ECE/CS 541
Computer System Analysis:
Intro to Queueing Theory

Mohammad A. Noureddine
Coordinated Science Laboratory
University of Illinois at Urbana-Champaign

Fall 2018

Learning Objectives

- Or what is this course about?

- At the start of the semester, you should have
  - Basic programming skills (C++, Python, etc.)
  - Basic understanding of probability theory (ECE313 or equivalent)

- At the end of the semester, you should be able to
  - Understand different system modeling approaches
    - Combinatorial methods, state-space methods, etc.
  - Understand different model analysis methods
    - Analytic/numeric methods, simulation
  - Understand the basics of discrete event simulation
  - Design simulation experiments and analyze their results
  - Gain hands-on experience with different modeling and analysis tools

Announcements

- **Midterm on Tuesday November 6, 2018**
  - In class
  - Closed book, one A4 sheet
  - Everything include **up until lecture on Tuesday October 30**

Outline for Today

- Introduction to Queues
- Some notation before we start
- Little's Law

Why Queues?

- Because the **DMV**!
  - Your trips to the DMV will now never be the same

- System generally have limited capacity to perform service
  - Queue are bound to be formed

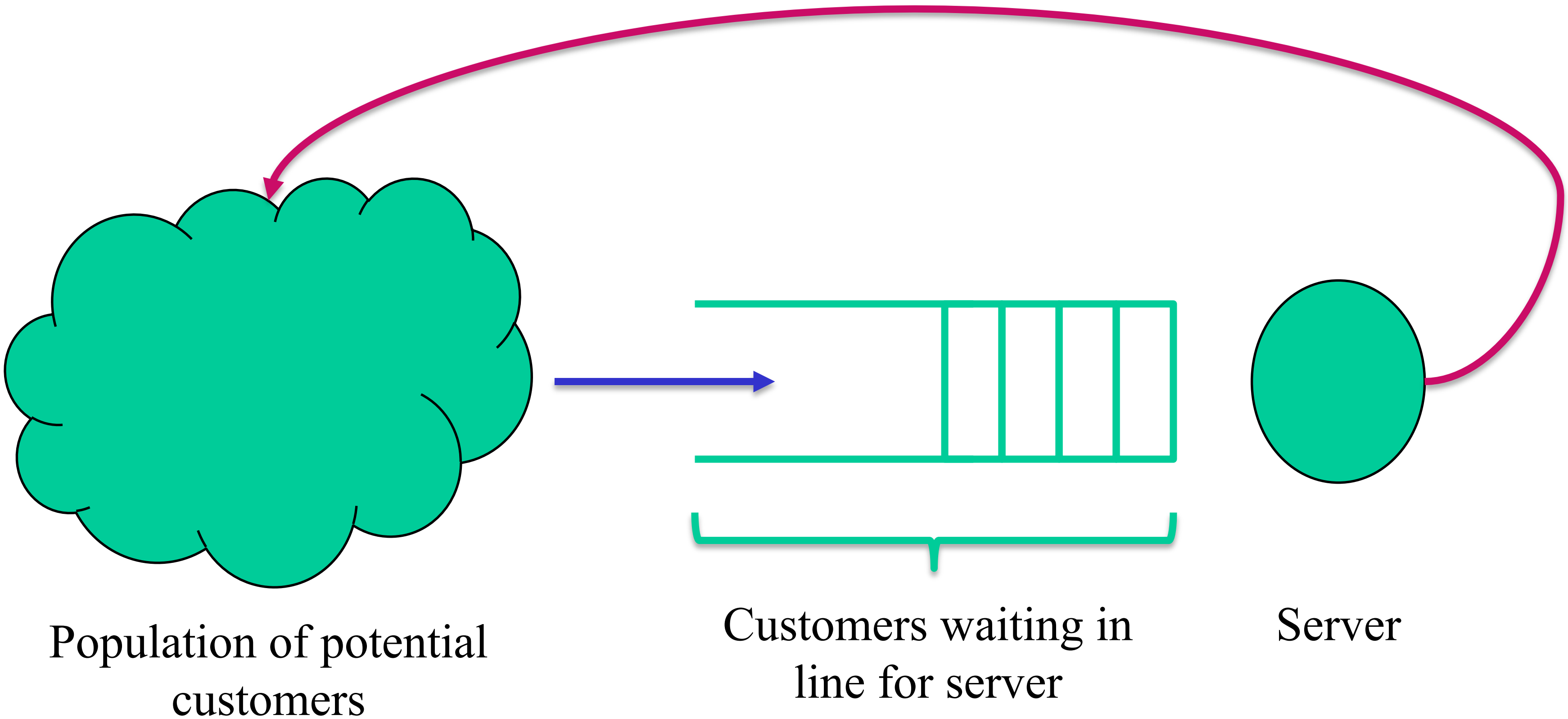
- We need tools to reason about queues so we can design them better (**allegedly**!)
  - Utilization
  - Number of jobs/customers in line
  - Waiting times

- Analytical solutions possible sometimes
- More complex models require simulation!

Objectives in this Course

- Introduce Queueing theory notation
- Understand Little's theorem
  - It's really not of little impact!
- Obtain closed form solutions for simple models
  - M/M/1 queues
  - M/M/c queues
  - M/M/m/B queues
- Insights into M/G/1 queues and some comparisons

Typical Queuing Models



Key Elements

- Customer/Job:
  - Anything that arrives at the system and requires service
  - Example: people, machines, trucks, packets, etc.

- Server:
  - Any resource that provides the requested service
  - Example: repairpersons, airport runways, router, web server, etc.

- Population: the population of potential customers
  - Can be finite: arrival rate will depend on the state of the queue
    - Example: arrivals stop when customers see a very long line
  - Can be considered to be infinite
    - The arrival rate will be independent of the state of the queue

System Capacity

- **Maximum** number of customers/jobs that can be waiting in the queue

- Can be limited:
  - e.g., network switches, routers
    - The number of packets waiting to be processed depends on the buffer size (limited by switch's memory)
  - e.g., TCP SYN backlog
    - Number of half-open TCP connections
    - Drop new connections when backlog is full (check out SYN flood attacks)

- Can be unlimited:
  - DMV: Yes, I would like to wait outside when it's 19^o
  - Teenagers waiting in the streets to spend their parents' hard-earned money on the new iPhone

Arrival Process

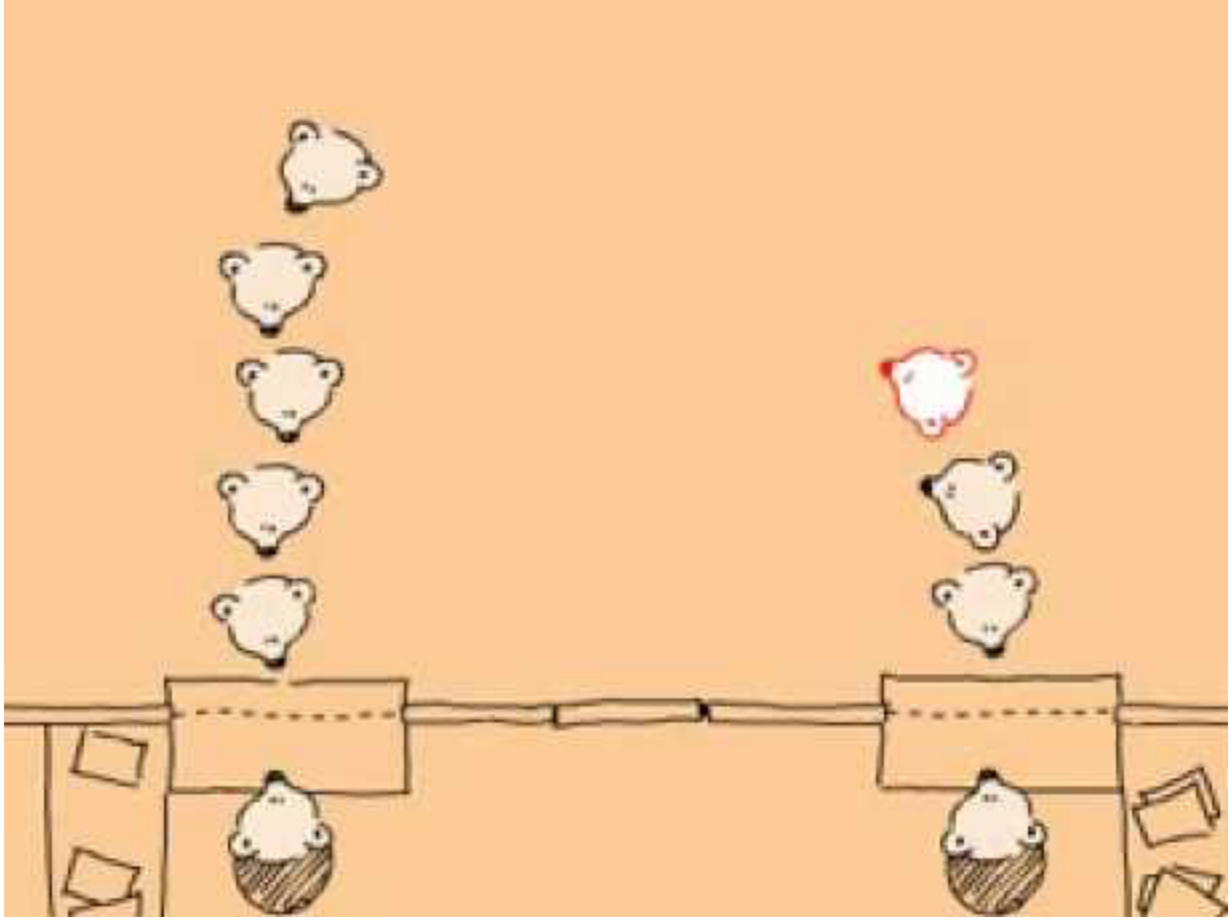
- For an **infinite population**, we can model the arrivals of customers or jobs at the system

- Random arrivals
  - For example, Poisson arrival process with rate λ
  - Inter-arrival times of customer are then i.i.d. exponentials with rate λ

- Scheduled arrivals:
  - Planes arriving at the runway (under ideal situations)
    - Definitely not at O'Hare.
  - By appointment

- At least one customer is always present
  - Sufficient raw material for an industrial machine

Queue Behavior



Queue Behavior & Discipline

- **Queue Behavior**
  - Actions of customers while in a queue waiting for service
    - Balk: leave when they see that the line is too long
      - Drive with an expired license
    - Renege: leave after being in the line when it's moving too slowly
    - Jockey: See previous slide

- **Queue discipline**:
  - Determination of which customer is chosen for service when server is free
    - First-in-first-out (FIFO)
    - Last-in-first-out (LIFO)
    - Service in random order (SIRO)
    - Shortest processing time first (SPT)
    - Service according to priority (PR)

Service Time & Mechanism

- We denote by S_i the service time of the $i'th$ arrival to the queueing system
  - Generally a random variable, but can also be constant
  - $\{S_i,\ i \geq 0\}$ are usually assumed to be i.i.d., random variables
    - Exponentials for the largest part of this class

- A system will generally consist of several servers along with one or more interconnected queues
  - There can be m servers that are working in parallel
  - A system is work conserving if a server is never idle while a customer arrives or is in the queue
    - i.e., agent will not call her friend to gossip while you're waiting at the door for her to sign your paycheck (true story)
      - At least I got to listen to the full story

Notation

- We will be using **Kendall's notation** for parallel server queues

$$A/B/m/B/K/SD$$

Interarrival time distribution

Server time distribution

Number of servers

System capacity

Population size

Queue Discipline

Examples

- M/M/1
  - Exponential interarrivals
  - Exponential service times
  - 1 server
  - Infinite population
  - Infinite buffer size
  - First in first out
- M/G/c/B
  - Exponential interarrivals
  - Exponential service times
  - c server
  - Finite buffer of size B

Implicit or "defaults" in case not specified

More Notation

- π_n, P_n: steady-state probability of having n customers in the system
- $P_n(t)$: probability of there being n customers in the system at time t
- λ: arrival rate
- μ: service rate of one server
- ρ: server utilization

- S_n: service time of $n'th$ arriving customer
- W_n: total time spent in the system by $n'th$ arriving customer
- W_n^Q: total time spent in the queue by customer n

- L: long-run time-average number of customers in the queue
- L_Q: long-run time-average number of customers in the queue
- W: long-run average time spent in the system **per customer**
- W_Q: long-run average time spent in the queue **per customer**

Little's Law

- Not a little result: part of the queueing fold literature for the past century
- Formal proof due to J.D.C. Little in 1961

$$L = \lambda W$$

Average number of customers in the system

Arrival rate

Average time spent in the system per customer

Little's Law

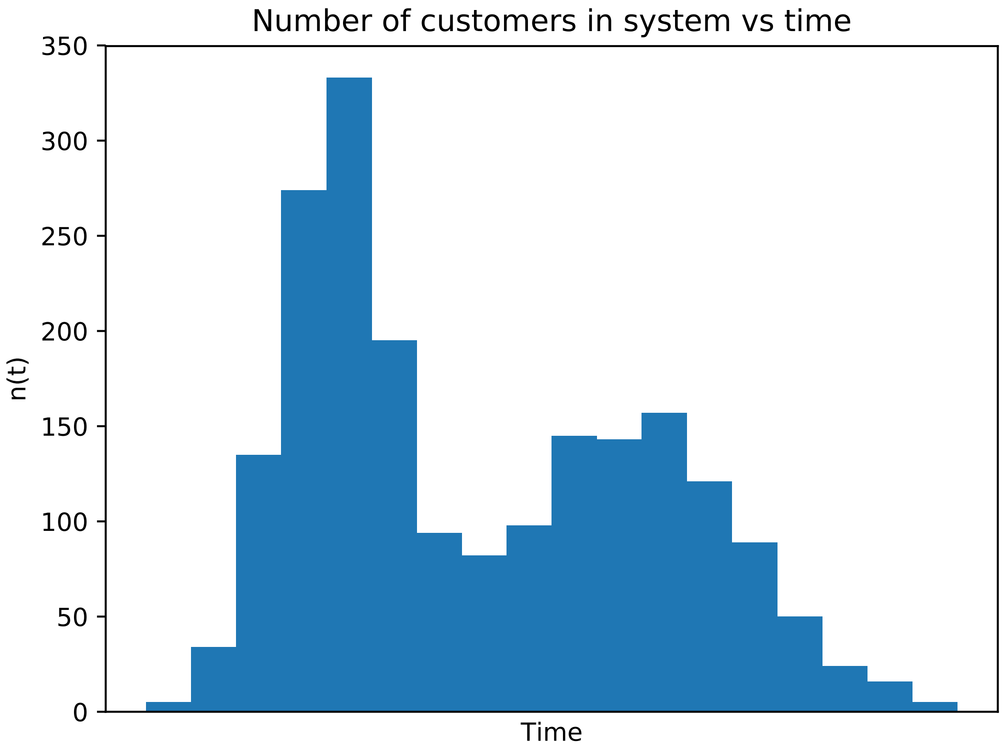
- (Average number in system) = (arrival rate) x (average time in system)
- $L = \lambda W$

- **Notice that we did not make any assumptions about the system**
  - No assumptions about arrival process
  - No assumptions about number of servers
  - No assumptions about queue discipline

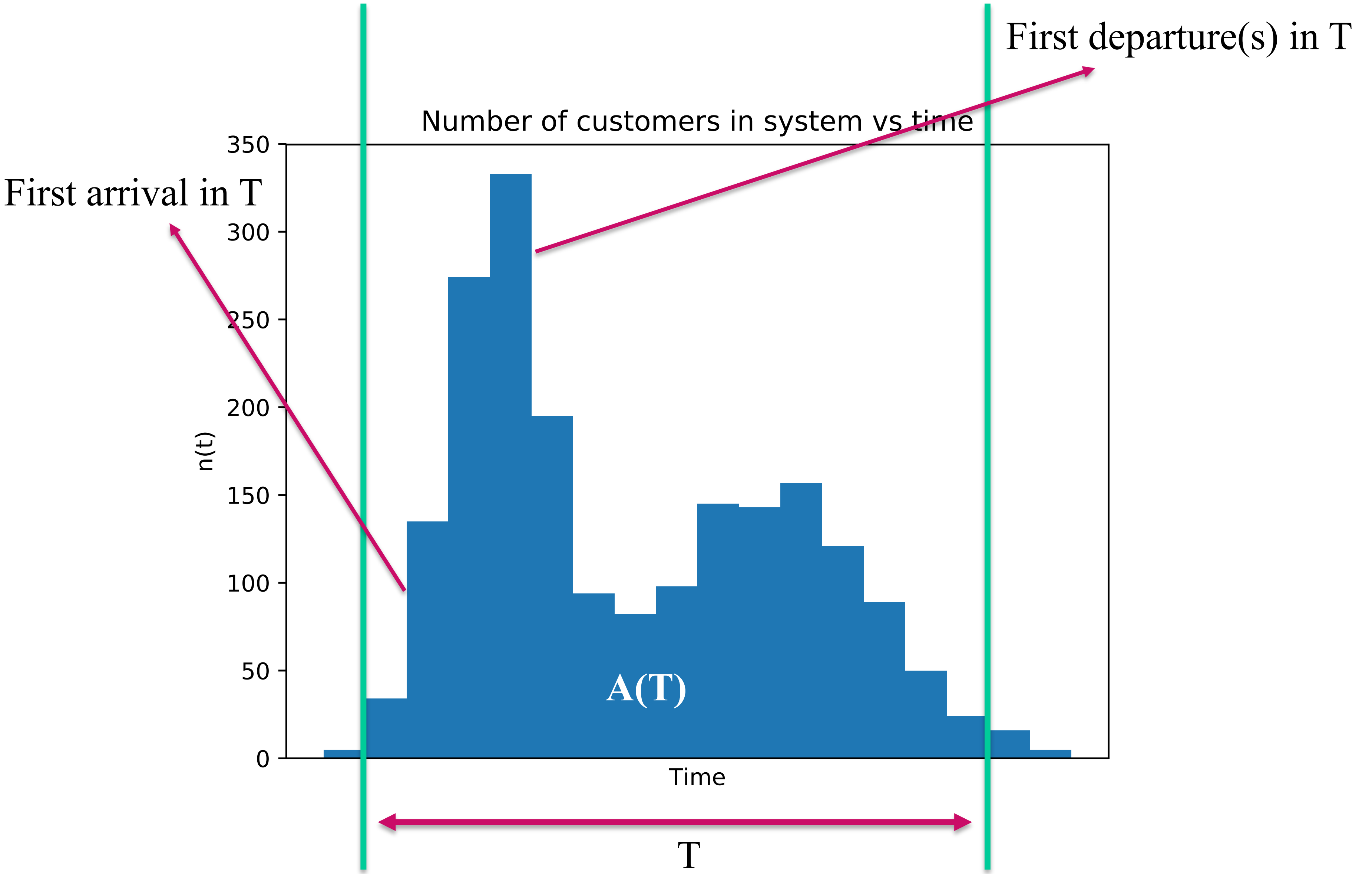
- Little's law applies to any "black box" queue assuming:
  - The system is work conserving
  - The system is stable, i.e., can reach a steady state
    - Arriving customer will eventually leave
    - Exit rate is equal to the arrival rate

Heuristic Proof

- Let $n(t)$ be the number of customers in the system up to time t
- Let T be a long period of time
- Let $A(T)$ be the area under the curve $n(t)$ over the time period T
- Let $N(T)$ be the number of arrivals in the time period T



Heuristic Proof



Heuristic Proof

- Average value of $n(t)$ over T is its integral over T divided by T, i.e.,

$$L(T) = \frac{A(T)}{T}$$

- At each time instant t, each customer of $n(t)$ is accumulating wait time, so we can obtain the average cumulative waiting time as $A(T)$, so

$$W(T) = \frac{A(T)}{N(T)}$$

- Also the arrival are countable over T, so we can estimate their rate as

$$\lambda(T) = \frac{N(T)}{T}$$

- By a slight manipulation, we can get that

$$L(T) = \lambda(T)W(T)$$

- In steady state as we send T to infinity, assuming quantities converge, we get

$$L = \lambda W$$