8. Intelligent Transportation: Dynamic Routing

金力 Li Jin
li.jin@sjtu.edu.cn
上海交通大学密西根学院
Shanghai Jiao Tong University UM Joint Institute

上海交通大学
SHANGHAI JIAO TONG UNIVERSITY

Recap

- Smart highways
  - Sensing technology
  - Control technology
- Traffic flow model
  - Flow-density relation
  - Cell transmission model
- Ramp metering
  - Flow stabilization
  - Throughput maximization
  - Delay minimization

Outline

- Background
  - Static & dynamic routing
  - Applications
- Single & parallel queues
  - Arrival process & service processes
  - Single queue
  - Parallel queues
- Queuing networks
  - Model
  - Bernoulli routing
  - Dynamic routing

Background: static routing

- Data:
  - Demand
  - Cost function
  - Capacity
- Decision variables:
  - Link flows
- Constraints:
  - Mass conservation
  - Link capacity
- Objective:
  - Minimize total flow cost

Background: static vs. dynamic routing

Static routing

- At each diverge, traffic assigned to downstream links with time-invariant fractions.
- Only depends on model data (demand & capacity)
- Independent of real-time traffic condition (open-loop)
- Not responsive to disruptions

Dynamic routing

- At each diverge, traffic assigned with time-variant fractions.
- Depends on both model data and real-time traffic condition (closed-loop)
- Can respond to disruptions
- Requires real-time sensing capabilities

Background: dynamic routing

- Data:
  - Demand, Cost function, Capacity
  - Real-time traffic state (e.g. queue size)
- Decision variables:
  - Routing for each vehicle/customer/job
- Constraints:
  - Mass conservation
  - Link capacity
- Objective:
  - Minimize total travel cost, typically starting from current state

Vehicle routing

- A vehicle starts its trip from origin to destination
- Baidu Map or AMAP suggests multiple possible routes
- Estimated travel time on each route is predicted
- Typically vehicles select the fastest route
- Such routing is responsive to traffic congestion & traffic incidents
- Dynamic routing!

Customer routing

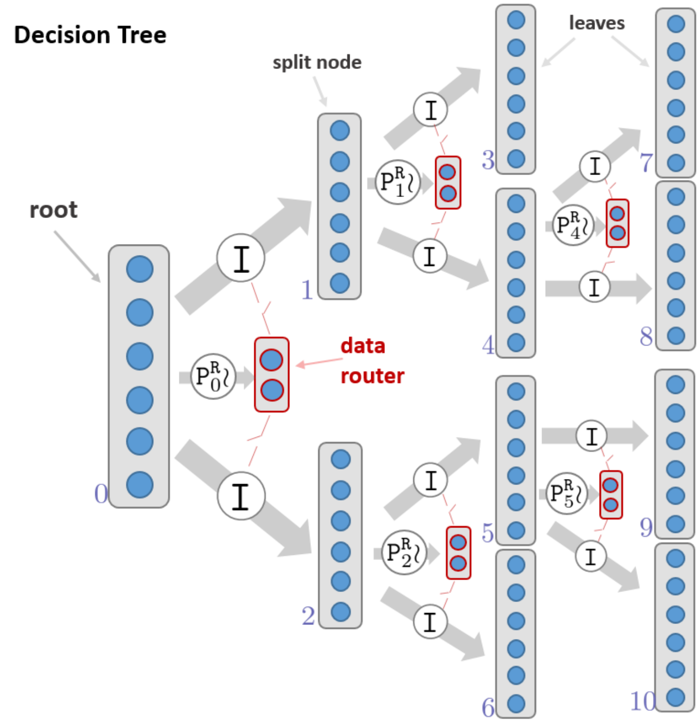
- Suppose that a supermarket has multiple cashiers
- Customers wait in separate queues for the cashiers
- When a customer arrives, he/she selects the shortest queue to join ("JSQ" policy)
- Or, joining the queue with the least items, one customer buying one week's supply vs. two customers buying two drinks

Air traffic management

- Air routes connecting airports
- Traffic flow on each route
- Traffic on different route may interact within sectors
- Sector can get congested
- Each flight needs to determine the path, i.e. sequence of sectors

Data job routing

- Jobs received by a router
- Router assigns each job to a server
- Jobs processed by a server is allocated to a further downstream server
- JSQ: a router always allocate an incoming job to the least busy server, i.e. the server with the shortest job queue



Outline

- Background
  - Static & dynamic routing
  - Applications
- Single & parallel queues
  - Arrival process & service processes
  - Single queue
  - Parallel queues
- Queuing networks
  - Model
  - Bernoulli routing
  - Dynamic routing

Queuing node model

- A node contains a server and a queuing space
- State $X(t)$ = # of customers waiting or being served at the node
- If $X(t) = 3$, then 2 customers are waiting and 1 being served.
- For ease of presentation, we consider discrete time

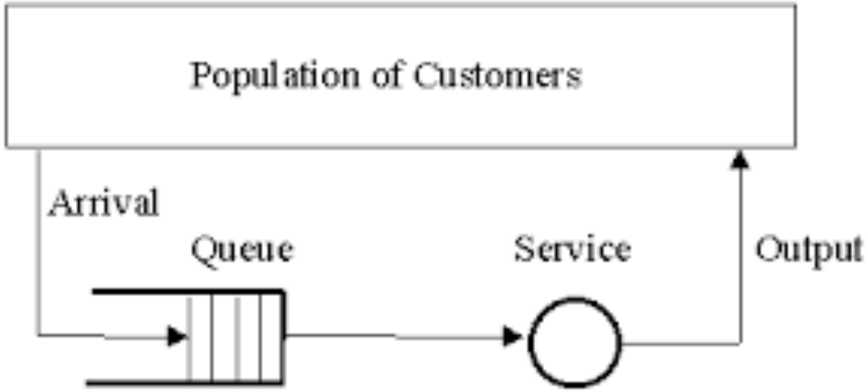


Figure 1

Arrival process

- We consider a stochastic arrival process at node i
- Bernoulli process:
  - A customer arrives at node i during one time step with probability $\lambda \in [0,1]$
  - No customers arrive at node i with probability $1-\lambda$
- Expected # of arrivals per time step = λ (demand)
- Probability mass function (PMF) for inter-arrival time

$$p_U(u) = (1-\lambda)^{u-1}\lambda, u \in \mathbb{Z}_{>0}.$$

- Expected inter-arrival time = $1/\lambda$.
- Distribution over T time steps:

$$p_N(n) = \binom{T}{n}\lambda^n(1-\lambda)^{T-n} = C_n^T\lambda^n(1-\lambda)^{T-n}$$

for $N = 0,1,\dots,T$

- Upon arrival, a customer
  - enters the server and begins its service if the server is empty
  - joins the queue and waits otherwise

Service process

- We consider a probabilistic service process:
  - Suppose that a customer is being served at time t
  - The customer finishes service and leaves the current node at time $t + 1$ with probability μ
  - The customer stays in the node and continues its service with probability $1 - \mu$
- As soon as a customer finishes service, the next customer enters the server and begins its service
- Otherwise, the subsequent customers have to continue waiting
- PMF for service time $p_V(v) = (1 - \mu)^{v-1}\mu$, $v \in \mathbb{Z}_{>0}$.
- Expected service time = $1/\mu$.

A single-node queuing system

- Consider an isolated node
- A single node with
  - Bernoulli arrivals with rate λ
  - Probabilistic service with rate μ
- State of the system: $X[t]$ = queue size at time t
- Given $X[t]$, we can predict the distributions for $X[t+1], X[t+2]$, ...

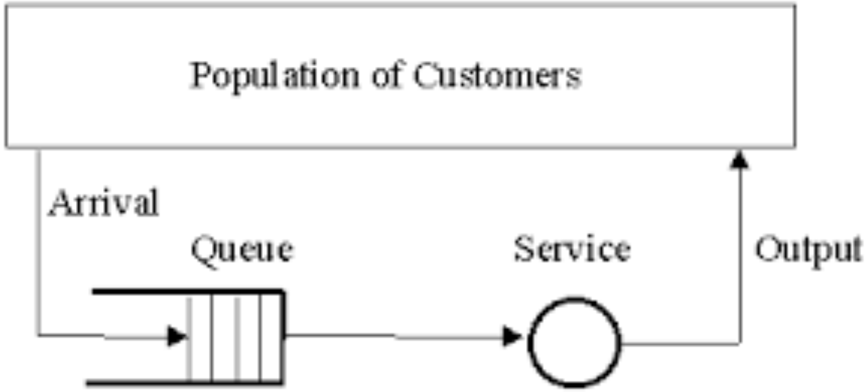


Figure 1

A single-node queuing system

- State transition probabilities (system dynamics)
- Suppose $X[t] = x > 0$

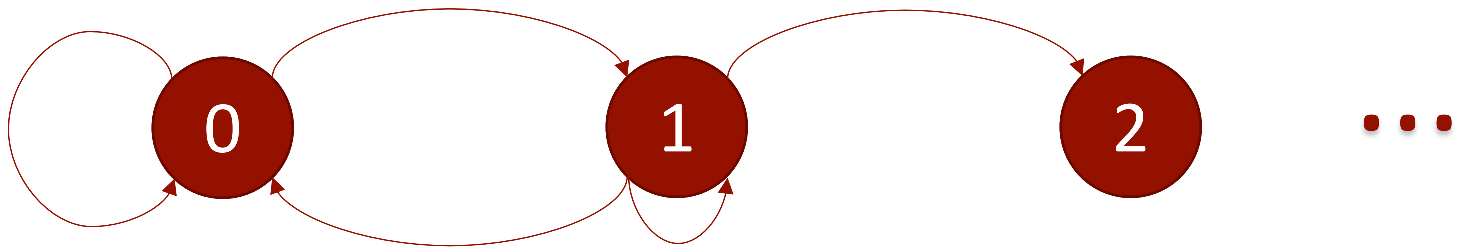
$$\Pr\{X[t+1] = x-1\} = \mu(1-\lambda)$$
$$\Pr\{X[t+1] = x+1\} = \lambda(1-\mu)$$
$$\Pr\{X[t+1] = x\} = (1-\lambda)(1-\mu) + \lambda\mu$$

- Interpretation of the three formulae?
- Suppose $X[t] = 0$

$$\Pr\{X[t+1] = 0\} = 1-\lambda$$
$$\Pr\{X[t+1] = 1\} = \lambda$$

- Can you draw the state transition diagram?

A single-node queuing system



A single-node queuing system

- Of particular interest is the steady-state behavior of the queuing system.
- Simplistically, steady-state behavior ≈ long time average. (*Ergodicity*)
- The most important performance metric for a queuing system is the steady-state or long time-average queue size

$$\bar{X} = \lim_{t\to\infty} \frac{1}{t} \sum_{s=0}^{t} X[s] = ?$$

- We say that the queuing system is
  - stable or convergent if $\bar{X} < \infty$
  - unstable if $\bar{X} = \infty$
- Can you make a scientific guess when is the queue stable?

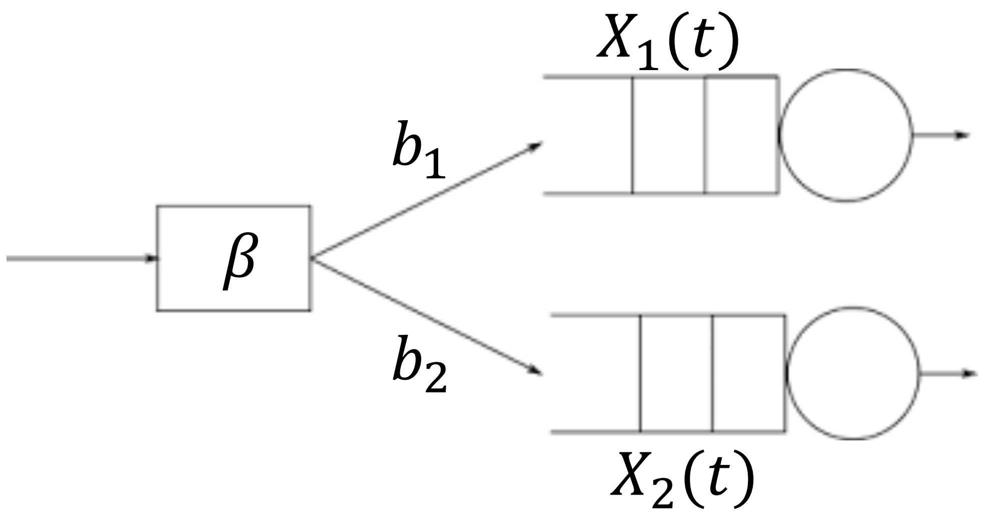
[Not required] A single-node queuing system

- Let p_{ij} be the transition probabilities, i.e.
$$p_{ij} = \Pr\{X(t+1) = j | X(t) = i\}$$
- Let π_i be the steady-state probabilities, i.e.
$$\lim_{t\to\infty} \Pr\{X(t) = i\} = \pi_i$$
- The steady-state equations for the queuing system is
$$p_{01}\pi_0 = p_{10}\pi_1$$
$$(p_{i-1,i} + p_{i,i+1})\pi_i = p_{i-1,i}\pi_{i-1} + p_{i+1,i}\pi_{i+1}, \qquad i = 1,2,\dots$$
$$\pi_0 + \pi_1 + \pi_2 + \cdots = 1.$$
- Then we can obtain π_i by solving the above equations
- $\bar{X} = \sum_{x=0}^{\infty} \pi_x x$
- Can you derive π_x?

Routing for parallel queues

- Customers arrive at a router with rate λ
- Router assigns customer to one of two parallel queues
- State: $X[t] = [X_1[t] X_2[t]]^T \in \mathbb{Z}^2_{\geq 0}$
- Routing policy:

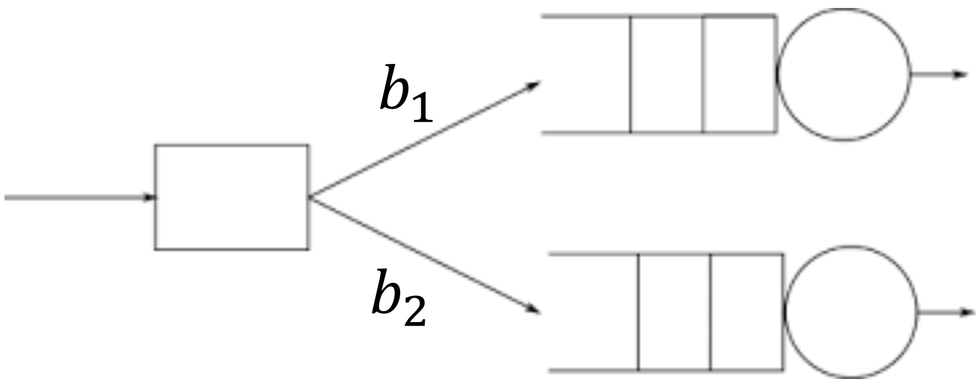
$$\beta : \mathbb{Z}^2_{\geq 0} \to [0,1]^2 \text{ or } \beta : \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \mapsto \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \text{ s.t. } b_1 + b_2 = 1.$$



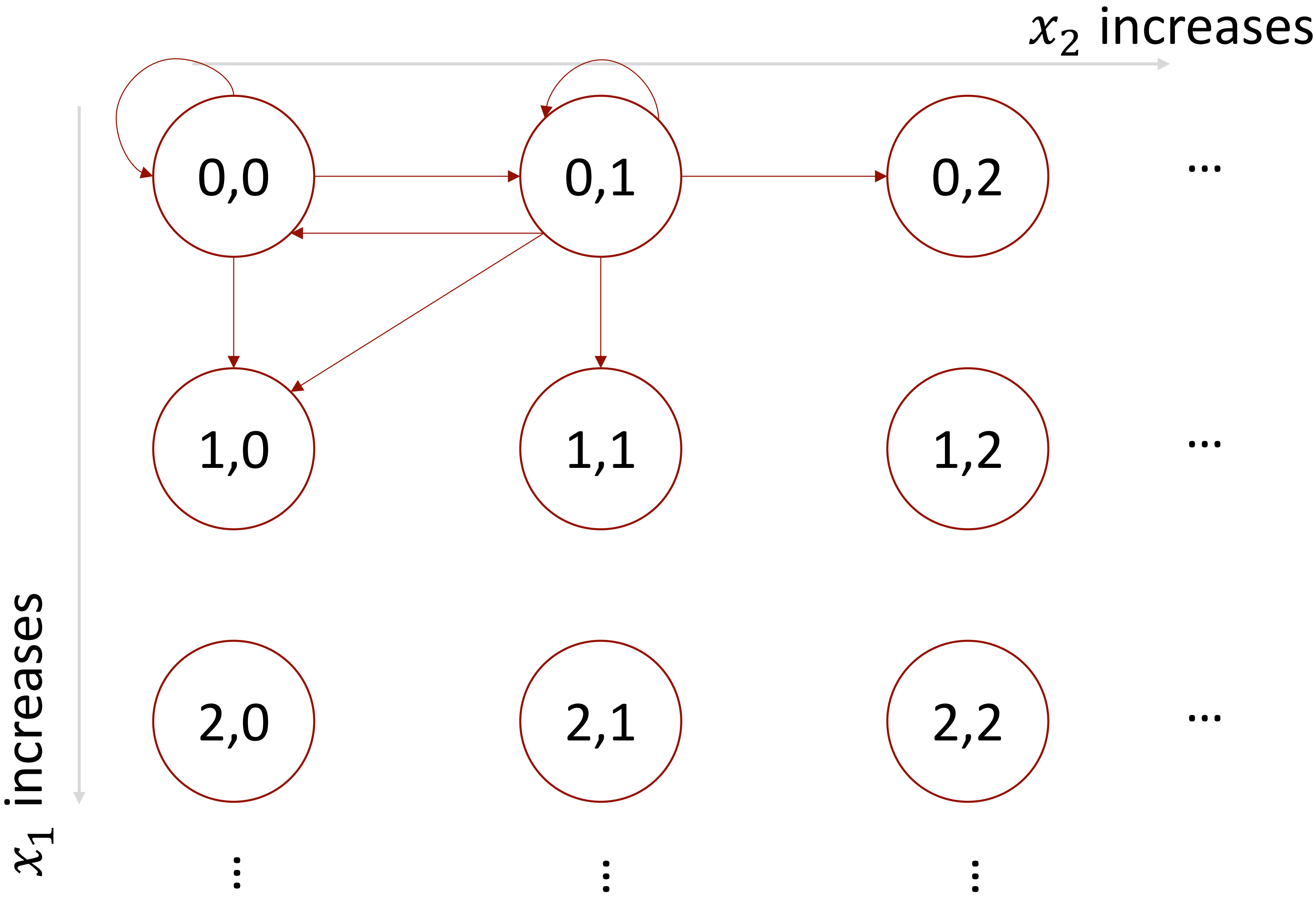
Bernoulli (static) routing:
$\beta(x)$ independent of x.
Dynamic routing:
$\beta(x)$ dependent of x

Bernoulli routing

- When a customer arrives, it is assigned to queue 1 with probability b_1 and queue 2 with probability b_2, respectively.
- $b_1 \in [0,1]$, $b_2 \in [0,1]$, $b_1 + b_2 = 1$
- Open-loop routing
- Independent of $X(t)$
- State transition diagram?

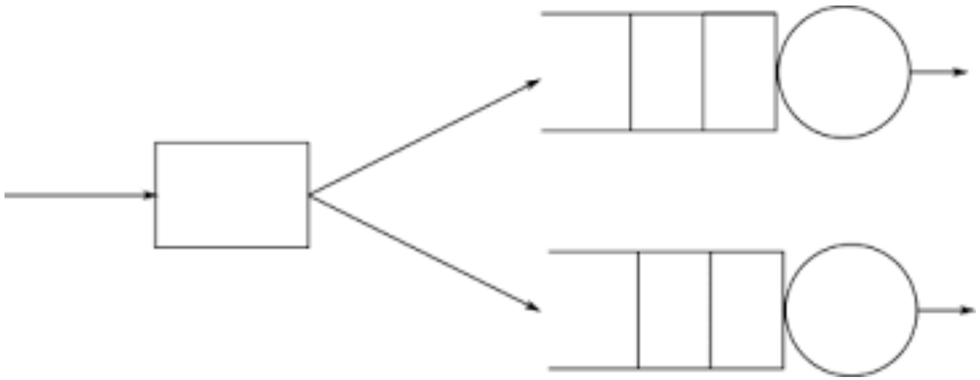


State-transition diagram (Bernoulli routing)



Dynamic routing: parallel queues

- Now suppose that we can route an incoming customer according to the real-time traffic state $X(t)$
- What is a good way of routing?
- Intuition:
  - If a queue is long, then do not add the customer to it.
  - If a queue is short, then it is OK to add the customer to it.
- ”Join the shortest queue” (JSQ) policy
- State transition diagram?



State-transition diagram (JSQ routing)

Dynamic routing: parallel queues

- JSQ policy:

$$\beta(x) = \begin{cases} [1\ 0]^T & x_1 < x_2 \\ [0\ 1]^T & x_1 > x_2 \\ ? & x_1 = x_2 \end{cases}$$

- Ties can be broken uniformly at random.
- That is, when $x_1 = x_2$, we set

$$\beta(x) = \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix}$$

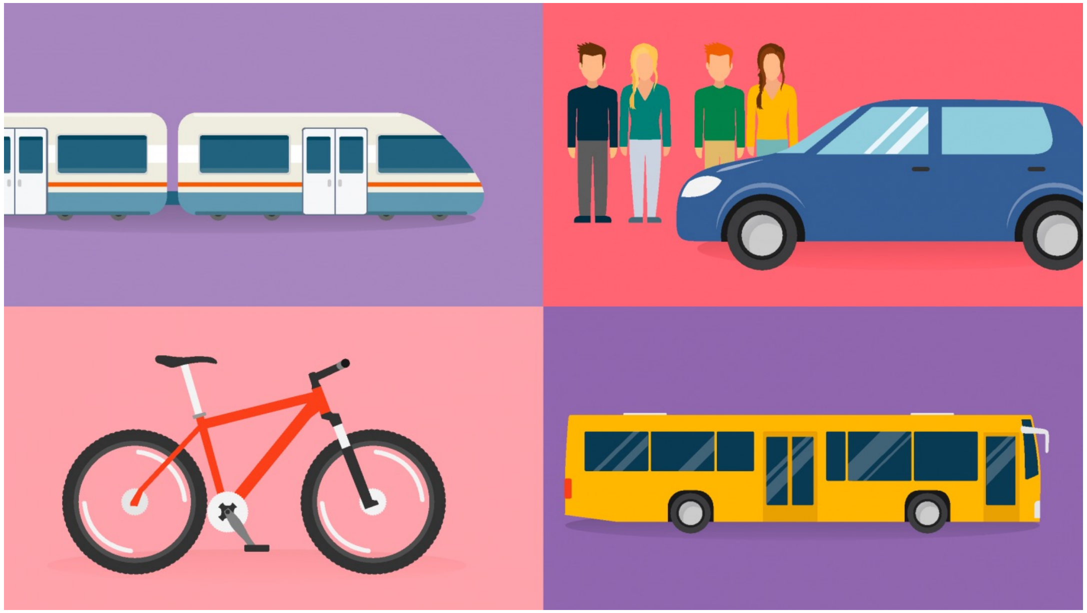
- Can adapt to randomness
- Can adapt to disruptions (e.g. server malfunction)

How routing actually works in transportation?

- There are two notions:
  1. What you want the travelers to do?
  2. How travelers would respond to your instruction?
- Usually, we are not able to force travelers to take a certain route...
- Instead, we can analyze travelers’ behavior and incentivize/encourage travelers to do a favored action.
- Incentivize = pay them somehow...
- Theoretical foundation: discrete choice theory

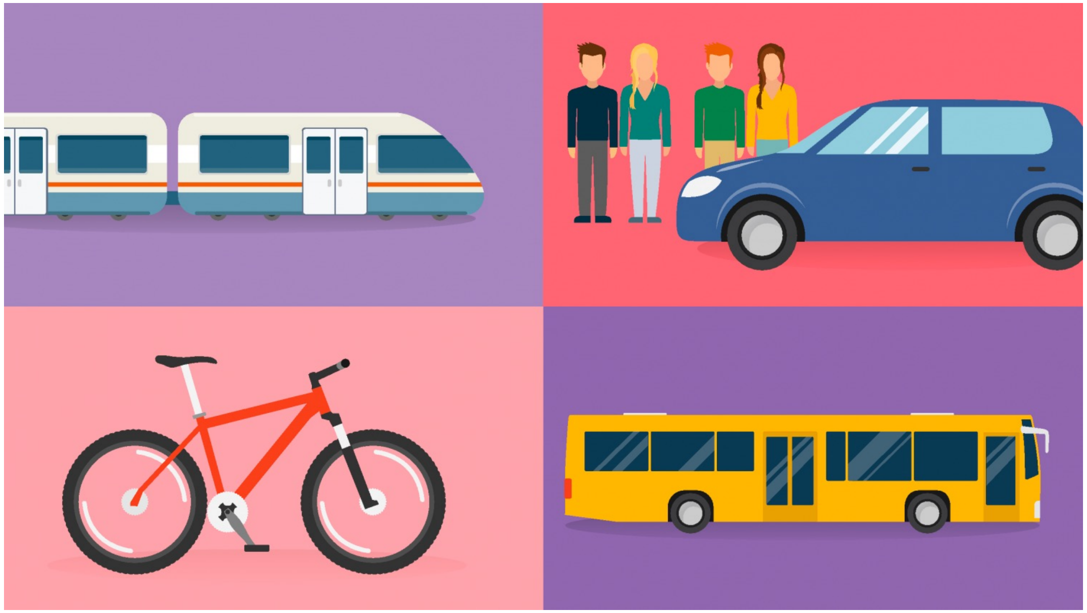
Discrete choice

- Customer chooses from K options
- A commuter chooses subway or bus
- A truck driver chooses departure time
- A driver chooses tolled or free roads



Discrete choice model

- Input: attributes of options
- Output: probability of choosing an option
- Tool: logistic regression



Logistic regression

- Logit function

$$\Pr\{G = k | X = x\} = \frac{\exp\left(\beta_{k0} + \beta_k^T x\right)}{\sum_{l=1}^{K} \exp(\beta_{l0} + \beta_l^T x)}$$

- Classification

$$G(x) = \arg\max_k \frac{\exp\left(\beta_{k0} + \beta_k^T x\right)}{\sum_{l=1}^{K} \exp(\beta_{l0} + \beta_l^T x)}$$

- Process of fitting coefficients β_{ki}: called logistic regression
- Use maximum likelihood

Basic discrete-choice model

- Utility: a quantification of customers' preferences
- Example 1: price of a product
- Example 2: price-to-quality ratio
- Example 3: travel time plus toll (value of time)
- Example 4: price plus comfort level
- Let $V_{i,n}$ be the nth customer's utility of the ith option
- The probability of choosing i is

$$P_{i,n} = \frac{\exp(V_{i,n})}{\sum_{m=1}^{K} \exp(V_{i,m})}$$

- High utility -> high probability of being chosen

Utility function

- In smart city settings, by far the most common form of utility function is linear

$$V_{i,n} = \sum_{k=1}^{K} \beta_{n,k} x_{i,k}$$

- For example, travel mode estimation

$$V_{subway} = b_0 + b_1x(\text{travel time}) + b_2x(\text{fare}) + b_3x(\text{crowdness})$$

- Signs of the coefficients?

Interpretation

- Coefficients $\beta_{i,k}$: quantifies the i-th customer's preferences with respect to the k-th feature
- $\beta_{i,k} > 0$ -> ?
- $\beta_{i,k} < 0$ -> ?
- Magnitude of $\beta_{i,k}$ captures the sensitivity

$$V_{subway} = b_0 + b_1 x(\text{travel time}) + b_2 x(\text{fare}) + b_3 x(\text{crowdness})$$

Estimation

- Use maximum likelihood
- Suppose we have observation for N customers
- Both features and their actual choices are recorded
- Probability that customer i chooses option n_i

$$P_{i,n_i} = \frac{\exp(\sum_{k=1}^{K} \beta_{n_i,k} x_{i,k})}{\sum_{n=1}^{K} \exp(\sum_{k=1}^{K} \beta_{n,k} x_{i,k})}$$

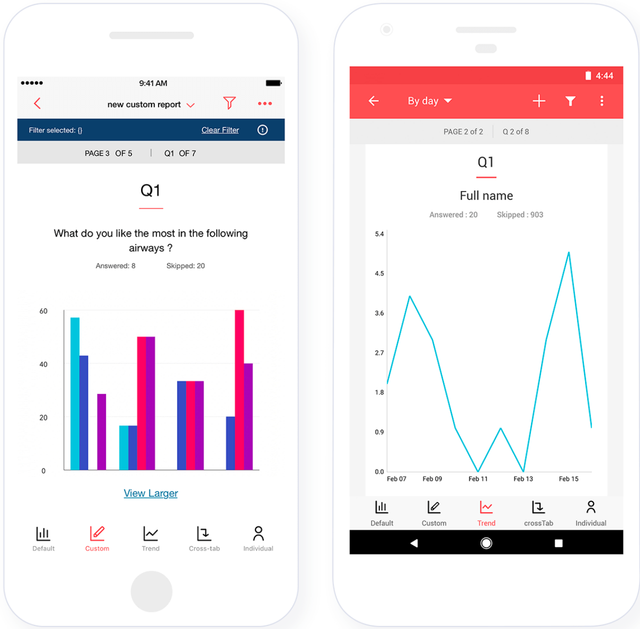
- Likelihood of their choices

$$lik(B) = \prod_{i=1}^{N} P_{i,n_i}(B)$$

- MLE: $\frac{\partial}{\partial \beta_{n,k}} lik(B) = 0$ for all n and for all k

Data

- Conventionally obtained from surveys
- Demographic information, technical and/or economical metrics, geographical information, etc.
- Modern ways of obtaining data
  - App-based surveys
  - “Big Data”
  - MTA trip records
  - Uber trip records
  - Airlines trip records…
- Important: people are increasingly concerned with efficiency vs. privacy



Mode choice

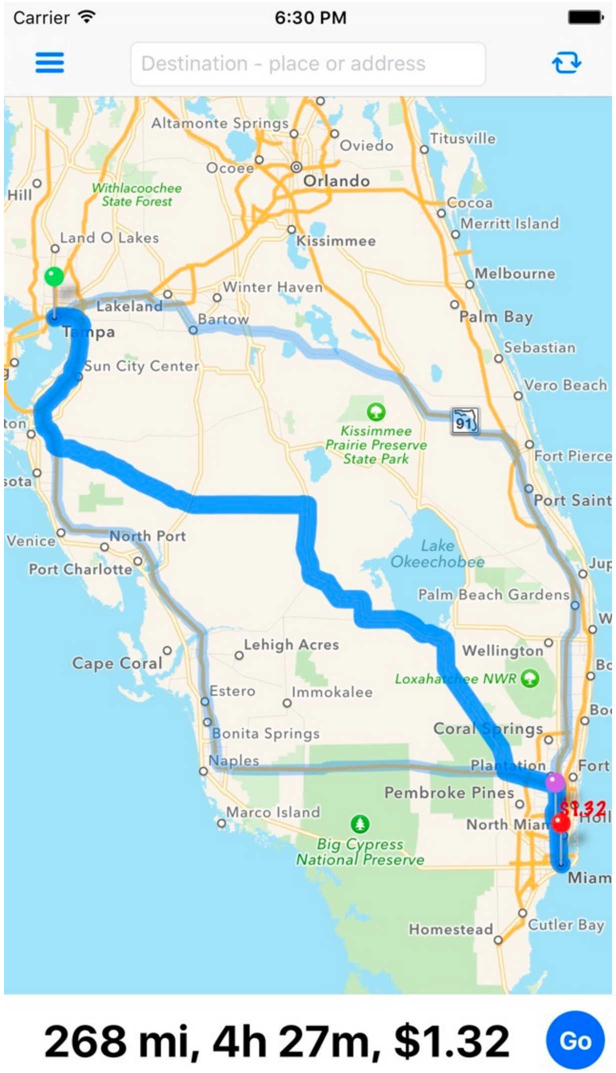
- Objective: develop a model explaining automobile ownership and commuting mode
- Application: justification for the Bay Area Rapid Transit (BART)
- Survey data
- $V=-0.0412c/w-0.0201T-0.0531T^0-0.89D^1-1.78D^3-2.15D^4$
- c=round-trip cost ($)
- w=passenger wage rate ($/min)
- T=in-vehicle travel time (min)
- T^0=out-of-vehicle time (min)
- D=alternative-specific dummies

Trip scheduling

- From a city manager's perspective, we prefer to balance the trip schedules of commuters rather than concentrate them in short peak hours
- Objective: understand how people schedule trips
- V=-0.106T-0.065SDE-0.254SDL-0.58DL
- T=trip time
- SDE=schedule delay early
- SDL=schedule delay late
- DL=late dummy

Route choice

- How do people select between toll and free routes?
- Important for setting congestion pricing
- $V=-0.862D^{tag}+0.0239Inc(D^{tag})-0.766ForLang(D^{tag})-0.789D^{3}-0.357c-0.109T-0.159R+0.074Male(R)+\text{other terms}$
- D^{tag}=alternative-specific dummies
- Inc=annual income
- ForLang=foreign language
- c=toll, T=travel time
- R=reliability



Outline

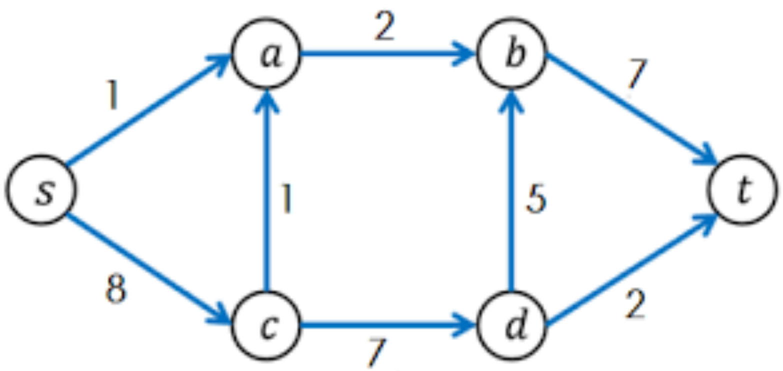
- Background
  - Static & dynamic routing
  - Applications
- Single & parallel queues
  - Arrival process & service processes
  - Single queue
  - Parallel queues
- Queuing networks
  - Model
  - Bernoulli routing
  - Dynamic routing

Network model

- Consider a network with nodes N and links E
- We use integers to label nodes
  - Node 1, 2,…
- We use pairs of integers to label links
  - Link (1,2), (2,3),…
- Directed link (i,j)
- A “customer” (vehicle/passenger/job) enters the network via an origin node (O) and exits via a destination node (D)
- OD is predefined, but route has to be determined in real time.
- Route = a sequence of nodes

Multi-class queuing network

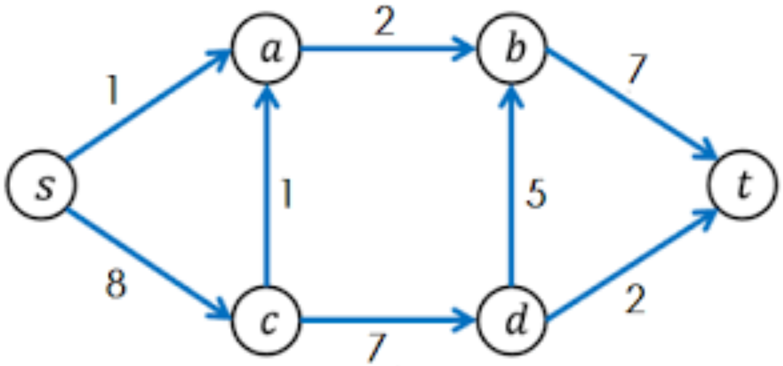
- Can we treat each customer in the same way?



- We classify customers according to their OD
- A set of classes (ODs) C
- Each class $c \in C$ has an origin o_c and a destination d_c
- Class-c arrival rate: $\lambda_c > 0$ at o_c
- Note: a node can be the origin/destination of multiple classes

Multi-class queuing network

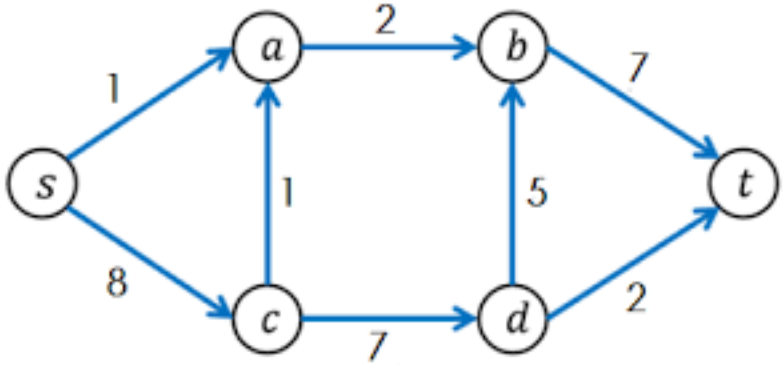
- System state: $X_{ij}^{c}(t),\ (i,j) \in E, c \in C$



- Compact notation: $X(t) = \left[X_{ij}^{c}(t)\right]_{(i,j)\in E, c\in C}$
- State space (set of states) $\mathcal{X} = \mathbb{Z}_{\geq 0}^{|E|\times|C|}$
- In general, service rate μ_i can also depend on class
- But we do not consider such complication in this lecture

Bernoulli routing

- When a customer leaves a server, it goes randomly to a downstream server with time-invariant probabilities



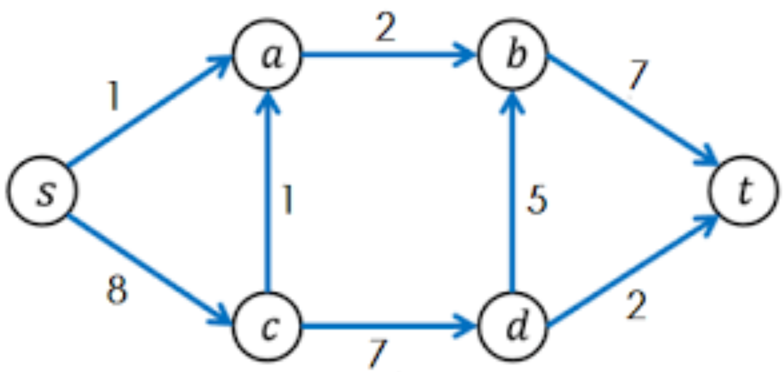
- For a node i with downstream nodes $Out(i;c)$ for class-c traffic, the routing probabilities are p_{ij}^c such that

$$p_{ij}^c \in [0,1] \text{ for all } j \in Out(i;c)$$

$$\sum_{j\in Out(i;c)} p_{ij}^c = 1$$

Dynamic routing

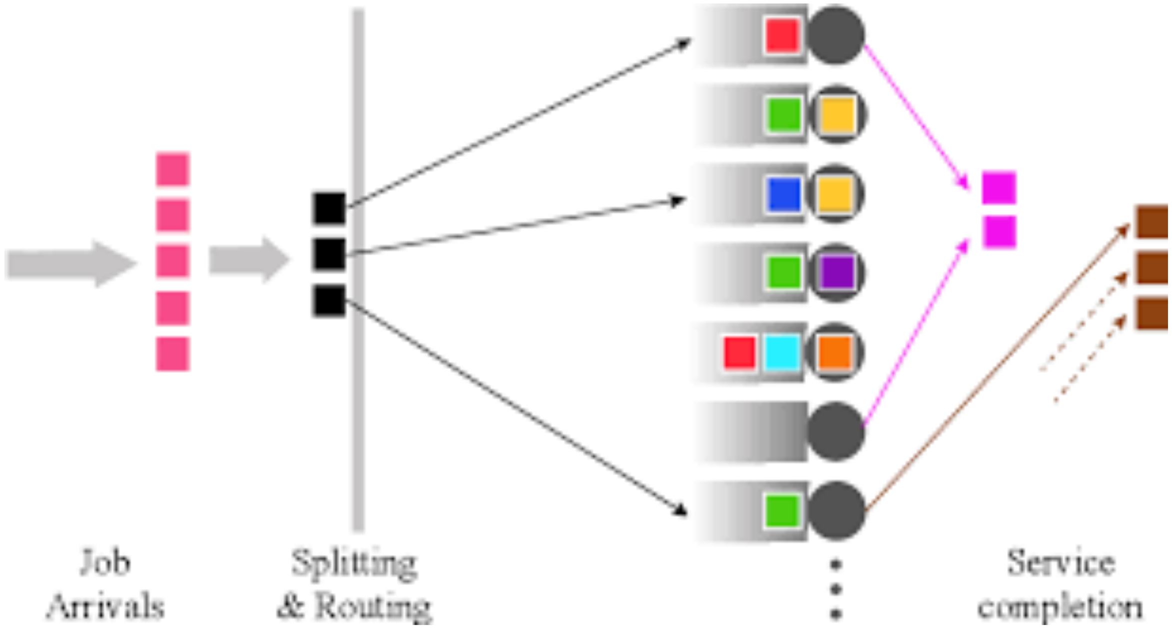
- When a customer leaves a node, its routing decision depends on real-time traffic conditions



- That is, the routing probabilities are functions of the traffic state, i.e. $\beta_{ij}^{c}: \mathcal{X} \to [0,1]$
- This is feedback control.
- Also a Markov decision process.

Does JSQ work on networks?

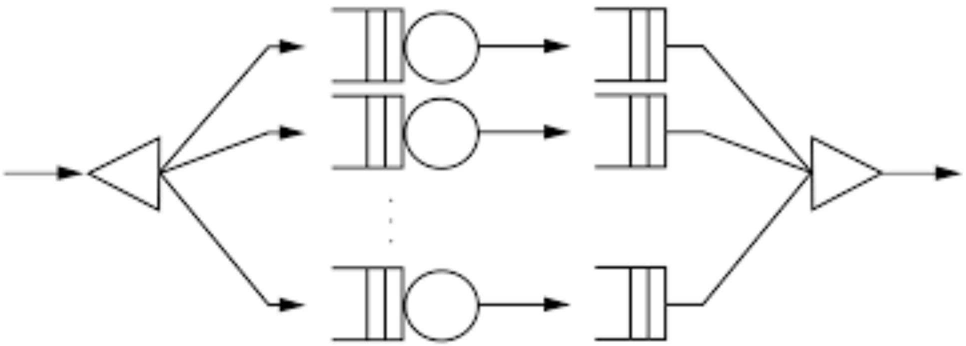
- Recall for two parallel queues, we can use the JSQ policy
- Now, suppose n parallel queues



- JSQ still works: stabilizing iff $\lambda < \sum_i \mu_i$

Does JSQ work on networks?

- However, JSQ can fail on more complex networks…



- Since JSQ is a localized routing policy, it cannot address further downstream congestions
- Fortunately, we can extend JSQ in a network setting
- "Join the shortest route" (JSR): when a customer enters the network, it selects the route with the minimal total # of customers thereon.

MDP formulation

- Consider a network with nodes N and edges E.
- Node $i \in N$, link $(i, j) \in E$ (i.e., from i to j).
- Consider a single origin-destination pair.
- State: traffic state on each link $X_{ij}[t] \in \mathbb{Z}_{\geq 0}$ for all $(i, j) \in E$.
- Action: routing destination at each node $A_{ij}[t] \in \mathrm{Out}(i, j)$ for all $(i, j) \in E$, where $\mathrm{Out}(i, j)$ is the set of downstream links to (i, j).
- Dynamics: the transition probability

$$p(x'|x, a) \qquad \text{for all } x, a, x'.$$

Summary

- Background
  - Static & dynamic routing
  - Applications
- Single & parallel queues
  - Arrival process & service processes
  - Single queue
  - Parallel queues
- Queuing networks
  - Model
  - Bernoulli routing
  - Dynamic routing