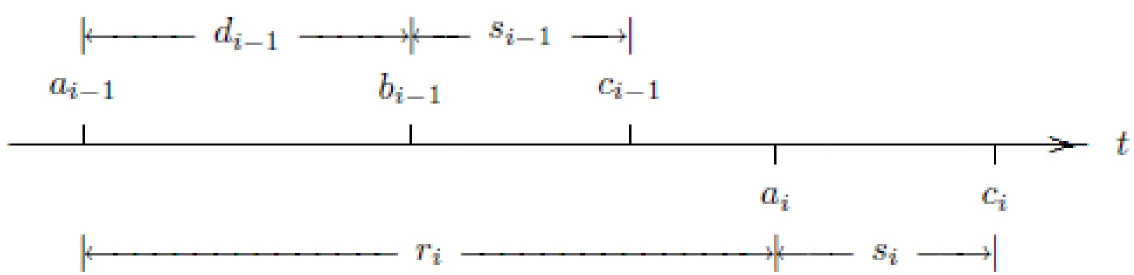
- Given the arrival times and service times, how may the delay times be computed? (Computational Model)

How do jobs experience delay?

- If $a_i < c_{i-1}$, job *i* arrives before job *i-1* completes



- If $a_i \geq c_{i-1}$, job *i* arrives after job *i-1* completes



Algorithm 1.2.1 Delay of Each Job (Single-Server FIFO Service Node with Infinite Capacity)

```
c_0 = 0.0;                      /* assumes that a_0 = 0.0 */
i = 0;
while ( more jobs to process ) {
    i++;
    a_i = GetArrival();
    if (a_i < c_{i-1})
        d_i = c_{i-1} - a_i;
    else
        d_i = 0.0;
    s_i = GetService();
    c_i = a_i + d_i + s_i;
}
n = i;
return d_1, d_2, ..., d_n;
```

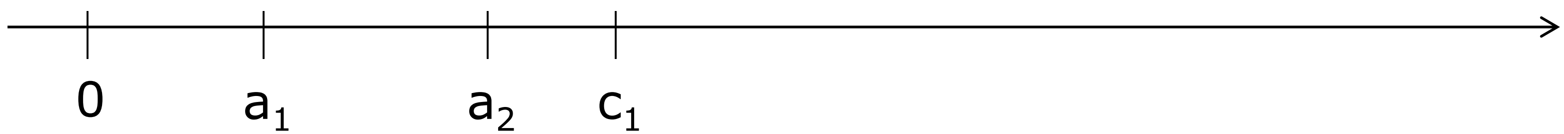
Trace-driven Simulation

- Simulation driven by external data (i.e., a trace)
- Trace can be a running record of a real system

Algorithm 1.2.1 Processing 10 Jobs

	i	1	2	3	4	5	6	7	8	9	10
read from file	a_i	15	47	71	111	123	152	166	226	310	320
from algorithm	d_i	0	11	23	17	35	44	70	41	0	26
read from file	s_i	43	36	34	30	38	40	31	29	36	30

- Running algorithm manually
  - $a_1 = 15, s_1 = 43, d_1 = ?$



  - $a_2 = 47, d_2 = ?$

Let's Answer a Few More Questions

- What were our goals and objectives?

Output Statistics

- Gain insight from various statistics!
- Examples
  - Job/Customer perspective: waiting time
  - Managing perspective: utilization
- Job-averaged statistics
- Time-average statistics

Job-Averaged Statistics (1)

- Average inter-arrival time

$$\bar{r} = \frac{1}{n}\sum_{i=1}^{n} r_i = \frac{a_n}{n}$$

  - Arrival rate: inverse of average inter-arrival time

- Average service time

$$\bar{s} = \frac{1}{n}\sum_{i=1}^{n} s_i$$

  - Service rate: inverse of average service time

Exercise L2-1

	i	1	2	3	4	5	6	7	8	9	10
read from file	a_i	15	47	71	111	123	152	166	226	310	320
from algorithm	d_i	0	11	23	17	35	44	70	41	0	26
read from file	s_i	43	36	34	30	38	40	31	29	36	30

- Examine the above 10 Jobs without running Algorithm 1.2.1
  - *Average inter-arrival time?*
  - *Average service time?*
  - *Arrival rate?*
  - Service rate?
  - *What conclusion can you draw from the above statistics?*
    - *Hint: compare arrival rate and service rate*

Job-Averaged Statistics (2)

- Average delay

$$\overline{d} = \frac{1}{n}\sum_{i=1}^{n} d_i$$

- Average wait

$$\overline{w} = \frac{1}{n}\sum_{i=1}^{n} w_i$$

- Since $w_i = d_i + s_i$

$$\overline{w} = \frac{1}{n}\sum_{i=1}^{n} w_i = \frac{1}{n}\sum_{i=1}^{n}(d_i + s_i) = \frac{1}{n}\sum_{i=1}^{n} d_i + \frac{1}{n}\sum_{i=1}^{n} s_i = \overline{d} + \overline{s}$$

Exercise L2-2

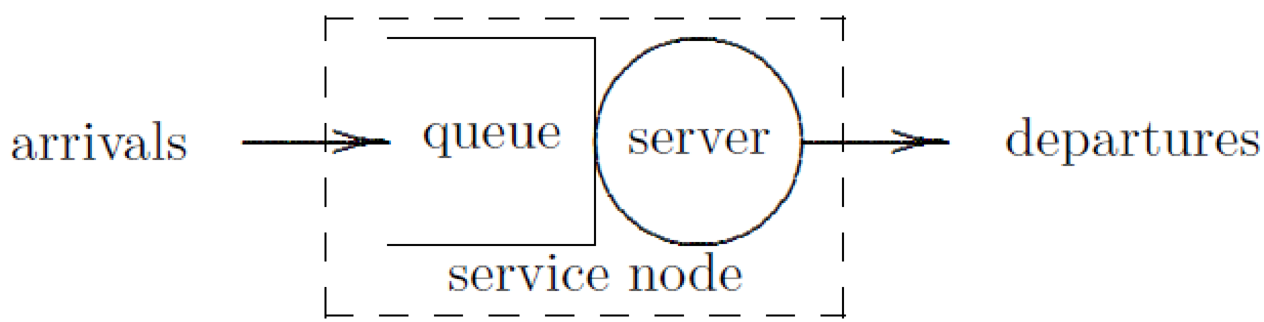
	i	1	2	3	4	5	6	7	8	9	10
read from file	a_i	15	47	71	111	123	152	166	226	310	320
from algorithm	d_i	0	11	23	17	35	44	70	41	0	26
read from file	s_i	43	36	34	30	38	40	31	29	36	30

- Use Algorithm 1.2.1 Processing 10 Jobs
  - Average delay?
  - Average wait?
  - Consistency check (part of verification)

$$\overline{w} = \frac{1}{n}\sum_{i=1}^{n} w_i = \frac{1}{n}\sum_{i=1}^{n}(d_i + s_i) = \frac{1}{n}\sum_{i=1}^{n} d_i + \frac{1}{n}\sum_{i=1}^{n} s_i = \overline{d} + \overline{s}$$

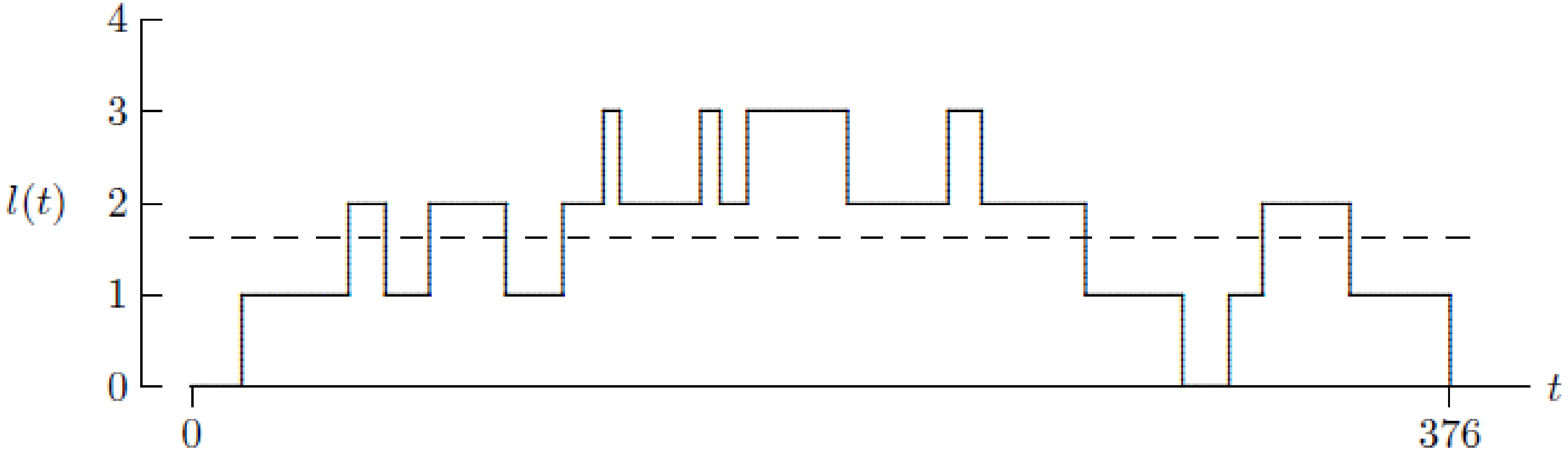
Time-Averaged Statistics (1)

- Defined by the area under a curve (integration)
- Single-Server Queue: Start with *statistics at time t*
  - $l(t)$: number of jobs in the service node at time t
  - $q(t)$: number of jobs in the queue at time t
  - $x(t)$: number of jobs in service at time t
- By definition: $l(t) = q(t) + x(t)$



Time-Averaged Statistics: Example of $l(t)$

	i	1	2	3	4	5	6	7	8	9	10
read from file	a_i	15	47	71	111	123	152	166	226	310	320
from algorithm	d_i	0	11	23	17	35	44	70	41	0	26
read from file	s_i	43	36	34	30	38	40	31	29	36	30



Time-Averaged Statistics (2)

- Defined by the area under a curve (integral)
  - Over the time interval $(0, \tau)$ the time-averaged number in the node

$$\bar{l} = \frac{1}{\tau}\int_0^{\tau} l(t)dt$$

  - Over the time interval $(0, \tau)$ the time-averaged number in the queue

$$\bar{q} = \frac{1}{\tau}\int_0^{\tau} q(t)dt$$

  - Over the time interval $(0, \tau)$ the time-averaged number in service

$$\bar{x} = \frac{1}{\tau}\int_0^{\tau} x(t)dt$$

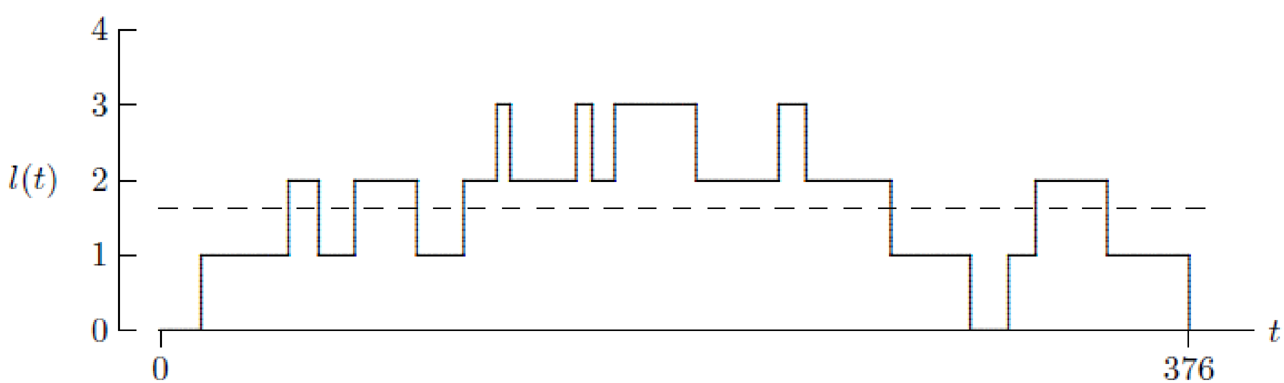
Time-Averaged Statistics (3)

- Defined by the area under a curve (integral)
  - Over the time interval $(0, \tau)$

$$\bar{l} = \frac{1}{\tau}\int_0^{\tau} l(t)dt \qquad \bar{q} = \frac{1}{\tau}\int_0^{\tau} q(t)dt \qquad \bar{x} = \frac{1}{\tau}\int_0^{\tau} x(t)dt$$

  - Since $l(t) = q(t) + x(t)$ for all $t > 0$,

$$\bar{l} = \bar{x} + \bar{q}$$



Job-Averaged and Time-Averaged Statistics

- Little's Equations
- If
  - (a) queue discipline is FIFO
  - (b) service node capacity is infinite, and
  - (c) service is idle both at $t=0$ and $t=c_n$,
- Then

$$\int_0^{c_n} l(t)dt = \sum_{i=1}^{n} w_i$$

$$\int_0^{c_n} q(t)dt = \sum_{i=1}^{n} d_i$$

$$\int_0^{c_n} x(t)dt = \sum_{i=1}^{n} s_i$$

Exercise L2-3

	i	1	2	3	4	5	6	7	8	9	10
read from file	a_i	15	47	71	111	123	152	166	226	310	320
from algorithm	d_i	0	11	23	17	35	44	70	41	0	26
read from file	s_i	43	36	34	30	38	40	31	29	36	30

- Use Little's Equations to calculate

$$\bar{q}$$

$$\bar{l}$$

$$\bar{x}$$

Server Utilization

- Sever utilization: time averaged number in service
  - Represents probability that the server is busy

$$\overline{x} = \frac{1}{\tau}\int_0^{\tau} x(t)dt$$

Traffic Intensity

- Traffic intensity: ratio of arrival rate to service rate

$$\frac{1/\overline{r}}{1/\overline{s}} = \frac{\overline{s}}{\overline{r}} = \frac{\overline{s}}{a_n/n} = \left(\frac{c_n}{a_n}\right)\overline{x}$$

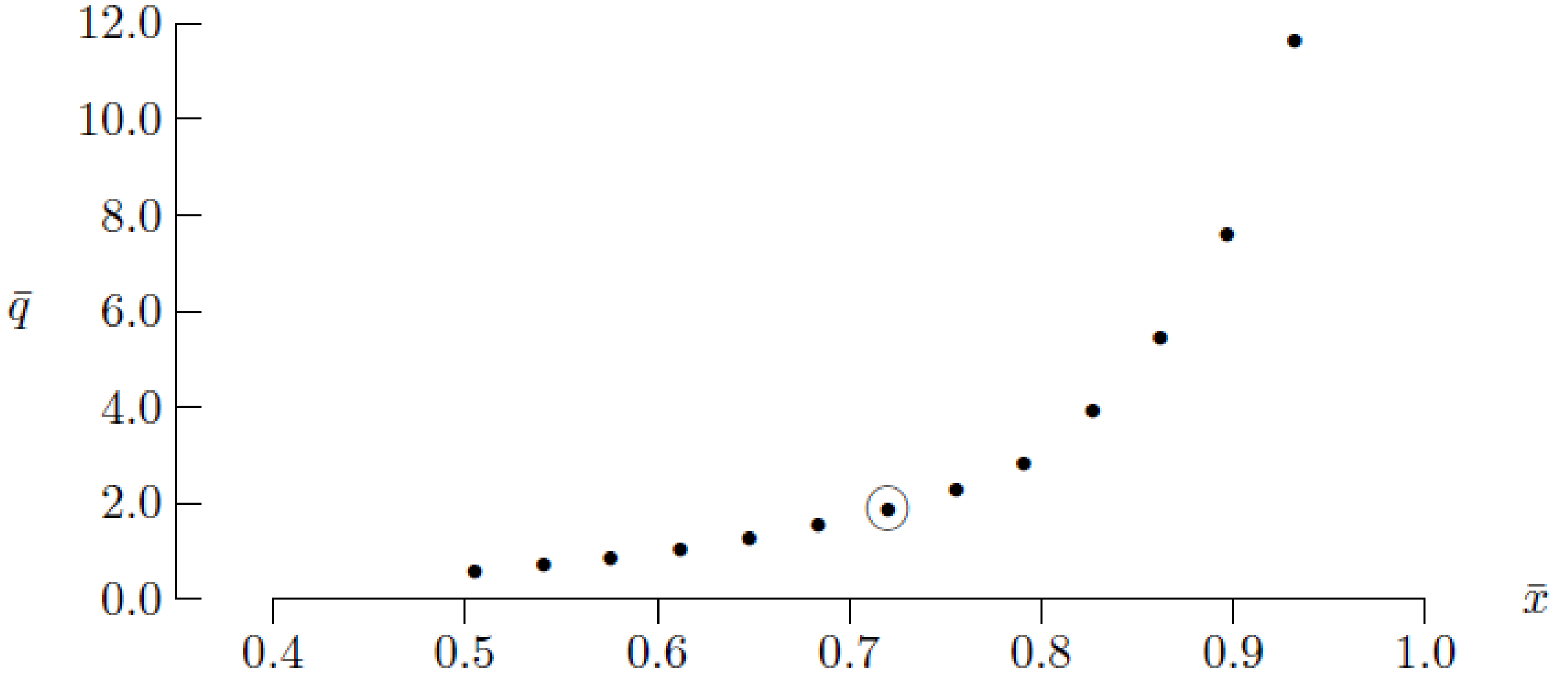
Large Trace?

- Write a program!
- Instructor demonstration in either of these two programming languages to implement Algorithm 1.2.1
  - C/C++
  - Java

Case Study

- Sven and Larry's Ice Cream Shoppe
  - Owners considering adding new flavors and cone options
  - Concerned about resulting service times and queue length
- Can be modeled as a single-server queue
  - ssq1.dat represents 1000 customer interactions
  - Direct consequence of adding new flavors and cone options
    - Service time per customer increases
  - What's the consequence?

Ice Cream Shoppe



Exercise: L2-4

- Develop a simulation program to implement Algorithm 1.2.1 in your favorite programming language
  - Let's call the program ssq1
  - In the program, output all job-average statistics
  - Add consistency check to the program
- Verify the program
  - Perform consistency check
  - Create a test case using exercises L2-1, L2-2, and L2-3 and apply the test case to your program
- Use a large trace
  - Run your program using the provided "large" trace as input, observe the output

Exercise: L2-5

- Modify your program *ssq1* to output the additional statistics

$$\bar{q} \quad \bar{l} \quad \bar{x}$$

- *As* in the case study (Sven and Larry's Ice Cream Shoppe), use this program to compute a table of the above three statistics for the traffic intensities that are 0.6, 0.7, 0.8, 0.9, 1.0, 1.1 and 1.2 times of original one in the input file
- Illustrate your result using Matlab/Octave, Excel, or any other graphing software of your choice
  - When illustrating the result, think about what message you want to convey in your graph

Summary

- Single-server queue
  - Concept model
  - Specification model
  - Simulation model and program
  - Numerical examples (Test cases for simulation program)
  - Job-averaged statistics
  - Time-averaged statistics
  - Applications
- Graphing consideration