CSCE 619-600
Networks and Distributed Processing
Spring 2017

Markov Chains IV

Dmitri Loguinov
Texas A&M University

March 7, 2017

Agenda

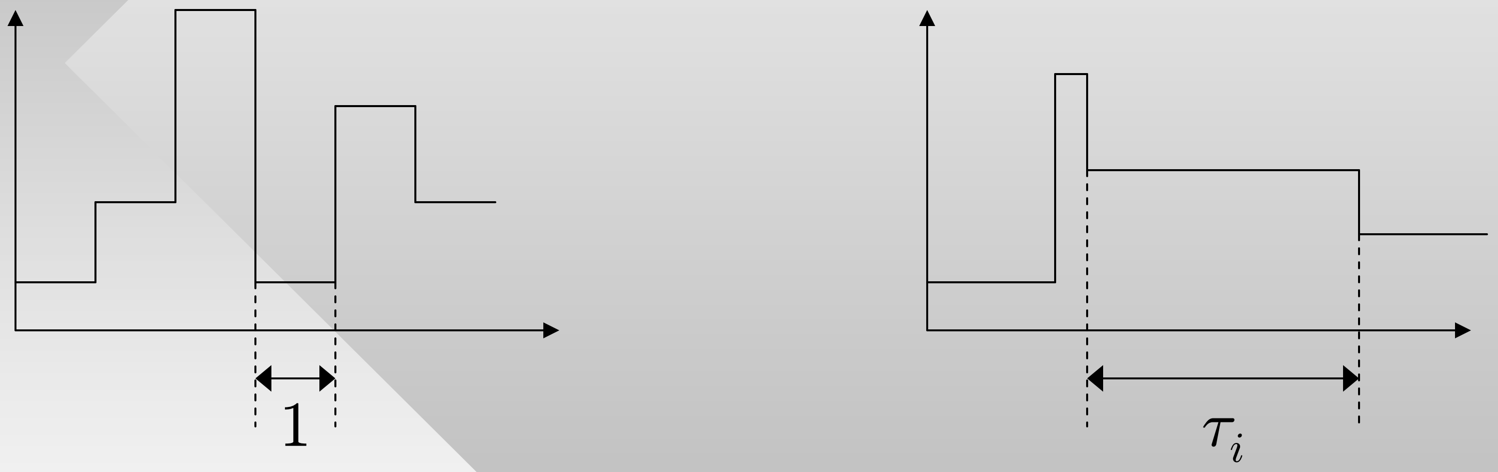
- Continuous chains
  - Definitions
  - M/M/1 queue
  - Loss probability, queue size distribution
- More examples
  - Google PageRank
  - Resilience of P2P networks
- Wrap-up

Continuous Chains

- Second type of Markov chains that we study
  - Recall that continuous-time chains (CTMC) make jumps after spending random amounts of time in each state
  - Each such duration is called *sojourn time* τ_i
- Note that sojourn times are independent; however, their distribution may very with state j
  - Thus, the chain may stay longer in some states that in others
- One classical example of CTMC is the Poisson arrival process $M(t)$ where τ_i are iid exponential variables

Continuous Chains 2

- Here is again an illustration of the differences between discrete and continuous chains



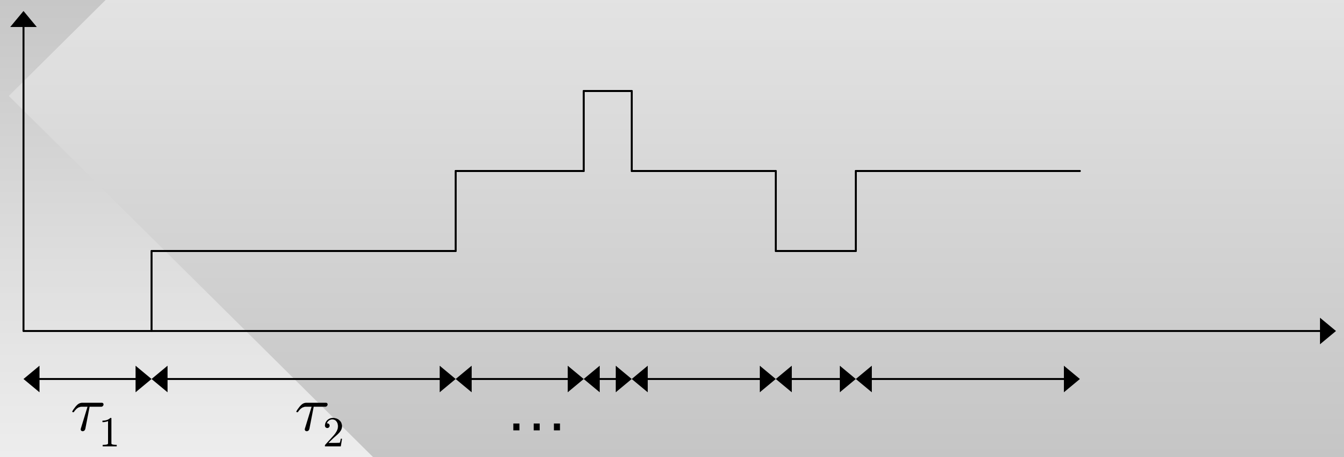
- One important class of CTMC are router queues
  - There are many types of queue models
  - They typically have 3 parameters as in $M/M/1$, which stand for the arrival process, departure process, and the number of routers serving the packets

Continuous Chains 3

- M stands for Markov (exponential delay), D for deterministic (constant inter-renewal delay), G for general (an arbitrary distribution)
  - $M/G/1$, $G/M/1$, $G/G/1$ are all examples with a single router
  - $M/M/c$ or $M/M/\infty$ allow more than one server to work on the packets in parallel
- Consider an $M/M/1$ queue
  - Packets arrive and depart as two Poisson processes with rates λ and μ, respectively
  - State of the chain $N(t)$ at time t is the number of packets in the queue

Continuous Chains 4

- Queue size over time



- Once the process is in state j, the delay to the next packet (i.e., transition to state $j+1$) is an exponential random variable X with CDF $1 - e^{-\lambda x}$
  - We say that the transition rate j → $j+1$ is λ

Continuous Chains 5

- Similarly, since packet service times are also random, the delay before finishing transmission of the current packet (i.e., going from state j to $j-1$) is an exponential variable Y with distribution $1-e^{-\mu x}$
  - We call this transition rate μ
- Why are service times random?
  - Could be many reasons, but one of them is packet size is assumed to be exponentially distributed
  - Not always a realistic assumption for the Internet
- In ATM networks, for example, packet size is fixed and one may use $M/D/1$ queues instead

Continuous Chains 6

- What is the distribution of sojourn times τ_i?
- Notice that τ_i is the minimum between the delay to receive a new packet and serve the next packet
  - Which means $\tau_i = \min(X,Y) \sim$ exp(λ+μ)
- Also straightforward to get transition probabilities

$$p_{i,i+1} = \frac{\lambda}{\lambda+\mu} = p \qquad p_{i,i-1} = 1-p = \frac{\mu}{\lambda+\mu} = q$$

  - Except boundary cases when $i = 0$

Continuous Chains 7

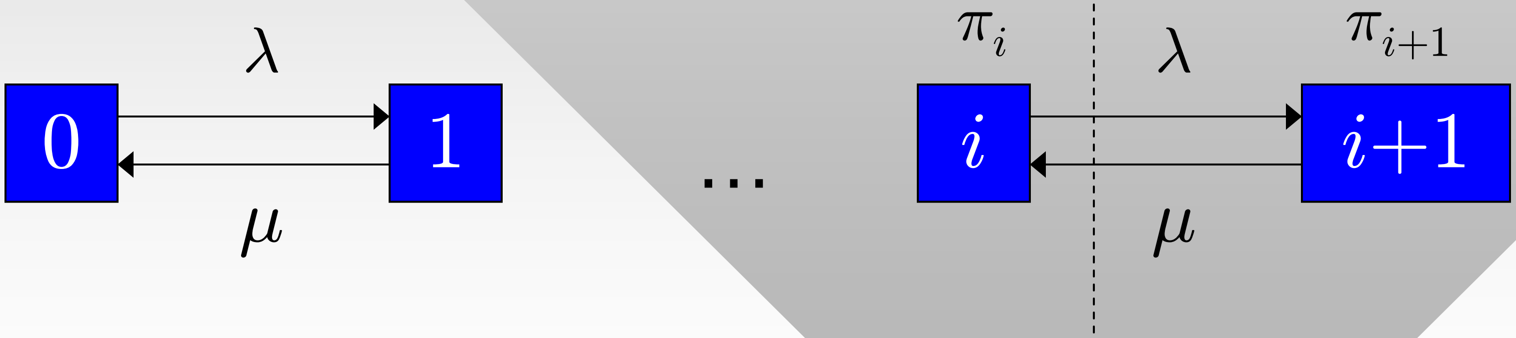
- Thus, the one-step probability matrix of $M/M/1$ queues has the following shape

$$P = \begin{pmatrix} 0 & 1 & 0 & \dots \\ q & 0 & p & \dots \\ 0 & q & 0 & \dots \\ \dots & \dots & \dots & \dots \end{pmatrix}$$

- For all states to be positive, we must have $p < q$ (or equivalently, $\lambda < \mu$)
  - This makes the arrival rate smaller than the service rate
  - In such cases, the queue is said to be “stable”

Continuous Chains 8

- While generally continuous-time chains are more complex that discrete chains, $M/M/1$ queues allow a simple derivation of π
- Recall that transition rates across any boundary of a stable queue must be equal
  - Thus, we can apply similar principles to continuous chains:



  - Rate of transitions from state i to $i+1$ equals π_i (i.e., fraction of time in state i) times the departure rate λ along the link

Continuous Chains 9

- Therefore, we can write our balance equations:

$$\lambda\pi_i = \mu\pi_{i+1}$$

- Recursively expanding:

$$\pi_i = \left(\frac{\lambda}{\mu}\right)^i \pi_0 = \rho^i \pi_0,\ \ \rho = \frac{\lambda}{\mu}$$

- Next we need to determine π_0

$$1 = \sum_{i=0}^{\infty} \pi_i = \pi_0 \sum_{i=0}^{\infty} \rho^i = \frac{\pi_0}{1-\rho}$$

Continuous Chains 10

- Thus, the fraction of time queue size equals i packets is simply:

$$\pi_i = (1 - \rho)\rho^i$$

- Metric $\rho < 1$ is called *traffic intensity* and solely determines the distribution of queue size
  - For example, given $\rho = 0.8$, the probability to find the queue empty is 20% and with 1 packet 16%
  - Also notice that we can view ρ as the *link-utilization factor* since $P(\text{queue is busy}) = 1 - \pi_0 = \rho$
- For $\rho = 0.8$, we can conclude that 8.6% of the time queue size exceeds 10 packets

Continuous Chains 11

- For 95% link utilization, the buffer exceeds 20 packets at least 34% of the time
- Next assume finite buffers of capacity L packets
  - Thus, all transitions from state L occur into state $L-1$
- The only difference is in normalizing π_0

$$1 = \sum_{i=0}^{L} \pi_i = \pi_0 \sum_{i=0}^{L} \rho^i = \frac{\pi_0(1-\rho^{L+1})}{1-\rho}$$

$$\pi_i = \frac{(1-\rho)\rho^i}{1-\rho^{L+1}}$$

Continuous Chains 12

- How many arrivals are now lost?
  - Applying PASTA, the number of customers who observe the chain in state L equals the probability to find the chain in that state!
  - Since all such customers (packets) are lost, we have that the packet-loss rate is equal to π_L:

$$P(\text{packet lost}) = \pi_L = \frac{(1-\rho)\rho^L}{1-\rho^{L+1}}$$

- For example, $\rho = 0.8$ and $L = 10$, we have 2.35% of packets are dropped
  - If $\rho = 0.95$, we will lose 6.9% of arriving packets

Continuous Chains 13

- From all this data, we may want to design a buffer such that the probability of an overflow is below a certain threshold
  - For 1% packet loss and 80% link utilization, buffer size must be at least 14 packets
  - For the same loss and 99% link utilization, we need at least 69 packets
- Practical limitations of $M/M/1$
  - Delays between arrival are usually not 1) iid or 2) exponential
  - Service times are not exponential either
- Bursty traffic requires much larger queues
  - Often ISPs run at 20% or less link utilization on backbone

Example: Google PageRank

- Search engines need a method to rank the importance of pages on the web
- PageRank is one such technique proposed in 1998 under the following intuition:
  - Assumption 1: links pointing to a page indicate endorsement of its content
  - Assumption 2: users randomly surf the web, clicking each outgoing link from a page with equal probability
  - Assumption 3: users may randomly decide to teleport to another page using bookmarks or search-engine results
- Goal: design an algorithm that takes the webgraph G and outputs the ranks of all pages

Google PageRank 2

- Denote by u^+ the set of page u's out-degree neighbors and by u^- the set of its in-degree neighbors
- A "random surfer" keeps clicking on out-going links
  - The probability that the surfer jumps from page u to v is:

$$p_{uv} = \frac{1}{|u^+|}, \text{ if } u \to v$$

u's out-degree

  - The random walk defines a discrete-time MC with transition probability matrix $P = (p_{uv})$, where each state u is a webpage
  - Is this good enough? What if the surfer visits a page with zero out-degree, or enters a loop, or gets bored

Google PageRank 3

- Define a damping factor α to be the probability that the surfer will continue their random walk
  - With probability $1-\alpha$, the surfer jumps to some other page
- The transition probability from page u to v is then:

$$p_{uv} = \begin{cases} \alpha \frac{1}{|u^+|} + (1-\alpha)\frac{1}{N} & u \to v \\ (1-\alpha)\frac{1}{N} & \text{otherwise} \end{cases}$$

number of pages in the graph

- There exists a stationary distribution π for this MC, which satisfies $\pi = \pi P$ $\quad P = (p_{uv})$
- It is then easy to see that

$$\pi_u = \alpha \sum_{v \in u^-} \frac{\pi_v}{|v^+|} + \frac{1-\alpha}{N}$$

PageRank of page u $\quad$ u's in-neighbors

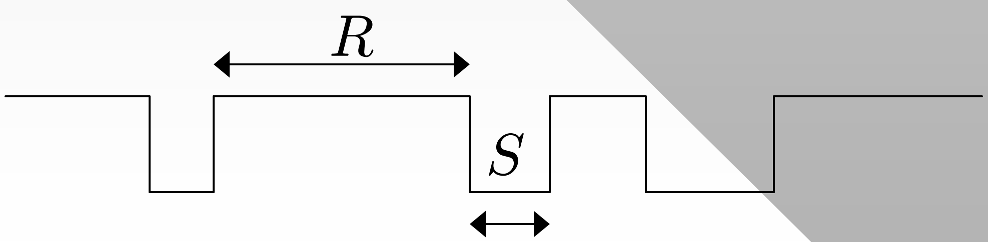
Google PageRank 4

- PageRank π can be computed via matrix operations
  - However, the matrix size $N \times N$ is huge
- In practice, π is obtained by iterating the computation until it converges
  - Assign any random initial vector to π^0
  - While $| \pi^{i+1} - \pi^i | > \epsilon$: $\pi^{i+1} = \pi^i P$
  - After K iterations, obtain the final PageRank π^K
- Dealing with zero out-degree (dangling) nodes
  - Eliminate from the graph, possibly in several iterations
  - Add edges back to in-degree neighbors
  - Teleport with probability 1 (common case in practice)

Example: Resilience of P2P Networks

- Recall from HW3 that each joining user v selects k neighbors from among the existing users
  - Denote by R_i the residual lifetime of neighbor i
  - If neighbor replacement is not allowed, time to isolation $T = \max(R_1, R_2, \ldots, R_k)$
  - Then, the probability that v is isolated is $P(T < L)$, where L is the lifetime of v

- Consider that once a neighbor fails, a replacement is found after some random search delay S
  - The i-th neighbor ON/OFF process:

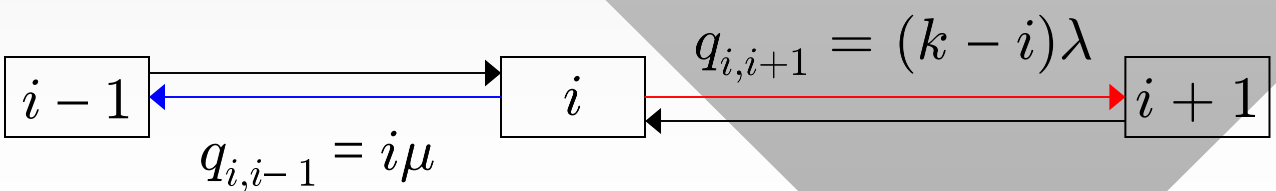


Resilience of P2P Networks 2

- Denote by $W(t)$ the number of neighbors at time t
  - Clearly, this is a continuous-time stochastic process
- Assume that user lifetimes are exponential with rate μ and search delays are exponential with rate λ
  - Neighbor residual lifetimes $R \sim \exp(\mu)$
  - Search delays $S \sim \exp(\lambda)$
  - Then, process $W(t)$ is a Markov chain
- By manipulating this CTMC, we examine the delay T_{00} between visits to state 0
  - Variable T_{00} is a good approximation for isolation time T

Resilience of P2P Networks 2

- Given that $W(t) = i$, there are i live neighbors and $k - i$ dead in search of a replacement
- Transition from i to $i + 1$ is triggered by finding a replacement
  - Transition delay from i to $i + 1$ is $\min(S_1, \ldots, S_{k-i})$, which is exponential with rate $(k - i)\lambda$, for $i < k$
- Transition from i to $i - 1$ is caused by neighbor failure
  - Transition delay from i to $i - 1$ is $\min(R_1, \ldots, R_i)$, which is exponential with rate $i\mu$, for $i > 0$



Resilience of P2P Networks 3

- Compute the stationary distribution π of process $W(t)$
- Recall that the transition rates across any boundary must be equal:

$$\pi_i q_{i,i-1} = \pi_{i-1} q_{i-1,i}$$

transition rate from state i–1 to i

  - The above are called balance equations. It follows that

$$\pi_i = \pi_0 \rho^i \frac{k!}{i!(k-i)!}, \qquad \rho = \frac{\lambda}{\mu}$$

- After normalization, we have:

$$\pi_i = \binom{k}{i} \frac{\rho^i}{(1+\rho)^k}, \qquad i = 0, 1, \ldots, k$$

Resilience of P2P Networks 4

- For continuous-time Markov chains, the expectation of delay T_{jj} between visits to state j is given by:

$$E[T_{jj}] = \frac{1}{\pi_j q_j}$$

transition rate out of state j

$q_0 = k\lambda$

- The mean delay T_{00} between visits to state 0 is $1/\pi_0 q_0$
  - Node isolation probability $\phi = P(L < T)$ is close to $E[L]/E[T_{00}]$, where $E[L]$ is the mean user lifetime
  - Compare isolation probability ϕ for exponential lifetimes with $E[L] = 0.5$ hours and $k = 8$

	Model $E[L]/E[T_{00}]$	ϕ with replacement	ϕ without replacement
$E[S] = 0.1$	1.5×10^{-5}	2.4×10^{-5}	$1/(k+1) = 0.111$
$E[S] = 0.01$	8.3×10^{-12}	8.7×10^{-12}	$1/(k+1) = 0.111$

Significant improvement!

Wrap-up

- We skipped a lot of technical definitions and associated derivations in chapters 3-4
- Even the simplest stochastic queues require non-trivial modeling efforts and half of a semester of derivations
- Midterm includes today's lecture
  - Practice computing π for continuous chains