Performance Evaluation of Queueing Networks - Outline

- Introduction - networks of queues are the example family of systems to be studied
- Deterministic models including network calculus
- Dynamic routing
- Review of elements of probability & statistical confidence, overview of simulation
- Stationary (and ergodic and stable) models
- Markovian models in continuous and discrete time
- Parallel and distributed processing, fork-join queues
- Markov decision processes
- Constrained optimization and duality with examples

Performance Evaluation of Queueing Networks - Outline (cont)

- Queueing system models have been used in a wide range of applications including computer/communication networking, computation, supply chain and logistics.
- The focus of this course will be (unambiguous) theoretical derivations of performance objectives based on models of queueing system and their workloads.
- To this end, we will review the basic, relevant elements of probability theory.
- We will also discuss performance evaluation based on simulation.
- Simulation is useful when system or workload complexity precludes simple models that lead to close-form analytical results for the performance objectives.
- We also will review the use of statistical confidence when reporting the results of a simulation study.
Performance Evaluation of Queueing Networks - Outline (cont)

- In the following, our approach to performance evaluation will be to will consider models of increasing detail:
  1. Deterministic, including worst-case analysis
  2. Stationary and ergodic
  3. Stationary Markovian

- We will demonstrate how increased model complexity (assumed suitable for the physical system under consideration) leads to more refined and detailed performance results.

- We will not consider non-Markovian stochastic models such as self-similar models exhibiting long-range dependence.

- Also, we will not consider stochastic models that are time-varying nor those that possess deterministic (e.g., time-of-day/day-of-week) trends.

Deterministic models of queues and queuing networks

- Arrivals, departures and queue occupancy
- Traffic shaping - token buckets, service curves
- Flow scheduling
- Network calculus
Queues - preliminaries

- A queue or buffer is simply a waiting room with an identified arrival process and departure (completed “jobs”) process.

- Work is performed on jobs by servers according to a service policy.

- In some applications, jobs arriving to the queue will be packets of information; in others, the arrivals will represent calls attempting to be set-up in the network.

- Some jobs may be blocked from entering the queue (if the queue’s waiting room is full) or join the queue and be expelled from the queue before reaching the server.

- For jobs reaching the server, their queueing delay plus service time is called their sojourn time, i.e., the time between the arrival of the job to the queue and its departure from the server.

- We will consider queues that serve jobs in the order of their arrival known as first come, first serve (FCFS) or first in, first out (FIFO).

Arrivals, departures, and queue occupancy

- Over the time interval \((0, t]\), the counting process
  - \(\{A(0, t] \mid t \in \mathbb{R}^+\}\) represents the number of jobs arriving at the queue,
  - \(\{D(0, t] \mid t \in \mathbb{R}^+\}\) represents the number of departures from the queue,
  - \(\{L(0, t] \mid t \in \mathbb{R}^+\}\) represents the number of jobs blocked (lost) upon arrival.

- Let \(Q(t)\) be the number of jobs in the queueing system at time \(t\); i.e.,
  - the occupancy of the queue plus the number of jobs being served at time \(t\);
  - including the arrivals at \(t\) but not the departures at \(t\).

- We assume no jobs with zero sojourn time.
Arrivals, departures, and queue occupancy (cont)

- Clearly, a previously "arrived" job is either queued or has departed or has been blocked, i.e.,
  \[ Q(0) + A(0, t) = Q(t) + D(0, t) + L(0, t). \]
- If we take the origin of time to be \(-\infty\), we can simply write
  \[ Q(t) = A(-\infty, t] - D(-\infty, t] - L(-\infty, t]. \]

Basic assumptions

- We’ll typically assume that:
  - Servers are nonidling (or "work conserving") in that they are busy whenever \( Q(t) > 0 \).
  - A job’s service cannot be preempted by another job.
  - Jobs may only be blocked upon arrival to a queue.
  - All servers associated with a given queue work at the same, constant rate (otherwise, need to define the work each job brings).
- Thus, we can unambiguously define \( S_i \) to be the service time required by the \( i^{th} \) job.
- In addition, each job \( i \) will have the following two quantities associated with it:
  - its arrival time to the queueing system \( T_i \), assumed to be a nondecreasing sequence in \( i \) (\( \forall i, T_i \leq T_{i+1} \)), and
  - its departure (service completion) time from the server \( V_i \) if the job is not lost (blocked upon arrival).
Queue workload (not blocked jobs)

- Let $R_i(t)$ be the residual amount service time required by the $i^{th}$ job at time $t$.
- Clearly, $0 \leq R_i(t) \leq S_i$ for all $i$, $t$; $R_i(t) = 0$ for $t > V_i$; $R_i(t) = S_i$ for $t < V_i - S_i$.
- The total work-to-be-done (or workload) at time $t$, $W(t)$, is simply the sum of the service times of all queued jobs and residual service times of all jobs being served at time $t$.
- For jobs $i$ that are not lost (i.e., not dropped upon arrival), let $V_i$ be the departure time of the job from the server.
- Clearly, $V_i - S_i$ is the time at which the $i^{th}$ job enters a server and, for all $t$ and $i \in J_S(t)$,
  $$R_i(t) = V_i - t.$$  
- Clearly, a job $i$ is in the queue but not in service if $T_i \leq t < V_i - S_i$.

Parameterizing queue arrival and departure processes

- The arrival process $A$ is parameterized above as $\{T_i, S_i\}_{i \in \mathbb{Z}}$ or $\mathbb{Z}^+$.
- The queueing discipline determines how jobs are enqueued and in which order they are served (dequeued), i.e., the dynamics of queue $Q$ and workload $W$ processes.
- The departure process $D$, parameterized by $\{V_i, S_i\}$ is determined by both the queueing discipline and the arrival process.
- For a given arrival process and queueing discipline, we are typically interested in determining the "system" processes $Q$ and $W$ only in terms of the arrival parameters, i.e., not using the departure times $V_i$ as these may not be known a priori.
Lossless queues

- Now assume the queue we have just introduced is lossless, i.e., \( L(-\infty, t] = 0 \) for all \( t \).
- Define the indicator \( 1_B = 1 \) if \( B \) is true, else \( = 0 \). Since,
  \[
  A(s, t] = \sum_i 1\{T_i \in (s, t]\}, \quad \text{and} \quad D(s, t] = \sum_i 1\{V_i \in (s, t]\},
  \]
  we get (by recalling \( \forall i, V_i > T_i \) by assumption) that
  \[
  Q(t) = A(-\infty, t] - D(-\infty, t] = \sum_i 1\{T_i \leq t < V_i\}.
  \]
- The sojourn time is the total delay experienced by the \( i \)th job, \( V_i - T_i \), i.e., the departure time minus the arrival time.
- Again, this sojourn time consists of two components: the queueing delay, \( V_i - T_i - S_i \), plus the service time, \( S_i \).
- Expressions will be derived for quantities of interest such as the number of jobs in the queue, the workload, and job sojourn times.
- The objective is to express quantities of interest in terms of the job arrival times and service times alone.

The case of no waiting room

- Suppose the queueing system consists only of the servers and no waiting room.
- Thus, if the job flow is demultiplexed (demux’ed) to one of \( K \) servers, the queueing system can only hold \( K \) jobs at any given time.
- Since the system is assumed lossless: for all jobs \( i \),
  \[
  V_i = T_i + S_i
  \]
- Since there are infinitely many servers (\( K = \infty \)), the system is always lossless and so the number of jobs queued and the workload are
  \[
  Q(t) = \sum_i 1\{T_i \leq t < T_i + S_i\} = \sum_i 1\{T_i \in (t - S_i, t]\}
  \]
  \[
  W(t) = \sum_i 1\{T_i \in (t - S_i, t]\} R_i(t)
  \]
- In the following figure, note how the negative slope of the workload sample path is proportional to the number of jobs currently queued.
The case of no waiting room - example sample path

Now suppose that the queue has a waiting room and only a single server.

Clearly, if the waiting room was infinite in size, the queue would be lossless irrespective of the job arrival and service times.

For the following example sample path, note that upon arrival of the $i^{th}$ job at time $T_i$, $Q$ increases by 1 and $W$ increases by $S_i$.

The process $Q$ is piecewise constant and, due to the action of the server, $W(t)$ has zero time derivative if $Q(t) = 0$ (i.e., $W$ is constant) and otherwise has time derivative $-1$ for any $t$ that is not a job arrival time.

Upon departure of the $i^{th}$ job, $Q$ decreases by 1.
Lossless single-server queue: Departure-times recursion

- **Theorem:** For a work-conserving, single-server, lossless FIFO queue, the $i^{th}$ job’s departure time
  \[ V_i = \max\{V_{i-1}, T_i\} + S_i \]
  for all jobs $i \in \mathbb{Z}^+$, where $V_0 \equiv 0$.

- **Proof:** For the $i^{th}$ job arriving at the lossless queue, there are two cases.
  - If $T_i > V_{i-1}$, then:
    - job $i - 1$ has already departed the queue by time $T_i$.
    - So, $Q(T_i) = 0$ and,
    - when the $i^{th}$ job joins the queue, it immediately enters the server.
    - So, it departs $S_i$ seconds after it arrives, i.e., $V_i = T_i + S_i$.
  - On the other hand, if $T_i \leq V_{i-1}$,
    - job $i - 1$ is present in the queue (and immediately ahead of the $i^{th}$ job) when the $i^{th}$ job joins the queue.
    - Thus, the $i^{th}$ job will depart the queue $S_i$ seconds after job $i - 1$, i.e., $V_i = V_{i-1} + S_i$. 
- Note that, by subtracting $T_i$ from both sides of the departure-times recursion, we get a statement involving the sojourn times $V_i - T_i$ and the interarrival times $T_i - T_{i-1}$:

\[
V_i - T_i = \max\{V_{i-1} - T_i, 0\} + S_i = \max\{(V_{i-1} - T_{i-1}) - (T_i - T_{i-1}), 0\} + S_i,
\]

where $T_0 \equiv 0$.

- An immediate consequence of the FIFO nature of a single-server queue is this relation to workload:

\[
V_i = T_i + W(T_i).
\]

- Again, here we take the work brought by each job $i$, $S_i$, as its required service time.

- Also note that the time at which the $i^{th}$ job enters the server is

\[
\max\{V_{i-1}, T_i\}
\]

Single server and constant service times

- Suppose each job requires the same amount of service, i.e., for some constant $c > 0$, $S_i = 1/c$ for all $i$.

- So, the service rate of any server can be described as $c$ jobs per second. Further suppose that the (assumed lossless) queue has a waiting room.

- Because each job contributes $c^{-1}$ to workload upon its arrival, the number of jobs in the system in terms of the workload is, $\forall t$,

\[
Q(t) = \lceil cW(t) \rceil.
\]

- That is,

\[
\frac{1}{c}(Q(t) - 1)^+ < W(t) \leq \frac{1}{c}Q(t)
\]

recalling that $W(t)$ and $Q(t)$ include the work arriving at time $t$.

- So, $Q(t) = \lceil cW(t) \rceil$ follows because $Q(t)$ is integer valued.
Max-plus expression for workload

- **Theorem:** For a work-conserving, single-server, lossless, initially empty \((W(0) = 0)\) FIFO queue with constant service times,

\[
W(t) = \max_{0 \leq s \leq t} \left( \frac{1}{c} A[s, t] - (t - s) \right)
\]

for all times \(t \geq 0\), where the maximizing value of \(s\) is \(t\) if \(W(t) = 0\), else starting time of the busy period containing \(t\).

Max-plus expression for workload - proof

- We first define a notion of a queue **busy period** as an interval of time \([s, t]\) with \(s < t\) such that:
  - \(W(s-) = Q(s-) = 0\), i.e., the system is empty just prior to time \(s\),
  - \(W(r) > 0\) (and \(Q(r) > 0\)) for all time \(r \in [s, t]\), and
  - \(W(t) = Q(t) = 0\), i.e., the system is empty at time \(t\).

- Queue busy periods (each started by a job arrival to an empty queue) are separated by **idle periods**, which are intervals of time over which \(W\) (and \(Q\)) are both always zero.

- So, the evolution of \(W\) is an alternating sequence of busy and idle periods.
• Arbitrarily fix a time $t$ somewhere in a queue busy period, i.e., $Q(t), W(t) > 0$.

• Define $b(t)$ as the starting time of the busy period containing time $t$, so that, in particular, $b(t) \leq t$ and $W(b(t) -) = 0$.

• The total work that arrived over $[b(t), t]$ is $A[b(t), t]/c$ and the total service done over $[b(t), t]$ was $t - b(t)$.

• Since $W(s) > 0$ for all $s \in [b(t), t]$,
  \[
  W(t) = \frac{1}{c}A[b(t), t] - (t - b(t)).
  \]

• Furthermore, for any $s \in [b(t), t)$,
  \[
  W(t) = W(s-) + \frac{1}{c}A[s, t] - (t - s) \geq \frac{1}{c}A[s, t] - (t - s).
  \]

• Now consider a time $s < b(t)$.

• Since $W(b(t) -) = 0$, any arrivals over $[s, b(t))$ have departed by time $b(t)$; this implies that
  \[
  \frac{1}{c}A[s, b(t)) - (b(t) - s) \leq 0.
  \]

• Therefore,
  \[
  \frac{1}{c}A[s, t] - (t - s) = \frac{1}{c}A[s, b(t)) - (b(t) - s) + \frac{1}{c}A[b(t), t] - (t - b(t)) \leq \frac{1}{c}A[b(t), t] - (t - b(t)) = W(t).
  \]

• So, we have proved the desired result for the case where $W(t) > 0$.

• The other case, where $t$ is in an idle period (i.e., $Q(t), W(t) = 0$), is similarly proved. \(\Box\)
Max-plus expression for queue backlog

- Combining the last two results gives

\[ Q(t) = \lceil \max_{0 \leq s \leq t} A[s, t] - (t - s)c \rceil. \]

- Also, when the \( i^{th} \) job is in the server at time \( t \),

\[ W(t) = \frac{1}{c} \max \{Q(t) - 1, 0\} + V_i - t. \]

Single server and general service times

- Now consider a lossless FIFO single-server queue wherein the \( i^{th} \) arriving job has service time \( S_i \).

- Here,

\[ W(t) = \max_{0 \leq s \leq t} \sum_i S_i 1\{s \leq T_i \leq t\} - (t - s), \]

since

\[ A[s, t] = \sum_i S_i 1\{s \leq T_i \leq t\}. \]
Single server and general service times (cont)

• Alternatively focusing just on job arrival times, let $i(t)$ be the index of the last job arriving prior to time $t$, i.e.,

$$i(t) \equiv \max\{j \mid T_j \leq t\}.$$

• For this queue, the workload is given by

$$W(t) = \left(\max_{j \leq i(t)} \sum_{k=j}^{i(t)} S_k \right) - (t - T_j)^+,$$

where $(x)^+ \equiv \max\{x, 0\}$.

Queues in communication/computer networks

• Now consider packet queues/buffers in communication/computer networks operated by network providers.

• In particular, such queues reside in network switches and routers.

• At their network boundaries, network providers strike service-level agreements (SLAs) wherein the transmitting network agrees that his or her egress packet flow will conform to certain parameters.
A 3×3 Router

Linecards of a Router

Note: VOQs and VIQs about the switch fabric, and eTM in egress linecard
SLA parameters regarding packet flows

- A preferable choice of flow parameters would be those that are:
  - significant from a queueing perspective, simply to ensure conformity by the sending network, and
  - simple to police by the receiving network.

- We will see how useful the mean arrival rate (typically denoted by $\lambda$) is in terms of predicting the queueing behavior/performance.

- The mean arrival rate is, however, difficult to police as it is only known after the flow has terminated.

- Instead of the mean arrival rate, we consider flow parameters that are policeable on a packet-by-packet basis.

The burstiness of a packet-flow

- Suppose that when the flow of packets arrives to a dedicated FIFO queue
  - with a constant service rate of $\rho$ bytes per second (Bps),
  - the backlog of the queue never exceeds $\sigma$ bytes.

- One can define $\sigma$ as the burstiness of a flow of packets as a function of the rate $\rho$ used to service it.

- Such a definition for burstiness informs a node so that it can allocate both memory and bandwidth resources in order to accommodate such a regulated flow.

- Moreover, by limiting the burstiness of a flow, one also limits the degree to which it can affect other flows with which it shares network resources.

- Indeed, such traffic regulation was standardized by the ATM Forum and adopted by the Internet Engineering Task Force (IETF); see RFCs 2697 and 2698 at www.ietf.org
Token (leaky) buckets for packet-traffic shaping - preliminaries

- Suppose that at some location there is a flow of packets $A$ specified by the sequence of pairs $(T_i, l_i)$, where
  - $T_i$ is the arrival time of the $i^{th}$ packet in seconds ($T_{i+1} > T_i$) and
  - $l_i$ is the length of that packet in bytes (both work that the $i^{th}$ packet brings and memory it occupies in the queue).
- The total number of bytes that arrives over an interval of time $(s, t]$ is
  \[ A(s, t] = \sum_i l_i \mathbb{1}\{s < T_i \leq t \}. \]

Token (leaky) buckets for packet-traffic shaping (cont)

- Assume that this packet flow arrives to a token bucket mechanism.
- A token represents a byte and tokens arrive at a constant rate of $\rho$ tokens/s to the token bucket which has a limited capacity of $\sigma$ tokens.
- A (head-of-line) packet $i$ leaves the packet FIFO queue when $l_i$ tokens are present in the token bucket;
  - when the packet leaves, it consumes $l_i$ tokens, i.e., they are removed from the bucket.
- Note that this mechanism requires that $\sigma$ be larger than the largest packet length (again, in bytes) of the flow.
Token (leaky) buckets for packet-traffic shaping (cont)

- Let $A_o(s, t]$ be the total number of bytes departing from the packet queue over the interval of time $(s, t]$.

- The following result is directly proved by considering the maximal amount of tokens that can be consumed over an interval of time.

- **Theorem:** For all arrival processes $A$ to the packet queue,
  
  $$A_o(s, t] \leq \sigma + \rho(t - s), \quad \forall \ s \leq t.$$  

- Any flow $A_o$ that satisfies this inequality is said to satisfy a $(\sigma, \rho)$ constraint.

- In the jargon of the IETF RFCs, $\rho$ could be a sustained information rate (SIR), and $\sigma$ a maximum burst size (MBS).

- Alternatively, $\rho$ could be a peak information rate (PIR > SIR), in which case $\sigma$ would usually be taken to be the number of bytes in a (single) maximally sized packet (< MBS).

- Note that the mean departure rate over $(s, t]$ is $A_o(s, t]/(t - s) \leq \rho + \sigma/(t - s) \approx \rho$ for large $t - s$.

Bounded queue backlog if $(\sigma, \rho)$ constrained arrivals

- Let $W(t)$ be backlog at time $t$ of a queue with arrival flow $A_o$ and a dedicated server with constant rate $\rho$.

- **Theorem:** The flow $A_o$ is $(\sigma, \rho)$ constrained if and only if $W(t) \leq \sigma$ for all time $t$.

- **Proof:** The maximum queue size is
  
  $$\max_t W(t) = \max_t \max_{s: \ s \leq t} \{A_o(s, t] - \rho(t - s)\}.$$  

- Substituting the $(\sigma, \rho)$ inequality gives the result. $\square$
Traffic shaping and policing

- We have shown how the token bucket can delay packets of the arrival flow \( A \) so that the departure flow \( A_0 \) is \((\sigma, \rho)\) constrained.

- This is known as traffic shaping.

- The receiving network of the exchange of flows described above may wish to:
  - shape the flow using a \((\sigma, \rho)\) token bucket, or
  - police the flow by simply identifying (marking) any packets that are deemed out of the \((\sigma, \rho)\) profile of the flow, or
  - police the flow by dropping any out of profile packets.

- There are two main devices used for traffic policing.

- The first is a token-bucket device but without the packet queue: A packet is dropped or marked out of profile if and only if there are not sufficient tokens (according to its length) in the token bucket upon its arrival (no tokens consumed if dropped).

Traffic policing

- Alternatively, by the previous theorem, one can employ a policer as depicted above which does not delay any packets.

- A packet is dropped or marked out-of-profile if and only its arrival and inclusion in the virtual queue would cause its backlog \( Q \) to become larger than \( \sigma \);

- when this happens, the arriving packet is not included in the virtual queue.

- Note that the virtual queue can be maintained by simply keeping track of two state variables:
  - the queue length, \( Q \), upon arrival of the previous packet and
  - the arrival time, \( \alpha \), of the previous packet.
Traffic policing (cont)

Thus if a packet of length $l$ bytes arrives at time $T$ and is admitted into the virtual queue, then

$$Q \leftarrow \max\{Q - \rho(T - a), 0\} + l \quad \text{and} \quad a \leftarrow T.$$

This (event-driven) operation requires one multiplication operation per packet.

Alternatively, one could maintain the departure time $d$ of the most recently admitted packet instead of the queue occupancy $Q$.

Traffic policing: 2R3CM

If two such virtual queues are used, one for (SIR,MBS) and the other for PIR, then every packet has one of four fates:

- in-profile for both
- out-of-profile for PIR but in-profile for (SIR,MBS)
- in-profile for PIR but out-of-profile for (SIR,MBS)
- out-of-profile for both

Thus, one of three three different “colors” can be used to mark the out-of-profile packets (by setting a field in their headers).

This policing system with two virtual queues is called a two-rate, three-color marker (2R3CM)
- again, see RFCs 2697 and 2698 at www.ietf.org
• Suppose that at some location, \( N \) flows are to be multiplexed (scheduled) into a single flow.

• Similarly, scheduling sequences of jobs of variable work amounts.

• The flows are indexed by \( n \in \{0, 1, ..., N - 1\} \) below.

• Each flow \( n \) is assigned its own tributary FIFO queue with “relative allocation” \( f_n \), and the output flows of the tributary queues are multiplexed into the transmission FIFO queue.

• How the multiplexing occurs depends on the kinds of relative priorities of the flows.

\[
\begin{align*}
&\text{tributary queues} \\
&\text{FIFO queue 0} \\
&(a_0, l_0) \\
&\text{f_0} \\
&\text{\vdots} \\
&\text{\vdots} \\
&(a_{N-1}, l_{N-1}) \\
&\text{f_{N-1}} \\
&\text{FIFO queue N – 1} \\
\end{align*}
\]

\( \text{transmission queue} \)

\( c \text{ bytes/s} \)

\( \text{mux} \)

FIFO scheduling

• First suppose a system without tributary queues, \( i.e., \) all flows directly arrive to the transmission queue.

• In FIFO scheduling, packets are served in first-come first-served (first-in first-out) fashion.

• Hard to differentially manage per-flow service \( (f_n) \) this way - perhaps a differential rule for queue admission/blocking.

• Also, flows more readily “interfere” with each other.

• Note that FIFO queues without overtaking or push-out have minimal per-packet overhead: operations only at the head (join, block) or tail (serve) of the queue (doubly linked list).
Strict priority scheduling

- Now and hereafter suppose that each flow $n$ has a separate tributary FIFO queue/buffer so that “flow” and “queue” (or “transmission queue”) may be used interchangeably.
- In strict priority multiplexing, flows are ranked according to priority.
- A flow is served by the scheduler only if no packets of any higher priority flows are queued.
- Even when the volume of high priority traffic is limited (perhaps by a leaky bucket mechanism), there remains the potential problem of service starvation to lower priority flows.
- The problems with both priority and single FIFO-queue multiplexing can be solved by using a scheduler that can in some way allocate service bandwidth to a flow in order to prevent long-term service starvation.

Lottery scheduling

- Let $\ell_n$ be the size in bytes of the head-of-line packet of queue/flow $n$ and
- $c$ Bps be the capacity of server shared by the flows.
- Under a kind of (work-conserving) “lottery” scheduling, the discrete distribution
  \[
  \left\{ \frac{f_1/\ell_1}{Z}, \frac{f_2/\ell_2}{Z}, \ldots, \frac{f_N/\ell_N}{Z} \right\}
  \]
  is sampled to determine which queue to serve next, where the normalizing term
  \[ Z = \sum_{n=1}^{N} \frac{f_n}{\ell_n} \]
  and take $\ell_n = \infty$ when queue $n$ is empty.
- In addition to sampling this discrete distribution, the overhead of lottery scheduling involves recomputing its terms $(f_n/\ell_n)/Z$ when the size of a head-of-line packet changes, particularly when a queue becomes empty or nonempty.
Lottery scheduling (cont)

- Let $D_n$ and $D_m$ be the total number of bytes transmitted by two different flows $n \neq m$ over an interval of time $T$ during which both of their queues are never empty.

- By the law of large numbers, the throughput of non-idle flows $n$ will be proportional to $f_n$ in the long run, i.e.,

$$\lim_{T \to \infty} \left| \frac{D_n(T)}{f_n} - \frac{D_m(T)}{f_m} \right| = 0.$$

- But due to random sampling, it’s possible that a queue is starved of service, i.e.,

$$P(D_n(T) = 0) > 0 \text{ for (finite) } T > 0.$$

Deficit round-robin

- Under round-robin multiplexing (scheduling), time is divided into successive rounds, perhaps each not necessarily of the same time duration depending on which flows (tributary queues) are active.

- Each flow is visited once per round by the scheduler.

- Suppose that in each round there is a rule allowing for at most one packet per tributary queue to be transmitted into the transmission queue.

- A problem here is that flows with large-sized packets (e.g., large file transfers using TCP) will monopolize the bandwidth and starve out flows of small-sized packets (e.g., those of streaming media).

- Thus, one might want to regulate the total number of bytes that can be extracted from any given tributary queue in a round.

- This leads to the notion of deficit round-robin (DRR) scheduling.
Deficit round-robin - definition

- To describe a DRR mechanism, we need the following definitions.
- Let $L_{\text{max}}$ be the size, in bytes, of the largest packet and $L_{\text{min}}$ the size of the smallest.
- Here, the priority of a flow has to do with the fraction $f_n$ of the total link bandwidth $c$ bytes per second assigned to it, where we assume no overbooking:
  $$\sum_{n=1}^{N} f_n \leq 1.$$
- In practice, resources may be overbooked to exploit “statistical multiplexing”.
- Finally, let the minimal allotment of bandwidth to a queue be
  $$f_{\text{min}} = \min_n f_n$$

Deficit round-robin - definition (cont)

- Under DRR, at the beginning of each round, each nonempty FIFO queue is allocated a certain number of tokens.
- Packets departing a queue consume their byte length in tokens from the queue’s allotment.
- Queues are serviced in a round until their token allotment becomes insufficient to transmit their next head-of-line packet.
- For example, if a queue is allocated 8000 tokens at the start of a round and has six packets queued each of length 1500 bytes, then the first five of those packets are served leaving the trailing sixth packet at the head of the queue and $8000 - 5 \times 1500 = 500$ tokens unused.
- If it’s not empty, the $n^{th}$ queue is allocated
  $$\frac{f_n}{f_{\text{min}}} L_{\text{max}}$$ tokens
  at the start of a round, thereby ensuring that at least one packet from this queue will be transmitted in the round irrespective of the packet’s size.
- If a queue has no packets at the end of a round, its remaining token allotment may be reset to zero - in the following, assume that at most one round’s worth of tokens can carry over to the next.
Deficit round-robin - discussion and performance

- Note that the token allotments per round can be precomputed given service requirements $f_n$, where
- the $f_n$ themselves change at a much slower "connection-level" time scale than that of the transmission time required for a single packet ($L_{max}/c$).
- One could replace $f_{min}$ in the token allocation rule by the minimum bandwidth allocation among nonempty queues at the start of a round, but the result would be a significant amount of computation per round possibly precluding a high-speed implementation.
- Claim: If the $n^{th}$ queue is always not empty over $k$ consecutive rounds with constant $f_{min}$, then cumulative bytes $D_n(k)$ transmitted from this queue over this period satisfies
  $$k \frac{f_n}{f_{min}} L_{max} - L_{max} \leq D_n(k) \leq (k + 1) \frac{f_n}{f_{min}} L_{max}.$$  
- Proof: The upper bound is obtained assuming that all allocated tokens are consumed in addition to a maximal amount of carryover tokens from the round prior to the $k$ consecutive ones under consideration.
- The lower bound is obtained by assuming no carryover tokens from a previous round and a maximal number of unused tokens in the last round. □

DRR is rate-proportionally fair

- The previous claim demonstrates that DRR scheduling indeed allocates bandwidth consistent with the parameters $f_n$.
- Exercise: If two queues $n$ and $m$ are never idle over an interval of $k$ consecutive rounds, show that
  $$\left| \frac{D_n(k)}{f_n} - \frac{D_m(k)}{f_m} \right| \leq \max \left\{ \frac{1}{f_n}, \frac{1}{f_m} \right\} L_{max},$$  
  (bound doesn't depend on $k$) and
  $$\lim_{k \to \infty} \frac{D_n(k)}{D_m(k)} = \frac{f_n}{f_m}.$$  
- Exercise: Derive related bounds if $L_{max}$ is different for the two queues or if $f_{min}$ changes, i.e., $f_{min,i}$ for round $i$.
- Exercise: Explain a potential problem if more than one round’s work of unused tokens are allowed to accumulate for a flow.
Shaped VirtualClock

- We will now describe a scheduler
  - that employs timestamps to give packets service priority over others
  - but restricts consideration only to packets that meet an eligibility criterion
  - to limit the jitter of the individual output flows.

- This trait, which is lacking in DRR, is important for link channelization (partitioning a link into smaller channels) at network boundaries where SLAs are struck and policed.

- A general problem of timestamp based scheduling is that dequeue requires $O(\log N)$ dequeue complexity to determine the flow with smallest head-of-line/queue packet timestamp.

Shaped VirtualClock - definition

- For all $i$ and $n$, $(n, i)$ denotes the $i^{th}$ packet of the $n^{th}$ flow.

- Packet $(n, i)$ is assigned a service deadline $d_{n,i}$ and a service eligibility time $\varepsilon_{n,i}$. A packet is said to be eligible for service at time $t$ if $\varepsilon \leq t$.

- As with DRR, once a packet begins service, its service is not interrupted.

- Upon service completion of a packet, the next packet selected for service will be the one with the smallest deadline among all eligible packets.

- Assuming the queues are FIFO, only head-of-queue packets need to be considered by the multiplexing (scheduling) algorithm.

- Each packet $(n, i)$ has two other important attributes: its arrival time $a_{n,i}$ to the multiplexer and its size in bytes, $l_{n,i}$.

- Under what we will hereafter call shaped VirtualClock (SVC) scheduling, packet $(n, i)$’s eligibility time and deadline are

$$\varepsilon_{n,i} := \max\{d_{n,i-1}, a_{n,i}\} \quad \text{and} \quad d_{n,i} := \frac{l_{n,i}}{f_{n,C}}$$
SVC - performance evaluation - preliminaries

- That is, if the $n^{th}$ flow were instead to arrive to a queue with a dedicated server of constant rate $f_n c$ bytes per second, then packet $(n, i)$ would:
  - reach the head of the queue (and begin service) at its eligibility time $\varepsilon_{n,i}$ and
  - completely depart the server at its service deadline $d_{n,i}$.

- Recall the Lindley recursion of the packet departure times for this virtual queue $n$:

  $$d_{n,i} = \max\{d_{n,i-1}, a_{n,i}\} + \frac{l_{n,i}}{f_n c} = \varepsilon_{n,i} + \frac{l_{n,i}}{f_n c}$$

- Lemma: Just prior to the start time of a busy period of the multiplexer, the aggregate eligible work to be done of all $N$ virtual queues is zero.

- This lemma is used to prove a guaranteed-rate property of SVC.

SVC - guaranteed rate property

- Now recall $L_{\text{max}}$ is the maximum size of a packet (in bytes), a quantity that is typically about 1500 in the Internet.

- The following theorem demonstrates that SVC schedules bandwidth appropriately in our time-division multiplexing context:

- Theorem: For all $n$ and $i$, the time at which packet $(n, i)$ completely departs from the multiplexer is not more than

  $$d_{n,i} + \frac{L_{\text{max}}}{c}.$$

- This is a kind of guaranteed-rate result for the SVC multiplexer.

- Such results can easily be extended to an end-to-end guaranteed-rate property of a tandem system of such multiplexers.
SVC - output burstiness

- The SVC multiplexer also has an appealing property of bounding the jitter of every output flow.

- Consider any flow/queue \( n \) and note that the \( i \)th packet of this flow
  - will have completely departed the multiplexer between times \( \varepsilon_{n,i} + \frac{l_{n,i}}{c} \) and \( d_{n,i} + \frac{L_{\text{max}}}{c} \),
  - where \( l_{n,i}/c \) is its total transmission time.

- We can use this fact and the fact that \( d_{n,i} \leq \varepsilon_{n,i+1} \) to show:

  - **Theorem:** The cumulative departures from the \( n \)th queue of the multiplexer over an interval of time \([s, t] \) is less than or equal to \( f_{nc}(t - s) + 2L_{\text{max}} \) bytes.

- That is, the departure process is \((f_{nc}, 2L_{\text{max}})\)-constrained.

SVC - discussion

- Another perspective for SVC is that flows
  - just get what they pay for (i.e., service \( f_{nc} \)) and
  - either "use it or lose it", i.e., the scheduler is not obligated to distribute unreserved \((1 - \sum_{n} f_{nc})\) or currently reserved-but-unused resources (owing to idle flows/queues) to currently nonidling flows.

- This perspective may be that of a public, for-profit utility (ISP, cloud services provider).

- **Exercise:** How could DRR above be modified to limit output burstiness as SVC?
Fair scheduling with timestamps - SCFQ and SFQ

- There is a significant literature on fair scheduling including timestamp based
  - Self-Clocked Fair Queueing (SCFQ) [Golestani INFOCOM’94], and
  - Start-Time Fair Queueing (SFQ) [Goyal et al. SIGCOMM’96].
- They are “fair” in that they track work-conserving, rate based scheduling of a Generalized Processor Sharing (GPS) fluid traffic flow model that accounts for queue idle times.
- Let the start-time and finish-time of the $i^{th}$ packet of flow/queue $n$ be, respectively,
  \[ S_{n,i} = \max\{v(a_{n,i}), F_{n,i-1}\} \quad \text{and} \quad F_{n,i} = S_{n,i} + \frac{l_{n,i}}{f_n c}, \]
  where $v$ is the virtual time and $F_{n,0} = 0$.
- Under SCFQ, $v(t)$ is the finish-tag of the packet in service (or most recently in service if $t$ is during a server idle period), and the head-of-queue packets with the smallest finish-tag are serviced first.
- Under SFQ, $v(t)$ is the start-tag of the packet in service (or most recently in service if $t$ is during a server idle period), and the head-of-queue packets with the smallest start-tag are serviced first.

SCFQ and SFQ - fair scheduling performance & cost

- For SCFQ and SFQ, over an interval of time $T$ where both queues $n$, $m$ are never idle,
  \[ \forall n, m, T, \quad \left| \frac{D_n(T)}{f_n} - \frac{D_m(T)}{f_m} \right| \leq \left( \frac{1}{f_n} + \frac{1}{f_m} \right) L_{max}, \]
- Again, SCFQ and SFQ have $O(1)$ enqueue complexity because they use a simple virtual time function to compute the start-times and finish-times,
- but they have $O(\log(N))$ dequeue complexity, again because to decide which head-of-queue packet to serve, the one with smallest timestamp needs to be determined.
Deterministic network calculus

• A more powerful formulation of guaranteed service is given by the service curve concept on which a kind of “network calculus” is based for determining delay and jitter bounds for a packet flow as it traverses a series of multiplexed FIFO queues, each of which may be shared with other flows.


• Network calculus provides a succinct way
  – to describe the burstiness of job/packet arrival flows
  – and the service guarantees provided by tandem (lossless) multiplexers/schedulers,
  – to derive bounds on delay and queue backlog.

• The burstiness curves are typically piecewise linear in practice - recall token/leaky buckets.

• Extensions to time-varying envelopes have been developed.

• Extensions to stochastic settings (so that packet-by-packet policing is not possible), will be discussed later.

Convolution and deconvolution operators

• We will now revisit some previous calculations via the convolution \( \otimes \) and deconvolution \( \ominus \) operators,
  – as used in “min-plus” algebras
  – on flows, i.e., initially zero and non-decreasing (and hence non-negative) functions of continuous time \( t \in \mathbb{R}^+ := [0, \infty) \) (or \( t \in \mathbb{Z}^+ \) if time is discrete), i.e., \( X \) is a flow if
    \[
    \forall t \geq v \geq 0, \quad X(t) \geq X(v) \quad \text{with} \quad X(0-) = 0,
    \]
    e.g., cumulative arrivals or departures or maximum/minimum service.

• For any two flows \( X \) and \( Y \) at time \( t \geq 0 \):
  – \( X \) convolved with \( Y \) is, \( \forall t \geq 0, \)
    \[
    (X \otimes Y)(t) = \min_{0 \leq v \leq t} X(v) + Y(t - v) = (Y \otimes X)(t).
    \]
  – \( X \) deconvolved with \( Y \) is, \( \forall t \geq 0, \)
    \[
    (X \ominus Y)(t) = \max_{s \geq 0} X(t + s) - Y(s).
    \]
Basic properties of convolution and deconvolution

- $X \leq Y$ means $\forall t \geq 0, X(t) \leq Y(t)$.

- $X = \min\{Y \mid Y \in \mathcal{G}\}$ means $\forall t \geq 0, X(t) = \min_{Y \in \mathcal{G}} Y(t)$, i.e., $X$ is the largest such that $X \leq Y \forall Y \in \mathcal{G}$.

- The identity function of the convolution and deconvolution operators is the infinite step,
  
  $$u_\infty(t) = \begin{cases} 0 & \text{if } t \leq 0 \\ +\infty & \text{if } t > 0 \end{cases}$$

  i.e., for all flows $X$, $X \otimes u_\infty = X$ and $X \ominus u_\infty = X$.

- Convolution is commutative and associative.

- One can directly show that for all flows $X, Y, Z$:
  
  $$(X \otimes Y) \otimes Z = X \otimes (Y \otimes Z)$$
  
  $$X \otimes Y = \min\{Z \mid Z \otimes Y \geq X\}$$

  $\Rightarrow (X \otimes Y) \otimes Y \geq X$

  $\Rightarrow (X \otimes Y) \otimes Z \leq X \otimes (Y \otimes Z)$.

- **Exercise:** Prove the above identities.

---

**Exercise:** Delay function

- Define the delay function
  
  $$\Delta_d(t) = \begin{cases} 0 & \text{if } t \leq d, \\ +\infty & \text{if } t > d. \end{cases}$$

- That is, $\Delta_d(t) = u_\infty(t - d)$

- **Exercise:** Show that for any function $f$ and a constant $d \geq 0$,
  
  $\forall t, f(t - d) = (f \otimes \Delta_d)(t)$. 

---

© December 14, 2018 George Kesidis

59

© December 14, 2018 George Kesidis

60
Flow burstiness curves (traffic envelopes)

- Consider an initially empty, lossless queue in a network device with cumulative arrivals and departures over \([0, t]\) respectively denoted \(A(t)\) and \(D(t)\).

- A flow \(A\) is said to have burstiness bounded by (or an upper envelope) \(b_in\) if
  \[
  \forall t \geq v \geq 0, \quad A(t) - A(v) \leq b_in(t - v) \quad \Leftrightarrow \quad A \leq A \otimes b_in,
  \]
  more succinctly denoted as \(A \ll b_in\) (recall \(b_in\) is also non-increasing).

- Note that this is a bound on arrivals over any time-interval \((v, t]\).

- For example, if \(A\) is the output of dual token-bucket regulators, then \(b_in\) is piecewise-linear:
  \[
  b_in(r) = \min\{\sigma + \rho r, \epsilon + \pi r\}
  \]
  where the maximum “burst size” \(\sigma > \epsilon \geq 0\) (\(\epsilon\) is small) and the peak rate is greater than the “sustainable” rate, \(\pi > \rho\).

- In the following, we assume the arrival flow \(A \ll b_in\).

Service curves

- Now consider a single (lossless) queue of a multiplexer (mux) within a network device (e.g., a router).

- \(A\) and \(D\) respectively are the queue’s arrival departure flows.

- The cumulative departures \(D\) of a given queue depends on any service guarantees as scheduled by the mux and possibly (in the case of non-idling service) how the other queues are busy.

- If \(Q(0) = 0\), then the queue backlog at time \(t \geq 0\) is
  \[
  Q(t) = A(t) - D(t).
  \]

- In the special case where the queue receives exact, deterministic service at rate \(c > 0\):
  \[
  \forall t \geq 0, \quad Q(t) = \max_{0 \leq r \leq t} A(t) - A(r) - (t - r)c
  \]
  \[
  \Rightarrow D(t) = \min_{0 \leq r \leq t} A(r) + (t - r)c = (A \otimes s_0)(t),
  \]
  where the “service flow” \(s_0(t) = tc\) for all \(t \geq 0\).

- More generally, a scheduler is said to give the queue a minimum service-curve \(s_{\text{min}}\), respectively maximum service-curve \(s_{\text{max}}\), if for all arrival flows \(A\),
  \[
  D \geq A \otimes s_{\text{min}}, \quad \text{respectively} \quad D \leq A \otimes s_{\text{max}}.
  \]
Guaranteed rate property and minimum service curve

- **Exercise:** If a scheduler with guaranteed-rate property parameter $\mu$ (SVC has $\mu = \frac{L_{\max}}{c}$) for a queue with bandwidth allocation $c$, show that the queue has minimum service-curve
  \[ s_{\min}(t) = \max\{ct - c\mu, 0\}. \]
- Cruz’s Service-Curve Earliest Deadline First (SCED+) scheduler was designed to achieve output service-curves.

Output burstiness

- **Theorem:** If $A \ll b_{in}$ and the initially empty queue has minimum service-curve $s_{\min}$, then
  \[ D \ll b_{out} := b_{in} \odot s_{\min}. \]
- **Proof:** $\forall t \geq 0$:
  \[
  D(t) \leq A(t) \\
  \leq \min_{0 \leq v \leq t} A(v) + b_{in}(t - v) \\
  \leq \min_{0 \leq v \leq t} A(v) + \min_{0 \leq t - v \leq t-v} \left( s_{\min}(t - v - r) + \max_{x \geq 0} b_{in}(x + r) - s_{\min}(x) \right) \\
  = \min_{0 \leq r \leq t} \min_{0 \leq v \leq t-r} A(v) + s_{\min}(t - v - r) + b_{out}(r) \\
  = \min_{0 \leq r \leq t} b_{out}(r) + \min_{0 \leq v \leq t-r} A(v) + s_{\min}(t - v - r) \\
  \leq \min_{0 \leq r \leq t} b_{out}(r) + D(t - r)
  \]
  where we have switched the order of minimization for the first equality.
- Thus,
  \[ D \ll b_{out}. \]
Output burstiness via convolution and deconvolution

• We now redo the previous proof using convolution notation and basic properties:

\[
D \leq A \\
\leq A \otimes b_{\text{in}} \\
\leq A \otimes (s_{\text{min}} \otimes (b_{\text{in}} \ominus s_{\text{min}})) \\
\equiv A \otimes (s_{\text{min}} \otimes b_{\text{out}}) \\
\equiv (A \otimes s_{\text{min}}) \otimes b_{\text{out}} \\
\leq D \otimes b_{\text{out}}
\]

• **Exercise:** Prove the extension of this result to also account for maximum a service-curve \( s_{\text{max}} \) of the queue:

\[
D \ll (b_{\text{in}} \otimes s_{\text{max}}) \ominus s_{\text{min}}.
\]

---

Virtual delay processes (for arrivals) and delay jitter bound

• For a queue with arrival flow \( A \) and departure flow \( D \), at time \( t \geq 0 \),
  - the queue backlog is \( Q(t) = A(t) - D(t) \), i.e., ”vertical” difference between the flows, and
  - the virtual delay for a hypothetical arrival at time \( t \) is \( D^{-1}(A(t)) - t \), where \( D^{-1}(a) \)
    is the smallest time \( t \) such that \( D(t) = a \), i.e., ”horizontal” difference between the flows.

• Note that the virtual delay process does not depend on arrivals after \( t \) under FIFO queuing, and recall our discussion of a virtual-queue policer.
Virtual delay processes and delay jitter bound - theorem

• **Theorem:** If a queue has arrival flow $A \ll b_{in}$, minimum service-curve $s_{min}$, and maximum service-curve $s_{max}$, then

$$
\forall t \geq 0, \quad A(t - d_{min}) \geq D(t) \geq A(t - d_{max}),
$$

where

$$
d_{min} = \max\{x \geq 0 : s_{max}(x) = 0\},
$$

$$
d_{max} = \min\{z \geq 0 : s_{min}(x) \geq (b_{in} \otimes \Delta_{z})(x) = b_{in}(x - z) \forall x \geq 0\}.
$$

• Re. the virtual delays, this theorem implies that, $\forall t \geq 0$,

$$
D^{-1}(A(t - d_{min})) - (t - d_{min}) \geq d_{min} \quad \text{and} \quad D^{-1}(A(t - d_{max})) - (t - d_{max}) \leq d_{max}.
$$

Maximum delay - remarks

• Note that $d_{max}$ is the largest horizontal difference between $b_{in}$ and $s_{min}$.

• Also note that, equivalently,

$$
d_{max} = \min\{z \geq 0 : b_{in}(x - z) - s_{min}(x) \leq 0 \forall x \geq 0\} = \min\{z \geq 0 : \max_{x \geq 0} b_{in}(x - z) - s_{min}(x) \leq 0\} = \min\{z \geq 0 : (b_{in} \oplus s_{min})(-z) \leq 0\}.
$$

• Moreover, $s_{max} \leq \Delta_{d_{min}}$. 
Delay jitter bound - Proof via convolution notation

First,
\[
\forall t \geq 0, \quad D(t) \geq (A \otimes s_{\min})(t) \\
\geq (A \otimes (b_{\min} \otimes \Delta_{d_{\max}}))(t) \\
= ((A \otimes b_{\min}) \otimes \Delta_{d_{\max}})(t) \\
\geq (A \otimes \Delta_{d_{\max}})(t) \\
= A(t - d_{\max}).
\]

Finally,
\[
\forall t \geq 0, \quad D(t) \leq (A \otimes s_{\max})(t) \\
\leq (A \otimes \Delta_{d_{\min}})(t) \\
= A(t - d_{\min}).
\]

End-to-end network calculus - exercise

- Consider the tandem queues of static flow-routes across multiple network devices.
- Suppose a given end-to-end flow with network arrivals \( A \ll b_{\min} \) visiting FIFO queues indexed \( j \) on its path, where each queue \( j \) has minimum and maximum service curves respectively \( s_{\min,j} \) and \( s_{\max,j} \), and each queue \( j \) handles only the given flow.
- Extend the previous results on delay and output jitter from a single queue to the entire network of tandem queues as experienced by the given flow.
- See https://www.comm.utoronto.ca/~jorg/teaching/ece1545/
Dynamic routing

- Routing algorithms are highly distributed/decentralized in their response to network state because network operating conditions potentially involve:
  - a large scale with respect to traffic volume or geography or both, and/or
  - high variability in the traffic volume both at packet and connection/call level on short time-scales (possibly due in part to the routing algorithm itself), and/or
  - potentially high variability in the network topology due to, for example, node mobility, channel conditions, or node or link removals because of faults or energy depletion.

Additive path costs

- Routing algorithms often assume that costs (or “metrics”) $C_r$ of paths/routes $r$ are additive, i.e.,
  \[ C_r = \sum_{l \in r} c_l, \]
  where $c_l$ represents the cost of link $l$.

- Such nonnegative link costs include Boolean hops, i.e., $c_l = 1$ for all active links $l$ (leading to path costs $C_r$ that are hop counts as used in the Internet), and those based on estimates of access delays at the transmitting node of the link.
Path costs based on bottlenecks

- Alternatively, path costs could be based on the bottleneck link on the path, i.e.,
  \[ C_r = \max_{l \in r} c_l. \]

- In a multihop wireless context, such link costs include those based on residual energy \( e_l \) of transmitting nodes (in a multihop wireless context), e.g.,
  \[ e_l = \frac{1}{e_l} \]
  or an estimate of the lifetime of the transmitting node of the link.

Hybrid path costs

- More complex two-dimensional link metrics of the form \((c^x, c^y)\) may be employed to consider more than one quantity simultaneously, e.g.,
  - delay and energy, or
  - hop count and BGP policy domain factors.

- One can define (lexicographic) order
  \[ (c^x_1, c^y_1) \leq (c^x_2, c^y_2) \]
  to mean
  \[ c^x_1 \leq c^x_2 \text{ or } \{ c^x_1 = c^x_2 \text{ and } c^y_1 \leq c^y_2 \} \]
  and define the cost of path composed of links indexed 1 and 2 as
  \[ (c^x_1 + c^x_2, \max\{c^y_1, c^y_2\}) \]

- For example,
  - if \( c^x \in \{1, \infty\} \) in order to count hops of a path
  - and \( c^y \) is based on the residual energy of the transmitting node,
  - then the chosen paths will be those with the highest bottleneck energy among those with the shortest hop count to the destination.
Hybrid path costs - examples

- Or one can determine optimal paths
  - according to one metric (the primary objective) and
  - choose among these paths conditional on another metric (the secondary objective) being less than a threshold.

- For instance, suppose the primary objective is to minimize (bottleneck) energy costs and suppose a route \( r \) has \( C_x^r \) hops and \( C_y^r \) energy cost.

-Appending link \( l \) to \( r \), \( r' = r \cup \{l\} \), will be considered based on costs
  \[
  (C_x^{r'}, C_y^{r'}) = (c_x^l + C_x^r, \max\{c_y^l, C_y^r\})
  \]
  if \( c_x^l + C_x^r < \theta_x \) for some threshold \( \theta_x > 0 \).

- Otherwise it will set \( (C_x^{r'}, C_y^{r'}) = (\infty, \infty) \) and, consequently, the network will not use route \( r' \) nor any route \( r^* \) that uses \( r' \) (i.e., \( r' \subset r^* \)).

- Similarly, the network can find routes with minimal hop counts (primary objective) while avoiding any link with energy cost \( c_y \geq \theta_y > 0 \) (i.e., the residual energy of the transmitting node of the link \( e \leq 1/\theta_y \)).

Optimal routing frameworks: link states

- Within an autonomous system (AS) of the Internet, it may be feasible for routers to periodically flood the network with their link-state information.

- So, each router can build a picture of the entire layer-3 AS graph from which loop-free optimal (minimal-hop-count) intra-AS paths can be found by OSPF and ISIS interior-gateway routing protocols (IGPs) based on Dijkstra’s algorithm.

- A hierarchical OSPF framework can be employed on the component “areas” of a large AS.

- Under OSPF, each router \( z \) will forward packets ultimately destined to router \( v \) according along the subpath \( p \) to a neighboring (predecessor) router \( r_p \) of \( v \) that is
  \[
  \arg \min_p C_p + c_{(r_p, v)};
  \]
  where \( C_p \) is the path cost (hop count) of \( p \).

- Dijkstra’s algorithm works iteratively at each node \( z \) based on a consistent graph of the AS owing to flooded link-states:
  - optimal paths to nodes are found in order of increasing distance to \( z \),
  - and so a spanning tree rooted at \( z \) is built out from its leaves.
Optimal routing frameworks: distance vectors

- A distributed distance-vector approach involves computing (at \( z \)) optimal path cost from \( z \) to \( S \) as

\[
\arg \min_w c(z,w) + C(w,S),
\]

where \( c(z,w) \) is the single-hop/link cost of the link \((z,w)\) between \( z \) and its neighboring node \( w \), and \( C(w,S) \) is \( w \)'s current path cost to \( S \) as advertised to \( z \).

- Only nearest-neighbor communication is generally more scalable than flooding.

- In the Internet, the BGP and the IGP RIP are distance vector based.

- BGP maintains whole-path vectors to avoid loops and implement important inter-domain routing policies (that may take precedence over distance).

- Also, BGP employs route reflectors, poison reverse, dynamic minimum route advertisement interval (MRAI) adjustments, and other mechanisms to dampen the frequency of route updates, reduce responsiveness (to, e.g., changing traffic conditions, link or node withdrawals), and improve stability/convergence properties.

- So, both Dijkstra’s and the distributed Bellman-Ford algorithms use the fundamental “principle of optimality” (easily proved by contradiction): all subroutes of any optimal (minimum cost) route are themselves optimal.

---

Example - shortest path on a graph

- Suppose we are planning the construction of a highway from city A to city K.

- Different construction alternatives and their “edge” costs \( g \geq 0 \) between directly connected cities (nodes) are given in the following graph.

- The problem is to determine the highway (edge sequence) with the minimum total (additive) cost.

![Graph](image_url)
Bellman's principle of optimality - exercise

- If C belongs to an optimal (by edge-additive cost $J^*$) path from A to B, then the sub-path A to C and C to B are also optimal,
- *i.e.*, any sub-path of an optimal path is optimal (easy proof by contradiction).

![Diagram showing optimal paths between A, C, and B]

- Dijkstra’s algorithm uses the predecessor node of the destination (path penultimate node) & is based on complete link-state (edge-state) info consistently shared among all nodes:
  \[ J^*(A, B) = \min_C \{ J^*(A, C) + g(C, B) | C \text{ is a predecessor of } B \}, \]
  *i.e.*, C and B are adjacent nodes in the graph (endpoints of the same edge).

- The distributed Bellman-Ford algorithm uses the successor node of the path origin and only nearest-neighbor distance-vector information sharing:
  \[ J^*(A, B) = \min_C \{ g(A, C) + J^*(C, B) | C \text{ is a successor of } A \} \]

---

Review of Elements of Probability

- The sample space $(\Omega, \mathcal{F}, P)$.
- Random variables and their distributions.
- The law of large numbers.
- See slidedeck at http://www.cse.psu.edu/~kesidis/teach/Prob-4.pdf
Stochastic Processes - Introduction

• A stochastic (or random) process is a set of random variables indexed by a parameter (e.g., time, location).

• If the time parameter takes values only in $\mathbb{Z}^+$ (or any other countable subset of $\mathbb{R}$), the stochastic process is said to be discrete time; i.e.,

  $$\{X(t) \mid t \in \mathbb{Z}^+\}.$$  

• If the time parameter $t$ takes values over $\mathbb{R}$ or $\mathbb{R}^+$ (or any real interval), the stochastic process is said to be continuous time.

• The dependence on the sample $\omega \in \Omega$ can be explicitly indicated by writing $X_\omega(t)$.

• For a given sample $\omega$, the random object mapping $t \mapsto X_\omega(t)$, for all $t \in \mathbb{R}^+$ say, is called a sample path of the stochastic process $X$. 
Stochastic Processes - Introduction (cont)

- The state space of a stochastic process is simply the union of the strict ranges of the random variables \( \{ X(t) \mid t \in \mathbb{Z}^+ \} \).

- We will restrict our attention to stochastic processes with countable state spaces, typically \( \mathbb{Z}, \mathbb{Z}^+ \), or a finite subset \( \{0, 1, 2, \ldots, K\} \).

- Of course, this means that the random variables \( X(t) \) are all discretely distributed.

- We will also focus on continuous-time so that queueing systems we will consider will be a little easier to analyze.

Finite-dimensional distributions of a stochastic process

- Consider a stochastic process
  \[ X = \{ X(t) \mid t \in \mathbb{R}^+ \} \]
  with state space \( \mathbb{Z}^+ \).

- Let \( p_{t_1, t_2, \ldots, t_n} \) be the joint PMF of \( X(t_1), X(t_2), \ldots, X(t_n) \) for some finite \( n \) and different \( t_k \in \mathbb{R}^+ \) for all \( k \in \{1, 2, \ldots, n\} \), i.e.,
  \[ p_{t_1, t_2, \ldots, t_n}(x_1, x_2, \ldots, x_n) = P(X(t_1) = x_1, X(t_2) = x_2, \ldots, X(t_n) = x_n). \]

- This is called an \( n \)-dimensional distribution of \( X \).

- The family of all such joint PMFs is called the set of finite-dimensional distributions (FDDs) of \( X \).
Consistent finite-dimensional distributions

- A family of FDDs (on state-space \( \mathbb{Z}^+ \), with time \( t \in \mathbb{R}^+ \)) are consistent if one can marginalize (reduce the dimension) and obtain another, e.g.,

\[
p_{t_1, t_2, t_4}(x_1, x_2, x_4) \equiv \sum_{x_3 \in \mathbb{Z}^+} p_{t_1, t_2, t_3, t_4}(x_1, x_2, x_3, x_4).
\]

- Recall that consistency ought to hold simply because

\[
P(A) = \sum_{x_3 \in \mathbb{Z}^+} P(A, X(t_3) = x_3), \text{ where } A := \{X(t_1) = x_1, X(t_2) = x_2, X(t_3) = x_3\}.
\]

- Beginning with a family of consistent FDDs, Kolmogorov’s extension (or “consistency”) theorem is a general result demonstrating the existence of a stochastic process \( t \to X_\omega(t), \omega \in \Omega \), that possesses them.

Stationarity of a stochastic process

- A stochastic process \( X \) is said to be (strongly) stationary if all of its FDDs are time-shift invariant.

- That is, if

\[
p_{t_1, t_2, \ldots, t_n} \equiv p_{t_1 + \tau, t_2 + \tau, \ldots, t_n + \tau}
\]

for all integers \( n \geq 1 \), all \( t_k \in \mathbb{R}^+ \), and all \( \tau \in \mathbb{R} \) such that \( t_k + \tau \in \mathbb{R}^+ \) for all \( k \).
Stationary queues

- Consider the $i^{th}$ job arriving at time $T_i$ to a FIFO single-server, nonidling queue.

- The departure time of this job is given by
  \[ V_i = T_i + W(T_i), \]
  where $W$ is the queue’s workload process.

- If the queue is stationary, the sojourn times of the jobs are identically distributed.

- Indeed, suppose we are interested in the distribution or just the mean of the job sojourn times.

- One is tempted to identify the distribution of the sojourn times $V - T$ with the stationary distribution of $W$; because of the “PASTA” rule, this gives the correct answer for the M/M/1 queue, as discussed later.

- But in general the distribution of $W(T_n -)$ (i.e., the distribution of the $W$ process viewed just before a “typical” job arrival time $T_n$) is not equal to the stationary distribution of $W$ (i.e., viewed at at typical time).

Loynes’ construction of a stationary queue viewed at finite time ($0$)

- Consider a stationary marked point process on $\mathbb{R}$, where a mark $S$ is a random variable associated with an arrival time $T$ (point).

- The point process is stationary if for any interval of time $[r, t] \subseteq \mathbb{R}$, $r < t$, the distribution of the number and values of the marked points $(T_i - r, S_i)$ therein (i.e., $r \leq T_i \leq t$) depends on $t$ and $r$ only through $t - r$.

- Assume that the marks $S$ are the service times of the arrivals by a unit server (one unit of work per second), which do not depend on future arrivals/marks (i.e., are non-anticipative, causal).
• Suppose that the arrivals commence at some negative time \( r < 0 \), i.e., ignore arrivals at times \( T < r \).

• So that the work-to-be-done of a single-server queue at time 0 is

\[
W_r(0) = \max_{r \leq t \leq 0} \sum_{i : t \leq T_i \leq 0} S_i - ct,
\]

where \( c \) is the constant service rate of the queue and \( S_i \) is the service time of the \( i \)th job arriving at time \( T_i \).

• Note that as \( r \to -\infty \), \( W_r(0) \) monotonically increases.

• Loynes proved that if the arrival intensity is finite, i.e., \( \lambda = (\mathbb{E}(T_i - T_{i-1}))^{-1} < \infty \), and the queue is stable, i.e., \( c > \lambda \mathbb{E}S_i \), then this limit exists and is finite, i.e.,

\[
\lim_{r \to -\infty} W_r(0) \uparrow W(0) < \infty \text{ a.s.}
\]

the stationary queue on \( \mathbb{R} \) viewed at a typical (finite) time 0.

stationary queueing system viewed at typical time vs at typical job

• We will now explore the relationship between the stationary distribution of a queueing system (i.e., as viewed from a typical time) and the distribution of the queueing system at the arrival time of a typical job - we now illustrate the potential difference.

• Consider a stationary and ergodic point process on \( \mathbb{R} \) whose interarrival times \( \tau \) are discretely distributed as

\[
P(\tau = 5) = \frac{1}{4} \quad \text{and} \quad P(\tau = 10) = \frac{3}{4}.
\]

• Also consider a large interval of time \( H \gg 1 \) spanning \( N \) consecutive interarrivals.

• Consider an interarrival interval \( T_1 - T_0 \) viewed at a typical time 0, i.e., by definition \( T_0 < 0 \leq T_1 \) a.s.

• The probability of selecting such an interval of length say 5 is equal to the fraction of interarrivals of length 5 that cover \( H \).

• That is, since \( H \gg 1 \), by the law of large numbers \( H \approx N(5 \cdot \frac{1}{4} + 10 \cdot \frac{3}{4}) \), and so

\[
P(T_1 - T_0 = 5) = \frac{N \cdot 5(1/4)}{N(5(1/4) + 10(3/4))} = \frac{1}{7} \neq \frac{1}{4} = P(\tau = 5)
\]

• Later we’ll see that \( T_1 - T_0 \sim \tau \) when job arrivals are Poisson (PASTA).
A lossless, stationary, stable queue: input rate equals output rate

- Let $\lambda$ be the mean arrival rate and $\mu$ the mean service rate of jobs (data packets) to a stable queue, i.e.,
  $$\mu > \lambda.$$  

- **Theorem:** For a stable, lossless and stationary queue, the mean (net) arrival rate equals the mean departure rate in steady state, i.e.,
  $$\lambda := \lim_{t \to \infty} \frac{A(0, t)}{t} = \lim_{t \to \infty} \frac{D(0, t)}{t},$$
  where $A(0, t]$ and $D(0, t]$ are the cumulative arrivals and departures over $(0, t]$, respectively.

- **Proof:** The queue is stable implies that $Q(t)/t \to 0$ almost surely as $t \to \infty$.

- Since
  $$Q(0) + A(0, t] = Q(t) + D(0, t],$$
  - Dividing this equation by $t$ and letting $t \to \infty$ gives the desired result. \(\square\)

- **Note:** The mean departure rate of the stable queue ($\lambda$) is less than $\mu$, as the server is active only when $Q > 0$.

**Little’s result:** $L = \lambda W$

- Consider a causal (nonanticipative), stationary and and ergodic, lossless, and stable queueing system.

- Partition an interval of time of length $T \gg 1$ so that the number of jobs in the system is constant in each subinterval.

- That is, jobs arrive or depart the queueing system only at partition boundaries.

- Let $J$ be the number of departures of jobs over $[0, T]$.

- Let $t_k$ be the duration of the $k^{th}$ interval, so that $\sum_{k=1}^{K} t_k = T$.

- Let $n_k$ be the average number of jobs in the system during the $k^{th}$ interval.

- Thus, the time-average number of jobs in the system over $[0, T]$ is
  $$L \approx \frac{1}{T} \sum_{k=1}^{K} n_k t_k = \frac{1}{T} \sum_{k=1}^{K} n_k t_k.$$
Little’s result: \( L = \lambda W \) (cont)

- Assume any jobs initially in the system (i.e., \( Q(0) \)) or any that remain (i.e., \( Q(T) \)) are negligible compared to \( J \) when \( T \gg 1 \); so \( J \) is approximately the number of arrivals over \( T \) too.

- Thus,

\[
\lambda \approx J/T.
\]

- Similarly, the mean sojourn time (queueing delays plus service times) of jobs in the queueing system is

\[
W \approx \frac{1}{J} \sum_{k=1}^{K} n_k t_k,
\]

where the numerator is the total sojourn time of all jobs in the interval \([0, T]\).

- By substitution, we arrive at Little’s result: \( L = \lambda W \).

- A rigorous proof of Little’s result is based on a powerful conservation law for stationary marked point-processes, Campbell’s theorem.

Little’s result - discussion and example

- To reiterate, Little’s result relates
  - the average number of jobs in the stationary lossless queueing system (i.e., the average number of jobs viewed at a typical time 0)
  - to the mean sojourn time of a typical job.

- For example: We will see that the mean number of jobs in a stationary “M/M/1” queue is

\[
L = \frac{\rho}{1 - \rho},
\]

where \( \rho = \lambda/\mu < 1 \) is the traffic intensity.

- By Little’s result, the mean workload in the M/M/1 queue upon arrival of a typical job (i.e., the mean sojourn time of a job) is

\[
W = \frac{L}{\lambda} = \frac{1}{\mu - \lambda}.
\]
Little’s result: mean server busy-time

- Now consider again a lossless, FIFO, single-server queue \( Q \) with mean interarrival time of jobs \( 1/\lambda \) and mean job service time \( 1/\mu < 1/\lambda \).
- \textit{i.e.}, mean job arrival rate \( \lambda \) and mean job service rate \( \mu > \lambda \).
- Suppose the queue and arrival process are stationary at time zero.
- The following result identifies the \textit{traffic intensity} \( \lambda/\mu \) with the fraction of time that the stationary queue is busy.

Little’s result: mean server busy-time (cont)

- \textbf{Theorem}: For a stationary and stable \((\lambda < \mu)\) queue \( Q \),
  \[ P(Q(0) = 0) = 1 - \frac{\lambda}{\mu} \]
- \textbf{Proof}: Consider the server separately from the waiting room.
- Since the mean \textit{departure} rate of the waiting room is \( \lambda \) too, Little’s result implies that the mean number of jobs in the server is \( L = \lambda/\mu \).
- Finally, since the number of jobs in the server is Bernoulli distributed (with parameter \( L \)), the mean corresponds to the probability that the server is occupied (has one job) in steady state. \( \Box \)
- As above, note that the \textit{mean} departure rate is
  \[ \mu \cdot P(Q > 0) + 0 \cdot P(Q = 0) = \mu \cdot \rho = \lambda. \]
Recall that a scheduler acting on a queue is said to offer a service-curve $\beta$ if

- $\beta$ is nondecreasing with $\beta(0) = 0$,
- for all cumulative arrivals $A$ and for all times $t \geq 0$ such that the queue is always backlogged over $[0, t]$, the cumulative departures $D$ from that queue satisfy

$$D[0, t] \geq \min_{0 \leq s \leq t} A[0, s] + \beta(t - s) = \min_{0 \leq s \leq t} A[0, t - s] + \beta(s).$$

Now consider a queue occupancy process $Q$ with cumulative arrivals $A$ and a service rate of exactly $\rho$ bytes/s.

$A$ is said to have generalized stochastically bounded burstiness (gSBB, or strong SBB) with bound $f_\rho$ at $\rho$ if

$$\forall t \geq 0, \ P(Q_{\rho}(t) \geq \sigma) \leq f_\rho(\sigma),$$

where $f_\rho \geq 0$ is a nonincreasing function with $f_\rho(0) = 1$ and, as before, $Q(0) = 0$ and, for $t > 0$,

$$Q(t) = \max_{0 \leq s \leq t} A(s, t) - \rho(t - s);$$


Alternatively, we can work with a weaker definition: define $A$ as having (weak) SBB by $f_\rho$ at $\rho$ if

$$\forall t \geq 0, \ P(A(s, t) - \rho(t - s) \geq \sigma) \leq f_\rho(\sigma),$$


Probabilistic service curves - gSBB (cont)

- We denote $A \ll (\rho, f)$ if $A$ has gSBB with bound $f$ at constant service rate $\rho$.
- Clearly, $A \ll (\rho, f)$ implies $A \ll (r, f)$ for $r > \rho$.
- Also note that this definition reduces to the $(\sigma^*, \rho)$ constraint when $f(\sigma) \equiv u(\sigma - \sigma^*)$.
- Note that, unlike the gSBB, the deterministic $(\sigma, \rho)$ constraint is policeable on a packet-by-packet basis.

**Theorem:** For a queue with service curve $\beta$, if arrivals $A \ll (\rho, f)$, then departures $D \ll (\rho, g)$, where

$$g(x) \equiv f \left( x + \min_{s \geq 0} \{ \beta(s) + \rho t \} \right).$$

![Diagram](image)

**Proof:** Consider the backlog of a queue $\bar{Q}$ with arrivals $D$ and service rate $\rho$, so that

$$Q(t) = \max_{0 \leq s \leq t} D[0, t] - D[0, s] - (t - s) \rho$$

$$\leq \max_{0 \leq s \leq t} A[0, t] - \left( \min_{0 \leq u \leq s} A[0, u] + \beta(s - u) \right) - (t - s) \rho$$

$$= \max_{0 \leq s \leq t} \max_{0 \leq u \leq s} A[0, t - s + u] - \rho(t - s + u) + \rho u - \beta(u)$$

$$\leq Q(t) + \max_u \rho(u) - \beta(u).$$

Applying this inequality to the definition of gSBB proves the theorem. □

**Exercise:** Extend this theorem to an end-to-end result for a flow crossing tandem schedulers each giving the flow different service curves $\beta$. 

Probabilistic service curves - gSBB (cont)
Flow-balance equations - preliminaries

- Consider a stationary system consisting of a group of $N \geq 2$ lossless, single-server, work-conserving queueing stations.
- Jobs at the $n^{th}$ station have a mean required service time of $1/\mu_n$.
- The job arrival process to the $n^{th}$ station is a superposition of $N + 1$ component arrival processes.
- Jobs departing the $m^{th}$ station are forwarded to and immediately arrive at the $n^{th}$ station with probability $r_{m,n}$.
- Also, with probability $r_{m,0}$, a job departing station $m$ leaves the queueing network forever; here we use station index 0 to denote the world outside the network.
- Clearly, for all $m$,
  $$\sum_{n=0}^{N} r_{m,n} = 1.$$  
- Arrivals from the outside world arrive to the $n^{th}$ station at rate $\Lambda_n$; it's these interactions with the outside world that make the network open.

Flow balance equations (cont)

- Let $\lambda_n$ be the total arrival rate to the $n^{th}$ station.
- These are found by solving the so-called flow balance equations which are based on the notion of conservation of flow and require that all queues are stable, i.e.,
  $$\forall n, \mu_n > \lambda_n.$$  
- Since the mean arrival rate equals that of the mean departure rate, the flow balance equations are,
  $$\lambda_n = \Lambda_n + \sum_{m=1}^{N} \lambda_m r_{m,n}, \forall n \in \{1, 2, ..., N\}.$$  
- Note that the flow balance equations can be written in matrix form:
  $$\Lambda^T(I - R) = \Lambda^T,$$
  where the $N \times N$ matrix $R$ has entry $r_{m,n}$ in the $m^{th}$ row and $n^{th}$ column.
- Note: We could define the total throughput of the system $\lambda_0 = \sum_{m=1}^{N} \Lambda_m$ so that $r_{0,m} = \Lambda_m/\lambda_0$.  

© December 14, 2018 George Kesidis

101

102
Flow balance equations - solution requirements

- Thus,
  \[ \Lambda^T = \Lambda^T (I - R)^{-1}. \]
- Again, we are assuming that \( \lambda < \mu \) for stability.
- Also, we clearly require that \( \det(I - R) \neq 0 \), i.e., \( I - R \) is invertible.
- This (and stability and stationarity) requires that \( r_{m,0} > 0 \) for some station \( m \), i.e., jobs can exit the network and don’t on-average accumulate in it.
- Otherwise,
  - on average work accumulates in the system and so it cannot be stationary, and
  - \( R \) would be a stochastic matrix (all entries nonnegative and all rows sum to 1) so that 1 is an eigenvalue of \( R \) and, therefore, 0 is an eigenvalue of \( I - R \), i.e., \( I - R \) is not invertible.
- Note: It is possible to define stationary queueing systems that are closed, i.e., with \( r_{n,0} = 0 = r_{0,n} \) for all \( n \); in such systems there are no such stability requirements.

Flow balance equations - solution requirements

- We can also write a flow balance equation between the outside world and the queueing network as a whole by summing over the individual queueing stations \( n \in \{1, \ldots, N\} \) to get:
  \[ \sum_{n=1}^{N} \Lambda_n = \sum_{n=1}^{N} \lambda_n r_{n,0}, \]
  i.e., the total flow into the queueing network equals the total flow out of the network as the previous theorem.
- The flow balance equations hold in great generality.
- In the following, we will apply them to derive the stationary distribution of a special network with Markovian dynamics.
This example network has three lossless FIFO queues, queues 1 and 2 respectively have exogenous arrival rate $\Lambda_1$ and $\Lambda_2$ jobs per second.

- The mean service time at queue $k$ is $1/\mu_k$.
- The nonzero job routing probabilities are $r_{12} = r_{13} = \frac{1}{2}$, $r_{21} = r_{23} = \frac{1}{3}$, $r_{31} = r_{32} = r_{30} = \frac{1}{3}$, where again the subscript 0 represents the outside world.

### Flow balance equations - example (cont)

- **Assuming** that the queues are all stable, the flow balance equations are
  
  \[
  \begin{align*}
  \lambda_1 &= \Lambda_1 + \frac{1}{2} \lambda_2 + \frac{1}{3} \lambda_3, \\
  \lambda_2 &= \Lambda_2 + \frac{1}{2} \lambda_1 + \frac{1}{3} \lambda_3, \\
  \lambda_3 &= \frac{1}{2} \lambda_1 + \frac{1}{3} \lambda_2.
  \end{align*}
  \]

- Thus, in matrix form,
  
  \[
  \begin{bmatrix}
  1 & -\frac{1}{2} & -\frac{1}{3} \\
  -\frac{1}{2} & 1 & -\frac{1}{3} \\
  -\frac{1}{3} & -\frac{1}{3} & 1
  \end{bmatrix}
  \begin{bmatrix}
  \lambda_1 \\
  \lambda_2 \\
  \lambda_3
  \end{bmatrix}
  \begin{bmatrix}
  \Lambda_1 \\
  \Lambda_2 \\
  0
  \end{bmatrix},
  \]
  
  which implies
  
  \[
  \begin{bmatrix}
  \lambda_1 \\
  \lambda_2 \\
  \lambda_3
  \end{bmatrix}
  = \begin{bmatrix}
  1 & -\frac{1}{2} & -\frac{1}{3} \\
  -\frac{1}{2} & 1 & -\frac{1}{3} \\
  -\frac{1}{3} & -\frac{1}{3} & 1
  \end{bmatrix}^{-1}
  \begin{bmatrix}
  \Lambda_1 \\
  \Lambda_2 \\
  0
  \end{bmatrix},
  \]
  
  i.e., $\Delta = (I - R)^{-1} \Delta$.
Flow balance equations - example (cont)

• Given the total flow rates $\lambda$, the service rates $\mu_k$ need to be chosen so that $\mu_k > \lambda_k$ for all queues $k$ to achieve stability and stationarity (the flow balance equations hold).

• Note that the mean departure rate to the “outside world,” $\lambda_3$, will work out from the flow balance equations to be

$$\lambda_3 r_{30} = \Lambda_1 + \Lambda_2.$$

• Finally, the stability assumption requires that the service rates

$$\mu^T > \lambda^T = \Lambda^T (I - R)^{-1}.$$

Exercise - maximum throughput of a network processor

• Consider a NP with multiple internal engines/stations, e.g., for: 1. header checksum processing, 2. TTL decrement, 3. forwarding look-up, and 4. flow-based processing (e.g., policing, shaping, prioritizing - a flow engine).

• A NP needs to be able to operate at a “worst-case” prescribed packet (job) arrivals at rate;

• e.g., for an OC-48 line, 2.5 Gbps $= 7.8$ Mpps $= \lambda_0$, assuming the worst-case that all IP packets are 40 bytes long and all packets pass through the first four engines.

• Suppose all packets arriving to the 4th (flow) engine cause a flow lookup operation, and thereafter a number $N$ of different flow sub-engines, indexed 5 to $N + 5 - 1$, may be visited.

• Define the probabilities $r_0; r_{k,k+1} = 1 \forall k < 4; r_{k,0} = r_0 \forall k \geq 4; r_{4,j} = (1 - r_0)/N \forall j > 4; r_{k,j} = (1 - r_0)/(N - 1) \forall k \neq j \geq 5$.

• Exercise: In terms of $r_0, N$:

  - Find the average number of flow sub-engine visits by a packet.

  - Find the minimum service capacity of each engine and sub-engine so that $\lambda_0$ is the throughput of the NP.
Markovian queuing systems in continuous time

- Introduction
- Memoryless property of exponential distribution
- Finite-dimensional distributions and stationarity
- The Poisson counting process
- Poisson Arrivals See Time Averages (PASTA)
- Time-homogeneous Markov processes on countable state space (Markov chains)
- Fitting a Markov model to data
- Birth-death Markov chains
- Markovian queuing models: single queues and queuing networks

Markov modeling - state variables

- More complex performance metrics, such as the distribution of delays experienced by jobs, requires more detailed modeling of the (stationary) queueing system.
- Application of Markovian models begins with identifying state variables in the data (or the system that generated the data).
- The current state summarizes the past evolution of the data so that one need not remember the past in order to determine/predict the future evolution of the data/system.
- This is consistent with the notion of a finite-state machine in computer science.
- In deterministic linear circuits, the state variables are “outputs of integrators,” i.e., voltage across capacitors \( C \),

\[
\forall t \geq s, \quad v_C(t) = v_C(s) + \frac{1}{C} \int_s^t i_C(\tau) d\tau
\]

and currents through inductors.
- In a stochastic setting, continuous-time Markov processes have a special structure involving the (memoryless) exponential distribution.
Memoryless property of the exponential distribution

- If \( X \) is exponentially distributed, then
  \[
P(X > x + y \mid X > y) = P(X > x).
  \]
- The proof is an immediate consequence of the distribution of an exponential,
  \[P(X > x) = e^{-\lambda x},\] where \( E(X) = \frac{1}{\lambda} \) and \( u \) is the unit step, \( u(x) = 1\{x \geq 0\} \).
- This is the memoryless property and its simple proof is left as an exercise.
- For example, if \( X \) represents the duration of the lifetime of a light bulb, the memoryless property implies that, given that \( X > y \), the probability that the residual lifetime \( (X - y) \) is greater than \( x \) is equal to the probability that the unconditioned lifetime is greater than \( x \).
- So, in this sense, given \( X > y \), the lifetime has “forgotten” that \( X > y \).
- Only exponentially distributed random variables have this property among all continuously distributed random variables and only geometrically distributed random variables have this property among all discretely distributed random variables.

Minimum of independent exponentially distributed random variables

- If \( X_1 \sim \exp(\lambda_1) \) and \( X_2 \sim \exp(\lambda_2) \) are independent, then
  \[
  \min\{X_1, X_2\} \sim \exp(\lambda_1 + \lambda_2).
  \]
- Proof: Define \( Z = \min\{X_1, X_2\} \) and let \( F_Z(z) = P(Z \leq z) \), \( F_1 \), and \( F_2 \) be the CDF of \( Z \), \( X_1 \), and \( X_2 \), respectively.
- Clearly, \( F_Z(z) = 0 \) for \( z < 0 \) and for \( z \geq 0 \),
  \[
  1 - F_Z(z) = P(\min\{X_1, X_2\} > z)
  = P(X_1 > z, X_2 > z)
  = P(X_1 > z)P(X_2 > z) \text{ by independence}
  = \exp(- (\lambda_1 + \lambda_2) z)
  \]
  as desired. \(\square\)
Minimum of independent exponentially distr’d random variables (cont)

• Again, if \( X_1 \sim \exp(\lambda_1) \) and \( X_2 \sim \exp(\lambda_2) \) are independent, then

\[
P(\min\{X_1, X_2\} = X_1) = \frac{\lambda_1}{\lambda_1 + \lambda_2}.
\]

• Proof:

\[
P(\min\{X_1, X_2\} = X_1) = P(X_1 \leq X_2) = \int_0^\infty \int_0^{x_2} \lambda_1 e^{-\lambda_1 x_1} dx_1 \lambda_2 e^{-\lambda_2 x_2} dx_2
\]
\[
= \int_0^\infty \lambda_2 e^{-(\lambda_1 + \lambda_2) x_2} dx_2
\]
\[
= \frac{\lambda_1}{\lambda_1 + \lambda_2}
\]
as desired. □

• Two independent geometrically distributed random variables also have these properties.

A counting process on \( \mathbb{R}^+ \)

• A counting process \( X \) on \( \mathbb{R}^+ \) is characterized by the following properties:

(a) \( X \) has state space \( \mathbb{Z}^+ \),

(b) \( X \) has nondecreasing (in time) sample paths that are continuous from the right, i.e.,

\[
\lim_{t \downarrow s} X(t) = X(s), \quad and
\]

\[
\lim_{t \downarrow t-} X(t-) = \lim_{s \uparrow t} X(s).
\]

(c) \( X(t) \leq X(t-) + 1 \) so that \( X \) does make a single transition of size 2 or more, where \( t- \) is a time immediately prior to \( t \), i.e.,

\[
\lim_{t \downarrow t-} X(t-)
\]

• For example, consider a post office where the \( i^{th} \) customer arrives at time \( T_i \in \mathbb{R}^+ \). We take the origin of time to be zero and, clearly, \( T_i \leq T_{i+1} \) for all \( i \).
A counting process on $\mathbb{R}^+$ (cont)

- The total number of customers that arrived over the interval of time $[0, t]$ is defined to be $X(t)$.
- Note that $X(T_i) = i$, $X(t) < i$ if $t < T_i$, and $X(t) - X(s)$ is the number of customers that have arrived over the interval $(s, t]$,

$$X(t) = \sum_{i=1}^{\infty} 1\{T_i \leq t\} = \max\{i \mid T_i \leq t\}.$$
- Of course, $X$ is an example of a continuous-time counting process whose sample paths are continuous from the right.

The Poisson counting process - definition by interarrival times

- Now let the sequence of job interarrival times be $S_i = T_i - T_{i-1}$ for job indexes $i \in \{1, 2, 3, \ldots\}$, where

$$T_0 \equiv 0.$$
- A Poisson process is a continuous-time counting process whose interarrival times $\{S_i\}_{i=1}^{\infty}$ are mutually IID exponential random variables.
- Let the parameter of the exponential distribution of the $S_i$’s be $\lambda$, i.e., $\mathbb{E}S_i = \lambda^{-1}$ for all $i$.
- Since

$$T_n = \sum_{i=1}^{n} S_i,$$

$T_n$ is Erlang (gamma) distributed with parameters $\lambda$ and $n$. 
• $X(t)$ is Poisson distributed with parameter $\lambda t$.

• For this reason, $\lambda$ is sometimes called the intensity (or "mean intensity", "mean rate", or just "rate") of the Poisson process $X$.

• **Proof:** First note that, for $t \geq 0$,
  \[ P(X(t) = 0) = P(T_1 > t) = P(S_1 > t) = e^{-\lambda t}. \]

• Now, for an integer $i > 0$ and a real $t \geq 0$,
  \[ P(X(t) \leq i) = P(T_{i+1} > t) = \int_t^\infty \frac{\lambda^i e^{-\lambda z}}{i!} dz, \]
  where we have used the gamma PDF.

• By integrating by parts, we get
  \[ P(X(t) \leq i) = \frac{\lambda^i}{i!} (-e^{-\lambda t})|_t^\infty + \int_t^\infty \frac{\lambda^{i-1} e^{-\lambda z}}{(i-1)!} dz, \]
  \[ = \frac{(\lambda t)^i e^{-\lambda t}}{i!} + \int_t^\infty \frac{\lambda^{i-1} e^{-\lambda z}}{(i-1)!} dz. \]

• After successively integrating by parts in this manner, we get
  \[ P(X(t) \leq i) = \frac{(\lambda t)^i e^{-\lambda t}}{i!} + \cdots + \frac{(\lambda t)^0 e^{-\lambda t}}{1!} + \int_t^\infty \lambda e^{-\lambda z} dz, \]
  \[ = \sum_{j=0}^i \frac{(\lambda t)^j e^{-\lambda t}}{j!}. \]

• Now note that \{ $X(t) = i$ \} and \{ $X(t) \leq i - 1$ \} are disjoint events and
  \[ \{ X(t) = i \} \cup \{ X(t) \leq i - 1 \} = \{ X(t) \leq i \}. \]

• Thus,
  \[ P(X(t) = i) = P(X(t) \leq i) - P(X(t) \leq i - 1) \]
  \[ = \sum_{j=0}^i \frac{(\lambda t)^j e^{-\lambda t}}{j!} - \sum_{j=0}^{i-1} \frac{(\lambda t)^j e^{-\lambda t}}{j!} \]
  \[ = \frac{(\lambda t)^i e^{-\lambda t}}{i!}. \]
Increments of a Poisson Process

- $X$ is a Poisson process if and only if, for all $k$, all disjoint intervals $(s_1, t_1], (s_2, t_2], ...,$ $(s_k, t_k] \subset \mathbb{R}^+$, and all $n_1, n_2, ..., n_k \in \mathbb{Z}^+$,
  \[ P(X(t_1) - X(s_1) = n_1, X(t_2) - X(s_2) = n_2, ..., X(t_k) - X(s_k) = n_k) = \prod_{i=1}^{k} \frac{[\lambda(t_i - s_i)]^{n_i}}{n_i!} e^{-\lambda(t_i - s_i)}. \]

- $X(t_i) - X(s_i)$, called an increment of $X$, is the number of transitions of the Poisson process in the interval of time $(s_i, t_i]$.

- Thus, the Poisson process has independent (nonoverlapping) increments.

- Also, the increment over a time interval of length $\tau$ is Poisson distributed with parameter $\lambda \tau$.

Equivalent definitions of a Poisson process

- The Poisson process is the only counting process that
  - has Poisson distributed independent increments, or
  - has IID exponentially distributed interarrival times, or
  - possesses the conditional uniformity property.

- That is, all of these properties are equivalent.

- The memoryless property of the exponential distribution is principally responsible for the independent increments property of a Poisson process.
Poisson process - finite-dimensional distributions

• We will now drive the \( k(\geq 1) \)-dimensional distribution of a Poisson process by using the independent increments property.

• Consider times \( 0 \leq t_1 < t_2 < \cdots < t_k \) and

\[
P(X(t_1) = m_1, X(t_2) = m_2, \ldots, X(t_k) = m_k),
\]

where \( m_1, m_2, \ldots, m_k \in \mathbb{Z} \) and \( m_i \leq m_{i+1} \) for all \( i \) (otherwise the probability above would be zero).

• Define \( \Delta m_i := m_i - m_{i-1} \) and \( \Delta X_i := X(t_i) - X(t_{i-1}) \) to get

\[
P(X(t_1) = m_1, \Delta X_2 = \Delta m_2, \ldots, \Delta X_k = \Delta m_k)
= P(\Delta X_2 = \Delta m_2, \ldots, \Delta X_k = \Delta m_k | X(t_1) = m_1) P(X(t_1) = m_1)
= P(\Delta X_2 = \Delta m_2, \ldots, \Delta X_k = \Delta m_k) P(X(t_1) = m_1),
\]

where the last equality is by the independent increments property.

• By repeating this argument, we get that the above \( k \)-dimensional distribution is

\[
P(X(t_1) = m_1) \prod_{i=2}^{k} P(X(t_i) - X(t_{i-1}) = m_i - m_{i-1})
= \frac{(\lambda t_1)^{m_1}}{m_1!} e^{-\lambda t_1} \prod_{i=2}^{k} \frac{(\lambda (t_i - t_{i-1}))^{m_i - m_{i-1}}}{(m_i - m_{i-1})!} e^{-\lambda (t_i - t_{i-1})}.
\]

Poisson processes on \( \mathbb{R}^n \) for \( n \geq 1 \)

• A stationary Poisson process on the whole real line \( \mathbb{R} \) is defined by
  – a countable collection of points \( \{\tau_i\}_{i=-\infty}^{\infty} \),
  – where the interarrival times \( \tau_i - \tau_{i-1} \) are IID exponential random variables.

• Alternatively, we can characterize a Poisson process on \( \mathbb{R} \) by stipulating that
  – the number of points in any interval of length \( t \) is Poisson distributed with mean \( \lambda t \), and
  – that the number of points in nonoverlapping intervals is independent.

• This last characterization naturally extends to that of a Poisson point process on \( \mathbb{R}^n \) for all dimensions \( n \geq 1 \), i.e., a spatial Poisson process:

• If \( v(A) \) is the volume of \( A \subset \mathbb{R}^n \),
  – then the number of points in \( A \) is Poisson distributed with mean \( \delta v(A) \),
  – where \( \delta \) is the intensity of the Poisson process with \( [\delta] = \text{points/metre}^n \).
Example: Hand-off rates among wireless cells

- For this example, we need following result that the Poisson property is preserved by IID random shifts of the points.

- **Theorem:** If \( \{ \tau_i \} \) is a Poisson process in \( \mathbb{R}^n \) with intensity \( \delta \) and the random vectors \( \{ Y_i \} \) in \( \mathbb{R}^n \) are IID and a.s. bounded, then \( \{ \tau_i + Y_i \} \) is a Poisson process intensity \( \delta \) as well.

- In the two-dimensional plane \( \mathbb{R}^2 \) covered by roughly circular cells, assume each mobile takes a direct path through each cell.

- At a cell boundary, an independent and uniformly distributed random change of direction occurs for each mobile.

- A sample path of a single mobile is depicted in the following figure, where the dot at the center of a cell is its base station.
Example: Hand-off rates among wireless cells (cont)

- Further assume that the average velocities of a mobile through the cells are IID with density \( f(v) \) over \([v_{\text{min}}, v_{\text{max}}]\).

- The mobiles are initially distributed in the plane according to a spatial Poisson process with density \( \delta \) mobile nodes per unit area.

- Finally, assume that the cells themselves are also distributed in the plane so that, at any given time, the total displacements of the mobiles are IID.

- Note: The base stations could also be randomly placed according to a spatial Poisson process with density \( \delta' \ll \delta \) and the resulting circular cells approximate Voronoi sets about each of them.

- Exercise:
  (a) Find the mean rate \( \lambda_m \) of mobiles crossing into a cell of diameter \( \Delta \). Hint: consider the length of a chord and use Little’s result.
  (b) How would the expression differ in (a) if velocity and direction through a cell were dependent?

Cts-time, time-homog. Markov processes with countable state-space

- We will now define a kind of stochastic process called a Markov process.

- The Poisson process is a (transient) pure birth Markov process.

- A Markov process on a countable state space \( \Sigma \) is called a Markov chain.

- A Markov chain is a kind of random walk on \( \Sigma \). \((= Z^+ \text{ w.l.o.g.})\).

- It visits a state, stays there for an exponentially distributed amount of time, then makes a transition at random to another state, stays at this new state for an exponentially distributed amount of time, then makes a transition at random to another state, etc.

- All of these visit times and transitions are independent in a way that will be more precisely explained in the following.
**The Markov property**

- If, for all integers \( k \geq 1 \), all subsets \( A, B, B_1, \ldots, B_k \subset \Sigma \), and all times \( t, s, s_1, \ldots, s_k \in \mathbb{R}^+ \) such that \( t > s > s_1 > \cdots > s_k \),

\[
P(X(t) \in A \mid X(s) \in B, X(s_1) \in B_1, \ldots, X(s_k) \in B_k) = P(X(t) \in A \mid X(s) \in B),
\]

then the stochastic process \( X \) is said to possess the Markov property.

- If we identify
  - \( X(t) \) as a future value of the process,
  - \( X(s) \) as the present value,
  - and past values as \( X(s_1), \ldots, X(s_k) \),

then the Markov property asserts that the future and the past are conditionally independent given the present.

- In other words, given the present state \( X(s) \) of a Markov process, one does not require knowledge of the past to determine its future evolution.

---

**The Markov property (cont)**

- Any stochastic process (on any state space with any time domain) that has the Markov property is called a Markov process.

- As such, the Markov property is a "stochastic extension" of notions of state associated with finite-state machines and linear time-invariant systems.

- The Markov property as stated above is an immediate consequence of a slightly stronger and more succinctly stated Markov property: for all times \( s < t \) and any (measurable) function \( f \),

\[
E(f(X_t) \mid X_r, 0 \leq r \leq s) = E(f(X_t) \mid X_s).
\]
Sample path construction of a continuous-time Markov chain

- For a time-homogeneous Markov chain, consider each state \( n \in \mathbb{Z}^+ \) and let

\[
ES := \frac{1}{-q_{n,n}} > 0
\]

be the mean visiting time of the Markov process, i.e., \( q_{n,n} < 0 \).

- That is, a Markov chain is said to enter state \( n \) at time \( T \) and subsequently visit state \( n \) for \( S \) seconds if \( X(T^-) \neq n, X(t) = n \) for all \( T \leq t < S + T \), and \( X(S + T) \neq n \).

- Also, define the assumed finite set of states

\[
\mathcal{T}_n \subset \mathbb{Z}^+ \setminus \{n\}
\]

to which a transition is possible directly from \( n \).

Sample path construction of a Markov chain - transition rates

- For all \( m \in \mathcal{T}_n \), define \( q_{n,m} > 0 \) such that the probability of a transition from \( n \) to \( m \) is

\[
\frac{q_{n,m}}{q_{n,n}} > 0.
\]

- Thus, we clearly need to require that

\[
\sum_{m \in \mathcal{T}_n} \frac{q_{n,m}}{q_{n,n}} = 1,
\]

i.e., for all \( n \in \mathbb{Z}^+ \),

\[
\sum_{m \in \mathbb{Z}^+} q_{n,m} = 0,
\]

where \( q_{n,m} := 0 \) for all \( m \not\in \mathcal{T}_n \cup \{n\} \).
Sample path construction of a Markov chain - initial distribution

- Now let $T_i$ be the time of the $i^{th}$ state transition with $T_0 \equiv 0$, i.e., the process $X$ is constant on intervals $[T_{i-1}, T_i)$ and
  \[ X(T_{i-1}) = X(T_i) \neq X(T_i) \]
  for all $i \in \mathbb{Z}^+$.

- Let the column vector $\pi(0)$ represent the distribution of $X(0)$ on $\mathbb{Z}^+$, so that entry in the $n^{th}$ row is
  \[ \pi_n(0) = P(X(0) = n), \]
  i.e., $\pi(0)$ is the initial distribution of the stochastic process $X$.

Sample path construction of a Markov chain - alternative construction

- Suppose that $X(T_i) = n \in \mathbb{Z}^+$.

- To the states $m \in T_n$, associate an exponentially distributed random variable $S_i(n, m)$ with parameter $q_{n,m} > 0$ (recall this means $E S_i(n, m) = 1/q_{n,m}$).

- Given $X(T_i) = n$, the smallest of the random variables
  \[ \{S_i(n, m) \mid m \in T_n\} \]
  determines $X(T_{i+1})$ and the intertransition time $T_{i+1} - T_i$.

- That is, $X(T_{i+1}) = j$ if and only if
  \[ T_{i+1} - T_i = S_i(n, j) = \min_{m \in T_n} S_i(n, m). \]

- The entire collection of exponential random variables
  \[ \{S_i(n, m) \mid i \in \mathbb{Z}^+, n \in \mathbb{Z}^+, m \in T_n\} \]
  are assumed mutually independent.
Therefore, the inter-transition time $T_{i+1} - T_i$ is exponentially distributed with parameter

$$-q_{n,n} := \sum_{m \in \mathcal{T}_n} q_{n,m},$$

$$\Rightarrow \ E(T_{i+1} - T_i) = \frac{1}{-q_{n,n}} > 0 \text{ in particular.}$$

Also, the state transition probabilities

$$P(X(T_{i+1}) = j \mid X(T_i) = n) = P(S_i(n,j) = \min_{m \in \mathcal{T}_n} S_i(n,m)) = -\frac{q_{n,j}}{q_{n,n}}.$$

Note again that if a transition from state $n$ to state $j$ is impossible (has probability zero), $q_{n,j} = 0$.

**Note:** Parameters (rates) $q$ are not probabilities.

---

Conservativeness and time-homogeneity assumptions

- In the following, we assume that $-q_{n,n} < \infty$ for all states $n$, i.e., the Markov chain is conservative.

- Also, we have assumed that the Markov chain is temporally (time) homogeneous, i.e., for all times $s, t \geq 0$ and all states $n, m$:

$$P(X(s + t) = n \mid X(s) = m) = P(X(t) = n \mid X(0) = m).$$

- In summary, assuming the initial distribution $\pi(0)$ and the parameters

$$\{q_{n,m} \mid n, m \in \mathbb{Z}^+\}$$

are known, we have described how to construct a sample path of the Markov chain $X$ from a collection of independent random variables

$$\{S_i(n,m) \mid i \in \mathbb{Z}^+, \ n \in \Sigma = \mathbb{Z}^+, \ m \in \mathcal{T}_n\},$$

where $S_i(n,m)$ is exponentially distributed with parameter $q_{n,m}$.

- When a Markov chain visits state $n$, it stays an exponentially distributed amount of time with mean $-1/q_{n,n}$ and then makes a transition to another state $m \in \mathcal{T}_n$ with probability $-q_{n,m}/q_{n,n}$. 

---
Proof that thus constructed process is Markovian

- To prove that the processes thus constructed are Markovian, let
  - \( n := X(s) \) and
  - \( i \) be the number of transitions of \( X \) prior to the present time \( s \).

- Clearly, the random variables \( i, n \), and \( T_i \) (the last transition time prior to \( s \)) can be discerned from \( \{X_r, 0 \leq r \leq s\} \) and can therefore be considered "given" as well.

- The memoryless property of the random variable \( T_{i+1} - T_i \), distributed exponentially with parameter \( -q_{n,n} \), implies that
  
  \[
  P(T_{i+1} - s > x \mid T_{i+1} - T_i > s - T_i) = P(T_{i+1} - T_i > x + (s - T_i) \mid T_{i+1} - T_i > s - T_i) = P(T_{i+1} - T_i > x) = \exp(q_{n,n}x)
  \]
  
  for all \( x > 0 \).

- Note that \( \exp(q_{n,n}x) \) depends on \( \{X_r, 0 \leq r \leq s\} \) only through \( n = X(s) \).

Proof that thus constructed process is Markovian (cont)

- So, \( T_{i+1} - s \) is exponentially distributed with parameter \( -q_{n,n} \) and conditionally independent of \( s - T_i \) given \( \{X_r, 0 \leq r \leq s\} \).

- Furthermore, \( \{X_r, 0 \leq r < T_i\} \) is similarly conditionally independent of \( \{X_r, r \geq T_{i+1}\} \) given \( X(s) = n \) (by the assumed mutual independence of the \( \{S_i(n,m)\} \) random variables).

- Since the exponential distribution is the only continuous one that is memoryless, one can conversely show that the Markov property implies the qualities of the previous constructions.
The Poisson process is Markovian

- Clearly, a Poisson process with intensity \( \lambda \) is an example of a Markov chain.

- The transition rates of a Poisson process are

\[
\forall n \in \mathbb{Z}^+, \quad q_{n,m} = \begin{cases} 
\lambda & \text{if } m = n + 1, \\
-\lambda & \text{if } m = n, \\
0 & \text{else.}
\end{cases}
\]

Transition-rate matrix (generator) of a cts-time Markov chain

- The matrix \( Q \) having \( q_{n,m} \) as its entry in the \( n \text{th} \) row and \( m \text{th} \) column is called the transition rate matrix (or just “rate matrix” or “generator”) of the Markov chain \( X \).

- Note by definition of \( q_{i,i} < 0 \), the sum of the entries in any row of the matrix \( Q \) equals zero.

- The \( n \text{th} \) row of \( Q \) corresponds to state \( n \) from which transitions occur, and

= the \( m \text{th} \) column of \( Q \) corresponds to states \( m \) to which transitions occur.

- For \( n \neq m \), the parameter \( q_{n,m} \) is called a transition rate (or probability flux) because, for any \( i \in \mathbb{Z}^+ \), \( ES_i(n, m) = 1/q_{n,m} \).

- Thus, we expect that, if \( q_{n,m} > q_{n,j} \), then transitions from state \( n \) to \( m \) will tend to be made more frequently (at a higher rate) by the Markov chain than transitions from state \( n \) to \( j \).
• The transition matrix of a Poisson process with intensity $\lambda > 0$ is

$$Q = \begin{bmatrix}
-\lambda & \lambda & 0 & 0 & 0 & \cdots \\
0 & -\lambda & \lambda & 0 & 0 & \cdots \\
0 & 0 & -\lambda & \lambda & 0 & \cdots \\
\vdots & \vdots & \vdots & \vdots & \ddots & \ddots
\end{bmatrix}$$

Rate matrix - example 2

• Suppose the strict state space of $X$ is $\{0, 1, 2\}$ and the rate matrix is

$$Q = \begin{bmatrix}
-5 & 2 & 3 \\
0 & -4 & 4 \\
1 & 0 & -1
\end{bmatrix}$$

• $Q$ is just a $3 \times 3$ since the strict state space is just a (finite-sized) 3-tuple $\{0, 1, 2\}$, rather than all of the nonnegative integers, $\mathbb{Z}^+$.

• A direct transition from state 2 to state 1 is impossible (as is a direct transition from state 1 to state 0).

• Also, each visit to state 0 lasts an exponentially distributed amount of time with parameter 5 (i.e., with mean 0.2), a transition to state 1 then occurs with probability $\frac{2}{5}$ or a transition to state 2 occurs with probability $\frac{3}{5}$. 
Graphical depiction of a Markov chain's transition rates

- We can also represent the transition rates (and states) graphically by what is called a transition rate diagram.

- The states of the Markov chain are circled and arrows are used to indicate the possible transitions (labeled with assumed positive transition rates) between states.

- The transition rate itself labels the corresponding arrow (transition).

- For the two previous examples:

\[
\begin{align*}
0 & \quad 1 & \quad 2 & \quad 3 & \quad 4 & \quad \ldots \\
\lambda & & \lambda & \lambda & \lambda & \\
\end{align*}
\]

\[
\begin{align*}
0 & \quad 1 & \quad 2 \\
q_{0,0} = 1 & & q_{0,1} = 2 \\
& & q_{1,2} = 4 \\
& & q_{2,0} = 3 \\
\end{align*}
\]

The Kolmogorov equations

- Consider the Markov chain \( X \) on \( \mathbb{Z}^+ \) with rate matrix \( Q \) and initial distribution \( \pi(0) \).

- For \( \tau \in \mathbb{R}^+ \) and \( n, m \in \mathbb{Z}^+ \), define

\[
p_{n,m}(\tau) = P(X(s+\tau) = m \mid X(s) = n).
\]

- Again, we are assuming that the chain is temporally homogeneous so that the right-hand side of the above equation does not depend on time \( s \).

- The matrix \( P(\tau) \) whose entry in the \( n \)th row and \( m \)th column is \( p_{n,m}(\tau) \) is called the transition probability matrix.

- Finally, for all times \( s \in \mathbb{R}^+ \) and all states \( n \in \mathbb{Z}^+ \), define

\[
\pi_n(s) := P(X(s) = n).
\]

- So, the column vector \( \pi(s) \), whose \( i \)th entry is \( \pi_i(s) \), is the marginal distribution of \( X \) at time \( s \), i.e., the distribution (PMF) of \( X(s) \).
The Kolmogorov equations (cont)

- Conditioning on \( X(s) \) and using the law of total probability,
  \[
P(X(s + \tau) = m) = \sum_{n=0}^{\infty} P(X(s + \tau) = m \mid X(s) = n)P(X(s) = n)
  \]
  for all \( m \in \mathbb{Z}^+ \), i.e.,
  \[
  \pi_m(s + \tau) = \sum_{i=0}^{\infty} p_{n,m}(\tau)\pi_n(s) \quad \text{for all} \quad m \in \mathbb{Z}^+ .
  \]
- We can write these equations compactly in matrix form:
  \[
  \pi^T(s + \tau) = \pi^T(s)P(\tau),
  \]
  where \( \pi^T(s) \) is the transpose of the column vector \( \pi(s) \), i.e., \( \pi^T(s) \) is a row vector.

Moreover, any finite-dimensional distribution (FDD) of the Markov chain can be computed from the transition probability functions and the initial distribution.

- For example, for times \( 0 < r < s < t \),
  \[
P(X(t) = n, X(s) = m, X(r) = k)
  = P(X(t) = n \mid X(s) = m, X(r) = k)P(X(s) = m, X(r) = k)
  = P(X(t) = n \mid X(s) = m)P(X(s) = m \mid X(r) = k)P(X(r) = k)
  = p_{m,n}(t-s)p_{k,m}(s-r)\sum_{i} P(X(r) = k \mid X(0) = i)P(X(0) = i)
  = p_{m,n}(t-s)p_{k,m}(s-r)\sum_{i} p_{i,k}(r)\pi_i(0)
  \]
  where the second equality is the Markov property.

- In the second-to-last expression, we clearly see the transition from some initial state to \( k \) at time \( r \), then to state \( m \) at time \( s \) \( (s-r \text{ seconds later}) \), and finally to state \( n \) at time \( t \) \( (t-s \text{ seconds later}) \).
Computing the transition probability matrix with the rate matrix

• First note that a transition in an interval of time of length zero occurs with probability zero,
  \[ p_{n,m}(0) = 1 \{n = m\} \quad \forall n, m, \quad \text{i.e.,} \quad P(0) = I, \]
  where \( I \) is the (multiplicative) identity matrix, i.e., i.e., the square matrix with 1’s in every diagonal entry and 0’s in every off-diagonal entry.

• For states \( n \neq m \), a small amount of time \( 0 < \varepsilon \ll 1 \), and an arbitrarily chosen time \( s \in \mathbb{R}^+ \), consider
  \[ p_{n,m}(\varepsilon) = P(X(s + \varepsilon) = m \mid X(s) = n). \]

• Let \( V_n \) be the residual holding time in state \( n \) after time \( s \), i.e., \( X(t) = n \) for all \( t \in [s, s + V_n) \) and \( X(s + V_n) \neq n \).

Computing the TPM with the rate matrix (cont)

• The total holding time in state \( n \) is \( \sim \text{exp}(-q_{n,n}) \).

• So by the memoryless property, \( V_n \sim \text{exp}(-q_{n,n}) \) and for all \( m \neq n \),
  \[ p_{n,m}(\varepsilon) = P(V_n \leq \varepsilon) \times \frac{q_{n,m}}{-q_{n,n}} + o(\varepsilon). \]

• The first term on the RHS represents the probability that the Markov chain \( X \) makes only a single transition (from \( n \) to \( m \)) in interval of time \( (s, s + \varepsilon] \).

• Recall that the probability that \( X \) makes a transition to state \( m \) from state \( n \) is \( -q_{n,m}/q_{n,n} \).

• The symbol \( o(\varepsilon) \) ("little oh of \( \varepsilon \)"") represents a function satisfying
  \[ \lim_{\varepsilon \to 0} \frac{o(\varepsilon)}{\varepsilon} = 0, \]
  specifically here the probability that the Markov chain has two or more transitions in the interval of time \( (s, s + \varepsilon] \).
Computing the TPM with the rate matrix (cont)

- Substituting
  \[ P(V_n \leq \varepsilon) = 1 - \exp(\varepsilon q_{n,n}) = -\varepsilon q_{n,n} + o(\varepsilon), \]
gives for all \( m \neq n \),
  \[ p_{n,m}(\varepsilon) = q_{n,m} \varepsilon + o(\varepsilon) \]

\[ \Rightarrow \frac{p_{n,m}(\varepsilon) - p_{n,m}(0)}{\varepsilon} = q_{n,m} + \frac{o(\varepsilon)}{\varepsilon}, \]
where we recall that \( p_{n,m}(0) = 0 \) for all \( m \neq n \).

- Letting \( \varepsilon \to 0 \), we get
  \[ \forall m \neq n, \quad \dot{p}_{n,m}(0) = q_{n,m}, \]
where the left-hand side is the \textit{time derivative of} \( p_{n,m} \) at time 0.

The Kolmogorov backward equations

- Finally, since
  \[ p_{n,n}(\varepsilon) = 1 - \sum_{m \in \mathbb{Z}^+, m \neq n} p_{n,m}(\varepsilon), \]
we get, after differentiating with respect to time,
  \[ \dot{p}_{n,n}(0) = -\sum_{m \neq n} q_{n,m} = q_{n,n} < 0, \]
where we have used the \textit{definition} of \( q_{n,n} \).

- In matrix form,
  \[ \dot{P}(0) = Q. \]

- This statement can be generalized to obtain the \textit{Kolmogorov backward equations}:
  \[ \forall \tau \geq 0, \quad \dot{P}(\tau) = P(\tau)Q \quad \text{with} \quad P(0) = I. \]
The Kolmogorov backward equations - proof

- First, we have already established (for \( s = 0 \)) that
  \[ P(0) = IQ = P(0)Q, \]

- For \( s > 0 \), take a real \( \varepsilon \) such that \( 0 < \varepsilon \ll \min\{s, 1\} \). So,
  \[
  p_{n,m}(s) = P(X(s) = m \mid X(0) = n) = \frac{P(X(s) = m, X(0) = n)}{P(X(0) = n)} \\
  = \sum_{k=0}^{\infty} \frac{P(X(s) = m, X(s - \varepsilon) = k, X(0) = n)}{P(X(0) = n)} \times \frac{P(X(s - \varepsilon) = k, X(0) = n)}{P(X(s - \varepsilon) = k, X(0) = n)} \\
  = \sum_{k=0}^{\infty} \frac{P(X(s - \varepsilon) = k \mid X(0) = n)}{P(X(s) = m \mid X(s - \varepsilon) = k, X(0) = n)} \times P(X(s) = m \mid X(s - \varepsilon) = k) \\
  = \sum_{k=0}^{\infty} p_{n,k}(s - \varepsilon)p_{k,m}(\varepsilon)
  \]

The Kolmogorov backward equations - proof (cont)

- Therefore,
  \[
  p_{n,m}(s) = p_{n,m}(s - \varepsilon)p_{m,m}(\varepsilon) + \varepsilon \sum_{k \neq m} p_{n,k}(s - \varepsilon)q_{k,m} + o(\varepsilon) \\
  = p_{n,m}(s - \varepsilon)\left(1 - \varepsilon \sum_{i \neq m} q_{m,i}\right) + \varepsilon \sum_{k \neq m} p_{n,k}(s - \varepsilon)q_{k,m} + o(\varepsilon) \\
  = p_{n,m}(s - \varepsilon)\left(1 - \varepsilon \sum_{i \neq m} q_{m,i}\right) + \varepsilon \sum_{k \neq m} p_{n,k}(s - \varepsilon)q_{k,m} + o(\varepsilon) \\
  = p_{n,m}(s - \varepsilon)(1 + \varepsilon q_{m,m}) + \varepsilon \sum_{k \neq m} p_{n,k}(s - \varepsilon)q_{k,m} + o(\varepsilon).
  \]
The Kolmogorov backward equations - proof (cont)

- After a simple rearrangement we get
  \[
  \frac{p_{n,m}(s) - p_{n,m}(s - \varepsilon)}{\varepsilon} = p_{n,m}(s - \varepsilon)q_{m,m} + \sum_{k \neq m} p_{n,k}(s - \varepsilon)q_{k,m} + \frac{o(\varepsilon)}{\varepsilon}
  \]
  \[
  = \sum_{k=0}^{\infty} p_{n,k}(s - \varepsilon)q_{k,m} + \frac{o(\varepsilon)}{\varepsilon}.
  \]

- So, letting \( \varepsilon \to 0 \) in the previous equation, we get, for all \( n, m \in \mathbb{Z}^+ \) and all real \( s > 0 \),
  \[
  \dot{p}_{n,m}(s) = \sum_{k=0}^{\infty} p_{n,k}(s)q_{k,m}
  \]
  as desired. \( \Box \)

Kolmogorov forward equations

- Using a similar argument, one can condition on the distribution of \( X(\varepsilon) \), i.e., move forward in time from the origin.

- We will then arrive at the Kolmogorov forward equations:
  \[
  \dot{P}(s) = QP(s).
  \]
Transition probability matrix by matrix exponential

- Recall that
  \[ P(0) = I. \]

- Equipped with this initial condition, we can solve the Kolmogorov equations for the case of a finite state space to get, for all \( t \geq 0 \),
  \[ P(t) = e^{Qt}, \]
  where the matrix exponential
  \[ \exp(Qt) \equiv I + Qt + \frac{1}{2!} Q^2 t^2 + \frac{1}{3!} Q^3 t^3 + \cdots. \]

- Note that the terms \( t^k/k! \) are scalars and the terms \( Q^k \) (including \( Q^0 = I \)) are all square matrices of the same dimensions.

---

Transition probability matrix by matrix exponential (cont)

- Indeed, clearly \( \exp(Q0) = I \) and, for all \( t > 0 \),
  \[
  \frac{d}{dt} \exp(Qt) = Q + Q^2 t + \frac{1}{2!} Q^3 t^2 + \cdots
  = [I + Qt + \frac{1}{2!} Q^2 t^2 + \cdots]Q
  = \exp(Qt)Q,
  \]
  where, in the second equality, we could have instead factored \( Q \) out to the left to obtain the forward equations.

- In summary, for all \( s, t \in \mathbb{R}^+ \) such that \( s \leq t \), the distribution of \( X(t) \) is
  \[
  \pi^T(t) = \pi^T(s) P(t - s)
  = \pi^T(s) \exp(Q(t - s)).
  \]
• Consider an example where the TRM $Q$ has distinct real eigenvalues,

$$Q = \begin{bmatrix} -2 & 1 & 1 \\ 1 & -1 & 0 \\ 1 & 0 & -1 \end{bmatrix}.$$  

The corresponding transition rate diagram (TRD) is

![Diagram](image)

• The eigenvalues are the roots of $Q$’s characteristic polynomial:

$$\det(zI - Q) \equiv z(z + 1)(z + 3);$$

• Taking the eigenvalues $z \in \{0, -1, -3\}$ and then solving the right-eigenvectors $x$ from $Qx = zx$ gives:
  - $[1 \ 1 \ 1]^T$ is a right-eigenvector corresponding to eigenvalue $0$ (true for all rate matrices $Q$),
  - $[0 \ 1 \ -1]^T$ is a right-eigenvector corresponding to eigenvalue $-1$, and
  - $[2 \ -1 \ -1]^T$ is a right-eigenvector corresponding to eigenvalue $-3$.

• Combining these three statements in matrix form gives

$$Q \begin{bmatrix} 1 & 0 & 2 \\ 1 & 1 & -1 \\ 1 & -1 & -1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 2 \\ 1 & 1 & -1 \\ 1 & -1 & -1 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -3 \end{bmatrix} =: V\Lambda.$$
Computing matrix exponential by Jordan form

• Thus, we arrive at a Jordan decomposition of the matrix $Q$ for the special case of distinct eigenvalues:
  \[ Q = V \Lambda V^{-1}. \]

• So, for all integers $k \geq 1$,
  \[ Q^k = V \Lambda^k V^{-1}, \]
  where
  \[
  \Lambda^k = \begin{bmatrix}
  0 & 0 & 0 \\
  0 & (-1)^k & 0 \\
  0 & 0 & (-3)^k
  \end{bmatrix}
  \]
  \[
  \Rightarrow \exp(Qt) = V \exp(\Lambda t) V^{-1}
  \]
  \[
  = V \begin{bmatrix}
  1 & 0 & 0 \\
  0 & e^{-t} & 0 \\
  0 & 0 & e^{-3t}
  \end{bmatrix} V^{-1}.
  \]

• Note that we could have developed this example using left eigenvectors instead of right; e.g., the stationary distribution $\sigma^T$ is the left eigenvector corresponding to eigenvalue 0.

---

Stationary distribution of a Markov chain

• Suppose there exists a distribution $\sigma$ on the state space $\Sigma = \mathbb{Z}^+$ that satisfies the full balance equations
  \[
  \sigma^T Q = \sigma^T,
  \]
  \[ i.e., \sum_{n=0}^{\infty} \sigma_n q_{n,m} = 0 \text{ for all } m \in \mathbb{Z}^+, \]
  so that $\sigma$ is a nonnegative left eigenvector corresponding to $Q$’s zero eigenvalue (recall $Q 1 = 0$).

• Therefore, for all integers $k > 0$,
  \[
  \Rightarrow \sigma^T Q^k = \sigma^T, \quad \sigma^T P(t) = \sigma^T e^{Qt} = \sigma^T I = \sigma^T \forall t \in \mathbb{R}^+.
  \]

• Recall that $\pi(t)$ is defined to be the distribution of Markov chain $X(t)$.

• Therefore, if $\pi(0) = \sigma$, then $\pi(t) = \sigma$ for all real $t > 0$.

• So $\sigma$ is called a stationary or invariant distribution of the Markov chain $X$ with TRM $Q$.

• The Markov chain $X$ itself is said to be stationary if $\pi(0) = \sigma$. 

---
Stationary distribution of a Markov chain - balance equations

\[ \sigma^TQ = 0 \iff \forall \text{ states } m, \text{ probability flux into } m \text{ equals that out of } m: \]

\[ \sum_{n \neq m}^{\infty} \sigma_n q_{n,m} = \sigma_m (-q_{m,m}) \]

\[ = \sigma_m \sum_{n \neq m}^{\infty} q_{m,n} \]

Stationary distribution of a Markov chain - examples

- For the previous example 3-state TRM, the unique invariant distribution is uniform \( \sigma^T = \left[ \frac{1}{3} \quad \frac{1}{3} \quad \frac{1}{3} \right] \).

- To model the packet-flow generated by a voice source:
  - First let the talkspurt state be denoted by 1 and the silent state be denoted by 0, i.e., our modeling assumption is that successive talkspurts and silent periods are independent and exponentially distributed.
  - In steady state, the mean duration of a talkspurt is 352 ms and the mean duration of a silence period is 650 ms.
  - The mean number of packets generated per second is 22, i.e., 22 48-byte (ATM) payloads, or about 8 kbits per second on average.
  - Solving the balance equations for a two-state Markov chain gives the invariant distribution:

    \[ \sigma_0 = \frac{q_{1,0}}{q_{0,1} + q_{1,0}} \quad \text{and} \quad \sigma_1 = \frac{q_{0,1}}{q_{0,1} + q_{1,0}} \]

    - So, \( q_{1,0} = \frac{1}{0.352} \) and \( q_{0,1} = \frac{1}{0.650} \).
    - Finally, the mean transmission rate is \( 0 \cdot \sigma_0 + r \cdot \sigma_1 = 22 \) packets/s.
Existence and uniqueness of stationary distribution

• We now consider the properties of Markov chains that have bearing on the issues of existence and uniqueness of stationary distributions.

• By the definition of its diagonal entries, the sum of the columns of a rate matrix $Q$ is the zero vector.

• Thus, the balance equations $\sigma^T Q = 0^T$ are dependent.

• Obviously another requirement is $\sigma$ needs to be a PMF on the state space, $\sigma_i \geq 0$ for all $i$.

• That is, we replace one of the columns of $Q$, say the $i^{th}$, with a column all of whose entries are 1, resulting in the matrix $\tilde{Q}_i$ so that

$$\sigma^T \tilde{Q}_i = \epsilon^T,$$

where $\epsilon$ is a column vector whose entries are all zero except that the $i^{th}$ entry is 1.

• Thus, we are interested in conditions on the rate matrix $Q$ that result in the invertibility (nonsingularity) of $\tilde{Q}_i$ (for any $i$) giving a unique

$$\sigma^T = \epsilon^T \tilde{Q}_i^{-1}.$$

Doeblin’s theory for Markov chains - recurrence & transience

• First note that the quantity

$$V_X(i) \equiv \int_0^\infty 1\{X(t) = i\} dt$$

represents the total amount of time the stochastic process $X$ visits state $i$.

• A state $i$ of a Markov chain is said to be recurrent if

$$P(V_X(i) = \infty \mid X(0) = i) = 1,$$

i.e., the Markov chain will visit state $i$ infinitely often with probability 1.

• On the other hand, if

$$P(V_X(i) = \infty \mid X(0) = i) = 0,$$

i.e., $P(V_X(i) < \infty) = 1$ so that the Markov chain will visit state $i$ only finitely often, then $i$ is said to be a transient state.

• All states are recurrent in the previous example of a 3-state TRM, whereas all states are transient for the Poisson process.

• If all of the states of a Markov chain are recurrent, then the Markov chain itself is said to be recurrent.
Positive and null recurrence

• Suppose that \( i \) is a recurrent state.
• Let \( \tau_i > 0 \) be the time of the first transition back into state \( i \) by the Markov chain.
• The state \( i \) is said to be positive recurrent if
  \[ E(\tau_i \mid X(0) = i) < \infty. \]
  On the other hand, if the state \( i \) is recurrent and
  \[ E(\tau_i \mid X(0) = i) = \infty, \]
  then it is said to be null recurrent; see
• If all of the states of the (temporally homogeneous) Markov chain are positive recurrent, then the Markov chain itself is said to be positive recurrent.
• cf. the example of a birth-death Markov chain with infinite state-space and the M/M/1 queue special case.

Irreducibility

• A Markov chain \( X \) or associated TRM \( Q \) is irreducible if there is a path from any state of the transition rate diagram to any other state of the diagram.
• The following example is an irreducible transition rate diagram.
Irreducibility (cont)

- The following transition rate diagram does not have a path from state 2 to state 0; therefore, the associated Markov chain is reducible.

\[
\begin{align*}
0 & \xrightarrow{q_{0.0}} 5 \\
1 & \xrightarrow{q_{1.0}} 0 \\
2 & \xrightarrow{q_{2.1}} 1 \\
3 & \xrightarrow{q_{3.2}} 4 \\
4 & \xrightarrow{q_{4.3}} 3 \\
5 & \xrightarrow{q_{5.1}} 1 \\
3 & \xrightarrow{q_{3.4}} 2 \\
4 & \xrightarrow{q_{4.1}} 2 \\
0 & \xrightarrow{q_{0.5}} 5
\end{align*}
\]

Irreducibility (cont)

- The state space of a reducible Markov chain can be partitioned into one transient class (subset) and a number of recurrent (or "communicating") classes.

- If a Markov chain begins somewhere in the transient class, it will ultimately leave it if there are one or more recurrent classes.

- Once in a recurrent class, the Markov chain never leaves it (when a single state constitutes an entire recurrent class, it is sometimes called an absorbing state of the Markov chain).

- For the previous reducible example, \( \{0, 5\} \) is the transient class and \( \{1, 2\} \) and \( \{3, 4\} \) are recurrent classes.

- Irreducibility is a property only of the transition rate diagram (i.e., whether the transition rates are zero or not); irreducibility is otherwise not dependent on the values of transition rates.

- If the Markov chain has a finite number of states, then all recurrent states are positive recurrent and the recurrent and transient states can be determined by the TRD’s structure.
Existence and uniqueness of stationary distribution

- **Theorem:** If a continuous-time Markov chain is irreducible and positive recurrent, then there exists a unique stationary (invariant) distribution.

- In the following theorem, the associated Markov chain $X(t) \sim \pi(t)$ is not necessarily stationary.

- **Theorem:** For any irreducible and positive recurrent TRM $Q$ and any initial distribution $\pi(0)$,

$$\lim_{t \to \infty} \pi^T(t) = \lim_{t \to \infty} \pi^T(0) \exp(Qt) = \sigma^T,$$

where $\sigma$ is the (unique) invariant of $Q$.

- That is, the Markov chain will converge in distribution to its stationary $\sigma$.

- For this reason, $\sigma$ is also known as the steady-state distribution of the Markov chain $X$ with rate matrix $Q$.

Existence and uniqueness of stationary distribution (cont)

- Consistent with the previous theorem, if $Q$ is the TRM of an irreducible and positive recurrent Markov chain, then

$$\lim_{t \to \infty} \exp(Qt) = \begin{bmatrix} \sigma^T \\ \sigma^T \\ \vdots \\ \sigma^T \end{bmatrix},$$

where $\sigma$ is the unique invariant of $Q$.

- Note that this limit is a matrix of rank 1.

- Also, for any summable function $g$ on $\mathbb{Z}^+$,

$$Eg(X(t)) = \sum_{i=0}^{\infty} \pi_i(t) g(i) \to \sum_{i=0}^{\infty} \sigma_i g(i) \text{ as } t \to \infty.$$
Time-reversed Markov chain

- Consider a Markov chain $X$ on (the entire) $\mathbb{R}$ with TRM $Q$ and unique stationary distribution $\sigma$.

- The stochastic process that is $X$ reversed in time is
  \[ Y(t) \equiv X(-t) \text{ for } t \in \mathbb{R}. \]

- **Theorem:** The time-reversed Markov chain of $X$, $Y$, is itself a Markov chain and, if $X$ is stationary, the transition rate matrix of $Y$ is $R$ whose entry in the $n_{th}$ row and $m_{th}$ column is
  \[ r_{m,n} = \frac{q_{n,m} \sigma_n}{\sigma_m}, \]
  where $q_{n,m}$ are the transition rates of $X$.

- It is easy to show that the reverse-time chain $Y(t) \equiv X(-t)$ also has stationary distribution $\sigma$; clearly, this should be true since the fraction of time that $Y$ visits any given state would be the same as (the forward-time chain) $X$.

**Theorem on time-reversed Markov chains - proof**

- First note $R$ is indeed a transition rate matrix because the balance equations
  \[ \sum_{n=0}^{\infty} r_{m,n} = 0. \]

- Consider an arbitrary integer $k \geq 1$, arbitrary subsets $A, B, B_1, ..., B_k$ of $\mathbb{Z}^+$, and arbitrary times $t, s, s_1, ..., s_k \in \mathbb{R}^+$ such that $t < s < s_1 < \cdots < s_k$, i.e.,
  \[ -t > -s > -s_1 > \cdots > -s_k. \]
Theorem on time-reversed Markov chains - proof (cont)

- The transition probabilities for the reverse-time chain $Y$ are

$$P(Y(-t) \in A \mid Y(-s) \in B, Y(-s_1) \in B_1, \ldots, Y(-s_k) \in B_k) = P(X(t) \in A \mid X(s) \in B, X(s_1) \in B_1, \ldots, X(s_k) \in B_k)$$

$$= \frac{P(X(s) \in B, X(s_1) \in B_1, \ldots, X(s_k) \in B_k)}{P(X(s_k) \in B_k \mid X(t) \in A, \ldots, X(s_{k-1}) \in B_{k-1})} \times \frac{P(X(s) \in B, \ldots, X(s_{k-1}) \in B_{k-1})}{P(X(s) \in B, X(s_1) \in B_1, \ldots, X(s_{k-1}) \in B_{k-1})}$$

where the second-to-last equality is by the Markov property of $X$.

Theorem on time-reversed Markov chains - proof (cont)

- We can repeat this argument $k - 1$ more times to get

$$P(Y(-t) \in A \mid Y(-s) \in B, Y(-s_1) \in B_1, \ldots, Y(-s_k) \in B_k)$$

$$= \frac{P(X(t) \in A, X(s) \in B)}{P(X(s) \in B)} = \frac{P(X(t) \in A \mid X(s) \in B)}{P(Y(-t) \in A \mid Y(-s) \in B)}.$$

- So, we have just shown that $Y$ is Markovian.
We now want to find $R$ in terms of $Q$ and $\sigma$.

For $t < s$ (i.e., $-s < -t$), note that

\[
P(Y(-t) = n \mid Y(-s) = m) = \frac{P(X(t) = n \mid X(s) = m)}{P(X(s) = m)} \frac{P(X(t) = n)}{P(X(t) = n)}
\]

Since $X$ is stationary by assumption, this implies that

\[
p_{Y}^{\tau}(t - s) = p_{X}^{\tau}(s - t) \frac{\sigma_{n}}{\sigma_{m}},
\]

where $n \neq m$ and the left-hand side is the transition probability for $Y$.

Differentiating this equation with respect to $s - t = -t - (-s)$ and then evaluating the result at $s - t = 0$ gives

\[r_{m,n} = q_{n,m} \sigma_{n} \sigma_{m}
\]

---

**Time-reversible Markov chains and detailed balance equations**

- A Markov chain $X$ is said to be **time reversible** if

\[q_{m,n} = r_{m,n} := \frac{\sigma_{n}}{\sigma_{m}} q_{n,m} \quad \text{for all states} \quad n \neq m,
\]

i.e., the transition rates of the stationary (forward-time) Markov chain $X$, $q_{m,n}$, are the same as those of the reverse-time Markov chain $Y(t) = X(-t)$.

- These are the simplified **detailed balance equations** for a time-reversible Markov chain:

\[\sigma_{m} q_{m,n} = \sigma_{n} q_{n,m} \quad \text{for all states} \quad n \neq m.
\]

So, $X$ is time reversible if the average rate at which transitions from state $m$ to $n$ occur in reverse time equals the average rate at which transitions from state $n$ back to $m$ occur forward in time.

Many of the Markov chains subsequently considered will be time reversible.
Time-reversible Markov chains and detailed balance equations

- **Exercise:** Show that if a distribution \( \sigma \) satisfies the detailed balance equations for a rate matrix \( Q \), then it also satisfies the balance equations for the invariant distribution of \( Q \).

- Given an irreducible and positive recurrent rate matrix \( Q \), if one finds a distribution \( \sigma \) that satisfies detailed balance, the associated Markov chain is time reversible.

- That is, time reversibility is a property that holds if and only if the detailed balance equations are satisfied.

- Note that all two-state Markov chains are time reversible since the single balance equation is also a detailed balance equation.

---

Time-reversible Markov chains - examples

- The previous example 3-state TRM is trivially time-reversible since the stationary distribution is uniform and the TRM is symmetric.

- **Exercise:** Does every symmetric TRM have a uniform invariant distribution?

- Consider the following (asymmetric) TRM:

\[
Q = \begin{bmatrix}
-3 & 1 & 2 \\
1 & -2 & 1 \\
1 & 1 & -2
\end{bmatrix}.
\]

- Its invariant distribution is

\[
\sigma^T = \begin{bmatrix}
\frac{3}{12} & \frac{4}{12} & \frac{5}{12}
\end{bmatrix},
\]

so that

\[
\sigma_{1q_{1,2}} = \frac{3}{12} \cdot 1 \neq \frac{4}{12} \cdot 1 = \sigma_{2q_{2,1}}.
\]
Modeling time-series data using a Markov chain

• Consider a single sample path $X_\omega(t)$, $t \in [0, T]$, of a stationary process, where $T \gg 1$.
• We may be interested in estimating its marginal mean,
  \[ \mu \equiv \mathbb{E}X(t), \]
  by
  \[ \frac{1}{T} \int_0^T X_\omega(t) dt. \]
• If this quantity converges to the mean as $T \to \infty$ (for almost all sample paths $X_\omega$) then $X$ is said to be ergodic in the mean.
• If the stationary distribution of $X$, $\sigma$, can be similarly approximated because
  \[ \sigma_n = \lim_{T \to \infty} \frac{1}{T} \int_0^T 1\{X_\omega(t) = n\} dt, \]
  then $X$ is said to be ergodic in distribution.
• Such estimates are sometimes used even when the process $X$ is known not to be stationary assuming that the transient portion of the sample path will be negligible.

Fitting a Markov model to data - states

• Given sample path measurements, we now describe how to obtain the most likely TRM $Q$ for one or more measured sample paths (time series) of the physical process to be modeled.
• We first assume that the states themselves are readily discernible from the data.
• Quantization (aggregation) of the observed/physical states may be required to obtain a discrete state space if the physical state space is uncountable.
• Even if the physical state space is already discrete, it may be further simplified by judicious quantization/clustering.
• However, assuming the data was generated by a Markov process, excessive state aggregation may compromise its Markovian character.
Fitting a Markov model to data - pertinent statistics

• Given a space of $N$ defined states, one can glean the following information from sample-path data, $X_\omega$:
  
  – the total time duration of the sample path, $T$;
  – the total time spent in state $i$, $\tau_i$, for each element $i$ of the defined state space, i.e.,
  $$\tau_i = \int_0^T 1\{X(t) = i\}dt,$$
  – the total number of jumps taken out of state $i$ (i.e., the number of visits to state $i$), $J_i$, and
  – the total number of jumps out of state $i$ to state $j$, $J_{i,j}$.

Clearly,
$$T = \sum_i \tau_i$$
and, for all states $i$,
$$J_i = \sum_i J_{i,j}.$$

Most likely Markov model of data

• From this information, we can derive:
  
  – the sample occupation time for each state $i$,
    $$\sigma_i = \frac{\tau_i}{T} \text{ and } -\frac{1}{q_{i,i}} = \frac{\tau_i}{J_i},$$
  – the sample probability of transiting to state $j$ from $i$,
    $$r_{i,j} = \frac{J_{i,j}}{J_i}.$$

• From this derived information, we can directly estimate the “most likely” transition rates of the process:
  $$q_{i,j} = r_{i,j}(-q_{i,i}) \text{ for all } i \neq j.$$
Most likely Markov model of data (cont)

- This leaves us with the $N$ unknowns $q_{i,i}$ for $1 \leq i \leq N$.

- Want to use the $N$ quantities $\sigma_i$ to determine the residual $N$ unknowns $q_{i,i}$, but in order to do, so we need to assume that the physical process is stationary.

- If so, we can identify $\sigma$ as approximately equal to the stationary distribution of the Markov chain and so the balance equations hold:

$$\sigma^T Q = 0.$$  

- Given that the substitution $q_{i,j} = r_{i,j}(-q_{i,i})$ is used (for all $i \neq j$) in the balance equations, the result is only $N - 1$ linearly independent equations in $N$ unknowns $q_{i,i}$.

- Also consider the total “speed” of the Markov chain, i.e., the aggregate mean rate of jumps:

$$\sum_i \sigma_i(-q_{i,i}) = \frac{1}{T} \sum_i J_i.$$  

Fitting a Markov model to data - example

- For $N = 3$ states, consider sample path data leading to the following information.

- The time-duration and occupation times were observed to be:

<table>
<thead>
<tr>
<th>$T$</th>
<th>$\tau_0$</th>
<th>$\tau_1$</th>
<th>$\tau_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>20</td>
<td>50</td>
<td>30</td>
</tr>
</tbody>
</table>

- The total number of transitions out of each state were observed to be:

<table>
<thead>
<tr>
<th>$J_0$</th>
<th>$J_1$</th>
<th>$J_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>40</td>
<td>30</td>
</tr>
</tbody>
</table>

The specific transition counts $J_{i,j}$ were observed to be:

<table>
<thead>
<tr>
<th>from \ to</th>
<th>0</th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>-</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>1</td>
<td>10</td>
<td>-</td>
<td>30</td>
</tr>
<tr>
<td>2</td>
<td>20</td>
<td>10</td>
<td>-</td>
</tr>
</tbody>
</table>
• So, finding $q_{i,j}$ for all $i \neq j$ as above gives

$$Q = \begin{bmatrix}
q_{1,1} & \frac{5}{10}q_{1,1} & \frac{5}{10}q_{1,1} \\
-\frac{10}{40}q_{2,2} & q_{2,2} & -\frac{30}{40}q_{2,2} \\
-\frac{20}{30}q_{3,3} & -\frac{10}{30}q_{3,3} & q_{3,3}
\end{bmatrix}.$$ 

• Now the $q_{i,i}$ can be solved from the first two (independent) balance equations,

$$\frac{20}{100}q_{1,1} - \frac{50}{100} \cdot \frac{10}{40}q_{2,2} - \frac{30}{100} \cdot \frac{20}{30}q_{3,3} = 0,$$

$$-\frac{20}{100} \cdot \frac{5}{70}q_{1,1} + \frac{50}{100} \cdot \frac{50}{40}q_{2,2} - \frac{30}{100} \cdot \frac{20}{30}q_{3,3} = 0,$$

and the total speed equation,

$$\frac{20}{100}(-q_{1,1}) + \frac{50}{100}(-q_{2,2}) + \frac{30}{100}(-q_{3,3}) = \frac{1}{100}(10 + 40 + 30).$$

• The resulting solution is

$q_{1,1} = -\frac{72}{55}$,  $q_{2,2} = -\frac{128}{775}$,  and  $q_{3,3} = -\frac{56}{55}$.

• These are the “maximum likelihood” transition rates given the data.
Birth-death Markov chains

• We now define an important class of Markov chains on $\Sigma = \mathbb{Z}^+$ that are called birth-death processes.

• The terminology comes from Markovian population models wherein
  - $X(t)$ is the number of living individuals at time $t$,
  - a birth, represented by a state change from $i \geq 0$ to $i+1$, is at rate $q_{i,i+1} = \lambda_i$, and
  - a death, represented by a state change from $i > 0$ to $i-1$, is at rate $q_{i,i-1} = \mu_i$.

  ![Birth-death process diagram](image)

Birth-death processes with finite state space

• Consider a finite state space
  $$\Sigma = \mathbb{Z}^+_K \equiv \{0, 1, 2, ..., K\}$$

  and transition rates
  - $\lambda_i > 0$ for all $i \in \{0, 1, 2, ..., K-1\}$ and
  - $\mu_i > 0$ for all $i \in \{1, 2, ..., K\}$ but
    - $\mu_0 = 0$ and $\lambda_K = 0$.

• So, the finite birth-death process has an $(K+1) \times (K+1)$ transition rate matrix
  $$Q = \begin{bmatrix}
    -\lambda_0 & \lambda_0 & 0 & 0 & 0 & \cdots & 0 \\
    \mu_1 & -\mu_1 - \lambda_1 & \lambda_1 & 0 & 0 & \cdots & 0 \\
    0 & \mu_2 & -\mu_2 - \lambda_2 & \lambda_2 & 0 & \cdots & 0 \\
    \vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\
    0 & 0 & \cdots & 0 & \mu_{K-1} & -\mu_{K-1} - \lambda_{K-1} & \lambda_{K-1} \\
    0 & 0 & \cdots & 0 & 0 & \mu_K & -\mu_K \\
  \end{bmatrix}. \ (1)$$
Birth-death processes with finite state space (cont)

- Note that this rate matrix is irreducible.
- The finiteness of the state space implies that the birth-death process is also positive recurrent.
- We will now compute the stationary distribution $\sigma$ which is a vector of size $K + 1$ by solving
  \[
  \sigma^T Q = \Omega,
  \]
  which is a compact representation for the following system of $K + 1$ balance equations:
  \[
  \begin{align*}
  -\lambda_0 \sigma_0 + \mu_1 \sigma_1 &= 0, \\
  \lambda_{i-1} \sigma_{i-1} - (\mu_i + \lambda_i) \sigma_i + \mu_{i+1} \sigma_{i+1} &= 0 \quad \text{for } 0 < i < K, \\
  \lambda_{K-1} \sigma_{K-1} - \mu_K \sigma_K &= 0.
  \end{align*}
  \]

- Exercise: Check whether this Markov chain is time reversible and detailed balance holds.

Birth-death processes with finite state space (cont)

- The solution to these equations is given by
  \[
  \sigma_i = \sigma_0 \prod_{j=1}^{i} \frac{\lambda_{j-1}}{\mu_j} \quad \text{for } 0 < i \leq K,
  \]
  where and $\sigma_0$ is chosen as a normalizing term (i.e., so that $\sum_{n \geq 0} \sigma_n = 1$).
  \[
  \sigma_0 = \left(1 + \sum_{i=1}^{K} \prod_{n=1}^{i} \frac{\lambda_{n-1}}{\mu_n}\right)^{-1}.
  \]
- Exercise: Check whether this Markov chain is time reversible and detailed balance holds.
Birth-death processes with finite state space - example

- Consider the example where
  \[ \lambda_i = \lambda \quad \text{and} \quad \mu_i = i \cdot \mu \]
  for some positive constants \( \lambda \) and \( \mu \).
- Define the constant
  \[ \rho \equiv \frac{\lambda}{\mu} \]
- In this case the stationary distribution is a truncated Poisson,
  \[ \sigma_i = \sigma_0 \frac{\rho^i}{i!} \quad \text{for} \quad 1 \leq i \leq K \quad \text{and} \quad \sigma_0 = \left( \sum_{n=0}^{K} \frac{\rho^n}{n!} \right)^{-1}. \]

Birth-death processes with infinite state space

- The balance equations \( \mathbf{\sigma}^T \mathbf{Q} = 0 \) for an infinite state space \( \mathbb{Z}^+ \) with transition rates \( \lambda_i > 0 \) for all \( i \geq 0 \) and \( \mu_i > 0 \) for all \( i \geq 1 \) are
  \[ -\lambda_0 \sigma_0 + \mu_1 \sigma_1 = 0 \]
  and, for \( i > 0 \),
  \[ \lambda_{i-1} \sigma_{i-1} - (\mu_i + \lambda_i) \sigma_i + \mu_{i+1} \sigma_{i+1} = 0. \]
- As for the finite case, the infinite birth-death process is irreducible.
- Assuming for the moment that it is positive recurrent as well, we can solve the balance equations to get
  \[ \sigma_i = \sigma_0 \prod_{j=1}^{i} \frac{\lambda_{j-1}}{\mu_j} \quad \text{for} \quad i > 0. \]
- Choosing \( \sigma_0 \) to normalize (so \( \sum_{i=0}^{\infty} \sigma_i = 1 \)), we get
  \[ \sigma_0 = \left( 1 + \sum_{i=1}^{\infty} \prod_{n=1}^{i} \frac{\lambda_{n-1}}{\mu_n} \right)^{-1}. \]
Birth-death processes with infinite state space - recurrence

- The condition for positive recurrence is that

\[ \sum_{i=1}^{\infty} \prod_{n=1}^{i} \frac{\lambda_{n-1}}{\mu_n} < \infty \]

because \( \sigma_0 > 0 \) under this condition and, therefore, \( \sigma \) is a well-defined distribution (PMF) on \( \mathbb{Z}^+ \).

- Otherwise (i.e., if \( \sigma_0 = 0 \)), the Markov chain is null recurrent or transient (even though the Markov chain is irreducible).

Birth-death processes with infinite state space - example

- We now consider the example where \( \lambda_i = \lambda \) and \( \mu_i = \mu \) for all \( i \) and constants \( \lambda, \mu > 0 \).
- Again define the constant

\[ \rho \equiv \frac{\lambda}{\mu} \]

- The invariant is geometric when \( \rho < 1 \): \( \sigma_i = (1 - \rho)^i \rho^i \) for \( i \geq 1 \).
- Note that

\[ \sigma_0 = \left( \sum_{i=0}^{\infty} \rho^i \right)^{-1} = 1 - \rho > 0 \]

if and only if \( \rho < 1 \), which is the condition for positive recurrence in this example.
Birth-death processes with infinite state space - example (cont)

- That is, if $\lambda < \mu$ this process is positive recurrent.
- If $\lambda > \mu$ the process is transient, i.e., each state is visited only finitely often a.s.
- If $\lambda = \mu$ the process is null recurrent, i.e., though each state is visited infinitely often a.s., the expected time between visits is infinite.

A queue described by an underlying Markov chain - notation

- The previous example of a birth-death Markov chain is also called the “M/M/1” queue, where
  - The first “M” in this notation means that the job interarrival times are Memoryless; i.e., the job arrival process is a Poisson process which has exponential (memoryless) interarrival times $T_n - T_{n-1}$.
  - The second “M” means that the job service times, $S_n$, are independent and identically distributed exponential (Memoryless) random variables - also, the service times are independent of the arrival times.
  - The “1” means that there is one work-conserving server.
- The queue is implicitly assumed to have an infinite capacity to hold jobs; indeed, “M/M/1” and “M/M/1/$\infty$” specify the same queue.
- So, the M/M/1 queue is lossless.
- When a general distribution is involved, the terms “G” or “GI” are used instead of “M”; “GI” denotes general and IID.
- So, an M/GI/1 queue has a Poisson job arrival process and IID job service times of some distribution that is not necessarily exponential.
Forward-equation applications to b-d processes

- Consider an interval $J = \{i, i+1, \ldots, j-1, j\} \subset \mathbb{Z}$ of states, where $i < j$.

- Suppose a birth-death Markov chain $X$ makes a transition into the interval at state $i$ at time $t$, i.e., $X(t) = i$ and $X(t-) = i-1 \notin J$.

- Let $Z_k$ be the first time that $X$ makes a transition to state $k$ after time $t$, i.e.,
  $$Z_k = \inf\{s \geq t \mid X(s) = k\}.$$

- Note that $Z_i = t$ by the definition of $t$ above.

- Also, by the assumed temporal homogeneity, the distribution of $Z_i - t$ does not depend on $t$;

- so we take $t = 0$ to simplify notation in the following.

- Assume that the birth-death process is such that there is a positive probability that it will exit the interval $J$ at either end.

- We will show how the Kolmogorov equations can be used to compute the probability that the Markov chain exits the interval $J$ at $i$.

---

Prob. a b-d process exits an interval at a given end

- For $k = i-1, i, \ldots, j+1$, define
  $$g(k) = P(Z_{i-1} < Z_{j+1} \mid X(0) = k).$$

- So, we want $g(i)$ or $g(j)$.

- First note that $g(i-1) = 1$ and $g(j+1) = 0$.

- Now consider a positive real $\varepsilon \ll 1$

- By a forward-conditioning argument, for $i \leq k \leq j$,
  $$g(k) = \sum_m P(Z_{i-1} < Z_{j+1} \mid X(0) = k, \ X(\varepsilon) = m) \times P(X(\varepsilon) = m \mid X(0) = k)$$
  $$= \sum_m P(Z_{i-1} < Z_{j+1} \mid X(\varepsilon) = m)P(X(\varepsilon) = m \mid X(0) = k)$$
  $$= g(k)(1 + \varepsilon q_{k,k}) + \sum_{m \neq k} g(m)q_{k,m}\varepsilon + o(\varepsilon),$$

  where the second equality above is just the Markov property itself.
- Recall that

\[ q_{k,k} = - \sum_{m \neq k} q_{k,m} = -q_{k,k-1} - q_{k,k+1}. \]

- Therefore, get the following set of \( j - i + 1 \) equations in as many unknowns (\( g(k) \) for \( i \leq k \leq j \)),

\[
\sum_{m=k-1}^{k+1} g(k)q_{k,m} = 0 \quad \text{for} \quad i \leq k \leq j,
\]

with boundary conditions \( g(i - 1) = 1 \) and \( g(j + 1) = 0 \).

- The unique solution of these equations can be found by, e.g., the systematic method of \( Z \)-transforms - in particular, the desired quantity \( g(i) \) can be found.

- For the example where \( \exists \) constant \( q > 0 \) s.t., for all \( k \in J \),

\[ q_{k,k+1} = q = q_{k,k-1} \]

the solution is

\[ g(k) = Ak + B \]

for all \( k \in J \) and for some constants \( A \) and \( B \) found using the boundary conditions, i.e.,

\[
\begin{align*}
1 &= g(i - 1) = A(i - 1) + B, \\
0 &= g(j + 1) = A(j + 1) + B.
\end{align*}
\]

- Therefore, \( A = -1/(j - i + 2) \), \( B = (j + 1)/(j - i + 2) \), and

\[ g(k) = \frac{j - k + 1}{j - i + 2}. \]
Mean time to return to a given state by a b-d process

- Considering that there are only finitely many states less than a given state $i$, state $i$ is positive recurrent only if $h_i(i+1) < \infty$, where

$$h_i(j) \equiv E(Z_i | X(0) = j).$$

- Again, by using a forward-equation argument,

$$h_i(j) = \frac{1}{q_{j,j+1} + q_{j,j-1}} h_i(j+1) + \frac{q_{j,j-1}}{q_{j,j+1} + q_{j,j-1}} h_i(j-1)$$

$$= \frac{1}{\lambda + \mu} h_i(j+1) + \frac{\mu}{\lambda + \mu} h_i(j-1)$$

for all $j > i$, with (by definition)

$$h_i(i) \equiv 0.$$

- Intuitively, the first term on the right-hand side is the mean visiting time of state $j$ and that the coefficient of $h_i(j \pm 1)$ is the probability of transitioning from $j$ to $j \pm 1$ in one step.

- If we define

$$\eta_i(j) \equiv h_i(j) - h_i(j+1),$$

then above equations for $h$ become

$$\eta_i(j) = \frac{1}{q_{j,j+1} + q_{j,j-1}} \eta_i(j-1) = \frac{1}{\lambda + \mu} \eta_i(j-1).$$

- Iterating, we get

$$\eta_i(j) = \frac{1}{\lambda} \left( 1 + \frac{1}{\rho} \right) + \frac{1}{\rho} \eta_i(j-2) = \frac{1}{\lambda} \sum_{k=0}^{j-1} \frac{1}{\rho^k} + \frac{1}{\rho^{j-i}} \eta_i(i).$$

- Multiplying through by $\rho^{j-i}$ and then rewriting in terms of $h_i$ gives

$$\rho^{j-i}(h_i(j) - h_i(j+1)) = -h_i(i+1) + \frac{1}{\lambda} \sum_{k=1}^{j-1} \rho^k,$$

where we note that $\eta_i(i) = h_i(i) - h_i(i+1) = -h_i(i+1)$. 
Mean time to return to a given state by a b-d process

- Now consider this equation as \( j \to \infty \).
- First note that the difference \( h_i(j) - h_i(j + 1) \to 0 \).
- Now if \( \rho = 1 \), then clearly requires that \( h_i(i + 1) = \infty \) since the summation on the right-hand side is tending to infinity, i.e., state \( i \) and, by the same argument, all other states are not positive recurrent.
- If \( \rho < 1 \), the summation on the right-hand side converges and the left-hand side tends to zero as \( j \to \infty \) so that

\[
0 = -h_i(i + 1) + \frac{1}{\lambda} \frac{\rho}{1 - \rho},
\]

i.e.,

\[
h_i(i + 1) = \frac{1}{\mu - \lambda}.
\]

Forward-equation applications - further reading

- A more general statement along these lines for birth-death Markov chains is given at the end of Section 4.7 of [Karlin & Taylor, “A First Course...”, 2nd Ed., 1975].
- Explore use of backward equation for similar problems.
- Explore these problems for discrete-time birth-death processes.
The M/M/1 queue

- The previous example birth-death process is the M/M/1 queue with
  - Poisson job arrivals of rate $\lambda$ jobs per second and
  - identically distributed exponential service times with mean $1/\mu$ seconds that are mutually independent and independent of the arrivals.

- That is, the job interarrival times are independent and exponentially distributed with mean $1/\lambda$ seconds and, therefore, for all times $s < t$, $A(s, t)$ is a Poisson distributed random variable with mean $\lambda(t - s)$.

- The mean arrival rate of work is $\lambda/\mu$ and the service rate is one unit of work per second.

- Or, the mean service rate can be described as $\mu$ jobs per second.

- So, the queue (job) occupancy, $Q$, is a birth-death Markov process with infinite state space $\mathbb{Z}^+$. 

The M/M/1 queue (cont)

- When the traffic intensity

$$\rho \equiv \frac{\lambda}{\mu} < 1,$$

$Q$ is the positive recurrent birth-death process with $\rho$-geometric stationary distribution.

- So, the stationary mean number of jobs in (backlog of) the system is

$$L = \frac{\rho}{1 - \rho} = \frac{\lambda}{\mu - \lambda},$$

- and, by Little’s formula, the stationary mean sojourn time of jobs is

$$W = \frac{L}{\lambda} = \frac{1}{\mu - \lambda}.$$

- For the M/M/1 queue, we can obtain the stationary distribution of the sojourn time, cf. PASTA.
Certain other queuing models

- Little’s formula can be used to obtain the Pollaczek-Khintchine formula for the mean sojourn time of a job through a stationary $M/G/1$ queue,

$$\frac{\lambda E(S^2)}{2(1 - \lambda E(S))},$$

where, as above, $S$ is distributed as the service time of a job and $\lambda$ is the (Poisson) job arrival rate.

- The important example of an $M/M/K/K$ queue, i.e., a queue with $K$ servers and no waiting room, is described later.

- In many queueing models, the queue occupancy itself is not Markovian but is based on an “underlying” Markov chain, e.g., where non-exponential service times or interarrival times are modeled as i.i.d. exponential phases (PH).

- Also, a Markov embedding can be used to find the sojourn time distribution of the $M/G/1$ queue.

Generalized (strong) stochastically bounded burstiness

- Recall the notion of generalized stochastically bounded burstiness in a stationary setting.

- For stationary queues with backlog $Q$, Poisson arrivals at rate $\lambda$, and deterministic service rate $\mu$:

$$P(Q > x) \leq \frac{1}{x} E(Q) \quad \text{by Markov’s inequality}$$

$$= \frac{1}{x} \lambda \frac{\mu^{-2}}{2(1 - \lambda \mu^{-1})} := f(x)$$

where the last equality by Little’s theorem and the the Pollaczek-Khintchine formula.

- A tighter gSBB bound $f$ can be computed for the $M/D/1$ queue and for more complex types of arrival models based on Markov processes, e.g., Markov-modulated or hidden-Markov; see

  - C.-S. Chang, “Stability, Queue Length, …,” *IEEE TAC* 39(5), May 1994, and
The “arrival theorem” - Poisson Arrivals See Time Averages (PASTA)

- For a causal (nonanticipative), stationary and ergodic, and stable queue, suppose the job arrival times form a Poisson point process.

- PASTA: If $Q$ is the state of such a queuing system and $T \in \mathbb{R}$ is distributed as a Poisson arrival time, then $Q(T^-)$ is distributed as the stationary distribution of $Q$.

- To see why, let $\lambda$ be the intensity of the Poisson arrivals and consider an interval $A$ of time length $T = |A|$ and a small subinterval $a \subset A$ of length $t = |a|$. and let $N$ be the number of Poisson arrivals.

- Given a single Poisson arrival occurs in $A$, the prob. that a Poisson arrival occurs in $a$ is
  \[ P(N(a) = 1 \mid N(A) = 1) = \frac{\lambda t e^{-\lambda t}}{\lambda T e^{-\lambda T}} = \frac{t}{T}. \]
  i.e., equal to the probability that a randomly chosen (typical) time in $A$ is also in $a$.

- A rigorous proof of PASTA is based on a powerful conservation law for stationary marked point processes, Palm’s theorem.

PASTA - sojourn time of stationary M/M/1 queue

- Recall that stationary distribution (i.e., at a typical time) of the number of jobs in an M/M/1 queue with traffic intensity $\rho = \lambda/\mu < 1$ is geometric with parameter $\rho$.

- By PASTA, the distribution of the number of jobs in the queue just before the arrival time $T$ of a typical job is also geometric with parameter $\rho$:
  \[ P(Q(T^-) = i) = (1 - \rho)^i \rho^i, \quad \forall i \in \mathbb{Z}^+. \]

- Note that $Q(T) = Q(T^-) + 1 \geq 1$.

- Thus, we can obtain the distribution of the stationary sojourn time $w$ as:
  $\forall i \geq 0, \text{Erlang}(\mu, i + 1)$ (i.e., $\Gamma(\mu, i + 1) = \text{sum of } i + 1 \text{ IID exp(\mu) random variables}$) with probability $(1 - \rho)^i \rho^i$.

- Exercise: Verify that
  \[ W = \text{E}w = \frac{1}{\mu - \lambda}. \]
The stationary M/M/K/K queue

- Consider a queue with Poisson arrivals, IID exponential service times, K servers, and no waiting room.
- That is, a lossy M/M/K/K queue described by a finite-state birth-death Markov chain.
- Since the capacity to hold jobs equals the number of servers, there is no waiting room (each server holds one job).
- Again, let \( \lambda \) be the rate of the Poisson job arrivals and let \( 1/\mu \) be the mean service time of a job.
- Suppose that there are \( n \) jobs in the system at time \( t \), i.e., \( Q(t) = n \).
- As before, we can show \( Q \) is a birth-death Markov chain.

\[
\begin{array}{cccccc}
0 & 1 & 2 & \cdots & K-1 & K \\
\mu & \mu & \mu & \cdots & \mu & \mu \\
\lambda & \lambda & \lambda & \cdots & \lambda & \lambda \\
\end{array}
\]

The stationary M/M/K/K queue (cont)

- Indeed, suppose \( Q(t) = n > 0 \) and suppose that the past evolution of \( Q \) is known (i.e., \( \{Q(s) \mid s \leq t\} \) is given).
  - By the memoryless property of the exponential distribution, the residual service times of the \( n \) jobs are exponentially distributed random variables with mean \( 1/\mu \).
  - Therefore, \( Q \) makes a transition to state \( n-1 \) at rate \( n\mu \), i.e., for \( 0 < n \leq K \)
    \[
    q_{n,n-1} = n\mu.
    \]
- Now suppose \( Q(t) = n < K \).
  - Again by the memoryless property, the residual interarrival time is exponential with mean \( 1/\lambda \).
  - Therefore, \( Q \) makes a transition to state \( n+1 \) at rate \( \lambda \), i.e., for \( 0 \leq n < K \)
    \[
    q_{n,n+1} = \lambda.
    \]
- Thus, the stationary distribution of \( Q \) is the truncated Poisson given before:
  \[
  \sigma_i = \sigma_0 \frac{\rho^i}{i!} \quad \text{for } 1 \leq i \leq K, \quad \text{and} \quad \sigma_0 = \left( \sum_{i=0}^{K} \frac{\rho^i}{i!} \right)^{-1}.
  \]
Erlang’s blocking formula for the stationary M/M/K/K queue

• Now consider a stationary M/M/K/K queue.

• Suppose we are interested in the probability that an arriving job is blocked (dropped) because, upon its arrival, the system is full, i.e., every server is occupied.

• Note above that when we assumed a “lossless” queue, we meant internally lossless.

• More formally, we want to find $P(Q(T_n-) = K)$, where we recall that $T_n$ is the arrival time of the $n^{th}$ job.

• Since the arrivals are Poisson, we can invoke PASTA to get

$$P(Q(T_n-) = K) = \sigma_K = \sigma_0 \frac{\rho^K}{K!} =: \mathcal{E}(\rho, K),$$

which is called Erlang’s blocking or Erlang B formula.

• Note that the traffic intensity for this system is $\rho/K = \lambda/(\mu K)$.

• Also, the mean sojourn time of all admitted arrivals is $W = 1/\mu$.

Erlang’s blocking formula for the stationary M/M/K/K queue

• For more general (non-exponential) service time distributions, it can be shown that Erlang’s blocking formula still holds.

• Therefore, given the mean service time $1/\mu$, Erlang’s result is said to be otherwise “insensitive” to the service time distribution.

• Finally note that, by Little’s theorem, the mean number of busy servers in steady state is

$$L = \lambda(1 - \sigma_K) \frac{1}{\mu} = \rho(1 - \sigma_K),$$

where $\lambda(1 - \sigma_K)$ is the mean rate of arrivals that are admitted (by PASTA), i.e., successfully join the queue.

• Exercise: check that $L = EQ = \sum_{i=0}^{K} i \sigma_i$. 

© December 14, 2018 George Kesidis
- Modeling a call center as an M/M/K/K queue, customers calling when all servers are occupied will be blocked with probability given by Erlang’s blocking formula (indicated by slow busy signal).

- If we add a waiting room of $W \geq 1$ jobs, then we have a M/M/K/(K + W) queue, with blocking (fast busy signal) probability, by PASTA, equal to the stationary

\[
\sigma_{K+W} = P(Q = K + W) = \left(1 + \sum_{i=1}^{K+W} \prod_{n=1}^{i} \frac{\lambda_n}{\mu_n}\right)^{-1} = \left(\sum_{i=0}^{K} \frac{\rho^K}{\rho^K} + \frac{\rho^K}{K!} \sum_{j=1}^{W} \frac{\rho^K}{K!}\right)^{-1},
\]

where $\sigma_0 = P(Q = 0) = \left(1 + \sum_{i=1}^{K+W} \prod_{n=1}^{i} \frac{\lambda_n}{\mu_n}\right)^{-1}$, $\rho = \lambda/\mu$, the birth rate $\lambda_n = \lambda$, and death rate

\[
\mu_n = \begin{cases} 
  n\mu & \text{if } 1 \leq n \leq K \\
  K\mu + (n-K)\delta & \text{if } K \leq n \leq K + W
\end{cases}
\]

\[\mu_n = \begin{cases} 
  n\mu & \text{if } 1 \leq n \leq K \\
  K\mu & \text{if } K \leq n \leq K + W
\end{cases}
\]

In steady-state, the total arrival rate equals the total "departure" rates due to blocking, abandonment or successful service:

\[
\lambda = \lambda \sigma_{K+W} + \sum_{q=K+1}^{K+W} (q - K)\delta\sigma_q + \sum_{q=1}^{K+1} (q \land K)\mu\sigma_q.
\]

- So, probabilities of successful service and abandonment (departure due to impatience) are, respectively,

\[
S(K, W) := \lambda^{-1} \sum_{q=1}^{K+W} (q \land K)\mu\sigma_q \quad \text{and} \quad A(K, W) := \lambda^{-1} \sum_{q=K+1}^{K+W} (q - K)\delta\sigma_q,
\]

and again $\sigma_{K+W}$ is the probability of blocking upon arrival.
For a call center, one can consider the optimization problem of the form
\[
\max_{K,W} r_s S(K, W) - c_a A(K, W) - c_b \sigma_{K+W} - K,
\]
where \( c_a \geq 0 \), resp. \( c_b \geq 0 \), is the cost of abandoned, resp. blocked, customers per unit server, and \( r_s \) is the reward for served customers per unit server.

Normally, \( c_a > c_b \) as customer who abandons after being on hold may naturally be more irate than one who is immediately blocked.

**Exercise:** Verify that \( \sigma_{K+W} \) decreases in \( K \) and \( W \), \( A \) decreases in \( K \) but increases with \( W \), and \( S \) increases in both \( K \) and \( W \).

---

**Exercise: Delays in memoried, multiserver systems - Erlang C formula**

- Now consider an \( M/M/K \) (i.e., \( M/M/K/\infty \)) system with infinite waiting room.

- Again, the traffic intensity here is \( \lambda/(K\mu) = \rho/K \), with \( \rho/K < 1 \) required for stability.

- The Erlang C formula gives the probability that an arriving job experiences positive queueing delay:

\[
C(\rho, K) := \frac{\rho^\rho}{\sum_{k=0}^{K-1} \frac{\rho^k}{k!} + \frac{\rho^\rho}{K!(1-\rho/K)}} = \frac{\mathcal{E}(\rho, K)}{1 - \frac{\rho}{K}(1 - \mathcal{E}(\rho, K))}.
\]

**Exercise:** Use PASTA to prove the Erlang C formula.

**Exercise:** Use Little’s theorem to prove the mean sojourn time is

\[
\frac{1}{\lambda} \left( \rho + C(\rho, K) \frac{\rho/K}{1 - \rho/K} \right).
\]

- Note: The Erlang C formula works only for exponential service times, unlike the Erlang blocking (B) formula which is insensitive to service distribution type.
Markovian queueing networks with static routing

- We now introduce two classical Markovian queueing network models:
  - loss networks modeling circuit-switched networks (e.g., the former telephone network, MPLS networks) with static routing, and
  - Jackson networks that can be used to model packet-switched networks and packet-level processors, with purely randomized routing with static routing probabilities.

- Both will be shown to have “product-form” invariant distributions.
• The previous Figure depicts a network with 13 enumerated links.

• Note that the cycle-free routes connecting nodes (end systems) \( m \) and \( n \) are

\[
\begin{align*}
  r_1 &= \{1, 3, 6, 9\}, \\
  r_2 &= \{1, 4, 13, 9\}, \\
  r_3 &= \{1, 4, 5, 6, 9\}, \\
  r_4 &= \{1, 3, 5, 13, 9\},
\end{align*}
\]

where we have described each route by its link membership as above. We will return to this example in the following.

---

**Loss networks - preliminaries**

• Consider a network connecting together a number of end-systems/users.

• Bandwidth in the network is divided into fixed-size amounts called *circuits*, e.g., a circuit could be a 64 kbps channel (voice line) or a T1 line of 1.544 Mbps.

• Let \( L \) be the number of network links and let \( c_l \) circuits be the fixed capacity of network link \( l \).

• Let \( R \) be the number of distinct bidirectional routes in the network, where a route \( r \) is defined by a group of links \( l \in r \).

• Let \( \mathcal{R} \) be the set of distinct routes so that \( R = |\mathcal{R}| \).

• Finally, define the \( L \times R \) matrix \( A \) with Boolean entries \( a_{l,r} \) in the \( l^{th} \) row and \( r^{th} \) column, where

\[
a_{l,r} = \begin{cases} 
1 & \text{if } l \in r, \\
0 & \text{if } l \notin r.
\end{cases}
\]

• That is, each column of \( A \) corresponds to a route and each row of \( A \) corresponds to a link.
• The four routes $r_1$ to $r_4$ for the previous example network (of 13 links) are described by the $13 \times 4$ matrix

$$A = \begin{bmatrix}
1 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 \\
1 & 0 & 0 & 1 \\
0 & 1 & 1 & 0 \\
0 & 0 & 1 & 1 \\
1 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 1 & 0 & 1
\end{bmatrix}.$$
Loss networks - preliminaries (cont)

- If an existing connection on route $r$ terminates at time $t$,
  \[ X(t) = X(t-) - e_r. \]
- Similarly, if a connection on route $r$ is admitted into the network at time $t$,
  \[ X(t) = X(t-) + e_r. \]
- Clearly, a connection cannot be admitted (i.e., is blocked) along route $r^*$ at time $t$ if any associated link capacity constraint is violated, i.e., if, for some $l \in r^*$,
  \[ \sum_{r \mid l \in r} X_r(t-) = c_l. \]
- An $R$-vector $x$ is said to be a feasible state if it satisfies all link capacity constraints, i.e., for all links $l$,
  \[ (Ax)_l = \sum_{r \mid l \in r} x_r \in \{0, 1, 2, \ldots, c_l\}. \]
- Thus, the state space of the stochastic process $X$ is
  \[ S(\xi) = \{x \in (\mathbb{Z}^+)^R \mid Ax \leq \xi\}. \]

Loss networks - example (cont)

- For the previous network example of $L = 13$ links, note that link 1 is common to all $R = 4$ routes.
- We now illustrate how link capacities $c$ are used to determine whether route occupancies $x$ are feasible via corresponding link occupancies $Ax$.
- For example,
  \[ x = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \]
  is feasible if the capacities $c_l \geq 4$ for all links $l$ but is not feasible if $c_1 < 4$ because
  \[ (Ax)_1 = 4. \]
• In addition to assuming that each route \( r \) has Poisson connection arrivals with rate \( \lambda_r \), also assume independent and exponentially distributed connection lifetimes with mean \( 1/\mu_r \).

• Now it is easily seen that the stochastic process \( X \) is a Markov chain wherein
  - the state transition \( x \rightarrow x + e_r \in S(\xi) \) occurs with rate \( \lambda_r \) and
  - the state transition \( x \rightarrow x - e_r \in S(\xi) \) occurs with rate \( x_r \mu_r \).

• **Theorem:** The loss network \( X \) is time reversible with stationary distribution on \( S(\xi) \) given by the product form

\[
\sigma(x) = \frac{1}{G(\xi)} \prod_{r \in R} \frac{\rho_r^{x_r}}{x_r!}, \quad \text{where } \rho_r = \frac{\lambda_r}{\mu_r},
\]

\( \xi \) is the \( L \)-vector of link capacities, and

\[
G(\xi) = \sum_{x \in S(\xi)} \prod_{r \in R} \frac{\rho_r^{x_r}}{x_r!}
\]

is the normalizing term (partition function) chosen so that \( \sum_{x \in S(\xi)} \sigma(x) = 1 \).

---

**Loss networks - proof of product-form invariant**

![Diagram of a Markov process with states and transitions](image)

• Assuming \( x, x + e_r \in S(\xi) \) for some \( r \in R \), a generic detailed balance equation is

\[
\lambda_r \sigma(x) = (x_r + 1) \mu_r \sigma(x + e_r).
\]

• The theorem statement therefore follows if the claimed \( \sigma \) satisfies this equation.

• So, substituting the claimed expression for \( \sigma \) and canceling from both sides the normalizing term \( G(\xi) \) and all terms pertaining to routes other than \( r \) gives

\[
\lambda_r \frac{\rho_r^{x_r}}{x_r!} = (x_r + 1) \mu_r \frac{\rho_r^{x_r+1}}{(x_r + 1)!}.
\]

• This equation is clearly seen to be true after canceling \( x_r + 1 \) on the right-hand side, then canceling \( \rho_r^{x_r}/x_r! \) from both sides, and finally recalling \( \rho_r \equiv \lambda_r/\mu_r \).

• Note how the normalizing term \( G \) depends on \( \xi \) through the state space \( S(\xi) \).
• An arriving connection at time $t$ is admitted on route $r$ only if a circuit is available on all of $r$’s edges, i.e., only if
  
  - $(A \mathbf{X}(t-))_l \leq c_l - 1$ for all $l \in r$, where
  - $(A \mathbf{X})_l$ represents the $l^{th}$ component of the $L$-vector $A \mathbf{X}$.

• Consider the $L$-vector $A \mathbf{e}_r$, i.e., the $r^{th}$ column of $A$ whose $l^{th}$ entry is
  
  $$a_{l,r} = (A \mathbf{e}_r)_l = \begin{cases} 1 & \text{if } l \in r, \\ 0 & \text{if } l \notin r. \end{cases}$$

• Thus, the $L$-vector $c - A \mathbf{e}_r$ has $l^{th}$ entry
  
  $$c_l - (A \mathbf{e}_r)_l = \begin{cases} c_l - 1 & \text{if } l \in r, \\ c_l & \text{if } l \notin r. \end{cases}$$

• Theorem: The steady-state probability that a connection is blocked on route $r$ is
  
  $$B_r = 1 - \frac{G(c - A \mathbf{e}_r)}{G(c)}.$$

• Proof: First note that $B_r$ is 1 minus the stationary probability that the connection is admitted (on every link $l \in r$).

• Therefore, by PASTA,
  
  $$B_r = 1 - \sum_{\mathbf{z} \in A \mathbf{X} \leq c - A \mathbf{e}_r} \sigma(\mathbf{z})$$
  
  $$= 1 - \frac{1}{G(c)} \sum_{\mathbf{z} \in S(c - A \mathbf{e}_r)} \prod_{r \in R} \frac{\rho_r^{z_r}}{z_r!},$$

  from which the expression for exact blocking directly follows by definition of the normalizing term $G$. $\blacksquare$
• The computational complexity of the partition function $G$ grows rapidly as the network dimensions ($L$, $R$, $N$, etc.) grow.

• We now formulate an iterative method for determining approximate blocking probabilities under the assumption that the individual links block connections independently.

• Consider a single link $l^*$ and let $b_l$ be its unknown blocking probability.

• For the moment, assume that the link blocking probabilities $b_l$ of all other links $l \neq l^*$ are known.

• Consider a route $r$ containing link $l^*$.

• By the independent blocking assumption, the incident load (traffic intensity) of $l^*$ from this route, after blocking by all of the route’s other links has been taken into account, is

$$\rho_r \prod_{l \in r \mid l \neq l^*} (1 - b_l).$$

• Thus, the total load of link $l^*$ is reduced/thinned by blocking to

$$\sum_r \rho_r \prod_{l \in r \mid l \neq l^*} (1 - b_l) \equiv \hat{b}_{l^*}(\hat{b}_{-l^*}),$$

where $\hat{b}_{-l}$ is the $(L - 1)$-vector of link blocking probabilities not including that of link $l$.

• By the independent blocking assumption, the blocking probability of link $l^*$ must therefore be the reduced-load approximation,

$$b_{l^*} = \mathcal{E}(\hat{b}_{l^*}, c_{l^*}) \quad \forall l^* \in \{1, 2, \ldots, L\},$$

where again $\mathcal{E}$ is Erlang’s blocking formula

$$\mathcal{E}(\rho, c) \equiv \mathcal{E}_c(\rho) \equiv \frac{\rho^c / c!}{\sum_{j=0}^{c} \rho^j / j!}.$$
Fixed-point iteration for approximate connection blocking (cont)

- Clearly, the link blocking probabilities \( b \) must simultaneously satisfy the reduced load approximation for all links \( l^* \), i.e., giving a system of \( L \) equations in \( L \) unknowns.

- Approaches to numerically finding such an \( L \)-vector \( b \) include Newton’s method and the following fixed-point iteration (method of successive approximations).

- Beginning from an arbitrary initial \( b^0 \), after \( j \) iterations set
  \[
  b^j_l = \mathcal{E}(\hat{\rho}_l(b^{j-1}_l), c_l) \quad \text{for all links } l.
  \]

- Brouwer’s fixed-point theorem gives that a solution \( b \in [0, 1]^L \) exists.

- Uniqueness of the solution follows from the fact that this solution is the minimum of a convex function.

Fixed-point iteration for approximate connection blocking (cont)

- Given the link blocking probabilities \( b \), under the independence assumption, the route blocking probabilities are
  \[
  B_r = 1 - \prod_{l \in r} (1 - b_l)
  = \sum_{l \in r} b_l + o \left( \sum_{l \in r} b_l \right),
  \]
  i.e., if \( \sum_{l \in r} b_l \ll 1 \), then
  \[
  B_r \approx \sum_{l \in r} b_l.
  \]

- That is, the blocking probability is approximately additive.

- Now, instead of “jobs” (connections or calls) occupying circuits on every link of a route and without waiting rooms, in the following we will consider jobs as spatially localized packets.
Interconnection networks

- Requests arrive for connections between fabric inlet and outlet nodes/terminals.
- Connections may be blocked if there are no free paths between the requested nodes.
- As an exercise, express Example 3 of Section 1.12 as the loss-network model described above.

Stable open networks of queues

- Again consider an idealized packet-switched network where
  - the forwarding decisions, made at each node forwarding the packets (jobs), are independently random, and
  - the service times at the nodes that a given packet visits are independently random with distribution depending on the forwarding node.
- Consider a group of $N \geq 2$ lossless, single-server, work-conserving queueing stations.
- Packets at the $n^{\text{th}}$ station have a mean required service time of $1/\mu_n$ for all $n \in \{1, 2, \ldots, N\}$, and external arrival rate $\Lambda_n$, where $\Lambda_n > 0$ for at least one station $n$ (open network).
- The packet arrival process to the $n^{\text{th}}$ station is a superposition of $N + 1$ component arrival processes.
- Packets departing the $m^{\text{th}}$ station are forwarded to and immediately arrive at the $n^{\text{th}}$ station with probability $r_{m,n}$.
- Also, with probability $r_{m,0}$, a packet departing station $m$ leaves the queueing network forever; here we use station index 0 to denote the world outside the network.
- Clearly, for all $m$, $\sum_{n=0}^{N} r_{m,n} = 1$. 
The flow balance equations

- Defining $\lambda_n$ as the total arrival rate to the $n^{th}$ station, recall the flow balance equations based on the notion of conservation of flow and require that all queues are stable, i.e., $\mu_n > \lambda_n$ and

$$\lambda_n = \Lambda_n + \sum_{m=1}^{N} \lambda_m r_{m,n}, \quad \forall n \in \{1, 2, \ldots, N\},$$

or in matrix form,

$$\Delta^T (I - R) = \Delta^T \Rightarrow \Delta^T = (I - R)^{-1} \Delta^T < \mu^T$$

- Also recall the conditions under which $I - R$ is nonsingular, including that $r_{m,0} > 0$ for at least one station $m$ (open network), so $R$ is a strictly sub-stochastic matrix.

- Exercise: Show there is “aggregate” flow balance between the outside world and the network:

$$\sum_{m \neq 0} \Lambda_m = \sum_{m \neq 0} \lambda_m r_{m,0}.$$
Open Jackson networks - Markovian transition rates

- Consider a vector $\mathbf{x} = (x_1, \ldots, x_N)^T \in (\mathbb{Z}^+)^N$ and define the following operator $\delta$ mapping $(\mathbb{Z}^+)^N \rightarrow (\mathbb{Z}^+)^N$.

- If $x_m > 0$ and $1 \leq n, m \leq N$, then $\delta_{m,n}$ represents a packet departing from station $m$ and arriving at station $n$:
  
  $$ (\delta_{m,n} \mathbf{x})_i = \begin{cases} 
  x_i & \text{if } i \neq m, n, \\
  x_m - 1 & \text{if } i = m, \\
  x_n + 1 & \text{if } i = n, 
  \end{cases} $$

  i.e., $\delta_{m,n} \mathbf{x} \equiv \mathbf{x} - e_m + e_n$.

- If $x_m > 0$ and $1 \leq m \leq N$, then $\delta_{m,0}$ represents a packet departing the network to the outside world from station $m$:
  
  $$ (\delta_{m,0} \mathbf{x})_i = \begin{cases} 
  x_i & \text{if } i \neq m, \\
  x_m - 1 & \text{if } i = m. 
  \end{cases} $$

- If $1 \leq n \leq N$, then $\delta_{0,n}$ represents a packet arriving at the network at station $n$ from the outside world:
  
  $$ (\delta_{0,n} \mathbf{x})_i = \begin{cases} 
  x_i & \text{if } i \neq n, \\
  x_n + 1 & \text{if } i = n. 
  \end{cases} $$

Open Jackson networks - Markovian transition rates (cont)

- In the following TRD fragment: $\Lambda_m > 0$ for the transition at left; and both $x_m > 0$ and $r_{m,n} > 0$ for the transition at right:
  
  ![Diagram](https://via.placeholder.com/150)

- For example, for $N = 4$ stations, suppose the network is currently in state $\mathbf{x} = [17 \ 5 \ 0 \ 6]^T$, so that:
  
  - Assuming $\Lambda_1 > 0$, an exogenous arrival to station 1 causes transition
    
    $$ \mathbf{x} \xrightarrow{\Lambda_1} \delta_{0,1} \mathbf{x} = [18 \ 5 \ 0 \ 6]^T $$
  
  - Assuming $r_{2,4} > 0$, a departure from station 2 that arrives to station 4 causes transition
    
    $$ \mathbf{x} \xrightarrow{\mu_{2,4}} \delta_{2,4} \mathbf{x} = [17 \ 4 \ 0 \ 7]^T $$
  
  - A departure from station 3 is impossible because it’s empty ($x_3 = 0$)
The transition rate matrix of the Jackson network is given by the following equations:

\[ q(\underline{x}, \delta_{m,n} \underline{x}) = \begin{cases} 
\mu_m r_{m,n} & \text{if } 1 \leq m \leq N, \ 0 \leq n \leq N, \ x_m > 0, \\
\Lambda_n & \text{if } m = 0, \ 1 \leq n \leq N. 
\end{cases} \]

Theorem: The stationary distribution of an open Jackson network is product form,

\[ \sigma(\underline{x}) = \frac{1}{G} \prod_{n=1}^{N} \rho_n^{x_n}, \]

where, for all \( n \), the traffic intensity

\[ \rho_n \equiv \frac{\lambda_n}{\mu_n} < 1 \]

for stability and the normalizing term (partition function) is

\[ G = \sum_{\underline{x} \in (\mathbb{Z}^+)^N} \prod_{n=1}^{N} \rho_n^{x_n}. \]

Proof of Jackson’s theorem

We need to verify the (full) balance equations for the claimed invariant distribution \( \sigma \),

\[ \forall \underline{x}, \sum_{\underline{y}} \sigma(y)q(y,x) = 0 \]

Recall the balance equations for the Jackson network “at” state \( \underline{x} \), i.e., corresponding to \( \underline{x} \)’s column in the network’s transition rate matrix.

To do this, we consider states from which the network makes transitions into \( \underline{x} \), including:

- \( \delta_{n,m} \underline{x} \) for stations \( n \) such that \( x_n > 0 \), where
  \[ q(\delta_{n,m} \underline{x}, \underline{x}) = \mu_m r_{m,n}; \]
- \( \delta_{0,n} \underline{x} \) for all stations \( n \), where
  \[ q(\delta_{0,n} \underline{x}, \underline{x}) = \Lambda_n; \]
- all other transitions that do not occur in one step, i.e., \( q = 0. \)

Note that \( \delta_{n,m} \delta_{n,m} \underline{x} = \underline{x} \).
Proof of Jackson’s theorem (cont)

- Therefore, the balance equations at $x$ are

$$
\sum_{\{n \mid x_n > 0\}} \left[ q(\delta_{n,0}, x)\sigma(\delta_{n,0}, x) + \sum_{m=1}^{N} q(\delta_{n,m}, x)\sigma(\delta_{n,m}, x) \right] \\
+ \sum_{m=1}^{N} q(\delta_{0,m}, x)\sigma(\delta_{0,m}, x) \\
= \left( \sum_{\{m \mid x_m > 0\}} \left[ \mu_m \sigma(x) + \sum_{n=1}^{N} \mu_m \sigma(\delta_{m,n}, x) \right] + \sum_{n=1}^{N} \Lambda_n \right) \sigma(x).
$$

Proof of Jackson’s theorem (cont)

- Substituting the transition rates we get

$$
\sum_{\{n \mid x_n > 0\}} \left[ \Lambda_n \sigma(\delta_{n,0}, x) + \sum_{m=1}^{N} \mu_m r_{m,n} \sigma(\delta_{n,m}, x) \right] \\
+ \sum_{m=1}^{N} \mu_m r_{m,0} \sigma(\delta_{0,m}, x) \\
= \left( \sum_{\{m \mid x_m > 0\}} \left[ \mu_m r_{m,0} + \sum_{n=1}^{N} \mu_m r_{m,n} \right] + \sum_{n=1}^{N} \Lambda_n \right) \sigma(x) \\
= \left( \sum_{\{m \mid x_m > 0\}} \mu_m r_{m,0} + \sum_{n=1}^{N} \Lambda_n \right) \sigma(x)
$$
Proof of Jackson’s theorem (cont)

• The theorem is proved if we can show that the claimed product-form invariant distribution \( \sigma \) satisfies this balance equation at every state \( x \).

• Substituting it and factoring out \( \sigma(x) \) on the left-hand side, we get

\[
\sum_{\{n \mid x_n > 0\}} \left[ \frac{1}{\rho_n} + \sum_{m=1}^{N} \frac{\mu_m r_{m,n} \rho_m}{\rho_n} \right] + \sum_{m=1}^{N} \frac{\mu_m r_{m,0} \rho_m}{\rho_n} = \sum_{\{m \mid x_m > 0\}} \mu_m + \sum_{n=1}^{N} \Lambda_n.
\]

• Substituting \( \lambda_m = \mu_m \rho_m \), we get

\[
\sum_{\{n \mid x_n > 0\}} \left[ \frac{1}{\rho_n} \left( \Lambda_n + \sum_{m=1}^{N} \lambda_m r_{m,n} \right) \right] + \sum_{m=1}^{N} \lambda_m r_{m,0} = \sum_{\{m \mid x_m > 0\}} \mu_m + \sum_{n=1}^{N} \Lambda_n.
\]

• Finally, substitution of the flow balance equations implies this equation does indeed hold. \( \square \)

• Note that the product form implies the queue occupancies are statistically independent in steady state.

Jackson’s theorem - exercise

• If the network has the following properties

\[ r_{m,n} > 0 \iff r_{n,m} > 0 \quad \text{and} \quad \Lambda_n > 0 \iff r_{n,0} > 0, \]

determine whether Jackson’s theorem for product-form invariant distribution holds by detailed balance, i.e., whether the open Jackson network is time-reversible.
If this Jackson network is stable with the forwarding probabilities and exogenous arrival rates indicated, the steady-state distribution of the number of jobs in e.g. the third queue is

\[ P(Q_3 = k) = \rho_3^k (1 - \rho_3), \]

where \( \rho_3 = \lambda_3 / \mu_3 < 1 \) and \( \lambda_3 \) is found by solving the flow-balance equations

\[ \Delta^T = (I - R)^{-1} \Delta = \left[ \begin{array}{ccc} 1 & -r_{1,2} & -r_{1,3} \\ -r_{2,1} & 1 & -r_{2,3} \\ -r_{3,1} & -r_{3,2} & 1 \end{array} \right]^{-1} \begin{bmatrix} \Lambda_1 \\ \Lambda_2 \\ 0 \end{bmatrix} < \mu^T, \]

where \( r_{3,1} + r_{3,2} = 1 - r_{3,0} < 1 \).

Thus, the mean number of jobs in the third queue is \( L_3 := EQ_3 = \rho_3 / (1 - \rho_3) \).

Little’s formula and open Jackson networks

To find the mean sojourn time \( W \) of jobs through the network in steady-state:

1. First use Jackson’s theorem to find the mean number of jobs at each station \( n \),
   \[ L_n = EQ_n = \rho_n / (1 - \rho_n). \]
2. Then use Little’s formula \( W = \sum_{n=1}^N L_n / \sum_{n=1}^N \Lambda_n \)

If instead we are just interested in the mean sojourn time \( W^{(k)} \) of jobs arriving from the outside world to station \( k \) (i.e., class-\( k \) jobs):

1. Solve the flow balance equations for each class of jobs
   \[ (\Delta^{(k)})^T = (I - R)^{-1} e_k^T \Lambda_k \]
   where \( e_k \) is the unit vector with a 1 in the \( k \)th entry and otherwise zero entries
2. So, the average number of class-\( k \) jobs in station \( n \) is
   \[ L_n^{(k)} := \frac{\lambda_n^{(k)} L_n}{\sum_{j=1}^N \lambda_j^{(j)} L_n} = \frac{\lambda_n^{(k)} L_n}{\lambda_n L_n} \text{ with } L_n = \frac{\rho_n}{1 - \rho_n} \text{ as above} \]
3. Finally, by Little’s formula \( W^{(k)} = \sum_{n=1}^N L_n^{(k)} / \Lambda_k \)
Overview of discrete-time Markov chains

- We now consider Markov processes in discrete time on countable state spaces, i.e., discrete-time Markov chains.

- We covered continuous time Markov chains first because applications are somewhat simpler.

- For example, in a queueing network operating in discrete time, it would be possible, e.g., that an arrival occurs at one station, while a departure from a second station arrives to a third, all in the same (discrete) time-slot.

- Recall that a stochastic process $X$ is said be “discrete time” if its time domain is countable, e.g., $\{X(n) \mid n \in D\}$ for $D = \mathbb{Z}^+$ or for $D = \mathbb{Z}$, where, in discrete time, we will typically use $n$ instead of $t$ to denote time.

- In discrete time, the Markov property is defined as in continuous time, relying on the memoryless property of the (discrete) geometric distribution - see http://www.cse.psu.edu/~kesidis/teach/Prob-4.pdf
Markovian counting process in discrete-time

- If the random variables $B(n)$ are IID Bernoulli distributed for, say, $n \in \mathbb{Z}^+$, then $B$ is said to be a Bernoulli process on $\mathbb{Z}^+$.
- Assume that for all time $n$, constant $P(B(n) = 1) := q$.
- Thus, the duration of time $B$ visits state 1 (respectively, state 0), is geometrically distributed with mean $1/(1 - q)$ (respectively mean $1/q$).
- The analog to the Poisson counting process can be constructed on $\mathbb{Z}^+$:
  \[
  X(n) = \sum_{m=0}^{n} B(m).
  \]
- The marginal distribution of $X$ follows a binomial distribution: for $k \in \{0, 1, 2, ..., n\}$,
  \[
  P(X(n-1) = k) = \binom{n}{k} q^k (1 - q)^{n-k}.
  \]
- Recall the law of small numbers relating the binomial to Poisson distributions:
  http://www.cse.psu.edu/~kesidis/teach/Prob-4.pdf
- The arrival theorem (PASTA in continuous time) holds for the Bernoulli process in discrete time.

One-step transition probabilities

- The one-step transition probabilities of a discrete-time Markov chain $Y$ are defined to be
  \[
  P(Y(n+1) = a \mid Y(n) = b),
  \]
  where $a, b$ are, of course, taken from the countable state space of $Y$.
- If these one-step transition probabilities do not depend on time $n$, then the Markov process $Y$ is time homogeneous.
- The one-step transition probabilities of a time-homogeneous Markov chain can be graphically depicted in a transition probability diagram (TPD).
- The transition probability diagrams of $B$ and $X$ are given below.
- Note that the nodes are labeled with elements in the state space and the branches are labeled with the one-step transition probabilities.
- Graphically unlike TRDs, TPDs may have “self-loops”, e.g.,
  \[
  P(X(n+1) = 1 \mid X(n) = 1) = q > 0.
  \]
Transition probability matrices (TPMs)

- From one-step transition probabilities of a discrete-time Markov chain, one can construct its transition probability matrix (TPM).

- i.e., the entry in the $a^{th}$ column and $b^{th}$ row of the TPM $\mathbf{P}(n+1)$ for $Y$ is $P(Y(n+1) = a \mid Y(n) = b) = P_{b,a}(n+1)$.

- For example, the Bernoulli process $B$ has state space $\{0, 1\}$ and TPM

$$\mathbf{P} = \begin{bmatrix} 1 - q & q \\ 1 - q & q \end{bmatrix}.$$

- That of the counting process $X$ defined above is

$$\mathbf{P} = \begin{bmatrix} 1 - q & q & 0 & 0 & 0 & \cdots \\ 0 & 1 - q & q & 0 & 0 & \cdots \\ 0 & 0 & 1 - q & q & 0 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}.$$

- Note that the previous two examples are time homogeneous.

Example transition probability matrix on $\{0, 1, 2\}$

Another example of a discrete-time Markov chain with state space $\{0, 1, 2\}$ and TPM

$$\mathbf{P} = \begin{bmatrix} 0.3 & 0.2 & 0.5 \\ 0.5 & 0 & 0.5 \\ 0.1 & 0.2 & 0.7 \end{bmatrix}.$$
TPMs are stochastic matrices

- All TPMs $P$ are row-stochastic matrices, i.e., they satisfy the following two properties:
  - All entries are nonnegative and real (the entries are all probabilities).
  - The sum of the entries of any row is 1, i.e., $P1 = 1$ by the law of total probability.
- Clearly, all such matrices have eigenvalue 1 with
- non-negative associated left eigenvector, which is of interest in the following.
- So, the PMF of the transition from state $k$ is given by
  - the $k^{th}$ row of the TPM, or
  - the labels of the out-bound arrows of state $k$ in the TPD.

TPMs and marginal distributions

- Given the TPM $P$ and initial distribution $\pi(0)$ of the process $Y$ (i.e., $P(Y(0) = k) =: \pi_k(0)$), one can easily compute the other marginal distributions of $Y$.
- For example, by conditioning on $Y(0)$ we can compute the distribution of $Y(1)$ as $\pi^T(1) = \pi^T(0)P$, i.e., for all $k$ in the state space $S$ of $Y$:

\[ \pi_k(1) := P(Y(1) = k) = \sum_{b \in S} P(Y(1) = k, Y(0) = b) \]
\[ = \sum_{b \in S} P(Y(1) = k \mid Y(0) = b)P(Y(0) = b) \]
\[ = \sum_{b \in S} P_{b,k} \pi_b(0) = (\pi^T(0)P)_k \]

- By induction, we can compute the distribution of $Y(n)$:

\[ \pi^T(n) = \pi^T(0)P^n. \]
- The quantity $P^n$ can be computed using similarity transform to its diagonal matrix of Jordan blocks.
- General finite-dimensional distributions can be found by sequential conditioning.
Time-inhomogeneous Markov chains

- Note that a time-inhomogeneous discrete-time Markov chain will simply have time-dependent transition probabilities.

- If \( P(n) \) the one-step TPM of \( Y \) from time \( n - 1 \) to time \( n \), then the distribution of \( Y(n) \) is

\[
\pi^T(n) = \pi^T(0)P(1)P(2)\cdots P(n).
\]

Forward Kolmogorov equations

For a time-inhomogeneous Markov chain \( Y \), the forward Kolmogorov equations in discrete-time can be obtained by conditioning on \( Y(1) \):

\[
\begin{align*}
(P(0,n))_{a,b} &\equiv P(Y(n) = a \mid Y(0) = b) \\
&= \sum_k \frac{P(Y(n) = a, Y(0) = b, Y(1) = k)}{P(Y(0) = b)} \\
&= \sum_k \frac{P(Y(n) = a, Y(0) = b, Y(1) = k)P(Y(1) = k, Y(0) = b)}{P(Y(1) = k, Y(0) = b)P(Y(0) = b)} \\
&= \sum_k P(Y(n) = a \mid Y(0) = b, Y(1) = k)P(Y(1) = k \mid Y(0) = b) \\
&= \sum_k P(Y(n) = a \mid Y(1) = k)P(Y(1) = k \mid Y(0) = b),
\end{align*}
\]

where the second-to-last equality is the Markov property.
Kolmogorov equations in Matrix form

- The Kolmogorov forward equations in matrix form are
  \[ P(0, n) = P(1)P(1, n). \]
- Similarly, the backward Kolmogorov equations are generated by conditioning on \( Y(n - 1) \):
  \[ P(0, n) = P(0, n - 1)P(n). \]
- Note that both are consistent with \( P(0, n) \equiv P(1)P(2) \cdots P(n) \),
- which simply reduces to \( P(0, n) = P^n \) in the time-homogeneous case.

Invariant distribution for the time-homogeneous case

- For a time-homogeneous Markov chain, we can define an invariant or stationary distribution of its TPM \( P \) as any distribution \( \sigma \) satisfying the balance equations in discrete time:
  \[ \sigma^T = \sigma^TP \]
  with \( \sum_i \sigma_i = \sigma^T1 = 1 \) and \( \forall i, \sigma_i \geq 0 \).
- Clearly, if the initial distribution \( \pi(0) = \sigma \) for a stationary distribution \( \sigma \), then \( \pi(1) = \sigma \)
  as well and, by induction, the marginal distribution of the Markov chain is \( \sigma \) forever,
- \( i.e., \pi(n) = \sigma \) for all time \( n > 1 \) and the Markov chain is stationary.
Invariant distribution - examples

- The counting process $X$ with binomially distributed marginal does not have an invariant distribution as it is transient.

- By inspection, the stationary distribution of the Bernoulli Markov chain is
  
  $$\sigma = \begin{bmatrix} 1 - q \\ q \end{bmatrix}.$$  

- The stationary distribution of the previous TPM on \{0, 1, 2\} is unique because it’s positive recurrent (only finite number of states), irreducible, and aperiodic.

- The invariant can be computed by solving
  
  $$\sigma^T(I - P) = 0,$$
  
  $$\sigma^T1 = 1.$$  

- Note that the first block of equations (three in this example) are equivalent to $\sigma^T = \sigma^TP$ and are linearly dependent, i.e., $I - P$ is singular since $P$ is row stochastic.

Example - computing an invariant distribution (cont)

- We can replace one of the columns of $I - P$, say column 3, with all 1’s (corresponding to $1 = \sigma^T1 = \sigma_0 + \sigma_1 + \sigma_2$) and replace $\sigma$ with $[0 0 1]^T$ to obtain three linearly independent equations:
  
  $$\sigma^T \begin{bmatrix} 0.7 & -0.2 & 1 \\ -0.5 & 1 & 1 \\ -0.1 & -0.2 & 1 \end{bmatrix} = [0 0 1] \Rightarrow \sigma^T = [0.20833 0.16667 0.625]$$  

- Suppose that this Markov chain on \{0, 1, 2\} has an initial distribution that is uniform, i.e., $\pi^T(0) = [1/3 \ 1/3 \ 1/3]$.

- The distribution at time 2 is
  
  $$\pi^T(2) = \pi^T(0)P^2 = \pi^T(0) \begin{bmatrix} 0.24 & 0.16 & 0.6 \\ 0.2 & 0.2 & 0.6 \\ 0.2 & 0.16 & 0.64 \end{bmatrix} = [0.21333 0.17333 0.61333]$$  

- So, we see that after just two time steps from uniform initial $\pi(0)$, the distribution is approximately the invariant, $\pi(2) \approx \sigma$.  

© December 14, 2018 George Kesidis

260
Recurrence, irreducibility and periodicity

- Individual states of a discrete-time Markov chain can be null recurrent, positive recurrent, or transient.
- We can call the Markov chain itself “positive recurrent” if all of its states are.
- Also, a discrete-time Markov chain can possess the irreducible property.
- Unlike continuous-time chains, all discrete-time chains also possess either a periodic or an aperiodic property through their TPDs (as with the irreducibility property).

Periodicity

- A state \( b \) of a time-homogeneous Markov chain \( Y \) is periodic if there is a time \( n > 1 \) such that:
  \[
  P(Y(m) = b \mid Y(0) = b) > 0 \iff m \text{ is a multiple of } n,
  \]
  where \( n \) is the period of \( b \).
- That is, given \( Y(0) = b \), \( Y(m) = b \) is only possible when \( m = kn \) for some integer \( k \).
- A Markov chain is said to be aperiodic if it has no periodic states; otherwise it is said to be periodic.
- The examples of discrete-time Markov chains considered previously are all aperiodic.
Periodicity - example

- This Markov chain is periodic with $n = 2$ being the period of state 2.

\[ \begin{align*}
0 & \xrightarrow{0.4} 1 \\
1 & \xrightarrow{0.6} 2 \\
2 & \xrightarrow{0.6} 1 \\
1 & \xrightarrow{0.4} 0 \\
0 & \xrightarrow{0.6} 2 \\
2 & \xrightarrow{0.6} 1 \\
1 & \xrightarrow{0.4} 0 \\
0 & \xrightarrow{0.6} 2 \\
2 & \xrightarrow{0.6} 1 \\
1 & \xrightarrow{0.4} 0
\end{align*} \]

- One can solve for the invariant distribution of this Markov chain to get the unique $\boldsymbol{\sigma} = [0.2 \ 0.3 \ 0.5]^T$.
- but the Markov chain is not stationary because, e.g., if $X(0) = 2$, then $X(n) = 2$ almost surely (i.e., $P(X(n) = 2 \mid X(0) = 2) = 1$) for all even $n$ and $X(n) \neq 2$ a.s. for all odd $n$.

Existence and uniqueness of invariant distribution

- **Theorem:** A time-homogeneous discrete-time Markov chain has a unique stationary (invariant) and steady-state distribution if and only if it is irreducible, positive recurrent and aperiodic.
- The proof of this basic statement of Doeblin is given in the 1968 book by Feller.
- The unique invariant $\boldsymbol{\sigma}$ is also the unique steady-state distribution because: if $P$ is the TPM (of an irreducible, positive recurrent and aperiodic Markov chain), then

\[
\lim_{n \to \infty} P^n = \begin{bmatrix}
\sigma_1^T \\
\sigma_2^T \\
\vdots \\
\sigma_k^T
\end{bmatrix}
\]

- Thus, for any initial distribution $\pi(0)$, $\lim_{n \to \infty} \pi(0)P^n = \sigma^T$. 

© December 14, 2018 George Kesidis
Birth-death Markov chains with finite state-space

- The counting process $X$ defined above is a “pure birth” process on $\mathbb{Z}^+$. 

- This TRD of a birth-death process on a finite state space $\{0, 1, \ldots, K\}$ (naturally assuming $q_k + p_k \leq 1$ for all $k$, where $p_0 = 0$ and $q_K = 0$).

Birth-death Markov chains with finite state-space (cont)

- The balance equations are

$$
(1 - q_0)\sigma_0 + p_1\sigma_1 = \sigma_0,
$$

$$
q_{n-1}\sigma_{n-1} + (1 - q_n - p_n)\sigma_n + p_{n+1}\sigma_{n+1} = \sigma_n \quad \text{for } 0 < n < K,
$$

$$
q_{K-1}\sigma_{K-1} + (1 - p_K)\sigma_K = \sigma_K,
$$

whose solutions are

$$
\sigma_i = \frac{\sigma_0}{\prod_{j=1}^{i} \frac{q_{j-1}}{p_j}}
$$

for $0 < i \leq K$ and $\sigma_0$ is chosen as a normalizing term

$$
\sigma_0 = \left(1 + \sum_{i=1}^{K} \prod_{n=1}^{i} \frac{q_{n-1}}{p_n}\right)^{-1}.
$$

- The example with $q_n \equiv q$ and $p_n = np$ again yields a truncated Poisson distribution for $\sigma$ with parameter $\rho = q/p$. 

Birth-death process on an infinite state-space

- The process will be positive recurrent if and only if
  \[ R \equiv \sum_{i=1}^{\infty} \prod_{n=1}^{i} q_{n-1} / p_{n} < \infty, \]
  in which case \( \sigma_0 = (1 + R)^{-1} \) and
  \[ \sigma_n = \sigma_0 \prod_{j=1}^{i} q_{j-1} / p_{j}. \]

- The example where \( p_n = p \) and \( q_n = q \) also yields a geometric, invariant stationary distribution with parameter \( \rho = q / p < 1 \).

Discrete-time M/M/1 queue

- Consider a FIFO queue with a single nonidling server and infinite waiting room in discrete time.
- Suppose that the job interarrival times are IID geometrically distributed with mean \( 1 / q \).
- The service times of the jobs are also IID geometric with mean \( 1 / p \), where \( \rho = q / p < 1 \).
- So, the number of jobs in the queue \( Q \) is a birth-death Markov chain, i.e., a discrete-time M/M/1 queue.
- From the invariant \( \text{geom}(\rho) \) distribution \( \sigma \), the mean number of jobs in the queue is
  \[ L = \sum_{k=0}^{\infty} i \sigma_i = \frac{\rho}{1 - \rho}. \]
- Thus, by Little’s formula in discrete time, the mean sojourn time is \( L / q = 1 / (p - q) \).
Discrete-time M/M/1 queue - simultaneous events

- However, our model of the discrete-time M/M/1 queue is not quite right as stated,
- because it’s possible that an arrival and departure occur simultaneously.
- For example, a one-step transition from state $k > 0$ to state $k + 1$ is the event that an
  arrival occurs but a departure does not, i.e., with one-step transition probability $q(1 - p)$.
- Considering such simultaneous events, the one-step TPM of the M/M/1 queue is

\[
P = \begin{bmatrix}
1 - q & q & 0 & 0 & 0 & \cdots \\
p(1 - q) & q + (1 - q)(1 - p) & q(1 - p) & 0 & 0 & \cdots \\
0 & p(1 - q) & q + (1 - q)(1 - p) & q(1 - p) & 0 & \cdots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \ddots 
\end{bmatrix}.
\]

- Exercise: Find the invariant distribution and mean sojourn time and compare to that of
  $\text{geom}(\rho)$.
- Exercise: Explore the discrete-time M/M/1 queue. Is it a birth-death Markov chain?

Discrete-time queues with constant service-rate - event ordering

- To show how the order in which events are accounted may impact a discrete-time queuing
  model,
- we now repeat a deterministic analysis for a single-server queue but in discrete time $n \in \mathbb{Z}^+$
  or $n \in \mathbb{Z}$.
- Suppose that the server works at a normalized rate of $c$ jobs per unit time and that $a(n)$
  is the amount of work that arrives at time $n$.
- If we assume that, in a given unit of time, service on the queue is performed prior to
  accounting for arrivals (in that same unit of time), then the work to be done at time $n$ is

\[
W(n) = (W(n - 1) - c)^+ + a(n),
\]

where, again,

\[
(\xi)^+ \equiv \max\{0, \xi\}.
\]
- Thus, work cannot begin on a job in the time-slot in which it arrives.
Cut-through discrete-time queues with constant service-rate

- Alternatively, if the arrivals are counted before service in a time slot,
  \[ W(n) = (W(n-1) - c + a(n))^+; \]
  these dynamics are sometimes called "cut-through" because it’s possible that arrivals to empty queues can depart immediately, incurring no delay.

- By induction, under cut-through
  \[ W(n) = \max_{-\infty < m \leq n} [A(m, n) - c(n - m)] , \]
  where
  \[ A(m, n) \equiv a(m + 1) + a(m + 2) + \ldots + a(n). \]

- For the dynamics without cut-through,
  \[ W(n) = a(n) + \max_{-\infty < m \leq n} [A(m, n) - c(n - m)] , \]
  where
  \[ A(m, n) \equiv a(m + 1) + a(m + 2) + \ldots + a(n - 1). \]

- One can show that the time \( m \) that achieves the maximum is the start time of the workload’s busy period of the queue that contains time \( n \).

Markov models of discrete-time queues with constant service-rate

- Suppose \( c, W(0) \in \mathbb{Z}^+ \) and that \( a \) is a stationary, \( \mathbb{Z}^+ \)-valued process such that
  \[ c > E_a(n), \]
  i.e., so that the \( W \) queue is stable.

- Given the (stationary) distribution \( \alpha \) of \( a(n) \), we can compute \( W \)’s TPM on \( \mathbb{Z}^+ \).

- For the case of cut-through: for all \( j, i \in \mathbb{Z}^+ \),
  \[
P(W(n) = j | W(n-1) = i) = \sum_{k \geq 0} \alpha_k 1\{j = (i - c + k)^+\} \]
  \[ = \sum_{k \geq 0} \alpha_k 1\{j = (i - c + k)^+\} \]
Discussion - modeling queueing networks in discrete-time

- Again, in a queueing network operating in discrete time, it would be possible, e.g., that an arrival occurs at one station, while a departure from a second station arrives to a third, all in the same (discrete) time-slot.

- So, a discrete-time analog of a continuous-time model would not simply be the “jump chain” of transitions of the latter (i.e., for all states \( a \neq b \), the TPM \( P_{a,b} = Q_{a,b}/Q_{a,a} \) so that \( \forall a, P_{a,a} = 0 \), where \( Q \) is the TRM of the continuous-time Markov model of the network).

- Rather, a much larger number of state transitions would need to be considered to account for the possibility of such simultaneous events in discrete time.

- Moreover, the order of occurrence of such simultaneous events in a time slot (unit of discrete time) would need to be specified to clarify the the dynamics of the system state.

Example of fitting a discrete-time Markov chain to data

- Consider a known/given corpus of typical passwords which a hacker could use to guess at a password, i.e., a “dictionary attack.”

- Each password, an ordered list of alpha-numeric characters, is modeled as the trajectory of a common Markov chain modeling (generating) the given corpus.

- In a second-order model, the state of the Markov chain is an ordered pair (bigram) of characters, e.g., "1a", "bS", "dA", "%2".

- We can augment the character set to include a symbol, say \( \varepsilon \) indicating the termination of the password, i.e., all bigrams of the form "\( x\varepsilon \)" are absorbing: \( P_{x\varepsilon,x\varepsilon} \equiv 1 \).

- Using the corpus, directly count the number of times
  - \( N_{xy} \) that each bigram \( xy \) appears (anywhere in the password),
  - \( N_{xyz} \) that each trigram \( xyz \) appears.

- Define the Markov transition probabilities on bigrams, \( P_{xy,yz} = N_{x,y,z}/N_{xy} \).

- Also, let \( \pi_{xy} \) be the fraction of the corpus’ passwords beginning with the bigram \( xy \).
Rejecting passwords using a generative model

- Let $w(k)$ be the $k^{th}$ character of password $w$ and $l(w)$ be the length of $w$, where $w(l(w) + 1) \equiv \varepsilon$.

- Given the transition probabilities $P_{xyz}$ learned from a document corpus, the likelihood $L(w)$ of any given password $w$ can be assessed,

$$L(w) = \prod_{k=1}^{l(w)-1} P_{w(k)w(k+1), w(k+1)w(k+2)}$$

- From the given corpus of passwords, we can compute the mean and variance of $L(w)$ for passwords of the same length $l = l(w)$: $\mu(l), \sigma^2(l)$, respectively.

- A newly suggested password $\tilde{w}$ could be rejected if, e.g., $L(\tilde{w}) \geq \mu(l(\tilde{w})) - 2\sigma(l(\tilde{w}))$ ($> 0$ depending on the password corpus), i.e., if its likelihood is within two standard deviations of the mean of known passwords of the same length.

- Additionally, a minimum length for new passwords is typically required.


Web-page ranking via discrete-time Markov chain

- Web search results are prioritized, e.g., pages can be listed in order of the number of other pages which link to them as in Google’s PageRank, i.e., a measure of the “popularity” of the page.

- Such measures of popularity are important for setting the price of advertising on commercial web sites.

- A simple iterative procedure for determining the relative popularity of web pages is as follows.
Inferring relative popularity through page links

- For a population of \( N \) pages numerically indexed 1, 2, ..., \( N \), let \( d_i \) be the number of different pages which are linked-to by page \( i \), i.e., \( i \)'s out-degree.

- Define the \( N \times N \) stochastic matrix \( P \) with entries \( P_{i,j} = \frac{1}{d_i} \) if \( i \) links to \( j \), otherwise \( P_{i,j} = 0 \) (with \( P_{i,i} = 0 \) for all \( i \), i.e., a "pure jump" chain).

- Define the popularity/rank of \( \pi_i \geq 0 \) of page \( i \) so that:

\[
\forall i, \quad \pi_i = \sum_j \pi_j P_{j,i} \quad \text{and} \quad 1 = \sum_j \pi_j.
\]

- Note how the \( j \) contributes to \( i \)'s popularity, but that contribution is reduced through division by the total out-degree \( d_j \) of \( j \).

- **Exercise:** Relate \( \pi \) to the "eigenvector centrality" of the web-page graph.

Inferring relative popularity through page links (cont)

- In matrix form, the first set of equations is simply \( \pi^T = \pi^T P \).

- So, \( \pi \) is the invariant distribution of a discrete-time Markov chain on the web pages with transition probabilities \( P \).

- *i.e.*, a random walk on the graph formed by the \( N \) web pages as vertices and the links between them as directed edges, with time corresponding to the number of transitions to other web pages (clicked-on web links).
The stationary distribution as page ranks

• The marginal distribution of the Markov chain at time $k$, $\pi(k)$ satisfies the Kolmogorov equations

$$ (\pi(k))^T = (\pi(k - 1))^T P, $$

• i.e., $\pi_i(k)$ is the probability that the random walk is at page $i$ at time $k$.

• If $P$ is aperiodic and irreducible then there is a unique stationary/invariant distribution $\pi$ such that that $\lim_{k \to \infty} \pi(k) = \pi$.

Google’s PageRank

• Google’s PageRank considers a parameter that models how web surfers do not always select links from web pages but may select links from among their bookmarks.

• Suppose that a bookmark selection occurs with probability $b$ and that the probability of specific bookmarked page selected is $b/N$.

• To this end, instead of $P$, an alternative is to use the stochastic matrix

$$ \tilde{P} := (1 - b)P + (b/N)1, $$

where $1$ is the $N \times N$ matrix all of whose entries are 1.

• With $0 < b \leq 1$, $\tilde{P}$ will be irreducible and aperiodic irrespective of $P$. 
Google’s PageRank (cont)

• But since scalable computation of $\pi$ may rely on sparseness of non-zero entries in $P$, we can retain $P$ and simply adjust the rank of page $i$ to be given by $(1 - b)\pi_i + b/N$.

• More precisely, we adjust the iteration to the affine

$$(\tilde{\pi}(k))^T = (1 - b)(\pi(k - 1))^T P + (b/N)\mathbf{1}^T,$$

where $\mathbf{1}$ is a column vector of 1s.

• This leads to a unique stationary distribution.

$$\tilde{\pi}^T = (b/N)\mathbf{1}^T [I - (1 - b)P]^{-1},$$

where $I$ is the $N \times N$ identity matrix and $I - (1 - b)P$ is non-singular for $0 < b \leq 1$ because $P$ is a stochastic matrix.

• Typically, most of the entries of $I - (1 - b)P$ are zero, so computationally efficient methods for inverting sparse matrices can be applied.

Caching Systems - Outline

• Independent Reference Model (IRM)
• LRU eviction
• MRU and kRU eviction
• RE eviction (RANDOM), FIFO, IRP (CLIMB)
• Some numerical results
• A RE caching network
• Approximating the hitting probabilities of a LRU cache
• Simulation exercises
Markov model of Least Recently Used (LRU) eviction policy

- The stationary state-space $\mathcal{R}$ of a LRU cache is the set of $B$-permutations of $\{1, 2, ..., N\}$ where
  - $N$ is the number of objects that could be cached and
  - $B$ objects is the capacity of the cache with $N > B > 0$ (typically $N \gg B$) and
  - the objects assumed identically sized.
- For $r \in \mathcal{R}$, define $r(k)$ as the element of $r$ in the $k^{th}$ position.
- The entries of $r$ are ranked in order of their position in $r$:
  - the most recently accessed (LRU) object being $r(1)$,
  - the oldest object in the cache being $r(B)$, and
  - uncached objects $n$ are denoted $n \not\in r$.
- In a transient regime, the cache may be in a state $\not\in \mathcal{R}$ with fewer than $B$ objects cached.

LRU eviction (cont)

- For a single node, we assume that demand process for object $n \in \{1, 2, ..., N\}$ is Poisson with intensity $\lambda_n$.
- These Poisson demands are assumed independent.
- Let the total demand intensity be
  \[ \Lambda = \sum_{n=1}^{N} \lambda_n. \]
- So, this is the classical “Independent Reference Model” (IRM) with query probabilities $p_n = \lambda_n/\Lambda$. 

December 14, 2018 George Kesidis
283

December 14, 2018 George Kesidis
284
Cache misses under LRU and cache hits, IRM model

- For LRU, a cache miss of object \( r(1) \) at state \( M_n^{-1}(r) \) resulting in a transition to state \( r \in \mathcal{R} \) occurs at rate \( \lambda_{r(1)} \), where \( n \notin r \) and

\[
(M_n^{-1}(r))(k) = \begin{cases} 
  n & \text{if } k = B \\
  r(k + 1) & \text{if } k < B 
\end{cases}
\]
i.e., \( n \notin r \) is the oldest object in the cache in state \( M_n^{-1}(r) \).

- A cache hit of object \( r(1) \) at state \( H_k^{-1}(r) \) resulting in a transition to state \( r \) occurs at rate \( \lambda_{r(1)} \) where \( 1 \leq k \leq B \) and

\[
(H_k^{-1}(r))(\ell) = \begin{cases} 
  r(1) & \text{if } \ell = k \\
  r(\ell + 1) & \text{if } \ell < k \\
  r(\ell) & \text{if } k < \ell \leq B 
\end{cases}
\]
i.e., \( r(1) \) is the \( k \)th youngest object in the cache in state \( H_k^{-1}(r) \) and \( H_1^{-1}(r) = r \).

- As commonly assumed with the IRM, we also assume that

- cache misses cause the query to be forwarded, possibly to a server holding the requested object, and once resolved, the object is reverse-path forwarded so that caches that missed it can be updated; and

- the required time for this query resolution process is negligible compared to the inter-querying times of the caching network.

LRU stationary invariant distribution

- The unique invariant distribution of the LRU Markov chain is [W.F. King’71]:

\[
\pi(r) = \prod_{k=1}^{B} \frac{\lambda_{r(k)}}{\Lambda - \sum_{i=1}^{k-1} \lambda_{r(i)}}
\]
for \( r \in \mathcal{R} \), where \( \forall k, \sum_{i=k}^{k-1} \ast \equiv 0 \).

- **Proof**: The full balance equations are: \( \forall r \in \mathcal{R}, \)

\[
(\Lambda - \lambda_{r(1)})\pi(r) = \sum_{n \not\in r} \lambda_{r(1)}\pi(M_n^{-1}(r)) + \sum_{j=2}^{B} \lambda_{r(1)}\pi(H_j^{-1}(r)).
\]

- Under the claimed invariant, for all \( n \not\in r \),

\[
\pi(M_n^{-1}(r)) = \frac{\lambda_n}{\Lambda - \sum_{i=2}^{B} \lambda_{r(i)}} \prod_{k=2}^{B} \frac{\lambda_{r(k)}}{\Lambda - \sum_{i=2}^{k-1} \lambda_{r(i)}}
\]
and for all \( j \in \{2, 3, ..., B\} \),

\[
\pi(H_j^{-1}(r)) = \prod_{k=2}^{j} \frac{\lambda_{r(k)}}{\Lambda - \sum_{i=2}^{k-1} \lambda_{r(i)}} \cdot \frac{\lambda_{r(1)}}{\Lambda - \sum_{i=2}^{k-1} \lambda_{r(i)}} \cdot \prod_{k=j+1}^{B} \frac{\lambda_{r(k)}}{\Lambda - \sum_{i=1}^{k-1} \lambda_{r(i)}}
\]
LRU invariant under IRM - proof (cont)

- Substituting into the full balance equations, and after some term cancellation, we see that
the claimed invariant satisfies the full balance equation if and only if

\[
1 = \prod_{k=3}^{B+1} \frac{x - \sum_{i=1}^{k-1} \lambda_r(i)}{x - \sum_{i=2}^{k} \lambda_r(i)} + \sum_{j=2}^{B} \prod_{k=3}^{j} \frac{x - \sum_{i=1}^{k-1} \lambda_r(i)}{x - \sum_{i=2}^{k} \lambda_r(i)} \cdot \frac{\lambda_r(1)}{x - \sum_{i=2}^{j} \lambda_r(i)}.
\]

- To see why this is true, suppose we’re given initially that the first cache entry is \( r(2) \).

- Now sequentially, according to the claimed invariant distribution, object \( r(1) \) attempts to
enter the cache after \( r(2) \).

- If it fails to enter in the \( k \)th attempt, then \( r(k+2) \) is placed in the cache instead and
\( r(1) \) tries again.

- The summand with \( j = 2 \) is the probability that \( r(1) \) enters in the second position right
after \( r(2) \): \( \lambda_r(1)/(x - \lambda_r(2)) \).

- Generally, the summand for \( j \in \{2, 3, ..., B\} \) is the probability \( r(1) \) enters in the \( j \)th
position (after having failed to enter in one of the more highly ranked ones).

- The first term of the right-hand-side is the probability \( r(1) \) fails to enter the cache.

- So, the above equation must generally hold by the law of total probability.

LRU invariant - comments

- Finally, since the stationary LRU Markov chain is irreducible on \( \mathcal{R} \), there is a unique invari-
ant. Q.E.D.

- Generally, the LRU Markov chain is neither time-reversible (nor quasi-reversible).

- Obviously, more popular objects (larger \( \lambda \)) are more likely stored, and the LRU invariant is
uniform in the special case that all the mean querying rates \( \lambda_n \) are the same.

- By PASTA, the stationary hit probability of object \( n \) in a LRU cache is

\[
h_n = \sum_{r : n \in r} \pi(r).
\]
Considering objects with different lengths

- To account for objects of different lengths for capacity-driven caches (with ranked objects) like LRU,
- simply consider a "complete-rankings" LRU variation, where
- the ranking of all objects is maintained whether the objects are cached or not.
- That is, the state-space $\mathcal{R}$ is now the set of permutations of all $N$ objects.

**Corollary:** The unique stationary invariant $\pi$ of complete-rankings LRU is given by King’s formula with $B$ replaced by $N$.

- Additionally consider the different sizes $\ell_n$ bytes of objects $n$, where the cache capacity $B$ is in bytes. The number of objects in the cache is given by

$$K(r) = \max\{K \mid \sum_{k=1}^{K} \ell_{r(k)} \leq B, \ 1 \leq K \leq N\}.$$ 

- So, the hit probability of object $n$ when the objects are of variable length is

$$h_n = \sum_{r : r(n) \leq K(r)} \pi(r).$$

Most Recently Used (MRU) and $k$RU eviction

- Again define the state-space $\mathcal{R}$ as the set of $B$-permutations of $\{1, 2, \ldots, N\}$.

- Under MRU, a cache hit of object $r(1)$ at state $H_{k}^{-1}(r)$ resulting in a transition to state $r$ occurs at rate $\lambda_{r(1)}$ where $1 \leq k \leq B$ and $(H_{k}^{-1}(r))(\ell)$ is given as LRU.

- But for MRU, a cache miss of object $r(1)$ at state $M_{n}^{-1}(r)$ resulting in a transition to state $r \in \mathcal{R}$ occurs at rate $\lambda_{r(1)}$, where $n \notin r$ and

$$\left(M_{n}^{-1}(r)\right)(k) = \begin{cases} n & \text{if } k = 1 \\ r(k) & \text{if } k > 1 \end{cases}$$

i.e., $n \notin r$ is the youngest object in the cache in state $M_{n}^{-1}(r)$.
MRU eviction

- **Theorem:** The unique invariant distribution of the MRU Markov chain is, for \( r \in \mathcal{R} \),

\[
\pi(r) = \frac{\lambda_r(1)}{\Lambda} \cdot \frac{1}{\binom{N-1}{B-1}} \prod_{k=2}^{B-1} \frac{\lambda_r(k)}{\Lambda} - \sum_{i=1}^{k-1} \lambda_r(i) - \sum_{n \notin r} \lambda_n.
\]

- **Exercise:** Give a proof by adapting that given above for LRU.

- To interpret MRU’s invariant:
  - \( \lambda_r(1) \) is chosen with probability \( \lambda_r(1)/\Lambda \);
  - then the remaining \( B - 1 \) objects in \( r \) are chosen from the remaining \( N - 1 \) objects uniformly at random with probability \( \binom{N-1}{B-1}^{-1} \);
  - finally, the order of the remaining items \( \lambda_r(2), \lambda_r(3), \ldots \) are determined as the LRU invariant distribution.

Object hit probabilities under MRU

- Consider a MRU cache under the IRM that is “synchronized” so that a query for object \( n \) occurs at time 0.

- Thus, immediately thereafter, \( n \) is the MRU object in the cache.

- The next query for object \( n \) will be at time \( T_n \sim \exp(\lambda_n) \).

- Again, under MRU eviction, the only way an object \( n \) is evicted is when a cache miss occurs immediately after a query for \( n \), i.e., a cache miss when \( n \) is the MRU object. So, the stationary hit probability \( h_n \) of object \( n \) equals the probability that a hit occurs at time \( T_n \), which is
  - the probability that no other queries occurred in the interval \((0, T_n)\) plus
  - the probability that a query does occur in \((0, T_n)\) and the first such query is a hit.

- Thus, we can write \( \forall n, \)

\[
h_n = E \left( e^{-T_n \sum_{i \neq n} \lambda_i} \left( 1 - e^{-T_n \sum_{i \neq n} \lambda_i} \right) \sum_{j \neq i \neq n} \frac{\lambda_j h_{jn}}{\sum_{i \neq n} \lambda_i} \right),
\]

where \( h_{jn} \) is the probability that a query is a hit on \( j \) given that object \( n \) is MRU.

- Note that by marginalizing the MRU invariant, we get that the probability that object \( n \) is MRU is \( p_n = \lambda_n/\Lambda \).
Object hit probabilities under MRU (cont)

- So, we have shown the following.

- **Proposition:** For a MRU-eviction cache under the stationary IRM: \( \forall n, \ h_n = p_n + \sum_{j \neq n} p_j h_{jn} = \sum_j p_j h_{jn} \), where \( p_j = \lambda_j / \sum_i \lambda_i \) and \( h_{jj} = 1 \); equivalently, a kind of balance equation: \( \forall n, \sum_j p_j h_{nj} = \sum_j p_j h_{jn} \).

\( kRU \) eviction under IRM

- "\( k \)th Recently Used" (\( kRU \)) is a simple generalization of LRU and MRU wherein object \( r(k) \), for some fixed \( k \in \{1, 2, \ldots, B\} \), is evicted upon cache miss;

- otherwise cache insertion (at rank 1) upon misses and promotion (to rank 1) and demotions (by 1) upon hits are the same as both MRU and LRU.

- That is, \( BRU \) is LRU and \( 1RU \) is MRU.

- **Corollary:** The invariant distribution of \( kRU \) is

\[
\pi(r) = \prod_{j=1}^{k} \frac{\lambda_{r(j)}}{\lambda - \sum_{i=2}^{j} \lambda_{r(i)}} \times \frac{1}{(B-k) \prod_{j=k+1}^{B-1} \Lambda \lambda - \sum_{i=1}^{j-1} \lambda_{r(i)}} \prod_{j=k+1}^{B-1} \Lambda \lambda - \sum_{n \notin r} \lambda_n.
\]

- **Exercise:** Give a proof by adapting that given above for LRU & MRU.

- The \( kRU \) problem is similar to the permutation-valued Markov chains ranking all objects, not just those cached [Hendricks’76].
Incremental Rank Progress (IRP, CLIMB)

- Under IRP, a query for object \( n \) results in its rank improved by just one (or zero if the object is already ranked first) and
- missed objects enter the cache at lowest rank.

**Theorem:** IRP under IRM is time-reversible with unique stationary invariant

\[
\pi(r) = \frac{\prod_{k=1}^{B} \lambda_{r(k)}^{B+1-k} \prod_{r' \in \mathcal{R}} \prod_{n \in r'} \lambda_{n}}{\sum_{r' \in \mathcal{R}} \prod_{k=1}^{B} \lambda_{r'(k)}}.
\]

**Exercise:** Prove by detailed balance.

Random Eviction (RE, RANDOM)

- A cache miss for object \( n \) results in \( n \) inserted into the cache and evicting of an object \( \ell \) selected uniformly at random from the cache.
- Suppose that a cache miss of object \( n \) at state \( M_{\ell,n}^{-1}(r) \) results in a transition to state \( r \in \mathcal{R} \) at rate \( B^{-1} \lambda_n \), where \( n \in r \), \( n \notin M_{\ell,n}^{-1}(r) \), \( \ell \in M_{\ell,n}^{-1}(r) \), and \( \ell \notin r \).
- The cache state \( r \) does not change if a cache hit occurs.
- The stationary state-space \( \mathcal{R} \) is the set of \( B \)-combinations of \( N \) different objects.

**Theorem:** The RE Markov chain is time-reversible with unique stationary invariant distribution

\[
\pi(r) = \frac{\prod_{n \in r} \lambda_n \prod_{n \notin r} \lambda_n}{\sum_{r' \in \mathcal{R}} \prod_{n \in r'} \lambda_n}.
\]

**Exercise:** Prove by detailed balance.
FIFO caching

• Under FIFO eviction, the object that has been in the cache longest is evicted.
• So, as for LRU, a cache miss of object $r(1)$ at state $M^{-1}(n)$ results in a transition to state $r \in R$ occurs at rate $\lambda_{r(1)}$, where $n \notin r$ and

$$M^{-1}(r)(k) = \begin{cases} n & \text{if } k = B \\ r(k+1) & \text{if } k < B \end{cases}$$

i.e., $n \notin r$ is the oldest (ranked $B$) object in the cache in state $M^{-1}(r)$.

• The state of the cache does not change upon cache hit.
• Exercise: Show that the full balance equations are: $\forall$ $B$-permutations $r$,

$$\left( \sum_{n \notin r} \lambda_n \right) \pi(r) = \lambda_{r(1)} \sum_{n \notin r} \pi(M^{-1}(r)),$$

so that the invariant is as that of RE,

$$\pi(r) = \prod_{k \in r} \lambda_k \bigg/ \sum_{r'} \prod_{k \in r'} \lambda_k.$$ 

• Note that, unlike RE, FIFO is not time-reversible and is permutation valued.
• The stationary aggregate hit rate of FIFO is less than LRU [Chrobak & Noga ’99].

Exactly computed hit probabilities for small $N, B$

$k$RU cache hit probabilities $h_{n_k}$ and popularity $\lambda_n$ versus object index $n$ for a cache of size $B = 6$, $N = 12$ objects, and Zipf popularity distribution with exponent $\alpha = 0.75$,

• where LRU=6RU and MRU=1RU.
Exactly computed hit probabilities for small \( N, B \) (cont)

- Cache hit probabilities \( h \) versus popularity \( \lambda \) for a cache of size \( B = 3 \), \( N = 12 \) objects, and
- Zipf popularity with exponent \( \alpha = 0.75 \).

---

**MRU with lower-bounded inter-query times**

- MRU is used when demand for hot (most popular) objects is such that they are not likely to be needed again soon after they are queried for.
- Consider the case where inter-query times are lower bounded by a constant \( D \) – specifically, inter-query times equal \( D \) plus an exponentially distributed quantity, so \( D = 0 \) corresponds to the IRM.
- The following table \( kRU \) aggregate hit rate for \( N = 12 \) objects, cache of capacity \( B = 6 \) objects, and Zipf popularity distribution with exponent \( \alpha = 0.75 \).

<table>
<thead>
<tr>
<th>( k ) (MRU)</th>
<th>( D = 0 )</th>
<th>( D = 1 )</th>
<th>( D = 2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.52</td>
<td>0.4578</td>
<td>0.45</td>
</tr>
<tr>
<td>2</td>
<td>0.54</td>
<td>0.4213</td>
<td>0.40</td>
</tr>
<tr>
<td>3</td>
<td>0.56</td>
<td>0.4014</td>
<td>0.35</td>
</tr>
<tr>
<td>4</td>
<td>0.58</td>
<td>0.4026</td>
<td>0.31</td>
</tr>
<tr>
<td>5</td>
<td>0.60</td>
<td>0.4187</td>
<td>0.29</td>
</tr>
<tr>
<td>6 (LRU)</td>
<td>0.62</td>
<td>0.4423</td>
<td>0.29</td>
</tr>
</tbody>
</table>
A network of RE caches

• To illustrate the difficulties with capacity-driven caching networks, now consider the simplest ones based on RE (combination-valued cache state).

• Though RE caches are time-reversible, a tree of independent local caches $q$ (of size $B_q$) whose collective query-misses are forwarded to an Internet cache (size $b$) is not time-reversible and its non-local nodes do not operate under the IRM.

• To see why it’s not time-reversible, consider a cache miss of object $n$ of local cache $q$ in state $r_q$, so that object $n_q$ is evicted, and suppose it’s also a miss on the Internet cache in state $R$, so that object $n$ is evicted;

• this can be reversed with one query (so that states $r_q$ and $R$ are restored) only if $n = n_q$.

• Note that, in practice, plural Internet caches could forward queries (with TTL) among themselves before forwarding to the origin server(s).

A network of RE caches

• Proposition: The invariant distribution $\pi$ of the RE caching tree with Internet cache size $b = 1$ satisfies

$$
\pi(R|r) = \frac{\sum_q 1\{R \in r_q\} \Lambda_q \pi_r / B_q}{\sum_q \Lambda_q \pi_r}
$$

where

$$
\Lambda_q \pi_r = \sum_{\ell \notin x} \lambda_q \ell,
$$

$\sum_q (...) \equiv 0$, and indicator $1X = 1$ if $X$ is true otherwise $= 0$.

• The proof is by full balance equations.

• In steady state, $R \subset \cup_q r_q$ a.s., i.e., if $\forall q, R \notin r_q$ then $\pi(R|r) = 0$.

• Note that the above invariant is the eviction probability of object $R$ upon local cache miss in local cache state $r$.

• Since the capacity of the Internet cache is one object ($b = 1$), it could obviously be operating any eviction policy.
A network of RE caches (cont)

- One can identify the incident mean rate of queries for object \( n \) to the Internet cache,

\[
\hat{\lambda}_n := \sum_q \lambda_{q,n} (1 - h_{q,n}) = \sum_q \lambda_{q,n} \sum_{r \neq n \in r_q} \pi(r_q),
\]

where \( 1 - h_{q,n} \) is the stationary miss probability of local cache \( q \) for object \( n \) under RE.

- According to this proposition, \( \pi(R) \) does not depend on the \( \hat{\lambda}_n \) in the way the IRM invariant \( \pi(r_q) \) depends on the \( \lambda_{q,n} \), i.e.,

\[
\pi(R) = \sum_q \pi(R|r) \pi(r) = \sum_q \pi(R|r) \prod_q \pi(r_q) \neq \frac{\hat{\lambda}_R}{\sum_n \hat{\lambda}_n}.
\]

- So, the RE network is not product form.

LRU caching - approximate stationary hitting probabilities

- For large caches, the stationary hit probabilities involve significant computational complexity.

- Assume IRM and all data objects have the same size.

- In the framework of Palm’s theorem and PASTA, suppose a particular data-object \( n \) is requested at time zero.

- Let \( h_n(B) \) be the probability that the next request for object \( n \) is a cache hit.

- By the memoryless property of exponential random variables,

\[
h_n(B) = P(X_n(\tau_n) < B),
\]

where \( X_n(t) \) is the number of different data-objects requested in time-interval \((0, t)\) except object \( n \): for \( t > 0 \),

\[
X_n(t) := \sum_{i=1,i \neq n}^N 1\{\tau_i < t\},
\]

and mutually independent \( \tau_i \sim \exp(\lambda_i) \).

- Note that \( X_n(t) \) is a.s. non-decreasing in \( t \) and does not depend on \( \tau_n \).
LRU caching - hit probability & eviction times (cont)

• So, we can write the cache hit probability for object \( n \),
  \[ h_n(B) = P(X_n(\tau_n) < B) = P(\tau_n < T_n(B)) = \mathbb{E}(1 - e^{-\lambda_n T_n(B)}), \]
  where we can interpret
  \[ T_n(B) := X_n^{-1}(B) \]
as the eviction time of object \( n \).

• Note that \( \forall n, \tau_n \) and \( T_n(B) \) are independent.

Since at \( T_n(B) \) there are \( B \) objects in the cache (otherwise object \( n \) is not evicted) and \( X_n(T_n(B)) = B \) a.s. (by definition),
\[
B = \mathbb{E}X_n(T_n(B)) = \sum_{i=1, i \neq n}^{N} \mathbb{E}1\{\tau_i < T_n(B)\} = \sum_{i=1, i \neq n}^{N} P(X_n(\tau_i) < B) \\
= \sum_{i=1, i \neq n}^{N} \mathbb{E}(1 - e^{-\lambda_i T_i(B)}).
\]

• Exercise: Working with independent Booleans, show for (constant) \( t > 0 \),
  \[
  E(X_n(t)) = \sum_{i \neq n} 1 - e^{-\lambda_i t} =: \mu_n(t) \quad \& \quad \text{var}(X_n(t)) = \sum_{i \neq n} e^{-\lambda_i t}(1 - e^{-\lambda_i t}) =: \sigma_n^2(t).
  \]
LRU Caching - Working Set (WS) approximation

- In [Denning & Schwartz’72, Fagin’77, Che et al.’02], [Denning and Schwartz, Commun. ACM, 1972],
- for large $B$ (but still $B \ll N$), the following "Working Set" (WS) approximations were made:
  - $\forall n, \var(T_n(B)) \approx 0$, i.e., $T_n(B) \approx t_n(B)$ (a constant) and $\forall n, t_n(B) = t(B)$, i.e., they are all the same constant, so that
    \[ h_n(B) \approx 1 - e^{-\lambda_n t(B)} \] and
    \[ \forall n, B \approx \sum_{i=1, i \neq n}^N h_i(B). \]
  - $\forall n, \lambda_n \ll \Lambda = \sum_i \lambda_i$, so that
    \[ B \approx \sum_{i=1}^N h_i(B) \] and $t(B)$ is the unique (why?) solution to the deterministic equation,
    \[ B = \sum_{i=1}^N (1 - e^{-\lambda_i t(B)}). \]

LRU - Justifying the WS approx. [Fricker et al., Proc. ITC 2012]

- Let the standard normal CDF be $F(t) := \int_{-\infty}^t e^{-x^2/2} dx / \sqrt{2\pi}$, $t \in \mathbb{R}$.
- By the central limit theorem (CLT), for large $N \gg 1$ and constant $t$, $X_n(t) \sim N(\mu_n(t), \sigma_n^2(t))$ approximately. So,
  \[
  h_n(B) = P(\tau_n \leq T_n(B)) = \int_0^\infty P(u \leq T_n(B))\lambda_n e^{-\lambda_n u} du \\
  = \int_0^\infty P(X_n(u) \leq B)\lambda_n e^{-\lambda_n u} du \\
  \approx \int_0^\infty P(\tilde{\mu}(u) \leq B)\lambda_n e^{-\lambda_n u} du \\
  = \int_0^\infty \{u \leq t(B)\} \lambda_n e^{-\lambda_n u} du \\
  = 1 - e^{\lambda_n t(B)}, \quad \text{where:}
  \]
- $\tilde{\mu}(u) := \sum_{i=1}^N 1 - e^{-\lambda_i u} \approx \mu_n(u)$ $\forall n$, $t(B) := \tilde{\mu}^{-1}(B)$,
- the first (CLT) approximation requires uniform accuracy in $u$, and
- the second (step) approximation is discussed below.
- The following figure roughly illustrates the step approximation,  
\( \forall n, \ P(\frac{B-\mu(u)}{\sigma(u)}) \approx 1\{\bar{\mu}(u) \leq B\} =: 1\{u \leq t(B)\} \).

- The WS approximation is shown to be asymptotically accurate in [R.Fagin’77].

---

**Caching - Discussion**

- A commonly used model for popularity is the Zipf law,  
  \( \lambda_n \propto \rho(n)^{-\alpha} \), where \( \alpha > 0 \)  
  and \( \rho(n) \) is the popularity rank of object \( n \), i.e., \( \rho(n') = 1 \) if \( n' = \text{argmax}_n \lambda_n \) is unique  
  and \( \rho(n'') = N \) if \( n'' = \text{argmin}_n \lambda_n \) is unique.

- If object \( n \) has size \( s_n \) bytes and \( B \) is also measured in bytes, then \( t(B) \) now solves,  
  \[ B = \sum_{n=1}^{N} (1-e^{\lambda_n t(B)}) s_n = \sum_{n=1}^{N} h_n(B) s_n, \]  
  for large \( B \ll \sum_{n=1}^{N} s_n \).
Exercise: Simulation

- Show your code and results.
- Write an event-driven simulator of an LRU cache under IRM capable that numerically estimates the hit probabilities of the $N$ different objects and associated confidence interval.
- Write a time-driven simulator of an LRU cache where the object inter-query times are each a positive constant plus an exponential.
- Modify your simulators to find the performance of MRU both in terms stationary hit probabilities for individual objects and the overall (across all objects) mean hit probability.

Review of Statistical Confidence

- The central limit theorem
- Statistical confidence
- See slidedeck at http://www.cse.psu.edu/~kesidis/teach/Prob-4.pdf
Simulation - Discussion

- Motivation: to explore beyond what currently can be proved or numerically computed from (tractable) models, and involve data/parameters and mechanisms of scenarios more representative of the "real world"
- Event-driven or time-driven simulation
- Random number generation
- Assessing performance metrics with confidence
- Markov-chain Monte Carlo (MCMC)
- Parallel and distributed simulation
  - load balancing (proactive and reactive methods)
  - synchronization and rollback
  - dynamic time-warping
- Quick simulation by
  - modeling-based techniques, e.g., state aggregation, fluid modeling
  - statistical techniques, particularly importance sampling

Simulating a sample path of a discrete-time \((n)\) Markov chain \(x\)

\[
\begin{align*}
n &= 0 \\
u &= \text{rand}() \\
x(0) &= F^{-1}(\text{init}, u) \\
\textbf{while } n < \text{max\_simulation\_time} \textbf{ do} \\
& \quad n \leftarrow n + 1 \\
& \quad u = \text{rand}() \\
& \quad x(n) = F^{-1}(x(n - 1), u) \\
\textbf{end while}
\end{align*}
\]

where

- the \text{rand} function returns IID (continuously) uniform\([0,1]\) samples,
- \(F^{-1}(\text{init}, \cdot)\) is the inverse CDF of the initial distribution, and
- \(F^{-1}(x, \cdot)\) is the inverse CDF of PMF that’s the \(x^{th}\) row of TPM \(P\),
- \(e.g.,\) for a uniform initial on state-space \(\{0, 1, 2\}\): if \(u < 1/3\) then \(F^{-1}(\text{init}, u) = 0\), else if \(u < 2/3\) then \(F^{-1}(\text{init}, u) = 1\), else \(F^{-1}(\text{init}, u) = 2\).
Simulating a sample path of a continuous-time $(t)$ Markov chain $x$

\begin{verbatim}
\texttt{n = 0}
\texttt{u = rand()}
\texttt{x(0) = F^{-1}(\text{init}, u)}
\texttt{t(0) = 0}
\texttt{while t(n) < max\_simulation\_time do}
  \texttt{u = rand()}
  \texttt{t(n) = t(n - 1) + log(1 - u)/Q(x(n), x(n))}
  \texttt{u = rand()}
  \texttt{x(n) = F^{-1}(x(n - 1), u)}
\texttt{end while}
\end{verbatim}

- where $t(n)$ is the $n$th jump/transition time, and
- $F^{-1}(x, n)$ is the CDF of the $x$th row of the jump chain with TPM: $\forall i$, $P_{,i} = 0$; and $\forall j \neq i$, $P_{i,j} = -Q_{i,j}/Q_{i,i}$.

Continuous-time Markov chain simulation by uniformization

- For any $q > \max_j -Q_{i,j}$, instead of the jump chain above, use the (non-jump) TPM $P = I + Q/\gamma$.
- $\forall n > 0$, $t(n) - t(n - 1)$ are IID $\exp(\gamma)$ random variables, i.e.,
  \[ t(n) = t(n - 1) + \log(1 - u)/\gamma. \]
- So, the number of iterations of the while loop over an interval of (continuous) time $[0, t]$ will be $\sim$ Poisson($\gamma t$).
- It follows that the TPM in continuous time,
  \[ \exp(Qt) = \sum_{n=0}^{\infty} P^n (\gamma t)^n / n! e^{-\gamma t}. \]
- \textbf{Exercise}: Verify this by using the definition of $P$.
- There is an alternative approach called perfect simulation.
Fork-join model of parallel computation - outline

- Motivation - MapReduce
- A single-stage, fork-join system
- A deterministic analysis
- A stationary analysis
- A two-server Markovian system - two M/M/1 queues with coupled arrivals
- Multi-server system
- Martingale approach

Parallel processing systems

- Decades of study on concurrent programming and parallel processing (including cluster computing), often in highly application-specific settings.

  Challenges include
  - resource allocation and load balancing so as to reduce delays at synchronization/barrier points,
  - dynamically deeming and dealing with straggler tasks,
  - redundancy for robustness/protection, and
  - maintaining consistent shared memory/state across processors while minimizing communication overhead,

- especially when dealing with feedback in the application itself.

- Today, popular platforms involve a group of Virtual Machines (VMs) mounted on multi-core/processor servers of a data center, or a group of data-centers forming a cloud.
Feed-forward parallel processing systems

- A certain family of jobs are best served by a particular arrangement of VMs/processors for parallel execution,

- In the following, we consider jobs that lend themselves to feed-forward parallel processing systems, e.g., many search/data-mining applications.

- Google’s MapReduce template for parallel processing with VMs (especially its open-source implementation Apache Hadoop) is a very popular such framework for search.

- In a single parallel processing stage, a job is partitioned into tasks (i.e., the job is “forked” or the tasks are demultiplexed); the tasks are then worked upon in parallel by different processors.

- Within parallel processing systems, there are often processing barriers (points of synchronization or “joins”) wherein all component tasks of a job need to be completed before the next stage of processing of the job can commence.

- The terminus of the entire parallel processing system is typically a barrier.

- Thus, the latency of a stage (between barriers or between the exogenous job arrivals to the first barrier) is the greatest latency among the processing paths through it.

MapReduce

- MapReduce is a multi-stage parallel-processing framework where each processor is a Virtual Machine (VM) mounted on a server (multiprocessor computer).

- In MapReduce, jobs arrive and are partitioned into tasks.

- Each task is then assigned to a mapper VM for initial processing (first stage).

- The results of mappers are transmitted (shuffled), in pipelined fashion with the mapper’s operation, to reducer stage.

- Reducer VMs combine the mapper results they have received and perform additional processing.

- A barrier exists before each reducer (after its mapper-shuffler stage) and after all the reducers (after the reducer stage).
Simple MapReduce example of a word-search application

- Two mappers that search and one reducer that combines their results.
- Document corpus to be searched is divided between the mappers.

Single-stage, fork-join systems - a deterministic analysis

- Consider a bank of $K$ parallel queues, with queue/processor $k$ is provisioned with service capacity $s_k$.

- Here let $A$ be the (fluid, positive time) cumulative input process of work that is divided among queues so that the $k^{th}$ queue has arrivals $a_k$ and departures $d_k$ in such a way that $\forall t \geq 0$,

$$A(t) = \sum_k a_k(t).$$

- Define the virtual delay processes for hypothetical departures at time $t \geq 0$ for queue $k$ as

$$\delta_k(t) = t - a_k^{-1}(d_k(t)),$$

where we define inverses $a_k^{-1}$ of non-decreasing functions $a_k$ as continuous from the left so that $a_k(a_k^{-1}(v)) \equiv a_k^{-1}(a_k(v)) \equiv v$.

- The following definition of the cumulative departures $D$ is such that the output ready for processing in the subsequent (reducer) stage is determined by the most “lagging” queue/processor: $\forall t \geq 0$,

$$D(t) = A(t - \max_k \delta_k(t))$$

$$= A \left( \min_k a_k^{-1}(d_k(t)) \right)$$
Delay bound under service and input-burstiness curves

- Assume the $k^{\text{th}}$ queue has service at least $s_{\min,k}$ and arrivals $a_k \ll b_{\text{in},k}$, i.e., conform to burstiness curve (traffic envelope) $b_{\text{in},k}$.

- Recall the convolution$(\otimes)$/deconvolution$(\ominus)$ identity is
  \[
  u_{\infty}(t) = \begin{cases} 
  0 & \text{if } t \leq 0 \\
  +\infty & \text{if } t > 0 
  \end{cases}
  \]

- The largest horizontal difference between $b_{\text{in},k}$ and $s_{\min,k}$ is
  \[
  d_{\max,k} = \min \{ z \geq 0 : \forall x \geq 0, s_{\min,k}(x) \geq (b_{\text{in},k} \otimes \Delta_z u_{\infty})(x) = b_{\text{in},k}(t - z) \}
  \]
  where the delay operator $(\Delta_d g)(t) \equiv g(t - d)$.

Simple deterministic delay-bound claim

- **Claim:** If $s_{\min,k}$ is a lower service curve of queue $k$ and $b_{\text{in},k}$ is a traffic envelop of arrivals $a_k$, then for all $t \geq 0$,
  \[
  D(t) \geq A(t - \max_k d_{\max,k}).
  \]

- Note that this claim simply states that the maximum delay of the system is the maximum delay among the queues.

- Equivalently, the service from $A$ to $D$ is at least $\Delta_d u_{\infty}$, where $d := \max_k d_{\max,k}$.
Proof of deterministic delay-bound claim

- By def’n of $d_{\text{max}, k}$, $\forall t \geq x \geq 0$ and $\forall k$,
  \[
  s_{\min, k}(t - x) \geq b_{\text{in}, k}(t - x - d_{\text{max}, k}) \\
  \Rightarrow a_k(x) + s_{\min, k}(t - x) \geq a_k(t - d_{\text{max}, k}) \\
  \Rightarrow (a_k \otimes s_{\min, k})(t) \geq a_k(t - d_{\text{max}, k}) \\
  \Rightarrow a_k^{-1}((a_k \otimes s_{\min, k})(t)) \geq t - d_{\text{max}, k}
  \]

  where we have used the fact that, $\forall k$, $a_k$ are nondecreasing.

- Thus,
  \[
  D(t) = A \left( \min_k a_k^{-1}(d_k(t)) \right) \\
  \geq A \left( \min_k a_k^{-1}((a_k \otimes s_{\min, k})(t)) \right) \\
  \geq A \left( \min_k t - d_{\text{max}, k} \right) \\
  = (A \otimes \Delta_d u_\infty)(t),
  \]

  where we have used the fact that $A$ is nondecreasing. $\square$

Single-stage, fork-join systems - a stationary analysis

- Claim: In the stationary regime at $t \geq 0$, if
  A1 service to queue $k$, $s_k \gg s_{\min, k}$ where
  \[
  \forall v \geq 0, \quad s_{\min, k}(v) := v\mu_k;
  \]
  A2 the demux/mapper divides arriving work roughly proportional to minimum allocated service resources $\mu_k$ to queue $k$ (strong load matching), i.e., $\forall k, \exists$ small $\varepsilon_k > 0$ such that $\forall v \leq t$,
  \[
  \left| a_k(t) - a_k(v) - \frac{\mu_k}{M}(A(t) - A(v)) \right| \leq \varepsilon_k \text{ a.s.,}
  \]
  where $M := \sum_k \mu_k$;
  A3 the total arrivals have generalized (strong) stochastically bounded burstiness,
  \[
  P(\max_{v \leq t} A(t) - A(v) - M(t - v) \geq x) \leq \Phi(x),
  \]
  where $\Phi$ decreases in $x > 0$;
  then $\forall x > 2M \max_k \varepsilon_k / \mu_k$,
  \[
  P(A(t) - D(t) \geq x) \leq \Phi(x - 2M \max_k \varepsilon_k / \mu_k).
  \]
A stationary analysis - proof of claim

\[ P(A(t) - D(t) \geq x) = P(A(t) - A(\min_k a_k^{-1}(d_k(t))) \geq x) \]
\[ = P(\min_k a_k^{-1}(d_k(t)) \leq A^{-1}(A(t) - x) =: t - z) \]
\[ = P(\exists k \text{ s.t. } d_k(t) \leq a_k(t - z)) \]
\[ = P(\exists k \text{ s.t. } a_k(t) - d_k(t) \geq a_k(t) - a_k(t - z) =: x_k) \]
\[ \leq P(\exists k \text{ s.t. } \max_{v \leq t} a_k(t) - a_k(v) - (t - v)\mu_k \geq x_k) \]

- where we have used the fact that \( A \) and the \( a_k \) are nondecreasing (cumulative arrivals) and the inequality is by assumption A1.

- Also, we have defined non-negative random variables \( z \) and \( x_k \) such that

\[ \sum_k x_k = x = A(t) - A(t - z). \]

\[ December 14, 2018 George Kesidis \]

327
So by using A2 (twice) then A3, we get
\[
P(A(t) - D(t) \geq x) \\
\leq P(\exists k \text{ s.t. } \max_{v \leq t} \frac{\mu_k}{M}(A(t) - A(v)) + \varepsilon_k - (t - v)\mu_k \geq \frac{\mu_k}{M}x - \varepsilon_k) \\
= P(\exists k \text{ s.t. } \max_{v \leq t}(A(t) - A(v)) - (t - v)M \geq x - 2\frac{M}{\mu_k}\varepsilon_k) \\
= P(\max_{v \leq t}(A(t) - A(v)) - (t - v)M \geq x - 2M \max_k \varepsilon_k/\mu_k) \\
\leq \Phi(x - 2M \max_k \varepsilon_k/\mu_k). \quad \square
\]

Exercise: numerically computing gSBB $\Phi$

- Compute $\Phi$ for the mapper (first) stage using Figure 3 (job arrival process) and Table 1 (individual job workloads) of


- Compute $\Phi$ for the reducer (second) stage as described in the previous discussion.
Discussion - load matching in a single processing stage

• Typically, the amount of allocated parallelism of a job at a stage is based on the size of the job’s input data-set to that stage, as that information is readily available operationally online.

• The execution time for the component tasks will, of course, greatly depend on other factors such as algorithmic/computational complexity.

• This is evident in a Facebook dataset where two jobs have about the same mean input data size but significantly different mean Map times (one is roughly double the other).

• This said, it’s likely that the same algorithm will be applied for all tasks of a given job at the same stage, so that effective load matching from job to task typically may be achieved.

  • i.e., when \( \forall k, l, \mu_k = \mu_l \).

• Note that the previous Claim allows for processors of different capacities \( \mu \).

• The following corollary involves a weaker form of the load matching assumption (A2).

Load matching in probability

• **Corollary:** If (A1), (A3) and (A2’) For each queue \( k \), there exists \( 0 \leq \varepsilon_k, \delta_k \leq 1 \) such that \( \forall v \leq t \)

  \[
P(\left| a_k(t) - a_k(v) - \frac{\mu_k}{M}(A(t) - A(v)) \right| > \varepsilon_k) < \delta_k,
  \]

  then \( \forall x > 2M \max_k \{\varepsilon_k/\mu_k\} \),

  \[
  P(A(t)-D(t) \geq x) \leq \Phi(x - 2M \max_k \{\varepsilon_k/\mu_k\}) + 2\delta_{\arg\max_k \varepsilon_k/\mu_k}.
  \]

• **Proof:**
  - The corollary is proved by applying the following simple result at where (A2) is used in the proof of the previous Claim.
  - If \( P(|X - Y| \geq \varepsilon) < \delta \) then

    \[
    P(X > \bar{X}) = P(X > \bar{X} | |X - Y| \leq \varepsilon)P(|X - Y| \leq \varepsilon) + P(X > \bar{X} | |X - Y| > \varepsilon)P(|X - Y| > \varepsilon) 
    \leq P(Y + \varepsilon > \bar{X}) + \delta.
    \]

  - Similarly, if also \( P(|\bar{X} - \bar{Y}| \geq \varepsilon) < \delta \) then

    \[
    P(X > \bar{X}) \leq P(Y + \varepsilon > \bar{Y} - \varepsilon) + 2\delta
    \leq P(Y > \bar{Y} - 2\varepsilon) + 2\delta. \quad \square
    \]
Redundant tasking: Releasing job after only $\kappa < K$ tasks complete

- The following extension is useful when tasking involves redundant work or simply when “good enough” solutions are adequate,
- so that a job can be forwarded when only a certain number $\kappa \leq K$ ($\kappa > 0$) tasks complete and the remaining $K - \kappa$ (straggling) tasks are cancelled.
- Its proof follows that of the previous Claim or Corollary with $\min_k$ interpreted as the $(K - \kappa + 1)^{th}$ smallest, and $\max_k$ interpreted as the $(K - \kappa + 1)^{th}$ largest.
- **Corollary**: If a job is completed upon completion of any $\kappa \leq K$ of its $K$ tasks, then the statements of the previous Claims and Corollary continue to hold with
  - $\max_k \{\varepsilon_k / \mu_k\}$ interpreted as the $(K - \kappa + 1)^{th}$ largest $\varepsilon_k / \mu_k$ and
  - $\delta \arg\max_k \varepsilon_k / \mu_k$ replaced by $\max_k \varepsilon_k / \mu_k \geq (K - \kappa + 1)^{th}$ largest $\varepsilon_k / \delta_k$.

---

**Discussion - Tandem parallel-processing stages**

- Let $x$ be the **mean job** arrival rate to a parallel processing stage $w$, and
- let $Z_{w,m}$ be the workload of the $m^{th}$ job, so $x E Z_{w} = \lim_{t \to \infty} A_{w}(t) = E A_{w}(t) = E A_{w}(t) / t$.
- At stage $w$, let $d_{k,w}$ be the amount of IT resource of type $k$ per unit (job) demand required to achieve the necessary service quality.
- Let $M_w := x d_{k^*(w),w}$, where $k^*(w)$ is the “bottleneck” or “dominant” IT resource required to achieve the necessary service quality at stage $w$.
- For stability, it’s required that $x E Z_{w} < M_w$, i.e., $E Z_{w} < d_{k^*(w),w}$, i.e., workloads $Z_{w}$ expressed in terms of bottleneck resource $k^*(w)$.
- Arrivals to the next stage $v$ are departures from the previous $w$ considering propagation delays if significant, $A_{v} = D_{w}$ where $x = E A_{v}(t) / (t Z_{v})$ too.
- Consider a network of parallel-processing stages (incl. re-entrant lines with feedback) handling a plurality of different workloads (job flows) $i$ as the one considered above, where stat mux gains may be exploited when setting aggregate service rate $M_w$.
Single-stage, fork-join systems - a Markovian analysis

- Jobs sequentially arrive to a parallel processing system of $K$ identical servers.
- The $i^{th}$ job arrives at time $t_i$ and spawns (forks) $K$ tasks.
- Let $x_{j,i}$ be the service-duration of the task assigned to server $j$ by job $i$.
- The tasks assigned to a server are queued in FIFO fashion.
- The sojourn (or response) time $D_{j,i} - t_i$ of the $i^{th}$ task of server $j$ is the sum of its service time ($x_{j,i}$) and its queueing delay:
  \[
  D_{j,i} = x_{j,i} + \max\{D_{j,i-1}, t_i\} \quad \forall \ i \geq 1, \ 1 \leq j \leq K
  \]
  \[
  D_{j,0} = 0
  \]
- The response time of the $i^{th}$ job is
  \[
  \max_{1 \leq j \leq K} D_{j,i} - t_i
  \]

Two-server ($K = 2$) system

- Suppose that jobs arrive according to a Poisson process with intensity $\lambda$, i.e.,
  \[
  t_i - t_{i-1} \sim \exp(\lambda) \quad \text{so that} \quad E(t_i - t_{i-1}) = \lambda^{-1}.
  \]
- Also, assume that the task service-times $x_{j,i}$ are mutually independent and exponentially distributed:
  \[
  x_{1,i} \sim \exp(\alpha) \quad \text{and} \quad x_{2,i} \sim \exp(\beta) \quad \forall i \geq 1.
  \]
- Let $Q_i(t)$ be the number of tasks in server $i$ at time $t$.
- $(Q_1, Q_2)$ is a continuous-time Markov chain.
Transition rates of \((Q_1, Q_2)\) with \(m, n \geq 0\)

\[
\begin{align*}
&\ x_{m-1, n} \quad \lambda_1 \{m > 0, n > 0\} \quad \alpha \{m > 0\} \quad \lambda \quad \beta_1 \{n > 0\} \\
&\ m, n \quad \lambda \quad \beta \\
&\ x_{m, n-1} \quad \beta \{n > 0\} \quad \lambda_1 \{m > 0, n > 0\} \quad \alpha \{m > 0\}
\end{align*}
\]

Stationary distribution of \((Q_1, Q_2)\)

- Assume that the system is stable, \(i.e., \lambda < \min\{\alpha, \beta\}\).
- For the Markov process \((Q_1, Q_2)\) in steady state, let the stationary
  \(p_{m,n} = P((Q_1, Q_2) = (m, n))\).
- The balance equations are
  \[
  (1 + \alpha \{m > 0\} + \beta \{n > 0\}) p_{m,n} = \lambda_1 \{m > 0, n > 0\} p_{m-1,n-1} + \alpha p_{m+1,n} + \beta p_{m,n+1}, \quad \forall m, n \in \mathbb{Z}^\geq 0,
  \]
  where
  \[
  \sum_{m=0}^{\infty} \sum_{n=0}^{\infty} p_{m,n} = 1.
  \]
Stationary distribution of \((Q_1, Q_2)\) (cont)

- The balance equations can be solved by two-dimensional moment generating function (Z transform) [Flatto & Hahn 1984]
  \[
P(z, w) = \sum_{m=0}^{\infty} \sum_{n=0}^{\infty} p_{m,n} z^m w^n, \quad z, w \in \mathbb{C}
\]

- Multiplying the previous balance equations by \(z^m w^n\) and summing over \(m, n\) gives \(P(z, w)\) in terms of boundary values \(P(z, 0)\) and \(P(0, w)\).

- In the load-balanced case where \(\alpha = \beta\) with \(\rho := \lambda/\alpha < 1\) [equ (6.5) of FH'84],
  \[
P(z, 0) = (1 - \rho)^{3/2} / \sqrt{1 - \rho z}.
\]

- From this, we can find the first two moments of \(p_{m,0}\),
  \[
  \sum_{m=0}^{\infty} m p_{m,0} = \frac{d}{dz} P(z, 0) \bigg|_{z=1} = \frac{1}{2} \rho \\
  \sum_{m=0}^{\infty} m^2 p_{m,0} = \frac{d}{dz} \frac{d}{dz} P(z, 0) \bigg|_{z=1} = \frac{1}{2} \rho + \frac{3}{4} \cdot \frac{\rho^2}{1 - \rho}
  \]

Job sojourn times

- Recall that a job is completed (departs the system) only when all of its tasks are completed (have been served).

- Some jobs have arrived but none of their tasks completed, while others have had only one task completed.

- So, in the two-server \((K = 2)\) case, \(|Q_1 - Q_2|\) represents the number of jobs queued in the system with just one task completed.

- Let \(q_k := P(Q_1 - Q_2 = k)\) in steady-state for \(k \in \mathbb{Z}\).

- Note that \(\forall k \geq 0,\)
  \[
  q_k = \sum_{m=k}^{\infty} p_{m,m-k}.
  \]
Job sojourn times in the load-balanced case

• Summing the balance equations for \((Q_1, Q_2)\) from \(m = k \geq 0\) with \(n = m - k\) gives
  \[
  (1 + \alpha + \beta)q_k - \beta p_{k,0} = q_k + \alpha q_{k+1} + \beta q_{k-1} - \beta p_{k-1,0}
  \]
  \[
  \Rightarrow \alpha(q_{k+1} - q_k) - \beta(q_k - q_{k-1}) = -\beta p_{k,0} + \beta p_{k-1,0}
  \]

• In the symmetric case (i.e., the servers are load balanced) where \(\alpha = \beta > \lambda\), this implies
  \[
  q_{k+1} - q_k = -p_{k,0}, \quad \forall k \geq 0
  \]
  where \(\forall k \in \mathbb{Z}, q_k = q_{-k}\).

• Thus,
  \[
  q_k = \sum_{m=k}^{\infty} p_{m,0}, \quad \forall k \geq 0.
  \]

Job sojourn times in the load-balanced case (cont)

• Consider jobs with no tasks completed and those completed tasks whose siblings are not completed for the load-balanced \((\alpha = \beta)\) case.

• By Little’s theorem the mean sojourn time of a job is:
  \[
  \frac{EQ_1}{\lambda} + \frac{E|Q_1 - Q_2|}{2\lambda} = \frac{1}{\alpha - \lambda} + \frac{1}{\lambda} \sum_{k=1}^{\infty} kq_k = \frac{1}{\alpha - \lambda} + \frac{1}{\lambda} \sum_{k=1}^{\infty} k \sum_{m=k}^{\infty} p_{m,0}
  \]
  \[
  = \frac{1}{\alpha - \lambda} + \frac{1}{\lambda} \sum_{m=1}^{\infty} p_{m,0} \sum_{k=1}^{m} k = \frac{1}{\alpha - \lambda} + \frac{1}{\lambda} \sum_{m=1}^{\infty} p_{m,0} \frac{m^2 + m}{2}
  \]
  \[
  = \frac{1}{\alpha - \lambda} + \frac{1}{4\lambda} + \frac{3}{8\lambda} \cdot \frac{\rho^2}{1 - \rho} + \frac{1}{4\lambda} \rho
  \]
  where
  \[
  \frac{\alpha - \lambda}{\lambda} = \frac{1 - \rho}{\rho},
  \]
  and we have used the first two moments of \(p_{m,0}\) computed above.
Job sojourn times in the load-balanced case - main result

So, the mean sojourn time of a job in the load-balanced ($\alpha = \beta$) case is:

\[
E \frac{Q_1}{\lambda} + \frac{E|Q_1 - Q_2|}{2\lambda} = \frac{1}{\alpha - \lambda} \left( \frac{3}{2} - \frac{1}{8\rho} \right),
\]

where

\[
\frac{1}{\alpha - \lambda}
\]

is just the mean number of jobs in a stationary M/M/1 queue.

Note that the delay factor above M/M/1 satisfies:

\[
\frac{11}{8} \leq \frac{3}{2} - \frac{1}{8\rho} \leq \frac{3}{2}.
\]

Bounds for $K > 2$ servers - Associated RVs

Again, consider the load balanced (i.i.d. exp($\alpha$) task service times) and stable ($\lambda < \alpha$) case.

To obtain an upper bound, it was argued in [Nelson and Tantawi 1988] that for all jobs $i$, all of its task sojourn times $\{S_{j,i} := D_{j,i} - t_i\}_{j=1}^K$ form an “associated” group of random variables.

Taking any monotonic function $g$ of each member group of an “associated” group of random variables $\{X_j\}$ leads to a group of random variables $\{g(X_j)\}$ that have (pairwise) non-negative covariance, $\text{cov}(g(X_j), g(X_l)) \geq 0$.

The following useful maximal inequality follows: $\forall x > 0$,

\[
P(\max_{1\leq j \leq K} S_{j,i} > x) \leq 1 - \prod_{j=1}^K P(S_{j,i} \leq x)
\]

i.e., the Bernoulli random variables $1\{S_{j,i} \leq x\}$ (a monotonically decreasing function of $S_{j,i}$) have non-negative covariance since

\[
P(\max_{1\leq j \leq K} S_{j,i} > x) = 1 - P(\max_{1\leq j \leq K} S_{j,i} \leq x).
\]
Bounds for \( K > 2 \) servers (cont)

- The stationary sojourn time \( S^{(K)} \) of a job has distribution satisfying, \( \forall x > 0 \):
  \[
P(S^{(K)} > x) = \lim_{i \to \infty} P(\max_{1 \leq j \leq K} S_{j,i} > x) 
  \leq 1 - \prod_{j=1}^{K} \lim_{i \to \infty} P(S_{j,i} \leq x),
\]
  where the last equality is for the M/M/1 queue.

- Using PASTA and conditioning on the number of jobs in a stationary M/M/1 queue (\( \sim \text{geom}(\rho) \)), one can show that the sojourn time of a job in steady-state \( \sim \exp(\alpha - \lambda) \), so that
  \[
P(S^{(K)} > x) \leq 1 - (1 - \exp((\alpha - \lambda)x))^K
\]

- Thus, one can show using
  \[
  ES^{(K)} = \int_{0}^{\infty} P(S^{(K)} > x)dx 
  \leq \int_{0}^{\infty} (1 - (1 - \exp((\alpha - \lambda)x))^K)dx =: H_K
  \]

Bounds for \( K > 2 \) servers - main result

- From the previous display, the mean sojourn time for the load-balanced case (\( \alpha = \beta \))
  \[
  ES^{(K)} \leq H_K.
  \]

- One can also show \( H_K = O(\log K) \), so that
  \[
  ES^{(K)} = O(\log K).
  \]

- Ignoring queuing delays, we get a simple lower bound
  \[
  ES^{(K)} \geq H_K/\alpha,
  \]
  giving some measure of tightness to the previous upper bound.
A martingale approach - background

• Following [Buffet and Duffield, JAP’94] consider a single queue with normalized service rate 1 and with jth job having service time xj and arrival time tj > tj−1.

• Define W as workload so that the queueing delay of the kth job is

\[
W(t_k-) = W(t_k) - x_k = \max_{l \leq k} \sum_{i=l}^{k-1} x_i - (t_k - t_i) = \max_{l \leq k} \sum_{i=l}^{k-1} (x_i - \tau_i)
\]

where the interarrival times \( \tau_i := t_{i+1} - t_i \) and 0 := \( \sum_{k}^{k-1} ... \).

• Stability requires \( E(x_i - \tau_i) < 0 \).

• If \( x_i - \tau_i \) are i.i.d. then for each \( k \in \mathbb{Z} \), we can choose the largest \( y > 1 \) so that \( Ey^{x_i - \tau_i} = 1 \) and

\[
Y_{k-1}^{(k)} := y \sum_{i=l}^{k-1} (x_i - \tau_i)
\]

is an (exponential) martingale for integers \( l \leq k \) with \( Y_{0}^{(k)} \equiv 1 \) and \( \forall i \geq 0, EY_{i}^{(k)} = 1 \).

• We can then use Doob’s maximal equality to obtain the bound,

\[
P(W(t_k) \geq \theta) = P(\max_{i \geq 0} Y_{i}^{(k)} \geq y^{\theta}) \leq y^{-\theta}.
\]

Martingale approach to a fork-join stage

• Let \( x_{j,i} \) be the duration of the jth task of ith job.

• The queueing delay of the kth job (time until the last of its tasks begins service) is therefore

\[
\max_j W_j(t_k-) = \max_j \max_{l \leq k} \sum_{i=l}^{k-1} (x_{j,i} - \tau_i)
\]

• By the union bound,

\[
P(\max_j W_j(t_k-) \geq \theta) \leq \sum_j P(W_j(t_k-) \geq \theta) \leq \sum_j y_j^{-\theta}
\]

• See [Rizk et al., SIGM.’15] for extensions to Markovian arrivals.

• Note that for a not work-conserving (blocking) case where the tasks of all future jobs \( l > k \) cannot start until all of job k complete, there is a single-queue equivalent:

\[
\max_{l \leq k} \sum_{i=l}^{k-1} (\max_j x_{j,i} - \tau_i) \geq \max_j W_j(t_k-).
\]
Markov decision processes (MDPs) - References

- Recall our previous discussion of
  - link-state and distance-vector routing and
  - discrete-time Markov chains.

Example - shortest path on a graph

- Suppose we are planning the construction of a highway from city A to city K.
- Different construction alternatives and their “edge” costs \( g \geq 0 \) between directly connected cities (nodes) are given in the following graph.
- The problem is to determine the highway (edge sequence) with the minimum total (additive) cost.
Recall Bellman’s principle of optimality

- If C belongs to an optimal (by edge-additive cost $J^*$) path from A to B, then the sub-path A to C and C to B are also optimal,

- i.e., any sub-path of an optimal path is optimal (easy proof by contradiction).

Dijkstra’s algorithm uses the predecessor node of the destination (path penultimate node), and is based on complete link-state (edge-state) info consistently shared among all nodes:

$$J^*(A, B) = \min_C \{ J^*(A, C) + g(C, B) \mid C \text{ is a predecessor of } B \},$$

i.e., C and B are adjacent nodes in the graph (endpoints of the same edge).

- The Iterated distributed Bellman-Ford algorithm instead uses the successor node of the path origin and only nearest-neighbor distance-vector information sharing:

$$J^*(A, B) = \min_C \{ g(A, C) + J^*(C, B) \mid C \text{ is a successor of } A \}$$

Discrete-time, deterministic scenario

- At “time” $n$,
  - $g_n(x_n, u_n) \geq 0$ is the cost,
  - $x_n$ is the state, and
  - $u_n$ is the control.

- State evolves according to

$$x_{n+1} = f_n(x_n, u_n), \quad \forall n \in \{0, 1, 2, \ldots, N-1\}.$$

- Given initial state $x_0$, the additive cost is

$$J_0(x_0, u_0) = \sum_{n=0}^{N-1} g_n(x_n, u_n) + g_N(x_N),$$

where $g_N$ is the terminal cost.

- Objective is to find the control $u_0 = \{u_n\}_{n=0}^{N-1}$ ($N$ decision variables $u_0$) that minimizes $J_0(x_0, u_0)$ - i.e., given the initial state $x_0$, dynamics $f$ and costs $g$,

$$\min_{u_0} J_0(x_0, u_0).$$
Discrete-time, deterministic scenario - problem variations

- We can, alternatively, maximize an additive total reward $J_0$ of rewards $g_n$ at $n$.
- Or, $J_0 = \max_{n \geq 0} g_n$ as maximum of signed rewards $g_n \in \mathbb{R}$.
- Or, $J_0 = \min_{n \geq 0} g_n$ as minimum of signed costs.

Discrete-time, deterministic scenario - backward induction

- The cost-to-go from time $k < N$ depends on the state $x_k$ and residual control $u_k = \{u_n\}_{n=k}^{N-1}$ ($N - k$ decisions),

$$J_k(x_k, u_k) = \sum_{n=k}^{N-1} g_n(x_n, u_n) + g_N(x_N)$$
$$= g_k(x_k, u_k) + J_{k+1}(x_{k+1}, u_{k+1})$$
$$= g_k(x_k, u_k) + J_{k+1}(f_k(x_k, u_k), u_{k+1}),$$

which we have written a function of just $x_k$, $u_k$ and $u_{k+1}$.

- Applying the optimality principle and state dynamics to minimize $J_0$,
  - we can work backward from time $N$ to find the optimal control $u^*_N$ before $u^*_k$,
  - thus finding $u^*_N = \{u^*_k, u^*_{k+1}\}$,

$$\forall x, u^*_N(x) = \arg\min_u J_{N-1}(x, u) = \arg\min_u g_{N-1}(x, u) + g_N(f_{N-1}(x, u))$$
$$\forall x, \forall k < N - 1, u^*_k(x) = \arg\min_u g_k(x, u) + J_{k+1}(f_k(x, u), u^*_{k+1}(f_k(x, u)))$$

- Note how optimal control at time $k < N$, $u^*_k$ depends on the current state $x = x_k$ and, for $k < N - 1$, on future optimal controls $u^*_{k+1}$ which are previously determined.
We will also model \( x \) as a Markov chain on its state space with transition probability matrix (TPM) \( P(k, u) \) which depends on the (not state anticipative) control at all times \( k \), i.e.,

\[ P_{ij}(k, u) = P(x_{k+1} = j \mid x_k = i, u_k = u), \]

and we’ve dispensed with the recursive update \( f_k \).

So, at each time \( k \) we choose from a (controlled) family of TPMs \( P(k, \cdot) \).

The marginal distribution \( \pi \) of \( x \) satisfies

\[ x_{k+1} \sim \pi^T(k + 1) = \pi^T(k)P(k, u_k). \]

Given the initial distribution \( \pi(0) \sim x_0 \), we wish to find the optimal control \( u_0 = \{u_n\}_{n=0}^{N-1} \) minimizing the expected additive cost

\[ V_0(\pi(0), u_0) := E_{\pi(0)}J_0(x_0, u_0) = E_{\pi(0)} \left( \sum_{n=0}^{N-1} g_n(x_n, u_n) + g_N(x_N) \right) \]

Recall that the expectation operator \( E \) is linear.

Given a state \( x \) governed by TPMs \( P \), we can write the principle of optimality for expected cost-to-go at time \( k < N \) as:

\[
V_k(\pi(k), \{u_k; u_{k+1}^*\}) := \min_{u_k} E_{\pi(k)} J_k(x_k, \{u_k; u_{k+1}^*\})
\]

\[
= \min_{u_k} E_{\pi(k)} g(x_k, u_k) + E_{\pi(k)} \left( \sum_{n=k+1}^{N-1} g_n(x_n, u_n^*) + g_N(x_N) \right)
\]

\[
= \min_{u_k} E_{\pi(k)} g(x_k, u_k) + E_{\pi(k)} V_{k+1}(\pi(k+1), u_{k+1}^*(\pi(k+1)))
\]

\[
= \min_{u_k} E_{\pi(k)} g(x_k, u_k) + V_{k+1}(\pi(k+1), u_{k+1}^*(\pi(k+1)))
\]

where in the last two equalities,

\[ \pi(k+1)^T = \pi(k)^T P(k, u_k). \]

Note how the minimizing \( u_k^* \) will depend on \( \pi(k) \sim x_k \) and future optimal controls \( u_{k+1}^* \).
To clarify:

\[
E_{\pi(k+1)}J_{k+1}(x_{k+1}, u_{k+1}) = \sum_x J_{k+1}(x, u_{k+1}) \pi_{k+1}(x)
= \sum_x J_{k+1}(x, u_{k+1}) \sum_{x'} \pi_k(x') P(x_{k+1} = x | x_k = x', u_k)
= \sum_x J_{k+1}(x, u_{k+1}) (\pi_k^T P(k, u_k))_x
= E_{\pi(k)}J_{k+1}(x_{k+1}, u_{k+1}),
\]

which depends on \(\pi_k\) and \(u_k\).
Discrete-time Markov decision processes - perturbations model (cont)

- Given the initial state $x_0$, the initial distribution $\alpha(0) \sim w_0$, and its TPM $P^{(w)}$, we wish to find the optimal control achieving the expected cost to minimize
  \[ V_0(x_0, \alpha(0), u^*_0) := \min_{\alpha_0} E_{\alpha(0)} J_0(x_0, u_0). \]

- So, we can write the principle of optimality for expected cost-to-go at time $k < N$ as:
  \[
  V_k(x_k, \alpha(k), \{u_k; u^*_k+1\}) := \min_{u_k} E_{\alpha(k)} J_k(x_k, \{u_k; u^*_k+1\})
  \]
  \[
  = \min_{u_k} E_{\alpha(k)} g(x_k, u_k, w_k) + E_{\alpha(k)} \left( \sum_{n=k+1}^{N-1} g_n(x_n, u^*_n, w_n) + g_N(x_N) \right)
  \]
  \[
  = \min_{u_k} E_{\alpha(k)} g(x_k, u_k, w_k) + V_{k+1}(\alpha(k + 1), w_{k+1}(\alpha(k + 1), x_{k+1})),
  \]
  where in the last equality,
  \[
  w_{k+1} \sim \alpha(k + 1)^T = \alpha(k)^T P^{(w)}(k) \quad \text{and} \quad x_{k+1} = f_k(x_k, u_k, w_k) \quad \text{with} \quad w_k \sim \alpha(k).
  \]

- Note how the minimizing $u_k$ will depend on $\alpha(k)$ and $x_k$ (and $u^*_k+1$).

Discrete-time Markov decision processes - perturbations model (cont)

For the special case of i.i.d. disturbances $w$:

- $\forall n$, $P^{(w)}(n) = I$,

- $w$ is stationary so that there is a distribution $\alpha$ such that, $\forall k$, $w_k \sim \alpha(k) = \alpha$ (does not depend on time $k$), and

- so indicating dependence of $V$ and $J$ on $\alpha$ may be suppressed.

© December 14, 2018 George Kesidis
Example - playing chess

- A strategic player plays against an opponent, where the (non-strategic) opponent does not change his actions in accordance with the current state.
- A draw fetches 0 points for both, a win fetches 1 point for the winner and 0 for the loser.
- They play $N$ independent games.
- If the scores are tied after $N$ games, then the players go to sudden death, where they play until one wins a game.

Example - playing chess - Timid and Bold strategies

- The (strategic) player can play "Timid", in that case draws a game with probability $p_d$ and loses with probability $1 - p_d$, i.e., cannot win playing Timid.
- The player can play "Bold", in that case wins a game with probability $p_w$ and loses with probability $1 - p_w$.
- Consideration of strategy is nontrivial when $p_d > p_w > 0$.
- Optimal strategy in sudden death? Play Bold (to win!)
Example - playing chess - set-up

- $u_k$: control is either Timid (0) or Bold (1) $\forall k$
- $w_k$: outcome of the $k^{th}$ game
  - Given Timid play, $P(w_k = 0|u_k = 0) = p_d$, $P(w_k = -1|u_k = 0) = 1 - p_d$
  - Given Bold play, $P(w_k = 1|u_k = 1) = p_w$, $P(w_k = -1|u_k = 1) = 1 - p_w$
- After $k$ games, strategic player leads by $x_k = w_{k-1} + x_{k-1}$ wins, with $x_0 := 0$.
- $S_k = \{-k, -(k - 1), ..., -1, 0, 1, ..., k - 1, k\}$: state space of $x_k$
- $N$: time horizon of optimization

Example - playing chess - reward function to optimize

- Now consider maximization of reward instead of minimization of cost.
- At time $N$, the probability of winning the whole match is
  \[
  EJ_N(x_N) = Eg_N(x_N) = \begin{cases} 
  0 & \text{if } x_N < 0 \\
  p_w & \text{if } x_N = 0 \quad \text{(need sudden death)} \\
  1 & \text{if } x_N > 0
  \end{cases}
  \]
- The probability of winning the whole match in $k < N$ games is zero (need to play at least $N$ games by rule) so
  \[
  Eg_k(x_k, u_k, w_k) = 0.
  \]
The Linear dynamics and Quadratic cost (LQ) framework

- Assume a perturbed model with linear-dynamics $f$ for state and perturbations $x_k, w_k \in \mathbb{R}^n$ and control $u_k \in \mathbb{R}^m$, i.e., there are deterministic matrix sequences $A_k \in \mathbb{R}^{n \times n}$, $B_k \in \mathbb{R}^{n \times m}$ such that

$$
\forall k < N, \quad x_{k+1} = A_k x_k + B_k u_k + w_k
$$

- Quadratic costs for non-negative definite matrices $0 \leq Q_k \in \mathbb{R}^{n \times n}$, $0 \leq R_k \in \mathbb{R}^{m \times m}$,

$$
g_N(x_N) = x_N^T Q_N x_N \quad \text{and} \quad g_k(x_k) = x_k^T Q_k x_k + u_k^T R_k u_k
$$

$$
J_j(x_j) = \sum_{k=j}^{N} g_k(x_k),
$$

where the cost-to-go at time $j$, $J_j$, depends on control $u_j = \{ u_k \}_{k=j}^N$.

- When $w$ is a zero-mean sequence of unit variance, can directly show that optimal linear control is $u_k(x_k) = L_k x_k$ where

$$
L_k = -(B_k^T K_{k+1} B_k + R_k)^{-1} B_k^T K_{k+1} A_k \quad \text{for } k < N, \quad \text{where}
$$

$$
K_N = Q_N \quad \text{and, with } K_k \text{ determined in backward order},
$$

$$
K_k = -A_k^T (K_{k+1} - K_{k+1} B_k (B_k^T K_{k+1} B_k + R_k)^{-1} B_k^T K_{k+1}) A_k + Q_k
$$

- The minimum resulting cost is

$$
V^*(x_0) = x_0^T K_0 x_0 + \sum_{k=0}^{N-1} E(w_k^T K_{k+1} w_k)
$$
Linear dynamics, Quadratic cost (LQ) - time-invariant case

- If $A_k = A$, $B_k = B$, $R_k = R$, $Q_k = Q$ (time-invariant/homogeneous case), then as time $k$ becomes large, $K_k$ converges to the steady-state solution of algebraic Riccati equation,

$$K = -A^T(K - KB(B^T KB + R)^{-1}B^T K)A + Q$$

- So, the LQ-optimal control is $u(x) = Lx$, where

$$L = -(B^T KB + R)^{-1}B^T KA$$

Optimal stopping problems

- Suppose in any time-slot, one of the control actions stops the system.

- The decision maker can terminate the system at a certain loss or choose to continue at a certain cost.

- The challenge will be when to stop so as to minimize the total/final expected cost.

- For example, decision maker possesses an asset that is the subject of sequential offers $w_n$ and the question is which offer to take?
Optional stopping example - asset selling problem

- A decision-maker has an asset for which he receives quotes/offers in every time-slot, \( w_0, \ldots, w_{N-1} > 0 \).
- Quotes are independent from slot to slot and identically distributed.
- If the offer is accepted, it is then invested to earn a fixed rate of interest \( r > 0 \).
- Control action \( u_k \) for \( k > 0 \) is to sell or not to sell at slot \( k \) based on offer \( w_k \).
- State is the offer in the previous slot if the asset is not sold yet, or a flag \( S < 0 \) if it was previously sold (terminating state),

\[
x_{k+1} = \begin{cases} S & \text{if sold in previous slots (< k + 1)} \\ w_k & \text{otherwise} \end{cases}
\]

Asset selling problem - rewards

- So, \( x_k \neq S \) means \( x_k = w_{k-1} > 0 \).
- Reward at \( N \) is

\[
J_N(x_N) = \begin{cases} g_N(x_N) = x_N = w_{N-1} & \text{if } x_N \neq S \text{ (not prev. sold, take final offer)} \\ 0 & \text{if } x_N = S \text{ (prev. sold)} \end{cases}
\]
- Terminal reward at step \( k < N \) is sale price plus interest till \( N \) if sale is made,

\[
g_k(x_k, u_k, w_k) = \begin{cases} (1 + r)^{N-k} x_k & \text{if } x_k \neq S \text{ and } u_k = \text{sell (at } k, \text{ so, } x_k = w_{k-1}) \\ 0 & \text{if } x_k = S \text{ or } u_k = \text{don’t sell (at } k) \end{cases}
\]
- So only one of the \( g_k \) will be nonzero.
- Reward-to-go at \( k < N \) is \( \max\{\text{sell at } k, \text{ don’t sell at } k\} \) if not previously sold or \( 0 \) if previously sold:

\[
V_k(x_k) = \begin{cases} \max\{(1 + r)^{N-k} w_{k-1}, EJ_{k+1}(w_k)\} & \text{if } x_k \neq S \text{ (} x_k = w_{k-1}) \\ 0 & \text{if } x_k = S \end{cases}
\]
Asset selling problem - optimal control is threshold

- Let the expected discounted future reward be
  \[ \alpha_k = EJ_{k+1}(w_k)/(1+r)^{N-k} = EJ_{k+1}(x_{k+1})/(1+r)^{N-k} \] when \( x_k \neq S \).
- So, \( J_k(x_k) = (1+r)^N \max\{x_k, \alpha_k\} \).
- So by backward induction, the optimal (maximizing reward) control strategy \( u_k \) is:
  - Accept the offer \((w_k)\) if \( x_k > \alpha_k \)
  - Reject the offer if \( x_k < \alpha_k \)
  - Act either way otherwise

Asset selling problem - threshold non-increasing in time

Theorem: \( \alpha_k \) is non-increasing function of \( k \), i.e., \( \forall k < N, \alpha_k \geq \alpha_{k+1} \).

Proof: We will show by backward induction that \( \forall x \geq 0 (x_k \neq S) \):
- \( \alpha_{N-1} := EJ_N(w)/(1+r) = Ew/(1+r) \), noting \( w_k \) are assumed i.i.d.
- \( \alpha_{N-2} := EJ_{N-1}(w)/(1+r)^2 = \max\{Ew/(1+r), EJ_N(w)/(1+r)^2\} \).
- Thus, \( \alpha_{N-1} := EJ_N(w)/(1+r) \leq EJ_{N}(w)/(1+r)^2 = \alpha_{N-2} \), i.e., we’ve established the base case.
- Assume \( \alpha_{k-1} := EJ_k(w)/(1+r)^{N-k+1} \geq EJ_{k-1}(w)/(1+r)^{N-k+2} =: \alpha_{k-2} \) for some arbitrary \( k \leq N \).
- Thus,
  \[ \alpha_{k-3} := EJ_{k-2}(x)/(1+r)^{N-k+3} = \max\{(1+r)^{-1}Ew, EJ_{k-1}(w)/(1+r)^{N-k+2}\} \leq \max\{(1+r)^{-1}Ew, EJ_{k}(w)/(1+r)^{N-k+1}\} \quad \text{(by inductive assumption)} \]
  \[ = EJ_{k-1}(w)/(1+r)^{N-k+2} =: \alpha_{k-2}. \]
- Q.E.D.
Asset selling problem - iterative computation of threshold

- Let $V_k(x_k) = J_k(x_k)/(1 + r)^{N-k}$ when $x_k \neq S$, i.e., $x_k = w_{k-1} > 0$, decision to sell has not been made, and threshold is still relevant.

  \[ V_N(x_N) = x_N \text{ (again, } x_N \neq S) \]

  \[ \forall k < N, \ V_k(x_k) = \max\{x_k, (1 + r)^{-1}E V_{k+1}(w)\} \text{ (again, } w_k \text{ assumed i.i.d.)} \]

- Since $\alpha_k := EV_{k+1}(w)/(1 + r)$,

  \[ V_k(x_k) = \max\{x_k, \alpha_k\} = \max\{w_{k-1}, \alpha_k\} \]

- Thus,

  \[ \alpha_k = EV_{k+1}(w)/(1 + r) \]

  \[ = E \max\{w, \alpha_k + 1\}/(1 + r) \]

  \[ = \left( \int_0^{\alpha_k+1} dF_w(z) + \int_{\alpha_k+1}^{\infty} zdF_w(z) \right)/(1 + r), \]

  where $F_w$ is the cumulative distribution function of $w$.

- Note that the first term is $\alpha_k+1P(w \leq \alpha_k+1) \leq \alpha_k+1 < \infty$ and the second term is $\leq Ew < \infty$ by assumption.

- So, $\alpha_k > 0$ is a bounded, monotonically non-increasing sequence, so it must converge.

- For large $k$, the sequence converges to solution $\alpha$ of

  \[ \alpha = \left( \int_0^\alpha dF_w(z) + \int_{\alpha}^{\infty} zdF_w(z) \right)/(1 + r) \]

  \[ = \left( \alpha P(w \leq \alpha) + \int_{\alpha}^{\infty} zdF_w(z) \right)/(1 + r) \]

Asset selling problem - iterative computation of threshold (cont)
Background on constrained optimization and duality

- Consider a primal optimization problem with a set of \( m \) inequality constraints: Find
  \[
  \arg \min_{x \in D} f_0(x),
  \]
  where the constrained domain of optimization is
  \[
  D \equiv \{ x \in \mathbb{R}^n \mid f_i(x) \leq 0, \forall i \in \{1, 2, \ldots, m\} \}.
  \]
- For the example of a loss network, the constraints are
  \[
  f_l(x) = (Ax)_l - c_l = \sum_{r \mid l \in r} x_r - c_l,
  \]
  where the index \( l \) corresponds to a link and \( m \) is the number of links in the network (and here \( x_r \in \mathbb{Z}^+ \)).
- To study the primal problem, we define the corresponding Lagrangian function on \( \mathbb{R}^{n+m} \):
  \[
  L(x, v) \equiv f_0(x) + \sum_{i=1}^{m} v_i f_i(x),
  \]
  where, by implication, the vector of Lagrange multipliers is \( v \in [0, \infty)^m \), i.e., non-negative \( v \geq 0 \).

Primal constrained optimization with Lagrange multipliers

- Theorem:
  \[
  \min_{x \in \mathbb{R}^n} \max_{v \geq 0} L(x, v) = \min_{x \in D} f_0(x) \equiv p^*.
  \]
- Proof: Simply,
  \[
  \max_{v \geq 0} L(x, v) = \begin{cases} 
  \infty & \text{if } x \notin D, \\
  f_0(x) & \text{if } x \in D,
  \end{cases}
  \]
  \( \square \)
- Note that if \( x \notin D \) then \( \exists \ i > 0 \) s.t. \( f_i(x) > 0 \) \( \Rightarrow \) optimal \( v_i^* = \infty \).
- So, we can maximize the Lagrangian in an unconstrained fashion to find the solution to the constrained primal problem.
Complementary slackness of primal solution

- Define the maximizing values of the Lagrange multipliers,
  \[ v^*(x) \equiv \arg \max_{v \geq 0} L(x, v) \]
  and note that the complementary slackness conditions
  \[ v^*_i(x) f_i(x) = 0 \]
  hold for all \( x \in D \) and \( i \in \{1, 2, \ldots, m\} \).

- That is, if there is slackness in the \( i \)th constraint, i.e., \( f_i(x) < 0 \), then there is no slackness in the constraint of the corresponding Lagrange multiplier, i.e., \( v^*_i(x) = 0 \).

- Conversely, if \( f_i(x) = 0 \), then the optimal value of the Lagrange multiplier \( v^*_i(x) \) is not relevant to the Lagrangian.

- Complementary slackness conditions lead to the Karush-Kuhn-Tucker necessary conditions for optimality of the primal solution.

The dual problem

- Now define the dual function of the primal problem:
  \[ g(v) = \min_{x \in \mathbb{R}^n} L(x, v). \]

- Note that \( g(v) \) may be infinite for some values of \( v \) and that \( g \) is always concave.

- **Theorem:** For all \( x \in D \) and \( v \geq 0 \),
  \[ g(v) \leq f_0(x). \]

- **Proof:** For \( v \geq 0 \),
  \[ g(v) \leq L(x, v) \leq \max_{v \geq 0} L(x, v) = f_0(x), \]
  where the last equality is the bound on \( L \) assuming \( x \in D \). \( \square \)
The dual problem (cont)

• So, by the previous theorem, if we solve the dual problem, i.e., find
  \[ d^* \equiv \max_{\nu \geq 0} g(\nu), \]
  then we will have obtained a (hopefully good) lower bound to the primal problem, i.e.,
  \[ d^* \leq p^*. \]

• Under certain conditions in this finite dimensional setting, in particular when the primal problem is convex and a strictly feasible solution exists, the duality gap
  \[ p^* - d^* = 0. \]

The dual problem for a linear program

• If \( f_0(\bar{x}) = \sum_{i=1}^{m} \phi_i x_i \) and all \( f_i(\bar{x}) = \xi_i + \sum_{j=1}^{n} \gamma_{i,j} x_j \) are linear functions, then the above primal problem, \( \min_\bar{x} f_0(\bar{x}) \text{ s.t. } f_i(\bar{x}) \leq 0 \ \forall i \), is called a Linear Program (LP).

• Exercise: Find an equivalent dual LP. Hint: first show the Lagrangian of the primal problem can be written as
  \[ L(\bar{x}, \nu) = \sum_{i=1}^{m} \xi_i v_i + \sum_{j=1}^{n} x_j \left( \phi_j + \sum_{i=1}^{m} v_i \gamma_{i,j} \right). \]

• LPs can be solved by the simplex algorithm (along feasible region boundaries) or by interior point methods.

• Some references:
Iterated subgradient method

- To use duality to find $p^*$ and $\mathbf{x}^* = \arg\max_{\mathbf{x}\in D}f_0(\mathbf{x})$ in this case, suppose that a slow ascent method is used to maximize $g$,

$$v_n = v_{n-1} + \alpha_1 \nabla g(v_{n-1}),$$

and between steps of the ascent method, a fast descent method is used to evaluate $g(v_n)$ by minimizing $L(\mathbf{x}, v_n)$,

$$\mathbf{x}_k = \mathbf{x}_{k-1} - \alpha_2 \nabla \mathbf{x} L(\mathbf{x}_{k-1}, v_n).$$

- The process described by such an ascent/descent method is called an iterative subgradient method.

- The step sizes $\alpha$ can be chosen dynamically, e.g., steepest ascent/descent (i.e., itself the result of optimization).

- Instead of slow ascent, the descent step can be projected on the feasible domain $D$.

KKT conditions

- Consider again a primal optimization problem with a set of $m$ inequality constraints: Find

$$\arg\min_{\mathbf{x}\in D}f_0(\mathbf{x}),$$

where the constrained domain of optimization is

$D \equiv \{\mathbf{x} \in \mathbb{R}^n \mid f_i(\mathbf{x}) \leq 0, \forall i \in \{1, 2, ..., m\}\}.$

- So the Lagrangian on $(\mathbf{x}, v) \in \mathbb{R}^n \times (\mathbb{R}^+)^m$ is

$$L(\mathbf{x}, v) \equiv f_0(\mathbf{x}) + \sum_{i=1}^{m} v_i f_i(\mathbf{x}).$$

and our objective is to find $\min_\mathbf{x} \max_{v \geq 0} L$.

- If $f_0$ is convex and, $\forall i \geq 1$, $f_i$ is linear, then the following Krush-Kuhn-Tucker (KKT) conditions are sufficient for optimality:

$$\forall j, \quad \partial L/\partial x_j = 0 \quad \text{and}$$

$$\forall i, \quad v_i f_i = 0 \quad \text{(complementary slackness)}. $$
Example - Max-Min Fair (MMF) allocation: problem set-up and def’n

- Suppose a set of $N$ processes require service from a set of $M$ cores (processors).
- Let $\delta_{n,m} \in \{0, 1\}$ indicate whether process $n \in N$ prefers core $m \in M$.
- Let $\phi_n$ be the weight or priority of process $n \in N$.
- Let $s_m$ be the capacity of core $m$.
- Finally, let $x_{n,m}$ be the fraction of core $m$ allocated to process $n$, where $\delta_{n,m} = 0 \Rightarrow x_{n,m} = 0$.

The normalized total allocation to process $n$ is

$$F_n := \frac{\sum_{m \in M} x_{n,m} \delta_{n,m} s_m}{\phi_n}.$$ 

- $x$ is a MMF allocation if the following condition holds: if $x_{n,m} > 0$, $\delta_{k,m} = 1$ and $F_k > F_n$, then $x_{k,m} = 0$.
- In other words, at a MMF allocation, all processes receiving positive allocation ($x > 0$) by any given core must have the same normalized total allocations ($F$).

Example - Max-Min Fair (MMF) allocation by constrained convex opt

- Consider the Lagrangian with Lagrange multipliers $v \geq 0$:

$$L = \sum_{n \in N} \phi_n g(F_n) + \sum_{m \in M} v_m \left( \sum_{n \in N} x_{n,m} - 1 \right) + \sum_{n,m} v_{n,m} (-x_{n,m})$$

where $g$ is strictly convex and $g'$ strictly increasing (e.g., $g(F) = -\log(F)$).

- The KKT conditions for optimality require that if $\delta_{n,m} > 0$ then

$$s_m g'(F_n) + v_m - v_{n,m} = 0 \Rightarrow F_n = (g')^{-1} \left( \frac{v_{n,m} - v_m}{s_m} \right)$$

where we note that $g'$ strictly increasing $\Rightarrow (g')^{-1}$ strictly increasing.

- If $x_{n,m} > 0$, then $v_{n,m} = 0$ by complementary slackness.

- Additionally, if $\delta_{k,m} = 1$ then

$$F_k = (g')^{-1} \left( \frac{v_{k,m} - v_m}{s_m} \right) \geq (g')^{-1} \left( \frac{-v_m}{s_m} \right) = F_n,$$

which is the definition of MMF allocation [Khamse-Ashari et al., GLOBECOM, 2016].

- So, the solution of the above convex optimization is the MMF allocations.
Example - load balancing in a network of parallel routes

• Consider a total demand of $\Lambda$ between two network end-systems having $R$ disjoint routes connecting them.

• On route $r$, the service capacity is $c_r$ and the fraction of the demand applied to it is $\pi_r$, where $\sum_r \pi_r = 1$ and $\forall r, c_r > \pi_r \Lambda$ (the latter for stability).

• Consider the problem of the routing decisions that minimize the mean number of jobs in the system,

$$N(\pi) = \sum_r \frac{\pi_r \Lambda}{c_r - \pi_r \Lambda},$$

where this expression is clearly derived from that of an M/M/1 queue.

• To find optimal $\pi$, we can first try to use a Lagrangian with just one of the inequality constraints, $\sum_r \pi_r \geq 1$:

$$L(\pi, q) = N(\pi) + v(1 - \sum_r \pi_r) = \sum_r \left( -1 + \frac{c_r}{c_r - \pi_r \Lambda} \right) + v(1 - \sum_r \pi_r).$$

• Note that for stable $\pi$, $L$ is increasing in every $\pi_r$.

• Since $L$ is concave in $\pi$, there will be zero duality gap allowing us to minimize over $\pi$ first.

---

Example - load balancing (cont)

• By the first-order necessary conditions, $\forall i$, $\partial L / \partial \pi_r = 0$, the minimizing

$$\pi_r^* = \frac{c_r}{\Lambda} - \sqrt{\frac{c_r}{\Lambda v}}$$

• To meet the equality constraint $\sum_r \pi_r^* = 1$ (maximize the dual function), the Lagrange multiplier $v$ is

$$\sqrt{\Lambda v} = \frac{\sum_r \sqrt{c_r}}{-1 + \sum_r c_r / \Lambda} \Rightarrow v = \left( \frac{\sum_r \sqrt{c_r}}{\sum_r c_r / \Lambda} \right)^2 \text{ and } \pi_r^* = \frac{c_r}{\Lambda} - \sqrt{\frac{c_r}{\Lambda v}} \left( -1 + \sum_j \frac{c_j}{\Lambda} \right),$$

where

- the first equality requires the system stability condition $\sum_r c_r > \Lambda$, and
- stability in each route is achieved, $c_r > \pi_r^* \Lambda$.

• Note that if route capacities $c_r$ are highly imbalanced, it’s possible that this $\pi_r^* < 0$ for routes $r$ with smallest $c_r$, in which case the constraints $\pi_r \geq 0$ need to be considered in the Lagrangian (exercise) - else if $c_r \approx c_s \forall r, s$, then $\pi_r^* \approx$ uniform ($\geq 0$).

• By Little’s theorem, $\pi^*$ also minimizes mean delay $\sum_r \pi_r \left( \frac{1}{c_r - \pi_r \Lambda} \right) = N(\pi) / \Lambda$.

• This model was extended to an end-user game in [Korilis et al. INFOCOM’97].
An “efficient” game among routed flows in a network


- Consider $R$ users sharing a network consisting of $m$ links (hopefully without cycles) each connecting a pair of nodes.

- We identify a single fixed route $r$ with each user, where, again, a route is simply a group of connected links.

- Thus, the user associated with each route could, in reality, be an aggregation of many individual flows of smaller users.

- Each link $l$ has a capacity of $c_l$ bits per second and each user $r$ transmits at $x_r$ bits per second.

- Link $l$ charges $\kappa_lX$ dollars per second to a user transmitting $X$ bits per second over it.

---

Noncooperative network-game formulation

- Suppose that user $r$ derives a certain benefit from transmission of $x_r$ bits per second on route $r$.

- The value of this benefit can be quantified as $U_r(x_r)$ dollars per second.

- A user utility function $U_r$ is often assumed to have the following properties: $U_r(0) = 0$, $U_r$ is nondecreasing, and, for elastic traffic, $U_r$ is concave.

- The concavity property is sometimes called a principle of diminishing returns or diminishing marginal utility.
Noncooperative network-game formulation (cont)

- Note that user \( r \) has net benefit (net utility)
  \[
  U_r(x_r) = x_r \sum_{l \in r} \kappa_l.
  \]

- Suppose that, as with the loss networks, the network wishes to select its prices \( \kappa \) so as to optimize the total benefit derived by the users, i.e., the network wishes to maximize “social welfare,” for example,
  \[
  -f_0(x) \equiv \sum_{r=1}^{R} U_r(x_r),
  \]
  subject to the link capacity constraints
  \[
  f_l(x) = (A x)_l - c_l \leq 0 \quad \text{for } 1 \leq l \leq m.
  \]

- We can therefore cast this problem in the primal form using the Lagrangian,
  \[
  L(x, v) \equiv f_0(x) + \sum_{i=1}^{m} v_i f_i(x).
  \]

Dual problem formulation

- Since all of the individual utilities \( U_r \) are assumed concave functions on \( \mathbb{R} \), \( f_0 \) is convex on \( \mathbb{R}^n \).

- Since the inequality constraints \( f_i \) are all linear, the conditions for zero duality gap are satisfied.

- So, we will now formulate a distributed solution to the dual problem in order to solve the primal problem.

- First note that, because of convexity, a necessary and sufficient condition to minimize the Lagrangian \( L(x, v) \) over \( x \) (to evaluate the dual function \( g \)) is
  \[
  \nabla_x L(x^*(v), v) = 0.
  \]
Solving the dual problem

- For the problem under consideration,
  \[ \frac{\partial L(x, v)}{\partial x_r} = -U'_r(x_r) + \sum_{l \in r} v_l = -U'_r(x_r) + (A^T v)_r. \]

- Therefore, for all \( r \),
  \[ x^*_r(v) = (U'_r)^{-1} \left( \sum_{l \in r} v_l \right) = (U'_r)^{-1}((A^T v)_r), \]
  where the right-hand side is made unambiguous by the above assumptions on \( U_r \).

Solving the dual problem - ascent-descent framework

- Assume that, at any given time, user \( r \) will act (select \( x_r \)) so as to maximize their net benefit, \( i.e., \)
  \[ \arg \max_{x \geq 0} U_r(x) - x \sum_{l \in r} \kappa_l = (U'_r)^{-1} \left( \sum_{l \in r} \kappa_l \right) =: y_r, \]
  where this quantity is simply \( x^*_r(\kappa) \).

- That is, the prices \( \kappa \) correspond to the Lagrange multipliers \( v \).

- So, the dual function
  \[ g(\kappa) = L(x^*(\kappa), \kappa), \]
  \( i.e., \) for fixed link costs \( \kappa \), the decentralized actions of greedy users minimize the Lagrangian and, thereby, evaluate the dual function.

- So, at fixed prices, the noncooperative game played by the users is efficient in that social welfare \( -f_0 \) is maximized at their Nash equilibrium.

- A Nash equilibrium is a set of play-actions \( x^* \) where no single user can benefit from unilateral defection.
Solving the dual problem - ascent-descent framework (cont)

- Following the ascent-descent framework of the dual algorithm, suppose that the network slowly modifies its link prices to maximize \( g(\kappa) \), where by “slowly” we mean that the greedy users are able to react to a new set of link prices well before they change again.

- To apply the ascent method to modify the link prices, we need to evaluate the gradient of \( g \) to obtain the ascent direction.

- Since
\[
\frac{\partial g(\kappa)}{\partial \kappa_l} = [(A\bar{x}^*(\kappa))_l - c_l] - \sum_r U'_r(x^*_r(\kappa)) \frac{\partial x^*_r(\kappa)}{\partial \kappa_l} + \sum_r \kappa_r \sum_{r'\in r} \frac{\partial x^*_r(\kappa)}{\partial \kappa_l}
\]

- for each link \( l \) the ascent rule for link prices becomes
\[
(\kappa_l)_n = (\kappa_l)_{n-1} + \alpha_1 ((A\bar{x}^*(\kappa_{n-1}))_l - c_l)
\]

or, in vector form,
\[
\kappa_n = \kappa_{n-1} + \alpha_1 (A\bar{x}^*(\kappa_{n-1}) - \underline{c})
\]

- Note that these link price updates depend only on “local” information such as link capacity and price and link demand, \((A\bar{x}^*(\kappa_{n-1}))_l\), where the latter can be empirically evaluated.

Solving the dual problem - ascent-descent framework (cont)

- Suppose that we initially begin with very high prices \( \kappa_0 \) so that demands \( x^*(\kappa) \) are very small.

- The action of the previous link-price updates will be to lower prices and, correspondingly, increase demand.

- The prices will try to converge to a point \( \kappa^* \), where supply \( \underline{c} \) equals demand \( A\bar{x}^*(\kappa^*) \).