CS4226 Midterm cheatsheet

By Khoo Jing Hong,Derrick

Network Performance

Little's Law:

L = λ * W,

where

L = average # of customers/packets in the system

λ = Arrival rate

W = Average sojourn time (time spent in the queue)

Network Queueing

Exponential distribution:

Exponential Distribution ✱✱✱ Poisson

- A continuous r.v. T follows/has an **exponential distribution** with parameter $\lambda > 0$, if, for $x \geq 0$
  - $F(x) = P\{T \leq x\} = 1 - e^{-\lambda x}$ or $\bar{F}(x) = P\{T > x\} = e^{-\lambda x}$ complementary cdf
  - $f(x) = \frac{dF(x)}{dx} = \lambda e^{-\lambda x}$
- $E[T] = \int_{-\infty}^{\infty} x f(x)\, dx = \frac{1}{\lambda}$ (Average)
- What is the unit and physical meaning of λ? frequency, Hz; How fast packets come into system
- Why is exponential a good distribution?
  - Memoryless property: $P\{T > s + t \mid T > s\} = P\{T > t\}$ conditional probability; Rate of arrival

6

-> <u>Memoryless property</u> (Future only depends on current state, and does not take into account the state of the past)

-> Model the **<u>arrival of packets</u>** as a poisson process(exponentially distributed as T with rate λ)

M/M/1 model:

M/M/1 Model (Markovian)

- Single server with a queue of infinite size
- Poisson arrival with rate λ
- Exponential i.i.d. service time with rate μ
- Arrival and service times are independent
- First-In-First-Out (FIFO) service discipline

no priority

12

Utilization, ρ = λ / μ

-> System stability: λ < μ

-> percentage of time that the "server" is busy, or

-> probability a random observation finds server busy

Probability that random obv finds i packets in system

Main Result (without proof)

- Denote π_i as
  - Percentage of time that exactly i packets or customers in the system, i.e., server + queue
  - Also the probability $P\{L = i\}$ that a random observation finds i packets in the system
  - For M/M/1 system, we have Geometric distribution
    $$\pi_i = P\{L = i\} = \rho^i (1 - \rho)$$
    L: DRV
- As a sanity check
  - what is the value for π_0? $1-\rho$ ∴ ρ: probability that server is busy
  - what is the sum of all the π_is? 1
  - follows a geometric distribution

16

-> **Geometric Distribution** also has memoryless property like exponential

Relevant formulae:

Sum of finite geometric sequence:

$$S_n = \sum_{i=1}^{n} a_i r^{i-1} = a_1 \left(\frac{1 - r^n}{1 - r} \right)$$

Sum of infinite geometric series:

$$S = \sum_{i=0}^{\infty} a_i r^i = \frac{a_1}{1 - r}$$

[note: if |r|≥1, the infinite series does not have a sum]

Focusing on M/M/1 System(Server + Q)

- Average # of packets in the system
  $$E[L] = \frac{\rho}{1 - \rho}, \quad \rho = \frac{\lambda}{\mu} = \text{utilization}$$
- Average sojourn time of packets (Delay)
  $$E[W] = \frac{1}{\mu - \lambda}$$
  $E(L) = \lambda E(W)$ by Little's Law

Focus on the queue $= \frac{1}{\mu}\left(\frac{1}{1-\rho}\right)$

Focusing on the M/M/1 Queue only

- Average # of packets in the queue $E[Q] = \frac{\rho^2}{1-\rho}$
- Avg. queueing delay of packets $E[D] = E[W] - \frac{1}{\mu}$

-> For M/M/1 model, ∃ positive relationship between throughput and queuing delay.

<u>Assumption:</u> M/M/1 model models a fixed sending rate (Like UDP)

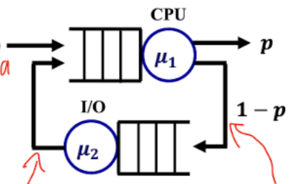
-> Throughput of stable M/M/1 system: λ

Burke's Theorem for Stable state M/M/1:
-> departure process is Poisson with rate λ
-> Effectively: **Arrival rate = Departure rate**

Solving Jackson Network
-> Calculate effective arrival rate for subsystem: Outside arrival rate + Feedback arrival rate.
-> Obtain a system of linear equations and resolve

A simple web server example

- What is the average number of jobs in the system? E[L]



Geometric formula

$$P\{L_i = x\} = \rho_i^x(1-\rho_i)$$

$$\Rightarrow E[L_i] = \frac{\rho_i}{1-\rho_i}$$

$$\Rightarrow E[L] = E[L_1] + E[L_2]$$

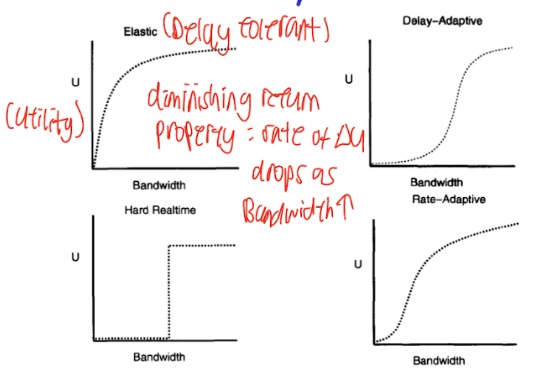
$$\begin{cases} \lambda_1 = \lambda + \lambda_2 \\ \lambda_2 = (1-p)\lambda_1 \end{cases} \Rightarrow \begin{cases} \lambda_1 = \frac{\lambda}{p} \\ \lambda_2 = \frac{(1-p)\lambda}{p} \end{cases} \Rightarrow \begin{cases} \rho_1 = \frac{\lambda}{p\mu_1} \\ \rho_2 = \frac{(1-p)\lambda}{p\mu_2} \end{cases}$$

30

Network Resource Allocation
Effective Bandwidth
-> More capacity of link relates to accommodating higher throughput and provides lower delay

Monotonically increasing property

How Does Utility Look Like?



Max-min fair allocation
-> Use water filling algorithm, start from small values and work way up
-> Idea is users of small demand get all they want, users of large demand evenly split the rest, until capacity of link.

-> Bottleneck resource: Resource is saturated, flow i has the maximum rate among all flows using resource r.

- flow cannot get more resource if allocation is fair; otherwise, it hurts flows with lower rates
-

Weighted Max-min fair allocation
-> Use local view method with constraints. Then use the smallest value in all local views and start from there until a link is saturated.
-> If demand is satisfied for a resource, split the remaining in accordance with ratio given

Bottleneck resource here means that X(i)/phi(i) is maximal, phi(i) here being the absolute value in the weighted vector, and the resource r is saturated.

Theorem: When each flow has an infinity demand under a network system, a flow allocation is max-min fair if and only if every flow has a bottleneck resource.