Basic Queueing Theory
M/M/* Queues

These slides are created by Dr. Yih Huang of George Mason University. Students registered in Dr. Huang's courses at GMU can make a single machine-readable copy and print a single copy of each slide for their own reference, so long as each slide contains the copyright statement, and GMU facilities are not used to produce paper copies. Permission for any other use, either in machine-readable or printed form, must be obtained from the author in writing.

Introduction

- Queueing theory provides a mathematical basis for understanding and predicting the behavior of communication networks.
- Basic Model

![Diagram of a queueing model]
- Major parameters:
  - interarrival-time distribution
  - service-time distribution
  - number of servers
  - queueing discipline (how customers are taken from the queue, for example, FCFS)
  - number of buffers, which customers use to wait for service
- A common notation: $A/B/m$, where $m$ is the number of servers and $A$ and $B$ are chosen from
  - $M$: Markov (exponential distribution)
  - $D$: Deterministic
  - $G$: General (arbitrary distribution)

M/M/1 Queueing Systems

- Interarrival times are exponentially distributed, with average arrival rate $\lambda$.
- Service times are exponentially distributed, with average service rate $\mu$.
- There is only one server.
- The buffer is assumed to be infinite.
- The queuing discipline is first-come-first-serve (FCFS).
System State

- Due to the memoryless property of the exponential distribution, the entire state of the system, as far as the concern of probabilistic analysis, can be summarized by the number of customers in the system, *i*.
  - the past/history (how we get here) does not matter
- When a customer arrives or departs, the system moves to an adjacent state (either *i*+1 or *i*-1).

In equilibrium,

- Let \( P_i = P\{\text{system in state } i}\)  
- We have \( \lambda P_i = \mu P_{i+1}\)

The rate of movements in both directions should be equal
Equations from the state transition diagram:

\[ \lambda P_0 = \mu P_1 \]
\[ \lambda P_1 = \mu P_2 \]
\[ \lambda P_2 = \mu P_3 \]
\[
\vdots
\]
\[ P_1 = \frac{\lambda}{\mu} P_0 = \rho P_0 \]
\[ P_2 = \frac{\lambda}{\mu} P_1 = \rho^2 P_0 \]
\[
\vdots
\]
\[ P_k = \rho^k P_0 \]

What is \( \rho \) ?

Since

\[
\sum_{k=0}^{\infty} P_k = \sum_{k=0}^{\infty} \rho^k P_0 = 1
\]

we have

\[
\frac{1}{1-\rho} P_0 = 1 \Rightarrow P_0 = 1 - \rho.
\]

That is, \( P_k = \rho^k (1 - \rho) \)

Note that \( \rho \) must be less than 1, or else the system is unstable.
Average Number of Customers

\[ N = \sum_{k=0}^{\infty} kP_k = (1 - \rho) \sum_{k=0}^{\infty} k \rho^k = ? \]

- Average delay per customer (time in queue plus service time):

\[ T = \frac{N}{\lambda} = \frac{1}{\mu - \lambda} \]

- Average waiting time in queue:

\[ W = \frac{1}{\mu - \lambda} - \frac{1}{\mu} = \frac{\rho}{\mu - \lambda} \]

- Average number of customers in queue:

\[ N_Q = \lambda W = \frac{\rho^2}{1 - \rho} \]
Applications

- Consider 24 computer users, each of which produces in average 48 packets per second.
- For every customer, the interarrival times of his packets are exponentially distributed.
- The lengths of packets are also exponentially distributed, with mean 125 bytes.

Scenario 1

- Users share a T1 line using the standard T1 time-division multiplexing.
- Assume that each user is associated with an infinite buffer (that is, queue).
- In a T1 line, it takes 1/8000 seconds to deliver (or serve) each byte.
- However, due to their variable lengths, the delivery (or service) times of packets are still exponentially distributed.
  - The average service rate $\mu = ?$
The system can be considered as 24 M/M/1 queues:

\[ \rho = \frac{48}{64} = 75\% \]

\[ N = \frac{0.75}{1 - 0.75} = 3 \]

\[ T = \frac{1}{64 - 48} = \frac{1}{24} \approx 42\text{msec} \]

Scenario 2

Users share a 1.544 Mbps line through an IP router.
- The (aggregated) arrival rate is $24 \times 48 = 1152$.  
The service rate is $24 \times 64 = 1536$.  
We have

$$\rho = \frac{48}{64} = 75\%$$

$$T = \frac{1}{1536 - 1152} = \frac{1}{384} \approx 2.4 \text{ msec}$$

This system is 24 times faster than TDM!

---

**Discussion**

- Flaws in the analysis?

- Still such a drastic difference in results convincingly reveals the inefficiency of TDM.

- This partly explains the momentum toward using the Internet as the universal information infrastructure.

- In general, allowing customers to share a pool of resources is far more efficient than allocating a fixed portion to each customer.
M/M/m Queueing Systems

All servers are identical, with service rate $\mu$

State Transition Diagram

Balance equations:

$$\lambda P_{i-1} = \begin{cases} 
  i \mu P_i, & \text{for } i \leq m \\
  m \mu P_i, & \text{for } i > m 
\end{cases}$$
Solution

\[ P_i = \begin{cases} 
  P_0 \frac{(m\rho)^i}{i!}, & \text{for } i \leq m \\
  P_0 \frac{m^m \rho^i}{m!}, & \text{for } i > m 
\end{cases} \]

Where \( \rho = \frac{\lambda}{m\mu} \).

Noticing that \( \sum_{i=0}^{\infty} P_i = 1 \), we have

\[ P_0 = \left[ \sum_{i=0}^{m-1} \frac{(m\rho)^i}{i!} + \frac{(m\rho)^m}{m!(1-\rho)} \right]^{-1} \]

The probability that an arriving customer has to wait in queue:

\[ P_Q = \sum_{i=m}^{\infty} P_i \]

\[ = \sum_{i=m}^{\infty} P_0 \frac{m^m \rho^i}{m!} \]

\[ = P_0 \frac{(m\rho)^m}{m!} \sum_{i=m}^{\infty} \rho^{i-m} \]

\[ = P_0 \frac{(m\rho)^m}{m!(1-\rho)} \]

This is known as the \textit{Erlang C formula}. 
Average number of waiting customers:

\[ N_Q = \sum_{i=m}^{\infty} (i-m) P_i = \sum_{i=0}^{\infty} i P_{i+m} \]

\[ = \sum_{i=0}^{\infty} i P_0 \frac{m^i \rho^i}{i!} = \frac{P_0 (m \rho)^m}{m!} \sum_{i=0}^{\infty} \frac{\rho^i}{i!} \]

\[ = \frac{P_0 (m \rho)^m}{m!} \frac{\rho}{(1-\rho)^2} \]

\[ = P_Q \frac{\rho}{1-\rho} \]

Average waiting time in queue:

\[ W = \frac{N_Q}{\lambda} = \frac{\rho P_Q}{\lambda (1-\rho)} \]

Average time in the system:

\[ T = \frac{1}{\mu} + W = \frac{1}{\mu} + \frac{\rho P_Q}{\lambda (1-\rho)} = \frac{1}{\mu} + \frac{P_Q}{m \mu - \lambda} \]

Average number of customers in the system:

\[ N = \lambda T = \frac{\lambda}{\mu} + \frac{\lambda P_Q}{m \mu - \lambda} = m \rho + \frac{\rho P_Q}{1-\rho} \]
M/M/1/K Queueing Systems

- Similar to M/M/1, except that the queue has a finite capacity of $K$ slots.
- That is, there can be at most $K$ customers in the system.
- If a customer arrives when the queue is full, he/she is discarded (leaves the system and will not return).

Analysis

Notice its similarity to M/M/1, except that there are no states greater than $K$. We have

$$P_i = \begin{cases} 
\rho^i P_0, & \text{for } 0 \leq i \leq K \\
0, & \text{for } i > K 
\end{cases}$$

Noticing that $\sum_{i=0}^{K} P_i = 1$ we have

$$P_0 = \frac{1 - \rho}{1 - \rho^{K+1}}$$
**Poisson Process**

Let random variable $N$ be a “counter” of the number of occurrences of a particular type of events. Clearly, the value of $N$ increases over time. Let $N(t)$ be the value of $N$ at time $t$. Moreover, if $N(0) = 0$, $N(t)$ is said to be a counting process.

The counting process $N(t)$ is said to be a **Poisson process** having rate $\lambda$ if the number of events in any interval of length $t$ is Poisson distributed with mean $\lambda t$. That is, for all $s, t \geq 0$

$$P\{N(t+s) - N(s) = n\} = e^{\lambda t} \frac{(\lambda t)^n}{n!}, \quad n = 0, 1, ...$$

**Discussion**

- The interarrival times of a Poisson process with rate $\lambda$ is exponentially distributed with average $1/\lambda$.
- The reverse is also true: if the interarrival times of events are exponentially distributed with average $1/\lambda$ then the event counting process is Poisson with rate $\lambda$.
- Thus, the customer arrival processes of $M/M/*$ queueing systems are Poisson.
- A Poisson customer count and exponentially distributed customer interarrival times are the two sides of the coin.
Sampling Poisson Arrivals

- Consider a Poisson customer arrival process of with average rate $\lambda$.
- Each customer can be classified as Type I or Type II, with probability $p$ and $1-p$ respectively.
- Then, the arrival process of Type I customers is also Poisson with average rate $p\lambda$.
- Likewise, the arrivals of Type II customers is Poisson with average rate $(1-p)\lambda$.

Application

- We know that the customer arrivals at a barbershop form Poisson process with average rate of 10 customers per hour.
- Among the customers, 40% are males and 60% are females.
- Then the interarrival times of male customers are exponentially distributed with an average rate of 4 per hour.
- The interarrival times of female customers are exponentially distributed with an average rate of 6 per hour.
Exercise

- Consider the router configuration below.

- The lengths of arriving packets are exponentially distributed with an average of 1000 bits.

Questions

- Argue that queues A and B are independent M/M/1 systems.

- Compute the average length of queue A in bits.

- For a packet destined for port 2, compute its expected time at the router (including transmission time).
Compute the average time a packet spent at the router (including transmission time).

Compute the average number of packets at the router (including the ones in transmission).

Merging Poisson Arrivals

- Given two exponential variables $X_1$ and $X_2$, with rates $\lambda_1$ and $\lambda_2$, the random variable $X = \min\{X_1, X_2\}$ is also exponential, with rate $\lambda_1 + \lambda_2$.
- Consider two Poisson arrivals, with average rates $\lambda_1$ and $\lambda_2$.
- The merged arrival process will also be a Poisson process, with the average rate $\lambda_1 + \lambda_2$.
Consider the router configuration below.

- The lengths of arriving packets are exponentially distributed with an average of 1000 bits.
  - Why do we care about packet lengths?
- Packet arrivals at ports 1 and 2 are exponentially distributed with average rates of 2000 and 3000 packets per second, respectively.
- The transmission rate of port 0 is 10Mbps.
- The whole system can be modeled as a single M/M/1 queueing system, with an arrival rate of 5000 and service rate of 10,000.
**Burke's Theorem**

In its steady state, an M/M/m queueing system with arrival rate $\lambda$ and per-server service rate $\mu$ produces exponentially distributed inter-departure times with average rate $\mu$. Application: Two cascaded, independently operating M/M/m systems can be analyzed separately.

![Diagram of M/M/1 queueing system with two servers](image)

**Pitfall**

- Consider the system below where the servers are transmission lines.

- Packets lengths are exponentially distributed with an average of 1000 bits.
- Can the two queues be analyzed separately? Why?
Discussion

In general: any “feedforward” network of independently-operating M/M/m systems can be analyzed in this system-by-system decomposition.

Question: How about networks that do contain “feedbacks”?

Answer: the interarrival times of some systems may not be exponentially distributed and thus cannot be analyzed as independent M/M/m queues
**Jackson's Theorem**

For an *arbitrary* network of $k$ M/M/1 queueing systems,

$$P(n_1, n_2, ..., n_k) = P_1(n_1) P_2(n_2) ... P_k(n_k),$$

where

$$P_j(n_j) = \rho_j^{n_j} / (1 - \rho_j).$$

That is, in terms of the number of customers in each system, individual systems act as if they are independent M/M/1 queues (they may not).

**Application**

Consider the network below. The arrival rate $\lambda$ and probabilities $p_1$ and $p_2$ are known.

We first compute the arrival rates $\lambda_1$ and $\lambda_2$:

$$\lambda_1 = \lambda + \lambda_2, \quad \lambda_2 = p_2 \lambda_1$$

$$\Rightarrow \lambda_1 = \lambda / p_1, \quad \lambda_2 = \lambda p_2 / p_1$$
Let $\rho_1 = \lambda_1 / \mu_1$ and $\rho_2 = \lambda_2 / \mu_2$. By Jackson's theorem,

$$P(i, j) = \rho_1^i (1 - \rho_1) \rho_2^j (1 - \rho_2)$$

And

$$N_1 = \frac{\rho_1}{1 - \rho_1}, \quad N_2 = \frac{\rho_2}{1 - \rho_2}$$

Total number of customers in system is

$$N = N_1 + N_2.$$ 

Average time in system is

$$T = \frac{N}{\lambda} = \frac{\rho_1}{\lambda (1 - \rho_1)} + \frac{\rho_2}{\lambda (1 - \rho_2)}.$$ 

Discussion

Consider the packet switching network below.

Can we cite the Jackson's theorem, model the transmission lines as servers, and analyze them as separate M/M/1 queues? Why?