

Models

Optimization, Queues & Graphs

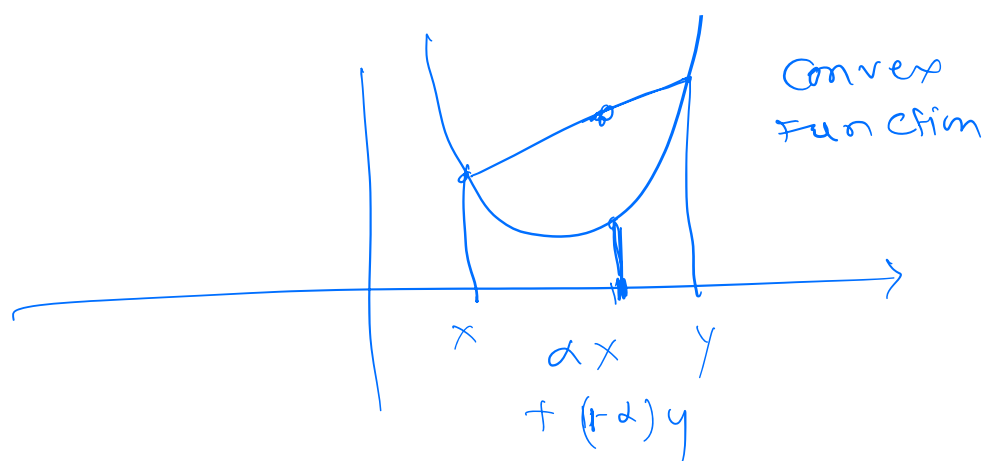
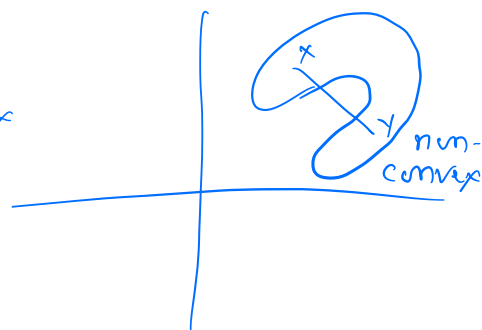
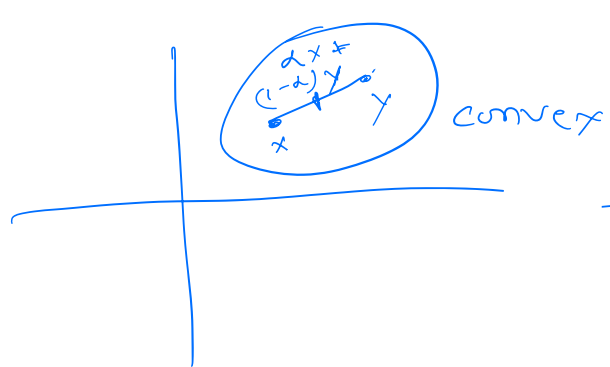
EECS 122, Spring 2024

Shyam Parekh

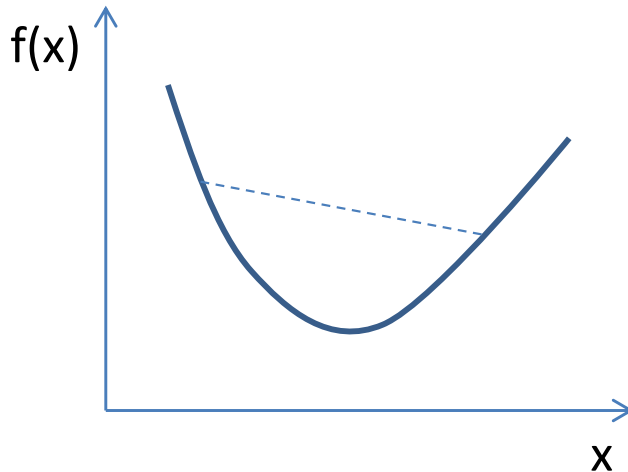
Optimization

Optimization: Basics

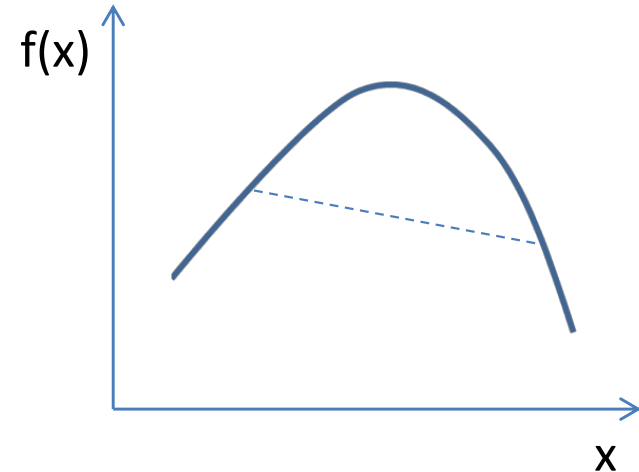
- **Convex Set:** A set $S \subseteq R^n$ is convex if $\alpha x + (1 - \alpha)y \in S$ whenever $x, y \in S$ and $\alpha \in [0, 1]$.
- **Convex Function:** A function $f(x): S \subseteq R^n \rightarrow R$ is a convex function if S is a convex set and the following condition holds for any $x, y \in S$ and $\alpha \in [0, 1]$:
$$f(\alpha x + (1 - \alpha)y) \leq \alpha f(x) + (1 - \alpha)f(y).$$
 $f(x)$ is strictly convex if the inequality is strict for all $\alpha \in (0, 1)$ and $x \neq y$.
- **Concave Function:** A function $f(x): S \subseteq R^n \rightarrow R$ is a concave (strictly concave) function if $-f$ is a convex (strictly convex) function.



Optimization: Basics (2)



Convex Function



Concave Function

Theorem: If $f(x)$ is convex (concave) and differentiable, then x^* is a global minimizer (maximizer) if $\nabla f(x^*) := (\frac{\partial f}{\partial x_1}(x^*), \frac{\partial f}{\partial x_2}(x^*), \dots, \frac{\partial f}{\partial x_n}(x^*)) = 0$

Constrained Optimization

- **Problem:** $\max_{x \in S} f(x)$ subject to

$$h_i(x) \leq 0, i = 1, 2, \dots, I,$$

$$g_j(x) = 0, j = 1, 2, \dots, J.$$

- This is referred to as the **primal** problem.
- **Lagrangian** for the above problem is defined as

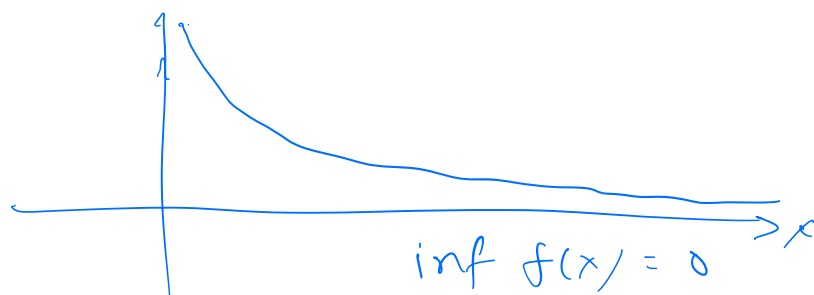
$$\rightarrow L(x, \lambda, \mu) = f(x) - \underbrace{\sum_{i=1}^I \lambda_i h_i(x)}_{\text{penalty}} + \underbrace{\sum_{j=1}^J \mu_j g_j(x)}_{\text{penalty}}, \lambda_i \geq 0 \forall i. \leftarrow$$

- **Lagrangian Dual Function** is defined to be

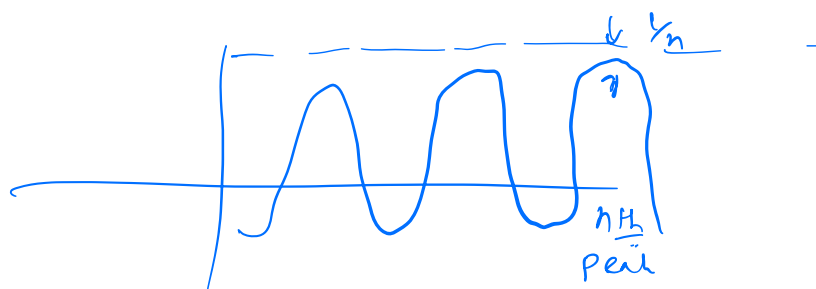
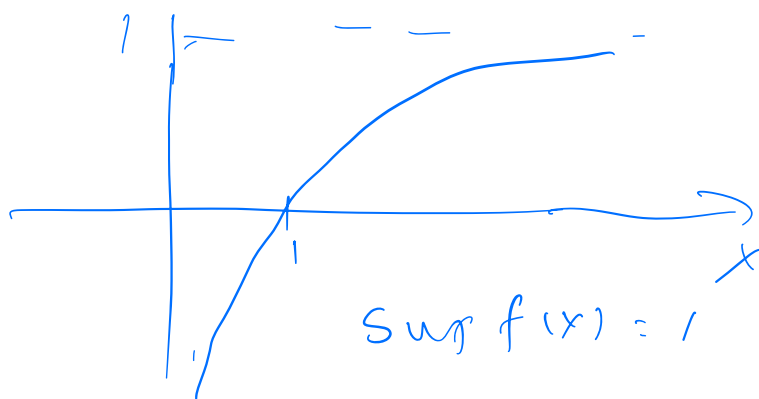
$$\underline{D}(\lambda, \mu) = \underline{\sup}_{x \in S} \underline{L}(x, \lambda, \mu). \quad \text{?}$$

- **Theorem:** $D(\lambda, \mu)$ is a convex function, and $D(\lambda, \mu) \geq f^*$, where $f^* = \max_{x \in S} f(x)$ subject to the constraints above.

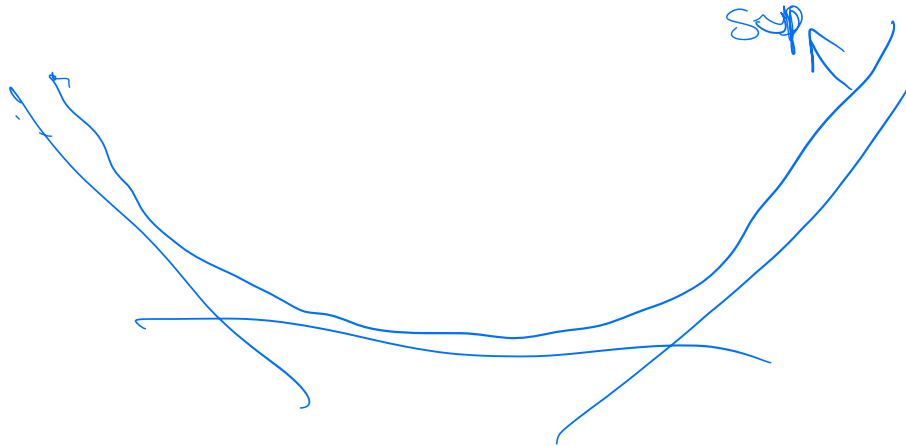
$$f(x) = \frac{1}{x}, \quad x > 0$$




$$f(x) = 1 - \frac{1}{x}, \quad x > 0$$



\sup = Lowest Upper Bound
 \inf = Highest Lower Bound.

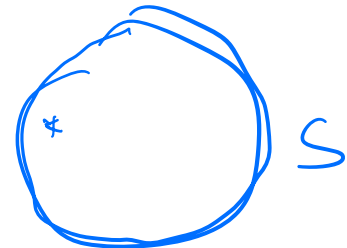


Constrained Optimization (2)

- Theorem above states that the dual function is an upper bound on the maximum of our optimization problem.
- We can get the best upper bound by solving the **Lagrange Dual Problem**: $\inf_{\lambda \geq 0, \mu} D(\lambda, \mu)$. 
 - Sometimes, the dual problem is easier to solve than the primal problem.
- Let $d^* = \inf_{\lambda \geq 0, \mu} D(\lambda, \mu)$. In general $d^* \geq f^*$. $d^* - f^*$ is called **duality gap**. We say that **strong duality** holds if $d^* = f^*$.

Constrained Optimization (3)

- **Theorem (Slater Conditions):** Strong Duality holds if
 - $f(x)$ is a concave function and $h_i(x)$ are convex functions;
 - $g_j(x)$ are affine functions (i.e., of the form $ax + b$); and
 - There exists an x that belongs to the relative interior of S such that $\underline{h_i(x)} < 0$ for all i and $g_j(x) = 0$ for all j .



Constrained Optimization (4)

Theorem (Karush – Kuhn – Tucker (KKT) Conditions): Assume that $f, h_i, i = 1, 2, \dots, I$, and $g_j, j = 1, 2, \dots, J$ are differentiable functions, and that Slater Conditions are satisfied. Let x^* be a point such that satisfies all the constraints. Such an x^* is a global maximizer for our optimization problem if and only if there exist constants $\lambda_i^* \geq 0$ and μ_j^* such that

$$\begin{aligned} \Rightarrow \frac{\partial f}{\partial x_k}(x^*) - \sum_i \lambda_i^* \frac{\partial h_i}{\partial x_k}(x^*) + \sum_j \mu_j^* \frac{\partial g_j}{\partial x_k}(x^*) &= 0, \forall k, \\ \lambda_i^* h_i(x^*) &= 0, \forall i. \end{aligned}$$

KKT-I
KKT-II

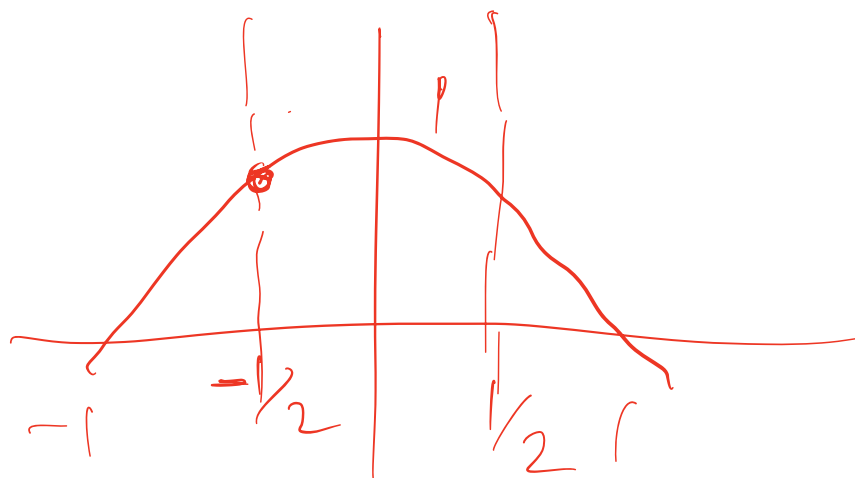
Furthermore, (λ^*, μ^*) is a global minimizer of the Lagrange dual problem if and only if the above two equations are satisfied.

Note: The second equation is referred to as the complementary slackness condition.

→ sup $L(x, \lambda, \mu)$

$$\frac{\partial L(x, \lambda, \mu)}{\partial x_i} = 0$$

$$f(x) = 1 - x^2$$



$$x + \frac{1}{2} \leq 0$$

$$h(x) = \begin{cases} x \leq -\frac{1}{2} \end{cases}$$

$$L(x, \lambda) = 1 - \underbrace{x^2}_{f(x)} - \lambda \left(\underbrace{x + \frac{1}{2}}_{h(x)} \right)$$

$$\frac{\partial L}{\partial x} = -2x - \lambda = 0 \quad \left. \begin{array}{l} \Rightarrow x = -\lambda/2 \end{array} \right\}$$

$$D(\lambda) = 1 - \frac{\lambda^2}{4} - \lambda \left(-\frac{\lambda}{2} + \frac{1}{2} \right)$$

$$= 1 + \frac{\lambda^2}{4} - \frac{\lambda}{2}$$

$$\frac{\partial D(\lambda)}{\partial \lambda} = \frac{\lambda}{2} - \frac{1}{2} = 0 \quad \}$$

$$\Rightarrow \lambda = 1$$

$$\Rightarrow x = -\frac{1}{2}$$

Resource Allocation and Utility Maximization

- The goal of resource allocation is to solve the Network Utility Maximization (NUM) problem:

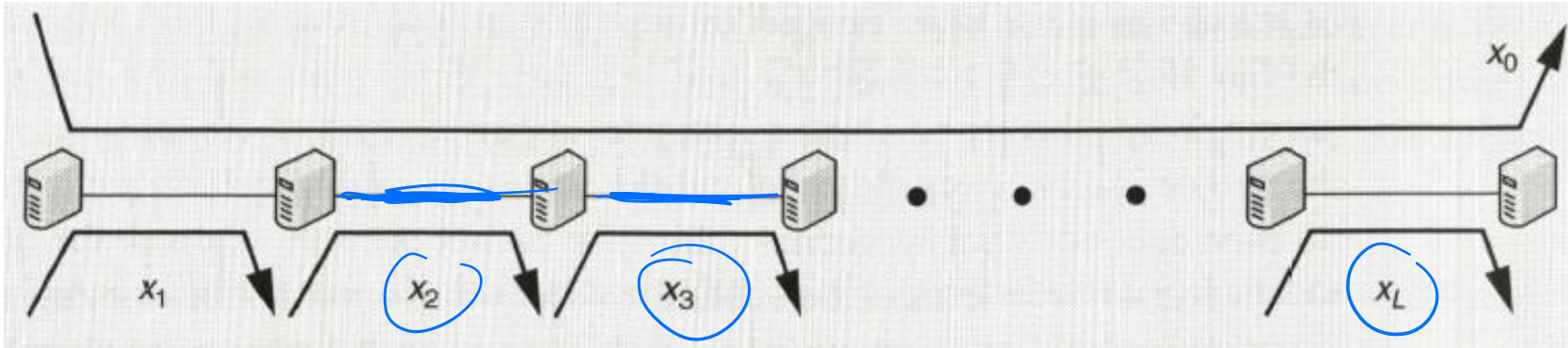
$$\max_{x_r} \sum_r U_r(x_r)$$

subject to the link capacity constraints

$$\sum_{r:l \in r} x_r \leq c_l, \forall l, \\ x_r \geq 0, \forall r.$$

- We assume the utility function to be increasing, continuously differentiable, and strictly concave.

Resource Allocation and Utility Maximization: Example



- Let $\mathbf{x} = (x_0, x_1, \dots, x_L)$, where x_l denotes rate for the connection l .
- Assuming log utility functions and link capacities of 1, we want to

$$\max_{\mathbf{x}} \sum_{r=0}^L \log x_r \quad f(\mathbf{x}) \leftarrow$$

subject to $x_0 + x_l \leq 1$, for $l = 1, 2, \dots, L$, and $\mathbf{x} \geq 0$.

(Unless specified, we assume \log to the base e , i.e., $\log = \ln$.)

- Here, the Lagrangian is given by $L(\mathbf{x}, \mathbf{p}) = \sum_{l=0}^L \log x_l - \sum_{l=1}^L p_l (x_0 + x_l - 1)$, where p_l 's denote the Lagrange multipliers.

$$\underbrace{\sum_{l=0}^L \log x_l}_{\lambda_l' s} - \underbrace{\sum_{l=1}^L p_l (x_0 + x_l - 1)}_{h_l(\mathbf{x})}$$

$$L(x, p) = \sum_{i=0}^L \ln(x_i) - \sum_{i=1}^L p_i (x_0 + x_i - 1)$$

$$\frac{\partial L}{\partial x_i} = 0 \quad i=0, \dots, L$$

$$\frac{\partial L}{\partial x_0} = \frac{1}{x_0} - \sum_{i=1}^L p_i = 0$$

$$\Rightarrow x_0^* = \sum_{i=1}^L p_i$$

$$\frac{\partial L}{\partial x_i} = \frac{1}{x_i} - p_i \quad i=1, \dots, L$$

$$= 0$$

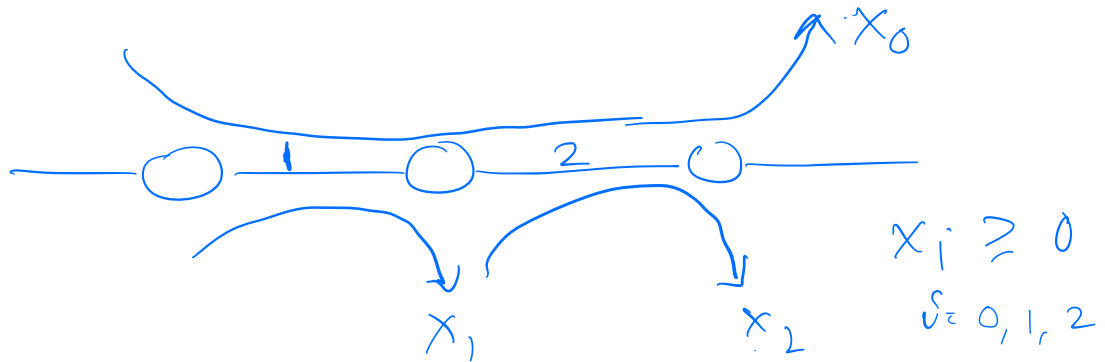
$$\Rightarrow x_i^* = \frac{1}{p_i^*} \quad i=1, \dots, L$$

Resource Allocation and Utility Maximization: Example (2)

- For the first KKT condition, we set $\frac{\partial L}{\partial x_r} = 0$ for each r . This gives

$$x_0^* = \frac{1}{\sum_{l=1}^L p_l^*}, x_l^* = \frac{1}{p_l^*}, \forall l \geq 1.$$

$$x_0^* = \frac{1}{\sum_{l=1}^L p_l^*} \quad x_l^* = \frac{1}{p_l^*} \quad l \geq 1$$
- The additional KKT conditions also require that
 $\Rightarrow p_l^* (x_0^* + x_l^* - 1) = 0$ and $p_l^* \geq 0, \forall l \geq 1.$
- Solving these equations, we get $p_l^* = \frac{L+1}{L}, \forall l \geq 1.$ \Leftarrow
- Hence the optimal data rates are $x_0^* = \frac{1}{L+1}$ and $x_l^* = \frac{L}{L+1}, \forall l \geq 1.$
- Observe that the optimal rate for a flow is the reciprocal of the sum of the Lagrange multipliers on its route.
 - If Lagrange Multipliers can be estimated “online” and fed back to each source, each source can estimate its optimal rate.



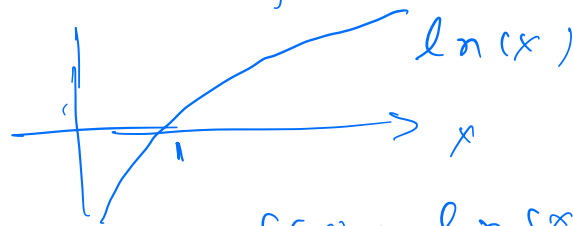
$$\max \sum_{i=0}^2 \ln(x_i) \quad \xrightarrow{f(\bar{x})}$$

$$x_0 + x_1 - 1 \leq 0$$

$$x_0 + x_2 - 2 \leq 0$$

$$\left(\begin{array}{l} h_1(\bar{x}) \leq 0 \\ h_2(\bar{x}) \leq 0 \end{array} \right)$$

① Concave Obj function



$$f(x) = \ln(x)$$

$$f'(x) = \frac{1}{x}$$

$$f''(x) = -\frac{1}{x^2}$$

② $h_i(\bar{x})$ convex

since $h_i(\bar{x})$ are linear

③ Boundary related cond. is met

\Rightarrow Slater Conditions are satisfied.

$$L(\bar{x}, \bar{\lambda}) = \sum_{i=0}^2 \ln(x_i) - \lambda_1(x_0 + x_1 - 1) - \lambda_2(x_0 + x_2 - 2)$$

$$\lambda_1(x_0 + x_1 - 1) = 0$$

$$\lambda_2(x_0 + x_2 - 2) = 0$$

$$\frac{\partial L(\bar{x}, \bar{\lambda})}{\partial x_i} = 0, \quad i = 0, 1, 2$$

①

$\bar{x}^*, \bar{\lambda}^*$ satisfying ① is our solution due to KKT

$$\frac{\partial L}{\partial x_0} = \frac{1}{x_0} - (\lambda_1 + \lambda_2) = 0$$

$$\Rightarrow x_0^* = \frac{1}{\lambda_1^* + \lambda_2^*}$$

$$\frac{\partial L}{\partial x_1} = \frac{1}{x_1} - \lambda_1 \Rightarrow x_1^* = \frac{1}{\lambda_1^*}$$

$$\frac{\partial L}{\partial x_2} = \frac{1}{x_2} - \lambda_2 \Rightarrow x_2^* = \frac{1}{\lambda_2^*}$$

$$\left. \begin{aligned} \lambda_1^* (x_0^* + x_1^* - 1) &= 0 \\ \lambda_2^* (x_0^* + x_2^* - 2) &= 0 \end{aligned} \right\}$$

Must be the case that

$$\lambda_1^* > 0, \lambda_2^* > 0$$

$$\Rightarrow x_0^* + x_1^* - 1 = 0$$

$$x_0^* + x_2^* - 2 = 0$$

$$\Rightarrow \left. \begin{aligned} x_1^* &= 1 - x_0^* \\ x_2^* &= 2 - x_0^* \end{aligned} \right\} \Rightarrow x_2^* = 1 + x_1^*$$

$$\Rightarrow \frac{1}{\lambda_2^*} = 1 + \frac{1}{\lambda_1^*}$$

$$= \frac{\lambda_1^* + 1}{\lambda_1^*}$$

$$\Rightarrow \lambda_2^* = \frac{\lambda_1^*}{1 + \lambda_1^*}$$

$$\begin{aligned}
 x_0^* &= \frac{1}{\lambda_1^* + \lambda_2^*} \\
 &= \frac{1}{\lambda_1^* + \frac{1}{1 + \lambda_1^*}} = \frac{1 + \lambda_1^*}{\lambda_1^{*2} + 2\lambda_1^*}
 \end{aligned}$$

$$x_1^* = \frac{1}{\lambda_1^*}$$

$$x_2^* = \frac{1}{\lambda_2^*} = \frac{1 + \lambda_1^*}{\lambda_1^*}$$

$$x_0^* + x_1^* = 1$$

$$\boxed{\frac{1 + \lambda_1^*}{\lambda_1^{*2} + 2\lambda_1^*} + \frac{1}{\lambda_1^*} = 1}$$

$$\Rightarrow \lambda_1^{*3} - 3\lambda_1^* = 0$$

$$\Rightarrow \lambda_1^* (\lambda_1^{*2} - 3) = 0$$

$$\lambda_1^* = 0 \text{ or } \sqrt{3} \text{ or } -\sqrt{3}$$


only valid sol_n $\lambda_1^* = \sqrt{3}$

$$\Rightarrow x_0^* = 0.4226$$

$$x_1^* = 0.5774$$

$$x_2^* = 1.5774$$

Utility Functions and Fairness

- There are many ways of defining fairness.
 - E.g., consider dividing a loaf of bread among three people.
 - Should we take into account their ages, sizes, preferences and/or willingness to pay, or just divide equally?
 - We consider the following notions of fairness for allocating network resources:
 - Proportional Fairness
 - Max-sum Fairness
 - Max-min Fairness
 - Minimum Potential Delay Fairness
- 

Utility Functions and Fairness (2)

- Recall that we want to maximize the sum of individual users' utilities, i.e., $\sum_r U_r(x_r)$.
- Different notions of fairness can be unified by considering the utility function of the following form:

$$U_r(x_r) = \frac{x_r^{1-\alpha}}{1-\alpha} \text{ if } \alpha \geq 0 \text{ and } \alpha \neq 1, \\ = \log x_r \text{ if } \alpha = 1.$$

Utility Functions and Fairness (3)

- Proportional Fairness:
 - $U_r(x_r) = \log x_r$.
 - Around the optimal point, we must have $\nabla f(x^*) \cdot (x - x^*) \leq 0$.
 - Hence, $\sum_r \frac{x_r - x_r^*}{x_r^*} \leq 0$ for a feasible point around x^* .
 - Implication: If one of the flow rates is increased by 1% from the optimal point, the sum of the relative changes for the other flows will be at most -1%.
 - Verify this for the previous example of L links and $L + 1$ flows.



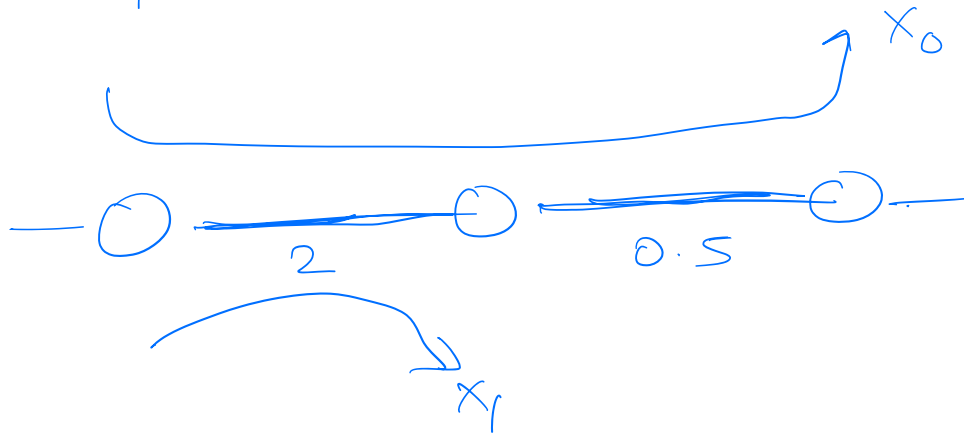
Utility Functions and Fairness (4)

- Max-sum Fairness:
 - With $\alpha = 0$, we maximize the sum of individual rate.
 - In the previous example of L links and $L + 1$ flows, we will have $x_0^* = 0, x_r^* = 1 \forall r \geq 1$.
- Max-min Fairness:
 - With $\alpha \rightarrow \infty$, we maximize the minimum of the allocations.
 - Property: If we attempt to increase the rate for one user, then a user with smaller or the same rate would suffer.
 - In the previous example of L links and $L + 1$ flows, we will have $x_r^* = 0.5 \forall r$.

$$\left(\frac{1}{1-\alpha} x_1^{1-\alpha} \right)$$

$$\frac{1}{1-\alpha} \left(\frac{1}{x_1^{\alpha-1}} + \frac{1}{x_2^{\alpha-1}} \right)$$

$$x_1 < x_2$$



$$x_0 = 0.5$$

$$x_1 = 1.5$$

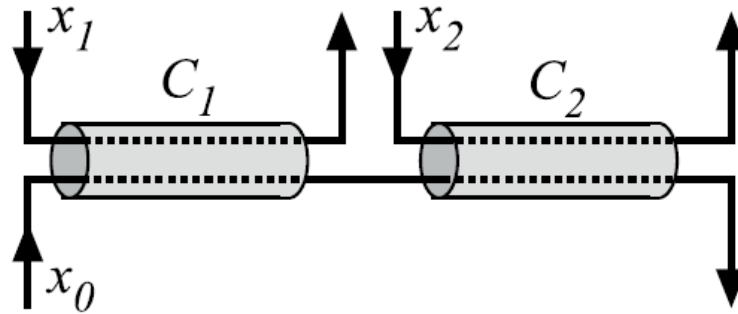
max-min

Utility Functions and Fairness (5)

- Minimum Potential Delay Fairness:
 - $U_r(x_r) = -1/x_r$.
 - With $\alpha = 2$, we minimize $\sum_r 1/x_r$.
 - $1/x_r$ can be thought of as the delay associated with transferring a file of unit size.
 - In the previous example of L links and $L + 1$ flows, we will have

$$x_0^* = \frac{\sqrt{L}-1}{L-1}, x_r^* = \frac{L-\sqrt{L}}{L-1} \forall r \geq 1.$$

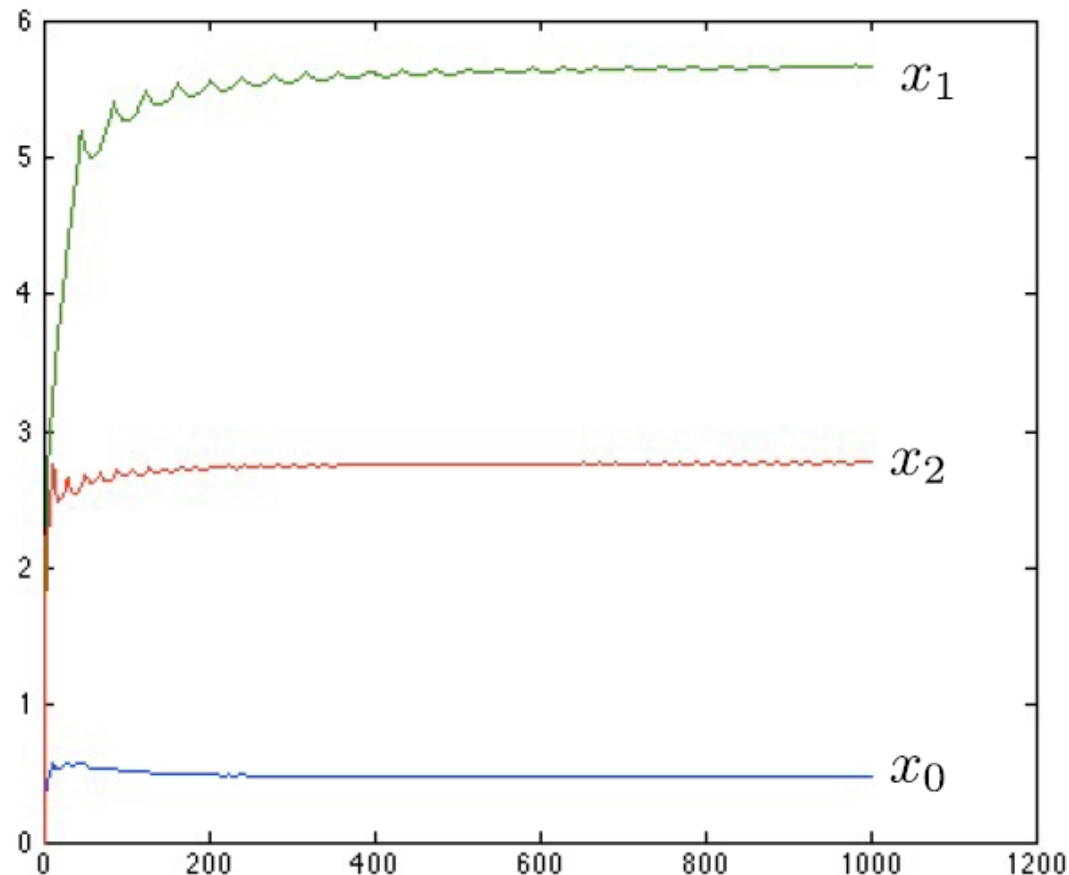
Utility Optimization and TCP



- We compare outcome of sources running AIMD vs. distributed utility maximization.
 - As discussed, AIMD is the window management algorithm underlying TCP.
- Assume $C_1 = 8$, $C_2 = 4$, and small buffers at the two nodes.

Utility Optimization and TCP (2)

- Outcome of AIMD (based on simulations):

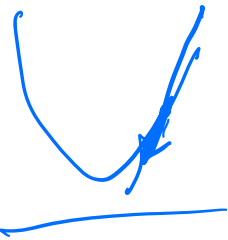


Utility Optimization and TCP (3)

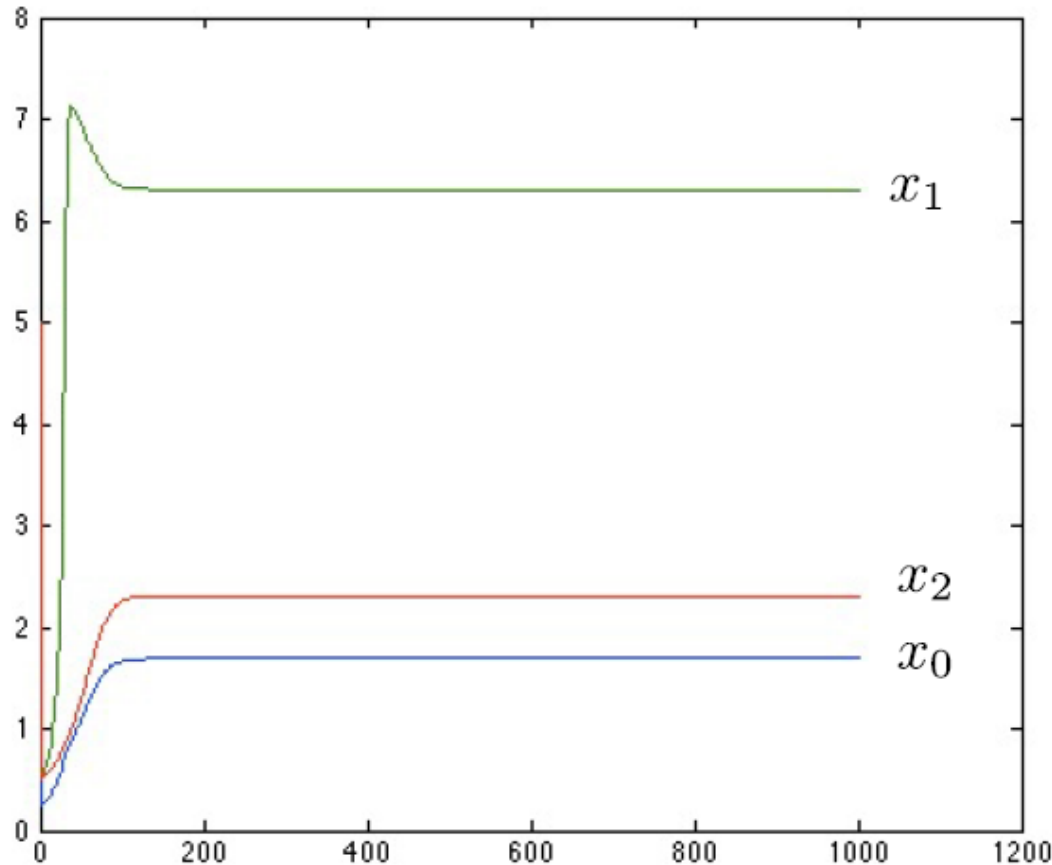
- Distributed Optimization with $U(x) = \log(x)$.
- Recall $L(x, \lambda)$ is given by

$$\sum_{i=0}^2 \log(x_i) - \lambda_1(x_0 + x_1 - C_1) - \lambda_2(x_0 + x_2 - C_2) \quad \}$$

- Maximization with respect to x and minimization with respect to λ can be achieved as follows:
 - Source i selects x_i to maximize $\log(x_i) - \gamma_i x_i$, where $\gamma_0 = \lambda_1 + \lambda_2, \gamma_1 = \lambda_1, \gamma_2 = \lambda_2$.
 - Network provides λ_1 and λ_2 by updating them as follows (using gradient projection):
 - $\lambda_i(n+1) = \max\{0, \lambda_i(n) - \alpha_n \frac{\partial}{\partial \lambda_i} L(x, \lambda(n))\}$ where $\alpha_n > 0$ determines the step size.
 - Hence, $\lambda_i(n+1) = \max\{0, \lambda_i(n) + \frac{1}{n}(x_0 + x_i - C_i)\}$, with $\alpha_n = \frac{1}{n}$.



Utility Optimization and TCP (4)



Results from Distributed Optimization

- More recent research suggests that a utility function different than $\log(x)$ can match the results better.

Queues

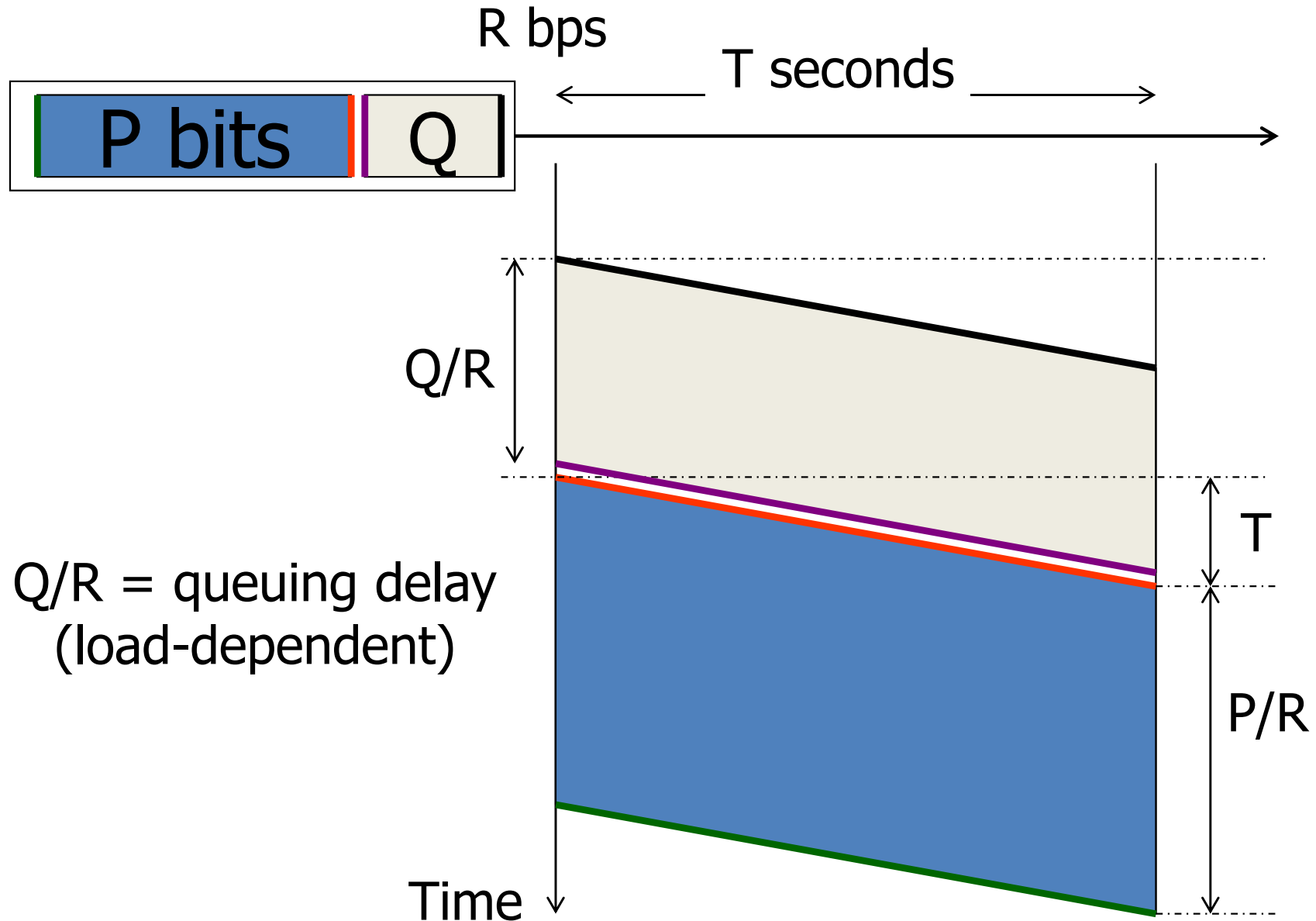
Origins of Queueing Theory

- Danish mathematician and engineer Agner Krarup Erlang (1878-1929) invented Queueing Theory.
 - He used this theory to analyze Copenhagen Telephone Exchange.
 - In his honor, the loading level of “calls” is measured in units of Erlangs.

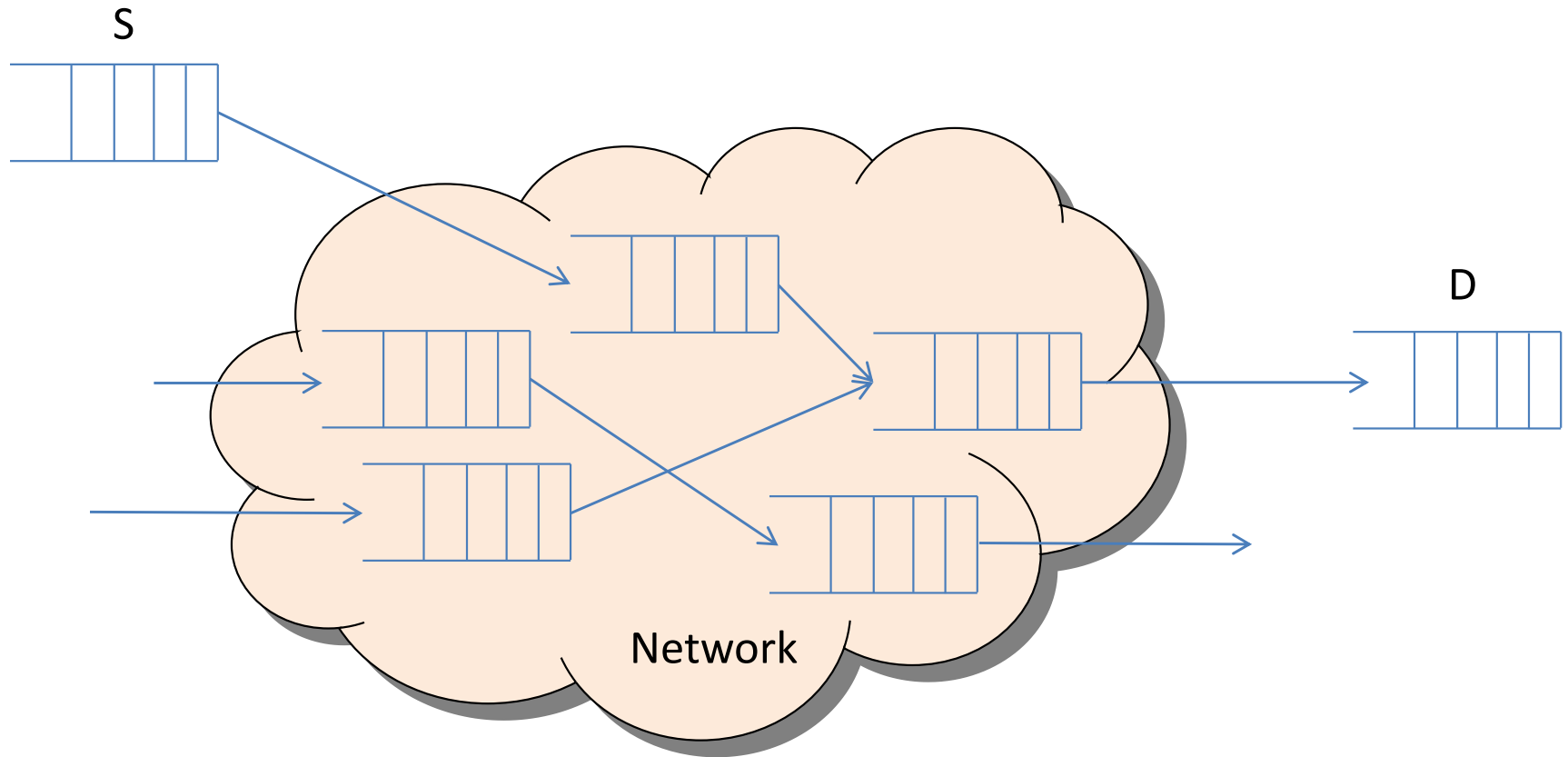


Queuing

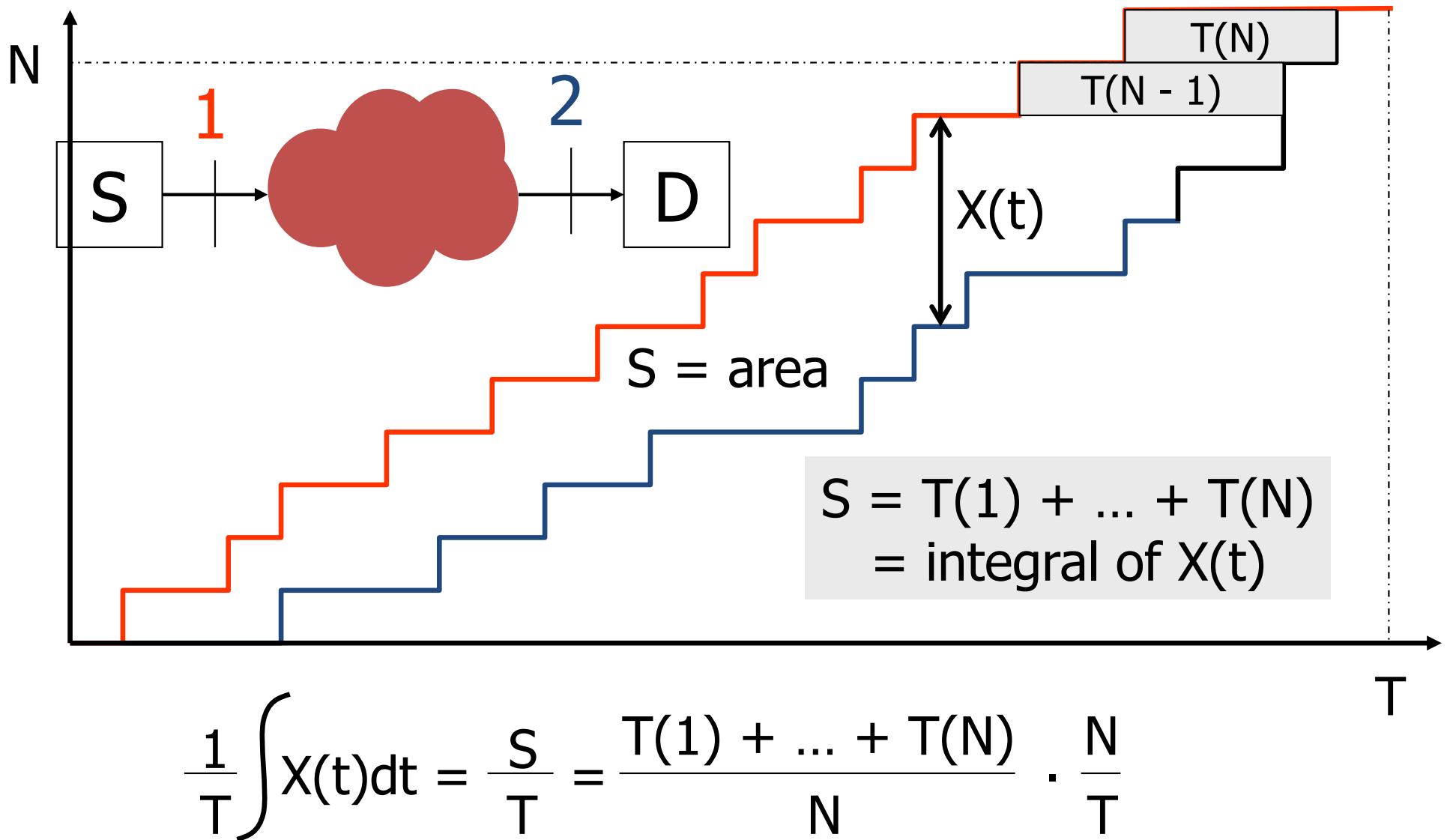
- Going over a link:



Queueing Network



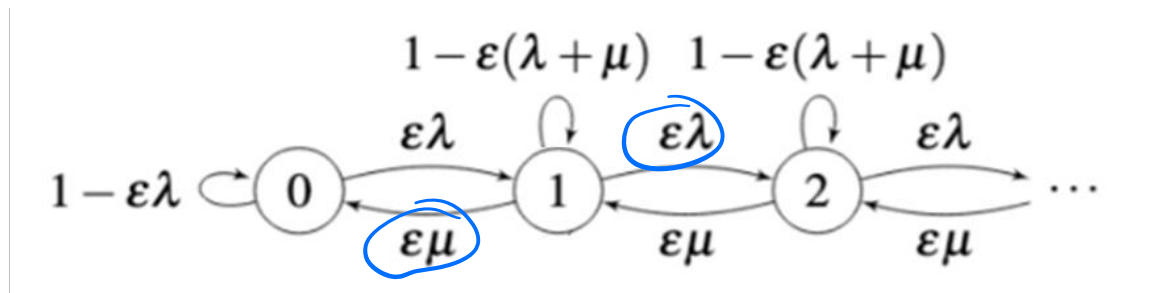
Fundamental Result: Little's Result



→ Average occupancy = (average delay) x (average arrival rate)

M/M/1 Queue

- Let $\lambda > 0$ be the packet arrival rate and $\mu > 0$ be the packet service rate.
- For stability of the queue, let's assume that $\lambda < \mu$.
- Consider an approximate DTMC with a small time step $\varepsilon > 0$.



Discrete-Time Markov Chain Approximation for an M/M/1 Queue

$$\begin{aligned}
 P(X \leq \varepsilon) &= 1 - e^{-\lambda \varepsilon} && \text{Taylor Expansion} \\
 &\downarrow \\
 \text{Exp}(\lambda) &= 1 - \left(1 - \lambda \varepsilon + \frac{(\lambda \varepsilon)^2}{2} + \dots \right)
 \end{aligned}$$

$\mathcal{O}(\varepsilon)$

M/M/1 Queue (2)

- Invariant Distribution

Cut 1:

$$\lambda \varepsilon \pi(0) = \mu \varepsilon \pi(1)$$

Cut n+1:

$$\begin{aligned} \lambda \varepsilon \pi(n) &= \mu \varepsilon \pi(n+1) \\ \Rightarrow \pi(n+1) &= \frac{\lambda}{\mu} \pi(n) = \left(\frac{\lambda}{\mu}\right)^{n+1} \pi(0) \end{aligned}$$

It can be shown that the DTMC is positive recurrent, and hence has a unique invariant distribution.

Since $\sum_{i=0}^{\infty} \pi(i) = 1$, $\sum_{i=0}^{\infty} \pi(0) \left(\frac{\lambda}{\mu}\right)^i = 1$.

This gives $\pi(n) = (1 - \rho)\rho^n$, $n \geq 0$, where $\rho = \frac{\lambda}{\mu}$.

$$\lambda < \mu \Leftrightarrow \rho < 1$$

M/M/1 Queue (3)

- Average # in the system:

$$E(X_t) = \sum_{n=0}^{\infty} n\pi(n) = \sum_{n=0}^{\infty} n(1-\rho)\rho^n = \frac{\rho}{1-\rho} = \frac{\lambda}{\mu-\lambda}.$$

$\rho < 1$

- Fact:** Poisson Arrivals See Time Average (PASTA).

- Average System Delay:

$$E(T) = \sum_{n=0}^{\infty} \frac{n+1}{\mu} \pi(n) = \sum_{n=0}^{\infty} \frac{n+1}{\mu} (1-\rho)\rho^n = \frac{1}{\mu-\lambda}.$$

- Alternatively, by Little's Result, $E(X_t) = \lambda E(T)$. Hence,

$$E(T) = \frac{1}{\lambda} E(X_t) = \frac{1}{\lambda} \frac{\lambda}{\mu-\lambda} = \frac{1}{\mu-\lambda}.$$

- **Fact:** System Delay is exponentially distributed with the average of $\frac{1}{\mu-\lambda}$.

- Let ρ = fraction of time the server is busy. Then, $\lambda\tau = \rho\tau\mu$ (# of arrivals = # of departures) for some large time τ . Hence, $\rho = \lambda/\mu$.

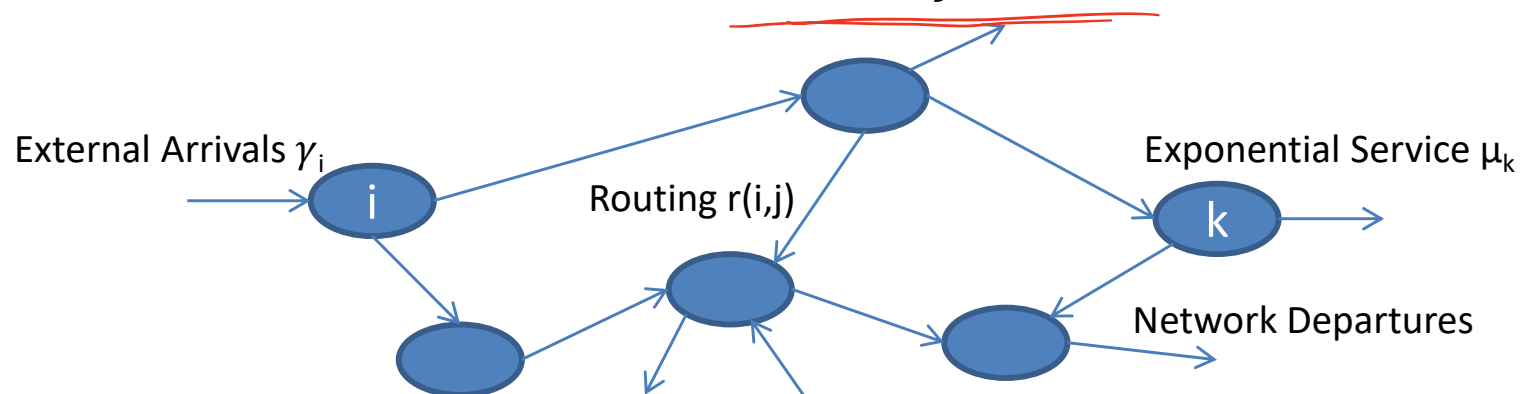
- Observe that average # with server = fraction of time the server is busy.

- Hence, average # in the queue (i.e., waiting) = $\frac{\rho}{1-\rho} - \rho = \frac{\rho^2}{1-\rho}$.

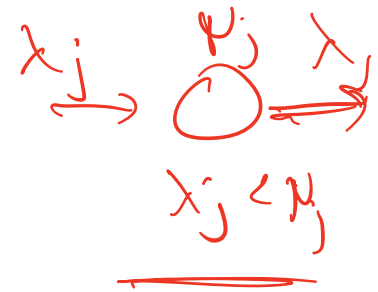
- Average queueing (waiting) time = $\frac{\rho}{1-\rho} \frac{1}{\mu} = \frac{\rho}{\mu-\lambda}$.

Jackson Network

- Definition:
 - Network of J ./M/1 queues,
 - External arrivals occur according to independent Poisson processes with rate γ_i into queue i ,
 - Service time at queue i is according to independent exponential distribution with rate μ_i , and
 - When a customer leaves queue i , independent of the past, he joins queue j with probability $r(i,j)$ and leaves the network with probability $1 - \sum_{j=1}^J r(i,j)$.



Jackson Network (2)



- Due to flow conservation, total arrival rate into queue i is given by

$$\lambda_i = \gamma_i + \sum_{j=1}^J \lambda_j r(j, i), \text{ for } i = 1, 2, \dots, J.$$

- Assume $\lambda_i < \mu_i$ for $i = 1, 2, \dots, J$.
- For $t \geq 0$, we define $X_t = (X_{1,t}, \dots, X_{J,t})$, where $X_{i,t}$ denotes total # of customers at node i .
 - X_t is a multidimensional CTMC.

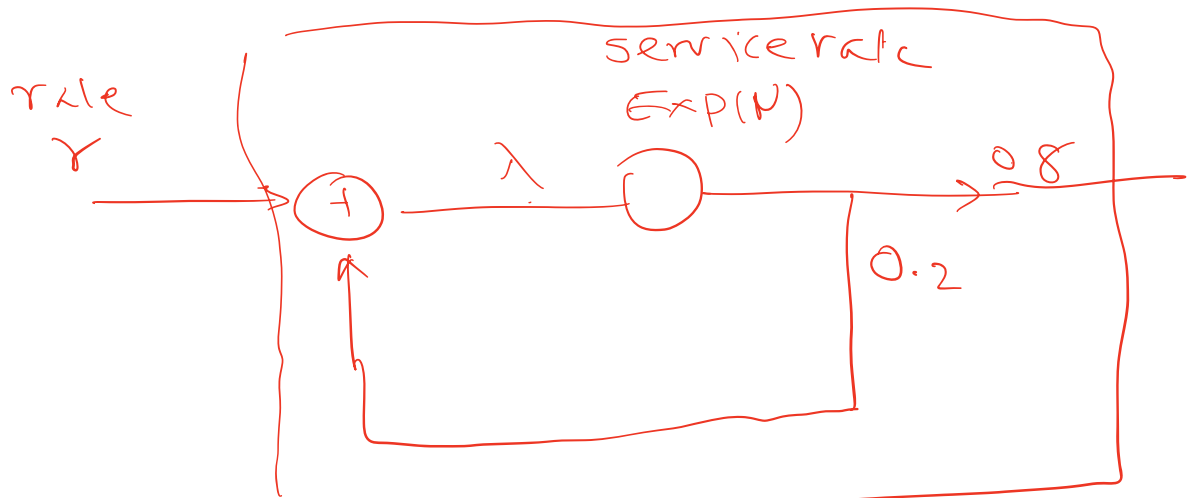
Jackson Network (3)

- Theorem: Assume that the solution $(\lambda_1, \dots, \lambda_J)$ is such that $\lambda_i < \mu_i$ for $i = 1, \dots, J$. Then, the CTMC X_t admits the following invariant distribution:

$$\pi(x_1, \dots, x_J) = \pi_1(x_1) \dots \pi_J(x_J), \text{ where}$$

For each j , $\pi_j(n) = (1 - \rho_j)\rho_j^n$, for $n \geq 0$ and $\rho_j = \frac{\lambda_j}{\mu_j}$.

- I.e., each queue behaves like an M/M/1 queue with appropriate utilization.



Find average delay

$$\gamma = 4, N = 10$$

assume $\lambda < N$: $\lambda = \gamma + 0.2\lambda$

$$\Rightarrow \lambda = 1.25\gamma$$

$$\Rightarrow \lambda = 5$$

see that $\lambda < N$ is valid.

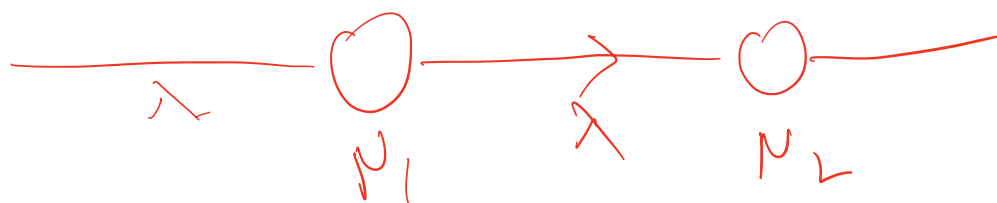
$$N : \text{Avg in queue} = \frac{\rho}{1-\rho}, \rho = 0.5$$

$$= 1$$

(Avg Delay)

$$D = \gamma = N = 1$$

$$\Rightarrow D = 0.25 \text{ seconds}$$



$$\lambda < p_1$$

Graphs

Beginning of Graph Theory

- Paper by L. Euler in 1736 on the Seven Bridges of Königsberg.



- Find a walk that crosses each bridge exactly once while starting and ending at the same spot (land mass).

No solution.

- How about starting and ending at different spots (land masses)?

No solution.

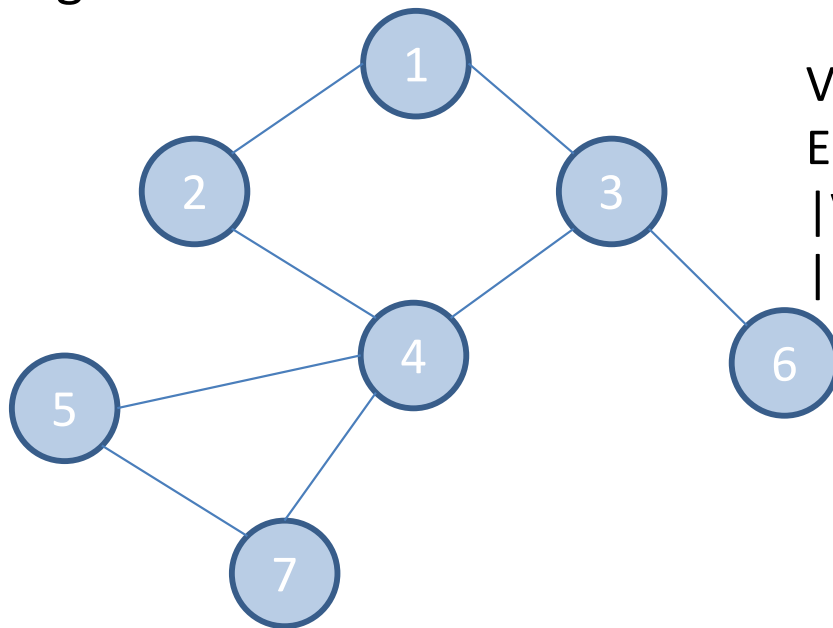
Additional Examples of Graphs

Application	Nodes	Edges
Internet	Routers	Links
Maps	Points of Interest	Roads
Air Travel	Airports	Routes
Social Network	People	Friendship
Retail Business	Stores, People	Purchase Likelihood
Genealogy	People	Parent-Child Relation
Epidemiology	People	Disease Spread Likelihood

Graphs: Definitions

- **Undirected Graphs**

- Notation: $G = (V, E)$
- V = Set of nodes
- E = Set of edges between each pair of nodes
- Graph size parameters: $|V|$ = Total # of nodes, and $|E|$ = Total number of edges



$V = \{1, 2, 3, 4, 5, 6, 7\}$

$E = \{1-2, 1-3, 2-4, 3-4, 3-6, 4-5, 4-7, 5-7\}$

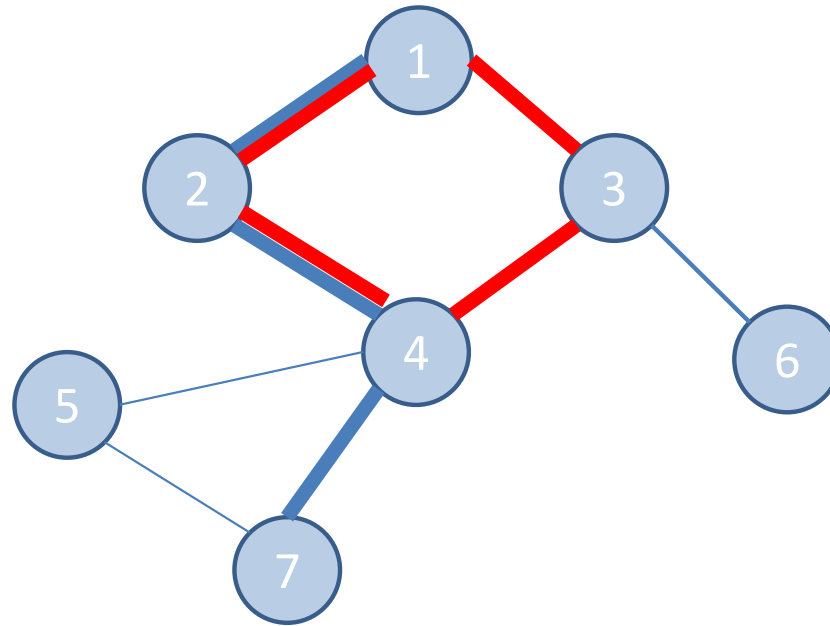
$|V| = 7$

$|E| = 8$

- **Directed Graphs:** Each edge is directional; edge $u-v$ indicates edge from node u to node v .

Graphs: Definitions (2)

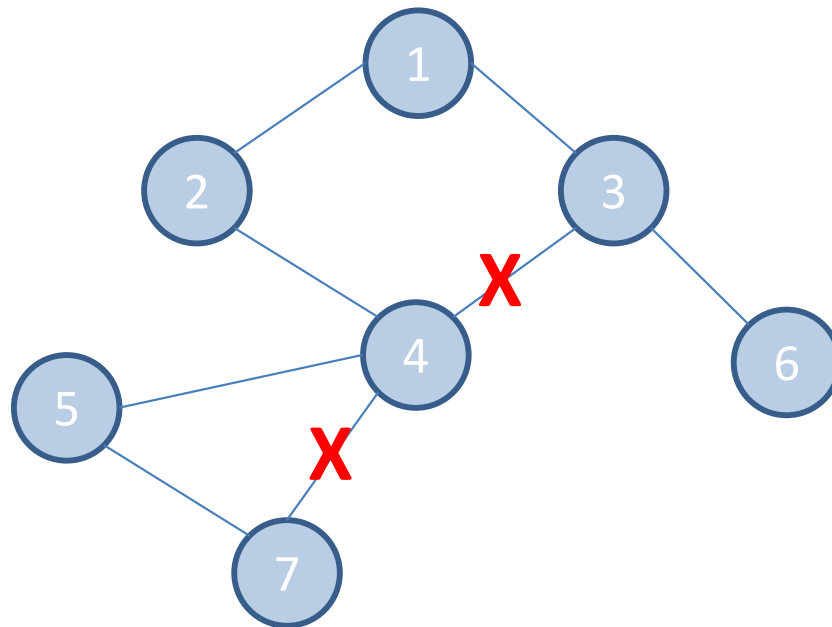
- **Adjacency Matrix:** $|V| \times |V|$ matrix where the element $(a, b) = 1$, if the edge $a-b$ exists, and 0, otherwise.
- **Incidence Matrix (Undirected Graph):** $|V| \times |E|$ matrix where the element $(x, y) = 1$, if edge y is incident to node x , and 0, otherwise.



- **Path:** A path is a sequence of nodes v_1, v_2, \dots, v_k s.t. the edge v_i-v_{i+1} exists for $i=1, 2, \dots, k-1$ (e.g., 1, 2, 4, 7 is a path).
 - A path is called **simple** if all vertices are distinct.
- **Cycle:** A cycle is a path s.t. all nodes except the first and the last nodes are distinct (e.g., 1, 3, 4, 2, 1 is a cycle).

Graphs: Definitions (3)

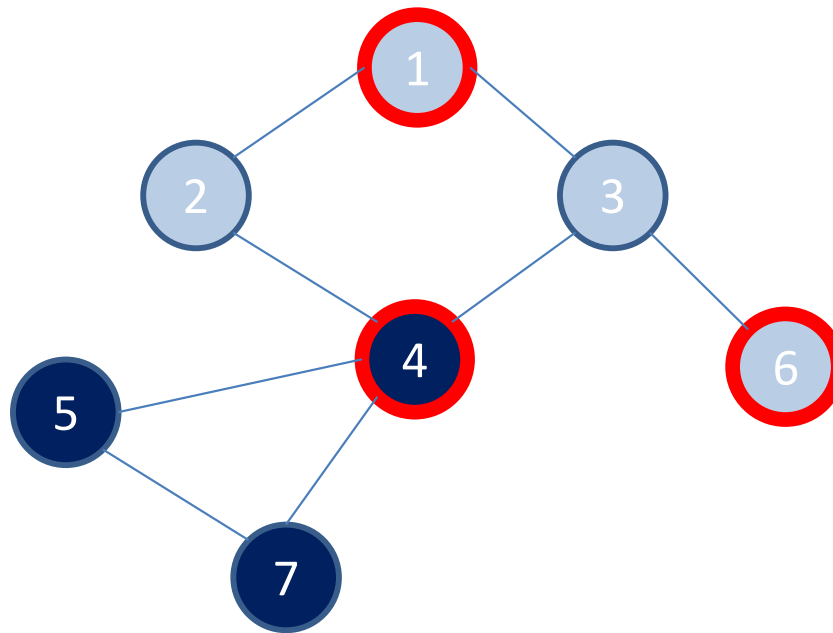
- **Connected Graph:** A graph is said to be connected if there is a path between any two nodes.
 - A directed graph is said to be **strongly connected** if for every pair of nodes u, v , there is path from u to v , and from v to u .



- **Tree:** An undirected graph is a tree if it's connected, and it does not have any cycles. (e.g., the graph after removing the edges 3-4 and 4-7 is a tree).

Graphs: Definitions (4)

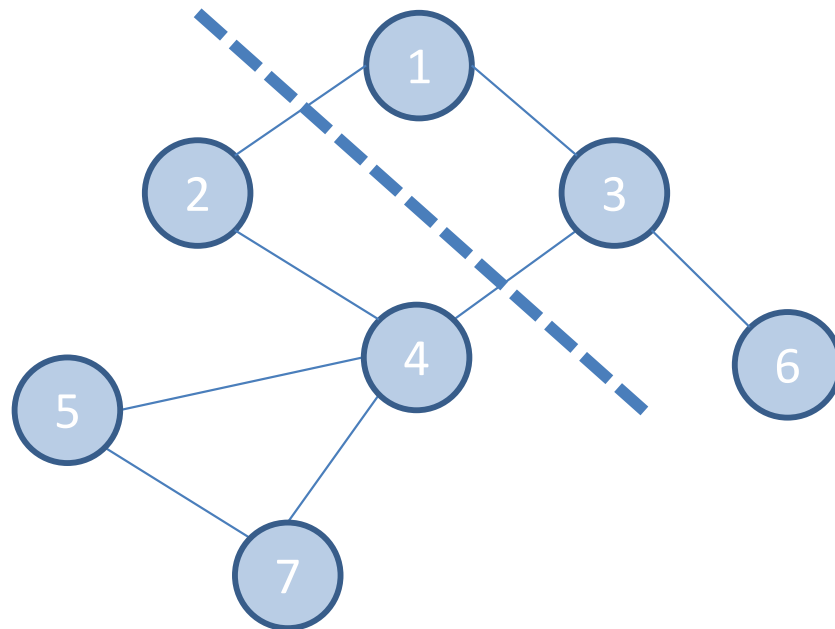
- **Clique:** A subset of vertices of an undirected graph is a clique if every pair of nodes in the subset have an edge connecting them (e.g., $\{4, 5, 7\}$ is a clique).



- **Independent Set:** A subset of vertices of an undirected graph is an independent set if no two nodes in the subset are adjacent (e.g., $\{1, 4, 6\}$ is an independent set).

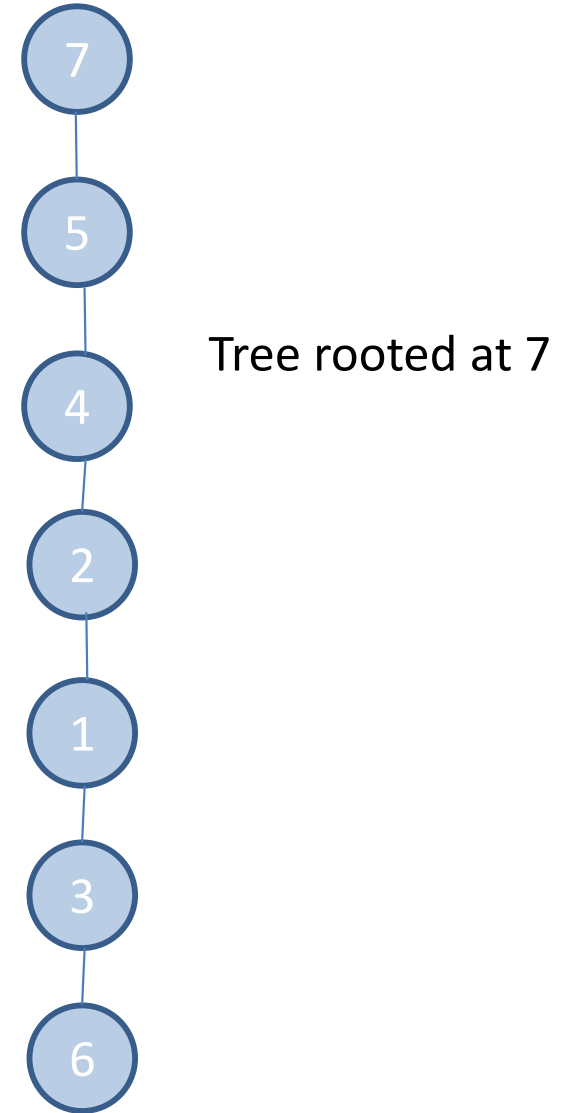
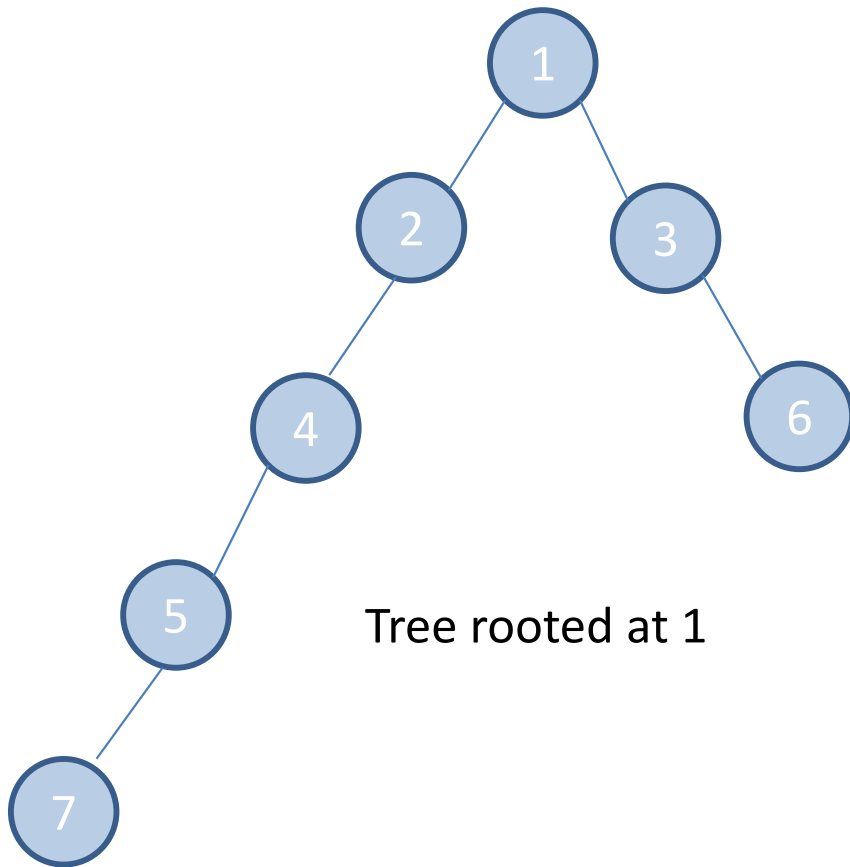
Graphs: Definitions (5)

- **Cut:** A partition of the nodes into two disjoint subset (e.g., {1, 3, 6} and {2, 4, 5, 7}).
- **Cut-set:** For a given cut, it's the set of edges whose end points are in the different subsets of the partition (e.g., {1-2, 3-4} for the cut shown).
- **x-y Cut:** An x-y cut is a partition where x belongs to one subset, and y belongs to the other.



Trees

- A **rooted tree** is just the graphic representation of the given tree by “hanging” the tree from the specified root.
 - A given tree can be rooted from each of its nodes.



Trees (2)

- **Theorem:**

Let G be an undirected graph on n nodes. Any two of the following statements imply the third.

(i) G is connected.

(ii) G does not contain a cycle.

(iii) G has $n - 1$ edges.

Graph Connectivity & Traversal

- Given the nodes s and t of a graph, we want to find whether s and t are connected, and if so how to go from s to t .
- For large graphs, we need a well specified method (algorithm).
- This is also called the Maze Solving Problem.
- We consider two algorithms:
 - Breadth-First Search (BFS)
 - Depth-First Search (DFS)

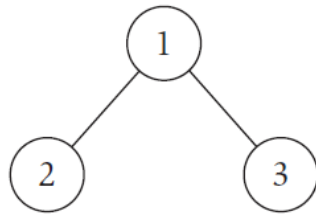
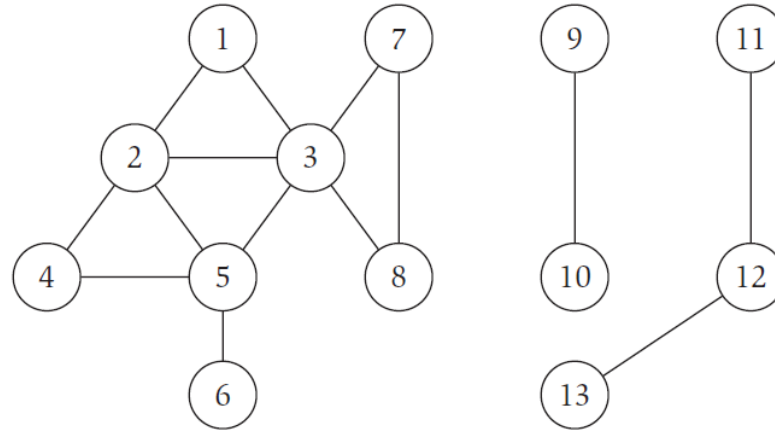
Breadth-First Search (BFS)

- Take s as the starting node.
- Identify all nodes connected to s (Layer 1).
- Next, identify all NEW nodes connected to the nodes in Layer 1.
 - This will be the Layer 2.
 - Examine the nodes in Layer 1 in any order.
- Continue recursively until no new nodes are found.
 - Algorithm is looking for all nodes connected to s .
- If t is found, s - t connectivity exists.
- Search provides the shortest path from s to t .
- Time Complexity: $O(V + E)$.

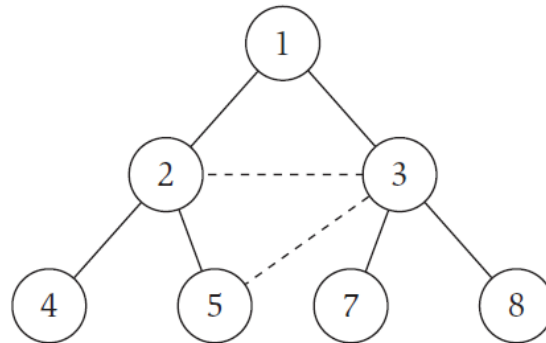
Breadth-First Search (BFS) (2)

- Example:

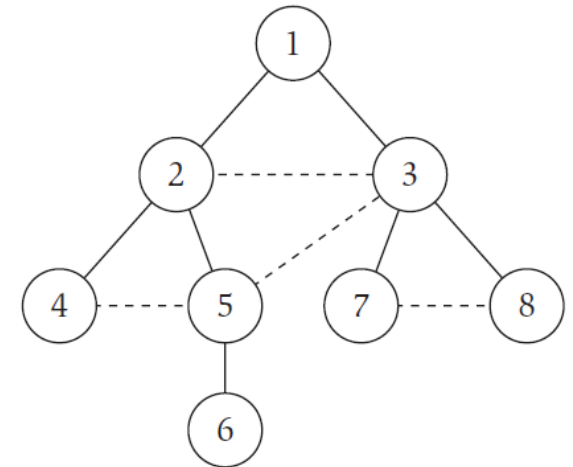
G



(a)



(b)



(c)

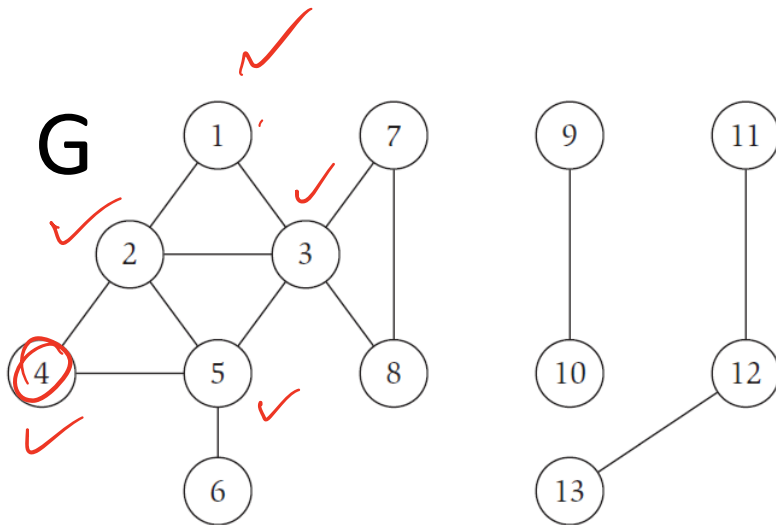
- Observe that the search is constructing a tree rooted at s (node 1).
- The dashed lines of the graph are not used by the tree.

Depth-First Search (DFS)

- Considers a different way of searching for connectivity.
- Start from s , and examine connectivity from the first edge out of s .
- Suppose that connects to node v . Then, follow the first edge out of v .
- Continue this way, until there is a “dead-end”, i.e., no edge to a new node.
- Then, backtrack to a node with an unexplored edge, and repeat.
- Time Complexity: $O(V + E)$.

Depth-First Search (DFS) (2)

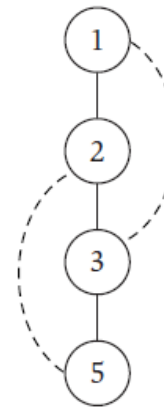
- Example:



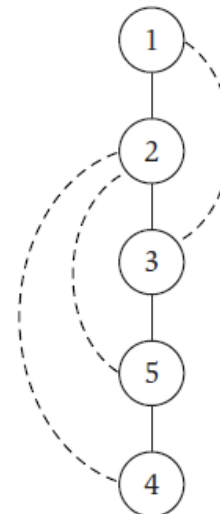
(a)



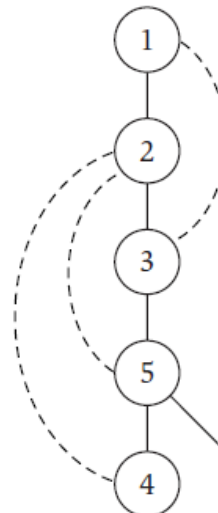
(b)



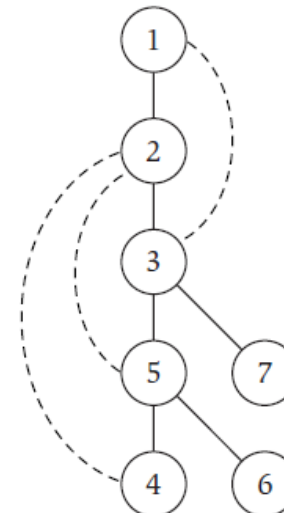
(c)



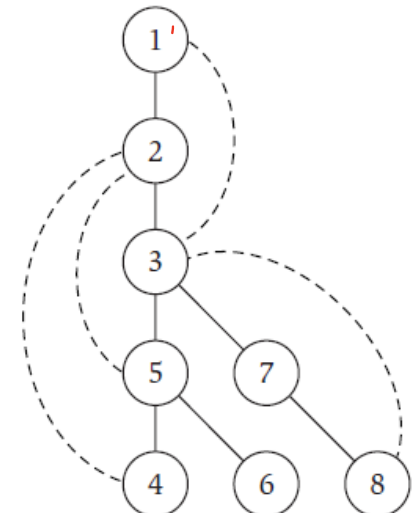
(d)



(e)



(f)

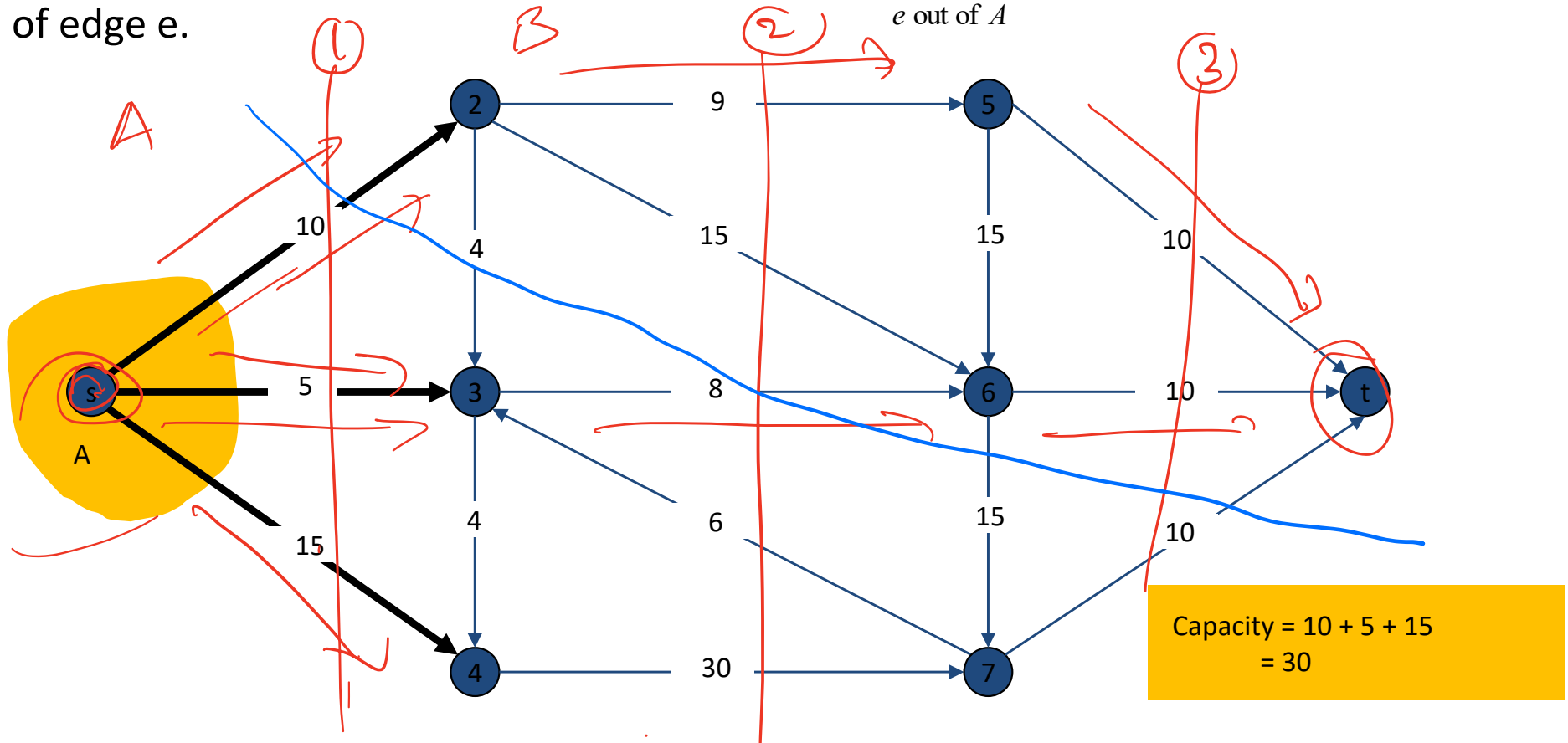


(g)

- Observe that the search is constructing a tree rooted at s (node 1).
- The dashed lines of the graph are not used by the tree.
- Does DFS find the shortest paths?

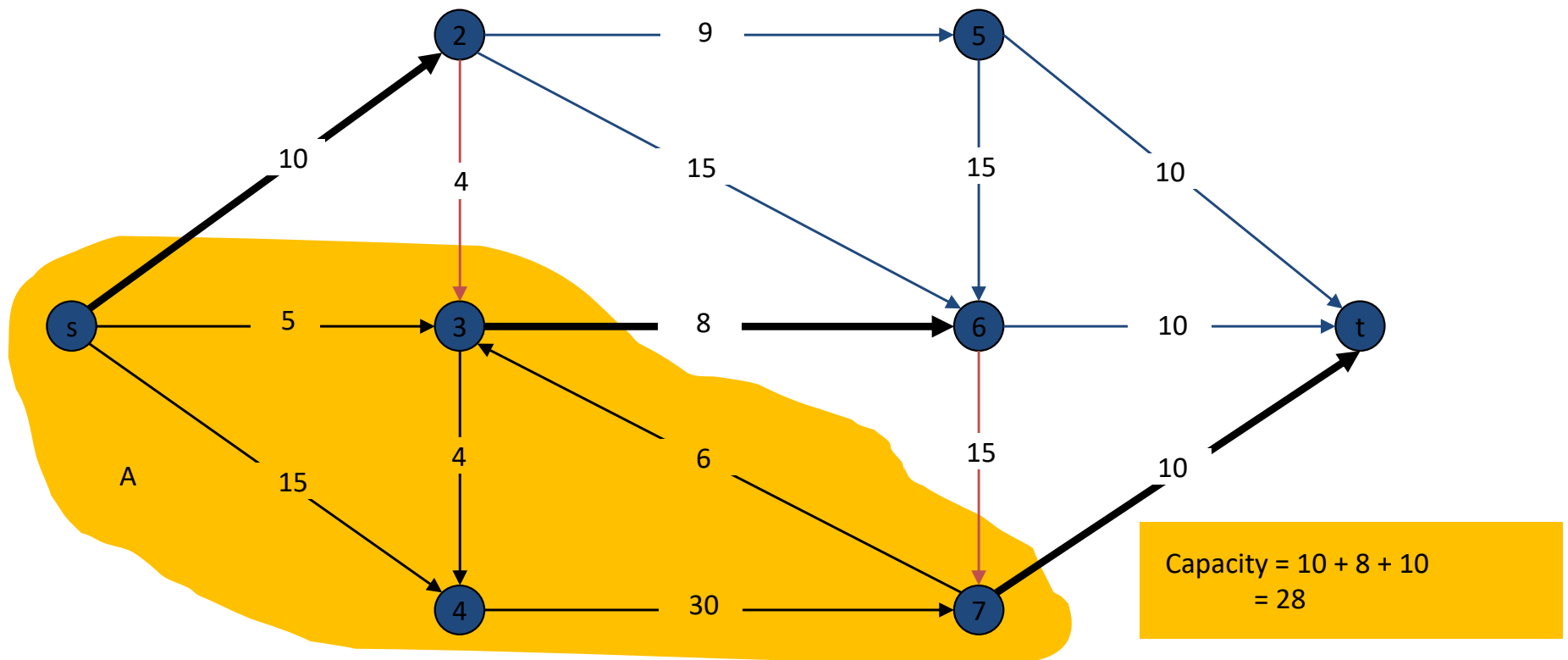
How large a flow can be?

- Answered by the max-flow min-cut theorem.
- To lead to the theorem, let's develop some concepts.
- Def. An **s-t cut** is a partition (A, B) of V with $s \in A$ and $t \in B$.
- Def. The **capacity** of a cut (A, B) is: $cap(A, B) = \sum_{e \text{ out of } A} c(e)$ where $c(e)$ is capacity of edge e .



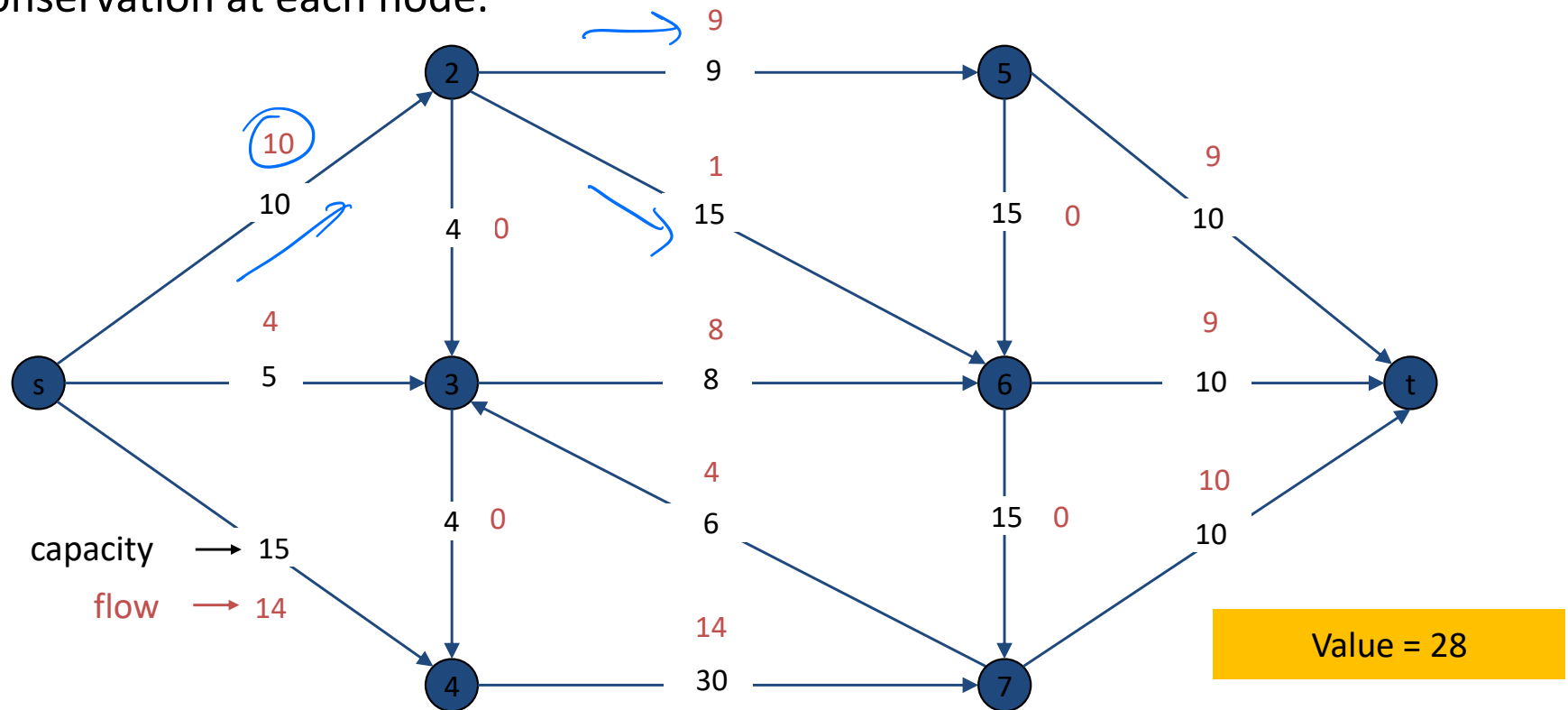
Minimum Cut Problem

- Min s-t cut problem. Find an s-t cut of minimum capacity.



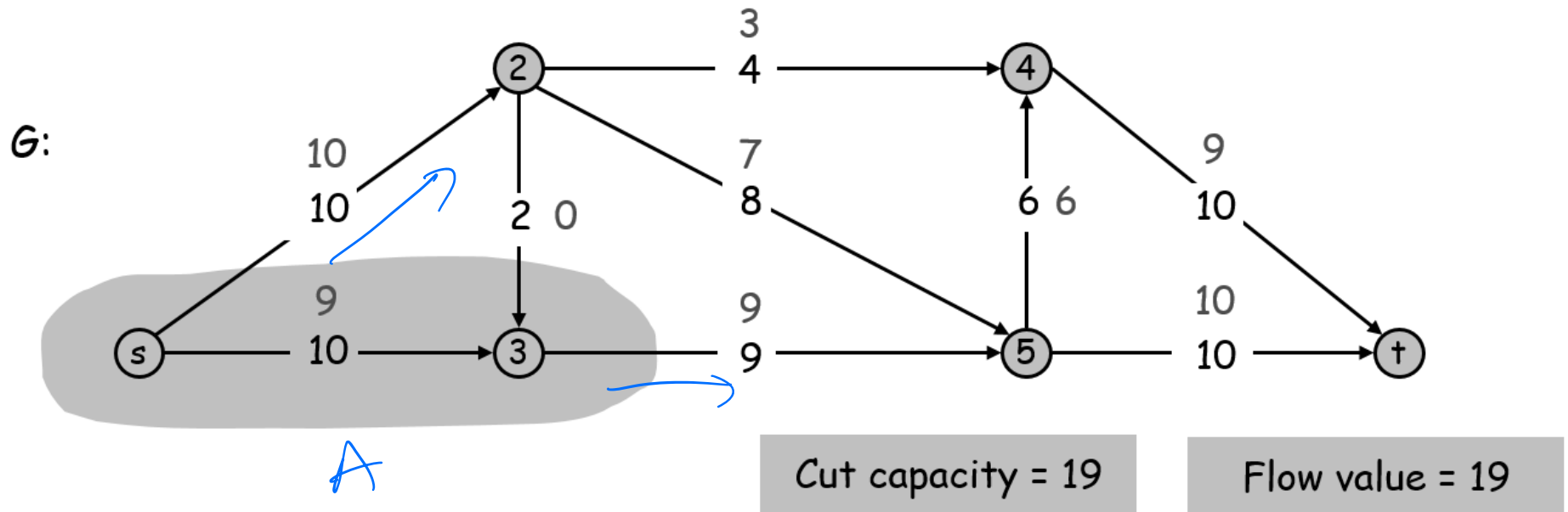
Maximum Flow Problem

- Max flow problem. Find s-t flow of maximum value.
- Numbers in red contribute to the overall flow from s to t while satisfying flow conservation at each node.



Max-Flow Min-Cut Theorem

- Max-flow min-cut theorem. [Elias-Feinstein-Shannon 1956, Ford-Fulkerson 1956]
The value of the max flow is equal to the value of the min cut capacity.

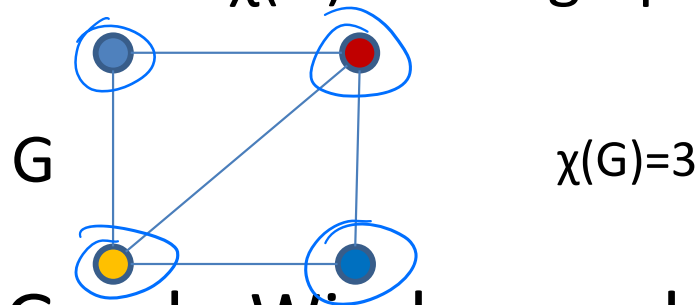


Example

(Solvable using Ford-Fulkerson Algorithm)


Chromatic Number

- Vertex Coloring Problem: Given a graph G , color the nodes such that no edge has the same color at the two ends.
 - Minimum number of colors required is called the Chromatic Number $\chi(G)$ of the graph.



- Interference Graph: Wireless nodes connected by an edge indicate interference between the two nodes.
 - Frequency Planning: $\chi(G)$ give the minimum number of carriers needed to avoid interference between the neighbors.

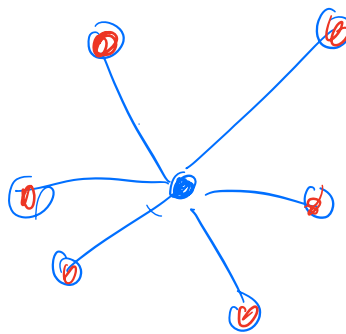
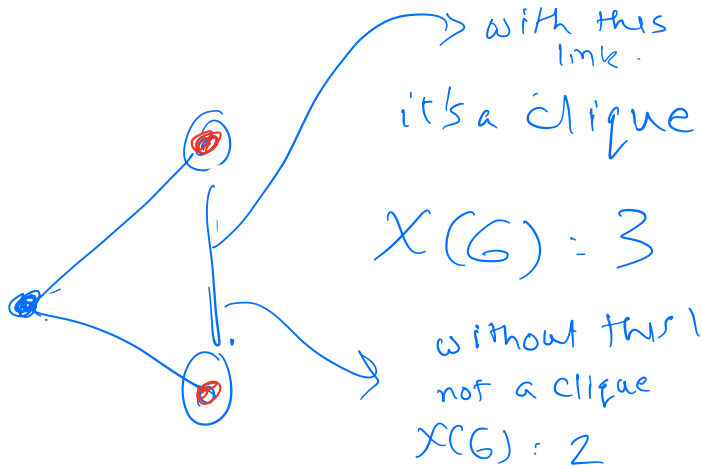
Vertex Coloring

- Vertex Coloring: Given an undirected graph G , find a function $f: V \rightarrow C$ where $C =$ set of colors, such that
 - For each edge $u-v$, $f(u) \neq f(v)$.
 - Chromatic Number $\chi(G)$ is the minimum size of C such that there is a Vertex Coloring to C .
- 

Chromatic Number: Well-known Results

- $\chi(G)$ of a planar graph G is at most 4.
- A graph G is a bipartite graph if and only if $\chi(G) = 2$.
- Let $\Delta(G)$ = maximum degree of vertices of G .
Then, $\chi(G) \leq \Delta(G) + 1$.
- Brook's Theorem: Let G be a connected graph.
If G is not a clique or a cycle of odd length,
 $\chi(G) \leq \Delta(G)$.
 - For a clique or an odd cycle, $\chi(G) = \Delta(G) + 1$.

$$\Delta(G) = 2$$



$$\Delta(G) = 6$$

$$\chi(G) = 2$$

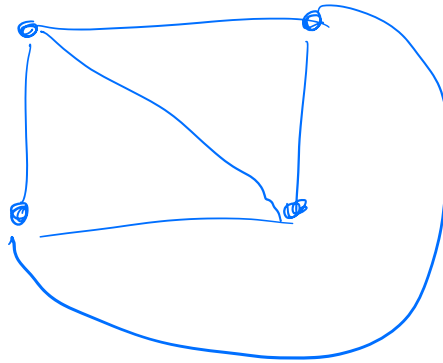
odd cycle.



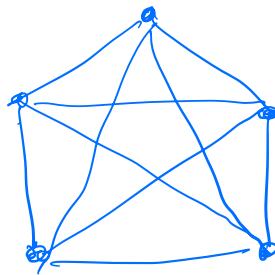
$$\Delta(G) = 2$$

$$\chi(G) = \Delta(G) + 1 = 3$$

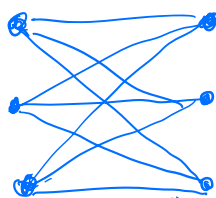
K_4 : Complete Graph of 4 nodes
is planar



K_5 : Complete Graph of 5 nodes
is not planar



$K_{3,3}$: Bipartite graph of
3 nodes on each side.
is not planar.



Appendix: Queues

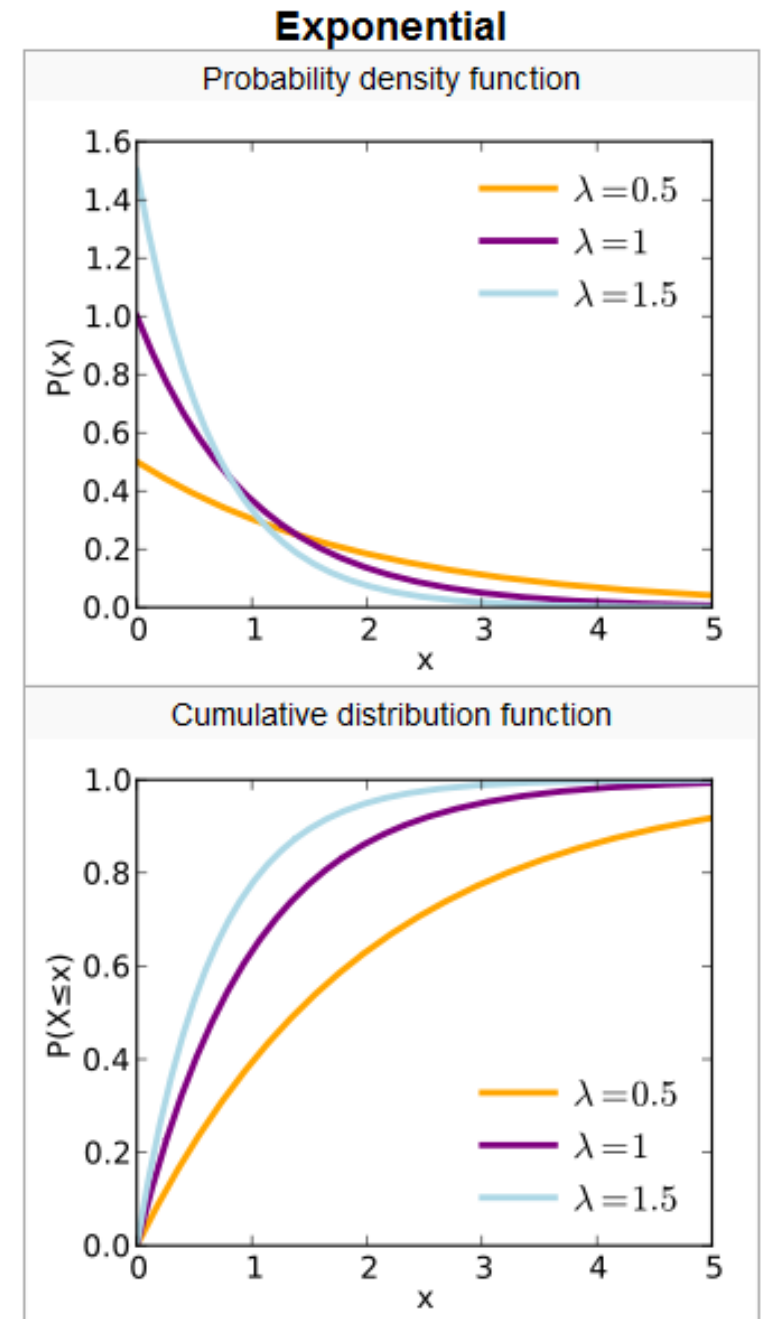
Exponential Distribution

- Definition: The random variable Z is exponentially distributed with rate $\lambda > 0$ if

$$P(Z \leq t) = 1 - e^{-\lambda t}, \text{ for } t \geq 0. \text{ (CDF or PDF)}$$

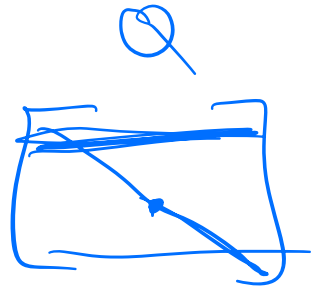
$$\text{(Or } P(Z > t) = e^{-\lambda t}, \text{ for } t \geq 0. \text{ (CCDF))}$$

- Facts for an exponentially distributed random variable Z :
 - Probability density function:
 $f_Z(t) = \lambda e^{-\lambda t}, t \geq 0. \text{ (pdf)}$
 - $E(Z) = \lambda^{-1}$.
 - $\text{Var}(Z) = \lambda^{-2}$. (Recall $\text{Var}(Z) = E(Z^2) - (E(Z))^2$)
 - $P(Z > t + s \mid Z > s) = P(Z > t)$ for all $s, t \geq 0$. (Memoryless property)



Continuous-Time Markov Chains

- Rate Matrix: Let \mathbf{X} be a countable set. A rate matrix $\mathbf{Q} = \{q(i,j), i, j \text{ in } \mathbf{X}\}$ of real numbers is such that
$$0 \leq q(i,j) < \infty \text{ for } i \neq j \text{ in } \mathbf{X}, \text{ and}$$
$$q(i) := -q(i,i) := \sum_{\{j \neq i\}} q(i,j) < \infty \text{ for all } i \text{ in } \mathbf{X}.$$
- Given a countable set \mathbf{X} , and a distribution π and a rate matrix \mathbf{Q} and on \mathbf{X} , we construct a Continuous-Time Markov Chain (CTMC) X_t , for $t \geq 0$ as follows:
 - Let π be the initial distribution, i.e., $P(X_0 = i) = \pi(i)$, for all i in \mathbf{X} .
 - Next, if $X_0 = i$, select a random time τ according to an exponential distribution with rate $q(i)$.
 - Let $X_t = i$ for $0 \leq t < \tau$.
 - At $t = \tau$, X_t jumps to j independently of τ such that
$$P(X_\tau = j \mid X_0 = i \text{ and } \tau) = \Gamma(i,j) := q(i,j)/q(i), \text{ for all } j \neq i.$$
 - Construction then resumes at $X_\tau = j$ in the same way as from $X_0 = i$ independent of the past.

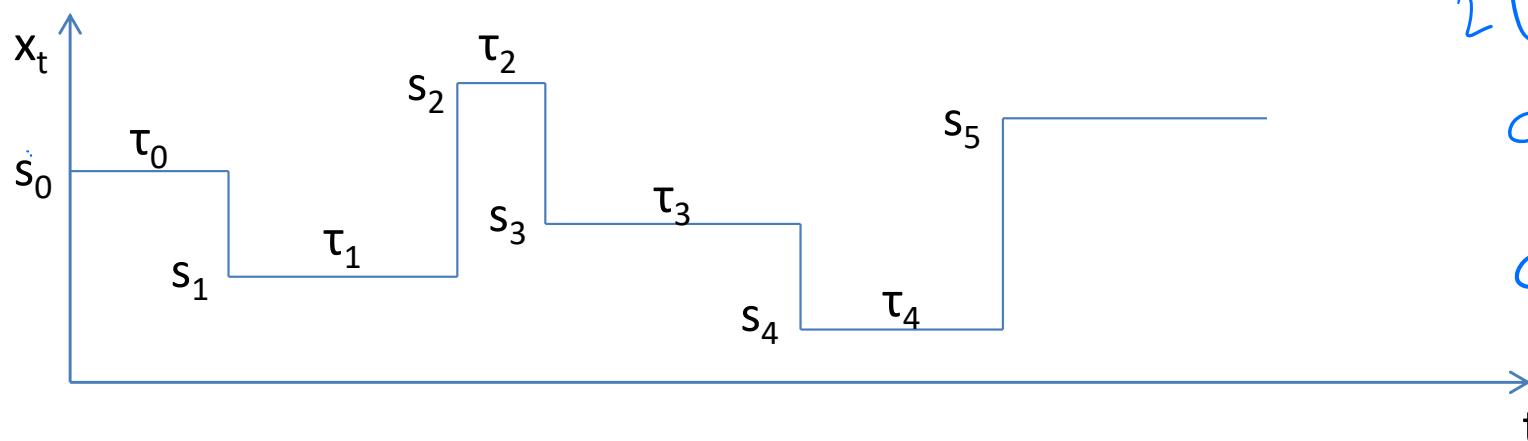


$$i \rightarrow j$$

$$a_{ij}/a_{ii}$$

Continuous-Time Markov Chains (2)

- Typical realization of a CTMC:



$$Q = \begin{bmatrix} q_{11} & q_{12} \\ q_{21} & q_{22} \end{bmatrix}$$

$$q_{11} = -q_{12}$$

$$q_{22} = -q_{21}$$

- A rate matrix Q on \mathbf{X} is irreducible if $q(i) > 0$ for all i in \mathbf{X} and if the transition probability matrix Γ defined by $\Gamma(i,j) = q(i,j)/q(i)$ if $i \neq j$, $= 0$ if $i = j$, is irreducible.

Continuous-Time Markov Chains (3)

- **Theorem:** Let X_t be an irreducible CTMC over \mathbf{X} with the rate matrix Q and initial distribution π . The distribution π is invariant, i.e.,

$$P(X_t = i) = \pi(i) \text{ for all } i \text{ in } \mathbf{X} \text{ and all } t \geq 0$$

if and only if

Π satisfies the following balance equations:

$$\sum_{i \text{ in } \mathbf{X}} \pi(i)q(i, j) = 0 \text{ for all } j \text{ in } \mathbf{X} \text{ (or } \pi Q = 0).$$

Continuous-Time Markov Chains (4)

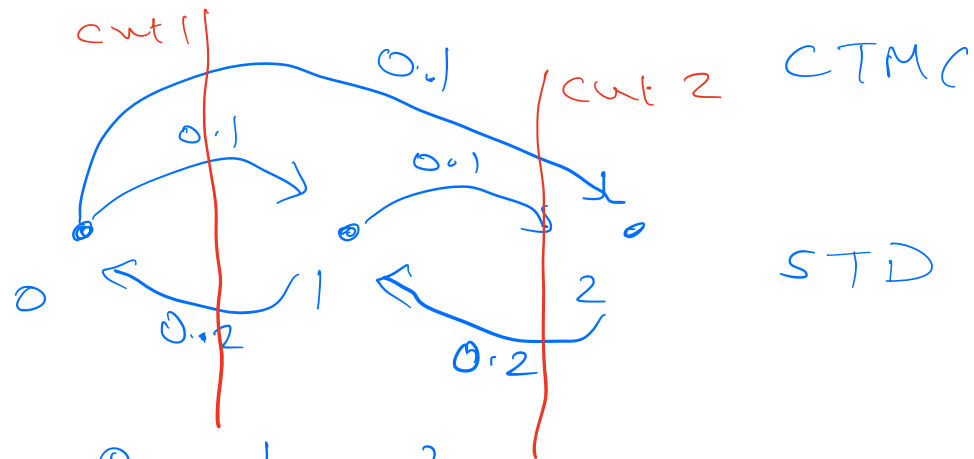
- **Theorem:** An irreducible CTMC has either no or exactly one invariant distribution*. If it is finite, it certainly has exactly one.

- **Theorem:** If the CTMC has exactly one invariant distribution* π , then for any initial distribution,

$$\lim_{t \rightarrow \infty} P(X_t = i) = \pi(i), \text{ for all } i \text{ in } \mathbf{X}, \text{ and}$$

$$\lim_{T \rightarrow \infty} \frac{1}{T} \int_0^T 1\{X_s = i\} ds = \pi(i), \text{ for all } i \text{ in } \mathbf{X}.$$

** True when the CTMC is positive recurrent, where positive/null recurrence and transience are defined similarly as for a DTMC.*



$$Q = \begin{matrix} & \begin{matrix} 0 & 1 & 2 \end{matrix} \\ \begin{matrix} 0 \\ 1 \\ 2 \end{matrix} & \begin{bmatrix} -0.2 & 0.1 & 0.1 \\ 0.2 & -0.3 & 0.1 \\ 0 & 0.2 & -0.2 \end{bmatrix} \end{matrix}$$

From state 1: $\text{Exp}(\overset{\text{rate}}{0.3})$ RV

@ end of timer w.p $\frac{0.1}{0.3} = \frac{1}{3}$ jump to state 2, & w.p $\frac{2}{3}$ jump to state 0.

\Rightarrow Two timers by $\text{Exp}(0.1)$ RV & $\text{Exp}(0.2)$ RV, & whoever wins, jump to that state

$$\pi Q \geq 0$$

$$\begin{bmatrix} \leftarrow \pi \rightarrow \\ \pi(0) \quad \pi(1) \quad \pi(2) \end{bmatrix} \begin{bmatrix} Q \end{bmatrix} = 0$$

$$\left(\text{DTMC} \Rightarrow \pi P = \pi \right)$$

$$\pi(0) \cdot 0.1 + \pi(0) \cdot 0.1 = \pi(1) \cdot 0.2$$

$$\Rightarrow \pi(0) = \pi(1)$$

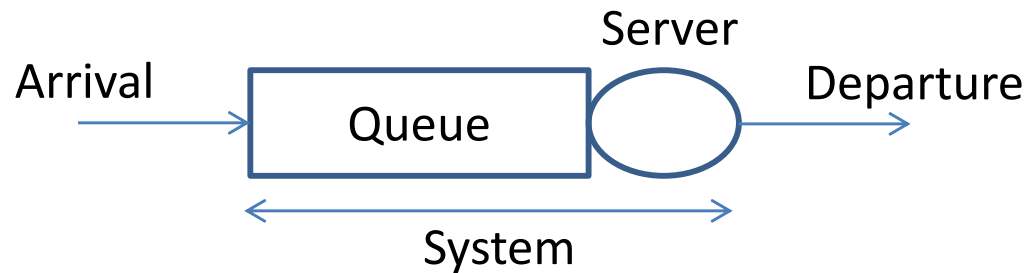
$$\pi(0) \cdot 0.1 + \pi(1) \cdot 0.1 = \pi(2) \cdot 0.2$$

$$\Rightarrow \pi(1) = \pi(2)$$

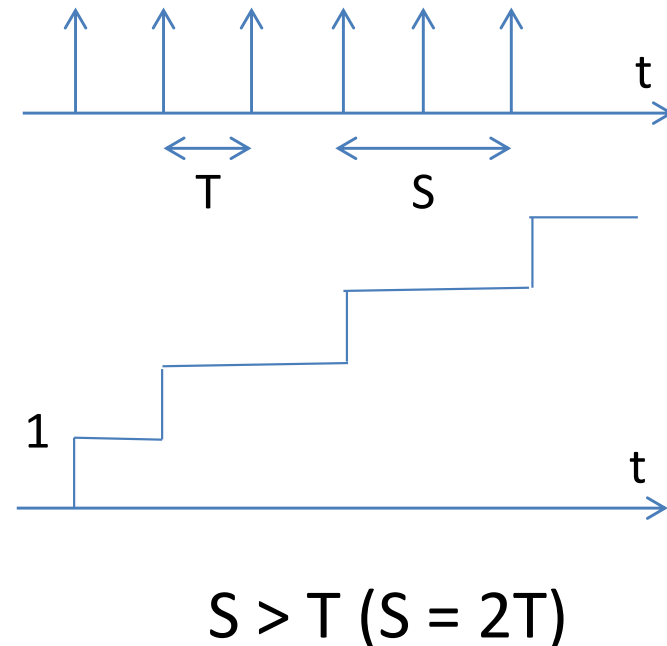
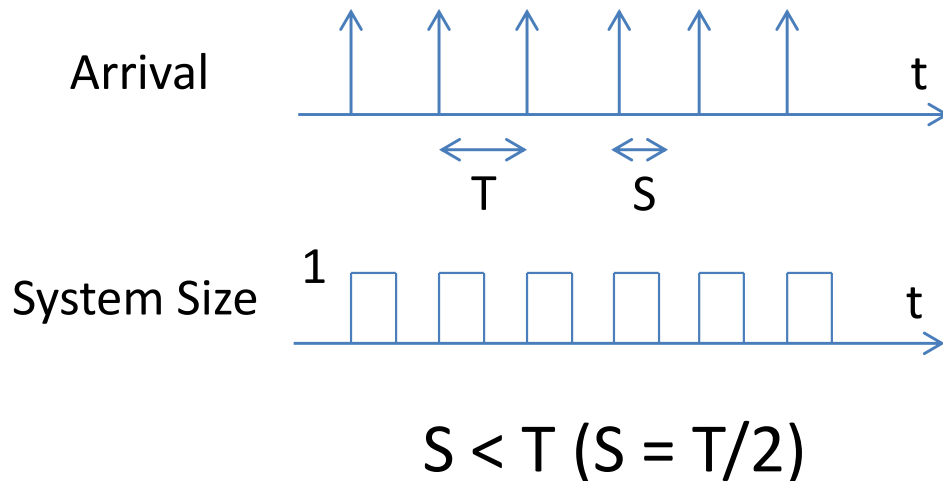
$$\pi(0) + \pi(1) + \pi(2) = 1 \quad \Leftarrow$$

$$\pi(0) = \pi(1) = \pi(2) = \frac{1}{3}$$

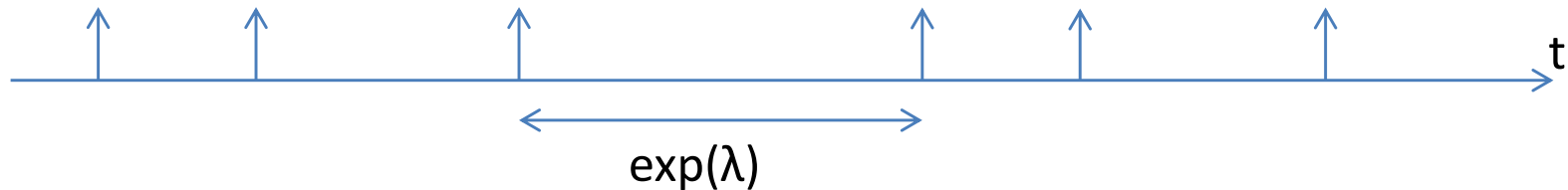
Deterministic Queue



- Consider a queue with a periodic customer (packet) arrival every T seconds, and assume that each customer (packet) requires exactly S seconds of service (transmission) time.



Poisson Process



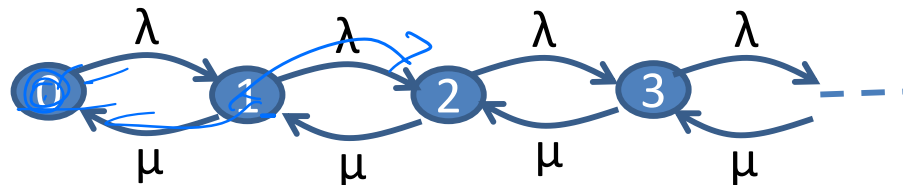
- A Poisson arrival process with rate λ is defined as the process that has independent and identically distributed (iid) inter-arrival times, each with exponential distribution with average of $1/\lambda$.
- **Fact:** Given a Poisson arrival process with rate λ , let $X_0 = 0$, and let $X_t = \#$ of arrivals in $[0, t]$. Then, X_t is a CTMC with $q(i, i) = -\lambda$, and $q(i, i+1) = \lambda$ for $i \geq 0$.
- **Fact:** In an interval of length T , # of arrivals from a Poisson process with rate λ has Poisson distribution with average of λT , i.e.,

$$P(\# \text{ arrivals} = k) = \frac{e^{-\lambda T} (\lambda T)^k}{k!}, \quad k = 0, 1, 2, \dots$$

- **Fact:** A Poisson process has independent increments, i.e., # of arrivals in disjoint intervals of lengths T_1 and T_2 are independently distributed Poisson random variables with average values of λT_1 and λT_2 , respectively.
- **Fact:** If X_i , $i = 1, 2$ are independent Poisson random variables with average values λ_1 and λ_2 , then $X_1 + X_2$ is Poisson random variable with average value of $\lambda_1 + \lambda_2$.
- Poisson process is a good tractable model for baseline evaluation – particularly good when modeling superposition of multiple packet streams.

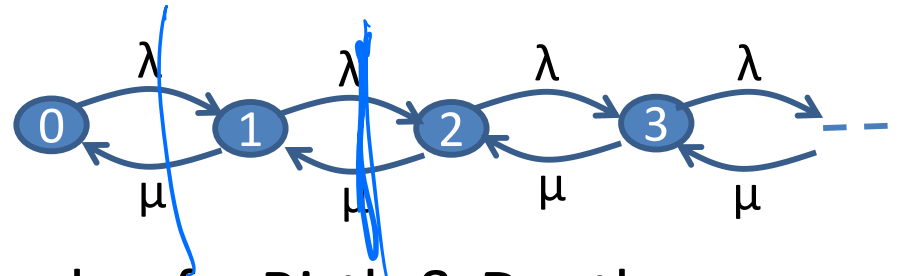
M/M/1 Queue

- First M means the memoryless Poisson process for arrivals, second M means memoryless exponential distribution for service times, and 1 indicates single server.
- Let X_t denote the # of packets/customers in the system.
- X_t can be modeled as a CTMC with the rate matrix Q shown below pictorially.
- Here λ is the Poisson arrival rate and $1/\mu$ is the average service time. **Assume $\lambda < \mu$.**



STD

M/M/1 Queue (2)



$$\lambda \pi(1) = \mu \pi(2)$$

$$\lambda \pi(n) = \mu \pi(n+1) \quad n \geq 0$$

- This is an example of a Birth & Death process.
- **Fact:** If X_1 and X_2 are independent and exponentially distributed with the rates λ_1 and λ_2 , respectively, then $\min\{X_1, X_2\}$ is exponentially distributed with the rate $(\lambda_1 + \lambda_2)$, and $\min\{X_1, X_2\} = X_i$ with probability $\lambda_i / (\lambda_1 + \lambda_2)$, $i = 1, 2$.
- Rate Matrix:

$$Q = \begin{pmatrix} -\lambda & \lambda & 0 & 0 & \dots \\ \mu & -(\lambda + \mu) & \lambda & 0 & \dots \\ 0 & \mu & -(\lambda + \mu) & \lambda & \dots \\ \dots & \dots & \dots & \dots & \dots \end{pmatrix} \leftarrow$$

M/M/1 Queue (3)

- For finding the invariant distribution π , we solve the balance equations assuming $\lambda < \mu$:


$$\sum_{i \in \mathbf{X}} \pi(i)q(i, j) = 0 \text{ for all } j \in \mathbf{X} \text{ (or } \pi Q = 0).$$

- This leads to

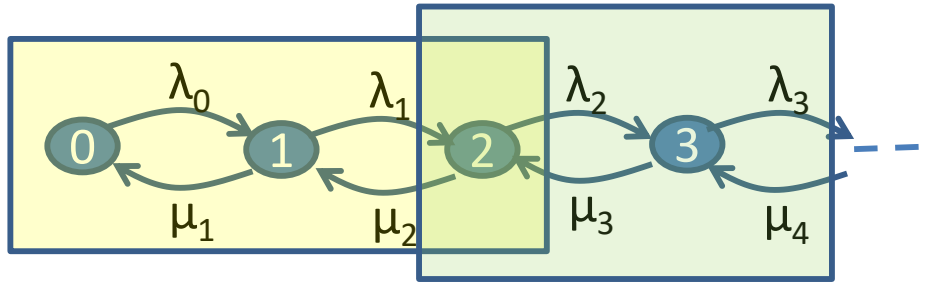
$$\pi(0)\lambda = \pi(1)\mu, \text{ and}$$

$$\pi(n)(\lambda + \mu) = \pi(n-1)\lambda + \pi(n+1)\mu, \text{ for } n \geq 1.$$

- Let $\rho = \lambda/\mu$. If $\rho < 1$, the unique solution for these equations is

$$\pi(n) = (1-\rho)\rho^n, n \geq 0.$$


Birth & Death Process



- Balance equations are valid for any closed set.
- For the set enclosed in green rectangle, we have

$$\pi(1)\lambda_1 + \pi(4)\mu_4 = \pi(3)\lambda_3 + \pi(2)\mu_2, \text{ or}$$
$$\pi(1)\lambda_1 - \pi(2)\mu_2 = -\pi(3)\lambda_3 + \pi(4)\mu_4$$

- Similarly, for the yellow rectangle, we have

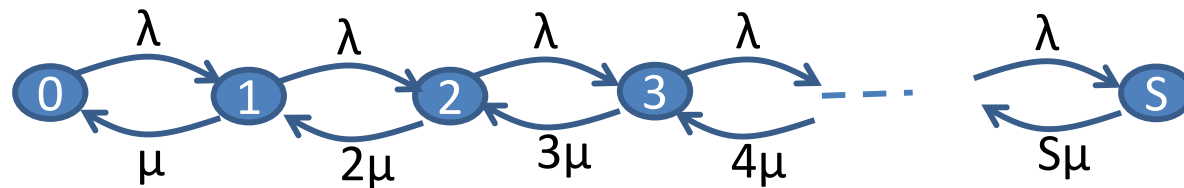
$$\pi(2)\lambda_2 = \pi(3)\mu_3.$$

- Using the latter recursively, we get

$$\pi(k) = \frac{\lambda_0 \lambda_1 \lambda_2 \dots \lambda_{k-1}}{\mu_1 \mu_2 \mu_3 \dots \mu_k} \pi(0).$$

Blocking Probability

- Consider M/M/S/S queue where the first S denotes # of servers, and second S denotes the max # of packets/customers/calls in the system.
- I.e., at an arrival, if all servers are busy, the arrival is dropped.



Blocking Probability

- Rate matrix is given by

$$Q = \begin{pmatrix} -\lambda & \lambda & 0 & 0 & \dots \\ \mu & -(\lambda + \mu) & \lambda & 0 & \dots \\ 0 & 2\mu & -(\lambda + 2\mu) & \lambda & \dots \\ \dots & \dots & \dots & \dots & \dots \end{pmatrix}$$

- The balance equations are:

$$\pi(0)\lambda = \pi(1)\mu, \text{ and}$$

$$\pi(n)(\lambda + n\mu) = \pi(n-1)\lambda + \pi(n+1)(n+1)\mu, \text{ for } n \geq 1.$$

- This leads to

- Blocking Probability = $\pi(S) = \frac{\rho^S/S!}{\sum_{n=0}^S \rho^n/n!}$, where $\rho = \frac{\lambda}{\mu}$.

- Note use of PASTA in above.

- This is referred to as the Erlang Loss or Erlang B formula.

- **Insensitivity:** For any service time distribution, with average service time = $1/\mu$, the above result is true. Remarkable!!!

Impact of General Service Time

- Service time variability increases delay.
 - Intuition: Barrier at $X_t = 0$, doesn't let the queue build-up get evened out.
- Consider an M/G/1 queue with average service time $1/\mu$ and variance σ^2 .
 - Recall coefficient of variation C is defined by
$$C^2 = \text{variance}/(\text{mean})^2 = \sigma^2 \mu^2.$$
 - Average queueing (waiting) time W is given by
$$W = \frac{(1 + C^2)}{2} \frac{\rho}{(1 - \rho)} \frac{1}{\mu} \text{ (Pollaczek – Khinchin Mean Value Formula)}$$
 - Observe that $C^2 = 1$ and 0 for exponential and deterministic service times, respectively.
 - For M/M/1, we have $W = \frac{\rho}{(1 - \rho)} \frac{1}{\mu}$.
 - For M/D/1, we have $W = \frac{\rho}{2(1 - \rho)} \frac{1}{\mu}$.