IEDA3250 Stochastic Models

Wei YOU



THE HONG KONG UNIVERSITY OF
SCIENCE AND TECHNOLOGY

Fall 2026

Contents

1. Introduction ..... 1.
   - 1.1. Same Average, Different Experience ..... 3.
   - 1.2. How to Describe an Uncertain System ..... 4.
   - 1.3. Why This Course Exists in the IEDA Curriculum ..... 9.
   - 1.4. Same Average, Different Outcomes ..... 11.
   - 1.5. From Random Quantities to Random Evolution ..... 25.
   - 1.6. What This Course Will Teach ..... 34.
   - 1.7. How We Will Learn ..... 35.
2. Markov Chains ..... 38.
   - 2.1. From Coin Tosses to One-Step Forecasts ..... 38.
   - 2.2. Discrete-Time Markov Chains ..... 49.
   - 2.3. More Examples of Discrete-Time Markov Chains ..... 63.
   - 2.4. Chapman–Kolmogorov Equations: Multi-Step Transitions ..... 75.
   - 2.5. Simple Random Walk: Fluctuations and Hitting Times ..... 89.
   - 2.6. Classification of States ..... 110.
   - 2.7. Long-run Behavior and Performance ..... 121.
3. Poisson Processes ..... 146.
   - 3.1. Events in Continuous Time ..... 146.
   - 3.2. A Model for Random Arrivals ..... 156.
   - 3.3. Forecasting Counts ..... 171.

3.4. Waiting for Arrivals ..... 179.
3.5. Arrivals Given a Total ..... 189.
3.6. Combining and Classifying Arrivals ..... 200.
3.7. Which Event Happens Next? ..... 212.
3.8. Concluding Remarks ..... 220.
4. Continuous-Time Markov Chains ..... 232.
4.1. Exponential Clocks and Markov States ..... 232.
4.2. Transition Rates and the Generator ..... 239.
4.3. Evolution of State Probabilities ..... 256.
4.4. Long-Run Flow and Performance ..... 266.
4.5. Birth-and-Death Models ..... 279.
5. Queueing ..... 285.
5.1. M/M/1: Occupancy and Waiting ..... 285.
5.2. Little's Law: Population, Throughput and Time ..... 311.
5.3. Finite Capacity and Admission ..... 328.
5.4. Resource Pooling ..... 333.
5.5. Beyond Exponential Service ..... 352.
6. Stochastic Inventory Models ..... 371.
6.1. Stocking for a Single Period: The Newsvendor Problem ..... 371.
6.2. Replenishment with a Stock Target ..... 388.
6.3. Fixed Ordering Fees and Batching ..... 406.

6.4. Replenishment with Lead Times ..... 419.
7. Markov Decision Processes ..... 441.
7.1. Finite-Horizon Markov Decision Processes ..... 441.
7.2. Infinite-Horizon Discounted Reward ..... 459.
A Proofs ..... 481.
B Further DTMC Calculations ..... 488.
C Detailed Balance and Reversibility ..... 492.
D CTMC Constructions and Calculations ..... 497.
E Queueing Reference ..... 513.
F Markov Decision Process Reference ..... 526.
F.1 Queue Admission Control ..... 533.

1. Introduction

A canteen counter completes **one order per minute**. Six students arrive during six minutes.

Smooth arrivals

One student arrives in each minute.

Time	0	1	2	3	4	5
Arrivals	1	1	1	1	1	1

Bursty arrivals

All six students arrive at the beginning.

Time	0	1	2	3	4	5
Arrivals	6	0	0	0	0	0

Predict before calculating

Poll 1.1

Both patterns have the same total demand and the same average demand rate. Will students experience the same waiting time?

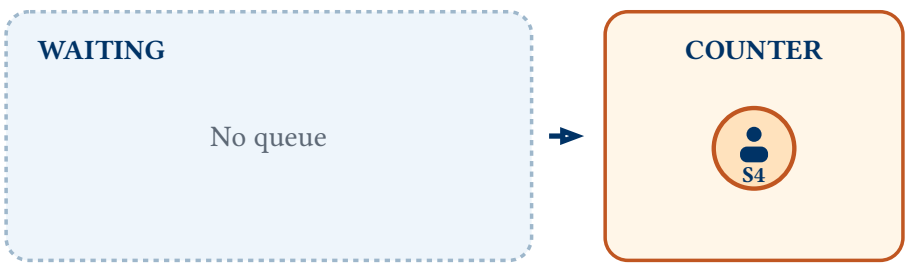
A). Yes: equal averages imply equal performance.

B). No: the timing of demand can change performance.

Canteen Queue Dynamics

Smooth arrivals

One student each minute

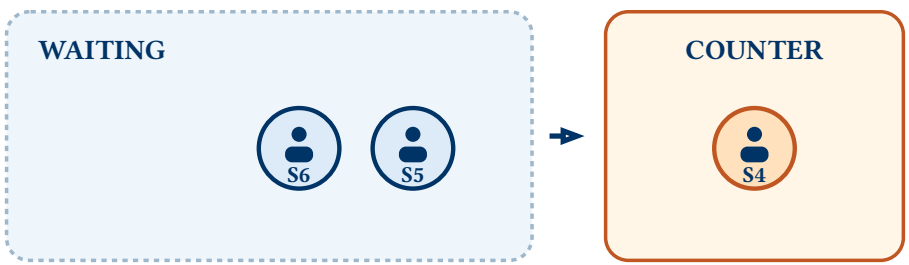


CUMULATIVE WAIT
0 min

STUDENTS DELAYED
0

Bursty arrivals

Six students together every 6 minutes



CUMULATIVE WAIT
6 min

STUDENTS DELAYED
3

1.1. Same Average, Different Experience

? Write down the waiting time of each student.

Smooth arrivals

Waiting times: $0, 0, 0, 0, 0, 0$

Average wait: 0 minutes

Bursty arrivals

Waiting times: $0, 1, 2, 3, 4, 5$

Average wait: $\frac{0+1+2+3+4+5}{6} = 2.5$ minutes

✓ Observation

Same average demand. Same capacity. Different waiting experience.

The average tells us how much work arrives overall. It does not tell us **when** the work arrives or how unfinished work accumulates.

1.2. How to Describe an Uncertain System

Three questions will organize every model in this course

Where is the system now?

Identify the **state** of the system.

Example: number of orders waiting.

What can happen next?

Identify the **random event** and how it *changes the state.*

Example: an arrival or a completed order.

What do we measure?

Identify the **performance measure**.

Example: waiting time and number of open counters.



Working definition

A **stochastic model** describes what uncertain events may occur, how they change the system, and what performance can result.

Random events → State evolution → Performance → Decision

1.2.1. A Reusable Model Card

Component	Question to ask
Clock	When can the system change—in steps or continuously?
State	What information are we tracking now?
Random event	What uncertain event may happen next?
Update rule	How does that event change the state?
Performance	What outcome do we care about?
Decision	What can an operator control?

Canteen queue model card

Example 1.1

- **Clock**: minutes.
- **State**: number of orders waiting and the order in service.
- **Random events**: new orders and service completions.
- **Update**: arrivals add work; completions remove work.
- **Performance**: average wait and probability of waiting more than five minutes.
- **Decision**: how many counters to open.

1.2.2. Model Detective: Food Delivery

Food delivery example **Poll 1.2**

A delivery platform tracks customers, merchants, and couriers in real time.

Identify:

- the **state** of the system;
- three uncertain quantities;
- one performance measure;
- one decision the platform can make.

Customer

Places an order and observes a promised time.

Merchant

Prepares the order with uncertain readiness time.

Courier

Travels through uncertain traffic and completes pickup and delivery.

1.2.3. One Possible Model—and the Pattern Across Industries

Food-delivery model card

Example 1.2

- **State**: pending orders, merchant readiness, courier locations and status.
- **Random events**: new orders, preparation times, travel times, cancellations.
- **Update**: orders arrive; merchants finish; couriers move, pick up, and deliver.
- **Performance**: delivery delay and probability of missing the promised time.
- **Decisions**: dispatch, batching, repositioning, and incentives.



Different industries can share the same stochastic structure.

System	State	Uncertainty	Performance	Decision
LLM service	queued workload	arrivals and output lengths	response delay	routing and capacity
Delivery	orders and couriers	prep and travel times	lateness	dispatch
Fashion	inventory by store	demand and returns	stockout / overstock	allocation and replenishment

1.3. Why This Course Exists in the IEDA Curriculum

IEDA3010

Formulate and optimize decisions once model inputs are specified.

IEDA3250

Describe uncertain inputs and evolving states; calculate performance distributions.

Later IEDA courses

Use stochastic models to optimize service, inventory, finance, pricing, and risk decisions.

Decisions can occur at different times

Some decisions are committed **before** uncertainty is revealed; others adapt as new information arrives.

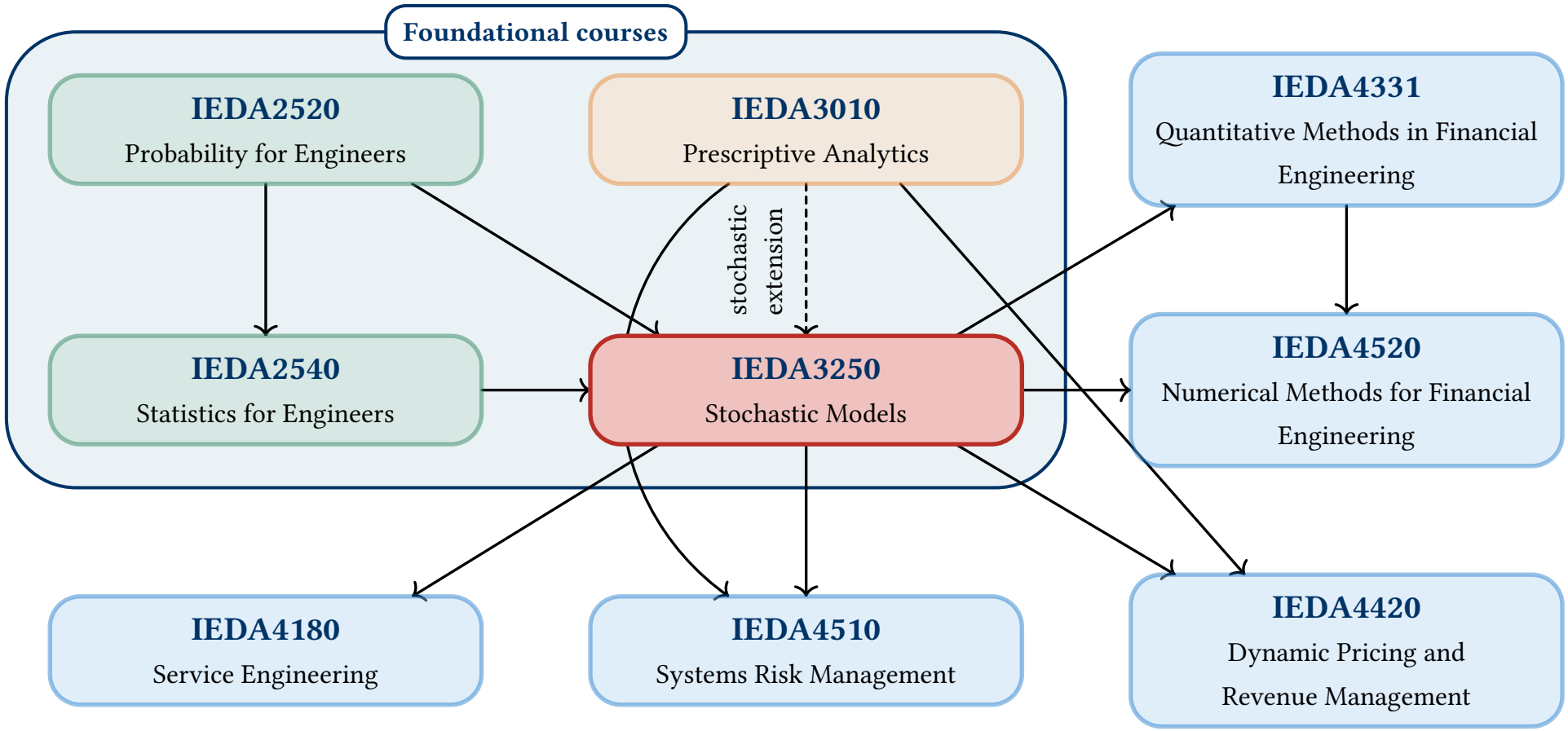


Figure 1.1: IEDA curriculum map: Stochastic Models in context

1.4. Same Average, Different Outcomes

Three familiar deterministic conclusions

Example 1.3

1. The mean bus gap is 20 minutes, so the mean passenger wait must be 10 minutes.
2. Average traffic flow is below road capacity, so persistent congestion should not occur.
3. Mean demand is 10, so inventory 10 should be the natural decision.

⚠ Each conclusion can fail—for a different reason

- A random observer can be biased toward long-lasting intervals.
- A loaded system can amplify small disturbances.
- Rare outcomes and asymmetric losses can change the best decision.

Conclusion

The distribution matters, but so does how the system transforms it.

1.4.1. Sampling Bias in Time: Waiting for the Bus

Perfectly regular service

Buses arrive exactly every 20 minutes.

A passenger arrives at a uniformly random time in a gap.

Expected wait: 10 minutes.

Irregular service

Bus gaps alternate between 10 and 30 minutes.

The mean gap is still

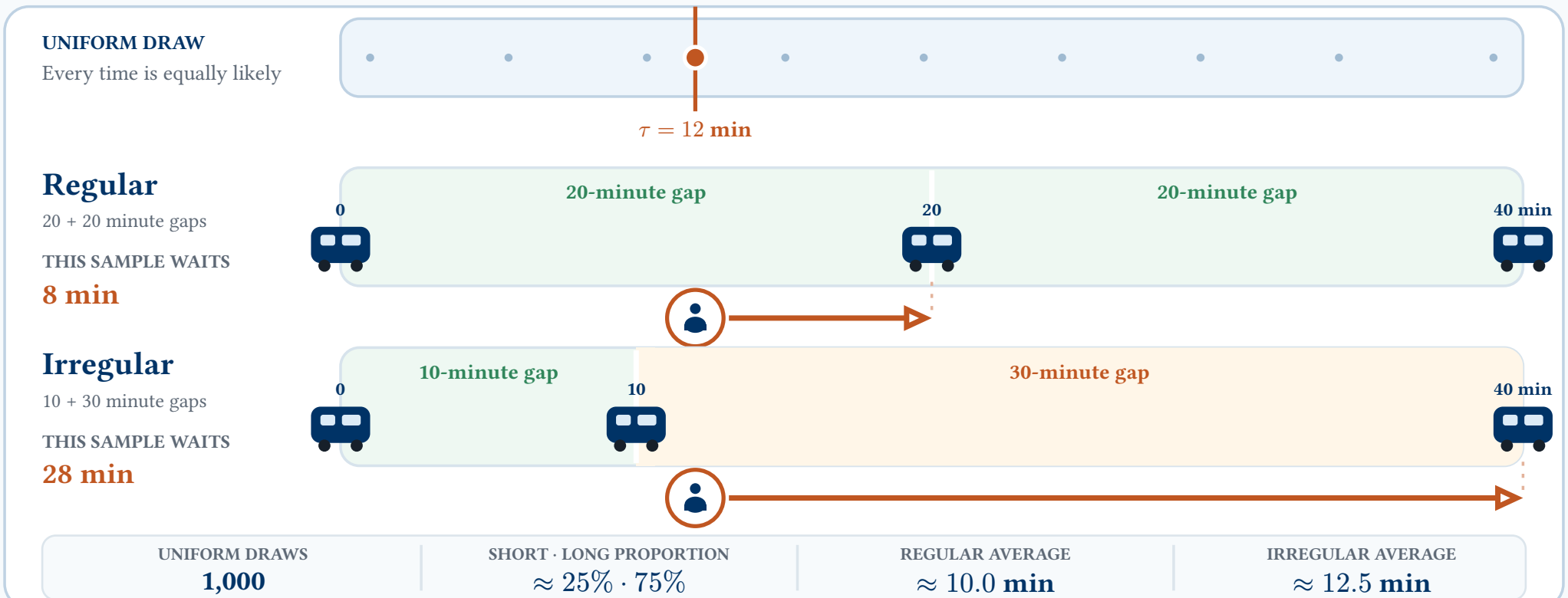
$$\frac{10+30}{2} = 20 \text{ minutes.}$$

Predict **Poll 1.3**

In the 10/30-minute pattern, is the average passenger wait still 10 minutes?

A). Less than 10 minutes.
B). Exactly 10 minutes.
C). Greater than 10 minutes.
D). The full pattern is still insufficient.

A Random Passenger Samples Clock Time



Bus Paradox

Over one 40-minute cycle:

Short gap

- occupies 10 of the 40 clock minutes;
- encountered with probability $\frac{10}{40} = \frac{1}{4}$;
- conditional average remaining wait is 5 minutes.

Long gap

- occupies 30 of the 40 clock minutes;
- encountered with probability $\frac{30}{40} = \frac{3}{4}$;
- conditional average remaining wait is 15 minutes.

$$\mathbb{E}[W] = \frac{1}{4} \times 5 + \frac{3}{4} \times 15 = 12.5 \text{ minutes.}$$

Reveal

The mean bus gap is 20 minutes, but the mean passenger wait is **12.5 minutes** $>$ 10 minutes.

Why the Bus Answer Changes

You are sampling clock time, not sampling bus gaps

Long gaps occupy more time on the clock. A passenger who arrives at a random time is therefore more likely to arrive during a long gap.

Length-biased sampling **Definition 1.1**

When observations are made at a random time, long-lasting intervals are more likely to be observed than short-lasting intervals.

Transfer the mechanism

There are two classes: one has 10 students and one has 90 students.

- Average class size across the two courses: $\frac{10+90}{2} = 50$.
- What class size is experienced by a randomly selected student?

$$\frac{10 \times 10 + 90 \times 90}{100} = 82.$$

Conclusion

A random observer over-samples large classes for the same reason a random passenger over-samples long bus gaps.

1.4.2. Amplification through Dynamics: Phantom Traffic

Prediction (multiple choice) **Poll 1.4**

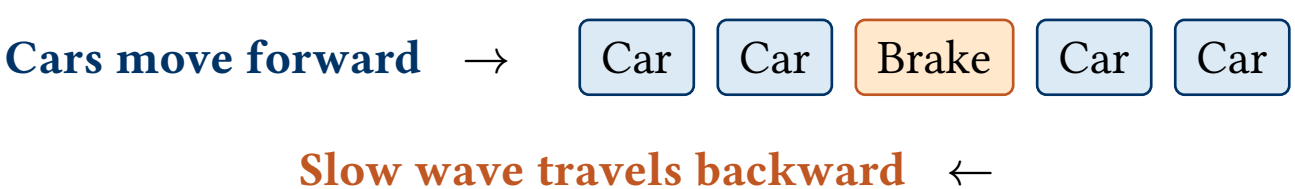
With no accident and no fixed bottleneck, how can a straffic jam form and persist?

A). Any single hard brake creates a permanent jam, regardless of traffic density.

B). A slow vehicle creates a fixed blockage at one location.

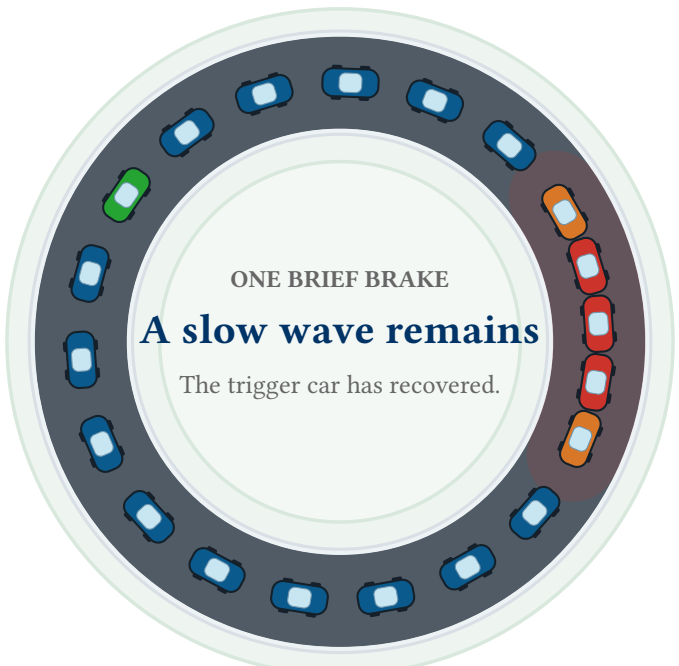
C). At high density, small speed differences and delayed reactions can amplify into a backward-moving wave.

D). The average speed is below the speed limit.



Random braking can create a traffic jam.

CARS MOVE CLOCKWISE



SLOWDOWN MOVES BACKWARD

Phantom Traffic: The Mechanism

1. Perturbation

One driver slows slightly.

2. Delayed reaction

The following driver reacts later and slows more.

3. Propagation

The disturbance travels backward through the traffic stream.

4. Persistence

At high density there is too little spacing to absorb the disturbance.

Slack is unused capacity or space that allows a system to absorb a disturbance.

Utilization is the fraction of available capacity used on average.



Conclusion

Near capacity, small disturbances can be amplified rather than absorbed.

Resilience Keeps Random Shocks from Cascading

The same amplification mechanism appears in global supply chains.

COVID-19: correlated global shock

Lockdowns, trade restrictions, and demand surges hit production and transport together. World merchandise trade fell **15% year-on-year** in 2020 Q2.

Strait of Hormuz: chokepoint shock

The 2026 disruption exposed the concentration: about **25% of seaborne oil trade** crossed this narrow route, and the shock spread into energy and shipping costs.



Conclusion

Resilience does not eliminate random shocks. It limits propagation and speeds recovery through buffers, alternative suppliers and routes, flexible capacity, and visibility.

Sources: WTO (2021) | UNCTAD (2026)

1.4.3. Tail Risk and Decisions: Pop Mart Inventory

Two demand worlds have the same mean demand, $\mathbb{E}[D] = 10$.

Stable world

$D = 8$ with probability 0.25;

$D = 10$ with probability 0.50;

$D = 12$ with probability 0.25.

Hype world

$D = 5$ with probability 0.75;

$D = 25$ with probability 0.25.

Most releases are modest, but an occasional viral surge is large.

Team decision **Poll 1.5**

In both worlds, choose between the same two inventory levels: $Q \in \{10, 15\}$.

- Every unsold unit costs 1 dollar.
- Every unit of unmet demand costs 4 dollars.

Which level minimizes expected loss in each world?

Pop Mart: Same Options, Different Decisions

$$\text{Loss}(Q, D) = (Q - D)^+ + 4(D - Q)^+$$

Stable world

Expected loss at $Q = 10$:

$$0.25(10 - 8) + 0.25 \times 4(12 - 10) = 2.5$$

Expected loss at $Q = 15$:

$$0.25(15 - 8) + 0.50(15 - 10) + 0.25(15 - 12) = 5.0$$

Choose $Q = 10$: $2.5 < 5.0$.

Hype world

Expected loss at $Q = 10$:

$$0.75(10 - 5) + 0.25 \times 4(25 - 10) = 18.75$$

Expected loss at $Q = 15$:

$$0.75(15 - 5) + 0.25 \times 4(25 - 15) = 17.5$$

Choose $Q = 15$: $17.5 < 18.75$.

Why the Same Mean Leads to Different Decisions

Why the choices differ

Both worlds have the same mean. The stable world favors $Q = 10$ because extra stock mostly becomes leftovers. The hype world favors $Q = 15$ because an occasional demand surge creates a large, expensive shortage when stock is too low.

Newsvendor problem

Definition 1.2

A decision maker commits inventory before uncertain demand is revealed and balances the costs of overage and shortage.

Conclusion

The best inventory level depends on the full demand distribution, not only its average.

1.4.4. Three Cases, Three Mechanisms

Case	Why the mean is insufficient	Mechanism	Later topic
Bus waiting	Long intervals are observed more often.	Sampling bias in time.	Renewal and Poisson processes.
Phantom traffic	Disturbances propagate through interactions.	Dynamic amplification near capacity.	Queueing and stability.
Pop Mart	Rare surges and asymmetric losses drive decisions.	Tail risk and decision consequences.	Inventory and MDPs.

✅ Central lesson

Operational performance depends on what can happen, how likely different outcomes are, and how the system transforms those outcomes over time.

1.5. From Random Quantities to Random Evolution

We have already used the ideas informally. We now give names to the mathematical objects.

Stochastic model — Definition 1.3

A **stochastic model** specifies:

- the uncertain inputs or events;
- the state of the system;
- how the state changes;
- the possible performance outcomes and their likelihoods.

A modeling choice, not a requirement of realism

A stochastic model is useful when unresolved variability materially affects the question being studied. Deterministic models (those in **IEDA 3010**) is still useful approximations.

1.5.1. Random Variable and Realization

During minute 1, an uncertain number of customers arrives. The queue then evolves minute by minute as arrivals add work and service completions remove work.

Let

$$A_1 = \text{number of customers arriving during minute 1.}$$

Before the minute

A_1 is uncertain and may take several possible values.

After observation

We observe one realized value, for example $A_1 = 3$.

Random variable **Definition 1.4**

A **random variable** is a numerical quantity whose value depends on an uncertain outcome.

Do not confuse the object with its realization: A_1 denotes the uncertain quantity; $A_1 = 3$ is one observed outcome.

1.5.2. Distribution

Distribution — Definition 1.5

The **distribution** describes which values of a random variable are possible and how likely they are.

a	0	1	2	3	4
$\mathbb{P}(A_1 = a)$	0.10	0.25	0.35	0.20	0.10

Idea

The random variable tells us **what is uncertain**. The distribution tells us **how the uncertainty is allocated across possible values**.

Discrete variables use probability masses. Continuous variables are often described using densities. Cumulative probabilities are summarized by a CDF.

Four Different Summaries of a Distribution

Mean

Probability-weighted center:

$$\mathbb{E}[X] = \sum_x x\mathbb{P}(X = x).$$

Standard deviation

Typical scale of fluctuation around the mean:

$$\sigma = \sqrt{\mathbb{E}[(X - \mathbb{E}[X])^2]}.$$

It has the same units as X.

Tail probability

Chance of exceeding a bad threshold:

$$\mathbb{P}(\text{delivery time} > 40 \ \ \text{min}).$$

Quantile

Example: the **95th percentile (p95)** is a threshold that approximately 95% of outcomes do not exceed.

⚠ One number cannot describe an entire distribution

Equal means do not imply equal spread, tails, or decision consequences.

A Preview: Coefficient of Variation

Suppose X has positive mean μ and standard deviation σ.

Coefficient of variation **Definition 1.6**

$$\mathrm{CV}(X) = \frac{\sigma}{\mu}.$$

It measures fluctuation **relative to the mean scale**. Because the units cancel, CV is dimensionless.

Same relative variability **Example 1.4**

- Mean 10 minutes, standard deviation 2 minutes: $\mathrm{CV} = 0.2$.
- Mean 100 minutes, standard deviation 20 minutes: $\mathrm{CV} = 0.2$.

Both fluctuate by about 20% of their mean scale.

Bus connection—a preview, not yet a theorem

Under a stationary renewal model, if L is a bus-gap length and W is the remaining passenger wait,

$$\mathbb{E}[W] = \frac{\mathbb{E}[L]}{2} \times (1 + \mathrm{CV}_L^2).$$

Regular gaps have $\mathrm{CV}_L = 0$; exponential gaps have $\mathrm{CV}_L = 1$.

1.5.3. Stochastic Process and Sample Path

Let $Q_0, Q_1, Q_2, \ldots$, where Q_n is the number of customers waiting after minute n.

Stochastic process

The complete time-indexed collection of uncertain queue lengths.

Sample path

One realized sequence/observed, for example

$$0, 2, 5, 3, 1, 0, \ldots$$

Stochastic process **Definition 1.7**

A **stochastic process** is a collection of random variables indexed by time or another ordered index.

Idea

Variables in a stochastic process may be independent or dependent. The process notation records their order and joint evolution.

State and Update Rule

How does the queue evolve from one minute to the next?

$$\textbf{next queue} = \textbf{current queue} - \textbf{completed service} + \textbf{new arrivals}$$

$$Q_{n+1} = \max(Q_n - S_{n+1}, 0) + A_{n+1}.$$

- Q_n: current queue length.
- S_{n+1}: number that can complete service.
- A_{n+1}: new arrivals.
- The maximum prevents a negative queue.

State **Definition 1.8**

The **state** is the information retained about the current system that is sufficient, under the model, to specify what may happen next.

Does alternative states lead to full information?

- Number of customer completed service? Number of customers that have arrived?

What Stochastic Processes Look Like

Random walk

$0 \to 1 \to 0 \to 1 \to 2 \to 1$

State moves up or down at discrete times.

Counting process

$0 \to 0 \to 1 \to 1 \to 2 \to 3$

The path increases when events occur.

Queue length

$0 \to 2 \to 5 \to 3 \to 1 \to 0$

Arrivals raise the path; service lowers it.

✅ This is what the course studies

Random evolution over time: where the system goes, what it hits, how long events take, what happens in the long run, and how decisions should respond.

1.6. What This Course Will Teach

Engineering question	Main topic
How does a state change from one step to the next?	Random walks and DTMCs
Will the system reach a target or failure state?	Hitting times and martingale ideas
How many events occur by time t?	Counting and Poisson processes
How long until the next event?	Interarrival and renewal models
How does a system jump in continuous time?	CTMCs
How does limited capacity create congestion and waiting?	Queueing models
When and how much should we order?	Inventory models
How should actions depend on the current state?	MDPs
How can a model be explored and checked?	Simulation and model criticism



Conclusion

Queueing and inventory models connect stochastic processes to decisions.

1.7. How We Will Learn

Observe → Predict → Model → Analyze → Check → Interpret

Mathematical support

Prerequisite tools will be refreshed when first needed: conditional probability, expectation and variance, geometric series, matrices, derivatives, integrals, and exponential functions.

LLM-assisted verification

Use the protocol: **Predict → Ask → Simulate → Verify → Explain.**

Verify with a calculation, simulation, boundary case, units check, or known special case.

Pause and recall

Modeling random systems

---

1. **Choose the measure.** Delivery A always takes 20 minutes; B takes 5 or 35 minutes, equally likely. Both average 20. Which better meets a 30-minute promise? What probability matters?
2. **Build an update.** A shop starts with X_n units. Demand D_{n+1} occurs before r units arrive at night; unmet demand is lost. Give the next stock level and a measure of shortage risk.
3. **Change the sampling.** A server runs 90 one-minute jobs and 10 nine-minute jobs, with no idle time. At a uniformly random time during this run, which type are you more likely to see?

For each question, identify what is observed, what changes, or what is measured.

Answers and reasoning

Modeling random systems

---

1. **Choose the measure.** Compare $\mathbb{P}(\text{delivery time} > 30)$. It is 0 for A and $\frac{1}{2}$ for B, so A better meets the promise. The mean hides the risk of being late.
2. **Build an update.** $X_{n+1} = \max(X_n - D_{n+1}, 0) + r$. The maximum accounts for lost sales; the delivery is added afterward. One risk measure is $\mathbb{P}(D_{n+1} > X_n)$.
3. **Change the sampling.** Equally likely: $90 \times 1 = 10 \times 9 = 90$ minutes for each type. A uniformly chosen job is short with probability 0.9, but sampling clock time gives longer jobs more weight.

The model must match the event order, the observation method, and the decision.

2. Markov Chains

2.1. From Coin Tosses to One-Step Forecasts

2.1.1. Paths, States, and Prediction

A queue length. An inventory level. A machine that is Up or Down.

We have described systems by their **state** and watched a **sample path** unfold. Now we ask a prediction question:

What information do we need now
to forecast the next step?



Takeaway

Our goal is not to guess one exact future. It is to describe the possible next states and their probabilities.

A game we can run

A token starts at 0; negative positions are allowed. Toss an **independent fair coin** each round.

Heads: move one place right. Tails: move one place left.

Two heads in a row — Poll 2.1

The first two tosses were heads. On the next toss, how does the chance of tails compare with heads?

A). Tails is more likely: it is "due."
B). Heads and tails are equally likely: the tosses are independent and fair.
C). Tails is less likely: heads is on a streak.

What assumption decides your answer?

Takeaway

Independence makes past tosses irrelevant; **fairness** gives H and T probability $\frac{1}{2}$ each. The same transition rule at every round gives **time-homogeneity**.

One run: what should we record?

The model

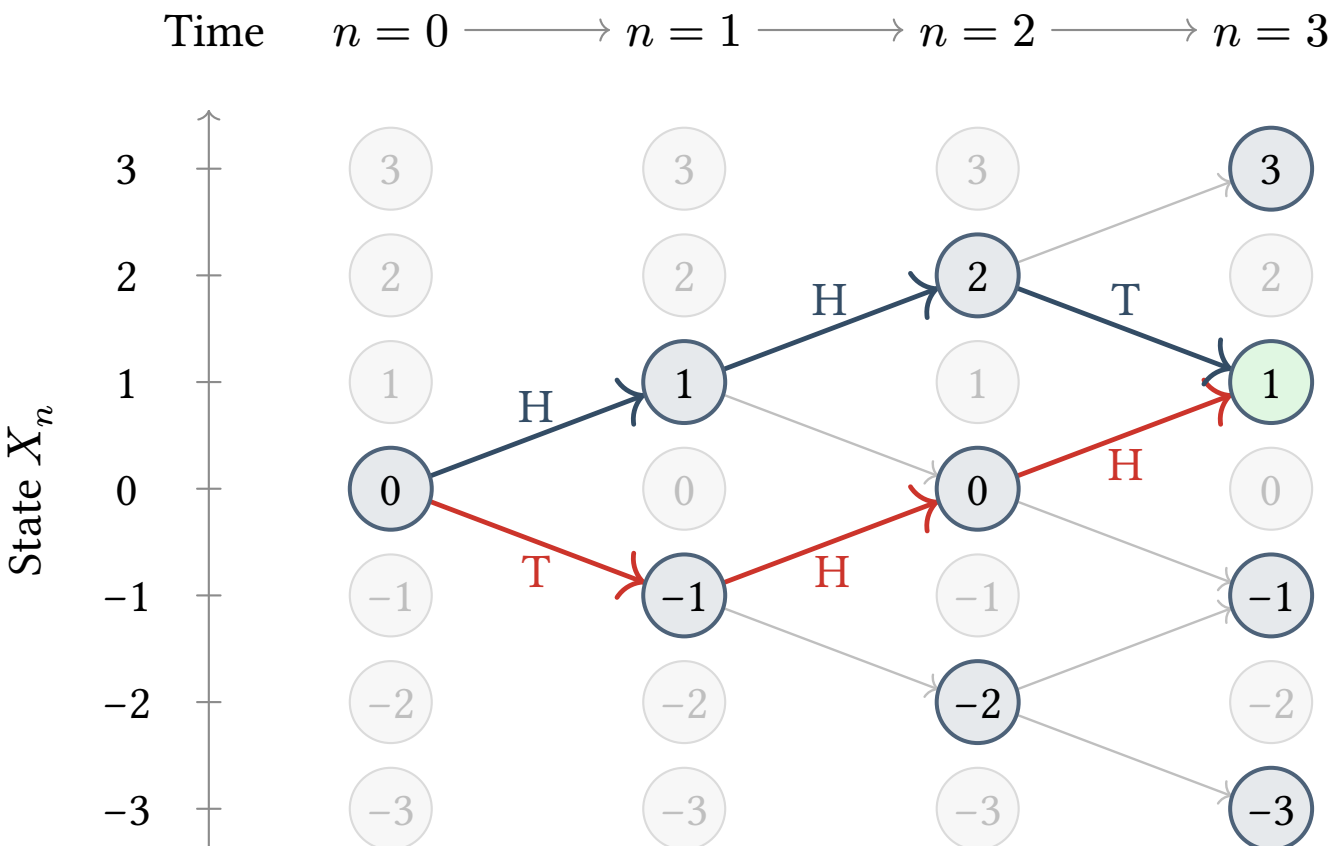
- **State**: X_n, the position after n tosses.
- **Initial state**: $X_0 = 0$.
- **State space**: $\mathcal{S} = \mathbb{Z}$, all integer positions.
- **Randomness**: independent tosses give an increment $\xi_n = +1$ (H) or -1 (T), each with probability $\frac{1}{2}$.
- **Transitions**: $X_{n+1} = X_n + \xi_{n+1}$.

One realization

- **Observed tosses**: H, H, T, T, H.
- **Sample path**: the sequence of realized states, $(X_0, \ldots, X_5) = (0, 1, 2, 1, 0, 1)$.
- **Current state**: $X_5 = 1$.
- **Last increment**: $\xi_5 = X_5 - X_4 = +1$.

Different histories, the same present

Both walks use independent fair tosses. Both have completed three rounds.



Run A: HHT

Run B: THH

Gray states cannot be reached from $X_0 = 0$ at that time.

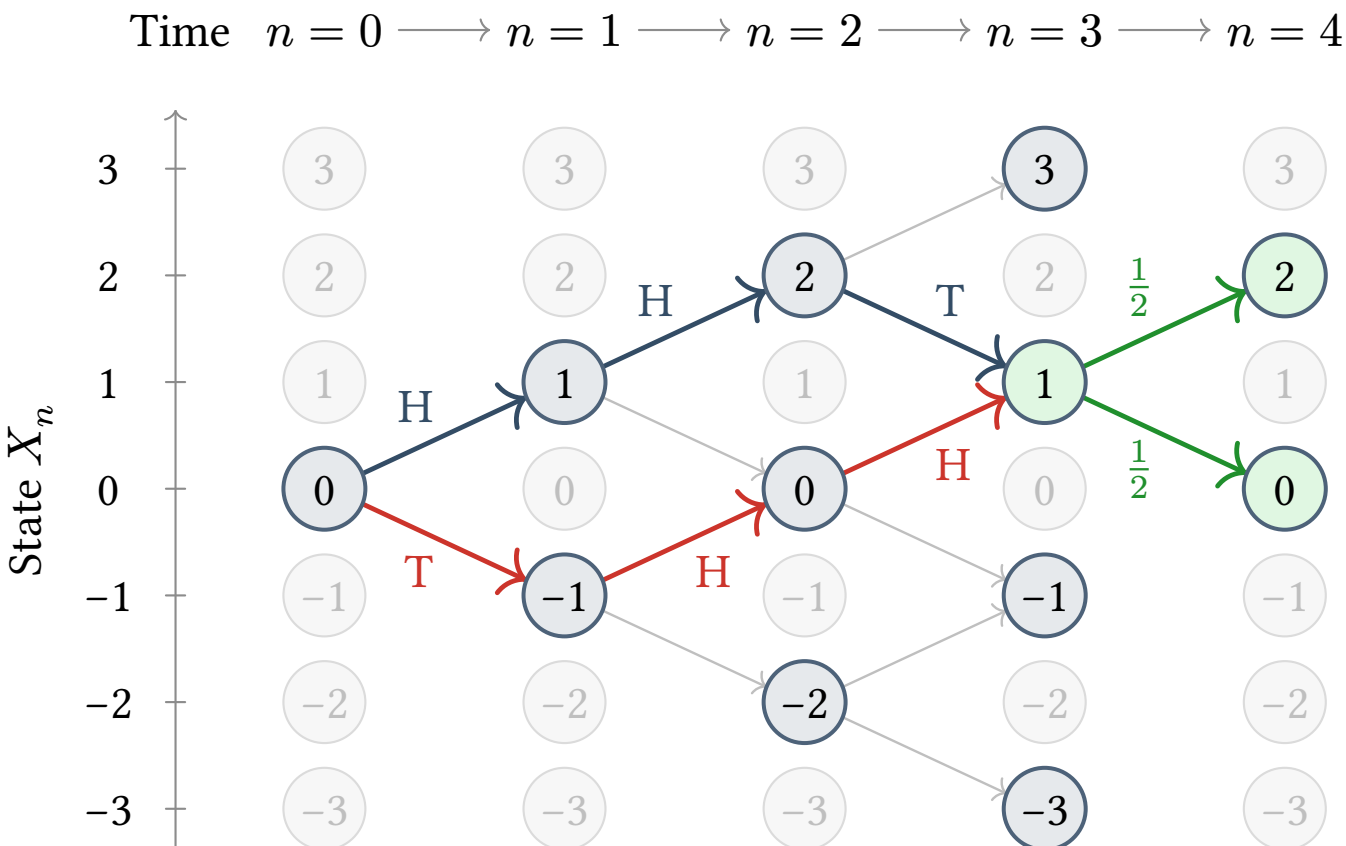
? Predict and explain

Which is more likely to be at 2 next: A, B, or neither?

Explain before calculating.

The next-step forecasts agree

Once we know that the position is 1, the earlier route does not change the forecast.



Run A: HHT

Run B: THH

At $n = 4$, gray states are impossible given $X_3 = 1$.

Takeaway

The same distribution does not mean the same outcome. A could next move to 2 while B moves to 0.

2.1.2. Independence and State Probabilities

X_n is the position after n tosses; ξ_{n+1} is the next increment.

Simple random walk **Definition 2.1**

Start at $X_0 = 0$ and update by

$$X_{n+1} = X_n + \xi_{n+1}. \tag{2.1}$$

The increments $\xi_1, \xi_2, \ldots$ are independent and identically distributed, with

$$\mathbb{P}(\xi_n = +1) = p, \quad \mathbb{P}(\xi_n = -1) = q = 1 - p.$$

Our fair-toss example has $p = \frac{1}{2}$. The state space is $\mathcal{S} = \mathbb{Z}$.

Probability refresher: Conditional probability

$\mathbb{P}(A \mid B)$ is the probability of A **given that B has occurred**. Restrict attention to B, then rescale the remaining probabilities to sum to 1.

$$\mathbb{P}(A \mid B) = \frac{\mathbb{P}(A \cap B)}{\mathbb{P}(B)}, \quad \mathbb{P}(B) > 0.$$

For two independent fair tosses, let A be "the second toss is H" and B be "the first toss is H."

Before knowing B

$$\Omega = \{\mathrm{HH}, \mathrm{HT}, \mathrm{TH}, \mathrm{TT}\}.$$

Each outcome has probability $\frac{1}{4}$.

After knowing B

$$B = \{\mathrm{HH}, \mathrm{HT}\}.$$

Each remaining outcome has probability $\frac{1}{2}$.

Only HH satisfies both A and B, so

$$\mathbb{P}(A \mid B) = \frac{1/4}{1/2} = \frac{1}{2}.$$

Probability refresher: Independence

Events: knowing that B occurred does not change the probability of A.

$$\mathbb{P}(A \cap B) = \mathbb{P}(A)\mathbb{P}(B).$$

Equivalently, $\mathbb{P}(A \mid B) = \mathbb{P}(A)$ when $\mathbb{P}(B) > 0$.

Discrete random variables: knowing Y does not change the **distribution** of X.

$$\mathbb{P}(X = x, Y = y) = \mathbb{P}(X = x)\mathbb{P}(Y = y) \quad \text{for all } x, y.$$

Equivalently, for every y with $\mathbb{P}(Y = y) > 0$,

$$\mathbb{P}(X = x \mid Y = y) = \mathbb{P}(X = x) \quad \text{for all } x.$$

Do independent tosses give independent positions?

? Predict and explain

The tosses are independent and fair. Starting from $X_0 = 0$, are X_1 and X_2 independent too?

Given $X_1 = 1$: $X_2 = 1 + \xi_2$, so only the second toss must be heads.

$$\mathbb{P}(X_2 = 2 \mid X_1 = 1) = \mathbb{P}(\xi_2 = 1 \mid \xi_1 = 1) = \mathbb{P}(\xi_2 = 1) = \frac{1}{2}.$$

Without conditioning: $X_2 = \xi_1 + \xi_2$, so reaching 2 requires **two independent heads**.

$$\mathbb{P}(X_2 = 2) = \mathbb{P}(\xi_1 = 1, \xi_2 = 1) = \mathbb{P}(\xi_1 = 1)\mathbb{P}(\xi_2 = 1) = \frac{1}{2} \times \frac{1}{2} = \frac{1}{4}.$$

Takeaway

No. The increments are independent, but the positions share ξ_1 and are **dependent**.

State distributions at $n = 1$ and $n = 2$

Start at $X_0 = 0$. The tosses are independent and fair.

One toss: $n = 1$

H and T each have probability $\frac{1}{2}$.

j	Toss	$\mathbb{P}(X_1 = j)$
-1	T	$\frac{1}{2}$
1	H	$\frac{1}{2}$

Thus $\mathbb{P}(X_1 = -1) = \mathbb{P}(X_1 = 1) = \frac{1}{2}$.

Two tosses: $n = 2$

Each pair has probability $\frac{1}{2} \times \frac{1}{2} = \frac{1}{4}$.

j	Tosses	$\mathbb{P}(X_2 = j)$
-2	TT	$\frac{1}{4}$
0	HT, TH	$2 \times \frac{1}{4} = \frac{1}{2}$
2	HH	$\frac{1}{4}$

Two disjoint routes end at 0:

$$\begin{aligned}\mathbb{P}(X_2 = 0) &= \mathbb{P}(\mathrm{HT}) + \mathbb{P}(\mathrm{TH}) \\ &= \frac{1}{4} + \frac{1}{4} = \frac{1}{2}.\end{aligned}$$

Unlisted states have probability 0. Add the probabilities of all routes ending at the same state.

State distribution at $n = 3$

There are eight three-toss sequences. Each has probability $\left(\frac{1}{2}\right)^3 = \frac{1}{8}$.

j	Toss sequences ending at j	$\mathbb{P}(X_3 = j)$
-3	TTT	$\frac{1}{8}$
-1	HTT, THT, TTH	$3 \times \frac{1}{8} = \frac{3}{8}$
1	HHT, HTH, THH	$3 \times \frac{1}{8} = \frac{3}{8}$
3	HHH	$\frac{1}{8}$

Check using $n = 2$: reach 1 from 0 with H, or from 2 with T. The third toss is independent of X_2, so

$$\begin{aligned}\mathbb{P}(X_3 = 1) &= \mathbb{P}(X_2 = 0) \times \frac{1}{2} + \mathbb{P}(X_2 = 2) \times \frac{1}{2} \\ &= \frac{1}{2} \times \frac{1}{2} + \frac{1}{4} \times \frac{1}{2} = \frac{3}{8}.\end{aligned}$$

Unlisted states have probability 0; $\frac{1}{8} + \frac{3}{8} + \frac{3}{8} + \frac{1}{8} = 1$.

2.2. Discrete-Time Markov Chains

2.2.1. The Markov Property and Time Homogeneity

Once the current state is known,
the earlier history does not change
the next-state distribution.

A state can contain memory. The question is whether it contains **enough** predictive information.

> **? Predict and explain**
>
> In the fair walk, would learning the previous toss change the next-position forecast once the current position is known?

A definition we can now read

Discrete-time Markov chain (DTMC) **Definition 2.2**

A discrete-time process $(X_n)_{n\geq 0}$ on a finite or countable state space $\mathcal{S}$ is a **Markov chain** if

$$\mathbb{P}(X_{n+1} = j \mid X_0 = i_0, \ldots, X_{n-1} = i_{n-1}, X_n = i) = \mathbb{P}(X_{n+1} = j \mid X_n = i). \tag{2.2}$$

The equality applies to histories with positive probability.

The left side uses **the whole history**. The right side uses **only the present**.



Takeaway

This equation states exactly what the two-history experiment illustrated.

Symmetric random walk: Markov property

Example 2.1

- **State**: $X_n \in \mathcal{S} = \mathbb{Z}$, starting from $X_0 = 0$.
- **Randomness**: independent increments with $\mathbb{P}(\xi_k = 1) = \mathbb{P}(\xi_k = -1) = \frac{1}{2}$.
- **Dynamics**: $X_{n+1} = X_n + \xi_{n+1}$.

The history $X_0, \ldots, X_n$ depends only on $\xi_1, \ldots, \xi_n$, so ξ_{n+1} is **independent of the whole history**.

For any history $X_0 = i_0, \ldots, X_n = i$ with positive probability,

$$\begin{aligned}
&\mathbb{P}(X_{n+1} = j \mid X_0 = i_0, \ldots, X_n = i) \\
&= \mathbb{P}(\xi_{n+1} = j - i \mid X_0 = i_0, \ldots, X_n = i) \\
&= \mathbb{P}(\xi_{n+1} = j - i) \\
&= \mathbb{P}(X_{n+1} = j \mid X_n = i).
\end{aligned}$$

This verifies (2.2): the next state is $i - 1$ or $i + 1$, each with probability $\frac{1}{2}$, depending only on the present state $X_n = i$.

Does the rule also stay the same over time?

The Markov property in (2.2) removes dependence on earlier history; the next-step law may still depend on n.

Time-homogeneous DTMC — Definition 2.3

A DTMC is **time-homogeneous** if its one-step transition probabilities

$$P_{ij} = \mathbb{P}(X_{n+1} = j \mid X_n = i) \tag{2.3}$$

do not depend on n. We collect them into the transition matrix $P = (P_{ij})$.



Takeaway

With a time-homogeneous DTMC, the same transition rule applies at every step. Hence, it suffices to specify the one-step transition probabilities P_{ij} for all $i, j \in \mathcal{S}$.

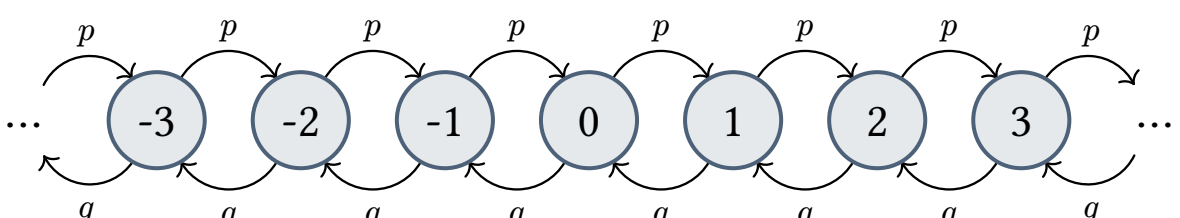
Random walk as a DTMC

Example 2.2

- **State space**: $\mathcal{S} = \mathbb{Z}$.
- **Randomness**: walk right/up with probability p or left/down with probability q.
- **Transitions**: from state i,
  - go to $i+1$ with probability p,
  - go to $i-1$ with probability $q \overset{\text{def}}{=} 1-p$.

- **Transition matrix**:

$$P = \begin{array}{c} \\ -2 \\ -1 \\ 0 \\ 1 \\ 2 \end{array} \begin{array}{c} \begin{array}{ccccccc} -3 & -2 & -1 & 0 & 1 & 2 & 3 \end{array} \\ \begin{bmatrix} \ddots & \ddots & \ddots & \ddots & \ddots & & \\ \ddots & q & 0 & p & 0 & 0 & \\ \ddots & 0 & q & 0 & p & 0 & \ddots \\ & 0 & 0 & q & 0 & p & \ddots \\ & & \ddots & \ddots & \ddots & \ddots & \ddots \end{bmatrix} \end{array}$$



Machine up/down as a DTMC

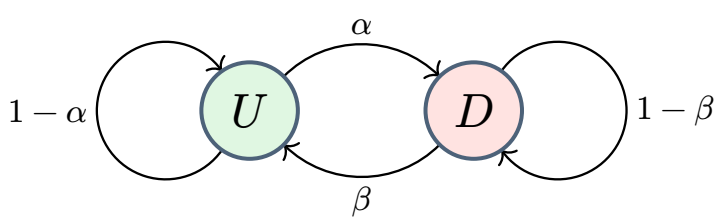
Example 2.3

- **State**: $X_n \in \{U, D\}$ = machine is **Up** / **Down** at day n.
- **Randomness**: If the machine is up, it may fail with probability α; if the machine is down, repairs are performed with probability β and restores it.
- **Transitions** (time-homogeneous):
  - $U \to D$ w.p. α (failure),
  - $D \to U$ w.p. β (repair),
  - otherwise it stays.

- **Transition matrix**:

$$P = \begin{array}{c} \\ U \\ D \end{array} \begin{array}{c} \begin{array}{cc} U & \quad D \end{array} \\ \begin{bmatrix} 1-\alpha & \alpha \\ \beta & 1-\beta \end{bmatrix} \end{array}$$

- **Transition diagram:**



Remarks:

- A state can have transitions to itself (e.g., machine stays up or down).
- **Markov assumption:** once today's status is known, earlier history is independent of tomorrow.
- **Time-homogeneity:** these same daily probabilities apply at every inspection.

Tomorrow's forecast is one row of the transition matrix

? One prediction problem

The machine is **Up on day 0**. What is the chance that it is **Down on day 1**? What about **day 2**?

Suppose the daily rules give the following transition matrix P. Rows specify **today** and columns specify **tomorrow**.

Today / tomorrow	U	D
U	0.8	0.2
D	0.4	0.6

Given Up today: use row U. Tomorrow is Up with chance 0.8 and Down with chance 0.2.

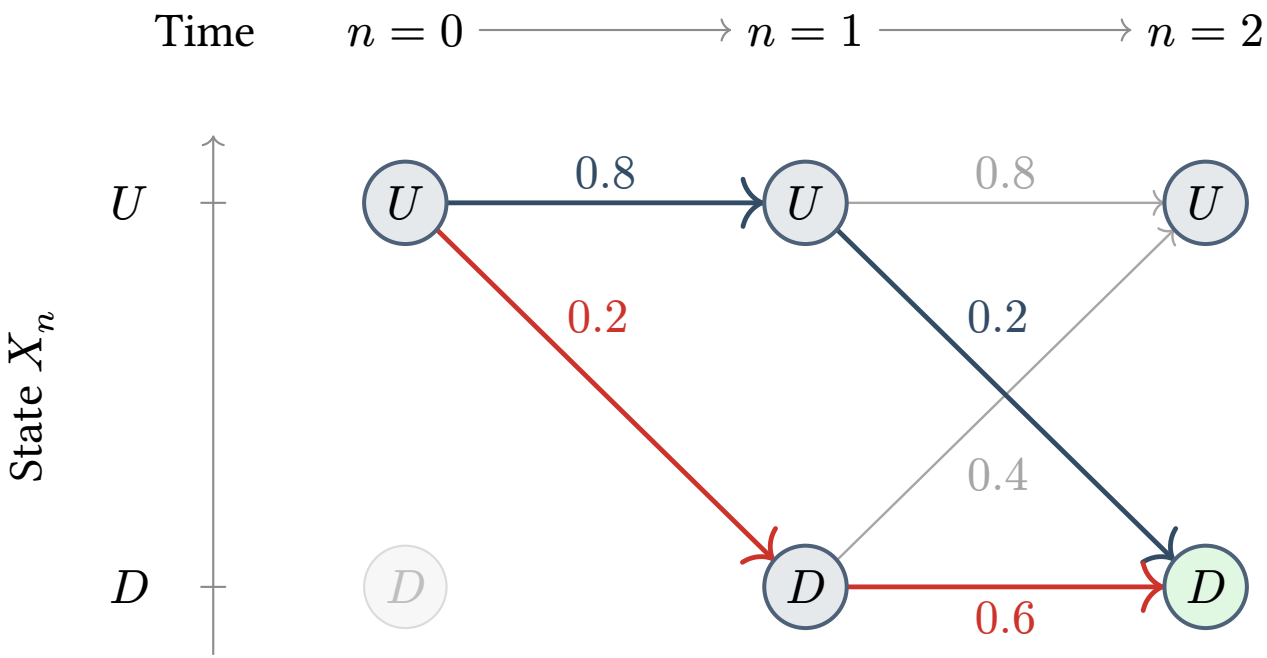
Given Down today: use row D. The chance of Up tomorrow is $P_{DU} = 0.4$.

Stochastic matrix **Definition 2.4**

A row-stochastic matrix is square, has nonnegative entries, and has each row summing to 1. Each row lists all mutually exclusive possibilities for the next state.

Two days ahead: two ways to be Down

Starting Up on day 0, the machine can be **Up or Down on day 1**. Both cases can lead to Down on day 2.



Via U: $U \to U \to D$

$$0.8 \times 0.2 = 0.16.$$

Via D: $U \to D \to D$

$$0.2 \times 0.6 = 0.12.$$

Total chance of Down:

$$0.16 + 0.12 = \mathbf{0.28}.$$

Probability refresher

1. Multiplication rule: both events occur (a logical "and")

$$\mathbb{P}(A \cap B) = \mathbb{P}(A)\mathbb{P}(B \mid A), \quad \mathbb{P}(A) > 0.$$

Multiply the chance of A by the chance of B **given** A.

2. Addition rule: disjoint alternatives (a logical "or")

$$\mathbb{P}(A \cup B) = \mathbb{P}(A) + \mathbb{P}(B), \quad A \cap B = \varnothing.$$

The events cannot both occur, so no outcome is counted twice.

3. Law of total probability: all possible cases

The B_k are disjoint, cover all outcomes, and have positive probability.

$$\mathbb{P}(A) = \sum_{k=1}^{m} \mathbb{P}(A \cap B_k) = \sum_{k=1}^{m} \mathbb{P}(B_k)\mathbb{P}(A \mid B_k).$$

Multiply within each case, then add across cases.

Applying the probability rules to the machine

Assume $X_0 = U$ throughout. To end Down on day 2, the machine passes through either U or D on day 1.

Machine path via U:

$$\begin{aligned}\mathbb{P}(X_1 = U, X_2 = D) &= \mathbb{P}(X_1 = U)\mathbb{P}(X_2 = D \mid X_1 = U) \\ &= 0.8 \times 0.2 = 0.16.\end{aligned}$$

20% of the 80% that are Up on day 1 go Down next: 16% of all runs.

Machine paths via U or via D:

$$\begin{aligned}\mathbb{P}(X_2 = D) &= \mathbb{P}(X_1 = U)\mathbb{P}(X_2 = D \mid X_1 = U) + \mathbb{P}(X_1 = D)\mathbb{P}(X_2 = D \mid X_1 = D) \\ &= 0.8 \times 0.2 + 0.2 \times 0.6 = 0.28.\end{aligned}$$

The day-1 states U and D are disjoint and exhaustive. This is the law of total probability with two cases.

Recording the full forecast

Starting Up, the day-2 Up probability is $1 - 0.28 = 0.72$. Both forecast columns use state order (U, D):

Time	Forecast, knowing $X_0 = U$	Forecast, knowing $X_0 = D$
Day 0	$\boldsymbol{\nu}^{(0)} = (1, 0)$	$\boldsymbol{\nu}^{(0)} = (0, 1)$
Day 1	$\boldsymbol{\nu}^{(1)} = (0.8, 0.2)$	$\boldsymbol{\nu}^{(1)} = (0.4, 0.6)$
Day 2	$\boldsymbol{\nu}^{(2)} = (0.72, 0.28)$	$\boldsymbol{\nu}^{(2)} = (0.56, 0.44)$
Day 3	$\boldsymbol{\nu}^{(3)} = (0.688, 0.312)$	$\boldsymbol{\nu}^{(3)} = (0.624, 0.376)$
⋮	⋮	⋮
Day 100	$\boldsymbol{\nu}^{(100)} =?$	$\boldsymbol{\nu}^{(100)} =?$

State probability vector **Definition 2.5**

The row vector $\boldsymbol{\nu}^{(n)}$ lists $\nu_j^{(n)} = \mathbb{P}(X_n = j)$ for each state j. Its entries are nonnegative and sum to 1. Here, (n) denotes the time and j the state.

In one run, X_2 will be **U or D**. Starting Up, $(0.72, 0.28)$ gives the chances of these outcomes.

What information does the forecast use?

? Down on day 2: three information scenarios

The machine is **Up on day 0**. In each case below, we want the probability that it is **Down on day 2**.

1. **No day-1 report.** We have not observed the machine on day 1.
2. **Day 1 observed Up.** The inspection reports that the machine is Up.
3. **Day 1 observed Down.** The inspection reports that the machine is Down.

For each case, **write the conditional probability**, then find its value in the forecast table. Which starting state, how many steps ahead, and which entry of the vector do we need?

The conditioning specifies the information

The future event is always $X_2 = D$. The **information after the conditioning bar** changes.

1. No day-1 report: consider all runs that start Up; either day-1 state is possible.

$$\mathbb{P}(X_2 = D \mid X_0 = U).$$

2. Day 1 observed Up: among those runs, consider only the ones Up on day 1.

$$\mathbb{P}(X_2 = D \mid X_0 = U, X_1 = U) = \mathbb{P}(X_2 = D \mid X_1 = U).$$

3. Day 1 observed Down: among those runs, consider only the ones Down on day 1.

$$\mathbb{P}(X_2 = D \mid X_0 = U, X_1 = D) = \mathbb{P}(X_2 = D \mid X_1 = D).$$

> 💡 **Why can we omit the day-0 state in the last two cases?**
>
> By the **Markov property**, once X_1 is known, X_0 adds no information about X_2. Without a day-1 observation, we still use the information $X_0 = U$.

Reading the answers from the forecast table

The relevant rows are shown below. In state order (U, D), the **second entry** gives the Down probability.

Table row	Starting state U	Starting state D
Day 1: one step	$(0.8, \mathbf{0.20})$	$(0.4, \mathbf{0.60})$
Day 2: two steps	$(0.72, \mathbf{0.28})$	$(0.56, 0.44)$

No day-1 report: two steps from U. $\mathbb{P}(X_2 = D \mid X_0 = U) = \mathbf{0.28}$

Day 1 observed Up: one step from U. $\mathbb{P}(X_2 = D \mid X_1 = U) = \mathbf{0.20}$

Day 1 observed Down: one step from D. $\mathbb{P}(X_2 = D \mid X_1 = D) = \mathbf{0.60}$

Why use the Day 1 row for a Day 2 question?

The row counts **steps after the known state**. Once day 1 is observed, only **one step** remains. The transitions are **time-homogeneous**, so the same one-step forecast applies from day 1 to day 2.

2.3. More Examples of Discrete-Time Markov Chains

2.3.1. Choosing the State Description

Cumulative grade average (CGA)

Is a student's **credit-weighted CGA** enough to predict the next semester's average?

Choice 1: record only the current CGA A_n. Let C_n be completed credits. Either starting record below is equally likely. Each new semester adds **15 credits at GPA 4.00**.

Starting record (A_0, C_0)	Current CGA A_1	Credits C_1	Next CGA A_2
$(2.00, 15)$	**3.00**	30	$\frac{150}{45} \approx 3.33$
$(2.80, 75)$	**3.00**	90	$\frac{330}{105} \approx 3.14$

Both histories reach $A_1 = 3.00$, but knowing the earlier CGA A_0 tells us which next CGA will occur. **The CGA process (A_n) is not Markov in this model**: earlier history changes the forecast.

Including completed credits in the state

Choice 2: record CGA and completed credits, $X_n = (A_n, C_n)$. Keep the same semester model. The current state now records both the average and its weight.

$$X_{n+1} = \left(\frac{C_n A_n + 15 \times 4}{C_n + 15}, C_n + 15 \right).$$

The two records now have **different current states**: $(3.00, 30)$ and $(3.00, 90)$. Each pair determines its next pair without needing earlier CGAs.



The augmented process satisfies the Markov property

Given (A_n, C_n), earlier history adds no information about the next state. **Choose a state that retains all the information needed for prediction.**

For random semester credits and GPA, the process (X_n) is Markov if their conditional distribution depends only on the current pair, rather than on earlier history.

When failure risk depends on operating age

CGA needed credits to predict the next average. For a machine, extra information can change the **probabilities** of future outcomes.

? Question

Suppose an Up machine's failure risk depends on the number of days since its last maintenance. Is Up/Down alone enough for prediction?

⚠ A conditional probability can be well-defined but omit information

$\mathbb{P}(X_{n+1} = D \mid X_n = U)$ is well-defined when $\mathbb{P}(X_n = U) > 0$. But it averages over the possible operating ages. Learning the age from the history can change the forecast, so the Up/Down process need not be Markov.

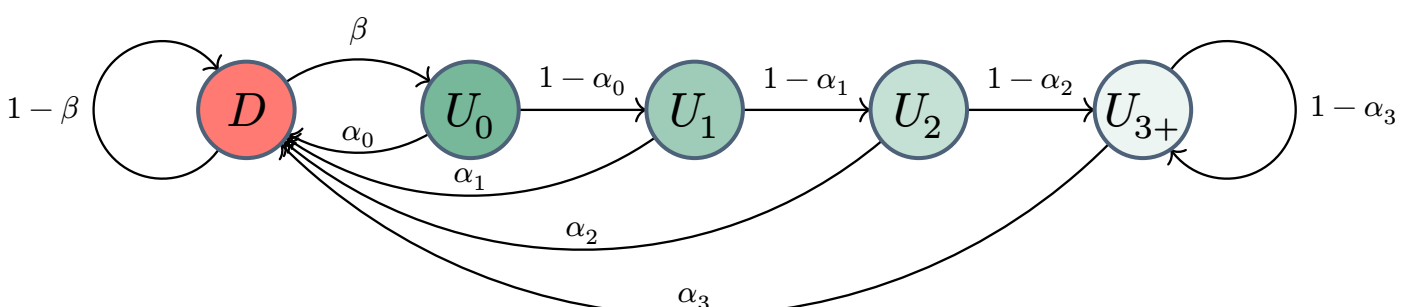
State augmentation: including operating age

Let U_k mean Up with operating age k, and let D mean Down. Use the state space $\{D, U_0, U_1, U_2, \ldots\}$.

In one possible model, α_k is the age-k failure probability:

- From U_k, move to D with probability α_k, or to U_{k+1} with probability $1 - \alpha_k$.
- From D, repair to U_0 with probability β, or stay D with probability $1 - \beta$.

These specified rules define a time-homogeneous DTMC on the augmented state space.



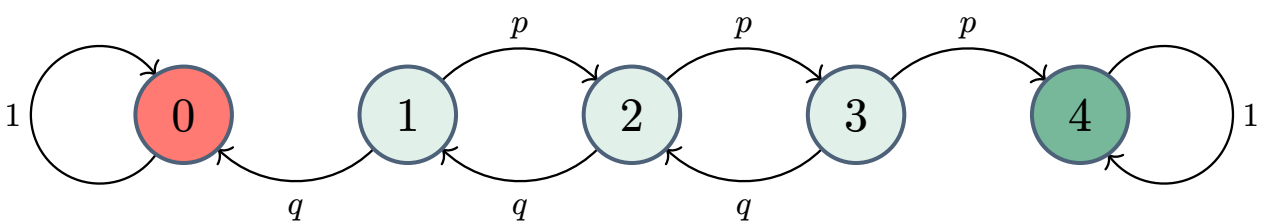
The diagram groups ages 3 and above, assuming $\alpha_k = \alpha_3$ for all $k \geq 3$.

2.3.2. Constructing Transition Probabilities

Gambler's ruin

Example 2.4

The state is the gambler's fortune, $\mathcal{S} = \{0, 1, 2, 3, 4\}$. At states $1, 2, 3$, each play adds \$1 with probability p or subtracts \$1 with probability $q = 1 - p$. At 0 or 4, play stops and the fortune stays fixed.



$$P = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{ccccc} 0 & 1 & 2 & 3 & 4 \end{array} \\ \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ q & 0 & p & 0 & 0 \\ 0 & q & 0 & p & 0 \\ 0 & 0 & q & 0 & p \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \end{array}$$

Absorbing states

Definition 2.6

A state that, once entered, is never left is called an absorbing state. In the gambler's ruin example, states 0 and 4 are absorbing states.

Transitions under a replenishment policy

Inventory chain: replenishment with a reorder point — Example 2.5

Let X_n be the **end-of-day** inventory after day n, and let D_{n+1} be the demand on day $n+1$.

(s, S)-policy:

- If $X_n \le s$, we place an order so that the **start-of-day** inventory becomes S.
- If $X_n > s$, we order nothing.
- Assume the replenishment order arrives before demand on day $n+1$ is realized. Daily demands are i.i.d. and independent of the initial inventory.

The dynamics can be written compactly as

$$X_{n+1} = \begin{cases} (X_n - D_{n+1})^+ & \text{if } X_n > s \\ (S - D_{n+1})^+ & \text{if } X_n \le s \end{cases}, \quad \text{where} \quad x^+ \stackrel{\text{def}}{=} \max\{x, 0\}.$$

So $(X_n)_{n \ge 0}$ is a DTMC on the state space $\mathcal{S} = \{0, 1, \ldots, S\}$.

From demand outcomes to transition probabilities

Suppose $s = 1$ and $S = 5$, and demand takes values $0, 1, 2, 3$ with

$$\mathbb{P}(D_{n+1} = 0) = 0.3, \quad \mathbb{P}(D_{n+1} = 1) = 0.4, \quad \mathbb{P}(D_{n+1} = 2) = 0.2, \quad \mathbb{P}(D_{n+1} = 3) = 0.1.$$

- **States**: $\mathcal{S} = \{0, 1, 2, 3, 4, 5\}$.
- **One-step transition matrix**:

$$P = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{cccccc} 0 & 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} 0 & 0 & 0.1 & 0.2 & 0.4 & 0.3 \\ 0 & 0 & 0.1 & 0.2 & 0.4 & 0.3 \\ 0.3 & 0.4 & 0.3 & 0 & 0 & 0 \\ 0.1 & 0.2 & 0.4 & 0.3 & 0 & 0 \\ 0 & 0.1 & 0.2 & 0.4 & 0.3 & 0 \\ 0 & 0 & 0.1 & 0.2 & 0.4 & 0.3 \end{bmatrix} \end{array}$$

How the model determines each row

Rows 0, 1, and 5: all start the next day with 5 units, so they have the same next-day distribution.

Row 2: demand 2 or 3 both leave no stock. Add these disjoint cases:

$$P_{2,0} = 0.2 + 0.1 = 0.3.$$

2.3.3. Forecasting the State

? Question

If we start with $X_0 = 5$, how do we forecast the **next-day** distribution of inventory?

Recall the **state distribution row vector**

$$\boldsymbol{\nu}^{(n)} \overset{\text{def}}{=} \left[\cdots \ \nu_j^{(n)} \ \cdots \right]_{j \in \mathcal{S}}, \quad \nu_j^{(n)} \overset{\text{def}}{=} \mathbb{P}(X_n = j), \quad n = 0, 1, 2, \ldots$$

With $X_0 = 5$, all probability mass is initially at state 5, so

$$\boldsymbol{\nu}^{(0)} = [0 \ \ 0 \ \ 0 \ \ 0 \ \ 0 \ \ 1].$$

Since we start at state 5 with probability 1, the distribution of the next state X_1 is given by *the row labeled* 5 (the sixth row in state order $0, 1, \ldots, 5$):

$$\boldsymbol{\nu}^{(1)} = [0 \ \ 0 \ \ 0.1 \ \ 0.2 \ \ 0.4 \ \ 0.3] = \boldsymbol{\nu}^{(0)} P.$$

Inventory forecast from an uncertain state

A probability-weighted average of rows

The row i of P describes the one-step distribution of the next state **if the current state is i**.

Therefore, when the current state is uncertain, the next-state distribution should be the **probability-weighted average** of the rows of P. The same reasoning as in the two-state machine gives

$$\boldsymbol{\nu}^{(n+1)} = \boldsymbol{\nu}^{(n)} P, \quad \text{equivalently,} \quad \nu_j^{(n+1)} = \sum_{i \in \mathcal{S}} \nu_i^{(n)} P_{ij}.$$

To understand it, being in state j at time $n+1$ requires being in *some* state i at time n and then transitioning from i to j.

The probability of this happening is the product of the probability of being in state i at time n (which is $\nu_i^{(n)}$) and the probability of transitioning from i to j (which is P_{ij}). Summing this product over all possible states i gives the total probability of being in state j at time $n+1$.

Inventory forecast two days ahead

Applying the same **one-step** update once more gives the two-day forecast:

$$\boldsymbol{\nu}^{(2)} = \boldsymbol{\nu}^{(1)} P = [0.05 \;\; 0.12 \;\; 0.22 \;\; 0.28 \;\; 0.24 \;\; 0.09].$$

! Memorize

Intuition: **current distribution $\Rightarrow$ push the probability mass through one step of P $\Rightarrow$ next distribution**.

Pause and recall

What must the state remember?

A device gives Success (S) or Failure (F). Its next result is S with probability 0.9 after SS, and 0.6 otherwise. Only the last two results matter.

1. **Keep one result?** Does the process of individual results have the Markov property? Compare histories ending in SS and FS.
2. **Keep two results.** From state FS, give the next pair and its probability for each possible outcome. Must two devices in FS have the same next result?
3. **An uncertain past.** Given that the latest result is S, the previous one was S with probability w. What is the next-success probability? Explain the multiplications and the addition.

Choose the state before calculating. The pair records the older result first.

Answers and reasoning

What must the state remember?

---

1. **Keep one result?** No. SS and FS share the latest result S, but give next-success probabilities 0.9 and 0.6. Earlier history changes the forecast.
2. **Keep two results.** The process of ordered pairs is Markov. From FS, the next pair is SS with probability 0.6 or SF with probability 0.4. Two devices have the same forecast; their outcomes can differ.
3. **An uncertain past.** $w \times 0.9 + (1 - w) \times 0.6$. Multiply each current-pair probability by its conditional success chance. Add because SS and FS are mutually exclusive and cover the possibilities.

Retain enough history for prediction; average over possible current states when uncertain.

2.4. Chapman–Kolmogorov Equations: Multi-Step Transitions

2.4.1. Composing Transitions

On Page 56, we found the machine's two-day Down probability:

$$\mathbb{P}(X_2 = D \mid X_0 = U) = 0.8 \times 0.2 + 0.2 \times 0.6 = 0.28.$$

We further extended the forecast to more days ahead on Page 59.

The core idea is to *condition on* the state k at time $n+1$, *sum* over k using the law of total probability (see Page 57), then apply the *Markov property* in (2.2).

> **? Question**
>
> For **any starting state i and destination j**, how do we combine the one-step probabilities in (2.3) to find $\mathbb{P}(X_{n+2} = j \mid X_n = i)$?

Two-step transition probabilities: For any states $i, j \in \mathcal{S}$, and any intermediate step $k \in \mathcal{S}$,

$$\begin{aligned}
\mathbb{P}(X_{n+2} = j \mid X_n = i) &= \sum_{k \in \mathcal{S}} \mathbb{P}(X_{n+2} = j, X_{n+1} = k \mid X_n = i) \\
&= \sum_{k \in \mathcal{S}} \mathbb{P}(X_{n+2} = j \mid X_{n+1} = k, X_n = i) \mathbb{P}(X_{n+1} = k \mid X_n = i) \\
&= \sum_{k \in \mathcal{S}} \mathbb{P}(X_{n+2} = j \mid X_{n+1} = k) \mathbb{P}(X_{n+1} = k \mid X_n = i) \\
&= \sum_{k \in \mathcal{S}} P_{ik} P_{kj}.
\end{aligned}$$

- The first equality is the **law of total probability**, summing over all possible intermediate states k at time $n+1$.
- The second equality uses the definition of conditional probability.
- The third equality uses the **Markov property** to eliminate dependence on $X_n = i$ when conditioning on $X_{n+1} = k$.

Two-step transitions by matrix squaring

Proposition 2.1

If P is the one-step transition matrix, then the two-step transition matrix is

$$P^{(2)} = PP = P^2, \quad P_{ij}^{(2)} = \sum_{k \in \mathcal{S}} P_{ik} P_{kj}.$$

The shared state k connects the first step's destination to the second step's starting state. For the machine, in state order (U, D),

$$P^2 = \begin{pmatrix} 0.8 & 0.2 \\ 0.4 & 0.6 \end{pmatrix} \begin{pmatrix} 0.8 & 0.2 \\ 0.4 & 0.6 \end{pmatrix} = \begin{pmatrix} 0.72 & 0.28 \\ 0.56 & 0.44 \end{pmatrix}.$$

The U, D entry is our earlier 0.28. The other entries give the remaining two-step forecasts.

Sanity check: random walk returns in two steps

Example 2.6

For a simple random walk on $\mathbb{Z}$ with $\mathbb{P}(\xi = +1) = p$ and $\mathbb{P}(\xi = -1) = q$, starting from 0 the only way to return to 0 in 2 steps is "up then down" or "down then up".

$$\mathbb{P}(X_2 = 0 \mid X_0 = 0) = pq + qp = 2pq.$$

? Question

How to compute the n-step transition probabilities $\mathbb{P}\left(X_{m+n} = j \,\middle|\, X_m = i\right)$?



n-step transition probabilities — Definition 2.7

For a time-homogeneous DTMC on state space $\mathcal{S}$, define

$$P_{ij}^{(n)} \overset{\text{def}}{=} \mathbb{P}(X_{m+n} = j \mid X_m = i), \quad n = 0, 1, 2, \ldots$$

which does not depend on m by time-homogeneity.

n-step transition probabilities — Poll 2.2

Given the one-step transition matrix P, how to compute the n-step transition matrix $P^{(n)}$?

A). P^n (matrix multiplication).
B). nP (scalar multiplication).
C). $P + P + \ldots + P$ (matrix addition).

The **Chapman–Kolmogorov equations** provide a method for computing n-step transition probabilities.

Chapman–Kolmogorov equations

Definition 2.8

For any $m, n \geq 0$ and any states i, j,

$$P_{ij}^{(m+n)} = \sum_{k \in \mathcal{S}} P_{ik}^{(m)} P_{kj}^{(n)}. \tag{2.4}$$

Proof of Equation (2.4): Fix an arbitrary start time r and condition on the **intermediate** state X_{r+m}. Using the law of total probability,

$$\begin{aligned} P_{ij}^{(m+n)} &= \mathbb{P}(X_{r+m+n} = j \mid X_r = i) \\ &= \sum_{k \in \mathcal{S}} \mathbb{P}(X_{r+m+n} = j, X_{r+m} = k \mid X_r = i) \\ &= \sum_{k \in \mathcal{S}} \mathbb{P}(X_{r+m+n} = j \mid X_{r+m} = k, X_r = i) \mathbb{P}(X_{r+m} = k \mid X_r = i) \\ &= \sum_{k \in \mathcal{S}} \mathbb{P}(X_{r+m+n} = j \mid X_{r+m} = k) \mathbb{P}(X_{r+m} = k \mid X_r = i), \end{aligned}$$

where the last line follows from the *Markov property*: given $X_{r+m} = k$ the future does not depend on the earlier history. Finally, by time-homogeneity (shift-invariance),

$$\mathbb{P}(X_{r+m} = k \mid X_r = i) = P_{ik}^{(m)}, \quad \mathbb{P}(X_{r+m+n} = j \mid X_{r+m} = k) = P_{kj}^{(n)}.$$

Substituting these into the sum gives (2.4). ∎

! Memorize

Let P be the one-step transition matrix. By Chapman–Kolmogorov equations, we have the matrix multiplication rule:

$$P^{(m+n)} = P^{(m)} \cdot P^{(n)} = \begin{bmatrix} P_{11}^{(m)} & P_{12}^{(m)} & \cdots & P_{1J}^{(m)} \\ P_{21}^{(m)} & P_{22}^{(m)} & \cdots & P_{2J}^{(m)} \\ \vdots & \vdots & \ddots & \vdots \\ P_{I1}^{(m)} & P_{I2}^{(m)} & \cdots & P_{IJ}^{(m)} \end{bmatrix} \cdot \begin{bmatrix} P_{11}^{(n)} & P_{12}^{(n)} & \cdots & P_{1J}^{(n)} \\ P_{21}^{(n)} & P_{22}^{(n)} & \cdots & P_{2J}^{(n)} \\ \vdots & \vdots & \ddots & \vdots \\ P_{I1}^{(n)} & P_{I2}^{(n)} & \cdots & P_{IJ}^{(n)} \end{bmatrix}, \quad \forall m, n \geq 0.$$

In particular, taking $n = 1$ and iterating gives the familiar "matrix power" representation:

$$P^{(n)} = P^{(n-1)} \cdot P = P^{(n-2)} \cdot P \cdot P = \cdots = P^n, \quad n = 0, 1, 2, \ldots$$

Once you know P, every multi-step transition probability is an entry of P^n. Together with an initial distribution, the transition matrix P specifies the finite-dimensional distributions of a time-homogeneous DTMC.

2.4.2. Evolving a State Distribution

? Question

So far, we have focused on **point-to-point** transitions. Often in application, we don't care about the exact path along which the system evolves, but rather the **distribution** of the state.

- **Example:** in gambling, we do not care how we got to our current fortune, only (the probability of) how much money we have now.
- **Question:** How does the distribution of the state evolve day by day?

Recall the **state distribution row vector**

$$\boldsymbol{\nu}^{(n)} \overset{\text{def}}{=} \begin{bmatrix} \cdots & \nu_j^{(n)} & \cdots \end{bmatrix}_{j \in \mathcal{S}}, \quad \nu_j^{(n)} \overset{\text{def}}{=} \mathbb{P}(X_n = j), \quad n = 0, 1, 2, \ldots$$

Discrete-time Kolmogorov forward equation

Theorem 2.1

Let $(X_n)_{n\geq 0}$ be a time-homogeneous DTMC with one-step transition matrix P. Then the one-step distribution update is

$$\boldsymbol{\nu}^{(n+1)} = \boldsymbol{\nu}^{(n)} P, \quad n \geq 0,$$

equivalently, componentwise,

$$\nu_j^{(n+1)} = \sum_{i \in \mathcal{S}} \nu_i^{(n)} P_{ij}, \quad j \in \mathcal{S}.$$

Iterating yields the n-step distribution

$$\boldsymbol{\nu}^{(n)} = \boldsymbol{\nu}^{(0)} P^n, \quad n = 0, 1, 2, \ldots$$

(with P^0 the identity matrix).

Proof of Theorem 2.1: Fix any $j \in \mathcal{S}$. By the law of total probability,

$$\nu_j^{(n+1)} = \mathbb{P}(X_{n+1} = j) = \sum_{i \in \mathcal{S}} \mathbb{P}(X_{n+1} = j, X_n = i).$$

Then factor the joint probability:

$$\mathbb{P}(X_{n+1} = j, X_n = i) = \mathbb{P}(X_{n+1} = j \mid X_n = i)\mathbb{P}(X_n = i) = P_{ij}\nu_i^{(n)}.$$

Summing over i gives

$$\nu_j^{(n+1)} = \sum_{i \in \mathcal{S}} \nu_i^{(n)} P_{ij},$$

which is exactly the matrix identity $\boldsymbol{\nu}^{(n+1)} = \boldsymbol{\nu}^{(n)} P$. Iterating over n yields

$$\boldsymbol{\nu}^{(n)} = \boldsymbol{\nu}^{(n-1)} P = \boldsymbol{\nu}^{(n-2)} P^2 = \cdots = \boldsymbol{\nu}^{(0)} P^n.$$

Transition as linear transformation

The transition matrix defines a linear transformation that maps the current probability vector to the next probability vector:

$$\boldsymbol{\nu}^{(n+1)} = \boldsymbol{\nu}^{(n)} P, \quad n \geq 0,$$

- **It preserves probability**: because P has nonnegative entries and each row sums to one, the output remains a valid probability vector:

$$\boldsymbol{\nu}^{(n+1)}\mathbf{1} = \left(\boldsymbol{\nu}^{(n)} P\right)\mathbf{1} = \boldsymbol{\nu}^{(n)}(P\mathbf{1}) = \boldsymbol{\nu}^{(n)}\mathbf{1} = 1, \quad n \geq 0.$$

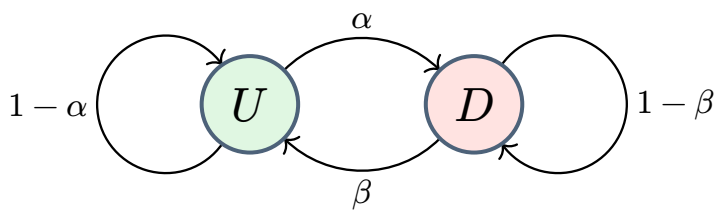
- **It preserves mixtures:** For $0 \leq a \leq 1$,

$$[a\boldsymbol{\nu}_1 + (1-a)\boldsymbol{\nu}_2]P = a(\boldsymbol{\nu}_1 P) + (1-a)(\boldsymbol{\nu}_2 P).$$

Mixing two current distributions and then evolving gives the same result as evolving each separately and then mixing.

? Question

Return to the machine in Example 2.3, now allowing general failure and repair probabilities α and β:



$$P = \begin{array}{c} \\ U \\ D \end{array} \begin{array}{c} \begin{array}{cc} U & \quad D \end{array} \\ \begin{bmatrix} 1-\alpha & \alpha \\ \beta & 1-\beta \end{bmatrix} \end{array}$$

Suppose the machine is **up** in day 0, i.e., $\boldsymbol{\nu}^{(0)} = (1, 0)$.

What is the probability it is **down** after 1 day? after 2 days?

After 1 day:

$$\boldsymbol{\nu}^{(1)} = \boldsymbol{\nu}^{(0)} P = [1 \;\; 0] P = [1-\alpha \;\; \alpha] \quad \Rightarrow \quad \mathbb{P}(X_1 = D) = \boldsymbol{\nu}^{(1)} \mathbf{1}_D = \alpha$$

Here, $\mathbf{1}_D$ is the *column vector* with 1 in the D component and 0 elsewhere.

After 2 days:

$$\boldsymbol{\nu}^{(2)} = \boldsymbol{\nu}^{(1)} P = (1,0) P^2, \text{where } P^2 = \begin{bmatrix} (1-\alpha)^2 + \alpha\beta & \alpha(2-\alpha-\beta) \\ \beta(2-\alpha-\beta) & (1-\beta)^2 + \alpha\beta \end{bmatrix}.$$

Thus,

$$\mathbb{P}(X_2 = D) = \boldsymbol{\nu}^{(0)} P^2 \mathbf{1}_D = [1 \;\; 0] P^2 \mathbf{1}_D = \alpha(2-\alpha-\beta).$$

- What if the distribution at time 0 is $\boldsymbol{\nu}^{(0)} = \left(\frac{1}{2}, \frac{1}{2}\right)$?

$$\mathbb{P}(X_2 = D) = \boldsymbol{\nu}^{(0)} P^2 \mathbf{1}_D = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \end{bmatrix} P^2 \mathbf{1}_D = \frac{1}{2}\alpha(2-\alpha-\beta) + \frac{1}{2}((1-\beta)^2 + \alpha\beta).$$

2.5. Simple Random Walk: Fluctuations and Hitting Times

2.5.1. Drift and Spread

We can now describe the walk as a Markov chain and make multi-step forecasts. We return to its engineering questions: how far does it spread, and when does it reach a target?

Platform growth — Example 2.7

A platform's trajectory can also look like "many small daily wins and losses."

- **State**: X_n = cumulative user base (or a running product "health score").
- **Randomness**: some days bring organic growth; other days bring churn (or incidents).
- **Net change**: ξ_{n+1} = net gain (growth minus churn) on day $n+1$.
- **Question**: Will we eventually "take off" past a target level, or drift down to a critical level?

Gambler's bankroll

Example 2.8

A gambler plays the same game repeatedly. Each round she either wins \$1 or loses \$1.

- **State**: X_n = bankroll after n rounds.
- **Randomness**: each round is a win/loss (think: "good day" vs "bad day").
- **Net change**: $\xi_{n+1} \in \{+1, -1\}$ is the increment in round $n+1$ ($+1$ = win, -1 = loss).
- **Questions**: Starting with i dollars, what is the chance of bankruptcy before reaching N (cashing out)? How long does it take (on average) to either go broke or cash out?

Refresher: Independent and identically distributed (i.i.d.)

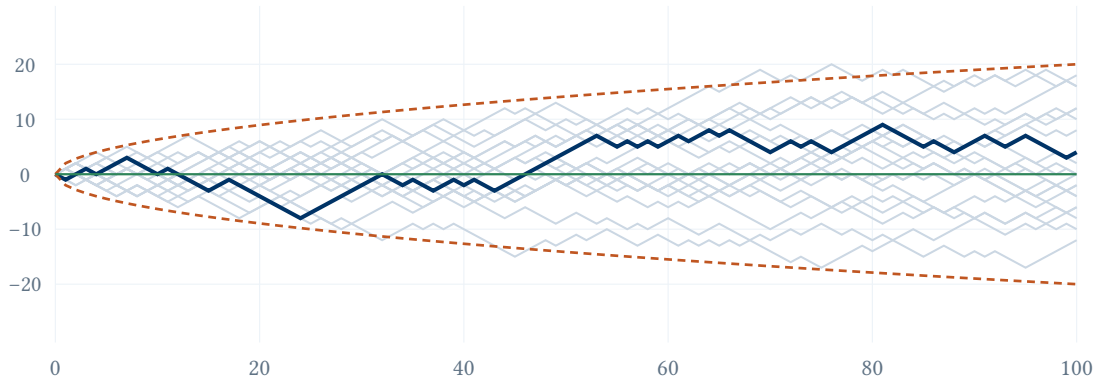
A sequence of random variables $X_1, X_2, \ldots$ is **i.i.d.** if it satisfies two properties:

- **Identically distributed**: all X_n share the same distribution.
  - *Intuition*: each step is generated by the **same mechanism** (same coin, same demand model, same service-time law).
- **Independent**: knowing past values gives no information about future ones.
  - *Intuition*: the randomness "resets" each time; there is no dependence across periods.

In a simple random walk, the increments $\xi_1, \xi_2, \ldots$ are i.i.d.:

- Each step has the same win/loss probabilities, and one step does not influence the next.
- p is the probability of an "up" step. The **trend** of the walk:
  - $p > \frac{1}{2}$: "up" steps are more likely (upward drift).
  - $p = \frac{1}{2}$: no systematic trend (fair game).
  - $p < \frac{1}{2}$: "down" steps are more likely (downward drift).

X_n: one path, independent comparison paths, and $\pm 2\sqrt{n}$



The curves are pointwise standard-deviation scales, not a simultaneous path bound.

? Question

Compare one realized path with many independent runs. What should “mean zero” tell us, and what does it not tell us?

The two-standard-deviation curves describe the spread at each fixed time. They are not an envelope that must contain the entire path.

Mean, variance, and distribution

To formalize the intuition about trends and long-run behavior, we compute the **mean** and **variance** of the state X_n after n steps.

Simple random walk mean and variance — Proposition 2.2

For a simple random walk $\{X_n\}$ with step probability p:

- **Mean**: $\mathbb{E}[X_n] = (2p-1)n$.
- **Variance**: $\mathrm{Var}(X_n) = 4pqn = 4p(1-p)n$.

Proof of Proposition 2.2: By linearity of expectation and i.i.d. increments,

$$\mathbb{E}[X_n] = \mathbb{E}\left[\sum_{i=1}^{n} \xi_i\right] = \sum_{i=1}^{n} \mathbb{E}[\xi_i] = n\mathbb{E}[\xi_1].$$

Since $\mathbb{E}[\xi_1] = (+1)\cdot p + (-1)\cdot q = p - q = 2p - 1$, we have $\mathbb{E}[X_n] = n(2p-1)$.

For the variance, since the ξ_i are independent,

$$\mathrm{Var}(X_n) = \mathrm{Var}\left(\sum_{i=1}^{n} \xi_i\right) = \sum_{i=1}^{n} \mathrm{Var}(\xi_i) = n\mathrm{Var}(\xi_1).$$

Now, $\mathbb{E}[\xi_1^2] = (+1)^2 \cdot p + (-1)^2 \cdot q = p + q = 1$, so

$$\mathrm{Var}(\xi_1) = \mathbb{E}[\xi_1^2] - (\mathbb{E}[\xi_1])^2 = 1 - (2p-1)^2 = 4pq = 4p(1-p).$$

Therefore, $\mathrm{Var}(X_n) = n\mathrm{Var}(\xi_1) = 4pqn = 4p(1-p)n$. ∎

Symmetric random walk

Example 2.9

A simple random walk with $p = \frac{1}{2}$ is called a **symmetric random walk**. Proposition 2.2 implies that,

- the symmetric random walk has zero mean $\mathbb{E}[X_n] = 0$;
  - The ensemble mean remains zero; a realized path need not remain near zero.
- but variance that **grows linearly** with time: $\mathrm{Var}(X_n) = n$.
  - The position of the state becomes *more and more spread out* over time.

Even when there is no systematic trend, the walk does **not** sit still. Random fluctuations accumulate ($\mathrm{Var}(X_n) = n$), so the state can wander far away and hit critical thresholds.

? Question

What is the distribution of the state?

Because $\xi_1, \xi_2, \ldots, \xi_n \in \{+1, -1\}$, the position $X_n = \sum_{i=1}^n \xi_i$ is determined by *how many* "+1 steps" occurred. Let

$$N_n \overset{\text{def}}{=} \sum_{i=1}^{n} \mathbb{1}(\xi_i = +1) \sim \text{Binomial}(n, p),$$

be the number of $+1$ increments among the first n steps. From (2.1), we have

$$X_n = X_{n-1} + \xi_n = (X_{n-2} + \xi_{n-1}) + \xi_n = \sum_{i=1}^{n} \xi_i = (+1)N_n + (-1)(n - N_n) = 2N_n - n.$$

X_n has a shifted-and-scaled binomial distribution

$$\mathbb{P}(X_n = x) = \mathbb{P}(2N_n - n = x) = \mathbb{P}\left(N_n = \frac{n+x}{2}\right) = \binom{n}{\frac{n+x}{2}} p^{\frac{n+x}{2}} q^{\frac{n-x}{2}},$$

for values x such that $x \in \{-n, -n+2, \ldots, n\}$; otherwise $\mathbb{P}(X_n = x) = 0$.

Probability refresher: Central Limit Theorem (CLT)

A sequence of random variables $\xi_1, \xi_2, \ldots$ is **i.i.d.**, each with mean μ and variance $\sigma^2 < \infty$. Then, as $n \to \infty$, the **standardized sum** converges to the standard normal distribution:

$$Z_n \overset{\text{def}}{=} \frac{\sum_{i=1}^{n} \xi_i - n\mu}{\sigma\sqrt{n}} \Rightarrow \mathbb{N}(0,1).$$

Application: The long-run behavior of X_n via CLT

Example 2.10

For $0 < p < 1$, a simple random walk with step probability p has $\mathbb{E}[\xi_i] = 2p - 1$ and $\mathrm{Var}(\xi_i) = 4pq$. By the Central Limit Theorem, as $n \to \infty$,

$$Z_n \overset{\text{def}}{=} \frac{X_n - n(2p-1)}{2\sqrt{pqn}} \Rightarrow \mathbb{N}(0,1) \quad \text{hence} \quad X_n \approx \mathbb{N}(n(2p-1), 4pqn).$$

2.5.2. Reaching a Target

? Question

Will a simple random walk hit a fixed target level T (can be positive or negative)?

- In practice, T is often a **threshold** (e.g., 0 for bankruptcy, N for successful cashout).

Hitting Time **Definition 2.9**

The **hitting time** τ_a of a level a is the *first time the process touches* a:

$$\tau_a \stackrel{\text{def}}{=} \min\{n \geq 0 : X_n = a\}. \tag{2.5}$$

If the chain never hits a, we set $\tau_a = \infty$ by convention.

Intuition: The walk's **drift** helps us analyze the hitting time in (2.5).

Upward drift ($p > \frac{1}{2}$): certainly hits positive targets

Example 2.11

Intuitively, the walk will eventually hit any fixed **positive** target $a \in \mathbb{Z}_{>0}$ because it tends to go up. Formally, the mean step is $\mathbb{E}[\xi_1] = p - q = 2p - 1 > 0$.

By the **(strong) law of large numbers**,

$$\frac{X_n}{n} = \left(\frac{1}{n}\right) \sum_{i=1}^{n} \xi_i \Rightarrow \mathbb{E}[\xi_1] = 2p - 1 > 0 \quad \text{as} \quad n \to \infty.$$

This says that, with probability 1, the ratio $\frac{X_n}{n}$ eventually stays close to $2p - 1$. In particular, along almost every sample path, X_n grows roughly linearly and hence $X_n \to \infty$. Therefore, for any fixed $a \in \mathbb{Z}_{>0}$, the walk must satisfy $X_n \geq a$ for all sufficiently large n.

Equivalently, the event "$X_n < a$ for all n" has probability 0, i.e.,

$$\mathbb{P}(\tau_a < \infty) = 1, \quad \forall a \in \mathbb{Z}_{>0}.$$

Negative drift ($p < \frac{1}{2}$): may never hit a positive target

Example 2.12

When $p < \frac{1}{2}$, the mean step is negative: $\mathbb{E}[\xi_1] = 2p - 1 < 0$. By the law of large numbers,

$$\frac{X_n}{n} \Rightarrow 2p - 1 < 0 \quad \text{as} \quad n \to \infty,$$

so along almost every sample path, X_n eventually trends downward and $X_n \to -\infty$, i.e.,

$$\mathbb{P}(\tau_a = \infty) > 0, \quad \forall a \in \mathbb{Z}_{<0}.$$

Warning

This does **not** mean the walk can never reach a positive target $a \in \mathbb{Z}_{>0}$. It can still get lucky early and jump up to a. The point is: there is a **strictly positive** chance it drifts down forever **without ever** reaching a.

Casino game where the house has a small edge — Example 2.13

Let X_n be your net winnings after n bets, with $X_0 = 0$, in a game with a small house edge.

- Assume you can continue playing indefinitely, with **no bankruptcy boundary**.
- Your goal is to reach a positive net-winnings target $a \in \mathbb{Z}_{>0}$.
- With negative drift, there is a nonzero chance you **never** reach a.

Question: How likely is it to ever hit a when the drift is negative ($0 < p < \frac{1}{2}$)?

Answer (appendix proof: Page 485): The probability is **less than 1** and decays *exponentially* in a:

$$\mathbb{P}(\tau_a < \infty) = \left(\frac{p}{q}\right)^a, \quad q \overset{\text{def}}{=} 1 - p.$$

Interpretation: each extra dollar in the target multiplies the success probability by $\frac{p}{q} < 1$. Successful paths may include losses along the way.

Change of measure: a heuristic proof sketch

Idea: solve the question using friendlier odds, then correct each path's probability back to the original odds. For easy arithmetic, take $p = \frac{1}{3}$, $q = \frac{2}{3}$, and target $a = 3$.

1. **Imagine an easier game.** Swap the win and loss probabilities. Now a win has probability $\frac{2}{3}$. The walk drifts up, so it eventually reaches $+3$ with probability 1.
2. **Compare the same sequence of wins and losses.** Under the original odds, each win is **half as likely** and each loss is **twice as likely**. Pair each loss with a win: $\left(\frac{1}{2}\right) \times 2 = 1$. Every path that first reaches $+3$ has three extra wins, leaving a factor $\left(\frac{1}{2}\right)^3 = \frac{1}{8}$.
3. **Add the corrected probabilities.** Here, every successful path has the same correction factor:

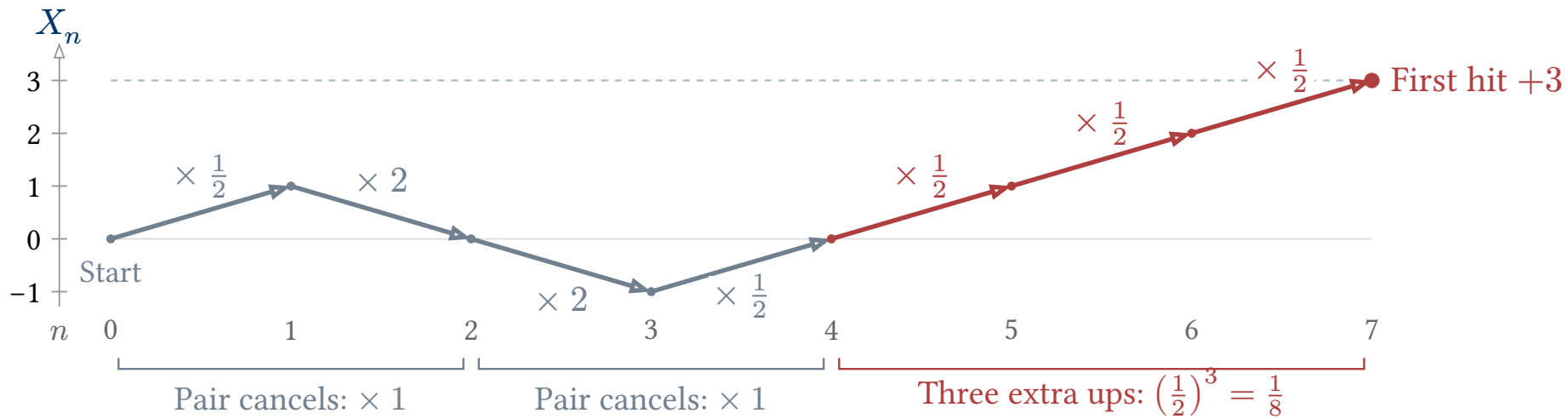
$$\text{Original success probability} = \left(\frac{1}{8}\right) \times \underbrace{1}_{\text{success in the easier game}} = \frac{1}{8}.$$

For general $0 < p < \frac{1}{2}$ and target a, the same argument gives a factor $\left(\frac{p}{q}\right)^a$.

Change of measure: a path and its correction factors

Original odds: $p = \frac{1}{3}, q = \frac{2}{3}$. **Swapped odds:** $p' = \frac{2}{3}, q' = \frac{1}{3}$.

Each edge shows the factor **original probability / swapped probability**.



Every first-hit path to $+3$ has three extra upward steps. Summing their weights gives

$$\mathbb{P}_{\text{original}}(\tau_3 < \infty) = \frac{1}{8} \times \underbrace{\mathbb{P}_{\text{swapped}}(\tau_3 < \infty)}_{1} = \frac{1}{8}.$$

Symmetric random walk: hitting probabilities and expected times

Starting from $X_0 = 0$, with no stopping boundary:

Drift	Positive target ($a > 0$)	Negative target ($a < 0$)
Upward ($p > \frac{1}{2}$)	Hits with probability 1	May never hit
None ($p = \frac{1}{2}$)	?	?
Downward ($p < \frac{1}{2}$)	May never hit	Hits with probability 1

With **no systematic drift**, how should we fill the two question marks?

Hitting probabilities under symmetric random walk — Poll 2.3

Consider a **symmetric** random walk ($p = \frac{1}{2}$) starting at $X_0 = 0$, so there is **no systematic drift**.

What is the probability of hitting a fixed target $a \in \mathbb{Z}_{>0}$?

A). Probability 0.
B). Probability strictly between 0 and 1.
C). Probability 1.

The symmetric random walk hits every level with probability 1

Let h be the probability of ever reaching $+1$ from 0. By shifting the starting point, h is also the chance of ever reaching **one level higher**.

Look at the first step:

- **Up to $+1$** (probability $\frac{1}{2}$): we have already reached the target.
- **Down to -1** (probability $\frac{1}{2}$): first reach 0, then $+1$. Each is the same one-level climb, with success probability h. At 0, future coin tosses start afresh. So both climbs succeed with probability $h \times h$.

$$h = \underbrace{\left(\frac{1}{2}\right) \times 1}_{\text{first step up}} + \underbrace{\left(\frac{1}{2}\right) \times h \times h}_{\text{first step down}}.$$

Rearranging gives $(h-1)^2 = 0$, so $\boldsymbol{h = 1}$.

Other targets: reaching $+a$ takes a such climbs, each finishing with probability 1. Negative targets follow by symmetry.

First-step analysis (or first-step conditioning)

Condition on the next step, then express the remaining problem using the same unknowns. This is a standard tool for finding hitting probabilities and expected times in stochastic processes.

Expected hitting time under symmetric random walk — Poll 2.4

Consider the same **symmetric** random walk ($p = \frac{1}{2}$) starting at $X_0 = 0$, with no stopping boundary.

What is the expected time $\mathbb{E}[\tau_a]$ to first hit a fixed target $a \in \mathbb{Z}_{>0}$?

A). Exactly a steps.
B). Finite, but greater than a steps.
C). Infinite.

The expected hitting time is infinite

Suppose reaching $+1$ from 0 takes a **finite** average of m steps. Every one-level climb has the same expected duration m, including any wandering along the way.

- **Up to $+1$** (probability $\frac{1}{2}$): no more steps are needed.
- **Down to -1** (probability $\frac{1}{2}$): reach 0, then $+1$. These two climbs take $2m$ more steps on average.

$$m = \underbrace{1}_{\text{first step}} + \underbrace{\left(\frac{1}{2}\right) \times 0}_{\text{already there}} + \underbrace{\left(\frac{1}{2}\right) \times (2m)}_{\text{two climbs left}} = 1 + m.$$

No finite m can satisfy this equation. Therefore $\mathbb{E}[\tau_{+1}] = \infty$.

For $+a$, even the first climb has infinite mean duration. Negative targets follow by symmetry.

Principle to remember

Relate the unknown to a simpler problem

First-step analysis (Page 105)

Condition on the first step. The remaining journey consists of copies of the same one-level problem.

For $h = \mathbb{P}_0(\tau_1 < \infty)$:

$$h = \frac{1}{2} + \frac{1}{2}h^2.$$

Change of measure (Page 102)

Swap p and $q = 1 - p$ ($0 < p < \frac{1}{2}$). Under the new odds, hitting $+a$ is certain.

Each first-hit path has a extra upward steps, giving the correction factor $\left(\frac{p}{q}\right)^a$:

$$\mathbb{P}_{\text{original}}(\tau_a < \infty) = \left(\frac{p}{q}\right)^a \times 1.$$

Use the structure of the paths to write an **exact identity**.

The monkey and the drunkard

Infinite Monkey Theorem **Theorem 2.2**

A monkey presses keys **independently and uniformly** on a finite keyboard. With probability 1, it eventually types **any fixed finite text** using those keys—even the complete works of Shakespeare.

Waiting for...	Eventual success	Expected waiting time
The monkey to type Shakespeare	Probability 1	Finite, but enormous
The symmetric walk to go from 0 to +1	Probability 1	Infinite

The monkey gets fresh chances. For a target of m characters and K equally likely keys, each disjoint block of m keystrokes succeeds with probability K^{-m}. Hence the waiting time T in keystrokes satisfies

$$\mathbb{E}[T] \leq mK^m < \infty.$$

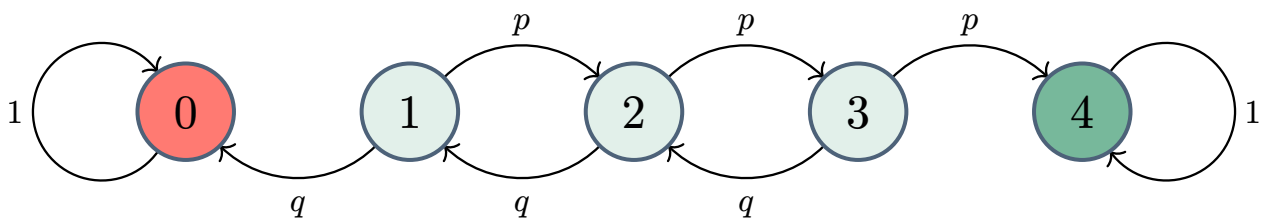
Eventual success, expected waiting time, and success before a deadline are different questions.

Source: Berkeley Data 140: Expected Waiting Times.

2.6. Classification of States

2.6.1. Returns and Repeated Visits

Recall the gambler's ruin DTMC in Example 2.4 with state space $\mathcal{S} = \{0, 1, 2, 3, 4\}$.



Suppose we start from $X_0 = 2$. Typical sample paths look like

$$2 \to 3 \to 2 \to 1 \to 2 \to 3 \to 4 \to 4 \to 4 \to \ldots$$

$$2 \to 1 \to 2 \to 3 \to 2 \to 1 \to 0 \to 0 \to 0 \to \ldots$$

- The chain *wanders* among the interior states $1, 2, 3$ for a while.
- Eventually it hits 0 (broke) or 4 (cash out), and then it is *stuck forever*.



Warning

These paths suggest two qualitatively different kinds of states:

- **Traps**: 0 and 4. Once entered, the chain keeps visiting the same state forever.
- **Waypoints**: 1, 2, 3. They may be visited many times, but there is always a chance we *leave and never come back* (because absorption occurs first).

We call traps **recurrent** and waypoints **transient**.

First return time **Definition 2.10**

Fix a state $i \in \mathcal{S}$. The **first return time** to i (after time 0) is

$$T_i^+ \stackrel{\text{def}}{=} \min\{n \geq 1 : X_n = i\}.$$

If the chain never returns to i, we set $T_i^+ = \infty$.

Transient vs recurrent states

Definition 2.11

Let $i \in \mathcal{S}$. Define the **return probability**

$$f_{ii} \overset{\text{def}}{=} \mathbb{P}(T_i^+ < \infty \mid X_0 = i).$$

This is the probability that, starting from i, the chain returns to i at some finite time.

- State i is **recurrent** if $f_{ii} = 1$ (starting from i, you return to i again with probability 1).
- State i is **transient** if $f_{ii} < 1$ (there is a nonzero chance you leave i and never come back).

In Example 2.4:

- The states 0 and 4 are absorbing, hence *recurrent*.
- The states $1, 2, 3$ are *transient*. For instance, starting from 1:
  - with probability q, the chain moves to 0 in one step and is absorbed;
  - therefore $\mathbb{P}(T_1^+ = \infty \mid X_0 = 1) \geq q > 0$, so $f_{11} < 1$.

Chance of returning to a state multiple times

Recall first-step analysis: We reused the same problem after a restart.

Here, restart at the first return. On $T_i^+ < \infty$, the chain is back at i, and the chance of returning **one more time** is again f_{ii}:

$$\mathbb{P}(\text{Return at least twice} \mid X_0 = i) = \mathbb{P}(T_i^+ < \infty \mid X_0 = i) \cdot f_{ii} = f_{ii}^2.$$

Restarting at a return uses the strong Markov property

- **Markov property**: restart from the present state at a **fixed time**, as after the first step.
- **Strong Markov property**: restart at a **stopping time**, such as the first return T_i^+. A stopping time can be recognized from the observed history, without looking into the future.

Recurrence means infinitely many returns

Starting from i, repeat the restart argument. For every $k \geq 1$,

$$\mathbb{P}(\text{at least } k \text{ returns after time } 0 \mid X_0 = i) = f_{ii}^k.$$

Letting $k \to \infty$ connects the return probability to long-run behavior:

- **Recurrent** ($f_{ii} = 1$): infinitely many returns with probability 1.
- **Transient** ($f_{ii} < 1$): $f_{ii}^k \to 0$, so only finitely many returns with probability 1.

Back to first-step analysis: the symmetric walk first moves to $+1$ or -1. Either state hits 0 with probability 1 (Page 105), so

$$f_{00} = \frac{1}{2} \times 1 + \frac{1}{2} \times 1 = 1.$$

Thus 0 is recurrent; by translation, **every integer state is recurrent**.

Recurrence does not imply a finite mean return time. After the first step, returning to 0 takes a one-level journey with infinite mean (Page 107). Hence $\mathbb{E}\left[T_0^+ \;\middle|\; X_0 = 0\right] = \infty$, despite infinitely many returns.

Number of visits to a transient state — Proposition 2.3

If i is transient, i.e., if $f_{ii} < 1$, then starting from i, the number of time periods that the process will be in state i has a geometric distribution with finite mean $\frac{1}{1-f_{ii}}$.

Proof: Let N_i be the number of visits to state i (including the initial time 0). Each time the process enters state i, there will be a positive probability, namely, $1 - f_{ii}$, that it will never again enter that state. Therefore, starting in state i, the probability that the process will be in state i for exactly n time periods equals

$$\mathbb{P}(N_i = n \mid X_0 = i) = f_{ii}^{n-1}(1 - f_{ii}).$$

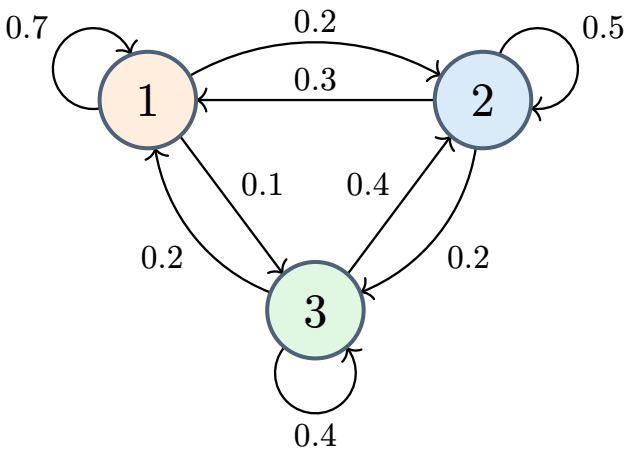
This is the probability mass function of a geometric distribution with a finite mean $\frac{1}{1-f_{ii}}$. ∎

Social mobility

Example 2.14

Let X_n be a family's social class in the nth generation, taking values 1 = lower, 2 = middle, or 3 = upper. In this (very simplified) model, social mobility follows a Markov chain.

$$P = \begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{bmatrix} 0.7 & 0.2 & 0.1 \\ 0.3 & 0.5 & 0.2 \\ 0.2 & 0.4 & 0.4 \end{bmatrix} \end{array}$$



Are the social classes recurrent or transient?

Poll 2.5

1. All classes are recurrent.
2. Upper class 3 is transient; classes 1 and 2 are recurrent.
3. All classes are transient.

Proof of all classes are recurrent: Note that, from any current class, the probability of moving to class 3 in the next generation is at least 0.1. Therefore,

$$\begin{aligned}\mathbb{P}(T_3^+ > n \mid X_0 = i) &= \mathbb{P}(X_1 \neq 3, ..., X_n \neq 3 \mid X_0 = i) \\ &\leq (1 - 0.1)^n = 0.9^n \to 0, \quad \text{as } n \to \infty, \quad \forall i \in \{1, 2, 3\}.\end{aligned}$$

Hence $f_{33} = \mathbb{P}(T_3^+ < \infty \mid X_0 = 3) = 1$, so class 3 is recurrent.

The same argument works for classes 1 and 2. Indeed, from any current class, the probability of moving to class 1 in one step is at least 0.2, and the probability of moving to class 2 in one step is also at least 0.2. Thus, for all $i \in \{1, 2, 3\}$,

$$\mathbb{P}(T_1^+ > n \mid X_0 = i) \leq 0.8^n \to 0, \quad \mathbb{P}(T_2^+ > n \mid X_0 = i) \leq 0.8^n \to 0, \quad \text{as } n \to \infty.$$

Therefore $f_{11} = f_{22} = 1$, and all three classes are recurrent. ∎

Recall from Definition 2.7 the n-step transition probability $P_{xy}^{(n)} = \mathbb{P}(X_n = y \mid X_0 = x)$.

Accessibility and communication — Definition 2.12

- State y is **accessible from** x, written $x \to y$, if there exists $n \geq 0$ such that $P_{xy}^{(n)} > 0$. In words, starting from x, there is a positive chance to reach y in finitely many steps.
- States x and y **communicate**, written $x \leftrightarrow y$, if both $x \to y$ and $y \to x$.

i Useful facts

1. Accessibility is transitive: if $x \to y$ and $y \to z$, then $x \to z$.
2. Communication is an equivalence relation: it is reflexive, symmetric, and transitive.
3. Recurrence and transience are class properties: communicating states have the same classification.
4. If x is recurrent and $x \to y$, then necessarily $y \to x$; otherwise the chain could leave x and fail to return with positive probability.

Enrichment: changing the dimension

The same “drunkard” behaves very differently once you change the dimension:

- **1D**: steps left/right — a drunken ant living on a rope.
- **2D**: steps on a grid — a drunken human wandering in city blocks.

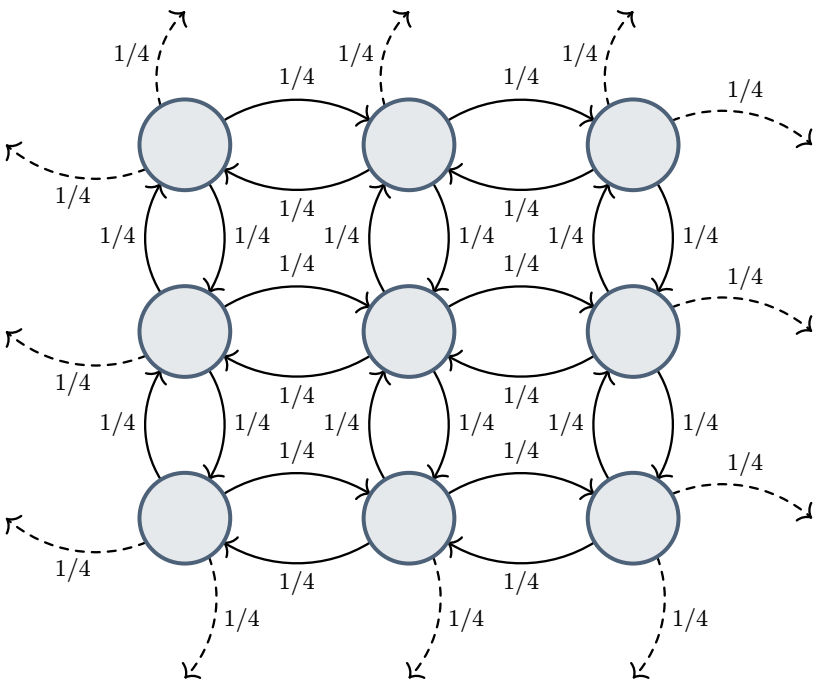


Figure 2.1: State transition diagram of a simple random walk in two-dimensional space.

Fun fact: drunkard's walk in higher dimensions

- **1D (drunken ant) and 2D (drunken human)**:
  - With probability 1, the drunkard returns to the starting point infinitely many times,
  - and will eventually visit any fixed location (given enough time).
  - Such states are called **recurrent**.
- **3D and higher (drunken bird)**:
  - With probability < 1, the drunkard ever returns to the starting point.
  - In fact, there is a nonzero chance they wander off and **never** come back.
  - Such states are called **transient**.

Warning

Adding an extra degree of freedom (moving in 3D instead of 2D) makes "getting lost forever" possible. This is one of the most famous **phase transitions** in stochastic models.

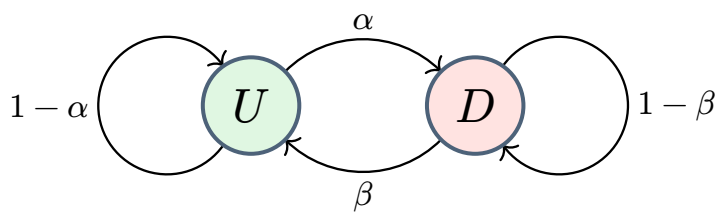
2.7. Long-run Behavior and Performance

2.7.1. Stationary Distributions

In Section 2.4, we forecast the machine's state using $\boldsymbol{\nu}^{(n)} = \boldsymbol{\nu}^{(0)} P^n$.

The same machine (Example 2.3)

$$P = \begin{bmatrix} 0.8 & 0.2 \\ 0.4 & 0.6 \end{bmatrix}, \quad (U, D).$$



? Question

After many days, how much does today's state matter? What fraction of days will the machine work, and what profit can it earn?

We will also return to **gambler's ruin**: the chance of reaching a target and the time it takes answer different operational questions.

The machine gradually forgets its starting state

Apply the same forecast rule to machines that start Up and Down. Only the Up probability is needed: the Down probability is its complement.

n	Up at day n, start Up	Up at day n, start Down
0	1.0000	0.0000
1	0.8000	0.4000
2	0.7200	0.5600
5	0.6701	0.6598
10	0.6667	0.6666

Both columns approach $\frac{2}{3}$. The candidate distribution is $\boldsymbol{\pi} = \left[\frac{2}{3}, \frac{1}{3}\right]$.

How can we find this distribution directly, without computing large powers of P?

Idea: Recall that $\boldsymbol{\nu}^{(n+1)} = \boldsymbol{\nu}^{(n)} P$. If $\boldsymbol{\nu}^{(n)}$ were to be unchanging, then $\boldsymbol{\nu}^{(n+1)} = \boldsymbol{\nu}^{(n)}$ and we would have $\boldsymbol{\nu}^{(n)} = \boldsymbol{\nu}^{(n)} P$.

A distribution that one more step leaves unchanged

Stationary distribution **Definition 2.13**

A probability **row vector** $\boldsymbol{\pi}$ is stationary if

$$\boldsymbol{\pi} = \boldsymbol{\pi} P, \quad \pi_j \geq 0, \quad \sum_j \pi_j = 1.$$

Think of a large collection of machines. If the fraction in each state is $\boldsymbol{\pi}$, one transition redistributes them into the **same proportions**.

Starting with $\boldsymbol{\nu}^{(0)} = \boldsymbol{\pi}$ therefore gives

$$\boldsymbol{\nu}^{(n)} = \boldsymbol{\pi} P^n = \boldsymbol{\pi} \quad \text{for every } n \geq 0.$$

Individual machines still fail and recover. **Stationary** describes their distribution.

Calculating the machine's stationary distribution

Write $\boldsymbol{\pi} = [\pi_U, \pi_D]$ in the order (Up, Down). By definition,

$$[\pi_U, \pi_D] = [\pi_U, \pi_D] \begin{bmatrix} 0.8 & 0.2 \\ 0.4 & 0.6 \end{bmatrix}.$$

Equating components and requiring probabilities to sum to one gives

$$\begin{cases} \pi_U = 0.8\pi_U + 0.4\pi_D \\ \pi_D = 0.2\pi_U + 0.6\pi_D \\ \pi_U + \pi_D = 1. \end{cases}$$

The first two equations are equivalent. Substitute $\pi_D = 1 - \pi_U$ into the first:

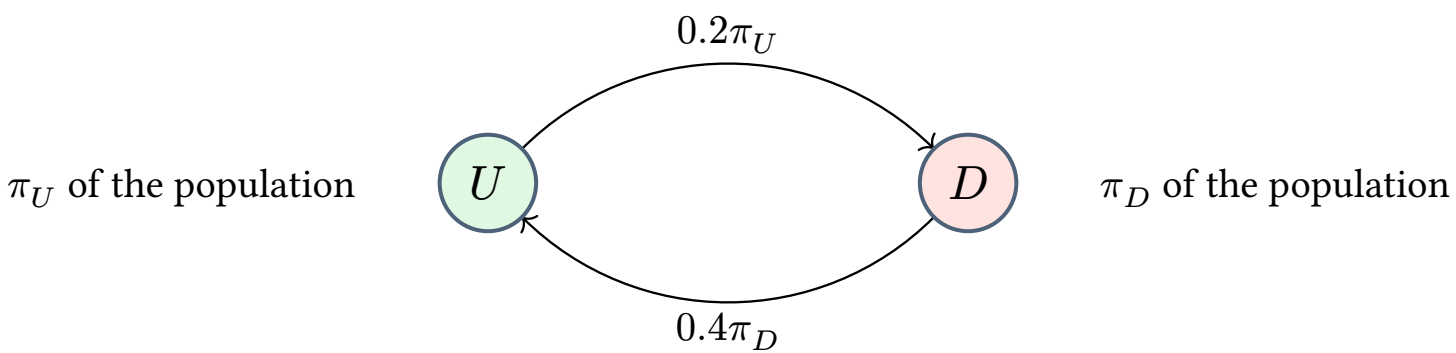
$$\pi_U = 0.8\pi_U + 0.4(1 - \pi_U) \quad \Rightarrow \quad \boldsymbol{\pi} = \left[\frac{2}{3}, \frac{1}{3}\right].$$

What keeps these proportions unchanged? The Up equation tells us.

The stationary equation balances probability flows

In the Up equation, $0.8\pi_U$ stays Up and $0.4\pi_D$ enters from Down. Subtract the part that stays:

$$\pi_U = 0.8\pi_U + 0.4\pi_D \quad \Leftrightarrow \quad \underbrace{0.2\pi_U}_{\text{out of Up}} = \underbrace{0.4\pi_D}_{\text{into Up}}.$$

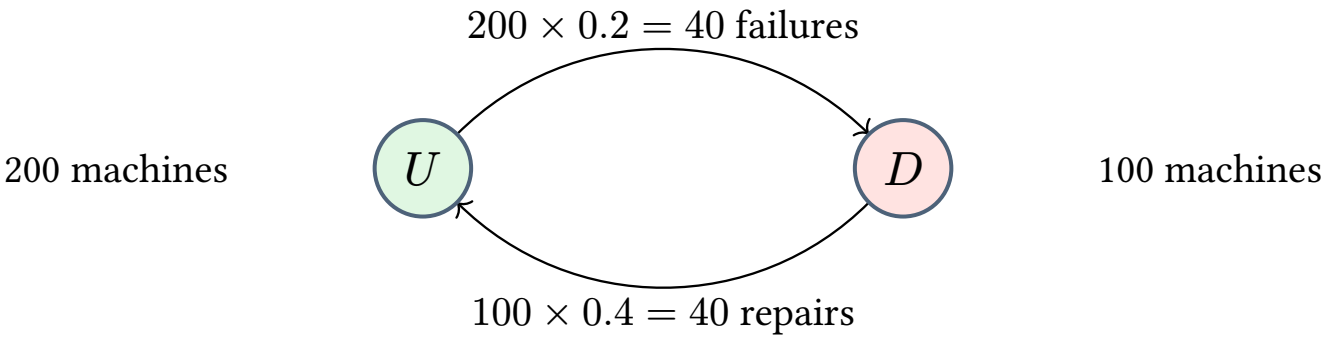


Each arrow carries **source probability × transition probability**, using the multiplication rule from our one-step forecasts.

Flow balance is the stationarity equation rearranged: Up receives exactly as much probability as it loses.

Balanced flows preserve the proportions

At the proportions we calculated, the **expected moves per step** among 300 machines are:



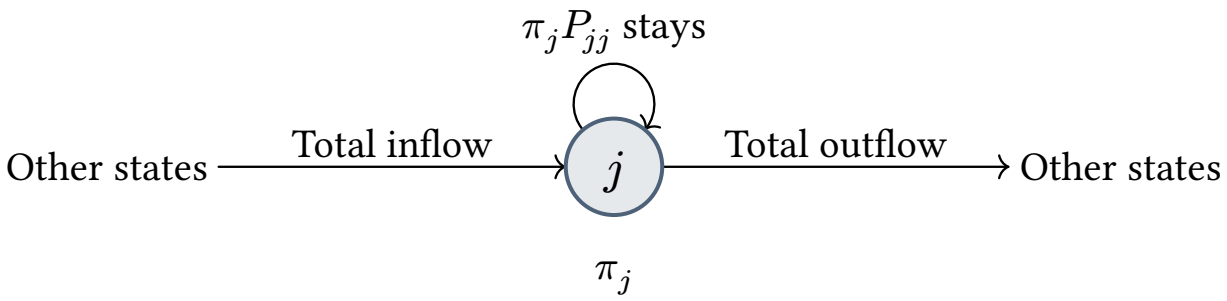
$$\text{Expected Up tomorrow} = 200 + \underbrace{40}_{\text{inflow}} - \underbrace{40}_{\text{outflow}} = 200.$$

With 150 machines in each state, there would instead be 60 expected repairs and 30 failures. Up would gain 30 machines in expectation. **Unequal flows change the proportions.**

Individual machines still move, and actual counts can fluctuate.

Principle to remember

Stationarity is flow balance at every state



$$\underbrace{\sum_{i \neq j} \pi_i P_{ij}}_{\text{inflow}} = \underbrace{\pi_j \left(1 - P_{jj}\right)}_{\text{outflow}}.$$

Subtracting the self-loop from $\pi_j = \sum_i \pi_i P_{ij}$ gives this balance.

Failure and repair probabilities determine the balance

For the earlier machine model, let $0 < \alpha, \beta < 1$:

$$P = \begin{bmatrix} 1-\alpha & \alpha \\ \beta & 1-\beta \end{bmatrix}.$$

Balancing $\alpha\pi_U = \beta\pi_D$ and normalizing gives

$$\pi_U = \frac{\beta}{\alpha+\beta}, \quad \pi_D = \frac{\alpha}{\alpha+\beta}.$$

Interpretation: increasing β while holding α fixed shifts stationary probability toward Up.

Multiplying **both** α and β by the same positive factor c leaves $\boldsymbol{\pi}$ unchanged, provided the probabilities stay valid.

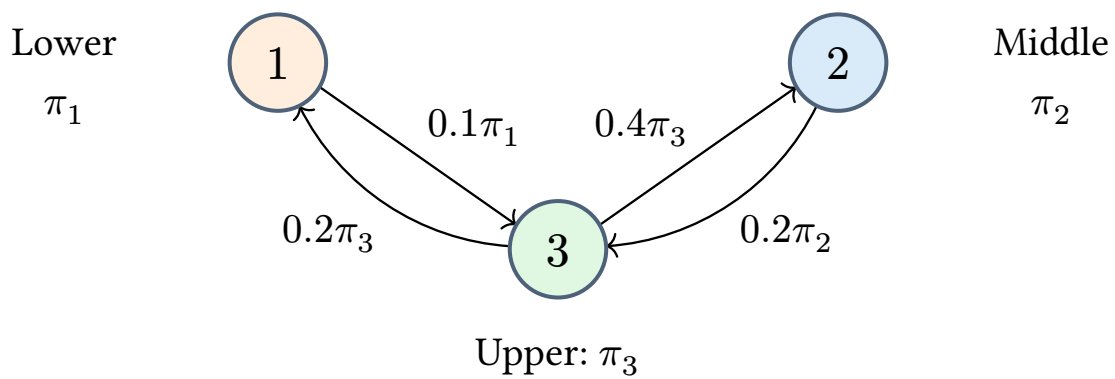
Stationary probabilities describe how time is divided between states. They do not determine the frequency or duration of outages.

Waiting times are studied by first-step analysis in Section 2.7.4.

Social mobility: several routes feed one state

Example 2.15

Return to Example 2.14. Only transitions involving upper class 3 are shown, labeled by **probability flow per generation**.



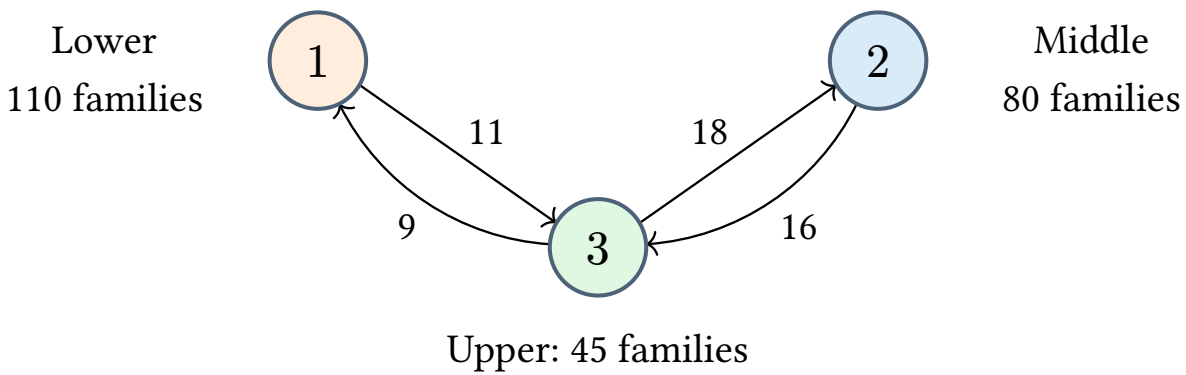
Total inflow to class 3 equals total outflow:

$$0.1\pi_1 + 0.2\pi_2 = (0.2 + 0.4)\pi_3.$$

The self-loop $0.4\pi_3$ stays inside class 3 and contributes to neither side.

Total balance does not require opposite arrows to match

The social-mobility equations give $\boldsymbol{\pi} = \left[\frac{22}{47}, \frac{16}{47}, \frac{9}{47}\right]$ (calculation: Page 481). Use 235 families in these proportions. Arrows show **expected moves involving class 3** per generation.



Class 3 receives $11 + 16 = 27$ families and loses $9 + 18 = 27$, in expectation. Yet the opposing flows between classes 1 and 3 are 11 and 9.

Global balance matches the totals at each state. Requiring $\pi_i P_{ij} = \pi_j P_{ji}$ for every pair is a stronger condition.

Pairwise balance and its interpretation are discussed in the appendix on Page 492–Page 496.

Stationarity and convergence answer different questions

Stationarity: which initial distribution stays unchanged?

$$\boldsymbol{\pi} = \boldsymbol{\pi} P.$$

Limiting distribution **Definition 2.14**

For an initial distribution $\boldsymbol{\nu}^{(0)}$, a distribution $\boldsymbol{\pi}$ is a **limiting distribution** if

$$\boldsymbol{\nu}^{(0)} P^n \to \boldsymbol{\pi} \quad (n \to \infty).$$

The limit may depend on $\boldsymbol{\nu}^{(0)}$.

For a finite chain, taking limits in $\boldsymbol{\nu}^{(n+1)} = \boldsymbol{\nu}^{(n)} P$ shows that any distributional limit must satisfy $\boldsymbol{\pi} = \boldsymbol{\pi} P$.

Does a stationary distribution always attract every starting distribution?

Alternation prevents convergence

A stationary distribution without convergence

Example 2.16

Set failure and repair probabilities to 1 in the machine model:

$$P = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad \pi = \left[\frac{1}{2}, \frac{1}{2}\right].$$

Starting Up gives

Day	State	Distribution
Even	Up	$[1, 0]$
Odd	Down	$[0, 1]$

The stationary distribution is unique, but the forecast keeps oscillating. Starting with a 50–50 mixture would preserve that mixture at every step.

Gambler's ruin remembers the starting state

Several stationary distributions — Example 2.17

Recall Example 2.4 with states $0, 1, 2, 3, 4$ and $0 < p < 1$. States 0 and 4 are absorbing.

Any mixture of the two absorbing states stays unchanged:

$$\boldsymbol{\pi}_a = [a, 0, 0, 0, 1-a], \quad 0 \le a \le 1.$$

Starting inside, the game eventually ends at 0 or 4. The final weights depend on the probability of reaching each endpoint.

Starting at 0 and starting at 4 already give different limits. The limiting distribution **depends on the initial state**.

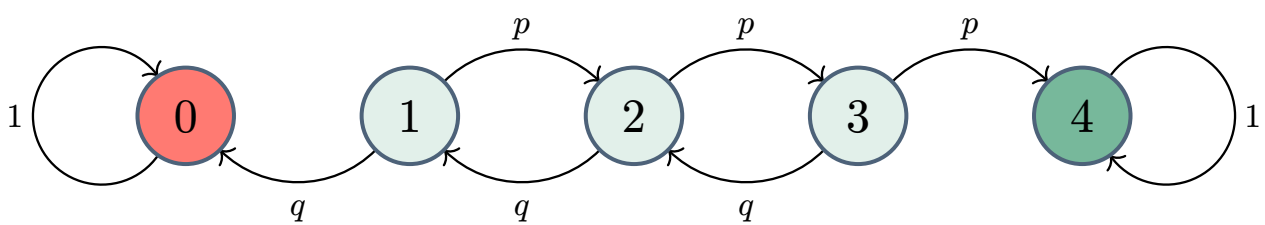
Later, first-step analysis will give the exact weights for each starting capital.

Communication identifies the possible traps

Recall Definition 2.12: two states communicate when each can reach the other.

Communicating and closed classes — Definition 2.15

A **communicating class** is a maximal group of mutually communicating states. A class is **closed** if no transition can leave it.



The classes are $\{0\}$, $\{1, 2, 3\}$, and $\{4\}$. The endpoint classes are closed. The interior class can be left forever.

This connects to Section 2.6: in a **finite** chain, states in closed classes are recurrent, and states outside the closed classes are transient.

Irreducibility keeps all states in one class

Irreducible chain **Definition 2.16**

A chain is **irreducible** when every state can reach every other state with positive probability. Its state space is one communicating class.

Earlier model	Check using positive-probability paths
Machine	Up can reach Down, and Down can reach Up. **Irreducible.**
Social mobility	All one-step probabilities are positive. **Irreducible.**
Gambler's ruin	Once at an endpoint, the chain cannot return inside. **Reducible.**

Only reachability matters for this check. An edge with a very small positive probability still counts.

Periodicity restricts the possible return times

The alternating machine can return to Up only after $2, 4, 6, \ldots$ steps. Its period is 2.

Period and aperiodicity — Definition 2.17

For a state i with a possible return, its period is

$$d(i) = \gcd\left\{n \geq 1 : P_{ii}^{(n)} > 0\right\}.$$

Period 1 means **aperiodic**. In an irreducible chain, all states share the same period.

Quick check: in an irreducible chain, one self-loop $P_{ii} > 0$ makes the period 1. Our original machine has $P_{UU} = 0.8$, so it is aperiodic.

A self-loop is sufficient. A chain can also be aperiodic without one.

A useful convergence theorem for finite chains

Stationarity and convergence **Theorem 2.3**

For a DTMC on a **finite** state space:

1. **Irreducible:** there is a unique stationary distribution $\boldsymbol{\pi}$, and every $\pi_j > 0$.
2. **Irreducible and aperiodic:** for every initial distribution,

$$\boldsymbol{\nu}^{(0)} P^n \to \boldsymbol{\pi}.$$

The limit is **independent of the initial distribution**.

The original machine and the social mobility model satisfy both conditions. The alternating machine satisfies only the first.

These are **sufficient conditions**. If they fail, inspect the chain before deciding what happens. The finite-state assumption matters: *our walk on all integers is outside this theorem* (state space is not finite).

The inventory model has a limiting distribution

Recall the $(s, S) = (1, 5)$ policy from Section 2.3. End-of-day stock is $X_n \in \{0, 1, 2, 3, 4, 5\}$, and demands $0, 1, 2, 3$ all have positive probability.

A reachability check using the policy:

- From any stock level, suitable demands lead to 0 or 1.
- From 0 or 1, replenishment followed by zero demand leads to 5.
- From 5, demands can lead to $2, 3, 4$ directly, and then to $0, 1$.

Thus every state communicates with 5. The chain is **irreducible**. Also, $P_{5,5} = 0.3 > 0$, so it is **aperiodic**.



Conclusion

The policy has a unique stationary distribution, and the stock distribution converges to it regardless of initial stock. We established this without computing P^n.

Stationary proportions describe time averages

A forecast $\mathbb{P}(X_n = j)$ describes a **specified time**. Along one sample path, the fraction of visits to state j is

$$\hat{\pi}_j^{(n)} = \frac{1}{n} \sum_{k=0}^{n-1} \mathbb{1}\{X_k = j\}.$$

Time averages **Theorem 2.4**

For a **finite irreducible** DTMC with stationary distribution $\boldsymbol{\pi}$, from any starting state,

$$\hat{\pi}_j^{(n)} \to \pi_j \quad \text{with probability 1, for every state } j.$$

Aperiodicity is not required.

For the alternating machine, the Up probability alternates between 1 and 0, but the fraction of days Up tends to $\frac{1}{2}$.

A snapshot may oscillate even when a time average settles down.

Stationary proportions give average rewards

Let r_i be the reward earned per day in state i. Grouping days by state gives

$$\frac{1}{n}\sum_{k=0}^{n-1} r_{X_k} = \sum_i \hat{\pi}_i^{(n)} r_i \to \sum_i \pi_i r_i.$$

Long-run average reward **Definition 2.18**

For a finite irreducible chain,

$$g = \sum_i \pi_i r_i = \boldsymbol{\pi}\boldsymbol{r},$$

where $\boldsymbol{r}$ is the reward column vector. The sample average converges to g with probability 1.

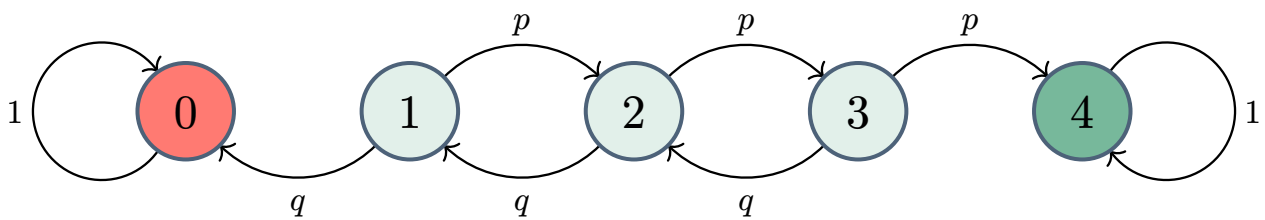
Our machine has $\boldsymbol{\pi} = \left[\frac{2}{3}, \frac{1}{3}\right]$, earns \$1200 per Up day, and loses \$300 per Down day:

$$g = \frac{2}{3}(1200) + \frac{1}{3}(-300) = 700 \quad \text{dollars per day.}$$

This average counts **all days**. A question about what happens **before a target** requires a stopping rule and first-step analysis.

Gambler's ruin: success and duration

Recall the game on $\{0, 1, 2, 3, 4\}$, with $0 < p < 1$ and $q = 1 - p$.



Recall the **hitting time** from Definition 2.9. For a target set A, write $\tau_A = \inf\{n \geq 0 : X_n \in A\}$. Write τ_j for the first visit to state j. Starting in A gives $\tau_A = 0$.

? Question

Starting at i, what is the chance of **reaching 4 first**? How many rounds until **either endpoint**?

First-step analysis: define the unknown and its boundary values, then condition on the next move. Given the next state, the Markov property lets us restart there.

Hitting probability: average the next-state chances

Write $\mathbb{P}_i$ for probability starting at i. Define

$$h_i = \mathbb{P}_i(\tau_4 < \tau_0), \quad h_0 = 0, \quad h_4 = 1.$$

First-step equation for success **Proposition 2.4**

After one move, success has chance h_{i-1} or h_{i+1}:

$$h_i = qh_{i-1} + ph_{i+1} \quad (i = 1, 2, 3).$$

For a fair game, $p = q = \frac{1}{2}$, so

$$h_{i+1} - h_i = h_i - h_{i-1}.$$

Equal increments between $h_0 = 0$ and $h_4 = 1$ give

$$(h_0, h_1, h_2, h_3, h_4) = \left(0, \frac{1}{4}, \frac{1}{2}, \frac{3}{4}, 1\right).$$

There is no extra 1: a move selects the remaining success chance. For fair play with endpoints $0, N$, the same reasoning gives $h_i = \frac{i}{N}$.

Expected hitting time: one round plus the remainder

Write $\mathbb{E}_i$ for expectation starting at i. Stop at $A = \{0, 4\}$ and let

$$t_i = \mathbb{E}_i[\tau_A], \quad t_0 = t_4 = 0.$$

First-step equation for duration **Proposition 2.5**

One round passes before the remaining wait begins:

$$t_i = 1 + qt_{i-1} + pt_{i+1} \quad (i = 1, 2, 3).$$

For fair play, symmetry gives $t_1 = t_3$. Thus

$$t_1 = 1 + \frac{1}{2}t_2, \quad t_2 = 1 + \frac{1}{2}(t_1 + t_3) = 1 + t_1.$$

Solving gives

$$t_1 = t_3 = 3, \quad t_2 = 4 \quad \text{rounds.}$$

The extra 1 counts the round just played. At an endpoint, no rounds remain. The transition probabilities are the same as for h_i; the quantity and boundary values differ.

Success, duration, and the limiting distribution

For the fair game, the same first-step method gives:

Start	Chance of reaching 4 first	Expected rounds to an endpoint
1	$\frac{1}{4}$	3
2	$\frac{1}{2}$	4
3	$\frac{3}{4}$	3

More capital raises the success chance. **Expected duration is largest in the middle.**

The game eventually stays at one endpoint. Hence the probabilities h_i also determine the limit promised earlier:

$$\mathbb{P}_i(X_n = \cdot) \to [1 - h_i, 0, 0, 0, h_i].$$

Starting at 2 gives the limiting distribution $\left[\frac{1}{2}, 0, 0, 0, \frac{1}{2}\right]$. Along **one run**, the fraction of time at 4 tends to 0 or 1. The chain is reducible, so the irreducible time-average theorem does not apply.

First-step equations for a target set

For disjoint targets A, B, set the boundary values first. Outside each row's stopping set, condition on the next state:

Unknown	First-step equation	Boundary values
$h_i = \mathbb{P}_i(\tau_A < \tau_B)$	$h_i = \sum_j P_{ij} h_j$	$h = 1$ on A; $h = 0$ on B
$t_i = \mathbb{E}_i[\tau_A]$	$t_i = 1 + \sum_j P_{ij} t_j$	$t = 0$ on A
Expected cost v_i before A	$v_i = c_i + \sum_j P_{ij} v_j$	$v = 0$ on A

Charge a finite cost c_i **before** each transition and stop charging on entering A. Taking $c_i = 1$ makes accumulated cost equal to elapsed time.

For a finite chain that reaches the relevant stopping set with probability 1, these equations have a unique solution: use $A \cup B$ for h, and A for t, v. The assumption must hold from every starting state.

A target need not be absorbing: we stop at its first visit. Define the quantity, the stopping rule, and the contribution from the first step.

3. Poisson Processes

3.1. Events in Continuous Time

A coin-toss round, a day of machine operation, an inventory review: our DTMC models move in **fixed steps**. Requests and failures need not wait for the next tick of that clock.



? A new prediction question

How many events will occur in a time window, and when will the next one occur?

The gateway receives requests, not time slots

An AI-service gateway records each incoming request before routing it for processing. We will first model **the arrival record**, without modeling service or routing.

Model component	Our choice
Event	One request reaches the gateway.
Clock	Elapsed time, measured in seconds or minutes.
Recorded state	The total number of requests received so far.
Question	How many requests arrive in the next ten seconds?



Model boundary

A completed request still belongs to the arrival record. Cumulative arrivals are **not** the number waiting or being processed.

Record when each request arrives

Here is one illustrative record over 30 seconds. Each dot marks an arrival; the observation starts at time 0.



Let T_n be the **time of the nth arrival**, measured from the start, and set $T_0 = 0$. For this record,

$$T_1 = 2, \quad T_3 = 11, \quad T_6 = 29 \quad \text{seconds.}$$

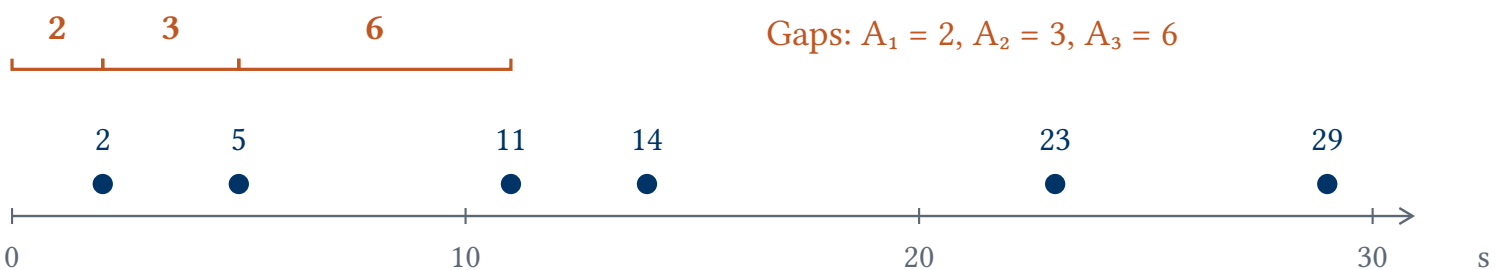
? Question

What does the subscript in T_3 count? What does its value measure?

The gaps tell the same story

The **interarrival time** A_n is the duration between arrivals $n-1$ and n:

$$A_n = T_n - T_{n-1}, \quad n \geq 1.$$



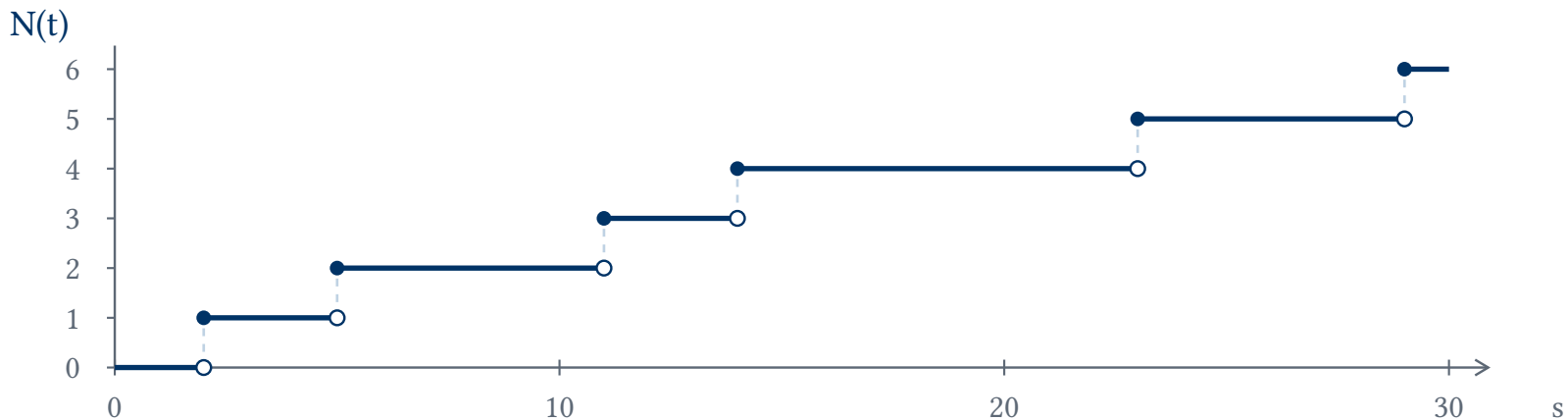
For this record, $A_1 = 2$, $A_2 = 3$, and $A_3 = 6$ seconds. Conversely, adding the gaps gives the arrival time:

$$T_n = A_1 + A_2 + \cdots + A_n.$$

An arrival time is measured **from the start**. A gap is measured **from the previous arrival**.

Count how many have arrived

$N(t)$ counts arrivals in $(0, t]$, including any arrival at time t.



$$N(10) = 2, \quad N(20) = 4, \quad N(T_3) = N(11) = 3.$$

Read the path

The integer count jumps by one at each arrival and stays constant between arrivals.

A window count is an increment

How many requests arrived **after time 10 and by time 20**?



Subtract the earlier cumulative count from the later one:

$$N(20) - N(10) = 4 - 2 = 2.$$

More generally, for $0 \le a < b$,

$$N(b) - N(a) = \text{number of arrivals in } (a, b].$$

Draw the observation window first. Its endpoints determine which events are counted.

Pause and recall

One record, three descriptions

The arrival times are $2, 5, 11, 14, 23, 29$ seconds.

1. Find $N(10)$ and $N(20) - N(10)$. State the time window in each case.
2. Find T_3 and A_3. Explain why their values differ.
3. At time 20, all received requests have finished processing. Is $N(20)$ now zero?

Pause here. Try each question before turning to the answers.

Answers and reasoning

One record, three descriptions

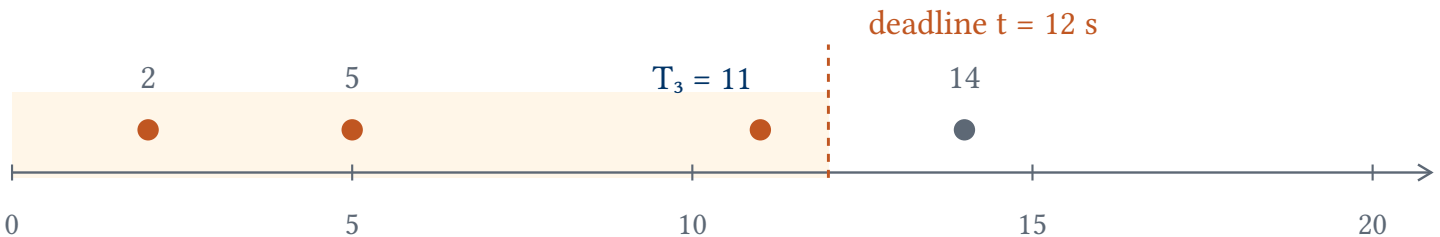
---

1. $N(10) = 2$ counts $(0, 10]$; $N(20) - N(10) = 2$ counts $(10, 20]$.
2. $T_3 = 11$ seconds from the start; $A_3 = 11 - 5 = 6$ seconds from the second arrival.
3. No. $N(20) = 4$ still counts every request received by that time. Completion does not erase an arrival.

The representation changes; the underlying arrival record does not.

One event, two descriptions

"The third request has arrived by time t" can be read in two ways.



The shaded window contains three arrivals.

For any positive integer n,

$$\{T_n \le t\} = \{N(t) \ge n\}.$$

Read the equivalence aloud

Arrival n occurs by time t **exactly when** the count by time t is at least n. This is true for the arrival record before we choose a probability model.

Principle to remember

Describe the event before choosing a formula

Start with the question in words, then identify the corresponding event.

At least three requests in ten seconds:

$$\{N(10) \geq 3\}.$$

The third request arrives within ten seconds:

$$\{T_3 \leq 10\} = \{N(10) \geq 3\}.$$

A different description can make the same probability easier to calculate.

3.2. A Model for Random Arrivals

3.2.1. Rates and Small Intervals

We now propose a model with a constant rate of **12 requests per minute** over the operating window of interest.

Interpret the rate **Poll 3.1**

Which statement is justified by this mean rate?

A). Every minute contains exactly 12 requests.
B). The expected count in a one-minute interval is 12.
C). Requests are always five seconds apart.
D). No gap between requests can exceed five seconds.

Explain your choice before calculating a probability.

A rate is an average, not a schedule

The answer is **B**. Individual counts and arrival times remain random. Write the rate and the window in compatible units:

$$\lambda = 12 \text{ per minute} = 0.2 \text{ per second}.$$

For a ten-second window, the expected count is

$$\lambda t = 12 \times \left(\frac{10}{60}\right) = 0.2 \times 10 = 2.$$

A rate

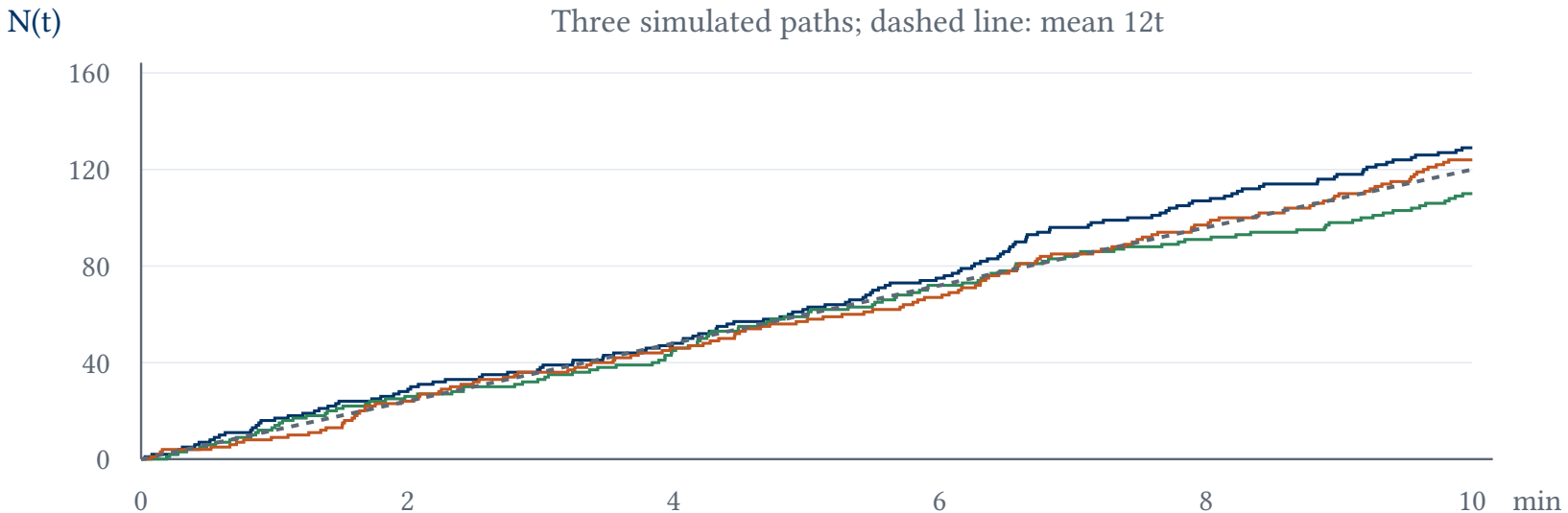
λ measures events per unit time.

A count parameter

λt is an expected number of events.

An expected count of 2 is **not** a probability and does not make the count equal to 2.

Random paths share a long-run rate



$$\mathbb{E}[N(t)] = 12t, \text{ with } t \text{ in minutes.}$$

Note how this differs from the random walk with a positive trend: the Poisson count is nondecreasing, while the random walk can go up or down.

A fresh chance of arrival

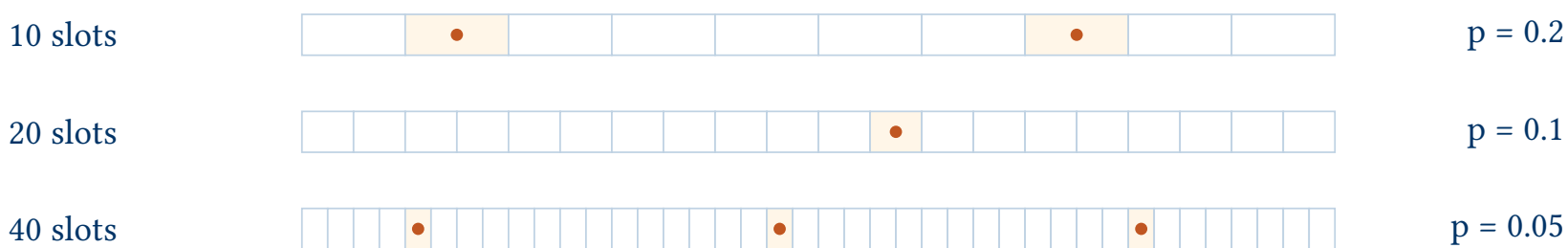
We want to model requests with a **fresh, steady chance of arrival** in each new short interval. This requires assumptions beyond the mean rate:

1. **Stable rate.** Equal short intervals have the same arrival chance. Doubling a very short interval roughly doubles that chance.
2. **History gives no advantage.** Earlier arrivals leave future chances unchanged. A recent request causes no burst or recovery period. A long silence does not make the next request due.
3. **Individual arrivals.** As an interval shrinks, the chance of multiple arrivals becomes negligible relative to the chance of one. Closely spaced arrivals remain possible.

A grid model for these arrivals

Split a window of length t into m short slots of length $h = \frac{t}{m}$. Individual arrivals give a **yes/no approximation**, the stable rate gives arrival chance $\approx \lambda h$, and the history assumption gives **independence**.

In the grid model, let each slot contribute one arrival with probability $p = \lambda h = \lambda \frac{t}{m}$. Choose m large enough that p is small.



Let B_m count occupied slots. The grid model gives exactly

$$B_m \sim \text{Binomial}\left(m, \lambda \frac{t}{m}\right).$$

Refine the grid without changing the mean

The mean stays fixed: $\mathbb{E}[B_m] = m \times \left(\lambda \frac{t}{m}\right) = \lambda t$.

For each fixed integer $k \geq 0$, as $m \to \infty$ (empty product $= 1$ for $k = 0$),

$$\begin{aligned}
\mathbb{P}(B_m = k) &= \binom{m}{k} \left(\lambda \frac{t}{m}\right)^k \cdot \left(1 - \lambda \frac{t}{m}\right)^{m-k} \\
&= \frac{(\lambda t)^k}{k!} \cdot \underbrace{\prod_{j=0}^{k-1} \left(1 - \frac{j}{m}\right)}_{\to 1} \cdot \underbrace{\left(1 - \frac{\lambda t}{m}\right)^m}_{\to \exp(-\lambda t)} \cdot \underbrace{\left(1 - \frac{\lambda t}{m}\right)^{-k}}_{\to 1} \\
&\to \exp(-\lambda t) \frac{(\lambda t)^k}{k!}.
\end{aligned}$$

The limiting model

Refining the grid removes the artificial slot size while preserving the mean. The assumptions built into the grid give a **Poisson** limiting count law.

The Poisson distribution describes a window count

Poisson distribution — Definition 3.1

A nonnegative integer-valued random variable K has distribution Poisson(θ) if

$$\mathbb{P}(K = k) = \exp(-\theta)\frac{\theta^k}{k!}, \quad k = 0, 1, 2, \cdots,$$

where $\theta > 0$ is its mean and $0! = 1$.

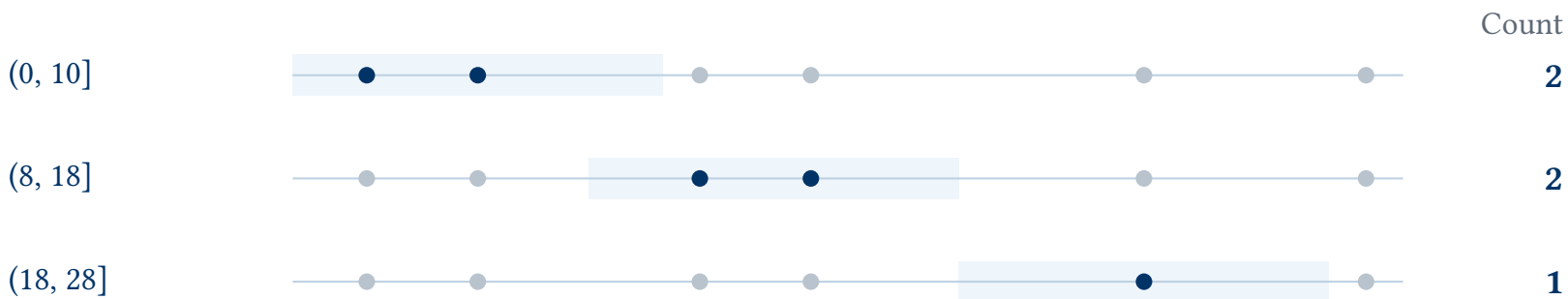
For a window of length t, our candidate parameter is $\theta = \lambda t$. With mean 2, the first three probabilities are

$$\mathbb{P}(K = 0) = \exp(-2), \quad \mathbb{P}(K = 1) = 2\exp(-2), \quad \mathbb{P}(K = 2) = 2\exp(-2).$$

We now state the assumptions about counts across different windows precisely.

Equal-length windows have the same law

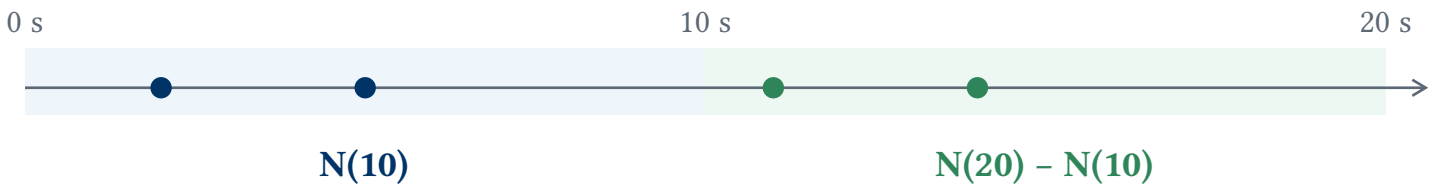
Our stable-rate model gives equal-length windows the same count distribution.



This is **stationary increments**: equal distributions, not equal realized counts.

Disjoint windows have independent counts

Our assumption that history leaves future chances unchanged becomes **independent increments**: counts in any collection of disjoint time intervals are independent.



Under this assumption, knowing the count in $(0, 10]$ does not change the distribution of the count in $(10, 20]$.

Two distinct assumptions

Stationary increments compare the laws of equally long windows. **Independent increments** describe the relationship between disjoint window counts.

The homogeneous Poisson process

Homogeneous Poisson process **Definition 3.2**

A counting process $\{N(t) : t \geq 0\}$ is a homogeneous Poisson process with rate $\lambda > 0$ if:

1. $N(0) = 0$;
2. it has independent increments;
3. for every $t \geq 0$ and $s > 0$,

$$N(t+s) - N(t) \sim \text{Poisson}(\lambda s).$$

The third condition specifies the count law and makes the increments stationary: the parameter depends on the **length** s, not the starting time t.

Throughout this chapter, an unqualified **Poisson process** means this homogeneous model.

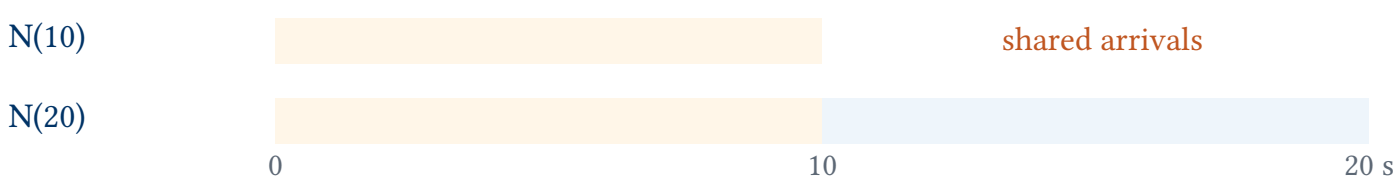
Independent increments, dependent totals?

The gateway follows a Poisson process. Time below is in seconds.

Which pair is independent? **Poll 3.2**

A). $N(10)$ and $N(20)$.
B). $N(10)$ and $N(20) - N(10)$.
C). Both pairs.
D). Neither pair.

Identify the two physical time windows in each pair.



Shared arrivals create dependence

B: $(0, 10]$ and $(10, 20]$ are disjoint. Cumulative counts share earlier arrivals:

$$N(20) = N(10) + \underbrace{N(20) - N(10)}_{\text{new arrivals}}.$$

For example, with $\lambda = 0.2$ per second,

$$\mathbb{P}(N(20) = 0) = \exp(-4) > 0,$$
$$\mathbb{P}(N(20) = 0 \mid N(10) = 1) = 0.$$



The same issue as the random walk

Independent tosses do not make successive positions independent. Independent new arrivals do not make cumulative counts independent.

In a short interval, one arrival is the main possibility

The Poisson count law makes our earlier short-interval intuition precise. As h approaches zero,

$$\mathbb{P}(N(h) = 0) = 1 - \lambda h + o(h),$$
$$\mathbb{P}(N(h) = 1) = \lambda h + o(h),$$
$$\mathbb{P}(N(h) \geq 2) = o(h).$$

Here $o(h)$ denotes a remainder whose ratio to h tends to zero. It is negligible **relative to h**.

Multiple arrivals are possible in a positive-length interval; their probability becomes smaller-order than h.

Later, CTMC transition rates will use this same short-time description.

Pause and recall

What makes the process Poisson?

---

1. Two ten-second windows have different observed counts. Does that contradict stationary increments?
2. Are counts in $(0, 20]$ and $(10, 30]$ independent under a positive-rate Poisson model? Explain using the windows.
3. Does $\mathbb{P}(N(h) \geq 2) = o(h)$ mean that two arrivals in a one-second interval are impossible?

Pause here. Try each question before turning to the answers.

Answers and reasoning

What makes the process Poisson?

---

1. No. Equal-length windows have the same **distribution**, not necessarily the same realized count.
2. No. Both counts include the random arrivals in $(10, 20]$.
3. No. The statement describes a limit as h tends to zero. For every $h > 0$, the probability of multiple arrivals is positive.

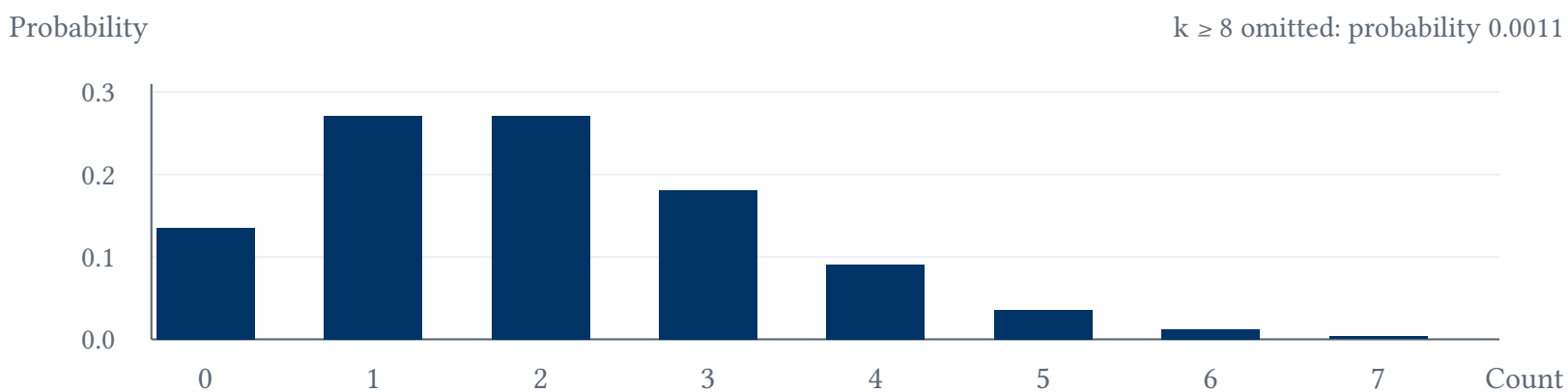
State the interval, the distributional claim, and the independence assumption separately.

3.3. Forecasting Counts

3.3.1. From Probabilities to a Decision

For a Poisson process, the counting random variable $N(t)$ has a Poisson distribution with parameter λt, hence mean and variance are

$$\mathbb{E}[N(t)] = \lambda t, \quad \mathrm{Var}(N(t)) = \lambda t.$$



The gateway's ten-second count has mean 2 and variance 2. Knowing the rate exactly still leaves **random variation in the count**.

No request in the next ten seconds

Use the proposed rate $\lambda = 12$ per minute. Let K count arrivals in the next ten-second window.

Model the count. Ten seconds is $\frac{1}{6}$ minute, so

$$K \sim \text{Poisson}\left(12 \times \frac{1}{6}\right) = \text{Poisson}(2).$$

Describe the event. No request means $K = 0$.

$$\mathbb{P}(K = 0) = \exp(-2) \approx 0.1353.$$

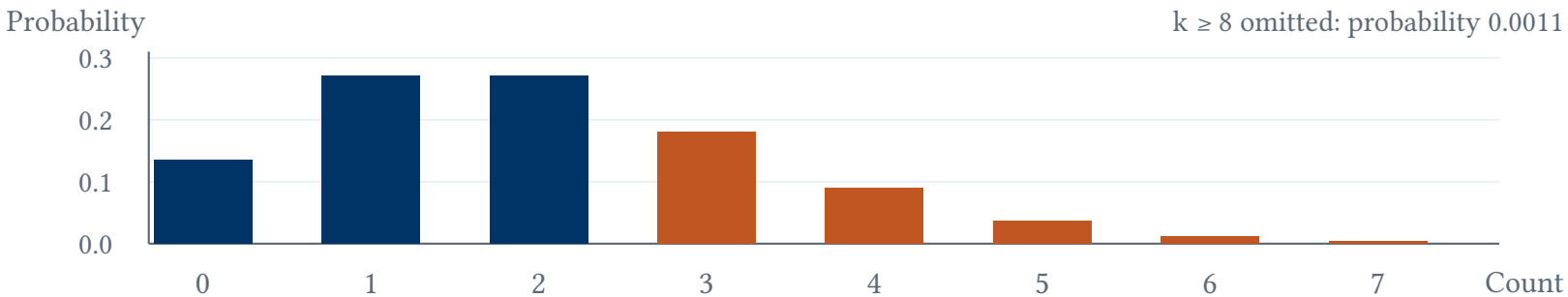


Interpretation

Under the model, about 13.5% of such windows contain no request, despite an expected count of 2.

At least three: count the easier complement

For the same $K \sim \text{Poisson}(2)$, "at least three" means $K \geq 3$, not $K = 3$.



$$\mathbb{P}(K \geq 3) = 1 - \sum_{k=0}^{2} \mathbb{P}(K = k) = 1 - \exp(-2)\left(1 + 2 + \frac{2^2}{2!}\right) \approx 0.3233.$$

The complement has only three cases: 0, 1, and 2.

Choose a request-count threshold

A monitor checks each fixed ten-second window and raises an alert when its count exceeds C. Under the baseline Poisson model, choose the **smallest integer C** with alert probability at most 5%.

C	2	3	4	5
$\mathbb{P}(K > C)$	0.3233	0.1429	0.0527	0.0166

Make and interpret the decision

Poll 3.3

Which threshold should be chosen? Does an alert prove that the arrival model is wrong?

The target is a **per-window count probability**, not a service-delay guarantee.

A tail probability becomes an operating rule

Choose $C = 5$: $C = 4$ still gives more than a 5% alert probability.

$$\mathbb{P}(K > 5) = 1 - \sum_{k=0}^{5} \exp(-2)\frac{2^k}{k!} \approx 0.0166.$$

Interpretation

About 1.66% of fixed ten-second windows trigger the alert even when the baseline model is correct. The integer threshold cannot in general achieve exactly 5%.

An alert is a reason to investigate, not proof that the model has failed. To predict response delay, we would also need service times and the current workload.

Which assumption needs checking?

The forecast used more than the mean rate. Connect each operational change to the assumption it challenges.

What changes?	What should we investigate?
Traffic rises predictably at noon.	Is a constant rate suitable for this window?
One request triggers follow-up requests.	Are disjoint window counts independent?
A client releases a batch simultaneously.	Are events arriving one at a time?

A few close arrivals on one path are possible under the Poisson model. Look for **repeated evidence** about the assumptions, not a visually perfect timeline.

Pause and recall

Model, calculate, interpret

Requests follow a Poisson process of rate 12 per minute.

1. What is the count distribution in 20 seconds? Give its mean and variance.
2. What is the probability of at least one request in five seconds?
3. Rewrite “the fifth request arrives within 20 seconds” as a count event. Does this tell us when that request finishes processing?

Pause here. Try each question before turning to the answers.

Answers and reasoning

Model, calculate, interpret

---

1. The parameter is $12 \times \frac{20}{60} = 4$. The distribution is $\text{Poisson}(4)$; its mean and variance are both 4.
2. The parameter is $12 \times \frac{5}{60} = 1$, so the probability is $1 - \exp(-1) \approx 0.6321$.
3. $\{T_5 \leq 20\} = \{N(20) \geq 5\}$. This concerns an arrival time, not a completion time; service is outside the present model.

Specify the event and the window, use the model, then interpret the result.

3.4. Waiting for Arrivals

3.4.1. From Counts to Waiting Times

The gateway still receives requests according to a Poisson process of rate λ. Let A_1 be the wait from time 0 to the first request.

An empty window and a long wait describe the same event:

$$\{A_1 > t\} = \{N(t) = 0\}, \quad \mathbb{P}(A_1 > t) = \exp(-\lambda t).$$

Exponential waiting time **Definition 3.3**

A nonnegative random variable A is $\mathrm{Exp}(\lambda)$, with **rate** $\lambda > 0$, if

$$\mathbb{P}(A > t) = \exp(-\lambda t), \quad t \geq 0.$$

The distribution function and density follow by complementing and differentiating:

$$F_A(t) = 1 - \exp(-\lambda t), \quad f_A(t) = \lambda \exp(-\lambda t), \quad t \geq 0.$$

The next gap starts a fresh exponential clock

The first gap A_1 is exponential. The same law applies to each later gap $A_n = T_n - T_{n-1}$.

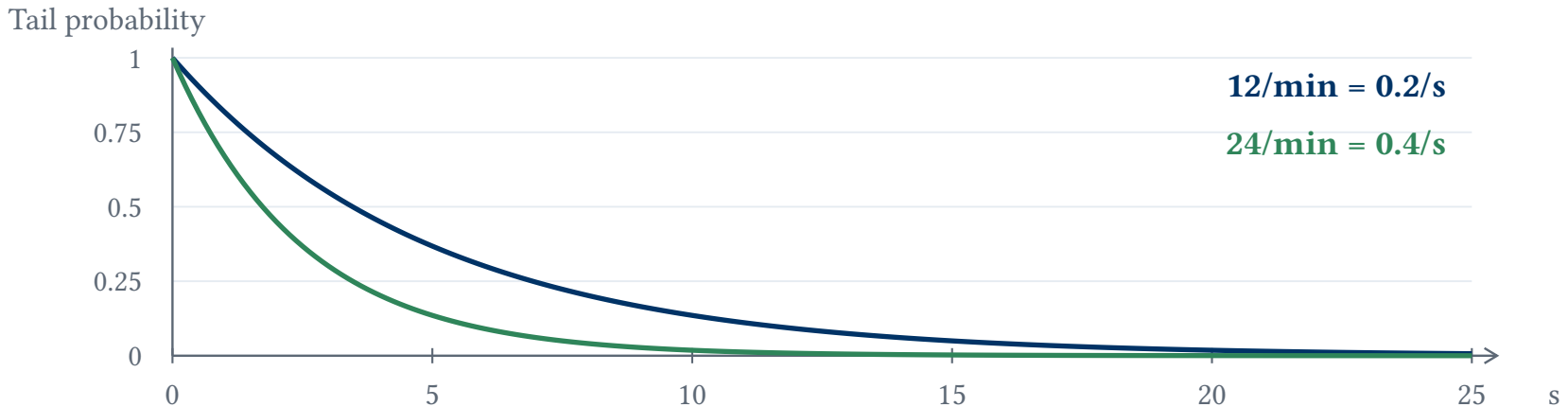
Interarrival representation — Proposition 3.1

A homogeneous Poisson process of rate λ has independent and identically distributed gaps

$$A_1, A_2, \cdots \sim \text{Exp}(\lambda).$$

Conversely, independent gaps with this law generate a Poisson process when we add them to obtain arrival times.

A higher rate makes long waits less likely



For any interarrival gap A_n,

$$\mathbb{E}[A_n] = \frac{1}{\lambda}, \quad \mathrm{Var}(A_n) = \frac{1}{\lambda^2}.$$

At $\lambda = 12$ per minute, the mean wait is $\frac{1}{12}$ minute, or **five seconds**. A mean of five seconds does not put an upper limit on the wait.

Ten seconds of silence: is a request now due?

Forecast the remaining wait **Poll 3.4**

The arrival rate is 12 per minute. Starting at time 0, you have observed **no request for ten seconds**.

Is the expected additional wait now:

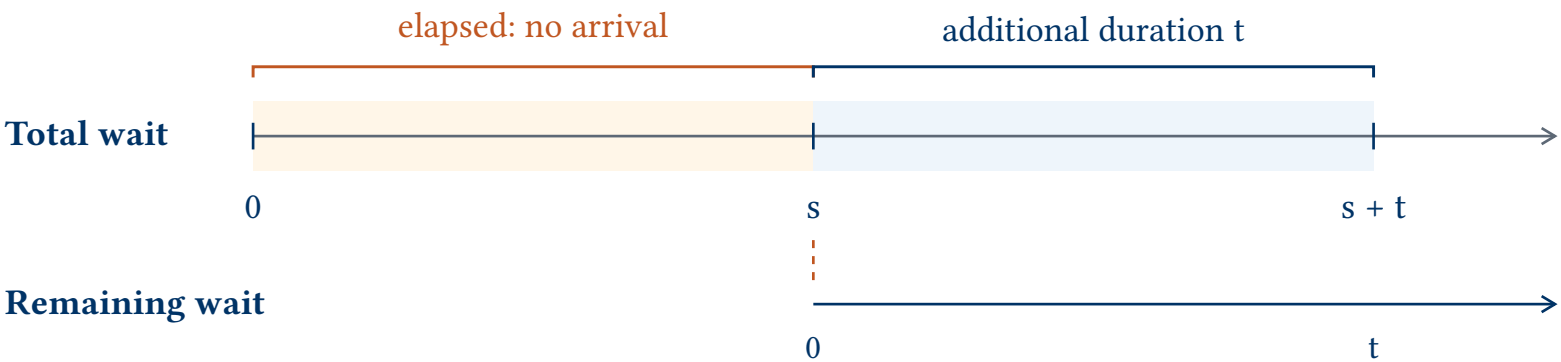
A). Less than five seconds?

B). Exactly five seconds?

C). More than five seconds?

Explain which information the ten-second silence gives you under the proposed model. Keep the **additional wait** separate from the **total wait from time 0**.

Memorylessness concerns the remaining duration



Memorylessness **Theorem 3.1**

For $A \sim \text{Exp}(\lambda)$ and $s, t \geq 0$,

$$\mathbb{P}(A > s + t \mid A > s) = \mathbb{P}(A > t) = \exp(-\lambda t).$$

Given survival through time s, the **remaining wait** $A - s$ has the same distribution as a fresh exponential wait. The elapsed time s has not disappeared from the total duration.

Conditioning explains the unchanged forecast

Since $\{A > s + t\}$ is contained in $\{A > s\}$,

$$\mathbb{P}(A > s + t \mid A > s) = \frac{\mathbb{P}(A > s + t)}{\mathbb{P}(A > s)} = \frac{\exp(-\lambda(s + t))}{\exp(-\lambda s)} = \exp(-\lambda t).$$

The poll answer is **B**. With time measured in seconds,

$$\mathbb{E}[A - 10 \mid A > 10] = 5, \quad \mathbb{E}[A \mid A > 10] = 15.$$

For example, the chance of waiting **at least another five seconds** is still $\exp(-1) \approx 0.3679$.

The bus forecast depends on the model

You saw a bus depart at time 0. Fifteen minutes later, no further bus has arrived. Compare two idealized services with the same mean gap of 20 minutes.

Model	What determines the next gap?	Mean remaining wait
Regular service	Every gap is 20 minutes.	5 minutes
Poisson service	Independent exponential gaps with mean 20 minutes.	20 minutes



The mean alone does not determine the forecast

A schedule makes elapsed time informative. An exponential gap has the memoryless property.

Here we know when the previous bus departed. Arriving at a random clock time is a different sampling question, revisited with renewal processes.

Should elapsed age be part of the state?

In the DTMC chapter, “machine Up” was not always enough information for prediction. The question was whether its age changes the failure forecast.

Exponential lifetime

Conditional on still working, the remaining lifetime has the same law at every age. Age can be omitted when it affects no other part of the model.

Age-dependent lifetime

If age changes the remaining-lifetime distribution, “Up” alone loses predictive information. Record age, or choose another adequate state.

Memorylessness can simplify the state.

Pause and recall

Remaining and total waiting times

Use $\lambda = 0.2$ per second and $A \sim \text{Exp}(\lambda)$.

1. Find $\mathbb{P}(A > 15 \mid A > 10)$. Which duration is five seconds?
2. Find the expected remaining wait and the expected total wait given $A > 10$.
3. A component has a fixed lifetime of 20 hours. After 15 hours of operation, is its remaining lifetime memoryless?

Pause here. Try each question before turning to the answers.

Answers and reasoning

Remaining and total waiting times

---

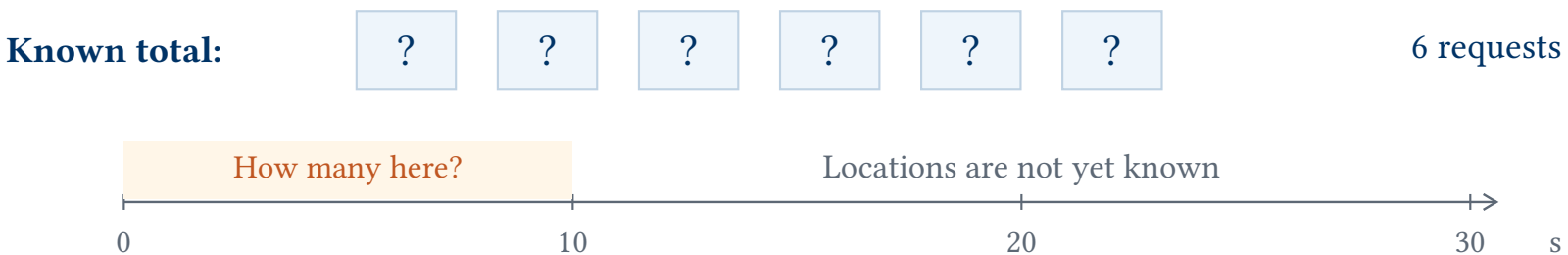
1. $\exp(-0.2 \times 5) = \exp(-1) \approx 0.3679$. The five seconds are **additional** to the first ten.
2. The expected remaining wait is 5 seconds; the expected total wait is $10 + 5 = 15$ seconds.
3. No. It has exactly five hours left. Age changes the forecast.

Condition on what has been observed, then specify whether the duration is total or remaining.

3.5. Arrivals Given a Total

3.5.1. What Does a Count Tell Us About Locations?

A log reports **exactly six requests in 30 seconds**, but their timestamps are hidden. The total is now known; the locations remain random.



> **Use the new information** **Poll 3.5**
>
> Is the number in the first ten seconds still Poisson(2)? Could it exceed six?

The same arrival model can give a different forecast after conditioning on observed information.

With one arrival, every location is equally likely

First suppose $N(T) = 1$. For $0 \le u \le T$, that arrival is in $(0, u]$ exactly when the two disjoint counts are 1 and 0. Recall that for every $t \ge 0$ and $s > 0$,

$$N(t+s) - N(t) \sim \text{Poisson}(\lambda s).$$

$$\begin{aligned}
\mathbb{P}(T_1 \le u \mid N(T) = 1) &= \frac{\mathbb{P}(N(u) = 1)\mathbb{P}(N(T) - N(u) = 0)}{\mathbb{P}(N(T) = 1)} \\
&= \frac{\lambda u \exp(-\lambda u) \exp(-\lambda(T-u))}{\lambda T \exp(-\lambda T)} \\
&= \frac{u}{T}.
\end{aligned}$$

Thus $T_1 \mid \{N(T) = 1\}$ is uniform on $(0, T)$.

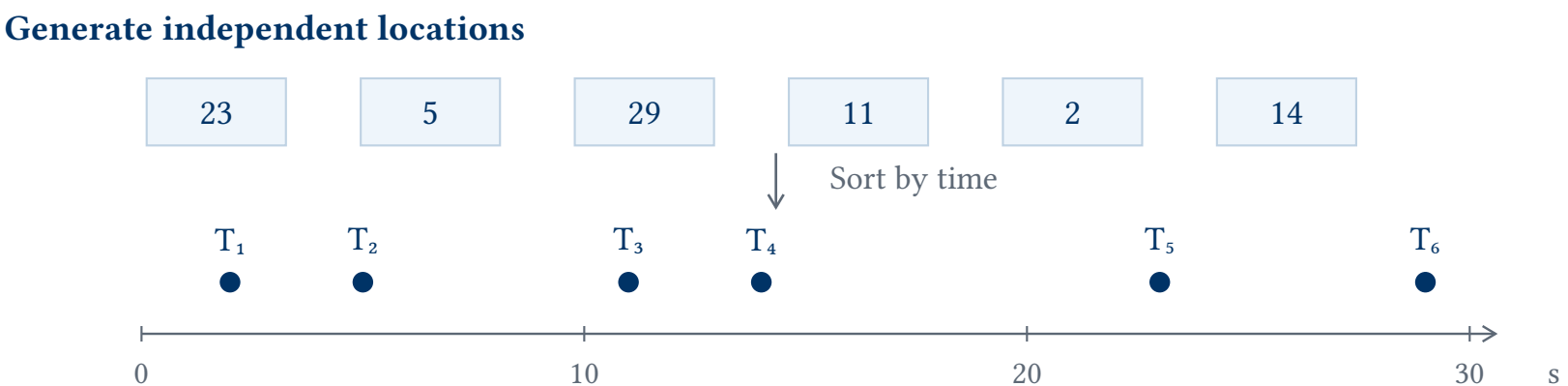
The rate cancels because we have already fixed the total count. Without that condition, T_1 is exponential, not uniform.

With several arrivals, generate locations and then sort

Conditional arrival locations

Proposition 3.2

Given $N(T) = n$, generate n independent $\text{Uniform}(0, T)$ points and sort them. The sorted points have the same joint distribution as $(T_1, \cdots, T_n)$ under this condition.



The **generated locations before sorting** are independent. The ordered arrival times satisfy $T_1 < \cdots < T_n$ and are not independent when $n \geq 2$.

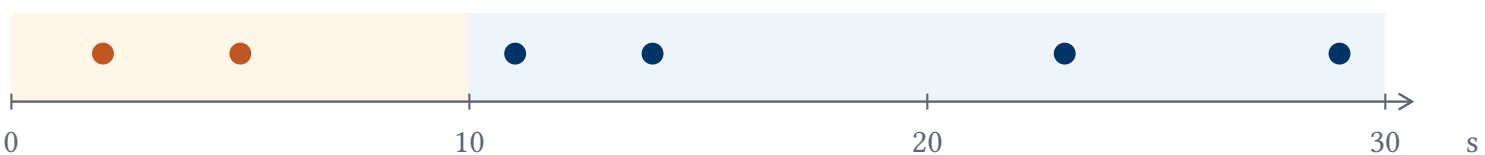
Proof: see the appendix on Page 486.

A subwindow count becomes binomial

Given $N(T) = n$, each of the n independent uniform locations falls in $(a, b] \subseteq (0, T]$ with probability $\frac{b-a}{T}$.

$$N(b) - N(a) \mid \{N(T) = n\} \sim \text{Binomial}\left(n, \frac{b-a}{T}\right).$$

Each uniform location falls in the first third with probability 1/3.



For six requests in 30 seconds, the first-ten-second count has distribution

$$N(10) \mid \{N(30) = 6\} \sim \text{Binomial}\left(6, \frac{1}{3}\right).$$

The answer to the poll is **no**: this conditional count cannot exceed six.

The mean can stay the same while the distribution changes

Let $K = N(10)$ with time in seconds and rate 0.2 per second. Compare the forecast before and after learning $N(30) = 6$.

Information	Law of K	Mean	Variance
No total supplied	$\text{Poisson}(2)$	2	2
Exactly six in 30 seconds	$\text{Binomial}\left(6, \frac{1}{3}\right)$	2	$\frac{4}{3}$

For example, after learning the total,

$$\mathbb{P}(K = 0 \mid N(30) = 6) = \left(\frac{2}{3}\right)^6 \approx 0.0878,$$

whereas without that information $\mathbb{P}(K = 0) = \exp(-2) \approx 0.1353$.

The conditional and unconditional problems happen to have the same mean here, but not the same law.

The earliest arrival is not a uniform location

What does sorting change? **Poll 3.6**

Given six requests in 30 seconds, is the chance that the **first** request occurs in the first ten seconds equal to $\frac{1}{3}$?

No. The first request is in that window if **at least one** of the six uniform points is there:

$$\mathbb{P}(T_1 \leq 10 \mid N(30) = 6) = 1 - \left(\frac{2}{3}\right)^6 \approx 0.9122.$$

A randomly selected request has a uniform location. The first request is selected by being **earliest**, not by a random choice of label.



Selection changes the distribution

Ask how the object was selected before assigning it a probability law.

Once the total is fixed, the windows share a constraint

Let $X = N(10)$ and $Y = N(30) - N(10)$. Without further information, these are independent counts in disjoint windows.

After conditioning on $N(30) = 6$,

$$X + Y = 6.$$

If $X = 4$, then $Y = 2$. A larger first-window count leaves fewer requests for the second window.

Before the total is supplied

$X \sim \text{Poisson}(2)$ and $Y \sim \text{Poisson}(4)$ are independent.

After the total is supplied

$X \sim \text{Binomial}\left(6, \frac{1}{3}\right)$ conditionally, and $Y = 6 - X$.

Principle to remember

State what is independent and what is known

Name the random quantities and the information on which you condition.

Before the total is known: disjoint Poisson window counts are independent.

Given the total: those counts must add to a fixed number.

A conditional question is a new probability question. Write the condition before reusing a familiar distribution.

Pause and recall

Arrival locations given a count

Six requests arrived during a 30-second window.

1. What is the conditional count distribution in the first 15 seconds?
2. What is the probability that all six requests arrived in the second half?
3. Are the six ordered arrival times independent uniform random variables? Explain the roles of generating and sorting.

Pause here. Try each question before turning to the answers.

Answers and reasoning

Arrival locations given a count

---

1. Binomial$\left(6, \frac{1}{2}\right)$, with mean 3.
2. $\left(\frac{1}{2}\right)^6 = \frac{1}{64}$. Each generated location must fall in the second half.
3. No. Generate six independent uniform locations, then sort them. Sorting gives ordered, dependent arrival times.

A fixed total leads to uniform locations, binomial subwindow counts, and dependence between complementary counts.

One arrival model now answers three kinds of questions

Question	Useful representation
How many by a fixed time?	Poisson increments
How long until the next or nth request?	Exponential gaps; a count event
Where were the arrivals, given their total?	Uniform locations; binomial subwindow counts

The next question is operational: **what happens when requests come from different regions or receive different routing labels?**

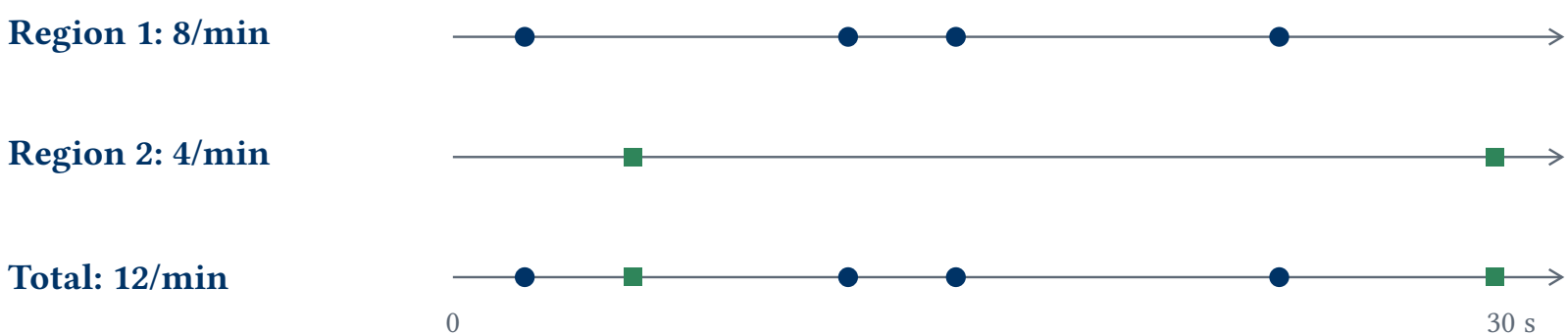
? Prepare the next step

Two independent regions send 8 and 4 requests per minute. What arrival model should the combined gateway use?

3.6. Combining and Classifying Arrivals

3.6.1. Independent Sources

Region 1 sends requests at rate 8 per minute; region 2 sends requests at rate 4 per minute. Assume their request streams are **independent Poisson processes**.



Let $N_1(t)$ and $N_2(t)$ be their cumulative counts. The gateway records

$$N(t) = N_1(t) + N_2(t).$$

We are merging arrival records, not modeling how the requests are processed.

Independent Poisson rates add

Superposition **Theorem 3.2**

The sum of independent Poisson processes with rates λ_1 and λ_2 is a Poisson process with rate $\lambda_1 + \lambda_2$.

Let $N(t) = N_1(t) + N_2(t)$. Check the three defining conditions:

1. **Starts at zero:** $N(0) = N_1(0) + N_2(0) = 0$.
2. **Independent increments:** disjoint windows use independent increments within each source, and the sources are independent. The sums for disjoint windows are therefore independent.
3. **Poisson increments:** for every $t \geq 0$ and $s > 0$, the source increments are independent $\text{Poisson}(\lambda_1 s)$ and $\text{Poisson}(\lambda_2 s)$ counts. Adding them gives

$$N(t+s) - N(t) \sim \text{Poisson}((\lambda_1 + \lambda_2)s).$$

Why the independence assumption matters

Suppose two logs record the **same** rate-r Poisson stream, where $r > 0$:

$$N_2(t) = N_1(t), \quad M(t) = N_1(t) + N_2(t) = 2N_1(t).$$

Every arrival is counted twice. The combined count is **always even**, so

$$\mathbb{P}(M(t) = 1) = 0.$$

If M were a rate-$2r$ Poisson process, it would instead satisfy

$$\mathbb{P}(M(t) = 1) = 2rt \exp(-2rt) > 0, \quad t > 0.$$

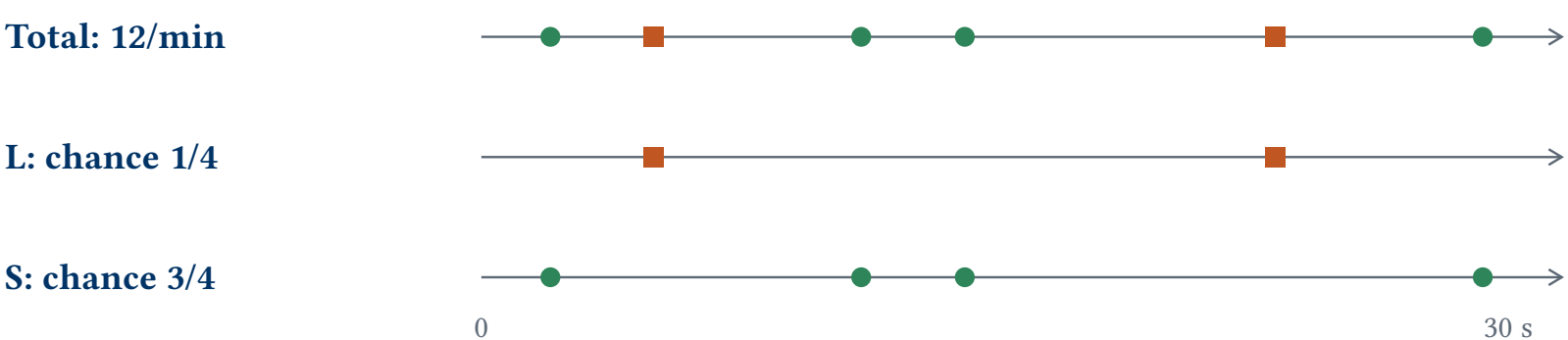


The missing assumption

Each log is Poisson, but the logs are **dependent**. **Without independence, adding Poisson streams need not give a Poisson process.**

Classify requests without moving their arrival times

The gateway receives a rate-λ Poisson stream. Independently for every request, a router chooses the **large model** with probability p and the **small model** with probability $1 - p$.



Let $N_L(t)$ and $N_S(t)$ count these two labels at arrival. The labels are independent across requests and independent of the arrival record.

Independent labels produce independent Poisson streams

Independent thinning **Theorem 3.3**

Under independent fixed-probability labeling, N_L and N_S are independent Poisson processes with rates

$$\lambda_L = \lambda p, \quad \lambda_S = \lambda(1-p).$$

At $\lambda = 12$ per minute and $p = \frac{1}{4}$, the rates are **3 and 9 per minute**. For a 30-second window,

$$\mathbb{P}(N_L(0.5) = 0) = \exp(-1.5) \approx 0.2231.$$

The expected wait for the next large-model request is

$$\frac{1}{3} \text{ minute} = 20 \text{ seconds}.$$

The total rate is preserved; the independent labels determine how it is divided.

Given the total, count the routing coin flips

Now the log tells us that exactly n requests arrived in a window of length t. Let $K = N(t)$, $K_L = N_L(t)$, and $K_S = N_S(t)$.

Each of the n independent labels chooses the large model with probability p, so

$$K_L \mid \{K = n\} \sim \text{Binomial}(n, p), \quad K_S = n - K_L.$$

For $n = 6$ and $p = \frac{1}{4}$,

$$\mathbb{P}(K_L = 0 \mid K = 6) = \left(\frac{3}{4}\right)^6 \approx 0.1780.$$

This differs from the unconditional value $\exp(-1.5) \approx 0.2231$ for a 30-second window.

? Question

Why does knowing K_L determine K_S after the total is fixed, but not before?

The two counts are independent before conditioning

The total already has the law $K \sim \text{Poisson}(m)$, where $m = \lambda t$. We now calculate the joint probability **without fixing K**.

For $K_L = k$ and $K_S = \ell$, the total must be $k + \ell$, with exactly k labels L:

$$\begin{aligned}\mathbb{P}(K_L = k, K_S = \ell) &= \mathbb{P}(K = k + \ell)\mathbb{P}(K_L = k \mid K = k + \ell) \\ &= \exp(-m)\frac{m^{k+\ell}}{(k+\ell)!}\binom{k+\ell}{k}p^k(1-p)^\ell.\end{aligned}$$

Canceling the factorials makes the joint probability a product:

$$\mathbb{P}(K_L = k, K_S = \ell) = \exp(-mp)\frac{(mp)^k}{k!}\exp(-m(1-p))\frac{(m(1-p))^\ell}{\ell!}.$$

These factors are the Poisson probabilities with means mp and $m(1-p)$, so **the two counts are independent**. Given $K = n$, they instead satisfy the fixed constraint $K_S = n - K_L$.

Verification of independent thinning

For any $t \geq 0$ and $s > 0$, write $\Delta_i = N_i(t+s) - N_i(t)$ for $i \in \{L, S\}$.

1. **Starts at zero:** $N_L(0) = N_S(0) = 0$.
2. **Independent increments:** disjoint windows use independent original counts and independent label choices. Thus their pairs (Δ_L, Δ_S) are independent.
3. **Poisson increments:** apply the preceding factorization to a window of length s:

$$\Delta_L \sim \text{Poisson}(\lambda p s), \quad \Delta_S \sim \text{Poisson}(\lambda(1-p)s).$$

Why the two streams are independent: Within each window, the factorization also makes Δ_L and Δ_S independent. Across any finite time partition, all L and S increments are therefore mutually independent. Cumulative counts are sums of these increments, so the two processes are independent.

Which routing rules satisfy the theorem?

Inspect the rule, not only the long-run fraction **Poll 3.7**

Which rules are covered by independent thinning?

A). Each request uses a fresh independent coin with $p = \frac{1}{4}$.

B). Requests alternate between the two models.

C). A request goes to the large model whenever the small-model queue is long.

For each rule, identify whether its label depends on previous labels, arrival times, or the current system state.

Receiving a certain fraction of requests is not enough to establish a Poisson arrival stream.

Alternation and congestion change the mechanism

Only **A** is covered by the theorem.

Rule	Why it differs
Alternation	Every later gap between two L-labeled requests sums two independent exponential gaps, so it is not exponential.
Congestion-based routing	The label depends on the current queue state instead of an independent fixed-probability coin. The ordinary thinning theorem gives no such conclusion.

For alternating labels, each stream has long-run rate $\frac{\lambda}{2}$, but this alone does not make it Poisson.

State-dependent routing connects the arrival model to queues and, later, control policies.

Pause and recall

Combine, split, or condition

Independent regions send 8 and 4 requests per minute; independently, each request chooses the large model with probability $\frac{1}{4}$.

1. What is the large-model count distribution in 20 seconds?
2. Given exactly eight total requests in that window, what is its conditional distribution?
3. Are the L and S counts independent before the total is known? After it is known?

Pause here. Try each question before turning to the answers.

Answers and reasoning

Combine, split, or condition

---

1. The large-model rate is 3 per minute, so the count is Poisson(1).
2. Binomial$\left(8, \frac{1}{4}\right)$.
3. They are independent before conditioning. Given total eight, they add to eight and are dependent.

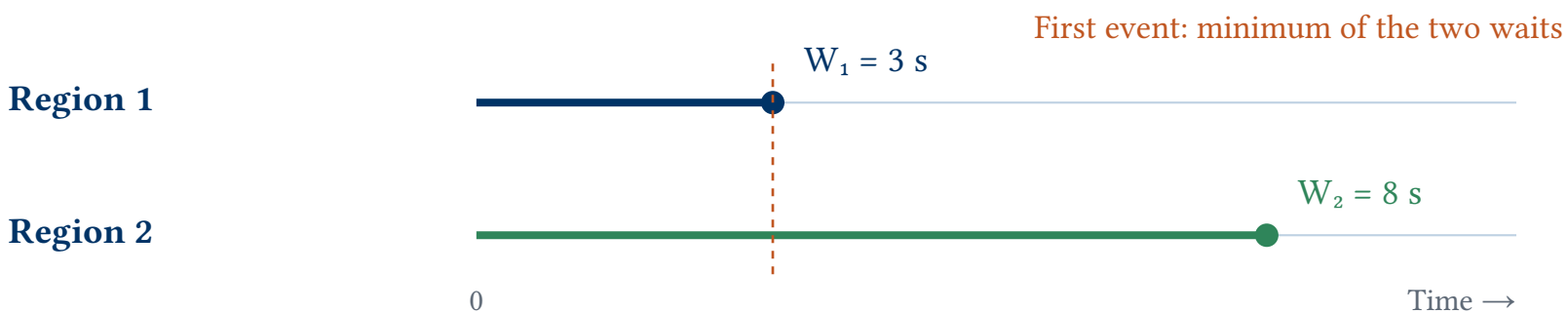
Specify the source independence, the labeling mechanism, and the information supplied.

3.7. Which Event Happens Next?

3.7.1. Competing Exponential Clocks

Let W_1 and W_2 be the waits from now to the next request from regions 1 and 2. Their independent streams give

$$W_1 \sim \text{Exp}(\lambda_1), \quad W_2 \sim \text{Exp}(\lambda_2).$$



The next gateway request arrives after $W = \min(W_1, W_2)$. Let J identify the region that sends it.

We need **two forecasts**: how long we wait and which event occurs.

No combined arrival means neither clock has finished

For the first request to arrive after time t, both clocks must exceed t:

$$\begin{aligned}\mathbb{P}(W > t) &= \mathbb{P}(W_1 > t, W_2 > t) \\ &= \exp(-\lambda_1 t)\exp(-\lambda_2 t) \\ &= \exp(-(\lambda_1 + \lambda_2)t).\end{aligned}$$

Thus $W \sim \mathrm{Exp}(\lambda_1 + \lambda_2)$.

At rates 8 and 4 per minute, the expected wait is

$$\mathbb{E}[W] = \frac{1}{8+4} \text{ minute} = 5 \text{ seconds}.$$

This is the waiting-time version of superposition.

Conditioning on clock 1's waiting time

Region 1 wins when $W_1 < W_2$. Condition on $W_1 = s$, then average over the possible values of s:

$$\begin{aligned}\mathbb{P}(W_1 < W_2) &= \int_0^\infty f_{W_1}(s)\mathbb{P}(W_1 < W_2 \mid W_1 = s)\,\mathrm{d}s \\ &= \int_0^\infty f_{W_1}(s)\mathbb{P}(W_2 > s \mid W_1 = s)\,\mathrm{d}s \\ &= \int_0^\infty f_{W_1}(s)\mathbb{P}(W_2 > s)\,\mathrm{d}s.\end{aligned}$$

1. **Total probability:** average the conditional winning probability using the density of W_1.
2. **Given $W_1 = s$:** the event $W_1 < W_2$ becomes $W_2 > s$.
3. **Independence:** knowing $W_1 = s$ does not change the probability that $W_2 > s$.

The faster source is more likely to win

Now substitute the exponential density and survival probability into this integral:

$$\begin{aligned}\mathbb{P}(J=1) = \mathbb{P}(W_1 < W_2) &= \int_0^\infty f_{W_1}(s)\mathbb{P}(W_2 > s)\,\mathrm{d}s \\ &= \int_0^\infty \lambda_1 \exp(-\lambda_1 s)\exp(-\lambda_2 s)\,\mathrm{d}s = \lambda_1 \int_0^\infty \exp(-(\lambda_1+\lambda_2)s)\,\mathrm{d}s \\ &= -\frac{\lambda_1}{\lambda_1+\lambda_2}\left[\exp(-(\lambda_1+\lambda_2)s)\right]_0^\infty \\ &= \frac{\lambda_1}{\lambda_1+\lambda_2}.\end{aligned}$$

Similarly, $\mathbb{P}(J=2) = \frac{\lambda_2}{\lambda_1+\lambda_2}$. Ties have probability zero.

Which region sends the next request? — Example 3.1

Region 1 wins with probability $\frac{8}{12} = \frac{2}{3}$ and region 2 with probability $\frac{1}{3}$.

Timing and event type can be sampled separately

Let $\Lambda = \lambda_1 + \lambda_2$. The same calculation, starting at time t, gives

$$\mathbb{P}(W > t, J = i) = \int_t^\infty \lambda_i \exp(-\Lambda s)\, \mathrm{d}s = \exp(-\Lambda t)\frac{\lambda_i}{\Lambda}.$$

This factors as $\mathbb{P}(W > t)\mathbb{P}(J = i)$. The winning time W and winning region J are independent.

An equivalent next-event simulation

Draw one $\mathrm{Exp}(\Lambda)$ waiting time. Independently choose region i with probability $\frac{\lambda_i}{\Lambda}$.

For any finite number of independent exponential clocks, add their rates and choose the winner in proportion to its rate.

Sequential events require a sum; competing events require a minimum

The gateway's total rate is 12 per minute. Two questions that mention "two" lead to different durations.

Target	Duration	Mean
Wait for two gateway requests in sequence.	$T_2 = A_1 + A_2$	10 seconds
Wait for the first request from either of two regions.	$W = \min(W_1, W_2)$	5 seconds

? Question

Are you waiting for several required events to accumulate, or for the first of several possibilities?

Write the event structure before deciding which rates or durations to add.

Pause and recall

Timing and type of the next event

Three independent Poisson sources have rates 2, 3, and 5 per minute.

1. What is the waiting-time distribution for the next combined event? Give its mean in seconds.
2. What is the probability that source 2 produces that event?
3. After no combined event for six seconds, does source 2 become more likely to win? Explain.

Pause here. Try each question before turning to the answers.

Answers and reasoning

Timing and type of the next event

---

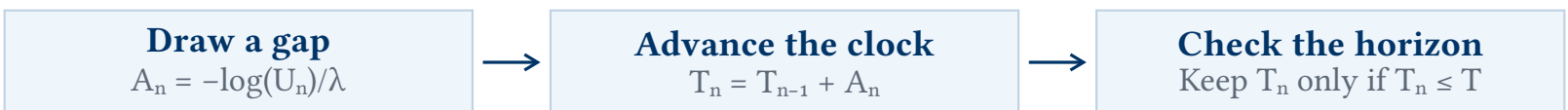
1. $\mathrm{Exp}(10)$ with time in minutes. Its mean is $\frac{1}{10}$ minute, or six seconds.
2. $\frac{3}{2+3+5} = 0.3$.
3. No. The winning time and source are independent. Equivalently, the surviving exponential clocks have unchanged remaining laws.

A next event has a waiting time and a type. Independent exponential clocks give both.

3.8. Concluding Remarks

3.8.1. Simulating a Poisson Process

To simulate a rate-λ Poisson process up to a fixed horizon T:



Repeat with a fresh independent draw; stop at the first overshoot.

Draw independent $U_n \sim \text{Uniform}(0,1)$ and set

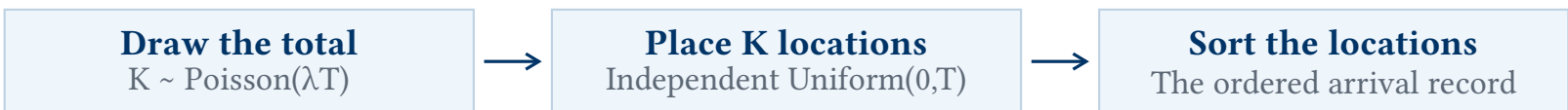
$$A_n = -\frac{\ln(U_n)}{\lambda}, \quad T_n = T_{n-1} + A_n.$$

Starting from $T_0 = 0$, retain the arrival times $T_n \le T$. Stop when the next cumulative time exceeds T; that final point is **not** an arrival in the observation window.

The exponential transform follows from $\mathbb{P}\left(-\frac{\ln(U_n)}{\lambda} > t\right) = \mathbb{P}(U_n < \exp(-\lambda t)) = \exp(-\lambda t)$.

Or draw a total count, then place its arrivals

For the same fixed observation horizon T, another exact construction is:



When K = 0, the arrival record is empty.

Draw $K \sim \text{Poisson}(\lambda T)$. Given $K = k$, draw k independent uniform locations in $(0, T)$ and sort them. For $K = 0$, the record is empty.

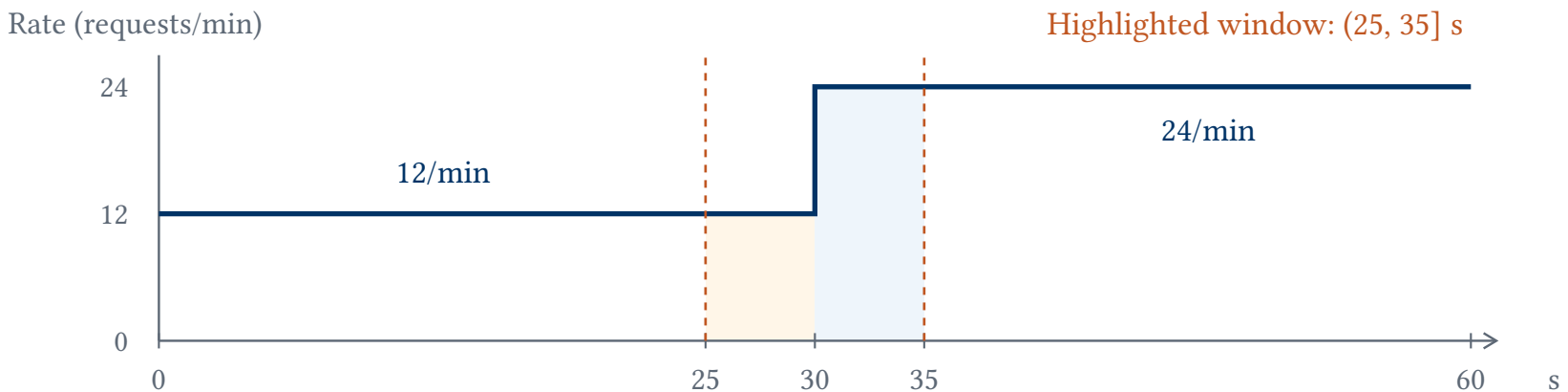
Same model, different construction

Exponential gaps reveal the record in chronological order. Poisson counts and uniform locations generate the whole fixed window at once.

Always drawing exactly λT locations would fix the total, not simulate its Poisson variation.

3.8.2. A Time-Varying Rate

A **nonhomogeneous Poisson process** starts at $N(0) = 0$, has independent increments, and uses a deterministic rate $\lambda(t) \geq 0$.

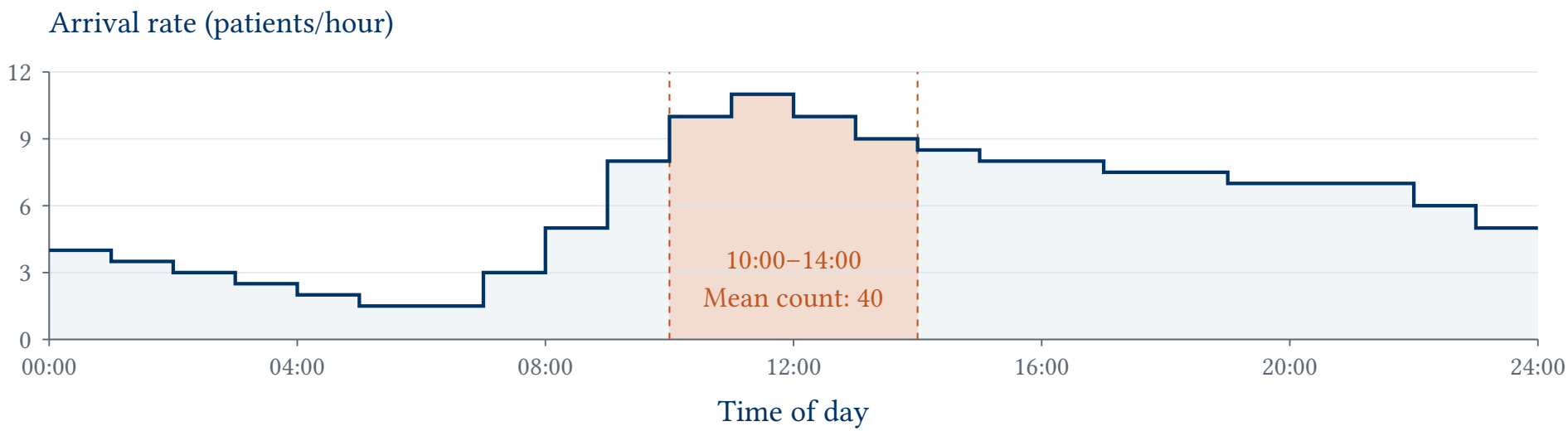


$$N(b) - N(a) \sim \text{Poisson}\left(\int_a^b \lambda(u)\,\mathrm{d}u\right).$$

The mean count is the **area under the rate curve**, assumed finite on bounded windows. Equal-length windows can therefore have different count distributions.

Emergency-department arrivals over a day

Illustrative hourly ED arrival rates for a weekday; t is hours after midnight.

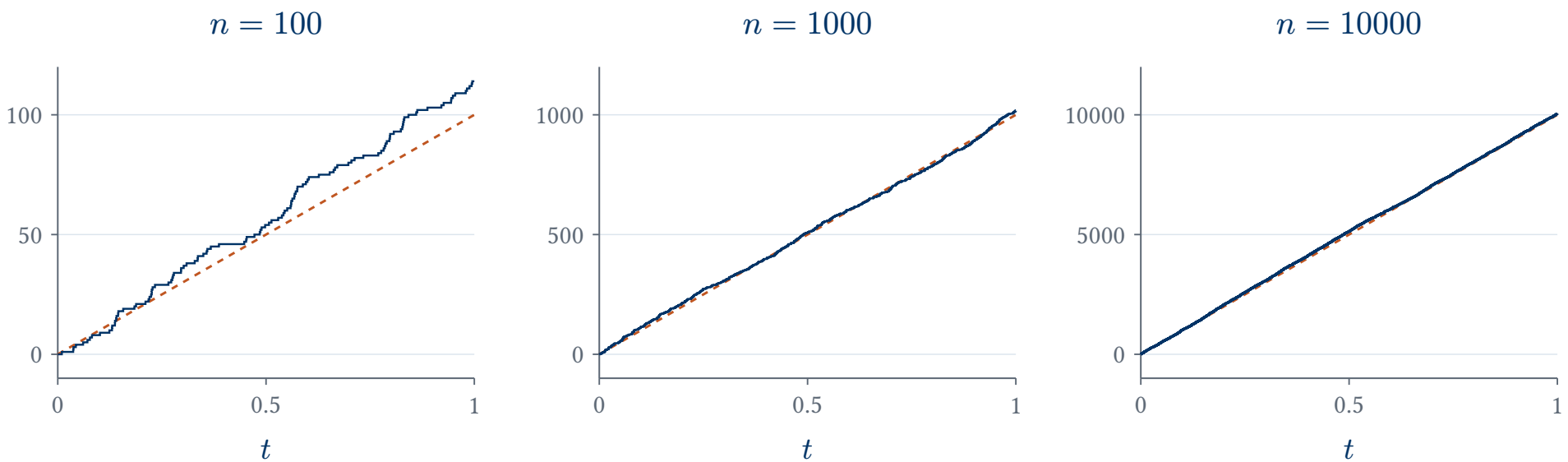


Under a time-varying Poisson model, the shaded 10:00–14:00 window gives

$$m = \int_{10}^{14} \lambda(t)\,\mathrm{d}t = 10 + 11 + 10 + 9 = 40, \quad N(14) - N(10) \sim \text{Poisson}(40).$$

3.8.3. Poisson Fluctuations and Brownian Motion

Start with a rate-1 process. To compare observation horizons n, plot $N(nt)$ against $0 \le t \le 1$.

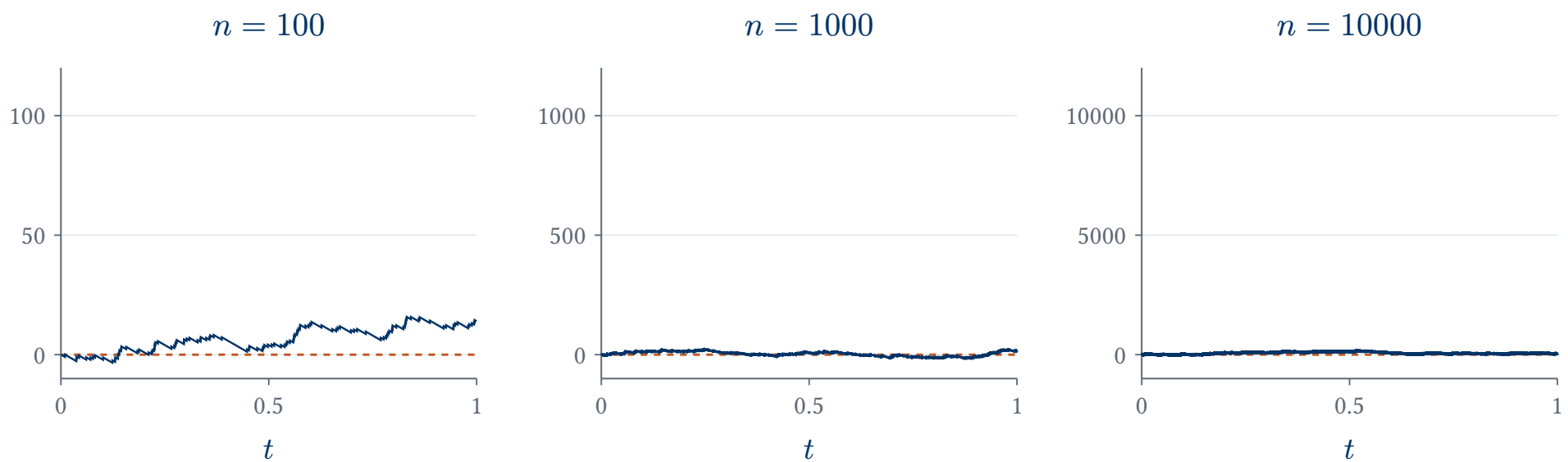


The **first-order trend** is nt (dashed). By the law of large numbers, $\frac{N(nt)}{n} \to t$ as $n \to \infty$.

Plotting approach adapted from W. Whitt, Experiencing Statistical Regularity (slides 7–16) and Stochastic-Process Limits, Ch. 1.

Subtract the trend; keep the vertical axes unchanged

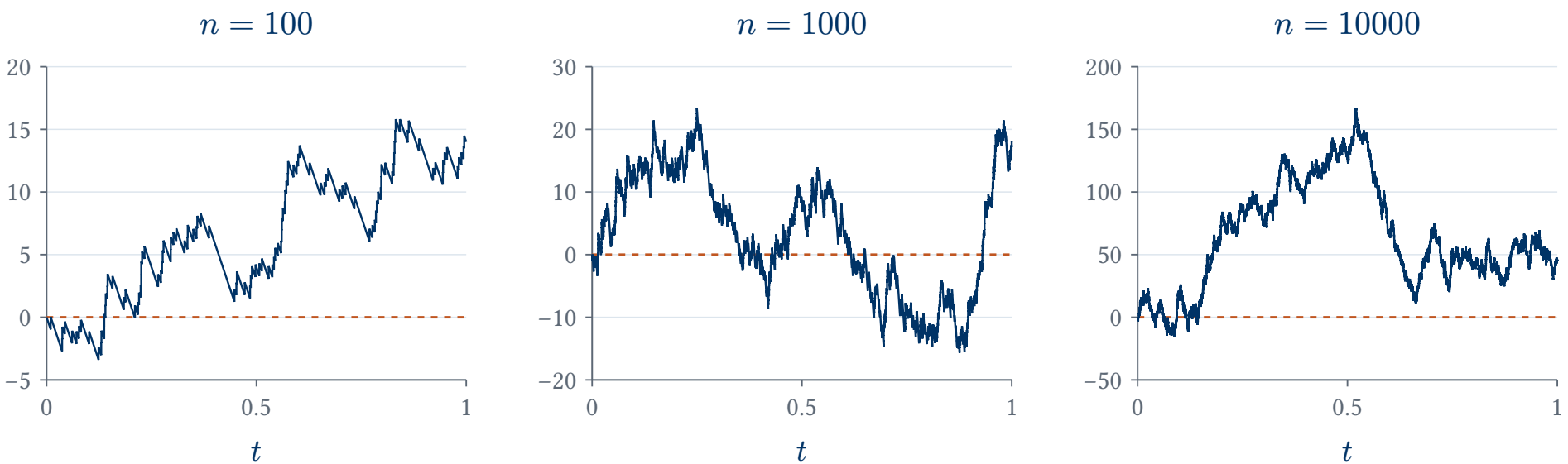
Now plot the **centered count** $D_n(t) = N(nt) - nt$. Each panel keeps exactly the same vertical range as on the previous slide.



The fluctuations look small on the scale of the total count. Centering removes the first-order trend, but this scale hides the remaining randomness.

Let the plotter enlarge the remaining fluctuations

These are the **same centered paths** $D_n(t)$. Only the vertical ranges change: each is selected automatically from that panel's minimum and maximum.



The fluctuations are now visible. Notice the different numbers on the vertical axes. **What common scaling rule would make their sizes comparable?**

The variance tells us to use a square-root scale

For a rate-1 process, $N(nt) \sim \text{Poisson}(nt)$. Subtracting a deterministic trend changes the mean but not the variance:

$$\mathbb{E}[D_n(t)] = 0, \quad \text{Var}(D_n(t)) = nt,$$
$$\text{SD}(D_n(t)) = \sqrt{n}\sqrt{t}.$$

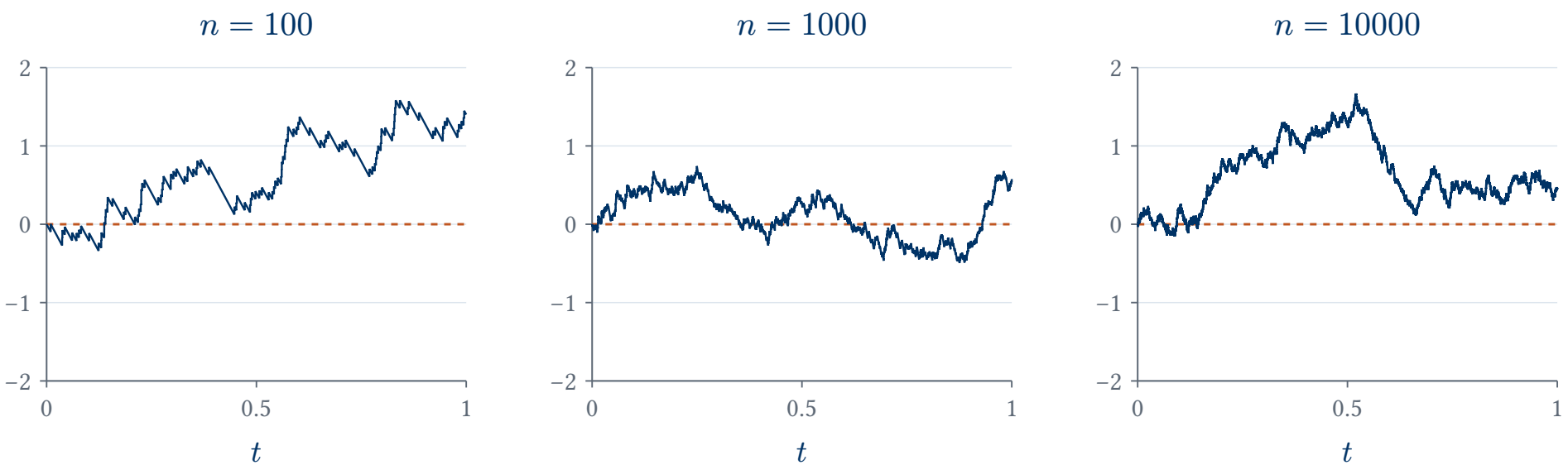
At the end of the observation window ($t = 1$):

Horizon n	**100**	**1,000**	**10,000**
Mean count n	100	1,000	10,000
Standard deviation $\sqrt{n}$	10	31.6	100

Thus the standard-deviation scale grows like $\sqrt{n}$, while the trend grows like n. Their size **relative to the trend** shrinks like $\frac{1}{\sqrt{n}}$.

Divide by the square root; compare on one common scale

$$Z_n(t) = \frac{N(nt) - nt}{\sqrt{n}}, \quad \mathbb{E}[Z_n(t)] = 0, \quad \mathrm{Var}(Z_n(t)) = t.$$



All three panels now use the same vertical range. The fluctuations have comparable size, while each upward jump has height $\frac{1}{\sqrt{n}}$.

The limiting fluctuations are Brownian motion

For $0 \le s < t$, the centered increment is a centered Poisson count. As $n \to \infty$, the ordinary central limit theorem gives

$$Z_n(t) - Z_n(s) = \frac{N(nt) - N(ns) - n(t-s)}{\sqrt{n}} \Rightarrow \text{Normal}(0, t-s).$$

Increments over disjoint intervals are independent for every n. This suggests a process with independent Gaussian increments.

> **Standard Brownian motion** **Definition 3.4**
>
> A process $B(t)$ with $B(0) = 0$, continuous sample paths, and independent increments such that $B(t) - B(s) \sim \text{Normal}(0, t-s)$ for $0 \le s < t$.

Its mean is zero and its variance at time t is t. Continuity is part of the process limit theorem, beyond the one-increment CLT.

A central limit theorem for the whole arrival path

Poisson functional central limit theorem (FCLT) **Theorem 3.4**

For a rate-1 Poisson process, as $n \to \infty$,

$$\left\{ \frac{N(nt) - nt}{\sqrt{n}} : 0 \le t \le 1 \right\} \Rightarrow \{B(t) : 0 \le t \le 1\}.$$

The convergence is in distribution of **random paths**.

The FCLT extends the ordinary CLT from a single count to the whole evolving fluctuation.

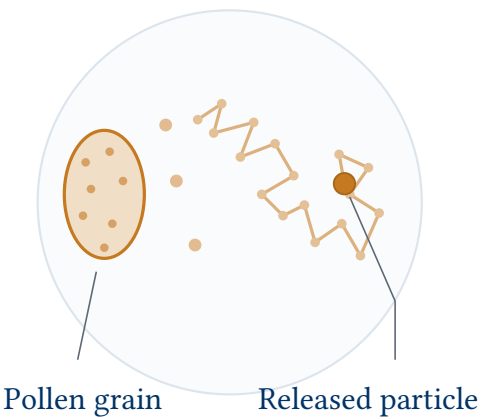
For a general rate $\lambda > 0$ and large n,

$$N_\lambda(nt) \approx \lambda nt + \sqrt{\lambda n} B(t).$$

Rate describes the trend; Brownian motion describes the square-root fluctuations.

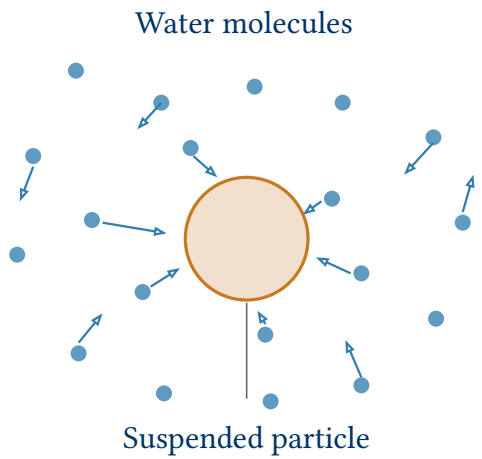
Historical remarks on Brownian motion

1827 — Robert Brown



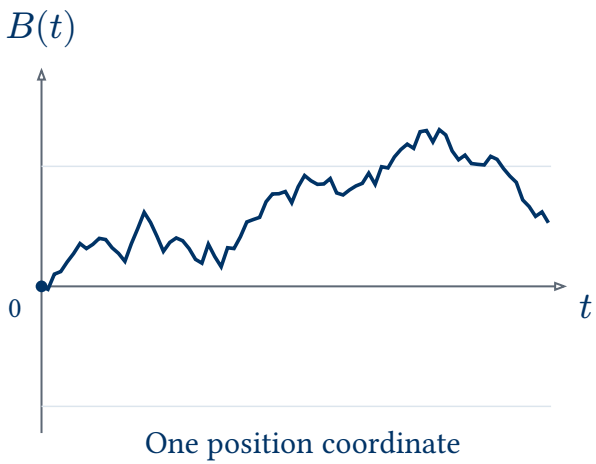
Observed tiny particles released from pollen moving irregularly in water.

1905 — Albert Einstein



Connected diffusion to thermal molecular motion; typical displacement grows like $\sqrt{t}$.

1923 — Norbert Wiener



Developed a rigorous probability model of continuous random paths: the **Wiener process**.

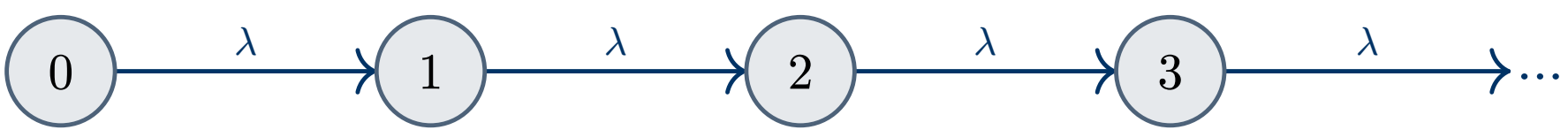
4. Continuous-Time Markov Chains

4.1. Exponential Clocks and Markov States

An **exponential clock** with rate $\lambda > 0$ per unit time rings after a random time H:

$$H \sim \mathrm{Exp}(\lambda), \quad \mathbb{P}(H > t) = e^{-\lambda t}, \quad \mathbb{E}[H] = \frac{1}{\lambda}.$$

In a Poisson process, let $N(t)$ count arrivals by time t. Each ring adds one to the count and starts a fresh, independent clock.



At count n, the **holding time** is the wait until the next arrival. Its rate is always λ, and the next state is always $n + 1$.

Why the Poisson count is Markov

Suppose $N(s) = n$ and the last arrival was a time units ago. Memorylessness gives the remaining wait:

$$\mathbb{P}(H > a + t \mid H > a) = e^{-\lambda t}.$$

The age a does not change this forecast. After the next arrival, subsequent clocks are again independent $\mathrm{Exp}(\lambda)$ variables.



The present count is enough

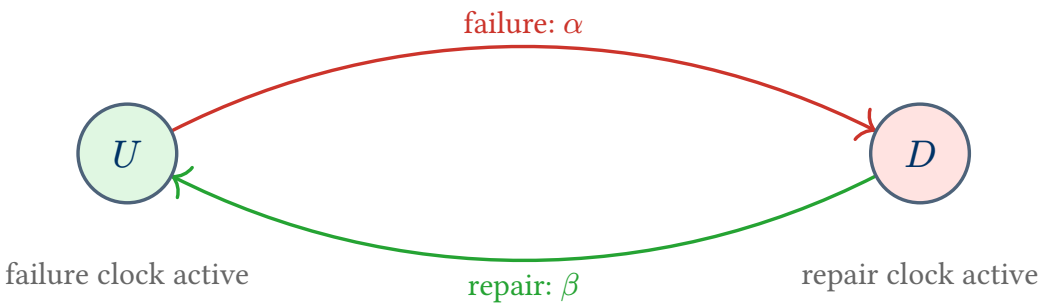
Given $N(s) = n$, earlier arrival times provide no additional information about future counts. This is the **Markov property**.

Equivalently, independent, stationary increments give, for $k = 0, 1, \cdots$,

$$\mathbb{P}(N(s+t) = n + k \mid N(s) = n, \text{past before } s) = e^{-\lambda t} \frac{(\lambda t)^k}{k!}.$$

One machine has one active clock in each state

A machine alternates between **Up (U)** and **Down (D)**. Its failure rate is $\alpha > 0$ and its repair rate is $\beta > 0$. Write H_U and H_D for the holding times in these states, with independent exponential durations on successive visits.



Current state	Holding time	Mean	Next state
Up	$H_U \sim \text{Exp}(\alpha)$	$\frac{1}{\alpha}$	Down
Down	$H_D \sim \text{Exp}(\beta)$	$\frac{1}{\beta}$	Up

In Chapter 2, α, β were probabilities per step. Here they are **rates per unit time**. Over a short duration h, the transition probabilities are approximately αh from Up to Down and βh from Down to Up.

Why Up or Down is enough for prediction

If the machine is Down, a repair that has lasted a time units has remaining law $\mathrm{Exp}(\beta)$:

$$\mathbb{P}(H_D > a + t \mid H_D > a) = e^{-\beta t}.$$

After repair, an independent failure clock starts. Memorylessness also applies while Up.



Keep the information needed for prediction

Repair and operating ages add no predictive information, so the Up/Down process is Markov.

? What if every repair lasts exactly two hours?

After 5 or 90 minutes of repair, the remaining wait is 115 or 30 minutes. “Down” alone is insufficient. Record the elapsed repair age, a continuous state component.

The continuous-time Markov chain definition

Write $X(t)$ for a process with a **finite or countable state space** $\mathcal{S}$ and continuous time $t \geq 0$.

Time-homogeneous CTMC — Definition 4.1

For states $i, j \in \mathcal{S}$ and times $s, t \geq 0$, the present state makes the future law independent of the earlier history:

$$\mathbb{P}(X(s+t) = j \mid X(s) = i, \text{past before } s) = \mathbb{P}(X(s+t) = j \mid X(s) = i).$$

Time-homogeneity means that this law depends only on elapsed time:

$$\mathbb{P}(X(s+t) = j \mid X(s) = i) = \mathbb{P}(X(t) = j \mid X(0) = i).$$

Our chains have piecewise-constant, right-continuous paths. We assume **non-explosion**: almost surely, finitely many jumps occur in every bounded time interval.

The Poisson count and the exponential Up/Down machine satisfy this definition.

Why a CTMC holding time is memoryless

Start with $X(0) = i$, and let H_i be the time of the **first departure** from i. If $H_i > s$, no departure has occurred, so $X(s) = i$.

1. **Markov property:** given $X(s) = i$, the past does not change the future law. Knowing that we have stayed in i throughout $[0, s]$ adds no information about when we will leave after s.
2. **Time-homogeneity:** starting from i at time s gives the same future law as starting from i at time 0. The chance of staying for another t time units is therefore $\mathbb{P}(H_i > t)$.

On $H_i > s$, the event $H_i > s + t$ means **no departure during the next t time units**. Hence

$$\mathbb{P}(H_i > s + t \mid H_i > s) = \mathbb{P}(H_i > t).$$

Here $H_i > s$ means no departure. The event $X(s) = i$ alone also includes paths that left and returned.

Memorylessness gives exponential holding times

Write $G(t) = \mathbb{P}(H_i > t)$. Multiplying the memoryless identity by $\mathbb{P}(H_i > s) = G(s)$ gives

$$G(s+t) = G(s)G(t).$$

Memoryless survival functions — Lemma 4.1

If $G(t) = \mathbb{P}(H_i > t)$ is right-continuous, $G(0) = 1$, and $G(s+t) = G(s)G(t)$, then

$$G(t) = e^{-q_i t} \quad \text{for some finite } q_i \geq 0.$$

Proof: see the appendix on Page 487.

Here q_i is the **total exit rate** from state i, also the rate of its exponential holding time. For $q_i > 0$, $H_i \sim \text{Exp}(q_i)$ with mean $\frac{1}{q_i}$. If $q_i = 0$, the state is **absorbing** and the process stays there forever.

For one machine, the clock also fixes the destination. With two machines, several clocks may compete.

4.2. Transition Rates and the Generator

Two machines: which state information matters?

Two identical machines share **one repair technician**. Each Up machine has a failure clock $\mathrm{Exp}(\alpha)$. The technician works on one repair with duration $\mathrm{Exp}(\beta)$, then starts any waiting repair. Operating times and active repair times are independent exponentials.

Let $X(t)$ be the **number of working machines**:

- $X(t) = 0$: both Down, configuration DD.
- $X(t) = 1$: one Up, configuration UD or DU.
- $X(t) = 2$: both Up, configuration UU.

In either UD or DU, one failure clock and one repair clock are active. Identical machines and memoryless clocks give the same future count law, so the three-state count process is Markov.

We use the active clocks in states $0, 1, 2$ to derive **transition rates** from short-time probabilities. These rates govern **probability flow** between states and form the **generator matrix** Q.

State 0: both machines Down

Only **one repair clock** is active, because there is one technician. Its completion leaves one machine working, so the next state is certainly 1.

$$H_0 \sim \mathrm{Exp}(\beta), \quad \mathbb{E}[H_0] = \frac{1}{\beta}.$$

For a short interval $h > 0$, recall that $o(h)$ denotes a remainder whose ratio to h tends to zero. Starting in state 0,

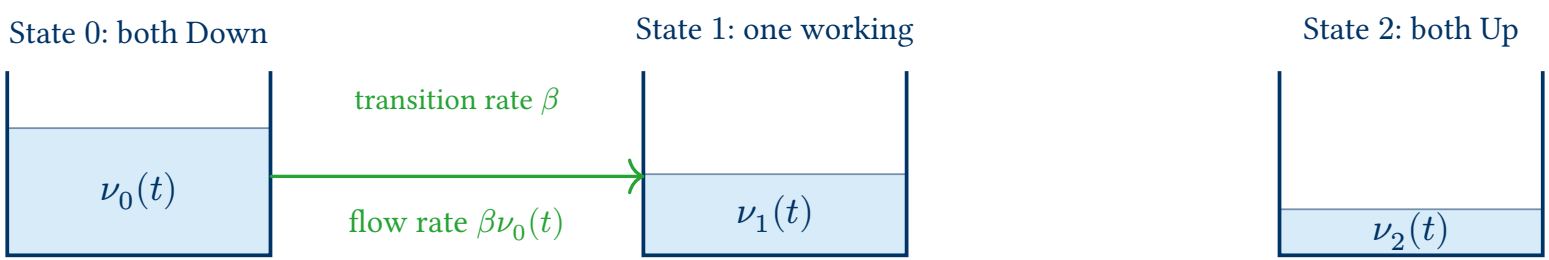
$$\mathbb{P}(H_0 \le h, \text{next state} = 1 \mid X(0) = 0) = 1 - e^{-\beta h} = \beta h + o(h).$$

Thus, for small h, the probability of a first move from state 0 to state 1 within time h is approximately βh: **exit rate $\times$ interval length**, since the next state is certainly 1.

Many copies of the system: probability as fluid

Run many independent copies of the **same two-machine system**, with one technician per copy and any chosen initial distribution. Represent the copies by equal drops with total volume 1.

Tank i holds **probability mass** $\nu_i(t) = \mathbb{P}(X(t) = i)$, approximated by the fraction of copies currently in state i.



Only flows out of state 0 are shown.

By memorylessness, about a fraction βh of tank 0 moves to tank 1 in the next short interval h:

$$\text{probability transferred} \approx \nu_0(t)\beta h, \quad \text{probability flow rate} = \beta\nu_0(t).$$

The transition rate $q_{0,1} = \beta$ is the **flow rate per unit probability mass**: doubling the mass doubles the flow rate.

Competing exponential clocks

Competing exponentials — Lemma 4.2

Let $E_k \sim \mathrm{Exp}(\lambda_k)$, $k = 1, \cdots, m$, be **independent**, with $\lambda_k > 0$. Write

$$H = \min_{1 \le k \le m} E_k, \quad \Lambda = \sum_{k=1}^{m} \lambda_k,$$

and let K be the index of the first clock to ring. Ties have probability zero. Then

$$H \sim \mathrm{Exp}(\Lambda), \quad \mathbb{P}(K = k) = \frac{\lambda_k}{\Lambda}.$$

The **waiting time H** and the **winning clock K** are independent.

The next state is independent of H when determined by K, even if several clocks share a destination.

Proof of Lemma 4.2:

For $t \geq 0$, all clocks are still waiting exactly when $E_k > t$ for every k:

$$\mathbb{P}(H > t) = \prod_{k=1}^{m} \mathbb{P}(E_k > t) = e^{-\Lambda t}.$$

For clock k to win at time u, it must ring while every other clock is still waiting:

$$\begin{aligned}\mathbb{P}(H \leq t, K = k) &= \int_0^t \lambda_k e^{-\lambda_k u} \prod_{\ell \neq k} e^{-\lambda_\ell u} \, \mathrm{d}u \\ &= \int_0^t \lambda_k e^{-\Lambda u} \, \mathrm{d}u = \frac{\lambda_k}{\Lambda}\left(1 - e^{-\Lambda t}\right).\end{aligned}$$

Letting $t \to \infty$ gives $\mathbb{P}(K = k) = \frac{\lambda_k}{\Lambda}$. For every $t \geq 0$ and k, the joint probability therefore factors:

$$\mathbb{P}(H \leq t, K = k) = \mathbb{P}(H \leq t)\mathbb{P}(K = k).$$

Thus **the waiting time and winning clock are independent**. ∎

State 1: two clocks and two destinations

A failure clock $F \sim \mathrm{Exp}(\alpha)$ races an independent repair clock $C \sim \mathrm{Exp}(\beta)$. Failure leaves both machines Down (state 0); repair leaves both Up (state 2).

Apply Lemma 4.2 with rates α and β. The holding time is $H_1 = \min(F, C)$, and J denotes the next state:

$$H_1 \sim \mathrm{Exp}(\alpha + \beta).$$

$$\mathbb{P}(J = 0) = \frac{\alpha}{\alpha + \beta}, \quad \mathbb{P}(J = 2) = \frac{\beta}{\alpha + \beta}.$$

Since the winning clock determines J, **the destination and holding time are independent**.

State 1: short-time transition probabilities

The probability of leaving state 1 within a short interval h is

$$\mathbb{P}(H_1 \leq h) = 1 - e^{-(\alpha+\beta)h} = (\alpha+\beta)h + o(h).$$

By **independence**, multiply by the probability of each destination:

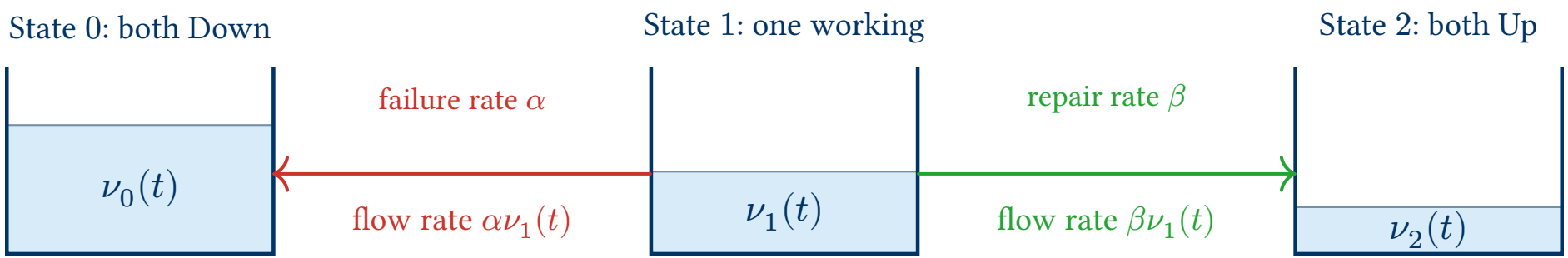
$$\mathbb{P}(H_1 \leq h, J = 0) = \mathbb{P}(H_1 \leq h)\mathbb{P}(J = 0) = [(\alpha+\beta)h + o(h)]\frac{\alpha}{\alpha+\beta} = \alpha h + o(h).$$

$$\mathbb{P}(H_1 \leq h, J = 2) = \mathbb{P}(H_1 \leq h)\mathbb{P}(J = 2) = [(\alpha+\beta)h + o(h)]\frac{\beta}{\alpha+\beta} = \beta h + o(h).$$

Thus, for small h, the probabilities of a first move from state 1 to state 0 or state 2 within time h are approximately αh and βh, respectively. Each approximation uses **exit rate $\times$ interval length**, multiplied by the **destination probability**.

State 1: two flows into different destinations

The two transition rates are $q_{1,0} = \alpha$ and $q_{1,2} = \beta$.



Only flows out of state 1 are shown.

$$\text{total outflow rate} = \alpha\nu_1(t) + \beta\nu_1(t) = (\alpha + \beta)\nu_1(t).$$

State 2: both machines Up

Independent failure clocks $F_1, F_2 \sim \mathrm{Exp}(\alpha)$ compete. Either failure sends the count to state 1.

Apply Lemma 4.2 with both rates equal to α:

$$H_2 = \min(F_1, F_2) \sim \mathrm{Exp}(2\alpha), \quad \mathbb{E}[H_2] = \frac{1}{2\alpha}.$$

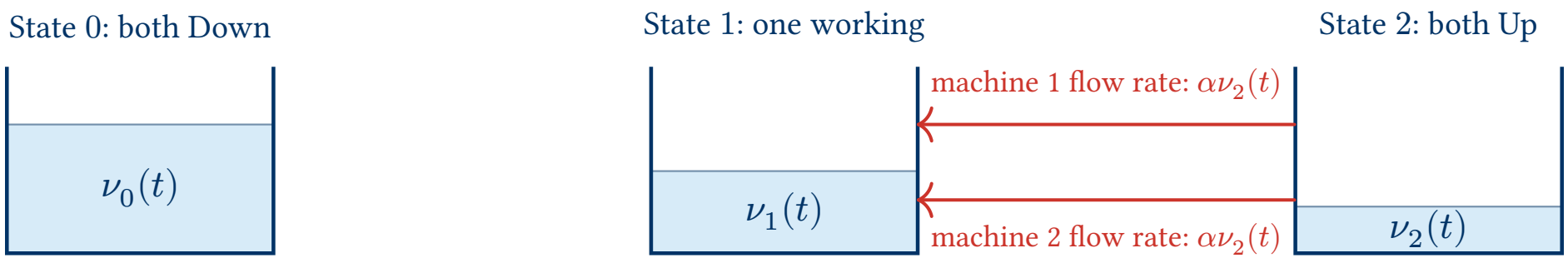
Each clock wins with probability $\frac{1}{2}$, but **the next state is certainly 1**. Starting in state 2, for a short interval h,

$$\mathbb{P}(H_2 \le h, \text{next state} = 1 \mid X(0) = 2) = 1 - e^{-2\alpha h} = 2\alpha h + o(h).$$

Thus, for small h, the probability of a first move from state 2 to state 1 within time h is approximately $2\alpha h$: **exit rate $\times$ interval length**, since the next state is certainly 1.

State 2: two flows into the same destination

Each failure clock contributes a probability flow rate $\alpha\nu_2(t)$ from state 2 into state 1.

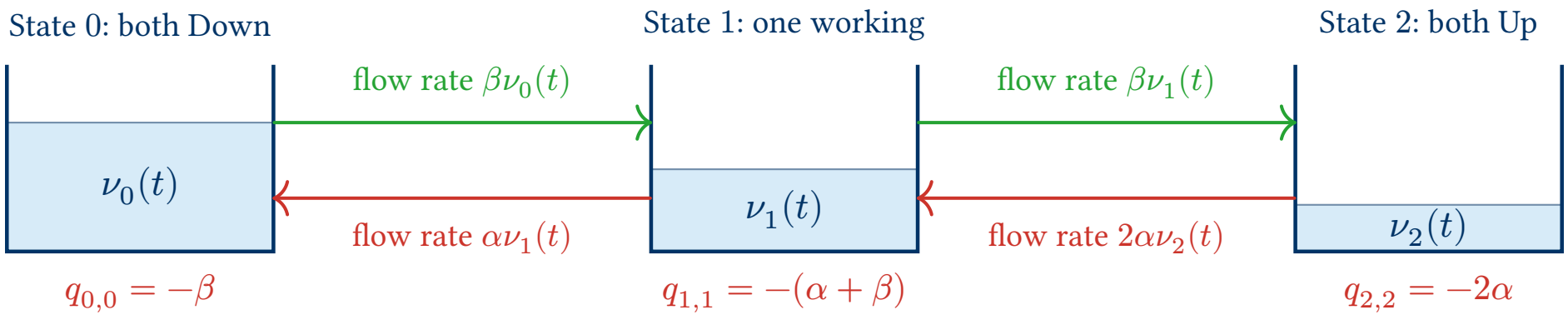


Only flows out of state 2 are shown.

$$\text{flow rate from 2 to 1} = \alpha\nu_2(t) + \alpha\nu_2(t) = 2\alpha\nu_2(t).$$

The combined transition rate is $q_{2,1} = 2\alpha$.

Putting it all together: The two-machine generator



Each probability flow rate equals source probability mass times transition rate. Collect the rates as off-diagonal entries of Q, with rows (source) and columns (destination) ordered $0, 1, 2$:

$$Q = \begin{pmatrix} -\beta & \beta & 0 \\ \alpha & -(\alpha+\beta) & \beta \\ 0 & 2\alpha & -2\alpha \end{pmatrix}.$$

Each diagonal entry is **minus the total exit rate** from its state.

The generator matrix

Generator matrix — Definition 4.2

The **generator matrix**, also called the **infinitesimal generator**, uses **row i** for the source and **column j** for the destination, with $i, j \in \mathcal{S}$. For $i \neq j$, $q_{ij} \geq 0$ is the **transition rate** (or jump rate), and is zero if no direct transition is possible.

$$q_i = \sum_{j \neq i} q_{ij} < \infty, \quad q_{ii} = -q_i, \quad Q = \big(q_{ij}\big).$$

Each row sums to zero. For $i \neq j$, the probability flow rate from state i to state j is $\nu_i(t) q_{ij}$.

For $q_i > 0$, wait an $\mathrm{Exp}(q_i)$ time, then independently choose $j \neq i$ with probability $\frac{q_{ij}}{q_i}$. If $q_i = 0$, stay there forever.

Transition probabilities at a fixed elapsed time

For a forecast at a **fixed elapsed time** t, define

Transition probability matrix **Definition 4.3**

$$P_{ij}(t) = \mathbb{P}(X(t) = j \mid X(0) = i), \quad P(t) = \big(P_{ij}(t)\big).$$

Each row sums to 1, and $P(0) = I$, where I is the identity matrix.

$P_{ij}(t)$ is the probability of being in state j at time t, starting from i, regardless of how many jumps occur during that interval.

The generator gives the initial rate of probability change

Two or more jumps have probability $o(h)$. Thus, for $j \neq i$, the first-jump calculation gives the probability of being in **state j at time h**, to first order.

The diagonal entries follow because each row sums to 1. In state order $0, 1, 2$:

$$P(h) = \begin{pmatrix} 1-\beta h & \beta h & 0 \\ \alpha h & 1-(\alpha+\beta)h & \beta h \\ 0 & 2\alpha h & 1-2\alpha h \end{pmatrix} + o(h) = I + hQ + o(h).$$

Here $o(h)$ applies entry by entry. Since $P(0) = I$,

$$Q = \lim_{h \to 0^+} \frac{P(h) - I}{h} = P'(0^+).$$

Short-time transition probabilities: the general result

Short-time transition probabilities — Proposition 4.1

Let $P(t)$ be the transition matrix of a **finite-state, time-homogeneous CTMC** with generator Q and exit rates $q_i = -q_{ii}$. As $h \to 0^+$,

$$P_{ij}(h) = q_{ij}h + o(h) \quad (j \neq i),$$
$$P_{ii}(h) = 1 - q_i h + o(h).$$

In matrix form, with $o(h)$ understood entry by entry,

$$P(h) = I + hQ + o(h), \quad P'(0^+) = Q.$$

All exit rates are bounded, so two or more jumps in time h have probability $O(h^2) = o(h)$. Thus, only **no-jump and one-jump paths** contribute to the first-order approximation.

Pause and recall

States, clocks, and forecasts

---

1. For one machine, write Q in state order U, D. Explain its first row.
2. For the two-machine count process, take $\alpha = 0.2$ and $\beta = 0.4$ per day. Find the holding-time law and next-state probabilities when $X(t) = 1$.
3. Add a second independent technician to that system. Which row of Q changes, and how?
4. Why can $q_{2,0} = 0$ while $P_{2,0}(t) > 0$ for $t > 0$? What is the initial slope $P'_{2,0}(0^+)$?

Pause here. Try each question before turning to the answers.

Answers and reasoning

States, clocks, and forecasts

---

1. $Q = \begin{pmatrix} -\alpha & \alpha \\ \beta & -\beta \end{pmatrix}$. The first row has failure rate α and diagonal entry $-\alpha$, minus the total exit rate.
2. The total rate is 0.6 per day, so $H_1 \sim \text{Exp}(0.6)$, with mean $\frac{5}{3}$ days. The next state is 0 with probability $\frac{1}{3}$, or 2 with probability $\frac{2}{3}$.
3. Only row 0 changes. Two active repairs give $q_{0,1} = 2\beta$ and $q_{0,0} = -2\beta$.
4. Two successive failures can leave both machines Down at time $t > 0$. They require two jumps, so $P_{2,0}(h) = o(h)$ and $P'_{2,0}(0^+) = q_{2,0} = 0$.

Clocks determine jump rates. Q gives initial rates of probability change; $P(t)$ gives transition probabilities.

4.3. Evolution of State Probabilities

State probabilities

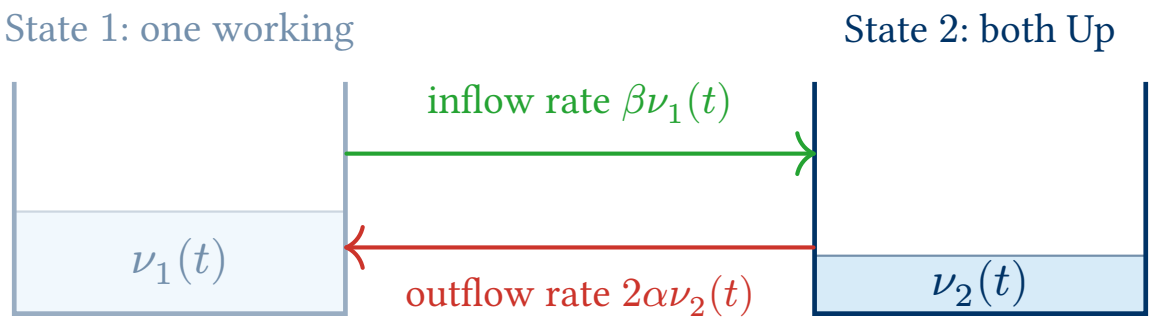
Continue with two machines and one repair technician, with an arbitrary initial distribution.

State probabilities **Definition 4.4**

The **state probability** at time t is $\nu_i(t) = \mathbb{P}(X(t) = i)$, for $i \in \mathcal{S}$.

Account for probability over a short interval

Follow tank 2, which represents both machines working. Tank 2 holds probability mass $\nu_2(t)$.



All flows into and out of tank 2.

Over a short interval h, each transferred amount is approximately **flow rate $\times$ h**:

- **Into tank 2:** repair transfers $\beta\nu_1(t)h$ from tank 1.
- **Out of tank 2:** either machine failing transfers $2\alpha\nu_2(t)h$ to tank 1.

The **change in the amount in tank 2** is the amount entering minus the amount leaving:

$$\nu_2(t+h) - \nu_2(t) = [\beta\nu_1(t) - 2\alpha\nu_2(t)]h + o(h).$$

The derivative is the net probability flow rate

Divide the net change by h to get the **average rate of change** in tank 2 over $[t, t+h]$:

$$\frac{\nu_2(t+h) - \nu_2(t)}{h} = \beta\nu_1(t) - 2\alpha\nu_2(t) + \frac{o(h)}{h}.$$

As $h \to 0^+$, the average rate approaches the **instantaneous rate**, and $\frac{o(h)}{h} \to 0$:

$$\nu_2'(t) = \lim_{h\to 0^+} \frac{\nu_2(t+h) - \nu_2(t)}{h} = \beta\nu_1(t) - 2\alpha\nu_2(t).$$

Follow probability flow rates

At every time t, the derivative is the instantaneous net probability flow rate into the tank. The amount increases when the inflow rate is larger and decreases when the outflow rate is larger.

The same accounting gives the forward equations

Apply inflow rate minus outflow rate to each of the three tanks:

$$\begin{aligned}
\nu_0'(t) &= \alpha\nu_1(t) - \beta\nu_0(t), \\
\nu_1'(t) &= \beta\nu_0(t) + 2\alpha\nu_2(t) - (\alpha+\beta)\nu_1(t), \\
\nu_2'(t) &= \beta\nu_1(t) - 2\alpha\nu_2(t).
\end{aligned}$$

With the row vector $\boldsymbol{\nu}(t) = (\nu_0(t), \nu_1(t), \nu_2(t))$ and the same generator Q,

$$\boldsymbol{\nu}'(t) = \boldsymbol{\nu}(t)Q.$$

These equations hold for **any initial probability distribution** $\boldsymbol{\nu}(0)$.

Combining two elapsed time intervals

Recall $P_{ij}(t) = \mathbb{P}(X(t) = j \mid X(0) = i)$. To combine durations s and t, condition on the state after time s.

Chapman–Kolmogorov equations **Theorem 4.1**

For a time-homogeneous CTMC on state space $\mathcal{S}$, and any $i, j \in \mathcal{S}$ and $s, t \geq 0$,

$$P_{ij}(s+t) = \sum_{k \in \mathcal{S}} P_{ik}(s) P_{kj}(t).$$

Compare the matrix form with the DTMC rule in (2.4):

DTMC: combine m and n steps	CTMC: combine durations s and t
$P^{(m+n)} = P^{(m)} P^{(n)}$	$P(s+t) = P(s)P(t)$

Proof of Theorem 4.1:

Fix $i, j \in \mathcal{S}$ and $s, t \geq 0$.

1. Total probability. Split the forecast according to $X(s) = k$:

$$P_{ij}(s+t) = \sum_{k \in \mathcal{S}} P_{ik}(s) \mathbb{P}(X(s+t) = j \mid X(s) = k, X(0) = i).$$

2. Markov property and time-homogeneity. The current state replaces the earlier history; only the remaining duration t matters:

$$\begin{aligned} \mathbb{P}(X(s+t) = j \mid X(s) = k, X(0) = i) &= \mathbb{P}(X(s+t) = j \mid X(s) = k) \\ &= P_{kj}(t). \end{aligned}$$

3. Matrix multiplication. Substitute into the first equation:

$$P_{ij}(s+t) = \sum_{k \in \mathcal{S}} P_{ik}(s) P_{kj}(t) = [P(s)P(t)]_{ij}.$$

Since i, j are arbitrary, $P(s+t) = P(s)P(t)$. ∎

Chapman–Kolmogorov gives the forward equations

For a **finite-state** CTMC, apply Theorem 4.1 to a final interval h and use Proposition 4.1:

$$\begin{aligned} P(t+h) &= P(t)P(h) \\ &= P(t)[I + hQ + o(h)] \\ &= P(t) + hP(t)Q + o(h). \end{aligned}$$

Subtract $P(t)$, divide by h, and let $h \to 0^+$:

$$\frac{P(t+h) - P(t)}{h} = P(t)Q + \frac{o(h)}{h} \to P(t)Q.$$

Kolmogorov forward equations **Theorem 4.2**

For a finite-state, time-homogeneous CTMC with generator Q, the transition matrix satisfies

$$P'(t) = P(t)Q, \quad P(0) = I.$$

At $t = 0$, the derivative is from the right.

Updating the state distribution

State distribution and its evolution — Proposition 4.2

Let $P(t)$ be the transition matrix of a finite-state CTMC with generator Q. For any initial row distribution $\boldsymbol{\nu}(0)$ and $t \geq 0$,

$$\boldsymbol{\nu}(t) = \boldsymbol{\nu}(0)P(t), \quad \boldsymbol{\nu}'(t) = \boldsymbol{\nu}(t)Q.$$

At $t = 0$, the derivative is from the right.

From any observation time:

DTMC: one step	CTMC: duration h
$\boldsymbol{\nu}^{(n+1)} = \boldsymbol{\nu}^{(n)} P$	$\boldsymbol{\nu}(t+h) = \boldsymbol{\nu}(t)P(h)$

Proof of Proposition 4.2:

Total probability. Conditioning on $X(0)$ gives

$$\nu_j(t) = \sum_i \nu_i(0) P_{ij}(t).$$

Differentiate. Apply Theorem 4.2:

$$\boldsymbol{\nu}'(t) = \boldsymbol{\nu}(0) P'(t) = \boldsymbol{\nu}(0) P(t) Q = \boldsymbol{\nu}(t) Q.$$

∎

For each state, this gives **inflow rate minus outflow rate**:

$$\nu_j'(t) = \sum_{i \neq j} \nu_i(t) q_{ij} - \nu_j(t) q_j.$$

A difference per step becomes a rate per unit time

Recall the DTMC distribution $\boldsymbol{\nu}^{(n)}$ and one-step matrix P from Theorem 2.1.

$\boldsymbol{\nu}^{(n)}$ describes the state after n steps; $\boldsymbol{\nu}(t)$ describes it after elapsed time t.

Transition matrix	DTMC	CTMC
Difference / derivative	$P^{(n+1)} - P^{(n)} = P^{(n)}(P - I)$	$P'(t) = P(t)Q$

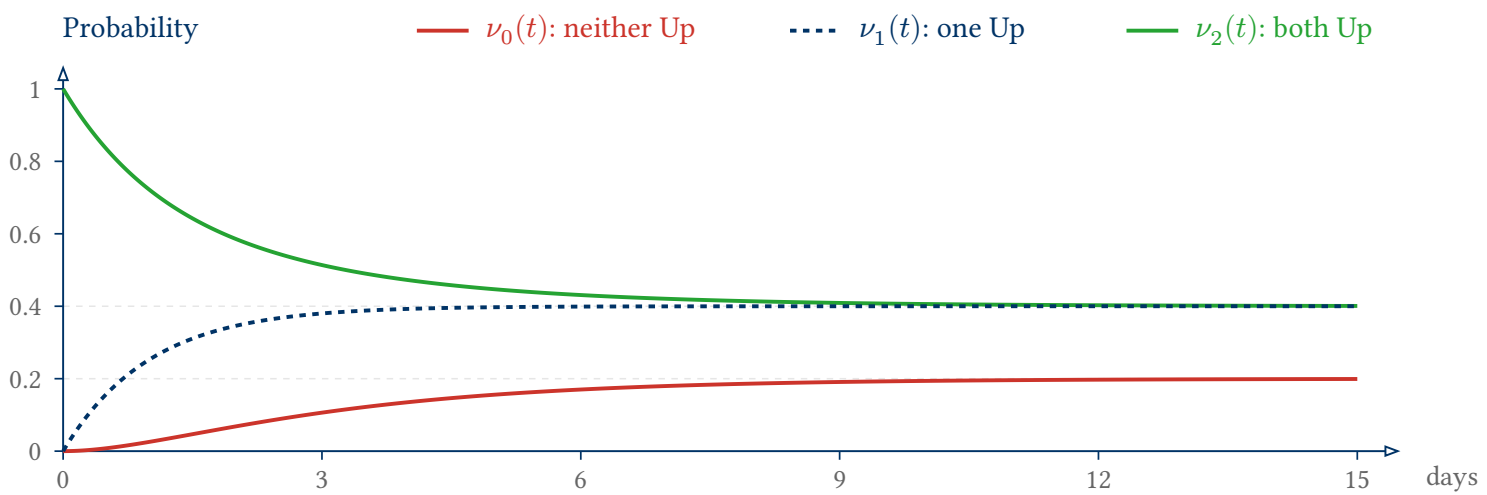
State distribution	DTMC	CTMC
Difference / derivative	$\boldsymbol{\nu}^{(n+1)} - \boldsymbol{\nu}^{(n)} = \boldsymbol{\nu}^{(n)}(P - I)$	$\boldsymbol{\nu}'(t) = \boldsymbol{\nu}(t)Q$

For a DTMC, subtraction gives the change over one step. For a CTMC, $P(h) \to I$ as $h \to 0$: changes shrink with h, and derivatives give rates **per unit time**, governed by Q.

4.4. Long-Run Flow and Performance

What do the reservoir amounts approach?

For illustration, take $\alpha = 0.2$, $\beta = 0.4$ per day and $\boldsymbol{\nu}(0) = (0, 0, 1)$ (both Up).



The state probabilities appear to settle near $(0.2, 0.4, 0.4)$, even as individual systems continue to fail and undergo repair.

Can we find this limiting distribution without solving the full transient forecast?

The limiting distribution

A CTMC is **irreducible** if every state can be reached from every other along positive-rate transitions.

Limiting distribution **Theorem 4.3**

A finite, **irreducible** CTMC has a *limiting distribution* $\boldsymbol{\pi}$ satisfying

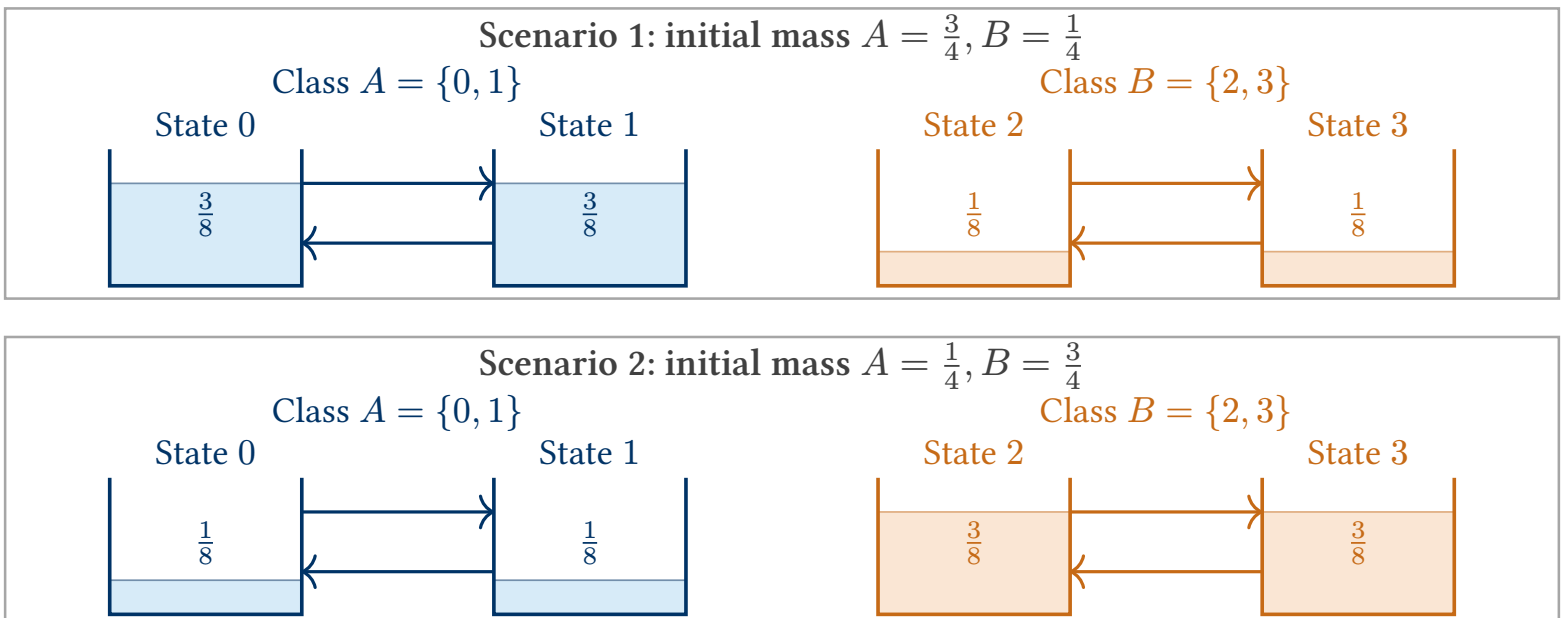
$$P_{ij}(t) \to \pi_j \quad (t \to \infty), \quad i, j \in \mathcal{S}.$$

No aperiodicity condition is needed.

The limit is **independent of the initial distribution**: $\boldsymbol{\nu}(t) \to \boldsymbol{\pi}$ for any $\boldsymbol{\nu}(0)$.

Reducible chains can have different limiting distributions

Four states form two disjoint closed pairs with unit jump rates. Tanks show **limiting amounts**.



Each class keeps its initial total, so the limit depends on $w = \nu_0(0) + \nu_1(0)$:

$$\boldsymbol{\nu}(t) \to \left(\frac{w}{2}, \frac{w}{2}, \frac{1-w}{2}, \frac{1-w}{2}\right) \quad (t \to \infty).$$

Stationary distributions of CTMC

Stationary distribution — Definition 4.5

A probability row vector $\boldsymbol{\pi}$ is a **stationary distribution** if

$$\boldsymbol{\pi} P(t) = \boldsymbol{\pi} \quad \text{for every } t \geq 0.$$

Starting with $\boldsymbol{\pi}$ keeps the state distribution fixed as time passes.

DTMC	CTMC
$\boldsymbol{\pi} P = \boldsymbol{\pi}$	$\boldsymbol{\pi} P(t) = \boldsymbol{\pi}$ for every $t \geq 0$
Hence $\boldsymbol{\pi} P^n = \boldsymbol{\pi}$ for all steps $n \geq 0$.	Every elapsed-time matrix $P(t)$ preserves $\boldsymbol{\pi}$.

DTMC definition: Definition 2.13. Distribution updates: DTMC Theorem 2.1, CTMC Proposition 4.2.

Constant reservoir amounts have zero derivative

By Definition 4.5, starting at $\boldsymbol{\pi}$ keeps $\boldsymbol{\nu}(t) = \boldsymbol{\pi}$ for all t. Recall that **the derivative is the instantaneous net probability flow rate** (Page 258).

Constant amounts have zero derivatives. For tank 2, the earlier derivative equation becomes

$$0 = \nu_2'(t) = \beta\pi_1 - 2\alpha\pi_2.$$

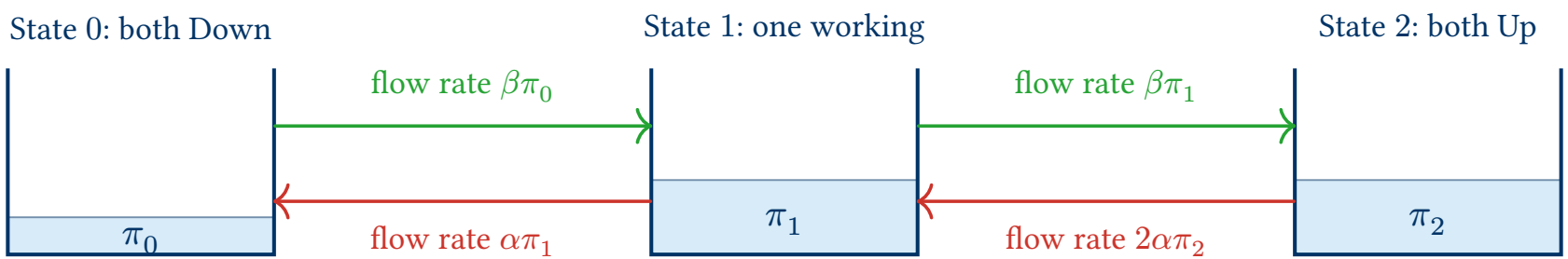
Apply Proposition 4.2 to every tank: $\boldsymbol{\nu}'(t) = \boldsymbol{\nu}(t)Q$. For a **finite-state** CTMC, stationarity is equivalent to

$$\boldsymbol{\pi}Q = \mathbf{0}, \quad \sum_j \pi_j = 1, \quad \pi_j \geq 0.$$

These are the **global balance equations**: total inflow rate equals total outflow rate at each state. With irreducibility, Theorem 4.3 gives a unique stationary distribution that is also the limiting distribution.

Write global balance for all three tanks

Set the derivatives in the two-machine forward equations (Page 259) to zero. At each tank, **total inflow rate equals total outflow rate**:



Probability flows while the amounts stay constant.

$$\begin{aligned}\text{Tank 0:}\quad & \alpha\pi_1 = \beta\pi_0,\\ \text{Tank 1:}\quad & \beta\pi_0 + 2\alpha\pi_2 = (\alpha+\beta)\pi_1,\\ \text{Tank 2:}\quad & \beta\pi_1 = 2\alpha\pi_2.\end{aligned}$$

Each outer tank has only one neighbor, so its global balance equates one inflow rate and one outflow rate.

Solve global balance, then normalize

Adding the tank 0 and tank 2 equations gives the tank 1 equation:

$$\alpha\pi_1 + \beta\pi_1 = \beta\pi_0 + 2\alpha\pi_2.$$

Thus only two balance equations are independent. Use the outer-tank equations and $\pi_0 + \pi_1 + \pi_2 = 1$. With $\alpha = 0.2$ and $\beta = 0.4$ per day,

$$\pi_1 = \left(\frac{\beta}{\alpha}\right)\pi_0 = 2\pi_0, \quad \pi_2 = \left(\frac{\beta}{2\alpha}\right)\pi_1 = \pi_1 = 2\pi_0.$$

Normalize the total probability:

$$\pi_0 + 2\pi_0 + 2\pi_0 = 1 \quad \rightarrow \quad \boldsymbol{\pi} = \left(\frac{1}{5}, \frac{2}{5}, \frac{2}{5}\right).$$

Available means at least one machine is working, so the long-run available-time proportion is

$$\pi_1 + \pi_2 = \frac{4}{5}.$$

Ergodicity: Long-Run Time Proportions.

The reservoir amounts describe many independent copies at a fixed time. For **one system observed from time 0 to T**, define its empirical time proportion by

$$\hat{\pi}_j(T) = \frac{\text{time spent in state } j}{T} = \frac{1}{T}\int_0^T \mathbb{1}\{X(t) = j\}\,\mathrm{d}t.$$

The indicator is one while the system occupies state j. Its integral adds those durations. For $j = 2$, we add every interval in which both machines are Up.

Ergodic theorem for finite irreducible CTMCs **Theorem 4.4**

Let $X(t)$ be a **time-homogeneous, irreducible CTMC** on a **finite state space** $\mathcal{S}$, with finite transition rates. It has a unique stationary distribution $\boldsymbol{\pi}$. For every initial distribution,

$$\hat{\pi}_j(T) \to \pi_j \quad \text{almost surely as } T \to \infty, \quad j \in \mathcal{S}.$$

A late snapshot across many copies and time proportions along one long path give the same $\boldsymbol{\pi}$.

Why is this an ergodic theorem?

Meaning. *Ergodic* is a technical English adjective. Here it means that time averages along a typical single path converge to averages under the stationary distribution.

Origin. The name comes from Boltzmann's *Ergode* in statistical mechanics. The common Greek explanation is *ergon* (work, associated with energy) + *hodos* (path or way). The exact derivation is disputed.

Why the name? The theorem establishes the equality of long-run time averages and equilibrium ensemble averages:

$$\underbrace{\hat{\pi}_j(T)}_{\text{time average along one path}} \quad \to \quad \underbrace{\pi_j}_{\text{probability at equilibrium}} \qquad (T \to \infty).$$

For our two-machine system, $\pi_2 = \frac{2}{5}$: almost every pair spends **40% of its time with both machines Up** in the long run.

American Heritage Dictionary.

Average reward rate

Consider a finite irreducible CTMC. While the system is in state i, reward accumulates at rate r_i.

Total reward is the sum of each state's rate multiplied by the time spent there. Dividing by T gives

$$\frac{\text{total reward earned by time } T}{T} = \sum_i r_i \hat{\pi}_i(T).$$

By ergodicity (Theorem 4.4), $\hat{\pi}_i(T) \to \pi_i$ almost surely. Thus the **long-run average reward rate** is

$$g = \sum_i \pi_i r_i.$$

Weight each state's reward rate by its **fraction of time**. If r_i is in dollars per day, so is g.

Example: average daily reward

In the two-machine system, each working machine earns 1,200 dollars per day. Each Down machine costs 300 dollars per day.

Number working i	Time proportion π_i	Reward rate r_i (dollars/day)
0	$\frac{1}{5}$	-600
1	$\frac{2}{5}$	900
2	$\frac{2}{5}$	2400

$$g = \frac{1}{5}(-600) + \frac{2}{5}(900) + \frac{2}{5}(2400) = 1200 \ \text{ dollars per day.}$$

Over a long period, the system earns an average net reward of **1,200 dollars per day**.

Pause and recall

Forecasts, observations, and performance

1. For an arbitrary initial distribution, identify the events described by $\nu_2(t)$ and $\nu_2(0)e^{-2\alpha t}$. Which probability is larger?
2. Suppose a working machine earns 1,200 dollars per day and a Down machine has no cost. What are the reward rates r_0, r_1, r_2 and the long-run average revenue per day?

Pause here. Try each question before turning to the answers.

Answers and reasoning

Forecasts, observations, and performance

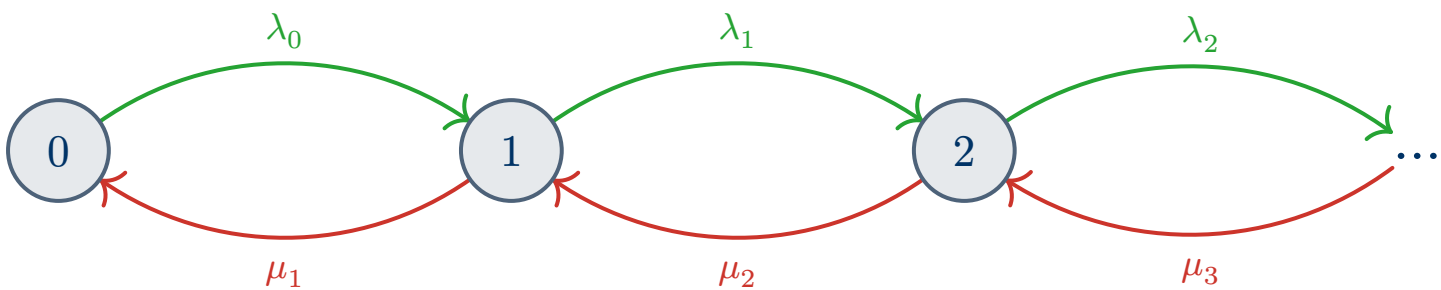
1. $\nu_2(t)$ is the probability that both machines are Up at time t; $\nu_2(0)e^{-2\alpha t}$ requires both to start Up and remain Up throughout. Thus $\nu_2(t) \geq \nu_2(0)e^{-2\alpha t}$: paths with repairs can also end in state 2.
2. $(r_0, r_1, r_2) = (0, 1200, 2400)$ dollars per day. The average revenue is $g = \frac{1}{5}(0) + \frac{2}{5}(1200) + \frac{2}{5}(2400) = 1440$ dollars per day.

Use state probabilities for snapshot questions and time proportions to average reward rates.

4.5. Birth-and-Death Models

A count that changes one at a time

A **birth-and-death process** is a CTMC on $\mathcal{S} = \{0, 1, 2, \cdots\}$. Each jump increases the count by one (a **birth**) or decreases it by one (a **death**).



In state n, the **total birth rate** is λ_n and the **total death rate** is μ_n:

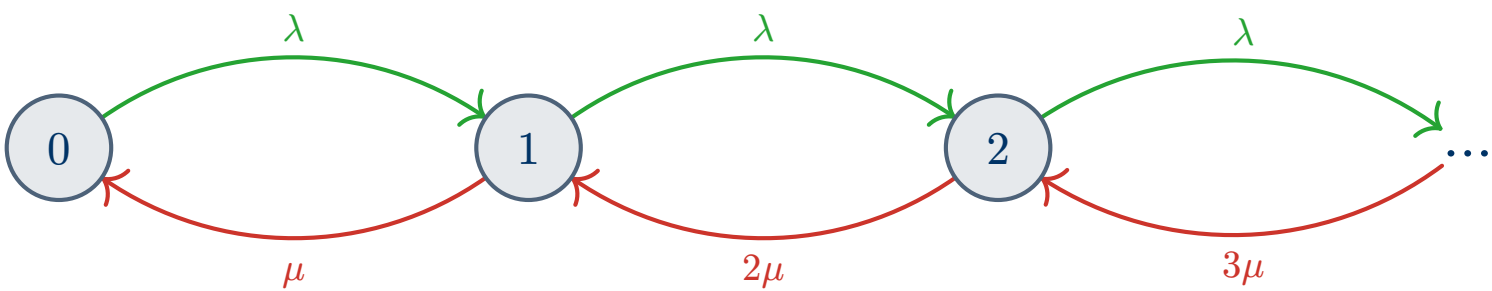
$$q_{n,n+1} = \lambda_n, \quad q_{n,n-1} = \mu_n, \quad q_{n,n} = -(\lambda_n + \mu_n).$$

The downward formula applies for $n \geq 1$. Set $\mu_0 = 0$ because the count cannot fall below zero. All other off-diagonal rates are zero.

A population with immigration and death

Individuals enter a habitat according to a Poisson process of rate $\lambda > 0$. Each has an independent exponential lifetime with rate $\mu > 0$, independent of immigration. The population changes only through immigration and death.

Let $X(t)$ count the individuals alive. With n individuals, ***n* independent death clocks** are active:



$$\lambda_n = \lambda, \quad \mu_n = n\mu \quad (n \geq 0).$$

Each individual's death rate is μ. The **total death rate** is $n\mu$. Memorylessness makes the current count sufficient, regardless of the individuals' ages.

The process is non-explosive: every death removes an initial resident or a previous immigrant.

Simplifying the global balance equations

For a non-explosive process, a stationary distribution $\boldsymbol{\pi}$ satisfies **global balance**:

$$\begin{aligned} \pi_0 \lambda_0 &= \pi_1 \mu_1, \\ \pi_{n(\lambda_n+\mu_n)} &= \pi_{n-1}\lambda_{n-1} + \pi_{n+1}\mu_{n+1} \quad (n \geq 1). \end{aligned}$$

For $n \geq 1$, add equations $0, \cdots, n$. Internal birth and death flows cancel:

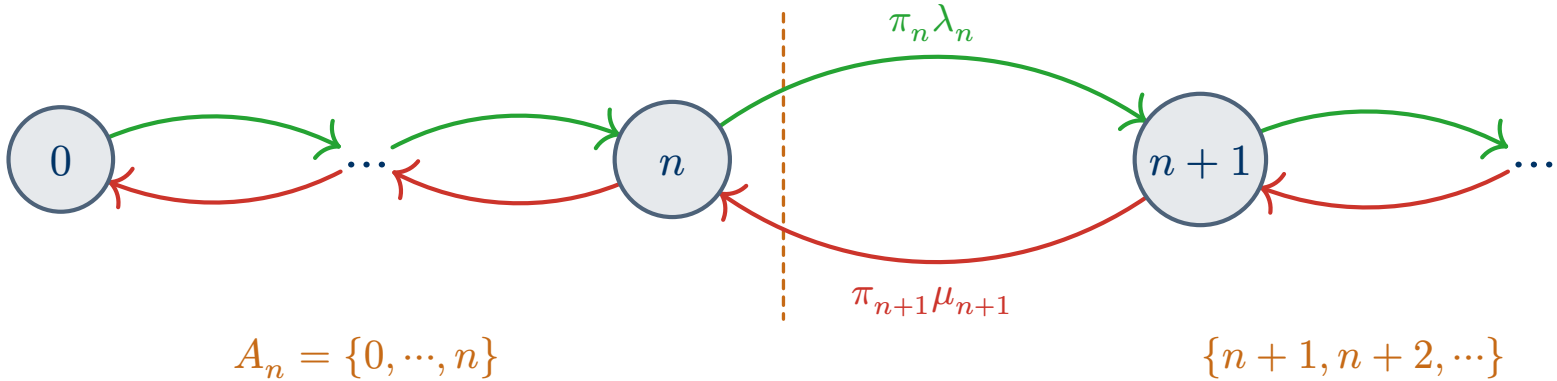
$$\cancel{\sum_{k=0}^{n-1} \pi_k \lambda_k} + \pi_n \lambda_n + \cancel{\sum_{k=1}^{n} \pi_k \mu_k} = \cancel{\sum_{k=0}^{n-1} \pi_k \lambda_k} + \cancel{\sum_{k=1}^{n} \pi_k \mu_k} + \pi_{n+1}\mu_{n+1}.$$

$$\pi_n \lambda_n = \pi_{n+1}\mu_{n+1} \quad (n \geq 0).$$

At $n = 0$, this is already the first global balance equation.

Probability flow across a cut

For any $n \geq 0$, split the states into $A_n = \{0, \cdots, n\}$ and the remaining states.



Only two transitions cross the cut:

- **Out of A_n:** $n \to n+1$, with probability flow rate $\pi_n \lambda_n$.
- **Into A_n:** $n+1 \to n$, with probability flow rate $\pi_{n+1} \mu_{n+1}$.

In stationarity, the probability inside A_n stays constant. Therefore, flow out equals flow in:

$$\pi_n \lambda_n = \pi_{n+1} \mu_{n+1} \quad (n \geq 0).$$

Population probabilities in terms of π_0

Substitute $\lambda_n = \lambda$ and $\mu_n = n\mu$ into the balance equations:

$$\pi_n \lambda = \pi_{n+1}(n+1)\mu \quad \rightarrow \quad \pi_{n+1} = \frac{\lambda}{(n+1)\mu}\pi_n.$$

Starting with π_0, compute the probabilities in order:

$$\pi_1 = \frac{\lambda}{\mu}\pi_0, \quad \pi_2 = \frac{1}{2!}\left(\frac{\lambda}{\mu}\right)^2 \pi_0, \quad \pi_3 = \frac{1}{3!}\left(\frac{\lambda}{\mu}\right)^3 \pi_0.$$

Repeated substitution gives

$$\pi_n = \frac{\pi_0}{n!}\left(\frac{\lambda}{\mu}\right)^n \quad (n = 0, 1, 2, \cdots).$$

The factorial comes from $\mu, 2\mu, \cdots, n\mu$. Normalization determines π_0.

A Poisson stationary population

The probabilities must sum to one. Using the exponential series,

$$1 = \sum_{n=0}^{\infty} \pi_n = \pi_0 \sum_{n=0}^{\infty} \frac{1}{n!} \left(\frac{\lambda}{\mu}\right)^n = \pi_0 e^{\frac{\lambda}{\mu}}.$$

Thus $\pi_0 = e^{-\frac{\lambda}{\mu}}$, and the stationary probabilities are

$$\pi_n = e^{-\frac{\lambda}{\mu}} \frac{1}{n!} \left(\frac{\lambda}{\mu}\right)^n \quad (n = 0, 1, 2, \cdots).$$

The population at a stationary time has a **Poisson distribution** with mean $\frac{\lambda}{\mu}$. This law exists for every $\lambda > 0$ and $\mu > 0$.

The general normalization check: for an irreducible, non-explosive birth-and-death process, the coefficients multiplying π_0 must have a finite sum for normalization to be possible.

The general product formula is in Appendix D (Page 511).

5. Queueing

5.1. M/M/1: Occupancy and Waiting

5.1.1. Stationary Counts

Requests at a gateway

A gateway receives requests that need processing. One processor handles one request at a time. Requests that arrive while it is busy join a waiting line.

Requests arrive at an average rate of **12 per minute**. A service takes **4 seconds on average**. These averages describe demand and processing speed. We also need a model for their randomness.

Let $L(t)$ be the **number of requests in the system**, including the one in service. We will first find its stationary distribution, then study an individual request's time in the system.

How many requests accumulate, and how long does a request wait?

The M/M/1 queue and its service rule

Model arrivals as a **Poisson process** with rate $\lambda = 12$ per minute. Service durations are independent **exponential** variables with rate $\mu = 15$ per minute, independent of arrivals. Thus a service has mean $\frac{1}{\mu} = 4$ seconds.

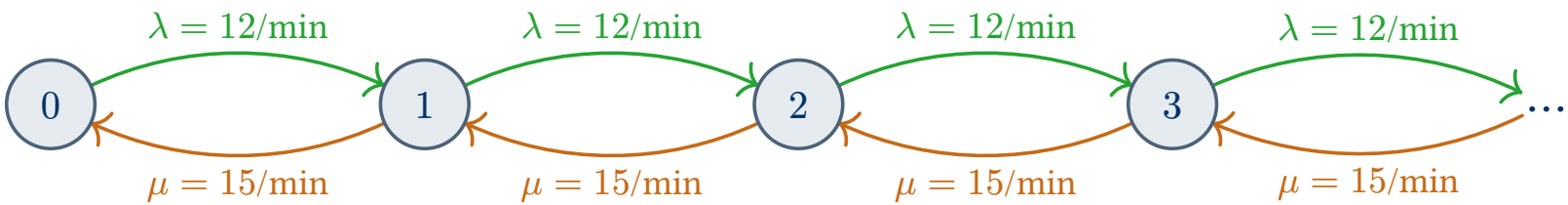
This is an **M/M/1 queue**: the two M's refer to **memoryless** interarrival and service times, and 1 denotes one processor.

- **First-come, first-served (FCFS):** requests are served in arrival order, without interruption.
- There is unlimited waiting space. Every arrival joins and stays until service finishes.
- The processor works whenever requests are present.

The service rule will matter when we turn from occupancy to a customer's wait.

The birth and death rates of one processor

Each arrival raises $L(t)$ by one. Each service completion lowers it by one. Exponential memorylessness makes $L(t)$ a CTMC on $\{0, 1, 2, \cdots\}$.



$L(t)$: requests in the system, including service

$$\lambda_n = \lambda \quad (n \geq 0), \quad \mu_n = \mu \quad (n \geq 1), \quad \mu_0 = 0.$$

In the immigration-death population, n individuals can leave independently, giving total rate $n\mu$. Here only **one service-completion clock** is active. Waiting requests do not depart on their own, so the total downward rate is μ whenever nonempty.

The jump rate is at most $\lambda + \mu$, ensuring non-explosion.

Balance equations and geometric weights

If a stationary distribution $\boldsymbol{\pi}$ exists, the birth-and-death balance from Section 4.5 gives

$$\pi_n \lambda = \pi_{n+1} \mu \quad (n \geq 0).$$

Write $\rho = \frac{\lambda}{\mu}$. Every adjacent ratio is the same:

$$\pi_{n+1} = \rho \pi_n.$$

Starting at state 0, repeated substitution gives

$$\pi_1 = \rho \pi_0, \quad \pi_2 = \rho^2 \pi_0, \quad \pi_3 = \rho^3 \pi_0, \quad \cdots$$

Thus the relative weights are $w_n = \rho^n$, and $\pi_n = \pi_0 \rho^n$ for every $n \geq 0$. We still need to choose π_0 so that the probabilities sum to one.

Stationary distribution and stability

Normalization requires $\pi_0 \sum_{n=0}^{\infty} \rho^n = 1$. The geometric sum is finite precisely when $\rho < 1$:

$$\sum_{n=0}^{\infty} \rho^n = \frac{1}{1-\rho}.$$

M/M/1 stability and stationary count — Theorem 5.1

The M/M/1 queue is **stable**, admitting a stationary probability distribution, if and only if $\lambda < \mu$ (equivalently $\rho < 1$). In stationarity, the **total number in the system** L, including the request in service, has a **geometric distribution** with parameter $1 - \rho$:

$$\mathbb{P}(L = n) = \pi_n = (1-\rho)\rho^n \quad (n = 0, 1, 2, \cdots).$$

At the gateway, $\rho = \frac{12}{15} = 0.8 < 1$, so $\pi_n = 0.2(0.8)^n$ for $n = 0, 1, 2, \cdots$.

From here on, we study this queue **in stationarity**. We can now connect its occupancy to an individual request's waiting and service.

Mean number of customers in the system

Mean number of customers in the system **Corollary 5.1**

For an M/M/1 queue with $\rho = \frac{\lambda}{\mu} < 1$,

$$\ell = \mathbb{E}[L] = \frac{\rho}{1-\rho} = \frac{\lambda}{\mu - \lambda}.$$

Derivation. Using $\pi_n = (1-\rho)\rho^n$,

$$\ell = (1-\rho)\sum_{n=1}^{\infty} n\rho^n = (1-\rho)\frac{\rho}{(1-\rho)^2} = \frac{\rho}{1-\rho}.$$

The sum follows by differentiating the geometric series. At the gateway, $\ell = \frac{0.8}{0.2} = 4$ requests on average.

How many are waiting rather than in service?

The number waiting is $L_q = \max(L - 1, 0)$, with mean $\ell_q = \mathbb{E}[L_q]$.

Number in the system	0	1	2	3	4
Number waiting	0	0	1	2	3
Number in service	0	1	1	1	1

Subtract one **only while the processor is busy**. Expanding the expectation makes this precise:

$$\ell_q = \sum_{n=1}^{\infty} (n-1)\pi_n = \sum_{n=1}^{\infty} n\pi_n - \sum_{n=1}^{\infty} \pi_n$$

$$= \ell - (1 - \pi_0) = \ell - \rho = \frac{\rho^2}{1-\rho}.$$

The subtracted term is the busy probability ρ. At the gateway, the **clock-time mean** is $\ell_q = 4 - 0.8 = 3.2$ waiting requests.

From system counts to a customer's wait

Stationary analysis describes **how many customers are in the system**. A customer arriving now cares about **how long it must wait**.

Let N be the number already present **just before arrival**, excluding the new customer. Under FCFS, these N customers must finish before its own service starts.

Connecting counts to waiting takes two steps:

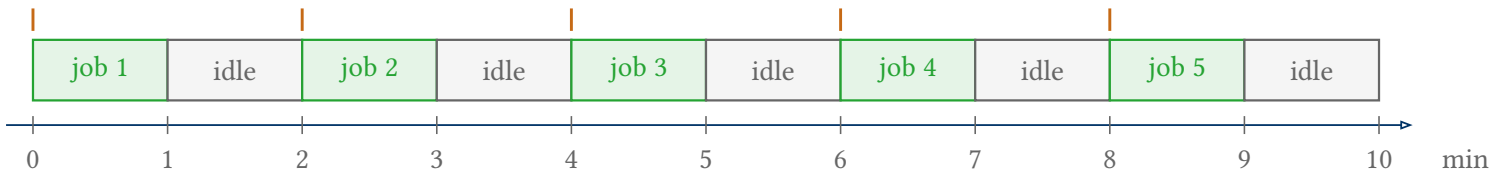
- **Arrival sampling:** relate N to the clock-time count L.
- **Service durations:** turn the work ahead into a waiting time.

We know $\mathbb{P}(L = n) = \pi_n$. **Can we use these same probabilities for what an arrival sees?** First compare two arrival patterns with the same average workload.

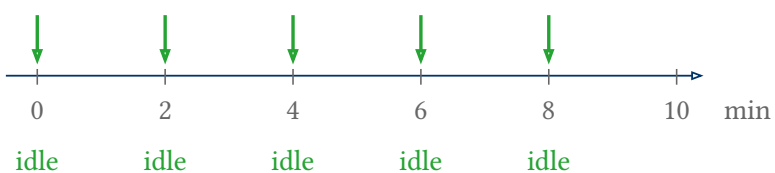
Arrivals need not see the time average

Both examples have **five requests per ten minutes** and **one-minute services**. First, arrivals are regular: one every **two minutes**.

Regular arrivals: every two minutes; one-minute services



Arrivals over the full ten-minute cycle



Clock-time view

$\frac{5}{10} = 50\%$ busy

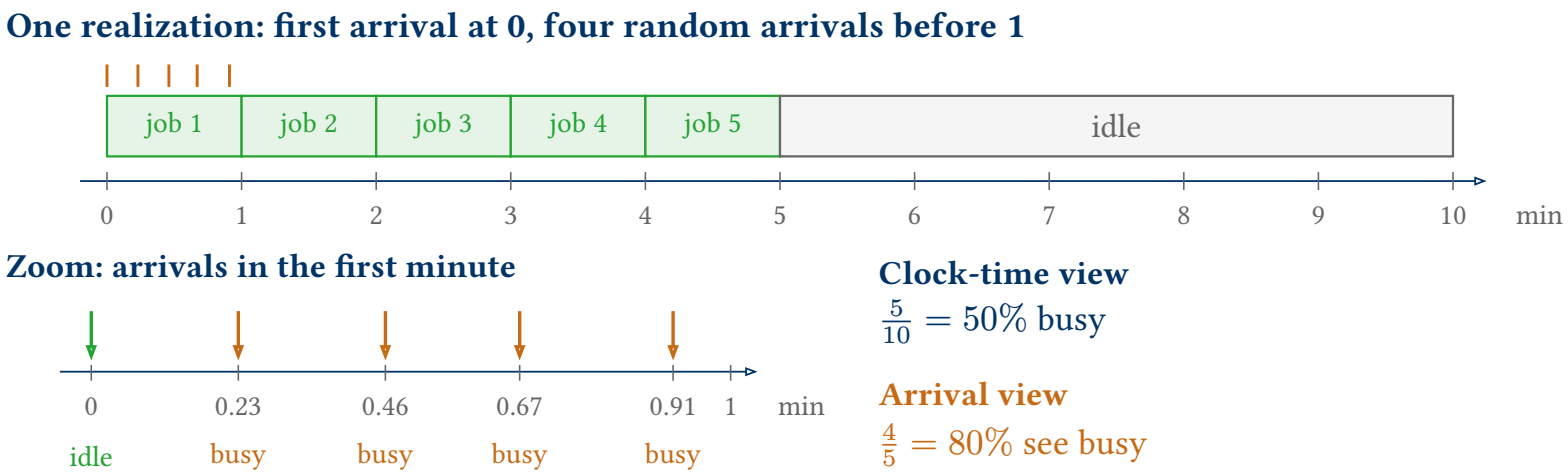
Arrival view

$\frac{0}{5} = 0\%$ see busy

The server is busy **half the time**, but **no arrival has to wait**.

Random bursts can favor busy periods

Keep the same average arrival rate and one-minute services. In each ten-minute cycle, one request arrives at **time 0**. The other four arrive **independently and uniformly within the first minute**.



All five arrive before the first service finishes. The first finds an idle server; the other four wait. Both schedules are busy **50% of clock time**, but random clustering raises the arrival-seen busy fraction from **0% to 80%**.

Only arrival timing has changed. **What would make arrival sampling neutral?**

Why Poisson arrivals sample time fairly

For a homogeneous Poisson process, future arrival increments are independent of the past. After any observed history, the chance of an arrival in the next short interval of length h is approximately λh, whether the queue is empty or crowded.

The queue reflects **past arrivals**. In a nonanticipating system, it cannot reveal the next arrival. Every state therefore receives arrivals at the **same rate per unit of time spent there**.

Over a long observation period of length H, if state n occupies about $\pi_n H$ time units,

$$\text{arrivals seeing state } n \approx \lambda \pi_n H,$$

$$\text{fraction of arrivals seeing state } n \approx \frac{\lambda \pi_n H}{\lambda H} = \pi_n.$$

Poisson arrivals can cluster by chance. Their next-arrival rate still stays λ, regardless of the past. For a server busy half the time, **50% of Poisson arrivals find it busy**.

Poisson arrivals see time averages

Poisson Arrivals See Time Averages (PASTA) **Theorem 5.2**

In stationarity, arrivals see the same state proportions as clock time, provided the system does not anticipate future arrivals.

$$\mathbb{P}(N = n) = \mathbb{P}(L = n) = \pi_n.$$

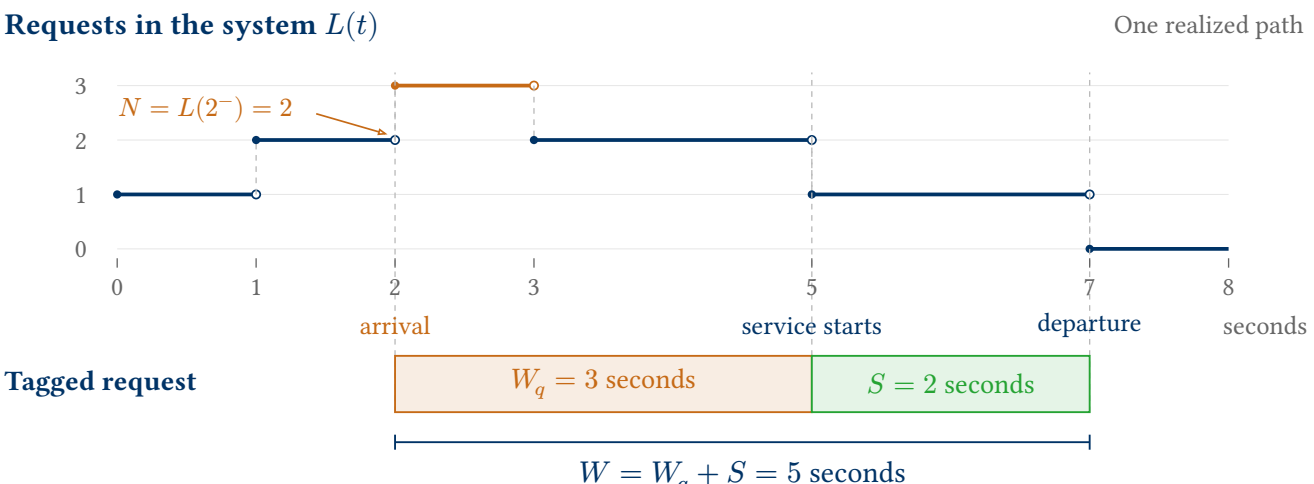
Thus the stationary probabilities also weight the number a request finds **just before arrival**.

Returning to the original gateway, $\mathbb{P}(N \geq 1) = \rho = 0.8$: **80% of arrivals find the processor busy and wait**.

Next, condition on $N = n$ and calculate the time needed for those n **service completions**.

One request: waiting, service and completion

Let W_q be the wait, S the service duration, and $W = W_q + S$ the **sojourn time** of a request finding N customers ahead.



This arrival sees $N = L(2^-) = 2$ **customers ahead**. It waits **3 seconds** and spends **5 seconds in the system**. These durations are one outcome; we next calculate their means.

From the number seen to the waiting time

Mean waiting time and mean sojourn time — Definition 5.1

The **mean waiting time** is $w_q = \mathbb{E}[W_q]$. The **mean sojourn time** is $w = \mathbb{E}[W]$, including service.

If an arrival sees $N = n \geq 1$, it waits for **one residual service and $n - 1$ full services**. FCFS keeps later arrivals behind it.

Exponential memorylessness gives the residual service the same mean $\frac{1}{\mu}$ as a full service. Thus each of the n stages contributes $\frac{1}{\mu}$ to the conditional mean wait:

$$\mathbb{E}[W_q \mid N = n] = \frac{n}{\mu}, \quad n = 0, 1, 2, \cdots.$$

For $N = 0$, service starts immediately. At the gateway, $N = 2$ gives an **8-second mean wait**; the request's own service adds another 4 seconds to its mean sojourn.

Average over what arrivals see

M/M/1 mean waiting and sojourn times — Theorem 5.3

In a stationary FCFS M/M/1 queue with $\rho = \frac{\lambda}{\mu} < 1$,

$$w_q = \frac{\ell}{\mu} = \frac{\rho}{\mu - \lambda}, \quad w = w_q + \frac{1}{\mu} = \frac{1}{\mu - \lambda}.$$

Derivation. PASTA supplies the weights $\mathbb{P}(N = n) = \pi_n$. Thus

$$w_q = \sum_{n=0}^{\infty} \mathbb{E}\left[W_q \mid N = n\right] \pi_n = \frac{1}{\mu} \sum_{n=0}^{\infty} n\pi_n = \frac{\ell}{\mu}.$$

N includes the customer in service. Adding the request's own service gives $w = w_q + \frac{1}{\mu}$.

At the gateway, arrivals see $\ell = 4$ **customers on average**, with four seconds per service stage:

$$w_q = 4 \times 4 = 16 \text{ seconds}, \quad w = 16 + 4 = 20 \text{ seconds}.$$

From conditional means to distributions

The same service stages also give the **full distributions**. Conditional on $N = n$, memorylessness gives independent $E_i \sim \mathrm{Exp}(\mu)$ stages:

$$W_q = \sum_{i=1}^{n} E_i, \quad W = \sum_{i=1}^{n+1} E_i.$$

For $n = 0$, the empty sum is zero: service starts immediately. A fixed positive number of stages gives an **Erlang distribution**.

The number of stages is random. By PASTA,

$$\mathbb{P}(N = n) = (1 - \rho)\rho^n, \quad n = 0, 1, 2, \cdots.$$

We therefore need the distribution of a **geometric number of exponential stages**.

A geometric sum of exponential stages

Geometric sum of exponentials — Lemma 5.1

Let G be geometric on $\{1, 2, \cdots\}$, with

$$\mathbb{P}(G = k) = p(1-p)^{k-1}, \quad k \geq 1, \quad 0 < p \leq 1.$$

Let $E_1, E_2, \cdots$ be independent $\mathrm{Exp}(\mu)$ variables, with $\mu > 0$, independent of G. Then

$$Y = \sum_{i=1}^{G} E_i \sim \mathrm{Exp}(p\mu).$$

Proof by Poisson thinning. Regard the E_i as gaps of a rate-μ Poisson process. Independently mark each event with probability p. The index of the first marked event has the law of G and is independent of the gaps.

The marked events form a Poisson process of rate $p\mu$. Its first-event time is Y, so

$$\mathbb{P}(Y > t) = e^{-p\mu t}, \quad t \geq 0.$$

Memorylessness survives geometric stopping

The lemma combines two memoryless mechanisms:

- **Exponential durations:** while a stage is unfinished, its remaining duration is again $\mathrm{Exp}(\mu)$.
- **Geometric stopping:** after each nonfinal stage, the chance that the next stage is final remains p.

While the total duration is unfinished, both mechanisms restart with the same laws. Thus the remaining total time has the original distribution:

$$\mathbb{P}(Y > s + t \mid Y > s) = \mathbb{P}(Y > t) = e^{-p\mu t}.$$

An independent geometric number of exponential stages is again exponential. This is another consequence of **memorylessness**.

A fixed sum of two exponential stages is Erlang: the geometric stopping rule is essential to this property.

The M/M/1 sojourn-time distribution

The sojourn time contains $N+1$ stages, including the request's own service. By PASTA, for $k \geq 1$,

$$\mathbb{P}(N+1=k) = (1-\rho)\rho^{k-1}.$$

Apply the lemma with $p = 1-\rho$ and stage rate μ. The resulting rate is $(1-\rho)\mu = \mu - \lambda$.

Stationary sojourn time **Corollary 5.2**

In a stationary FCFS M/M/1 queue with $\lambda < \mu$,

$$W \sim \mathrm{Exp}(\mu-\lambda), \quad \mathbb{P}(W > t) = e^{-(\mu-\lambda)t}, \quad t \geq 0.$$

This is a marginal law for one arriving request; successive requests' sojourn times can be dependent.

The M/M/1 waiting-time distribution

An arrival waits zero time when $N = 0$, which has probability $1 - \rho$. For $0 < \rho < 1$, conditional on having to wait,

$$\mathbb{P}(N = k \mid N \geq 1) = (1 - \rho)\rho^{k-1}, \quad k \geq 1.$$

Its wait contains N exponential stages. The geometric-sum lemma gives

$$W_q \mid \{W_q > 0\} \sim \mathrm{Exp}(\mu - \lambda).$$

Stationary waiting time **Corollary 5.3**

In a stationary FCFS M/M/1 queue with $\lambda < \mu$,

$$\mathbb{P}(W_q = 0) = 1 - \rho, \quad \mathbb{P}(W_q > t) = \rho e^{-(\mu-\lambda)t}, \quad t \geq 0.$$

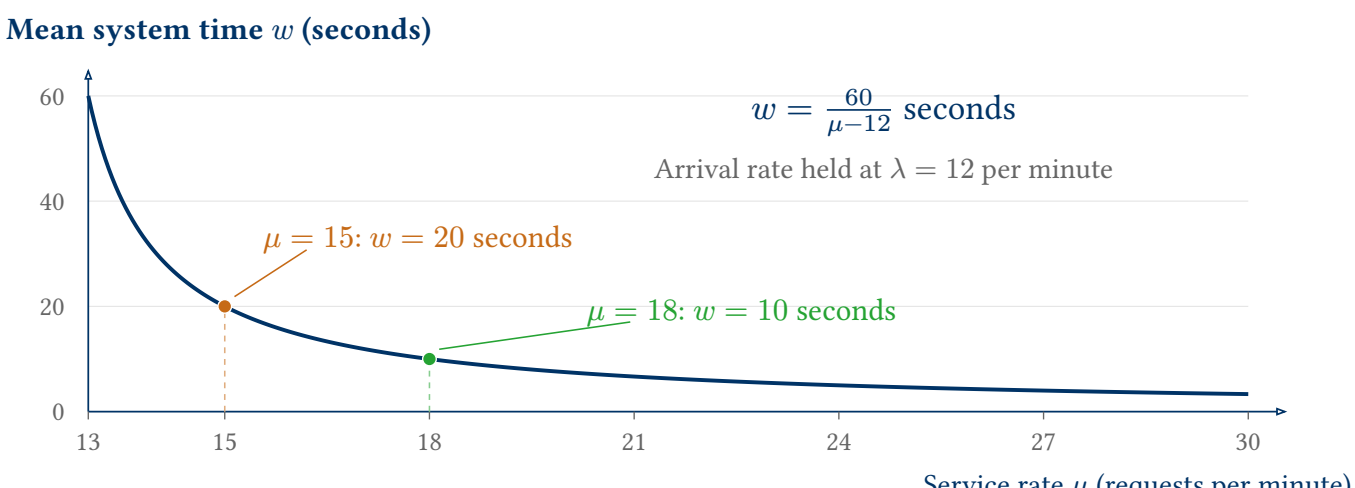
Waiting time has a **probability mass at zero** and an exponential positive part. At the gateway, 20% start service immediately and 80% wait.

5.1.5. Capacity and Targets

? How much does extra capacity reduce mean delay?

For the stationary M/M/1 gateway, hold $\lambda = 12$ per minute and increase μ from 15 to 18 per minute.

By Theorem 5.3, $w = \frac{1}{\mu - \lambda}$. A **20%** capacity increase doubles spare capacity, so w **halves from 20 to 10 seconds**.



? What completion time do 95% of arrivals meet?

At the original gateway ($\lambda = 12$, $\mu = 15$ per minute), the mean sojourn time is **20 seconds**. How often is that exceeded, and what 95% completion-time quote should we give?

By Corollary 5.2, $\mathbb{P}(W > t) = e^{-3t}$ when t is in minutes.

The fraction exceeding the mean is

$$\mathbb{P}(W > 20 \text{ s}) = e^{-3\times(\frac{1}{3})} = e^{-1} \approx 36.8\%.$$

For the 95% quote, set the tail probability to 5%:

$$e^{-3t_{0.95}} = 0.05 \quad \to \quad t_{0.95} = \frac{-\ln(0.05)}{3} \approx 0.999 \text{ min}.$$

The 95% quote is **about 60 seconds**, including waiting and service. About 5% of arrivals exceed it in the long run.

? How much capacity meets a 30-second target?

Keep $\lambda = 12$ per minute. What service rate μ makes **at least 95% of requests finish within 30 seconds**?

By Corollary 5.2, a deadline $t > 0$ with allowed exceedance probability $0 < \varepsilon < 1$ requires

$$e^{-(\mu-\lambda)t} \le \varepsilon \quad \leftrightarrow \quad \mu \ge \lambda + \frac{-\ln(\varepsilon)}{t}.$$

Here $\varepsilon = 0.05$ and $t = 0.5$ minutes, so

$$\mu \ge 12 + \frac{-\ln(0.05)}{0.5} \approx 17.99 \ /\text{min}.$$

Choosing $\boldsymbol{\mu = 18}$ **per minute** meets the target: its mean sojourn time is 10 seconds and its 95th percentile is about 30 seconds.

Principle to remember

From system counts to customer delay

For our FCFS M/M/1 queue, **waiting is the service work found ahead of an arrival**.

- **Stationary counts:** $\rho < 1$ gives a geometric count L in the system.
- **Arrival sampling:** PASTA gives the pre-arrival count N the same law.
- **Service work:** memorylessness makes the residual service exponential. N stages give W_q; $N + 1$ stages, including own service, give W.

A stable queue can still have long delays. As spare capacity $\mu - \lambda$ shrinks, both mean sojourn time and completion percentiles grow.

Use a tail probability to assess a completion-time target.

Pause and recall

Connecting counts, waiting and service targets

---

1. We know the stationary count law π_n. Can we use it for the count just before arrival? What justifies that step?
2. Why does an arrival seeing $N = n \geq 1$ wait for n service stages, although one service has already begun? How do we obtain the mean over all arrivals?
3. A fixed sum of several exponential stages is Erlang. Why is W exponential, while W_q can be zero?
4. A queue has $\rho < 1$. Is that enough to meet a completion-time target? What quantity should guide the capacity choice?

Explain the reasoning behind each step before checking the answers.

Answers and reasoning

Connecting counts, waiting and service targets

---

1. π_n describes clock time. PASTA transfers it to arrivals for a stationary, nonanticipating system with Poisson input.
2. The residual service also counts. Memorylessness gives each stage mean $\frac{1}{\mu}$; averaging with the PASTA weights yields $w_q = \frac{\ell}{\mu}$.
3. $N + 1$ is positive geometric, so the geometric-sum lemma gives exponential W. Waiting is zero for $N = 0$ and exponential conditional on being positive.
4. A stable queue can have long delays when $\mu - \lambda$ is small. A deadline target concerns the tail $\mathbb{P}(W > t)$, which a mean alone does not specify.

Next: how do mean population, throughput and mean sojourn time fit together?

5.2. Little's Law: Population, Throughput and Time

5.2.1. Population, Flow and Duration

The mean population and mean sojourn time in Section 5.1 satisfy

$$\ell = \frac{\lambda}{\mu - \lambda}, \quad w = \frac{1}{\mu - \lambda}, \quad \text{so} \quad \ell = \lambda w.$$

> **? Is this relationship special to M/M/1?**
>
> Would it survive if arrivals were not Poisson, service were not exponential, or customers were served in a different order?

The model gave us the individual averages. We will now connect them by **counting customers and the time they spend inside**.

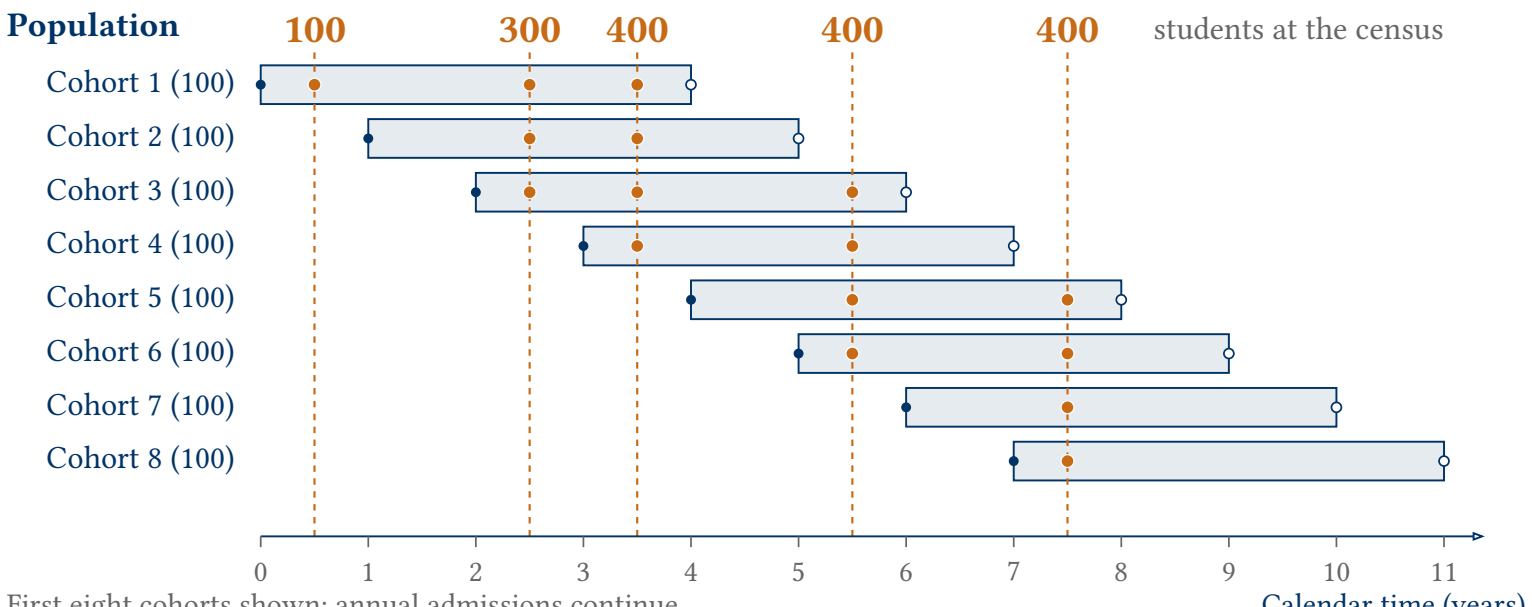
First consider a system whose population and duration are easy to picture: a university programme.

One hundred entrants a year, four years enrolled

? How many students are enrolled?

A programme admits **100 students each year**, and everyone stays **four years**.

Each vertical census line counts the cohorts still enrolled.



First eight cohorts shown; annual admissions continue.

From cohorts to Little's law

Once the fourth cohort enters, **four cohorts overlap**: $4 \times 100 = 400$ students. Each year, 100 students graduate and 100 new students enter, keeping the total at 400.

$$\underbrace{400 \ \ \text{students}}_{\ell} = \underbrace{100 \ \ \text{students/year}}_{\lambda_{\text{eff}}} \times \underbrace{4 \ \ \text{years}}_{w}.$$

Admissions supply the throughput λ_{eff}; years enrolled give the sojourn time w.

The cohort picture connects the number enrolled to both **how quickly students enter** and **how long they remain**.

The same relationship when stays vary

Each year, **100 students enter**: **50 stay three years** and **50 stay five years**.

- **Actual admission rate:** $\lambda_{\text{eff}} = 100$ students/year.
- **Mean total stay per entrant:** $w = \frac{50\times3+50\times5}{100} = 4$ years. Each entrant counts once in this average.

In steady operation, **three cohorts of 50** three-year students and **five cohorts of 50** five-year students overlap:

$$\ell = 3 \times 50 + 5 \times 50 = 400 \text{ students.}$$

Little's law in words

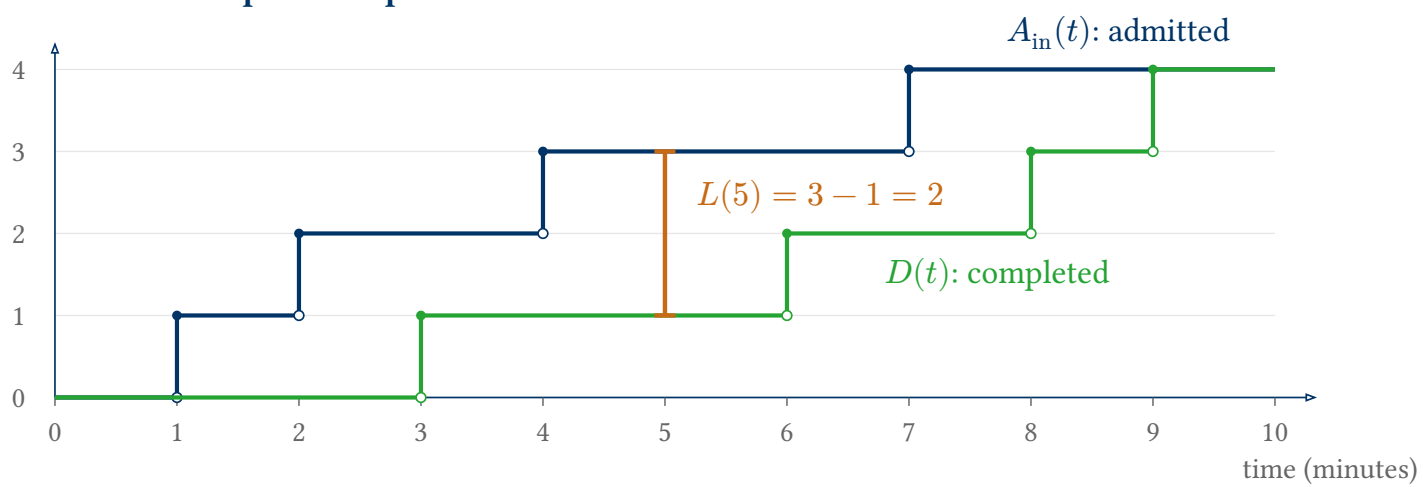
In steady operation, the average number in a system equals its throughput multiplied by the average time spent inside.

$$\ell = \lambda_{\text{eff}} w = 100 \times 4 = 400.$$

One record, two cumulative counts

Four requests arrive at minutes $1, 2, 4, 7$ and complete at $3, 6, 8, 9$. The system starts and ends empty in $[0, 10]$; no request abandons.

Admitted and completed requests



The vertical gap counts the requests still inside. At minute 5, **three have entered, one has left, and two remain**. We first identify the flow through this system, then count how long its customers stay.

Conservation identifies the throughput

Let $A_{\text{in}}(t)$ count admissions and $D(t)$ count all departures by time t. In the four-request record, every departure is a completion.

$$L(t) - L(0) = A_{\text{in}}(t) - D(t).$$

Divide by elapsed time. If finite long-run rates exist and accumulation grows more slowly than t,

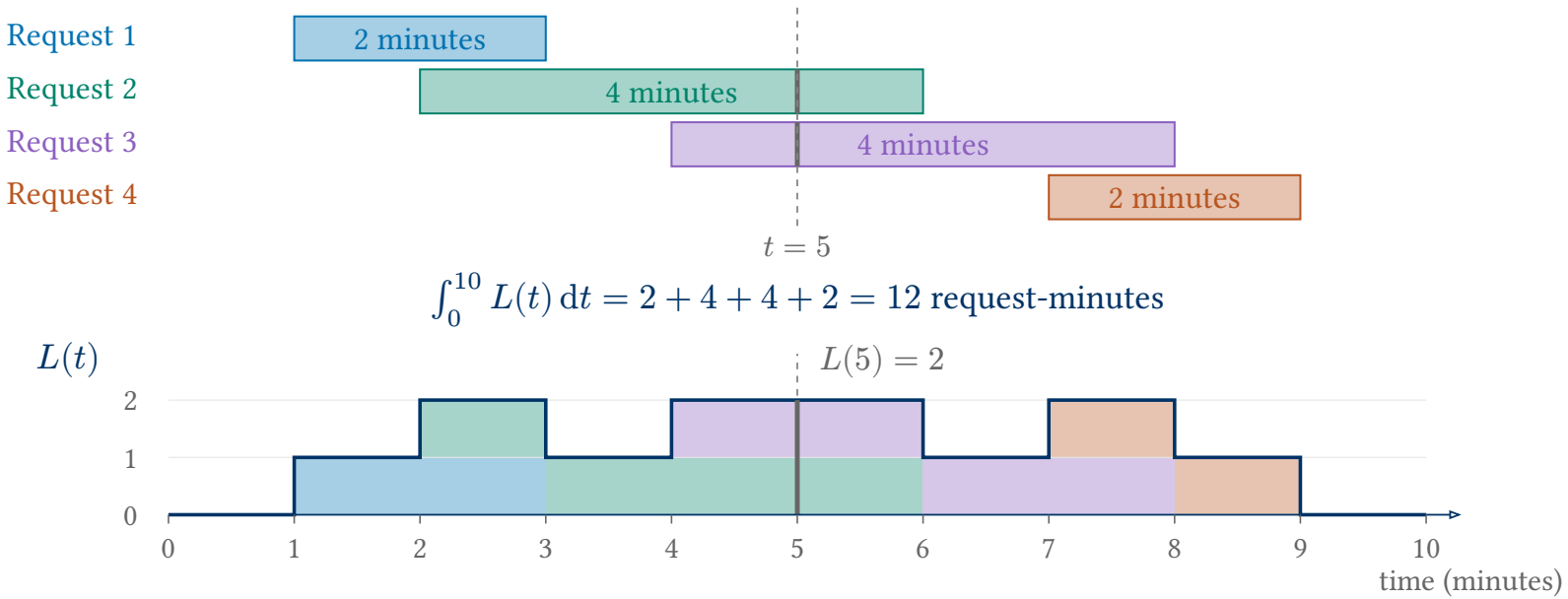
$$\frac{A_{\text{in}}(t)}{t} - \frac{D(t)}{t} = \frac{L(t) - L(0)}{t} \to 0.$$

Thus the admitted and departure rates agree. Write their common value as $\boldsymbol{\lambda_{\text{eff}}}$**, the throughput**. It measures the flow that actually passes through the system; service capacity is the maximum processing rate. If customers abandon, include that exit stream in $D(t)$.

Conservation identifies **how many pass through per unit time**. To connect this rate to population, we must also count **how long each customer remains**.

The same customer-time, counted two ways

Return to the same four requests. Count horizontally by each stay, or vertically by the number present.



One minute with two requests present contributes **two request-minutes**. Both counts give **12 request-minutes**: this is the area we will divide in two different ways.

Three averages from one area

For the complete record, $H = 10$ minutes and $m = 4$ requests contribute **12 request-minutes**.

Average	Calculation	Value
Population ℓ_H	$\frac{12}{10}$	1.2 requests
Throughput λ_H	$\frac{4}{10}$	0.4 requests/minute
Sojourn time w_H	$\frac{12}{4}$	3 minutes

For any window that starts and ends empty, with $m > 0$ complete stays $W_1, \cdots, W_m$,

$$\ell_H = \frac{1}{H}\int_0^H L(t)\,\mathrm{d}t = \frac{1}{H}\sum_{i=1}^{m} W_i = \underbrace{\frac{m}{H}}_{\lambda_H}\underbrace{\frac{1}{m}\sum_{i=1}^{m} W_i}_{w_H}.$$

The product is exact for this record. No distributional assumption entered the calculation.

Little's law in the long run

An arbitrary window can cut through a stay. A request arriving at minute 9 and leaving at minute 12 contributes **one minute** to the area over $[0, 10]$, but its full stay is **three minutes**.

Little's law **Theorem 5.4**

For the same customer population and boundary, suppose the long-run averages exist and are finite, with $\lambda_{\text{eff}} > 0$. If the customer-time discrepancy from stays crossing the window edges, divided by the window length, tends to zero, then

$$\ell = \lambda_{\text{eff}} w.$$

Here ℓ averages population over **clock time**, w averages stays over **customers**, and λ_{eff} is their flow through the system.

Poisson arrivals, exponential service and FCFS are not needed. PASTA relates arrival sampling to clock-time sampling; Little's law counts the same customer-time in two ways.

5.2.3. Applying Little's Law

? Which population matches the time we measure?

Apply Theorem 5.4 to waiting alone, service alone, or the whole system.

Follow all admitted customers, each of whom completes service. Let ℓ_s be the mean number in service.

Boundary	Mean number	Mean duration	Little's law
Waiting area	ℓ_q	w_q	$\ell_q = \lambda_{\text{eff}} w_q$
Service positions	ℓ_s	$\mathbb{E}[S]$	$\ell_s = \lambda_{\text{eff}} \mathbb{E}[S]$
Whole system	ℓ	w	$\ell = \lambda_{\text{eff}} w$

The parts add up: $\ell = \ell_q + \ell_s$ and $w = w_q + \mathbb{E}[S]$. Keep **zero waits** for customers served immediately when averaging w_q over all admitted customers.

Why two waiting formulas use different counts

For the original unlimited FCFS M/M/1 queue, $\lambda_{\text{eff}} = \lambda$. Compare Theorem 5.3 with Theorem 5.4:

Relationship	What is being counted?
$w_q = \frac{\ell}{\mu}$	The **work ahead of an arrival**: all customers present, including the remaining service of the one being served. PASTA and memorylessness give this M/M/1 result.
$\ell_q = \lambda w_q$	The **waiting population over time**: only customers who are waiting contribute to ℓ_q. Little's law counts their waiting-time area.

The customer in service contributes to an arrival's future wait, but is no longer part of the waiting population. Both formulas hold; they count different things.

The rate and mean must use the same customers

? Applications or admissions?

A steady programme receives 200 applications per year and admits 100 students per year. Admitted students stay four years on average. Can either rate give the mean enrollment?

Average over **admitted students**: $\lambda_{\text{eff}} = 100$ students/year and $w = 4$ years.

$$\ell = \lambda_{\text{eff}} w = 100 \times 4 = 400 \quad \text{students.}$$

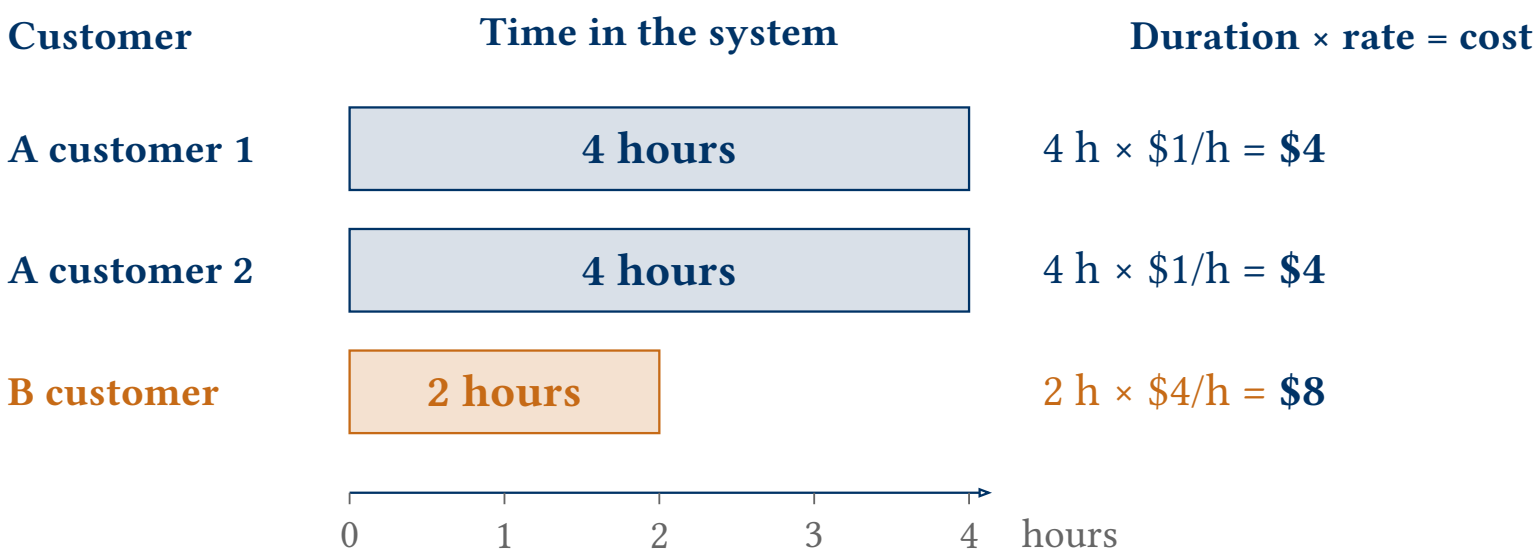
Average over **all applicants**: $\lambda_{\text{app}} = 200$/year, with **zero enrollment time** for rejected applicants.

$$w_{\text{app}} = \frac{100 \times 4 + 100 \times 0}{200} = 2 \quad \text{years}, \quad \ell = \lambda_{\text{app}} w_{\text{app}} = 200 \times 2 = 400.$$

Either rate works with its matching mean. Using 200×4 mixes all applicants with the mean stay of admitted students.

Class costs: price each customer's time

A costs **\$1 per customer-hour** and B costs **\$4 per customer-hour**. This four-hour cycle repeats.

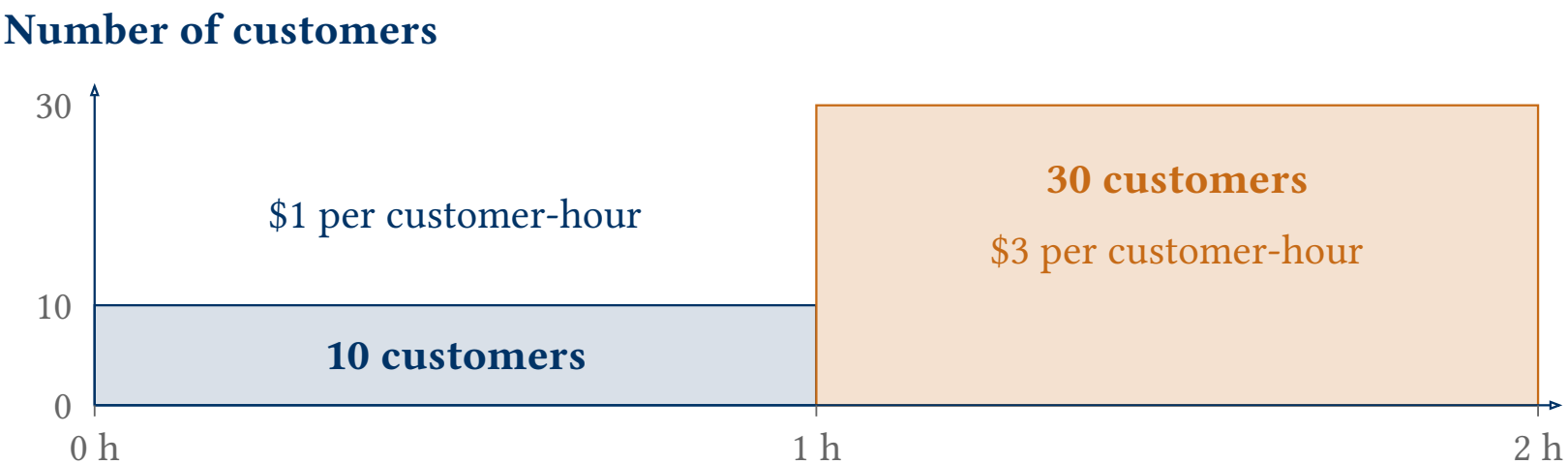


$$\text{total cost} = 4 + 4 + 8 = 16 \ \text{ dollars}, \quad \text{mean cost rate} = \frac{16}{4} = 4 \ \text{ dollars/hour}.$$

Here $\ell_A = 2$ and $\ell_B = 0.5$, so $c_A \ell_A + c_B \ell_B = 1 \times 2 + 4 \times 0.5 = 4$ dollars/hour.

Changing costs: calculate each hour separately

In each two-hour cycle, assume a cost of **$1 per customer-hour first**, then **$3**.



$10 \times \$1/\text{h} \times 1\ \text{h} = \mathbf{\$10}$ $30 \times \$3/\text{h} \times 1\ \text{h} = \mathbf{\$90}$

$$\text{mean cost rate} = \frac{10 + 90}{2} = 50 \ \text{ dollars/hour}.$$

For each interval, multiply **customers present × cost per customer-hour × hours**. **Add the interval costs, then divide by the total elapsed time.**

Principle to remember

Count the same customer-time

Mean population = throughput × mean time inside.

Count customers in the **same part of the system** for which you measure their time.

- **Which part?** The waiting area pairs ℓ_q with w_q. The whole system pairs ℓ with w.
- **Which customers?** The rate and mean duration must use the same customer group.
- **Include zero durations:** zero waits for immediate service, and zero enrollment time for rejected applicants when averaging over all applicants.

Conservation identifies the flow. Counting customer-time gives Little's law.

Little's law connects the averages. A model predicts how an intervention changes them.

Pause and recall

Population, throughput and time

---

1. What does customer conservation tell us? What additional observation gives Little's law?
2. Does Little's law need PASTA or exponential service? What matters when passing to long-run averages?
3. How should rejected requests and immediately served requests enter admitted-customer waiting calculations?
4. If processing becomes faster, can Little's law alone predict the new mean sojourn time?

Explain what is being counted before choosing a formula.

Answers and reasoning

Population, throughput and time

1. Conservation relates entry, exit and accumulation. Little's law also counts customer-time: the area under population equals the sum of complete stays.
2. Neither is needed. Use finite long-run averages and negligible customer-time contributions at the window edges, with a consistent population and boundary.
3. Rejected requests are outside the admitted population. Immediately served requests remain in it, with zero waiting time; use admitted throughput.
4. No. The identity links the new population, throughput and duration. A queueing model must determine how those quantities change.

Next: Section 5.3 uses these tools to study admission limits.

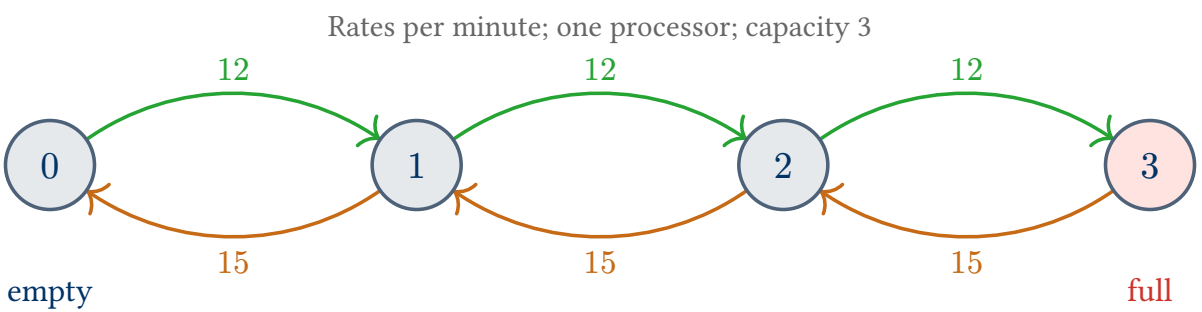
5.3. Finite Capacity and Admission

A gateway with a finite buffer

For the gateway in Section 5.1, keep offered arrivals at $\lambda = 12$/min and service at $\mu = 15$/min. Now impose **capacity three, including service**. An arrival finding it full leaves immediately.

This **M/M/1/3 queue** is a birth-and-death process on $\{0, 1, 2, 3\}$.

Birth rates are $\lambda_n = 12$ for $n < 3$, with $\lambda_3 = 0$. Death rates are $\mu_n = 15$ for $n = 1, 2, 3$, with $\mu_0 = 0$.



Arrivals at full capacity leave the state unchanged.

Rejected arrivals leave the count unchanged, so the exit rate from state 3 is 15/min.

Stationary occupancy with a finite buffer

The birth-and-death recursion gives $\pi_{n+1} = 0.8\pi_n$ for $n = 0, 1, 2$. The weights are $w_n = 0.8^n$, with total $Z_3 = 1 + 0.8 + 0.64 + 0.512 = 2.952$. Normalize by this finite sum: $\pi_n = \frac{w_n}{Z_3}$.

Requests present	0	1	2	3
Relative weight w_n	1	0.8	0.64	0.512
Probability π_n	$\frac{125}{369}$	$\frac{100}{369}$	$\frac{80}{369}$	$\frac{64}{369}$

$$\ell = \pi_1 + 2\pi_2 + 3\pi_3 = \frac{452}{369} \approx 1.225, \quad \ell_q = \pi_2 + 2\pi_3 = \frac{208}{369} \approx 0.564.$$

? How do we turn these counts into mean delays?

Little's law gives $w = \frac{\ell}{\text{rate}}$ and $w_q = \frac{\ell_q}{\text{rate}}$. For mean delays of **admitted requests**, should we use all **12 offered arrivals per minute**, or only the arrivals that **enter the gateway**? Why?

Offered, admitted and completed requests

To apply Little's law to **admitted requests**, first find their **entry rate**. By PASTA, the fraction of offered arrivals rejected is $\pi_3 = \frac{64}{369} \approx 17.34\%$.

Stream	Rate calculation	Requests/minute
Offered	λ	12
Rejected	$\lambda\pi_3$	2.08
Admitted	$\lambda_{\text{eff}} = \lambda(1-\pi_3)$	9.92

Every admitted request eventually completes service. In steady state, rate conservation gives **completed rate = admitted rate**:

$$\mu(1-\pi_0) = 15 \times \frac{244}{369} = 12 \times \frac{305}{369} = \lambda_{\text{eff}}.$$

Little's law now gives $\ell = \lambda_{\text{eff}} w$ and $\ell_q = \lambda_{\text{eff}} w_q$ for **admitted requests**. It needs only the average admitted rate. The admitted stream need not be Poisson.

Delay for requests that are admitted

Now divide each mean count by the **admitted throughput** $\lambda_{\text{eff}} \approx 9.919/\text{min}$. This gives the mean system time w and waiting time w_q **per admitted request**:

$$w = \frac{\ell}{\lambda_{\text{eff}}} \approx \frac{1.225}{9.919} \ \text{min} \approx 7.41 \ \text{s}.$$

$$w_q = \frac{\ell_q}{\lambda_{\text{eff}}} \approx 3.41 \ \text{s}, \quad w = w_q + 4 \ \text{s}.$$

What would dividing by 12 measure? It gives **6.12 seconds per offered request**, with zero time inside for rejected requests.

Our target is time **per admitted request**: divide by **9.92/min** to obtain **7.41 seconds**.

Has the smaller system improved service?

Hold the offered rate at 12/min and the processor rate at 15/min.

Measure	Unlimited capacity	Capacity three
Fraction rejected	0%	17.34%
Throughput	12/min	9.92/min
Actual busy fraction	80%	66.12%
Mean admitted system time	20 s	7.41 s
Mean admitted waiting time	16 s	3.41 s

Offered load $a = \frac{\lambda}{\mu} = 0.8$ is unchanged, but rejected requests create no service work. The finite system's actual utilization is $\frac{\lambda_{\text{eff}}}{\mu} = 1 - \pi_0 \approx 66.12\%$.

Admitted requests finish sooner, but roughly one in six offered requests is rejected. **Specify the objective before choosing a buffer limit.** Waiting space does not increase processing speed. Next, Section 5.4 adds processors and asks how requests should use them.

5.4. Resource Pooling

How can the gateway serve more demand?

Demand rises from **12 to 24 requests per minute**. One processor serves at **15 per minute** on average. If every request must be served, one processor cannot keep up. More waiting space only stores the backlog.

Suppose we add an identical second processor. Their **aggregate capacity is 30 per minute**, which exceeds demand.

> **? How should we operate the two processors?**
>
> 1. Give each processor its own queue? If so, how should we assign arriving requests?
> 2. Let both processors serve one common queue?

The same demand and the same processors can give **different waiting times**. We must decide how requests gain access to the available capacity.

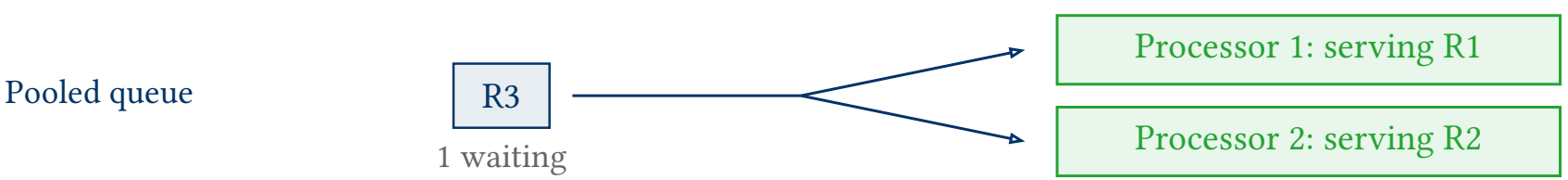
Can the same processors give different delays?

Dedicated queues: three requests, two processors



Requests wait while capacity at processor 2 is unused.

Resource pooling: the same three requests and two processors



Resource pooling lets the next waiting request use **any available processor**. With **dedicated queues**, each request stays with its assigned processor.

What is the dedicated-queue benchmark?

Assume Poisson arrivals, independent exponential services, unlimited buffers and FCFS.

Under **independent random routing**, each arrival chooses either processor with probability $\frac{1}{2}$.

Poisson thinning gives **two independent M/M/1 queues**, each with arrival rate 12/min. Each reproduces our original gateway model:

$$w_q = 16 \text{ s}, \quad w = 20 \text{ s}.$$

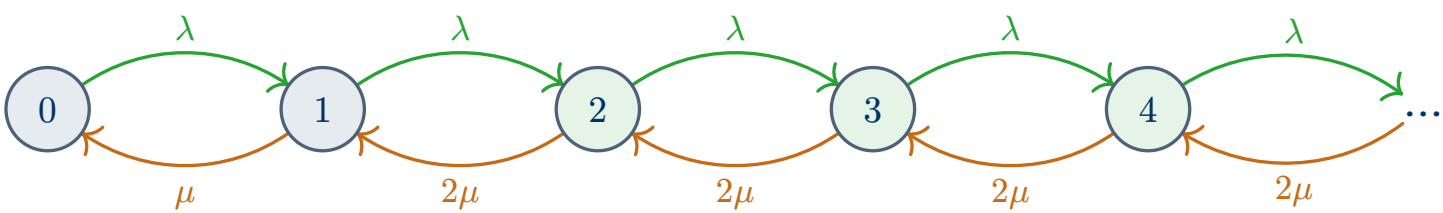
How much can pooling reduce these delays? The common queue is an **M/M/2** system.

How many processors can finish service?

For the pooled queue, let $L(t)$ count all requests present. With c identical processors, the total completion rate depends on how many are busy:

$$\mu_n = \min(n, c)\mu.$$

Two processors; transition rates per minute



State n counts all requests present, including those in service.

Rate per processor: μ. Total completion rate: $\min(n, 2)\mu$.

For two processors, state 1 has one active service clock; state 2 or higher has two. A third waiting request **does not create a third service clock**.

This is an **M/M/c** queue. Stability requires $\lambda < c\mu$.

What is the stationary distribution?

Write $a = \frac{\lambda}{\mu}$ for offered work in processor units and $\rho = \frac{\lambda}{c\mu} = \frac{a}{c}$. Here $a = 1.6$ and $\rho = 0.8$.

For $c = 2$, the birth-and-death recursion gives

$$w_0 = 1, \quad w_1 = a = 1.6, \quad w_2 = \frac{a^2}{2} = 1.28, \quad w_{2+k} = 1.28\rho^k.$$

Normalize the weights:

$$\pi_0 = \left(1 + 1.6 + \frac{1.28}{1 - 0.8}\right)^{-1} = \frac{1}{9}.$$

Below two requests, another request activates another processor. From state 2 onward, capacity is fixed at 2μ, so the tail has geometric ratio ρ.

The same recursion works for general c; we need its tail to calculate waiting.

When must an arrival wait?

In the pooled queue, an arrival waits exactly when it finds **all c processors busy**. Define the **waiting probability** and use PASTA:

$$p_{\text{wait}} = \mathbb{P}\big(W_q > 0\big) = \mathbb{P}(N \geq c) = \sum_{n=c}^{\infty} \pi_n.$$

Erlang C formula **Theorem 5.5**

For a stationary FCFS M/M/c queue with unlimited buffers, let $a = \frac{\lambda}{\mu}$ and $\rho = \frac{a}{c} < 1$. Then

$$\pi_0 = \left(\sum_{n=0}^{c-1} \frac{a^n}{n!} + \frac{a^c}{c!(1-\rho)} \right)^{-1}, \quad p_{\text{wait}} = \pi_0 \frac{a^c}{c!(1-\rho)}.$$

For two gateway processors, $p_{\text{wait}} = \left(\frac{1}{9}\right) \times \frac{1.28}{1-0.8} = 0.7111$.

Under independent random routing, an arrival waits whenever **its assigned processor** is busy, with probability **80%**, even if the other processor is idle.

Proof of Theorem 5.5:

1. Balance. The recursion $\lambda\pi_n = \min(n+1, c)\mu\pi_{n+1}$ for $n \geq 0$ gives

$$\pi_n = \pi_0 \frac{a^n}{n!} \quad (0 \leq n < c), \quad \pi_{c+k} = \pi_0 \frac{a^c}{c!}\rho^k \quad (k \geq 0).$$

2. Normalize. Since $\rho < 1$, $\sum_{k=0}^{\infty} \rho^k = \frac{1}{1-\rho}$, so

$$1 = \sum_{n=0}^{\infty} \pi_n = \pi_0 \left(\sum_{n=0}^{c-1} \frac{a^n}{n!} + \frac{a^c}{c!(1-\rho)} \right).$$

3. Apply PASTA. An arrival waits exactly when all c processors are busy. Thus

$$p_{\text{wait}} = \mathbb{P}(N \geq c) = \sum_{k=0}^{\infty} \pi_{c+k} = \pi_0 \frac{a^c}{c!} \sum_{k=0}^{\infty} \rho^k = \pi_0 \frac{a^c}{c!(1-\rho)}.$$

How long does a waiting request wait?

If an arrival finds $N = c + k$, it needs $k + 1$ completions before starting service. Until then, all c processors remain busy and completions occur at total rate $c\mu$.

Conditional on having to wait, the stationary tail gives

$$\mathbb{P}(N = c + k \mid N \geq c) = (1 - \rho)\rho^k, \quad k \geq 0.$$

By Lemma 5.1, the mean wait among those who wait is

$$\mathbb{E}\left[W_q \mid W_q > 0\right] = \frac{1}{1 - \rho} \cdot \frac{1}{c\mu} = \frac{1}{c\mu - \lambda}.$$

By the **law of total expectation**, condition on whether the arrival waits:

$$\begin{aligned} w_q = \mathbb{E}\left[W_q\right] &= (1 - p_{\text{wait}})\mathbb{E}\left[W_q \mid W_q = 0\right] + p_{\text{wait}}\mathbb{E}\left[W_q \mid W_q > 0\right] \\ &= (1 - p_{\text{wait}}) \times 0 + p_{\text{wait}} \frac{1}{c\mu - \lambda} = \frac{p_{\text{wait}}}{c\mu - \lambda}. \end{aligned}$$

Adding the request's own mean service time gives $w = w_q + \frac{1}{\mu}$.

How much delay does pooling remove?

For the pooled gateway, $w_q = \frac{0.7111}{30-24}$ minutes, or **7.11 seconds**.

Measure	Dedicated queues	Pooled queue
Total arrival rate	24/min	24/min
Total service capacity	30/min	30/min
Average processor utilization	80%	80%
Probability a request waits	80%	71.11%
Mean waiting time w_q	16 s	7.11 s
Mean system time w	20 s	11.11 s



Observation

The average amount of service work is unchanged, but waiting caused by inaccessible idle capacity is removed.

What is the waiting-time distribution?

M/M/c waiting-time distribution — Theorem 5.6

For a stationary FCFS M/M/c queue with unlimited buffers and $\rho = \frac{\lambda}{c\mu} < 1$, let p_{wait} be the Erlang C waiting probability (Theorem 5.5). For $t \geq 0$,

$$\mathbb{P}(W_q = 0) = 1 - p_{\text{wait}}, \quad \mathbb{P}(W_q > t) = p_{\text{wait}} e^{-(c\mu - \lambda)t}.$$

Proof of Theorem 5.6:

Given $W_q > 0$, an arrival needs $G = N - c + 1$ completions before service starts. PASTA and $\pi_{c+k} = \pi_c \rho^k$ give

$$\mathbb{P}(G = g \mid W_q > 0) = \frac{\pi_c \rho^{g-1}}{\frac{\pi_c}{1-\rho}} = (1-\rho)\rho^{g-1}, \quad g \geq 1.$$

Memorylessness gives independent $\text{Exp}(c\mu)$ stages, independent of the geometric count G. By Lemma 5.1, their sum is exponential with rate $(1-\rho)c\mu = c\mu - \lambda$. Weighting this conditional tail by p_{wait} gives the result.

? How many processors meet a waiting target?

With a pooled FCFS queue, demand at 24/min and service at 15/min per processor, find the smallest c such that **at least 95% of requests start service within 10 seconds**.

For each candidate c, first check **stability**: $c\mu > \lambda$, or $15c > 24$. Then compute p_{wait} using **Erlang C** (Theorem 5.5), with $a = 1.6$ and $\rho = \frac{1.6}{c}$.

By Theorem 5.6, using **10 seconds $=$ $\frac{1}{6}$ minute**, the target is

$$\mathbb{P}\left(W_q > 10 \text{ s}\right) = p_{\text{wait}} e^{-\frac{15c-24}{6}} \leq 0.05.$$

Three processors are the smallest choice

One processor is unstable. Two miss the target: **26.16%** wait over 10 seconds. For $c = 3$, $p_{\text{wait}} \approx 0.2738$ and Theorem 5.6 gives $\mathbb{P}\left(W_q > 10 \text{ s}\right) \approx 0.2738 e^{-3.5} \approx 0.83\% < 5\%$.

If queues stay dedicated, can routing help?

Load balancing routes arriving requests among processors to reduce workload imbalance.

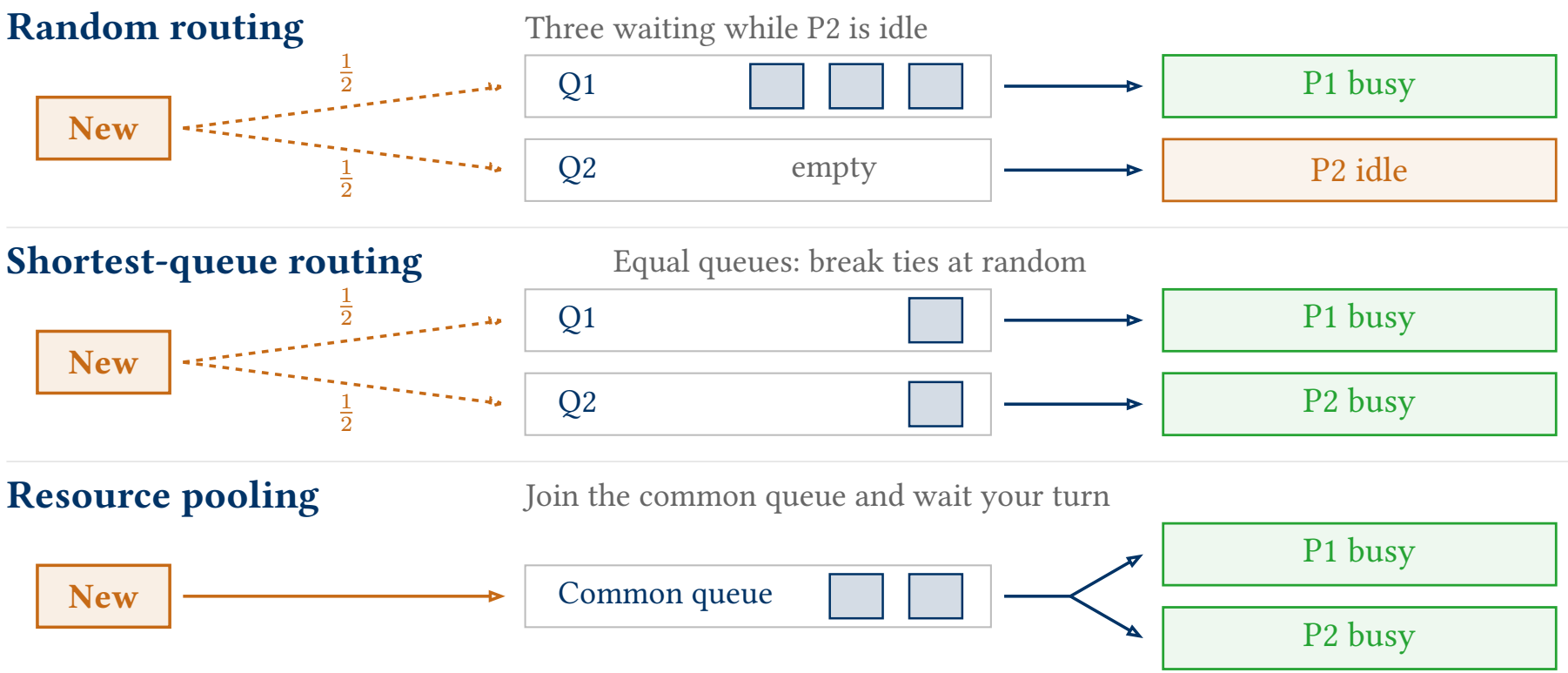
Routing rule	Decision at arrival
Random assignment	Independently choose either processor with probability $\frac{1}{2}$.
Round robin	Alternate between processors. This balances assignment counts, not actual workloads.
Join the shortest queue (JSQ)	Choose the processor with the fewest requests present, including the one in service.

With **dedicated queues**, a request stays with its assigned processor. With **resource pooling**, the first waiting request takes the next available processor.

Round robin changes the arrival processes, and shortest-queue routing couples the queues. Our **two independent M/M/1 formulas** apply to the independent random-routing benchmark.

How is the next request assigned?

Each design has four requests present. The orange box is a new request.



What ranking should we expect?

Keep the same demand and two identical processors, with negligible routing and sharing overhead.

> **? A prediction to test**
>
> For our example, we expect **mean waiting time** to increase in the order **pooling, JSQ, round robin, random routing**.

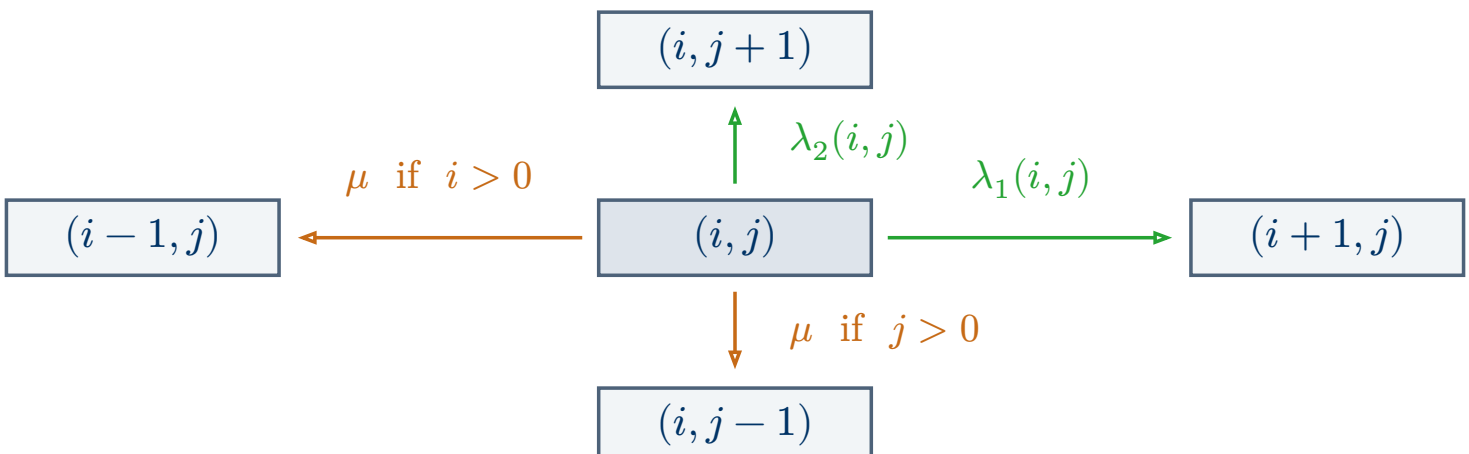
- **Pooling:** any idle processor can serve waiting work.
- **JSQ:** sends new work to the shorter queue, but cannot move requests already assigned.
- **Round robin:** balances assignment counts, but ignores current queues and variable service times.
- **Random routing:** can send a run of arrivals to a busy processor while the other is idle.

We will check this prediction numerically and see how close JSQ gets to pooling.

How can we analyze two-server JSQ?

Keep Poisson arrivals at $\lambda = 24$/min and independent exponential services at $\mu = 15$/min per processor. Both dedicated queues are FCFS, with **no switching after assignment**.

The state $(L_1(t), L_2(t)) = (i, j)$ counts requests at each processor, **including service**.



Arrivals choose the shorter queue: $(\lambda_1, \lambda_2) = (\lambda, 0)$ when $i < j$, $(0, \lambda)$ when $i > j$, and $\left(\frac{\lambda}{2}, \frac{\lambda}{2}\right)$ on a tie.

How close does JSQ get to pooling?

Keep $\lambda = 24$/min, two processors at $\mu = 15$/min each, and unlimited buffers. Every design has **80% average processor utilization** and **4 s mean service time**.

Measure	Random	RR	JSQ	Pooled
Mean waiting time w_q	16 s	11.38 s	7.82 s	7.11 s
Mean system time w	20 s	15.38 s	11.82 s	11.11 s
Time with an idle processor while requests wait	25.6%	17.54%	6.1%	0%

Random routing and RR can leave **idle capacity inaccessible to waiting work**. JSQ reduces this mismatch, although existing assignments can still become unbalanced.

Here, JSQ recovers **about 92% of pooling's mean-wait reduction** relative to random routing. Its mean wait is only **0.71 s above pooling**.

Which design should we choose?



Pool when feasible, use informed routing otherwise

1. **Pooling gives the smallest mean wait in our comparison.** It keeps processors busy whenever requests are waiting.
2. **Random routing and RR can waste opportunities:** balanced assignment counts do not balance realized service work. Waiting requests cannot use an idle processor elsewhere.
3. **Use current queue information:** JSQ directs new work toward the less occupied processor. It improves on random routing and RR in our comparison.
4. **JSQ comes close here:** mean waiting is **7.82 s**, versus **7.11 s** with pooling, recovering about **92%** of the pooling benefit over random routing.
5. **State the metric and assumptions:** all four designs have **80% average utilization**. The mean-delay ordering does not imply an ordering of every waiting-time probability.

Pause and recall

Resource pooling and load balancing

1. What changes when dedicated queues are replaced by a pooled queue?
2. Why must the two-server JSQ state record both queue populations?
3. Why does pooling reduce delay here while average utilization stays at 80%?
4. Does the smallest mean wait also imply the smallest probability of exceeding every waiting threshold?

Pause here. Try each question before turning to the answers.

Answers and reasoning

Resource pooling and load balancing

1. Each waiting request can use either processor. The number and speeds of processors stay fixed.
2. The same total can have different departure rates: $(2, 0)$ gives μ, while $(1, 1)$ gives 2μ.
3. It removes waiting caused by an inaccessible idle processor. Mean work per unit time stays the same.
4. No. A comparison of means does not determine the entire waiting-time distributions.

Next: keep mean service capacity fixed and change service variability.

5.5. Beyond Exponential Service

Same average, different experience

Recall the canteen: six students need one minute of service each. Compare six arrivals spread evenly over six minutes with six arriving together.

Arrival pattern	Individual waits (minutes)	Mean wait
One per minute	$0, 0, 0, 0, 0, 0$	0 minutes
All at once	$0, 1, 2, 3, 4, 5$	2.5 minutes

Average demand and average capacity alone did not determine waiting. Work accumulated during the burst; earlier wasted idle capacity could not be recovered for later use.

? Question

Now keep the gateway's **Poisson arrivals fixed**. Can variability in **service durations** change waiting even when mean service time stays at four seconds?

Three service models with the same mean

Compare alternative service-time models for **one** gateway processor. Each has mean service time four seconds and receives 12 requests/minute.

Model	Service duration S	$\mathbb{E}[S]$
Deterministic	Exactly 4 seconds	4 s
Exponential	Exponential with mean 4 seconds	4 s
Occasional long task	2 seconds with probability $\frac{16}{17}$; 36 seconds with probability $\frac{1}{17}$	4 s

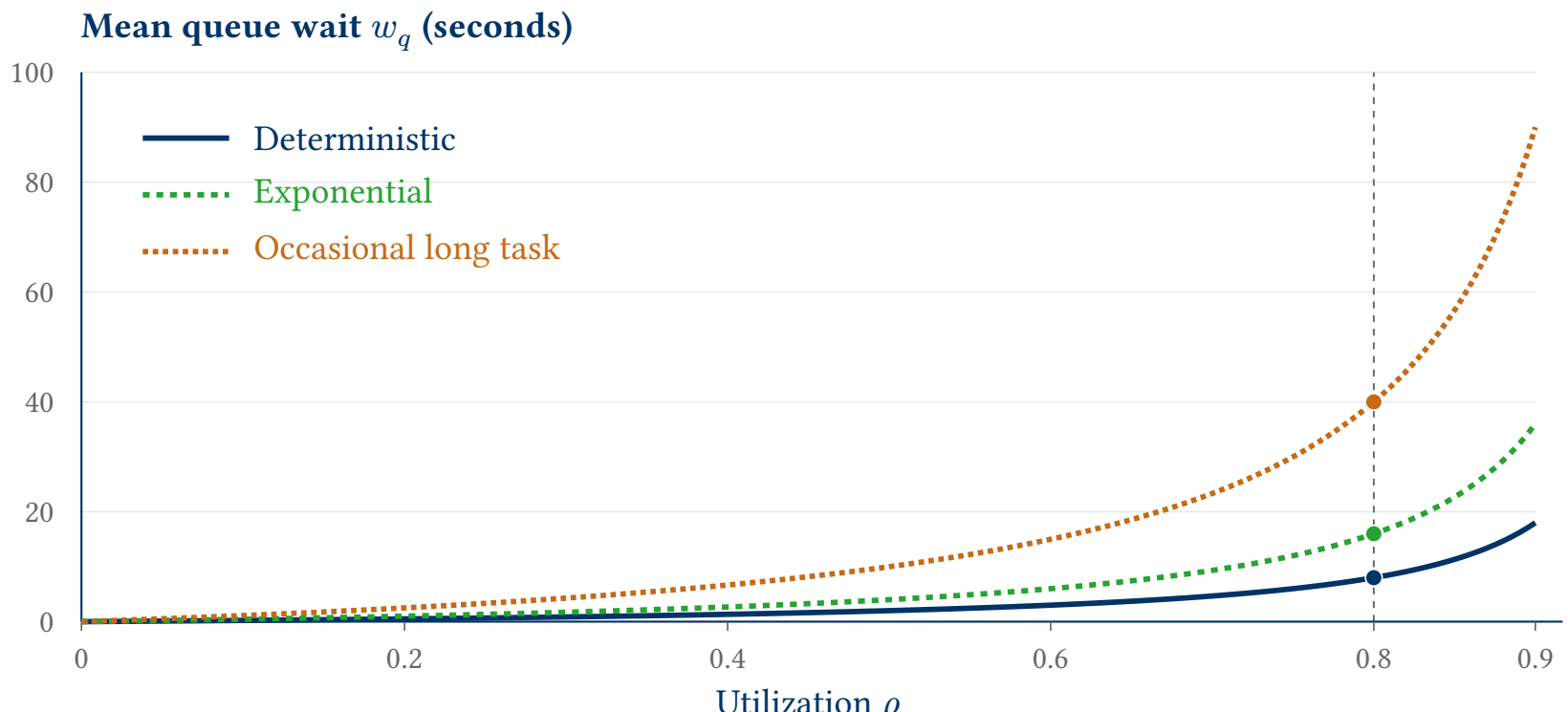
Mean waiting times — Poll 5.1

Predict the ordering of mean waiting times. All three have the same offered load, $\rho = 0.8$.

A). All three are equal.
B). Deterministic $<$ Exponential $<$ Occasional long task.
C). Occasional long task $<$ Deterministic $<$ Exponential.
D). Deterministic $<$ Occasional long task $<$ Exponential.

Mean waiting time as utilization rises

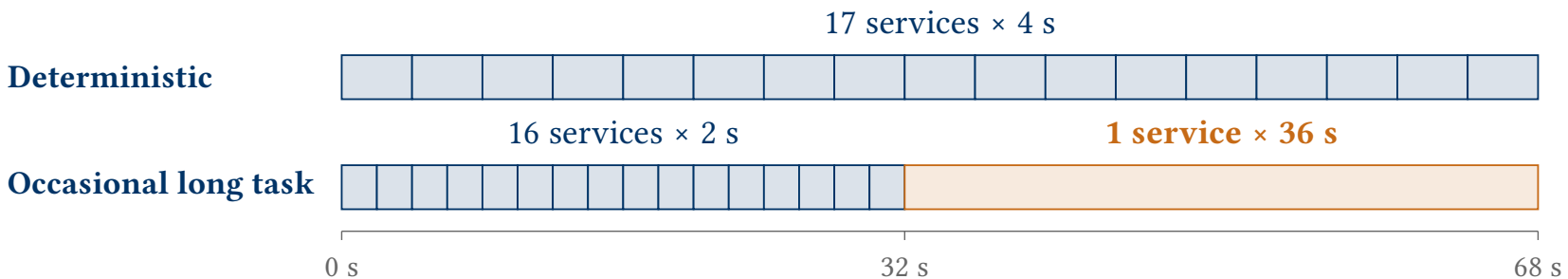
Keep mean service at four seconds and vary the Poisson arrival rate.



At $\rho = 0.8$, the mean queue waits are **8 s**, **16 s**, and **40 s**, respectively.

Why a rare long service raises the wait

Compare 17 services using the expected mix. Both rows take 68 seconds of service.



Only **1 in 17** jobs is long, but it takes **36 of 68 busy seconds**. Poisson arrivals that find the processor busy see jobs in proportion to their share of busy time, with half the job's duration left on average.

$$\frac{32}{68} \times 1 \text{ s} + \frac{36}{68} \times 18 \text{ s} = 10 \text{ s}.$$

The mean time left is **2 s** for deterministic service and **4 s** for exponential service.

At higher utilization, more requests arrive during a long service and the backlog takes longer to clear.

The M/G/1 model

Retain Poisson arrivals at rate λ and one FCFS processor that works whenever requests are present. Let successive service durations be independent and identically distributed copies of S, independent of arrivals.

The **G** allows a general service-time distribution. We assume

$$0 < \mathbb{E}[S] < \infty, \quad \rho = \lambda \mathbb{E}[S] < 1, \quad \mathbb{E}[S^2] < \infty.$$

The buffer is unlimited; no request is rejected or abandons. The model operates in stationarity.

The count $L(t)$ alone is generally no longer Markov: **the age of a service can change its remaining duration**. **PASTA and Little's law still apply.**

A service in progress is not a fresh service

Recall the bus example: gaps of 10 and 30 minutes occurred equally often, but a random passenger encountered the long gap three times as often.

Likewise, a random clock time during a busy period is more likely to lie inside a long service. It samples **service time**, not a uniformly chosen completed request.

For example, if service durations of 2 and 6 seconds occur equally often, a busy-time observation lies in them with probabilities $\frac{1}{4}$ and $\frac{3}{4}$.

$$\mathbb{E}[\text{residual service} \mid \text{busy}] = \frac{1}{4} \times 1 + \frac{3}{4} \times 3 = 2.5 \ \text{ s.}$$

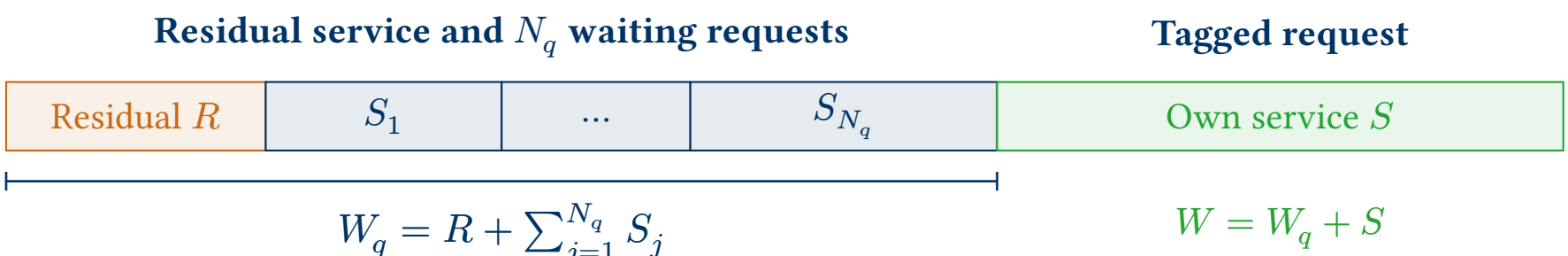
This exceeds half the mean service duration, $\frac{4}{2} = 2$ seconds. Long services are encountered more often and leave more work behind.

What work is ahead of an arrival?

? Question

What is the formula for the mean queue wait $w_q = \mathbb{E}[W_q]$ in a stationary M/G/1 queue?

Under FCFS, an arrival waits for the work already ahead. Let R be the service time left at arrival, zero if idle. Let N_q count the jobs already waiting and S_j be the service time of waiting job j.



To find the mean wait, start with the mean residual service $\mathbb{E}[R]$.

Why arrivals see the clock-time average

Let $R(t)$ be the service time left at clock time t, zero if idle. The request arriving at T_i sees $R(T_i^-)$, the value just before it arrives.

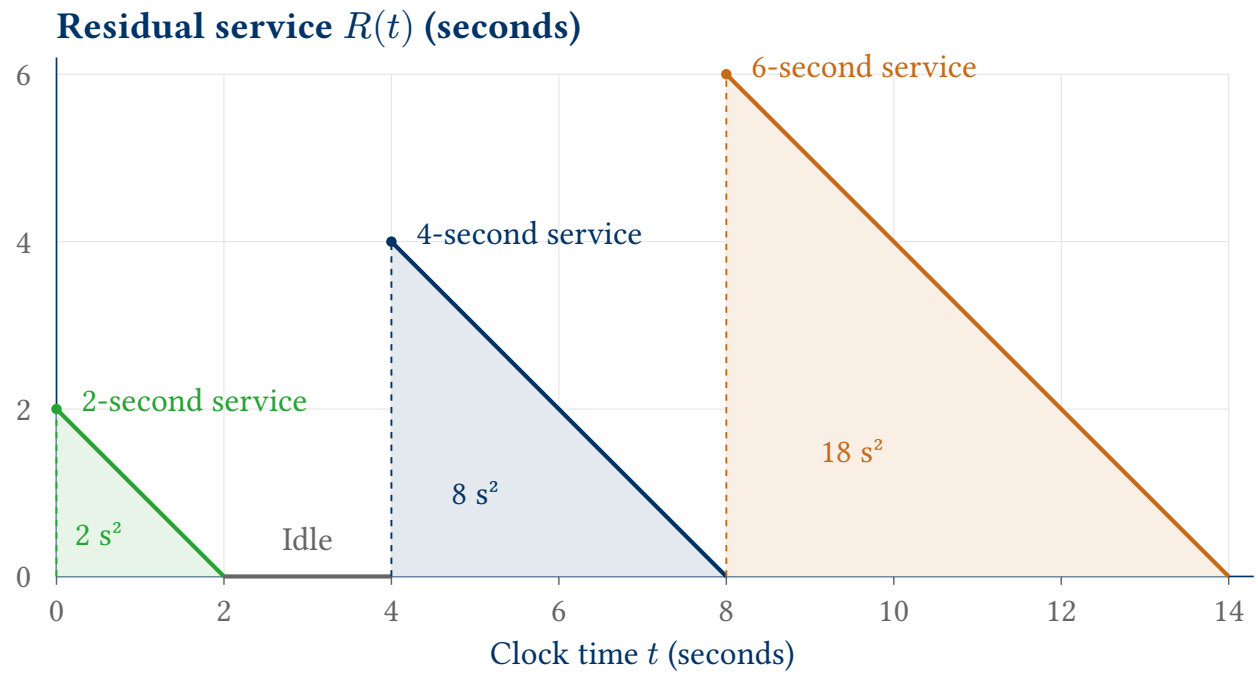
Poisson arrivals occur at the same rate λ whatever the current residual. By the PASTA theorem, arrivals see each range of residual times in the same proportion that the processor spends in that range. For example, if $R(t) > 4$ seconds for 10% of clock time, then 10% of arrivals see more than four seconds left in the long run.

$$\underbrace{\lim_{m\to\infty} \frac{1}{m} \sum_{i=1}^{m} R(T_i^-)}_{\text{average at arrivals}} = \mathbb{E}[R] = \underbrace{\lim_{H\to\infty} \frac{1}{H} \int_0^H R(t)\,\mathrm{d}t}_{\text{average over clock time}}.$$

We can therefore find the arrival mean by measuring the area under $R(t)$ and dividing by elapsed time.

Each service contributes a triangle

A service of length s contributes a triangle of area $\frac{s^2}{2}$. Doubling its duration **quadruples** the area.



The total area is $2 + 8 + 18 = 28$ s². The example averages $\frac{28}{12} \approx 2.33$ s over the 12 busy seconds.

Mean residual service when busy and over all time

By PASTA, an arrival finds the processor busy with probability $\rho = \lambda \mathbb{E}[S]$. Since $R = 0$ if idle,

$$\mathbb{E}[R] = \rho \mathbb{E}[R \mid \text{busy}].$$

To find the busy-time mean, let $B(H)$ be the total time the processor is busy during $[0, H]$. Divide the area under $R(t)$ by $B(H)$. The clock-time average divides the same area by H.

Each complete service adds $\frac{S_i^2}{2}$ to the area and S_i to busy time. This weights its mean residual $\frac{S_i}{2}$ by its duration. The law of large numbers gives the limiting ratio.

$$\mathbb{E}[R \mid \text{busy}] = \lim_{H \to \infty} \frac{\int_0^H R(t)\,\mathrm{d}t}{B(H)} = \lim_{n \to \infty} \frac{\sum_{i=1}^n S_i^2}{2\sum_{i=1}^n S_i} = \frac{\mathbb{E}[S^2]}{2\mathbb{E}[S]}.$$

Multiplying by ρ gives

$$\mathbb{E}[R] = \rho \mathbb{E}[R \mid \text{busy}] = \frac{\lambda \mathbb{E}[S^2]}{2}.$$

Little's law gives the Pollaczek–Khinchine formula

By PASTA, $\mathbb{E}[N_q] = \ell_q$. Under FCFS, the N_q queued jobs need $N_q \mathbb{E}[S]$ service on average.

Take expectations of the work ahead and use Little's law, $\ell_q = \lambda w_q$.

$$w_q = \mathbb{E}[R] + \ell_q \mathbb{E}[S] = \mathbb{E}[R] + \rho w_q.$$

Solve for w_q and substitute $\mathbb{E}[R]$.

Pollaczek–Khinchine mean-waiting formula **Theorem 5.7**

For the stationary FCFS M/G/1 model,

$$w_q = \frac{\lambda \mathbb{E}[S^2]}{2(1-\rho)}, \quad w = w_q + \mathbb{E}[S].$$

Separate the roles of variability and load

Define the **squared coefficient of variation** of service time:

$$c_s^2 = \frac{\mathrm{Var}(S)}{(\mathbb{E}[S])^2}, \quad \mathbb{E}[S^2] = (1 + c_s^2)(\mathbb{E}[S])^2.$$

The mean-waiting formula becomes

$$w_q = \frac{\rho}{1-\rho} \times \frac{1 + c_s^2}{2} \times \mathbb{E}[S].$$

At a fixed mean service time, higher variance means more waiting. As utilization approaches one, the factor $\frac{\rho}{1-\rho}$ magnifies the effect.

Deterministic service has $c_s^2 = 0$; exponential service has $c_s^2 = 1$. The units are carried by $\mathbb{E}[S]$; the other factors are dimensionless.

Four-second services can produce very different waits

Use $\lambda = 12/\text{min}$, $\mathbb{E}[S] = 4$ seconds and $\rho = 0.8$.

Service model	c_s^2	Mean wait w_q	Mean total w
Exactly 4 seconds	0	8 s	12 s
Exponential	1	16 s	20 s
2 or 36 seconds	4	40 s	44 s

For the occasional-long-task model,

$$\mathbb{E}[S^2] = \frac{16 \times 2^2 + 36^2}{17} = 80 \text{ s}^2, \quad c_s^2 = \frac{80 - 16}{16} = 4.$$

Under each service model, the processor completes 12 requests/minute and is busy 80% of the time. Those identical averages hide very different customer experiences.

Fixed service times can cut waiting without more capacity

Hold Poisson demand at 12 requests per minute.

Service option	Mean service	w_q	w
Original exponential service (15/min)	4 s	16 s	20 s
Deterministic service	4 s	8 s	12 s
Faster exponential service (18/min)	3.33 s	6.67 s	10 s

Fixed service times eliminate service variability. A faster processor reduces both mean service time and load. Either can reduce waiting; their costs and feasibility must be assessed separately.

For a **mean system-time target of 12 seconds**, both changes meet the modeled target. The table alone does not say which investment costs less.

What does deterministic service leave unresolved?

With service fixed at four seconds, M/D/1 still has a mean wait of **eight seconds**. Poisson arrivals can bunch together even when service is perfectly regular.

If arrivals were instead exactly five seconds apart and service exactly four seconds, an initially empty system would have **no waiting**. That is a different arrival model.

The P–K formula here is **exact for Poisson arrivals**. It is not a general formula for arbitrary arrival patterns.

A mean-waiting formula also does not provide a completion-time percentile. In general, we cannot reuse the M/M/1 exponential tail after changing the service distribution.

Kingman's approximation for general interarrival times

In a stationary FCFS GI/GI/1 queue, the interarrival times A and service times S form independent i.i.d. sequences with finite variances. Let $\rho = \frac{\mathbb{E}[S]}{\mathbb{E}[A]} < 1$.

The squared coefficient of variation of the arrival gaps is $c_a^2 = \frac{\mathrm{Var}(A)}{(\mathbb{E}[A])^2}$.

Kingman's heavy-traffic approximation **Theorem 5.8**

$$w_q \approx \frac{\rho}{1-\rho} \times \frac{c_a^2 + c_s^2}{2} \times \mathbb{E}[S].$$

For Poisson arrivals, $c_a^2 = 1$, and the expression reduces to the exact P–K formula.

With 4-second services and equally likely gaps of 3 or 7 seconds, $c_s^2 = 0$, $c_a^2 = \frac{4}{25} = 0.16$, and $\rho = 0.8$.

$$w_q \approx \frac{0.8}{0.2} \times \frac{0.16}{2} \times 4 \ \text{ s} = 1.28 \ \text{ s}.$$

Principle to remember

Variability and spare capacity act together

Long services occupy more of the clock and leave more residual work for arrivals. Their contribution depends on the **second moment**, not only the mean.

Reducing service variability can reduce waiting without increasing mean capacity.

Small capacity slack magnifies variability's effect. Stability alone does not even ensure a finite mean wait if the service-time second moment is infinite.

Compare the variability of demand and service, as well as their averages.

Pause and recall

Why equal averages produce different waiting

---

1. Why does a service of length s contribute $\frac{s^2}{2}$ to residual area?
2. Does $\mathbb{E}[R] = \lambda \frac{\mathbb{E}[S^2]}{2}$ include idle time?
3. For the gateway, why does deterministic service halve mean waiting but not eliminate it?
4. After changing service variability, can we still quote the M/M/1 95th percentile?

Pause here. Try each question before turning to the answers.

Answers and reasoning

Why equal averages produce different waiting

---

1. The residual path is a triangle with base s and height s.
2. Yes. $R = 0$ while idle; conditional on busy, divide this mean by ρ.
3. Its second moment halves relative to exponential service with the same mean; Poisson arrival variability remains.
4. No. P–K gives a mean, and the total-time distribution generally changes.

6. Stochastic Inventory Models

6.1. Stocking for a Single Period: The Newsvendor Problem

The Pop Mart decision revisited

In Chapter 1's Pop Mart example, we compared stocks of 10 and 15 units before uncertain demand. Both demand worlds had mean 10.

World	Demand distribution
Stable	$P(D=8)=0.25$, $P(D=10)=0.50$, $P(D=12)=0.25$
Hype	$P(D=5)=0.75$, $P(D=25)=0.25$

? What if we can choose any stock quantity?

Choose an integer $Q \geq 0$ before demand occurs. However, each leftover unit costs 1 dollar. Each unit of unmet demand costs 4 dollars.

Would a sixteenth unit help?

In the hype world, demand is usually only five. Suppose we already plan to stock 15 units.

? Predict before calculating

Would stocking a sixteenth unit increase or reduce expected loss? Write what happens to that extra unit under each demand outcome.

Demand	Probability	Consequence of the extra unit
5	0.75	Does it remain unsold or meet demand?
25	0.25	Does it remain unsold or meet demand?

Compare stocks 15 and 16 under the **same demand outcome**.

Marginal consequences and retail economics

With probability 0.75, the extra unit remains and costs 1 dollar. With probability 0.25, it prevents a shortage and saves 4 dollars.

$$\text{Change in expected loss} = 0.75 \times 1 - 0.25 \times 4 = -0.25.$$

The sixteenth unit **reduces expected loss by 25 cents**.

Where might these costs come from? Suppose a unit sells for 6, costs 2 to buy, and has salvage value 1 if unsold.

Consequence	Incremental cost	Amount
One excess unit	Purchase cost minus salvage	$c_o = 2 - 1 = 1$
One unit short	Missed contribution margin	$c_u = 6 - 2 = 4$

These costs already account for purchase cost. We do not add it again.

Why “newsvendor”?

A **newsvendor** sells newspapers. Each morning, the vendor chooses how many copies to buy **before knowing demand**.

- **Too many papers:** unsold copies have little or no value after the day ends.
- **Too few papers:** customers leave and potential sales are lost.

One stock decision, uncertain demand, and a short selling window: the same trade-off as in the Pop Mart example.



Jacob Epstein, *Buying a newspaper* (1902).
The Spirit of the Ghetto. Wikimedia Commons.

The newsvendor model

Choose stock Q before observing nonnegative demand D, whose distribution is known and has finite mean.

- $c_o > 0$: incremental cost of one excess unit.
- $c_u > 0$: incremental cost of being one unit short.

Newsvendor loss — Definition 6.1

The **realized loss** after demand is observed is

$$\ell(Q, D) = c_o(Q - D)^+ + c_u(D - Q)^+, \quad z^+ = \max\{z, 0\}.$$

The **expected loss** before demand is observed is

$$L(Q) = \mathbb{E}[\ell(Q, D)].$$

Choose a feasible Q to minimize $L(Q)$. Pop Mart uses integer quantities; the juice example will allow divisible quantities.

The value of one more unit

For this discrete model, both stock Q and demand D are nonnegative integers. Compare Q and $Q+1$ units under the **same realized demand**.

Demand event	Effect of the extra unit	Change in loss
$D \le Q$	Q units already meet demand; the extra unit is left over	$+c_o$
$D \ge Q+1$	The extra unit satisfies one unit of previously unmet demand	$-c_u$

$$\ell(Q+1, D) - \ell(Q, D) = \begin{cases} c_o & \text{if } D \le Q \\ -c_u & \text{if } D \ge Q+1 \end{cases}.$$

At $D = Q$, the original stock exactly meets demand. The added unit is left over, so **equality belongs in the first case**.

From realized loss to expected loss

Write $F_D(Q) = \mathbb{P}(D \le Q)$. Because demand is integer-valued, the two cases cover every outcome:

$$\mathbb{P}(D \ge Q+1) = 1 - F_D(Q).$$

The change in expected loss is the **probability-weighted average** of the two realized changes:

$$\begin{aligned} L(Q+1) - L(Q) &= \mathbb{E}[\ell(Q+1, D) - \ell(Q, D)] \\ &= (+c_o)\mathbb{P}(D \le Q) + (-c_u)\mathbb{P}(D \ge Q+1) \\ &= c_o F_D(Q) - c_u(1 - F_D(Q)) \\ &= (c_o + c_u)F_D(Q) - c_u. \end{aligned}$$

This increment does not decrease as Q grows. Expected loss falls until the increment becomes nonnegative.

The critical fractile

An extra unit reduces expected loss whenever

$$F_D(Q) < \frac{c_u}{c_o + c_u}.$$

Therefore the **smallest optimal integer quantity** is

$$Q^* = \min\{q \in \mathbb{Z}_{\geq 0} : F_D(q) \geq \frac{c_u}{c_o + c_u}\}.$$

The **critical fractile** $\frac{c_u}{c_o+c_u}$ selects a demand quantile.

If $F_D(Q) = \frac{c_u}{c_o+c_u}$, adding one unit changes expected loss by zero. Both Q and $Q+1$ are optimal; the formula selects the smallest optimum.

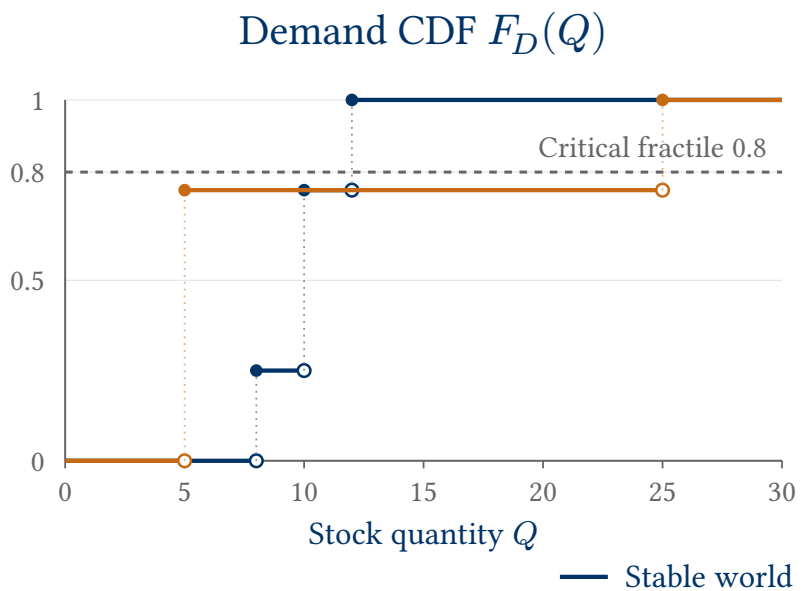
Solving the two Pop Mart worlds

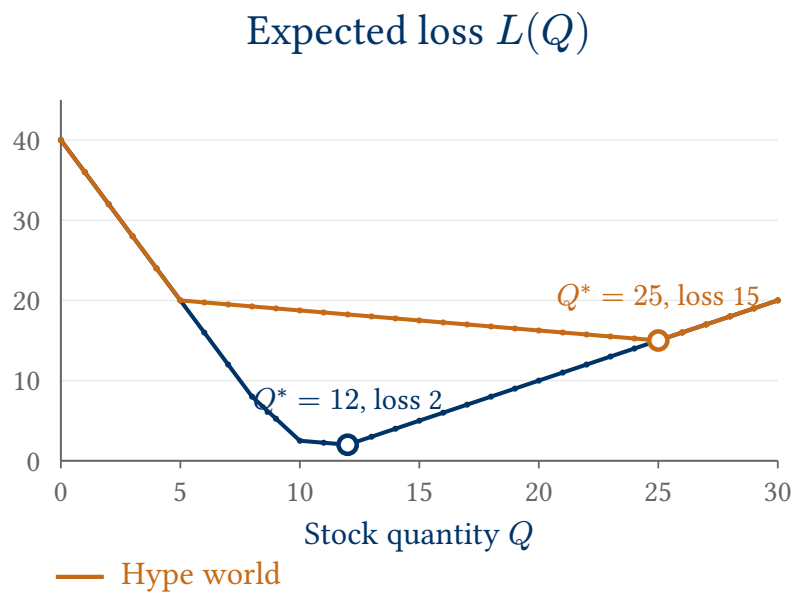
The critical fractile is $\frac{4}{1+4} = 0.8$.

World	CDF at demand values	Smallest Q^*	$L(Q^*)$
Stable	$F(8) = 0.25$, $F(10) = 0.75$, $F(12) = 1$	12	2
Hype	$F(5) = 0.75$, $F(25) = 1$	25	15

In the hype world, every addition from $Q = 5$ through $Q = 24$ changes expected loss by $0.75 - 4(0.25) = -0.25$. Stocking 25 is therefore understandable even though demand is usually five.

A demand quantile minimizes expected loss





Continuous demand: stocking fresh juice

A kiosk chooses $Q \geq 0$ litres before the day's demand. In this illustrative model, $D \sim \text{Uniform}(0, 20)$ litres and stock is divisible. Mean demand is still 10.

Keep the same retail economics as the Pop Mart example:

Selling price	Purchase cost	End-of-day salvage
6 dollars/litre	2 dollars/litre	1 dollar/unsold litre

For realized demand D, the day's profit is sales revenue plus salvage receipts minus purchase cost:

$$6 \min\{Q, D\} + (Q - D)^+ - 2Q.$$

Predict: should the best stock be below, at, or above the mean of 10 litres?

From sales profit to newsvendor loss

Demand = sales + unmet demand, while **stock = sales + leftovers**. Therefore

$$\min\{Q, D\} = D - (D - Q)^+,$$
$$Q = \min\{Q, D\} + (Q - D)^+ = D + (Q - D)^+ - (D - Q)^+.$$

Substitute these identities into the profit expression and collect terms:

$$\begin{aligned}
\text{Profit} &= 6\min\{Q, D\} + (Q - D)^+ - 2Q \\
&= 6[D - (D - Q)^+] + (Q - D)^+ - 2[D + (Q - D)^+ - (D - Q)^+] \\
&= 4D - (2 - 1)(Q - D)^+ - (6 - 2)(D - Q)^+ \\
&= 4D - [(Q - D)^+ + 4(D - Q)^+] = 4D - \ell(Q, D).
\end{aligned}$$

The loss in Definition 6.1 has **overage cost** $c_o = 2 - 1 = 1$ and **underage cost** $c_u = 6 - 2 = 4$. The benchmark $4D$ is profit if stock matches demand exactly.

How expected newsvendor loss changes

For the juice example, Definition 6.1 gives

$$L(Q) = \mathbb{E}[(Q - D)^+] + 4\mathbb{E}[(D - Q)^+].$$

Hold demand D fixed and consider a **small increase in stock** Q:

Demand	Effect of increasing stock	Loss slope
$D < Q$	Demand is already covered; leftovers increase.	$+1$
$D > Q$	Extra stock reduces unmet demand.	-4

Continuous demand gives $\mathbb{P}(D = Q) = 0$, so the boundary has zero probability. Average the two slopes using their probabilities:

$$\begin{aligned} L'(Q) &= (+1)\mathbb{P}(D < Q) + (-4)\mathbb{P}(D > Q) \\ &= F_D(Q) - 4(1 - F_D(Q)). \end{aligned}$$

The continuous critical fractile

Define $\Pi(Q)$ as the **expected daily profit** when stocking Q litres:

$$\begin{aligned}\Pi(Q) &:= \mathbb{E}[6\min\{Q, D\} + (Q - D)^+ - 2Q] \\ &= 4\mathbb{E}[D] - L(Q) = 40 - L(Q).\end{aligned}$$

Here $L(Q)$ is the **expected newsvendor loss** in Definition 6.1. Since 40 does not depend on Q,

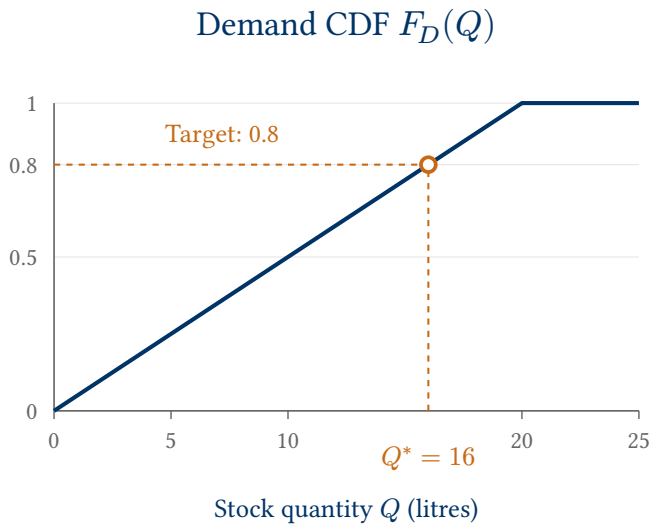
$$\begin{aligned}\Pi'(Q) &= 0 - L'(Q) \\ &= -[F_D(Q) - 4(1 - F_D(Q))] \\ &= 4(1 - F_D(Q)) - F_D(Q) = 4 - 5F_D(Q).\end{aligned}$$

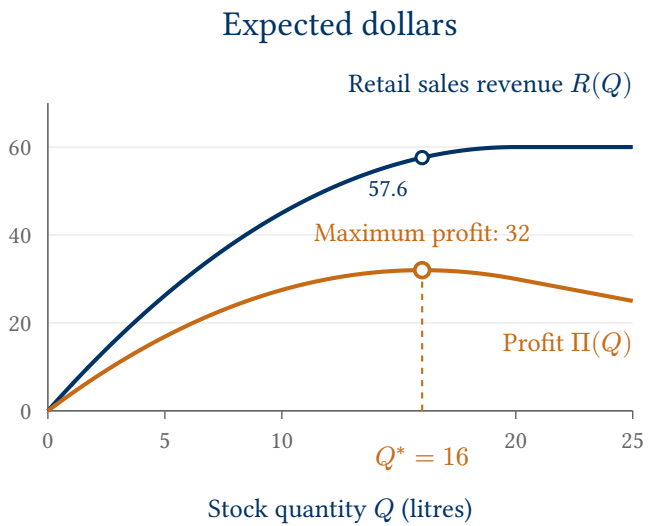
Marginal profit is **expected shortage cost saved** minus **expected extra leftover cost**.

With $D \sim \text{Uniform}(0, 20)$, we have $F_D(Q) = \frac{Q}{20}$ for $0 \le Q \le 20$. Marginal profit changes from positive to negative at

$$F_D(Q^*) = \frac{4}{5} = 0.8, \quad Q^* = 20 \times 0.8 = 16 \ \text{ litres.}$$

The quantile choice and its expected revenue





$R(Q) = 6\mathbb{E}[\min\{Q, D\}]$ is sales revenue; $4\mathbb{E}[D] = 40$ is profit if stock matches demand.

$$\Pi(Q) = \underbrace{R(Q)}_{\text{sales revenue}} + \underbrace{\mathbb{E}[(Q-D)^+]}_{\text{salvage receipts}} - \underbrace{2Q}_{\text{purchase cost}}$$

$$= 4\mathbb{E}[D] - \underbrace{[\mathbb{E}[(Q-D)^+] + 4\mathbb{E}[(D-Q)^+]]}_{L(Q)\text{: expected newsvendor loss}} = 40 - L(Q).$$

Pause and recall

Costs, quantiles, and profit

---

1. Why does maximizing expected profit in the juice example minimize newsvendor loss? Should we add purchase costs to $L(Q)$ again?
2. With new costs $c_o = 2$ and $c_u = 3$, find the best stock in the stable Pop Mart world and the uniform-demand juice model. Is the probability of serving all demand exactly 0.6 in both?
3. With the original juice prices, why does expected sales revenue rise from $Q = 16$ to $Q = 20$ while expected profit falls?

Pause here. Try each question before turning to the answers.

Answers and reasoning

Costs, quantiles, and profit

---

1. Expected profit is $40 - L(Q)$; 40 does not depend on stock. Purchase costs already enter c_o and c_u. Adding them again would double-count them.
2. The target is $\frac{3}{2+3} = 0.6$. Stable Pop Mart: choose $Q = 10$, with $F_D(10) = 0.75$. Juice: stock 12 litres, with $F_D(12) = 0.6$. A discrete CDF can jump past the target.
3. Four extra litres cost 8. Expected sales revenue increases by 2.4 and salvage receipts by 3.6: only 6 in total. Expected profit therefore falls by 2, from 32 to 30.

Translate profit into marginal costs, choose the demand quantile, and distinguish revenue from profit.

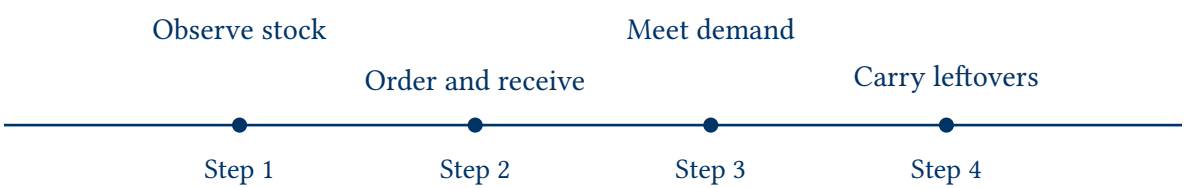
6.2. Replenishment with a Stock Target

The shop opens again tomorrow

In Section 6.1, we chose stock for one selling opportunity. Now unsold units carry over, and we can replenish each morning.

A shop stocking one item — Example 6.1

The item is *nonperishable*. The shop reviews stock **every morning**. Replenishment is **immediate and unlimited**. Unfilled requests leave: **lost sales**.



Daily demands are independent nonnegative integers with the same known distribution and finite mean. Each order is chosen **before today's demand**.

Notation for one day

The subscript t identifies the day.

Role	Symbol	Meaning
State	X_t	Stock on hand before the morning order
Decision	q_t	Order quantity, chosen before demand and received immediately
Exogenous randomness	D_t	Demand during the day, i.i.d. across days

Derived quantities:

- Y_t is the stock available after replenishment;
- U_t is the demand that could not be filled.

With **immediate delivery**, today's leftover stock becomes tomorrow's state X_{t+1}.

Replenishing to a stock target

We do not remove existing inventory. Given morning stock X_t, choose **available stock $Y_t \geq X_t$** and order the gap $q_t = Y_t - X_t \geq 0$. Today's sales and leftovers depend on Y_t and demand D_t.



Observation

With i.i.d. demand D_t and unchanged costs, the same Y_t gives the same expected daily performance. A natural policy aims for **constant stock S after replenishment**: the **stock target**.

For illustration, with target $S = 6$ and morning stock $X_t = 2$, order

$$q_t = 6 - 2 = 4 \quad \text{units.}$$

We will choose S by balancing the costs of leftovers and lost sales.

A constant target produces variable orders

Order-up-to, or base-stock, policy — Definition 6.2

Choose a target S for stock available after replenishment. Given morning stock X_t, the order quantity is

$$q_t = (S - X_t)^+, \quad z^+ = \max\{z, 0\}.$$

If morning stock exceeds S, order nothing and keep the excess.

Day	Morning X_t	Order q_t	Available	Demand	Leftovers
1	0	6	6	2	4
2	4	2	6	6	0
3	0	6	6	4	2
4	2	4	6	8	0

A quiet day leaves more stock, so the next order is smaller. Compare days 2 and 4: both end empty, but only day 4 has unfilled requests.

Stock remaining and unmet demand

After demand D_t, we record **stock remaining** and **unmet demand**:

After demand D_t	Quantity	What happens next
Stock remaining	$X_{t+1} = (Y_t - D_t)^+$	Carried into tomorrow
Unmet demand	$U_t = (D_t - Y_t)^+$	Customers leave

With $Y_t = 6$, demand $D_t = 4$ leaves **two units of stock**. Demand $D_t = 8$ instead leaves **two requests unfilled**.

Daily cost depends on morning inventory

Morning inventory **after replenishment** is $Y_t = X_t + q_t$. Charge $h = 1$ per leftover unit and $b = 4$ per unfilled request.

For demand D_t, the **daily cost** is

$$C_t = h(Y_t - D_t)^+ + b(D_t - Y_t)^+.$$

Before order X_t	Order q_t	Morning stock Y_t	Cost if $D_t = 4$	Cost if $D_t = 8$
2	4	6	2	8
4	2	6	2	8

Both choices give $Y_t = 6$ and incur the same cost under the same demand. With i.i.d. demand, **the same morning inventory Y_t gives the same daily cost distribution**.

Minimizing expected total cost

Policy **Definition 6.3**

A **policy** π chooses the order q_t using information available before demand D_t. It may depend on morning stock X_t and the observed past.

Examples of polices:

- **Order-up-to:** $q_t = (S - X_t)^+$.
- **Fixed daily order:** $q_t = 2$ every morning, regardless of X_t.

For a horizon of T days, $\mathbb{E}_\pi$ averages outcomes when following that policy. We choose π to minimize

$$\mathbb{E}_\pi \left[\sum_{t=1}^{T} C_t \right].$$

We count each day's holding and shortage costs, including the final day's holding cost. There is no additional terminal charge or salvage payment.

The daily cost has the familiar newsvendor form

Let D denote demand on a representative day. Let $g(y)$ be the **expected daily cost when y units are available before demand**:

$$g(y) = h\mathbb{E}[(y - D)^+] + b\mathbb{E}[(D - y)^+].$$

This has the same mathematical form as the expected loss in Section 6.1.

Newsvendor cost	Daily cost	Meaning in this shop
c_o	h	Cost of keeping a useful unit through one day
c_u	b	Cost charged when a request cannot be filled

To choose y, compare the consequences of **one more available unit**.

What does one more unit change?

Compare y and $y+1$ units under the **same demand outcome**. Stock and demand are integers.

Demand event	Consequence of the extra unit	Cost change
$D \leq y$	y units already serve everyone; the added unit remains	$+h$
$D > y$	The added unit fills a request that would otherwise be lost	$-b$

At $D = y$, the original stock exactly meets demand. The extra unit remains, so equality belongs in the first case.

> **? Which consequence is more likely?**
>
> For a low stock level, is the extra unit more likely to remain or to meet otherwise unfilled demand?

The expected change averages these two consequences using their probabilities.

Average the two marginal consequences

Let $F_D(y) = \mathbb{P}(D \le y)$ be the cumulative demand probability. Then $\mathbb{P}(D > y) = 1 - F_D(y)$.

By linearity, combine the holding terms and the shortage terms. For integer D and y:

$$
\begin{aligned}
g(y+1) - g(y) &= h\mathbb{E}\left[\underbrace{(y+1-D)^+ - (y-D)^+}_{\text{1 if } D \le y \text{ , 0 otherwise}}\right] + b\mathbb{E}\left[\underbrace{(D-y-1)^+ - (D-y)^+}_{\text{-1 if } D > y \text{ , 0 otherwise}}\right] \\
&= (+h)\mathbb{P}(D \le y) + (-b)\mathbb{P}(D > y) \\
&= hF_D(y) - b(1 - F_D(y)) \\
&= (h+b)F_D(y) - b.
\end{aligned}
$$

The extra unit reduces expected cost when

$$
(h+b)F_D(y) - b < 0 \quad \Longleftrightarrow \quad F_D(y) < \frac{b}{h+b}.
$$

The increment cannot decrease with y, so cost falls until adding a unit ceases to help.

Six is the best stock level for one day

Write $S^\star$ for the smallest optimal integer stock level. The marginal comparison gives

$$S^\star = \min\{s \in \mathbb{Z}_{\geq 0} : F_D(s) \geq \frac{b}{h+b}\}.$$

Daily demand D	2	4	6	8
Probability	0.10	0.40	0.40	0.10
Cumulative probability	0.10	0.50	0.90	1.00

The critical fractile is $\frac{4}{1+4} = 0.8$. Since $F_D(5) = 0.5 < 0.8 \leq F_D(6) = 0.9$, $\boldsymbol{S^\star = 6}$.

? Can we attain that choice every day?

We have found the best stock for **one day**.

A lower bound for every ordering policy

Consider any policy π that orders before seeing today's demand. Its choice can depend on the whole observed past.

Why the one-day bound applies. Once the available stock $Y_t = y$ is chosen, today's demand still has the same distribution as D because it is independent of the past. The expected cost is therefore $g(y) \geq g(S^\star)$.

If the policy chooses different values of Y_t on different histories, its expected cost averages such values of $g(y)$. Each is at least $g(S^\star)$, so $\mathbb{E}_\pi[C_t] \geq g(S^\star)$.

Add the daily bounds. By linearity of expectation,

$$\begin{aligned}\mathbb{E}_\pi\left[\sum_{t=1}^{T} C_t\right] &= \sum_{t=1}^{T} \mathbb{E}_\pi[C_t] \\ &\geq \sum_{t=1}^{T} g(S^\star) = T g(S^\star).\end{aligned}$$

Thus no ordering policy can have expected total cost below $T g(S^\star)$.

Base stock attains the bound every day

Start at the target. If $X_1 \le S^\star$, ordering $S^\star - X_1$ gives $Y_1 = S^\star$.

Keep tomorrow's optimum feasible. Whenever $Y_t = S^\star$,

$$X_{t+1} = (S^\star - D_t)^+ \le S^\star.$$

Immediate, unrestricted replenishment lets us order $q_{t+1} = S^\star - X_{t+1} \ge 0$ and restore $Y_{t+1} = S^\star$ every morning.

Attain the cost bound. We have $Y_t = S^\star$ for all t, so each day's expected cost is $g(S^\star)$. Expected total cost is exactly $Tg(S^\star)$, matching the lower bound.

> **Optimality over any number of days** **Theorem 6.1**
>
> Under our model and $X_1 \le S^\star$, the base-stock rule minimizes expected total cost over every finite horizon T. The minimum is $Tg(S^\star)$.

The cost of keeping a target of six

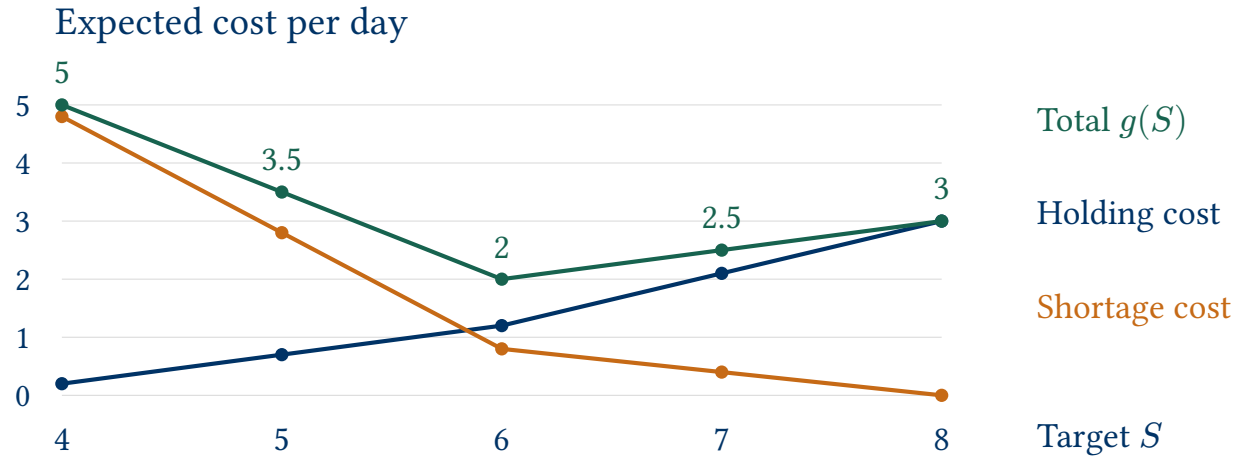
Every morning begins with six available units. The four demand outcomes give:

Demand	Probability	Leftovers	Unmet	Cost
2	0.1	4	0	4
4	0.4	2	0	2
6	0.4	0	0	0
8	0.1	0	2	8

$$\begin{aligned}
\text{Expected stock at day end:} \quad & \mathbb{E}[X_{t+1}] = 0.1(4) + 0.4(2) = 1.2 \ \text{ units}, \\
\text{Expected unmet demand:} \quad & \mathbb{E}[U_t] = 0.1(2) = 0.2 \ \text{ units/day}, \\
\text{Expected daily cost:} \quad & g(6) = \underbrace{1(1.2)}_{\text{holding cost}} + \underbrace{4(0.2)}_{\text{shortage cost}} = 2 \ \text{ cost units/day}.
\end{aligned}$$

Thus expected total cost over T days is $2T$. Individual days still have different outcomes.

Larger targets exchange shortages for holding cost



Target S	4	5	6	7	8
Expected cost/day	5	3.5	**2**	2.5	3

Mean demand is five, but topping up to five costs **3.5 per day**. Target eight eliminates shortages in this demand model, at a cost of **3 per day**. The best target balances holding and shortage costs.

Two measures of inventory service

Cycle service level — Definition 6.4

The probability of filling **all demand** in a replenishment cycle.

Example: Each cycle is one day. At target six, only demand eight causes a shortage.

$$\mathbb{P}(D \leq 6) = 0.9.$$

Unit fill rate — Definition 6.5

The long-run fraction of requested units supplied **immediately from stock**.

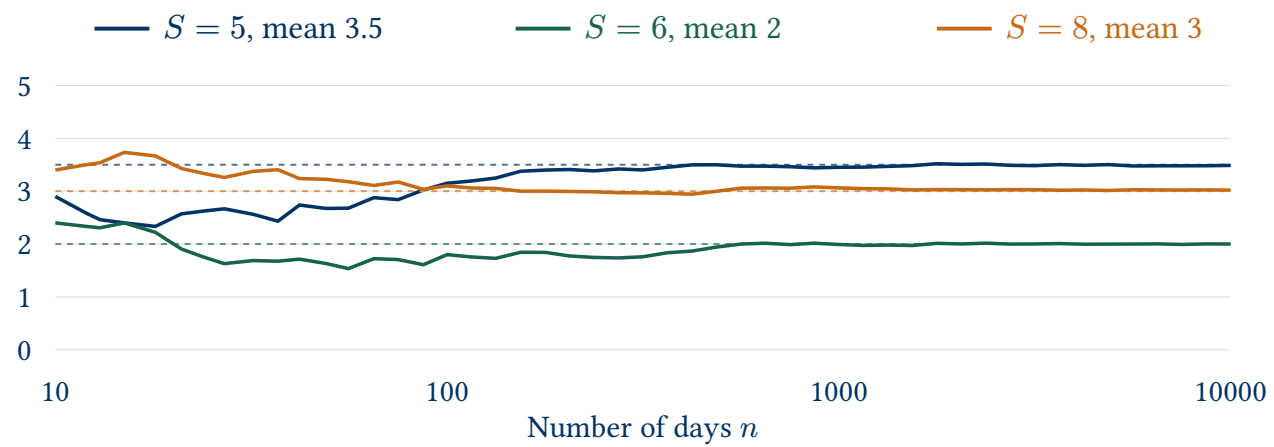
Example: Mean unmet demand is 0.2 units per day and mean demand is five.

$$\text{Fill rate} = 1 - \frac{\mathbb{E}[U_t]}{\mathbb{E}[D_t]} = 1 - \frac{0.2}{5} = 0.96.$$

Estimate fill rate by dividing total units supplied by total units requested.

Sample averages approach the expected performance

Each curve shows average cost after n days, $n^{-1} \sum_{t=1}^{n} C_t$. All targets face the same simulated demand stream.



Dashed lines show the exact means. Starting with an empty shelf, each target is attainable every day. Daily costs are then independent copies, so the **law of large numbers** explains convergence. A short run can still rank the targets differently.

Principle to remember

A daily optimum can be repeated when it remains attainable

- Carryovers determine **how much to order**.
- The newsvendor tradeoff determines the **target stock**.
- Immediate, unrestricted supply lets us restore that target every day.
- Independent, unchanged demand makes the same target appropriate each day.

What changes next? A fixed ordering fee makes frequent top-ups costly. A delivery delay prevents an order from immediately restoring the shelf.

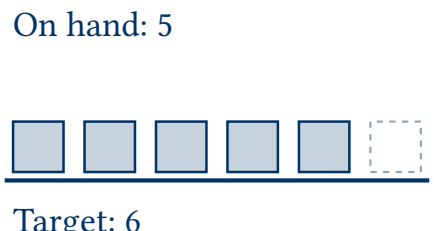
The base-stock result rests on the operating assumptions.

6.3. Fixed Ordering Fees and Batching

Would we pay for a delivery of one unit?

Model change: Keep daily review and immediate delivery. Let K be a fee paid once per positive order, regardless of its size. Use $K = 7$.

Five units remain. The earlier target was six, so the base-stock rule would order one.



? Order now or wait?

What might we gain by ordering today? What might we save by waiting and combining several replenishment needs?

When does waiting become too costly?

For morning stock $x < 6$, compare waiting with ordering up to the best stock level for one day, six. A positive order adds the fee $K = 7$, so bringing stock to six costs $7 + g(6) = 9$.

Recall the expected daily holding and shortage cost:

$$g(x) = h\mathbb{E}[(x - D)^+] + b\mathbb{E}[(D - x)^+], \quad h = 1, b = 4.$$

Stock x	Wait: $g(x)$	Order to six: $K + g(6)$	Lower-cost choice
1	16	9	Order
2	12	9	**Order**
3	8.5	9	**Wait**
5	3.5	9	Wait

Waiting costs less at three and more at two. Since $g(x)$ decreases up to six, the **one-day order trigger is two**: order at $x \leq 2$. At or above six, adding stock only raises cost.

Future holding, shortage, and ordering costs can change the trigger and the refill target.

Once we order, how far should we replenish?

Suppose inventory is low enough to justify paying for delivery.

Possible order	Consequence
Add only one unit	Another order may be needed very soon
Order several days' expected demand	Fewer orders and fewer fixed fees
Fill the storeroom	More units wait on the shelf and incur holding costs

We need both an **order trigger** and a **replenishment target**.



Observation

The trigger answers **when to order**. The target answers **how far to replenish once we order**.

In the continuing operation, both levels depend on future holding, shortage, and ordering costs. The best target can change when a fixed fee is introduced.

The (s, S) policy

Reorder point and order-up-to level **Definition 6.6**

Choose a **trigger s** and a higher **target S**, with $s < S$. At each morning review, if stock is at or below s, replenish to S. Otherwise, order nothing.

For current stock x, the order quantity $q(x)$ is

$$q(x) = \begin{cases} S - x & \text{if } x \leq s \\ 0 & \text{if } x > s \end{cases}.$$

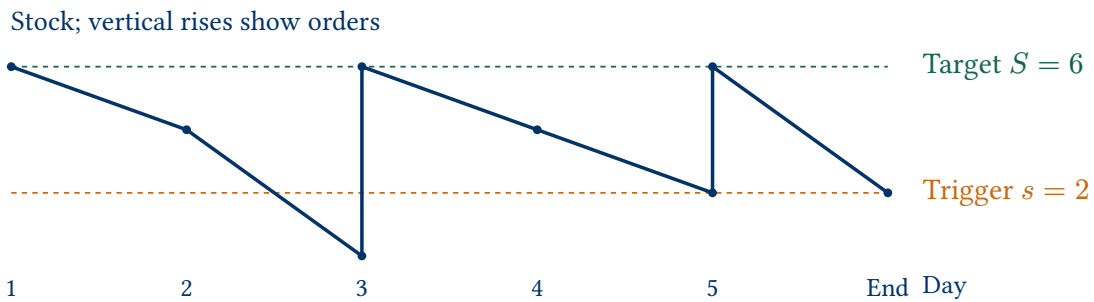
The one-day comparison gave $s = 2, S = 6$. Here is what that rule orders:

Morning stock	5	3	2	1	0
Order quantity	0	0	4	5	6

At stock three, we are below the target but still **above the trigger**, so we wait. At stock one, we replenish five units: the order need not equal $S - s$.

Waiting and replenishment along one demand path

Use $s = 2, S = 6$ and start with six on the shelf. Review every morning, before demand.



Day	Morning stock	Order	Demand	Leftovers
1	6	0	2	4
2	4	0	4	0
3	0	6	2	4
4	4	0	2	2
5	2	4	4	2

Stock can fall below s between reviews. The levels come from the one-day calculation. Repeated use can favor different levels.

A Markov chain for the $(2, 6)$ policy

Recall: iid daily demand takes values $2, 4, 6, 8$ with probabilities $0.1, 0.4, 0.4, 0.1$, respectively.

Starting from six, stock stays even and never exceeds six. Demand is at least two, so after the first day the morning stock is $X_t \in \{0, 2, 4\}$.

Orders depend only on current stock; iid demand makes $\{X_t\}$ a **Markov chain**.

The policy determines available stock.

Stock x	Order $q(x)$	Available
0	6	6
2	4	6
4	0	4

State order: 0, 2, 4.

$$P = \begin{pmatrix} 0.5 & 0.4 & 0.1 \\ 0.5 & 0.4 & 0.1 \\ 0.9 & 0.1 & 0 \end{pmatrix}.$$

First two rows: replenish to six. Demand six or eight leaves zero, with probability 0.5.

Last row: wait with four. Demand four, six, or eight leaves zero, with probability 0.9.

From stationary probabilities to daily cost

For continuing operation, compare **long-run average cost per day**. The stationary probability π_x is the fraction of mornings with stock x. Solving $\boldsymbol{\pi} P = \boldsymbol{\pi}$ and $\pi_0 + \pi_2 + \pi_4 = 1$ gives

$$(\pi_0, \pi_2, \pi_4) = \left(\frac{59}{110}, \frac{41}{110}, \frac{10}{110} \right).$$

Expected cost at each state, with $h = 1$, $b = 4$, and $K = 7$:

Stock x	Holding	Shortage	Order fee	Total $c(x)$
0	1.2	0.8	7	9
2	1.2	0.8	7	9
4	0.2	4.8	0	5

Weight daily costs by these stationary probabilities:

$$J(2, 6) = 9(\pi_0 + \pi_2) + 5\pi_4 = \frac{95}{11} \approx 8.64.$$

Choosing the refill target S

Keep $s = 2$. Use each target's Markov chain to calculate **long-run average costs per day**.

Target S	Holding	Shortage	Order fees	Total
6	1.11	1.16	6.36	8.64
8	2.12	1.36	4.64	8.11
10	**2.96**	**1.32**	**3.52**	**7.80**
12	4.00	0.89	2.99	7.88



Observation

Once we pay for an order, extra stock can **postpone the next order** and its fee. But units left waiting on the shelf incur holding costs.

Among these targets, **ten costs least**: at twelve, added holding outweighs the savings.

Choosing the trigger s

Keep $S = 10$ and compare triggers using **long-run average costs per day**.

Trigger s	Holding	Shortage	Order fees	Total
0	2.51	2.93	2.99	8.43
2	**2.96**	**1.32**	**3.52**	**7.80**
4	3.84	0.22	4.64	8.69

Wait longer: $s = 0$ saves fees and holding costs, but loses more sales. **Order sooner:** $s = 4$ avoids more shortages, but pays more in fees and holding.

Choose both levels together. Repeat the Markov-chain calculation for all integer pairs $0 \leq s < S \leq 30$. The minimum is **7.80 per day** at $(2, 10)$ and $(3, 10)$.

These triggers give the same orders once stock is even. This is the best result **within the searched policies**.

What service does the least-cost batching policy deliver?

$(2, 10)$ and $(3, 10)$ tied in the cost search. How often does the shop order under $(2, 10)$, and how much demand does it fill? Each rule starts empty and faces the same 10,000 simulated days. Delivery is immediate, with $h = 1$, $b = 4$, and $K = 7$.

Policy	Orders per 100 days	Mean day-end stock	Unit fill rate	Modeled cost per day
Top up to 6	100.00	1.22	96.06%	9.00
$(2, 6)$	90.56	1.12	94.16%	8.62
$(2, 10)$	**50.15**	**2.98**	**93.45%**	**7.80**
$(4, 10)$	65.82	3.85	98.91%	8.67

Among these four policies, which has the lowest estimated cost? Which would you use if the shop requires an estimated unit fill rate of at least **98%**?

How the 10,000-day run gives the $(2, 10)$ row

The shop placed **5,015** positive orders. Its day-end stock summed to **29,838** units over the run. Of the **49,800** units customers requested, the shop could not supply **3,260**.

Measure	Calculation
Orders per 100 days	$100 \times 5015 \div 10000 = 50.15$
Mean day-end stock	$29838 \div 10000 = 2.9838$ units
Unit fill rate	$1 - 3260 \div 49800 \approx 93.45\%$

Pool units across days for the fill rate. A day with eight requests contributes twice as many units as a day with four.

Add holding cost, shortage cost, and order fees over the run, then divide by days.

$$\text{Modeled cost per day} = \frac{1 \times 29838 + 4 \times 3260 + 7 \times 5015}{10000} = 7.7983.$$

Which batching policy fits the shop's objective?

Cost alone. $(2, 10)$ costs **7.80 per day**, the lowest of the four estimates. It places about 50 orders per 100 days and fills 93.45% of requested units.

At least 98% unit fill. Only $(4, 10)$ clears that threshold in this run. It costs **8.67 per day** and fills **98.91%**.

Both policies refill to ten. Raising the trigger from two to four lets the shop reorder sooner. It places about **16 more orders per 100 days**, and mean day-end stock rises from **2.98 to 3.85** units.

Batching balances fees, holding, and service

Operating change	Reason to reconsider the policy
Larger ordering fee	Combining needs into fewer orders becomes more attractive
Higher holding cost	Carrying large batches becomes more expensive
More costly shortages	Waiting with little stock becomes less attractive



Observation

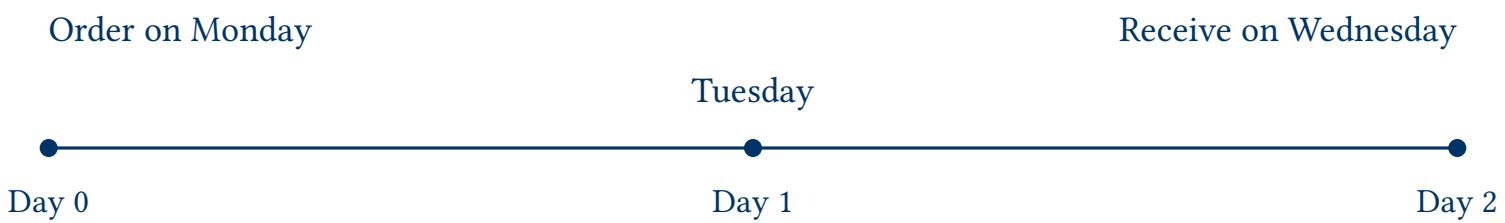
A fixed fee creates a reason to wait and batch orders. Holding and shortage costs limit how far that reasoning should be taken.

So far each delivery arrived before the day's customers. What changes if a Monday order first reaches the shelf on Wednesday?

6.4. Replenishment with Lead Times

Customers keep arriving while we wait

Model change: Keep daily review and lost sales. Orders arrive two mornings later. There is no fixed ordering fee.



Lead time is the interval from placing an order until that stock becomes available. Monday and Tuesday's customers arrive before Monday's order is received.

Each day the shop receives deliveries due, reviews stock and places its order, serves customers from the shelf, then carries leftovers into tomorrow.

If the shelf is empty on Monday, can Monday's new order serve Monday's customers?

Identical shelves can require different orders

Both shops have four units on the shelf. Shop A has ten on order and Shop B has none. **Should they order the same quantity?**

Inventory position with lost sales — Definition 6.7

On hand counts units available now. **On order** counts units still due from the supplier. The **inventory position** counts all supply already committed.

$$\text{IP} = \text{on hand} + \text{on order}.$$

Each shop aims for a **position of 18**.

Shop	On hand	On order	Position	New order
A	4	10	14	$18 - 14 = 4$
B	4	0	4	$18 - 4 = 14$

Order $q = (S - \text{IP})^+$. Counting only the shelf would make Shop A order its ten outstanding units twice.

What changes when the earlier order arrives?

Suppose Shop A receives the ten units before making a sale. **Which entries will change when the delivery arrives?**

Event	On hand	On order	Position
Initial situation	4	10	14
Order 4 more	4	14	18
Receive the earlier 10	14	4	**18**
Sell 6 units	8	4	12

A new order commits more supply and raises position. A receipt moves already counted units from **on order** to **on hand**.

Only stock on hand can meet demand now. Fulfilled sales reduce position. Unfilled requests leave without creating a future obligation.

Arrival dates determine near-term availability

Now suppose both shops have four on hand and ten on order, but Shop B's supplier is late.

Shop	On hand	On order	Position	Arrival
A	4	10	14	Tomorrow morning
B	4	10	14	In three days

? Are the shops equally protected?

Suppose nobody buys today. Which shop can fill a request for six units tomorrow, after the morning deliveries?

Both shops would order four to reach position 18. Only Shop A can fill tomorrow's request. Position is enough to apply this rule. Predicting service also requires the **delivery schedule**.

Shop A on Monday morning

Shop A reviews every morning, after receipts and before demand. A new order takes two days to arrive.

On Monday the shelf holds **four units**. An earlier order of **ten** is due Tuesday morning. To reach position 18, the shop orders

$$q = 18 - (4 + 10) = 4 \quad \text{units, due Wednesday morning.}$$

The shop orders again on Tuesday and Wednesday. Those orders arrive on Thursday and Friday, after the three days shown here.

? Six customers each want one unit on Monday

After ordering, the position is 18. How many customers can buy today? What happens to the remaining requests?

Monday's customers arrive before either delivery

Day	Morning receipt	Available before demand	Demand	Lost sales	Day-end stock
Monday	0	4	6	2	0

Four customers buy a unit each. The other two leave. All 14 units on order are still in the **delivery pipeline**.

> **? What will be on the shelf on Tuesday evening?**
>
> Ten units arrive Tuesday morning. Four customers each request one unit that day.

Tuesday's receipt refills the empty shelf

Day	Morning receipt	Available before demand	Demand	Lost sales	Day-end stock
Monday	0	4	6	2	0
Tuesday	10	10	4	0	6

Tuesday's four customers all buy a unit. Six units carry over to Wednesday.

? Wednesday brings Monday's order and demand for six

How much stock will remain? Across the three days, customers request only 16 units. How could Monday's position of 18 still leave sales unfilled?

Wednesday's receipt and the three-day result

Day	Morning receipt	Available before demand	Demand	Lost sales	Day-end stock
Monday	0	4	6	2	0
Tuesday	10	10	4	0	6
Wednesday	4	$6 + 4 = 10$	6	0	4

Of the 18 units committed on Monday, the shop sold **14** and finished with **four**. The two customers who left on Monday could not use Tuesday's or Wednesday's delivery.

How far ahead must today's plan protect?

Shop A reviews daily and delivery takes two days. **When can the next review's order first serve a customer?**

Decision	Placed	Available
Today's order	Monday	Wednesday morning
Next review's order	Tuesday	Thursday morning

Tuesday's order arrives Thursday morning, so Monday's decision must account for demand on **Monday, Tuesday, and Wednesday**.

> **Protection period** **Definition 6.8**
>
> In a periodic-review system, the protection period is the time from the current review until an order placed at the next review becomes available. It equals $R + L$, where R is the review interval and L is the fixed replenishment lead time.

In our example with daily review and two-day lead time, the protection period $R + L = 1 + 2 = 3$ days.

What changes if the shop reviews only on Mondays?

Model change: Orders can now be placed every Monday. Delivery still takes two days.



Why look nine days ahead when delivery still takes only two?

The shop waits seven days for another opportunity to order. An order placed then takes two more days to reach the shelf.

$$\text{Protection period} = R + L = 7 + 2 = 9 \text{ days}.$$

Weekly review may simplify ordering, but the shop must plan for six more days of demand than under daily review.

How much safety stock does a target of 18 provide?

Return to daily review and two-day delivery. Demand averages five units per day. Over the three-day protection period, expected demand is $3 \times 5 = 15$ units.

Safety stock — Definition 6.9

Safety stock is the part of the position target above expected demand over the protection period.

$$\underbrace{18}_{\text{position target}} = \underbrace{15}_{\text{expected demand}} + \underbrace{3}_{\text{safety stock}}.$$

Those three units are a planning allowance. Shop A had position 18 on Monday but only four units on the shelf. Its buffer could help customers only once the stock arrived.

? What cost and service does this buffer produce?

How much demand will the shop fill over many days? Could spreading its orders across reviews reduce holding cost?

A capped base-stock policy

Now start Monday with an **empty shelf and no outstanding orders**. If the shop orders 18 on Monday, all 18 arrive on Wednesday. At most eight can sell that day. Ten could arrive on Thursday with the same Wednesday sales.

The **target** is how much supply the shop wants to have committed. The **cap** is how much it chooses to order at one review, even if the supplier can ship more. Try a cap of six and a target of 16, one unit above mean three-day demand.

$$q = \min(6, (16 - \mathrm{IP})^+).$$

Position at review	Gap to 16	New order	Position after ordering
0	16	6	6
12	4	4	16
16	0	0	16

From position zero the first order gets the shop only to six. **How many units can be on order after a review?** Only yesterday's and today's orders can still be in transit, at most $6 + 6 = 12$ units.

What do we gain by changing the cap or the target?

On Wednesday's **empty shelf**, the sixth unit sells if demand is six or eight (50%). The seventh sells only at eight (10%). With holding cost 1 and lost-sale cost 4, the change in expected cost **that day** is

$$\text{Sixth:}\quad 0.5(1) - 0.5(4) = -1.50 \qquad \text{Seventh:}\quad 0.9(1) - 0.1(4) = +0.50.$$

This suggests six as a candidate cap. Leftovers and incoming deliveries change each day. The table gives **cost per day** when each rule runs for the same 10,000 simulated days, starting empty with no order fee.

Cap per order	Target 15	Target 16	Target 17
5	3.46	3.23	3.24
6	3.44	**3.17**	3.40
7	3.47	3.21	3.46

Read across the middle row. Why does cost fall as the target rises from 15 to 16, then rise at 17? Which costs does a larger buffer increase, and which can it reduce?

When is a capped base-stock policy optimal?

Suppose a shop reviews one item daily. Its supplier delivers any quantity up to a fixed limit $C > 0$ immediately, before demand. Daily demands are i.i.d. with a density on $[0, \infty)$ and a finite positive mean. Unfilled demand is lost. Purchasing, overnight holding, and lost sales each incur a constant positive cost per unit. There is no fixed order fee.

> **Capped base-stock optimality** **Theorem 6.2**
>
> For some fixed target $S^\star$, ordering $q(x) = \min\left(C, (S^\star - x)^+\right)$ whenever x units are on hand minimizes long-run average expected cost.

Proof idea. Dynamic programming gives a convex cost as a function of stock after ordering. The rule moves stock as close to its minimizing target as capacity allows. The proof for the long-run average is beyond this course. Huh and Janakiraman, Theorem 19.

Our shop waits two days for delivery and chooses its own cap. We compare its candidate rules by simulated cost and fill rate.

Which rule should the owner adopt?

With a cap of six, raising the target from 16 to 17 increased cost. How much more demand did it fill? All four policies below face the **same 10,000 simulated days**.

Ordering rule	Cost per day	Unit fill rate
Top up to 16	3.2460	93.72%
Top up to 16, at most six per order	3.1734	93.69%
Top up to 17, at most six per order	3.3975	95.79%
Top up to 18	3.7532	98.00%

? Make a recommendation

Which of these rules costs least? Which has the lowest estimated cost while filling at least 95% of requested units immediately?

What does the 95% requirement change?

On cost alone, choose target 16 with cap six. At **3.1734 per day**, it has the lowest estimated cost among the four rules and fills 93.69% of requested units.

For at least 95% unit fill, keep the cap at six and raise the target to **17**. It filled **95.79%** in this simulation, at an additional cost of $3.3975 - 3.1734 = 0.2241$ per day. It is the least costly displayed rule to clear 95%.

The uncapped target of 18 filled **98.00%** at **3.7532 per day**. The owner can compare the value of that additional service with its cost.

> **? How would the owner check service in the shop?**
>
> Yesterday six units were available and all six sold. Could unit fill have been 100%? Could it have been 75%? What would the owner need to record to tell?

Can the sales register verify a 95% unit fill rate?

When all six units sell, demand is **right-censored**. We know customers wanted **at least six**.

Units requested	Units sold	Units lost	Unit fill rate
6	6	0	$6 \div 6 = 100\%$
8	6	2	$6 \div 8 = 75\%$

To measure the actual fill rate, record **units sold** and **units requested but unfilled**, including the missing units in a partially filled request.

$$\text{Unit fill rate} = \frac{\text{total units sold}}{\text{total units sold} + \text{total unfilled units}}.$$

If customers leave without asking, their missing requests make the shop's recorded fill rate look too high. Demand estimation and forcast under censored data is a frontier research area.

How much stock does ordering in batches create?

Cycle stock — Definition 6.10

Cycle stock is the inventory a shop carries to meet ordinary demand between successive replenishments. It builds up when a batch arrives and falls as customers buy.

Suppose customers buy **exactly five units each day**. Deliveries arrive before opening, just as the previous batch runs out. Compare four days of sales.

Delivery schedule	Stock at the end of days 1–4	Mean day-end stock
20 units every four days	15, 10, 5, 0	7.5 units
10 units every two days	5, 0, 5, 0	2.5 units

Both schedules supply all 20 units requested. Smaller batches save five units of average overnight stock but require twice as many deliveries. A lower fee per order makes that trade-off more attractive.

How many units are on order? Little's law

Pipeline inventory counts units from order placement until receipt. Follow one unit through that interval. Each day it spends on order contributes one unit-day to the pipeline.

In steady operation, Little's law gives

$$\text{mean units on order} = \text{units received per day} \times \text{mean days from order to receipt per unit.}$$

Suppose the shop orders five units every morning and each delivery takes two days. After Monday's review, Sunday's five units are due Tuesday and Monday's five are due Wednesday.

$$5 \ \text{units/day} \times 2 \ \text{days} = 10 \ \text{units on order.}$$

One-day delivery cuts the average to **five units** at the same throughput. If some customers leave empty-handed, use the actual replenishment rate in this calculation. Unfilled requests never enter the pipeline.

What demand should the owner forecast on Monday?

Monday's order decision must cover demand until Thursday morning. Forecast the **total units customers will request from Monday through Wednesday**, using what the owner knows at Monday's review.

Expected demand	Mon	Tue	Wed	Three-day total
Shop's i.i.d. model	5	5	5	15
Known Wednesday promotion	5	5	8	18

In the i.i.d. model, a position target of 18 includes **three units of safety stock**. Suppose instead that a promotion raises Wednesday's forecast to eight. The same target of 18 now covers only the expected demand.

Keeping the three-unit buffer would give a **candidate target of 21**. Whether three extra units are enough depends on how far actual demand can exceed the forecast and when deliveries arrive.

Pause and recall

Replenishment with lead times

1. Monday's position is 18 and total demand through Wednesday is 16. Why can Monday sales still be lost?
2. With daily review and two-day delivery, why must Monday's plan cover three days?
3. What would you need to know to recommend an ordering rule and then check its unit fill rate in the shop?

Pause here. Try each question before turning to the answers.

Answers and reasoning

Replenishment with lead times

1. Only stock on the shelf can serve Monday's customers. Some of the 18 units arrive on Tuesday or Wednesday, after the customers have left.
2. The next review is Tuesday. An order placed then first serves customers on Thursday, so Monday's plan covers Monday through Wednesday.
3. To choose a rule, use on-hand stock, quantities due and their arrival dates, a demand model, and the owner's cost or service objective. To check fill, count units sold and units requested but unfilled over the same period.

7. Markov Decision Processes

7.1. Finite-Horizon Markov Decision Processes

Run or Maintain?

Inspect the machine at the beginning of each shift. It is **good** (G) or **worn** (W). Both actions are available in either condition.

Condition	Action	Output now	Next condition
G	Run	6	G or W, each with probability $\frac{1}{2}$
W	Run	2	W with probability 1
Either	Maintain	0	G with probability 1

Each action takes one shift. We maximize **expected total output over the remaining shifts**. The condition after the final shift earns no reward.

With one shift left, what should we do if the machine is worn? What if it is good?

The final-shift decision

Only output during this shift counts.

Condition	Run	Maintain	Best action
G	6	0	**Run**
W	2	0	**Run**

Run in either condition. Maintenance can improve the condition of the machine, but there is no later shift to benefit.

Would you still run a worn machine with **two shifts** left?

Two shifts remaining: worn machine

Compare the two actions now, then run in the final shift from the resulting condition.

Action now	Output now	Next condition	Final output	Total
Run	2	W	2	$2 + 2 = 4$
Maintain	0	G	6	$0 + 6 = \mathbf{6}$



Why maintenance can be optimal

Maintenance gives up 2 units now and gains 4 in the final shift by leaving the machine good. That future gain makes maintenance the better first action over two shifts.

Two shifts remaining: good machine

If we **run now**, this shift produces 6 units before any deterioration. In the final shift, run in whichever condition we observe.

Next condition	Probability	Final output	Weighted output
G	$\frac{1}{2}$	6	$\frac{1}{2} \times 6 = 3$
W	$\frac{1}{2}$	2	$\frac{1}{2} \times 2 = 1$

$$\text{Run now:} \quad 6 + (3 + 1) = \mathbf{10}.$$

$$\text{Maintain now:} \quad 0 + 6 = 6.$$

Run gives the larger expected total. The 6 units today are certain. Only the final shift's output needs to be averaged over the two possible conditions.

The optimal value function

Optimal finite-horizon value function — Definition 7.1

$V_h(i)$ is the **maximum expected total reward** over the remaining h shifts, starting in state i. Here, the reward is the machine's output.

Shifts left	$V_h(G)$	$V_h(W)$	Action at G	Action at W
0	0	0	None	None
1	6	2	Run	Run
2	10	6	Run	Maintain

$V_2(G) = 10$ counts both shifts under the best choices. With no shifts left, $V_0(G) = V_0(W) = 0$. Backward induction starts there and adds one shift at a time.

Three shifts: instructions for the operator

Start in G. Under Always Run, the chance of still being good halves each shift. The expected outputs in shifts 1, 2, and 3 are 6, $\frac{1}{2} \times 6 + \frac{1}{2} \times 2 = 4$, and $\frac{1}{4} \times 6 + \frac{3}{4} \times 2 = 3$, for a total of 13. Tomorrow we will inspect the machine. Can that information improve the instructions?

? A rule for every possible observation

Start with a **good** machine and **three shifts left**.

If you run today, what should the operator do tomorrow:

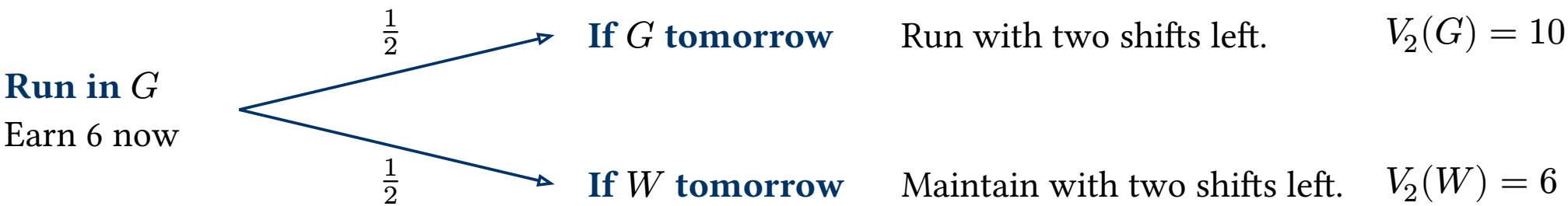
1. if the machine is good?
2. if it is worn?

What should the operator do in the final shift?

Tomorrow begins the two-shift problem we have already solved from either condition.

Tomorrow's action uses tomorrow's condition

Consider running today. A good machine produces 6 units this shift. Tomorrow's inspection tells us which two-shift solution to use.



In the final shift, run in either condition. Each branch value already includes that decision.

$$\text{Expected total} = 6 + \frac{1}{2} \times 10 + \frac{1}{2} \times 6 = \mathbf{14}.$$

Using tomorrow's observation raises expected output from 13 under Always Run to 14.

Three shifts: compare both first actions

Use $V_2(G) = 10$ and $V_2(W) = 6$. After today's action, follow the best two-shift rule.

State	Run now	Maintain now	Best value
G	$6 + \frac{1}{2} \times 10 + \frac{1}{2} \times 6 = 14$	$0 + 10 = 10$	$V_3(G) = 14$
W	$2 + 6 = 8$	$0 + 10 = 10$	$V_3(W) = 10$

The continuation after running from W: today produces 2 and leaves the machine worn. With two shifts left, maintain and then run:

$$2 + V_2(W) = 2 + (0 + 6) = 8.$$

With three shifts left, run in G and maintain in W. Tomorrow, inspect the machine again and use the two-shift rule.

Policies and observed information

Policy **Definition 7.2**

A policy specifies how to choose a feasible action using the information available at a decision. This can include previous states and actions. A fixed schedule ignores later observations.

A deterministic Markov policy for this problem specifies an action $f_h(i)$ using only state i and the number h of shifts left. Here is the rule we just calculated.

Shifts left h	If good: $f_h(G)$	If worn: $f_h(W)$
3	Run	Maintain
2	Run	Maintain
1	Run	Run

Choose the current action before seeing the next state. After observing that state, use the policy for the remaining shifts. The Always Run policy chooses Run regardless of the observations.

Components of a Markov decision process

Markov decision process (MDP) **Definition 7.3**

A Markov decision process specifies states, available actions, rewards, and transition probabilities. The current state and chosen action determine the reward model and the distribution of the next state.

Ingredient	Machine example
Decision epoch	Beginning of each shift
State i	Observed condition: G or W
Available actions $\mathcal{A}(i)$	Run and Maintain in either condition
Reward $r(i, a)$	Output in the current shift
Transition $P_{ij}(a)$	Probability of next state j given i and a
Objective	Maximize expected total output before the deadline

The Markov assumption

Let X_t be the condition at the beginning of shift t, and A_t the action chosen then.

Controlled Markov property — Definition 7.4

For a given current state and action, the next-state distribution is independent of earlier history.

$$\mathbb{P}(X_{t+1} = j \mid X_t = i, A_t = a, \text{earlier history}) = P_{ij}(a).$$

In this model, the reward is also determined by the current state and action. Two histories that give the same state and number of remaining shifts lead to the same optimization problem, so we can reuse its value.

Quick check. If machine age changes the chance that maintenance succeeds, what should the state record? Include age so the state and action determine the next-state probabilities.

Bellman recursion for a worn machine

Each action uses one shift. The next condition starts a problem with $h-1$ shifts left.

Action	Output now	Next condition	Best remaining value
Run	2	W	$V_{h-1}(W)$
Maintain	0	G	$V_{h-1}(G)$

$$V_h(W) = \max\left\{\underbrace{2 + V_{h-1}(W)}_{\text{Run}},\quad \underbrace{0 + V_{h-1}(G)}_{\text{Maintain}}\right\}.$$

Maintenance sacrifices 2 units now. It is better when its gain in future value exceeds that sacrifice:

$$V_{h-1}(G) - V_{h-1}(W) > 2.$$

Equality gives a tie.

Bellman recursion for a good machine

In the Markov-chain chapter, we conditioned on the first transition and restarted the problem from the next state. Here we do that once for **each possible action**.

Action	Output now	Expected best continuation
Run	6	$\frac{1}{2}V_{h-1}(G) + \frac{1}{2}V_{h-1}(W)$
Maintain	0	$V_{h-1}(G)$

$$V_h(G) = \max\left\{6 + \frac{1}{2}V_{h-1}(G) + \frac{1}{2}V_{h-1}(W), \quad 0 + V_{h-1}(G)\right\}.$$

First-step analysis with a choice

For each action, add today's reward to the probability-weighted continuation value for its possible next states. Compare those totals before choosing today's action. Tomorrow, observe the condition and follow its optimal continuation rule.

Bellman's principle of optimality

Bellman's principle of optimality **Proposition 7.1**

After each reachable next state, an optimal policy uses an optimal continuation for the remaining horizon.

If a continuation can be improved after an outcome of positive probability, improving it also raises the policy's expected total.

After running from G with three shifts left, tomorrow is worn with probability $\frac{1}{2}$.

- Running for both remaining shifts produces $2 + 2 = 4$.
- Maintaining and then running produces $0 + 6 = 6$.

$$\text{Expected gain from improving that branch:} \quad \frac{1}{2} \times (6 - 4) = 1.$$

This raises the expected total from 13 to 14. The branch starting good keeps its best continuation.

Finite-horizon Bellman equation

We observe the current state. There are finitely many states, each with a finite, nonempty set of actions. Rewards are finite, and the reward and transition models are the same at every decision epoch.

Finite-horizon Bellman equation — Theorem 7.1

With zero terminal reward, $V_0(i) = 0$. For $h \geq 1$,

$$V_h(i) = \max_{a \in \mathcal{A}(i)} \left\{ r(i,a) + \sum_j P_{ij}(a) V_{h-1}(j) \right\}.$$

Choosing a maximizing action $f_h(i)$ for every state and horizon gives an optimal deterministic Markov policy.

Today's action is chosen before the next state is known. That is why the maximum is outside the sum.

Once the state and remaining horizon are known, earlier history adds no predictive information. Randomizing averages the available action values, so a single maximizing action does at least as well.

Backward induction

An **MDP** specifies the decision model. **Dynamic programming** solves it by storing and reusing the answers to smaller remaining problems.

Step	Calculation
Initialize	$V_0(i) = 0$ for the machine. If final state i earns $s(i)$ at the deadline, set $V_0(i) = s(i)$.
Extend	For $h = 1, 2, \ldots$, use V_{h-1} to compare actions at every state
Store	Keep the value $V_h(i)$ and a maximizing action $f_h(i)$

Compute backward and use the policy forward

Start at the deadline and add one earlier shift at a time, so $h = 0, 1, 2, \ldots$. During operation, observe the state and use the stored action for the number of shifts still to come.

Four shifts remaining (exercise)

? Which action would you take?

Use $V_3(G) = 14$ and $V_3(W) = 10$.

Calculate the total for both actions in each state. Find $V_4(G)$, $V_4(W)$, $f_4(G)$, and $f_4(W)$.

State	Run now	Maintain now
G	$6 + \frac{1}{2} \times 14 + \frac{1}{2} \times 10$	$0 + 14$
W	$2 + 10$	$0 + 14$

Explain why every continuation value uses **three** remaining shifts.

Four shifts: values and decisions

State	Run now	Maintain now	Best action
G	$6+7+5=18$	$0+14=14$	Run
W	$2+10=12$	$0+14=14$	Maintain

$$V_4(G)=\max\{18,14\}=18, \quad V_4(W)=\max\{12,14\}=14.$$

Today's action consumes one of the four shifts. The best response after the transition therefore uses the three-shift values.

We compute the policy backward and follow it forward, using the observed condition and shifts remaining.

So far, a unit of output counts the same in every shift. How does the decision change if later output carries less weight?

7.2. Infinite-Horizon Discounted Reward

Net present value

Net present value adds cash flows in today's dollars. At 10% per year, paying \$100 now for \$110 in a year gives $-100 + \frac{110}{1.10} = 0$. For net cash flow C_t in period t and discount rate d per period,

$$\text{NPV} = C_0 + \gamma C_1 + \gamma^2 C_2 + \ldots, \quad \gamma = \frac{1}{1+d}.$$

The contract runs indefinitely. Each unit of output earns the same net cash amount, so maximizing expected NPV amounts to maximizing expected discounted output. Use $\gamma = \frac{1}{2}$ for the example.

? Does maintenance still pay?

A worn machine can produce 2 now or spend the shift on maintenance. Does the discounted gain in future output justify giving up those 2 units?

Discounted values and stationary policies

Let γ be the discount factor, with $0 \leq \gamma < 1$. Number the current shift $t = 0$.

Discounted value of a policy — Definition 7.5

For a policy f, the expected discounted reward starting in state i is

$$v^f(i) = \mathbb{E}_i^f \left[\sum_{t=0}^{\infty} \gamma^t r(X_t, A_t) \right].$$

The NPV formula discounts the reward in each shift. The expectation averages over the possible paths of machine conditions and rewards under f.

A **deterministic stationary policy** prescribes the same action $f(i)$ on every visit to state i. The rule stays fixed even as the distribution of machine conditions changes.

The same model applies next shift. Discount its value by γ to express it in today's terms.

The discounted Bellman optimality equation

Let $v^\star(i)$ be the maximum expected discounted reward starting in i.

Discounted Bellman optimality equation **Theorem 7.2**

Under our finite-model assumptions, with $0 \le \gamma < 1$, $v^\star$ is the unique solution to

$$v^\star(i) = \max_{a \in \mathcal{A}(i)} \left\{ r(i,a) + \gamma \sum_j P_{ij}(a) v^\star(j) \right\}.$$

An optimal deterministic stationary policy exists. It chooses a maximizing action in every state.

The same values appear on both sides because each decision starts the same discounted problem from its current state. Decreasing γ gives the later benefits of maintenance less weight.

Value iteration

Value iteration approximates $v^\star$. Start with $v^{(0)}(i) = 0$ in every state and use the completed value table to compute the next table.

$$v^{(k+1)}(i) = \max_{a \in \mathcal{A}(i)} \left\{ r(i,a) + \gamma \sum_j P_{ij}(a) v^{(k)}(j) \right\}.$$

Round	Shifts counted	What the value includes
0	None	Zero total in every state
1	$t = 0$	Best current reward, since $v^{(0)}(j) = 0$ for every next state j
2	$t = 0, 1$	Best expected total, with weight γ on the second shift's reward

For each action a, average the best k-shift values using its next-state probabilities. Discount this average by γ and add today's reward. Maximizing the total gives **$k+1$ shifts** of lookahead.

Round k counts shifts $t = 0, \ldots, k-1$, with zero value at the cutoff. We can compute these rounds before the machine operates.

Value iteration for the machine

Use $\gamma = \frac{1}{2}$. In each maximum, the first term is **Run** and the second is **Maintain**.

$$v^{(k+1)}(G) = \max\left\{6 + \frac{1}{4}v^{(k)}(G) + \frac{1}{4}v^{(k)}(W), \quad \frac{1}{2}v^{(k)}(G)\right\}.$$

$$v^{(k+1)}(W) = \max\left\{2 + \frac{1}{2}v^{(k)}(W), \quad \frac{1}{2}v^{(k)}(G)\right\}.$$

After running in G, each next condition has probability $\frac{1}{2}$, so its value gets weight $\gamma \times \frac{1}{2} = \frac{1}{4}$.

In the next update, choose Maintain at W if the gain in discounted continuation exceeds the 2 units sacrificed now.

$$\gamma\left[v^{(k)}(G) - v^{(k)}(W)\right] > 2.$$

Equality gives a tie.

The first update

Use $v^{(0)}(G) = v^{(0)}(W) = 0$. Only the current output remains in each comparison.

State	Run	Maintain	Value and action
G	$6 + 0 = 6$	$0 + 0 = 0$	$v^{(1)}(G) = 6$, Run
W	$2 + 0 = 2$	$0 + 0 = 0$	$v^{(1)}(W) = 2$, Run

With one shift counted, maintenance earns zero in either state. Running wins in both.

The second update

Use the **completed first value table**, $\left(v^{(1)}(G), v^{(1)}(W)\right) = (6, 2)$.

$$v^{(2)}(G) = \max\left\{6 + \frac{1}{4} \times 6 + \frac{1}{4} \times 2, \quad \frac{1}{2} \times 6\right\} = \max\{8, 3\} = 8.$$

$$v^{(2)}(W) = \max\left\{2 + \frac{1}{2} \times 2, \quad \frac{1}{2} \times 6\right\} = \max\{3, 3\} = 3.$$

At W, maintenance gains $\frac{1}{2}(6 - 2) = 2$ in future value and sacrifices 2 now. The two actions tie for this two-shift calculation.

Both state updates use $(6, 2)$, even if we calculate the good-state entry first.

A third update (exercise)

? Does a longer lookahead change the action?

Use $\left(v^{(2)}(G), v^{(2)}(W)\right) = (8, 3)$.

Calculate $v^{(3)}(G)$ and $v^{(3)}(W)$. Record the best action to take now in each state.

State	Run now	Maintain now
G	$6 + \frac{1}{4} \times 8 + \frac{1}{4} \times 3$	$\frac{1}{2} \times 8$
W	$2 + \frac{1}{2} \times 3$	$\frac{1}{2} \times 8$

The same continuation values must appear in both rows.

Maintenance becomes worthwhile

$$v^{(3)}(G) = \max\{8.75, 4\} = 8.75.$$

$$v^{(3)}(W) = \max\{3.5, 4\} = 4.$$

k	$v^{(k)}(G)$	$v^{(k)}(W)$	**Best action now in W**
0	0	0	None
1	6	2	Run
2	8	3	Either
3	8.75	4	Maintain
4	9.1875	4.375	Maintain

For $k \geq 1$, the lookahead starts with the current shift. In G, Run is best for each of these lookaheads.

In the third update, the future gain is $\frac{1}{2}(8 - 3) = 2.5 > 2$. Maintenance now recovers more future output than the 2 units it sacrifices.

Why value iteration converges

Convergence of value iteration — Theorem 7.3

For a finite MDP with $0 \le \gamma < 1$, value iteration converges from any finite initial table to the unique optimal value function $v^\star$. After $k \ge 1$ updates,

$$\max_i \left| v^{(k)}(i) - v^\star(i) \right| \le \gamma^k \max_i \left| v^{(0)}(i) - v^\star(i) \right|.$$

Discounting shrinks errors

The old table supplies the estimate of future rewards. Each update discounts that estimate by γ, so the influence of the starting table fades with repeated updates.

The machine earns between 0 and 6 units per shift. At $\gamma = \frac{1}{2}$, its discounted value is at most $\frac{6}{1-\gamma} = 12$. Starting from zero, ten updates give an error of at most $\frac{12}{1024} < 0.012$ units in either state.

Evaluating the maintenance policy

The latest value-iteration updates suggest the stationary policy

$$f(G) = \text{Run}, \quad f(W) = \text{Maintain}.$$

To test this candidate, first find its exact discounted value. Then use those values to compare all feasible actions.

? What does policy f achieve?

In state order (G, W), construct the transition matrix P^f and the reward vector r^f.

What discounted value does f achieve from each state? How could we verify its optimality?

A stationary policy induces a Markov reward process

In state order (G, W), select the transition row and reward for Run in G and Maintain in W.

State	Action	Selected transition row	Output
G	Run	$\left(\frac{1}{2}, \frac{1}{2}\right)$	6
W	Maintain	$(1, 0)$	0

$$P^f = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} \\ 1 & 0 \end{pmatrix}, \quad r^f = \begin{pmatrix} 6 \\ 0 \end{pmatrix}.$$

Evaluating a stationary policy

The prescribed actions determine the transition matrix P^f and reward vector r^f. We can now evaluate the resulting Markov reward process by first-step analysis.

Policy evaluation equations

Fix the stationary policy f. First-step analysis gives

$$v^f(i) = r(i, f(i)) + \gamma \sum_j P_{ij}(f(i)) v^f(j).$$

This follows the prescribed action, with no maximization. It combines reward now with discounted continuation under the same policy.

For the maintenance policy, a worn machine produces 0 and becomes good:

$$v^f(W) = \frac{1}{2} v^f(G).$$

A good machine produces 6 and may enter either condition:

$$v^f(G) = 6 + \frac{1}{4} v^f(G) + \frac{1}{4} v^f(W).$$

These are two linear equations. The first expresses $v^f(W)$ in terms of $v^f(G)$.

The maintenance policy's discounted value

Substitute $v^f(W) = \frac{1}{2} v^f(G)$:

$$v^f(G) = 6 + \frac{1}{4} v^f(G) + \frac{1}{8} v^f(G) = 6 + \frac{3}{8} v^f(G).$$

$$\frac{5}{8} v^f(G) = 6, \quad v^f(G) = \frac{48}{5} = 9.6.$$

$$v^f(W) = \frac{1}{2} \times 9.6 = 4.8.$$

These are the values of the prescribed policy. To test optimality, use them as continuation values for every feasible action.

Checking optimality by comparing actions

Repeated actions in value iteration suggest a policy. To verify it, evaluate the policy exactly, then compare all actions using its values $(9.6, 4.8)$.

State	Run now	Maintain now	Best action
G	$6 + \frac{1}{4}(9.6) + \frac{1}{4}(4.8) = 9.6$	$\frac{1}{2}(9.6) = 4.8$	Run
W	$2 + \frac{1}{2}(4.8) = 4.4$	$\frac{1}{2}(9.6) = 4.8$	Maintain

At W, maintenance gains $\frac{1}{2}(9.6 - 4.8) = 2.4$ in future value and sacrifices 2 now.

In each state, the prescribed action attains $v^f(i)$ and every alternative gives a smaller total. Thus v^f satisfies the discounted Bellman optimality equation.



The maintenance policy is optimal

The Bellman equation has a unique solution, so $v^f = v^\star = (9.6, 4.8)$.

Improving the always-run policy

The same comparison can improve a policy. Start with Always Run, $f_0(G) = f_0(W) = \text{Run}$. At $\gamma = \frac{1}{2}$ its values satisfy

$$v^{f_0}(W) = 2 + \frac{1}{2} v^{f_0}(W) \quad \Rightarrow \quad v^{f_0}(W) = 4,$$

$$v^{f_0}(G) = 6 + \frac{1}{4} v^{f_0}(G) + \frac{1}{4}(4) \quad \Rightarrow \quad \frac{3}{4} v^{f_0}(G) = 7 \quad \Rightarrow \quad v^{f_0}(G) = \frac{28}{3}.$$

Running now and then following Always Run is the original policy. What if we maintain just once?

State	Run now, then f_0	Maintain now, then f_0
G	$\frac{28}{3}$	$\frac{1}{2}\left(\frac{28}{3}\right) = \frac{14}{3}$
W	4	$\frac{1}{2}\left(\frac{28}{3}\right) = \frac{14}{3}$

Keep Run in G and switch to Maintain in W. From W, Always Run earns 4, and maintaining once before returning to Always Run earns $\frac{14}{3}$. Maintaining on every visit to W earns 4.8.

Policy iteration

Evaluate a deterministic stationary policy f exactly. Use its values to choose a new action in each state.

> **Policy improvement** **Proposition 7.2**
>
> In a finite discounted MDP, let $g(i)$ maximize $r(i,a) + \gamma \sum_j P_{ij}(a) v^f(j)$ over feasible actions a in each state i. Then $v^g(i) \geq v^f(i)$ for every i.

One shift under g followed by f is at least as good as f. At each next state the same argument lets us use g for one more shift without losing value. As the discounted tail vanishes, we obtain v^g.

Repeat with the improved policy. Keep the current action on ties. Each change strictly raises the value in at least one state. There are finitely many deterministic stationary policies, so iteration must stop.

When every action stays the same, the evaluated values satisfy the Bellman optimality equation. The policy is optimal, as we saw in the maintenance-policy comparison.

Review and Connections

The main ideas of an MDP

The central decision

An action earns reward now and changes where future decisions begin. Compare each action's immediate reward plus expected continuation value. Maintenance improves the next condition. Admission earns revenue but can leave a longer queue.

Concept	How we used it
State	Information for predicting rewards and next-state probabilities for each feasible action.
Policy	An action rule based on observations and, when relevant, time remaining.
Value	Expected reward from here under a given policy. An optimal policy maximizes it.
Bellman equation	Maximize immediate reward plus expected optimal continuation value, discounting when appropriate.

How the calculations fit together

Method	What do we calculate?
Backward induction	Optimal values and actions for each remaining horizon.
Value iteration	Successive approximations to optimal discounted values.
Policy evaluation	The value in each state under a prescribed policy.
Policy iteration	A sequence of policies obtained by evaluation and improvement.

Fixing a stationary policy gives a Markov reward process. Optimizing lets us choose which reward and transition row to use in each state.

? Have we established optimality?

Value iteration selects the same actions in two consecutive rounds. Does that establish optimality? What calculation would verify the candidate policy?

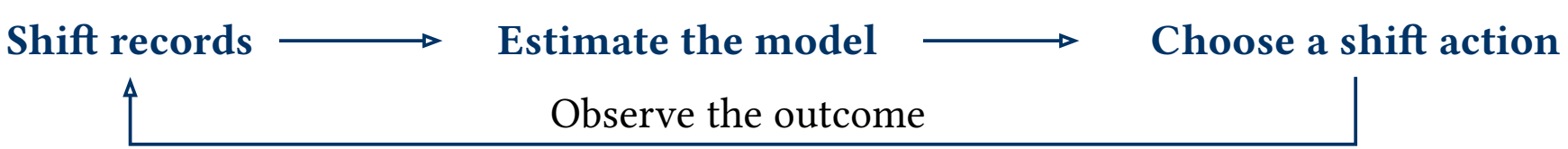
After discussion, revisit the worked verification.

What if maintenance success is unknown?

Previously, maintenance always succeeded. Now its success probability from W is fixed but unknown. Given this probability q, attempts are independent. Success leaves G and failure leaves W. Rewards, running transitions and $\gamma = \frac{1}{2}$ are as before.

Which shifts tell us about q? Record (condition, action, output, next condition). After at least one attempt from W, estimate

$$\hat{q} = \frac{\text{successful maintenance attempts from } W}{\text{maintenance attempts from } W}.$$



Model-based reinforcement learning uses the estimated probabilities $\hat{q}$ and $1 - \hat{q}$ to plan with the Bellman equation.

For fixed q, the current condition and action determine the next-state probabilities. Past observations help the operator estimate q and decide which action to take.

Actions also determine what we learn

? If we always run a worn machine, what can we learn about maintenance?

We never see a maintenance attempt from W, so every possible q fits those records.

Trying maintenance costs 2 units now, may restore the machine and gives one observation about q.

- **Exploitation.** Treat the current model estimate as correct and maximize expected discounted output. Maintenance can already be the best action.
- **Exploration.** Try an action because what its outcome teaches us could improve later decisions.



Planning and learning

An MDP describes the decision problem. Dynamic programming computes decisions from a given model. Reinforcement learning uses experience to learn good decisions. Our actions affect both future rewards and the experience available for learning.

Appendix

A Proofs ..... 481.
B Further DTMC Calculations ..... 488.
C Detailed Balance and Reversibility ..... 492.
D CTMC Constructions and Calculations ..... 497.
E Queueing Reference ..... 513.
F Markov Decision Process Reference ..... 526.

A Proofs

Proof of Example 2.15: Consider the social mobility chain from Example 2.14 with

$$P = \begin{bmatrix} 0.7 & 0.2 & 0.1 \\ 0.3 & 0.5 & 0.2 \\ 0.2 & 0.4 & 0.4 \end{bmatrix}.$$

A stationary distribution $\boldsymbol{\pi} = [\pi_1, \pi_2, \pi_3]$ (row vector convention) satisfies

$$\boldsymbol{\pi} = \boldsymbol{\pi} P, \quad \pi_1 + \pi_2 + \pi_3 = 1.$$

Step 1: expand $\boldsymbol{\pi} = \boldsymbol{\pi} P$ coordinate-wise. Since $\boldsymbol{\pi}_j = \sum_i \pi_i P_{ij}$ for each column j, we get the three balance equations

$$\begin{aligned} \pi_1 &= 0.7\pi_1 + 0.3\pi_2 + 0.2\pi_3, \\ \pi_2 &= 0.2\pi_1 + 0.5\pi_2 + 0.4\pi_3, \\ \pi_3 &= 0.1\pi_1 + 0.2\pi_2 + 0.4\pi_3. \end{aligned}$$

Step 2: move everything to the left to obtain a homogeneous linear system.

$$
\begin{aligned}
(1-0.7)\pi_1 - 0.3\pi_2 - 0.2\pi_3 &= 0,\\
-0.2\pi_1 + (1-0.5)\pi_2 - 0.4\pi_3 &= 0,\\
-0.1\pi_1 - 0.2\pi_2 + (1-0.4)\pi_3 &= 0.
\end{aligned}
$$

Equivalently (clearing decimals by multiplying by 10),

$$
\begin{aligned}
3\pi_1 - 3\pi_2 - 2\pi_3 &= 0,\\
-2\pi_1 + 5\pi_2 - 4\pi_3 &= 0,\\
-\pi_1 - 2\pi_2 + 6\pi_3 &= 0.
\end{aligned}
$$

Note that, these three equations are not independent; we will use two of them plus normalization.

Step 3: solve by elimination.

From the first equation

$$
3\pi_1 = 3\pi_2 + 2\pi_3 \quad \Rightarrow \quad \pi_1 = \pi_2 + \left(\frac{2}{3}\right)\pi_3.
$$

Plug this into the second equation $-2\pi_1 + 5\pi_2 - 4\pi_3 = 0$:

$$-2\left(\pi_2+\left(\frac{2}{3}\right)\pi_3\right)+5\pi_2-4\pi_3=0$$

so

$$(-2\pi_2+5\pi_2)+\left(-\left(\frac{4}{3}\right)\pi_3-4\pi_3\right)=0 \quad \Rightarrow \quad 3\pi_2-\left(\frac{16}{3}\right)\pi_3=0$$

hence

$$\pi_2=\left(\frac{16}{9}\right)\pi_3.$$

Then

$$\pi_1=\pi_2+\left(\frac{2}{3}\right)\pi_3=\left(\frac{16}{9}\right)\pi_3+\left(\frac{2}{3}\right)\pi_3=\left(\frac{16}{9}+\frac{6}{9}\right)\pi_3=\left(\frac{22}{9}\right)\pi_3.$$

Step 4: enforce normalization $\pi_1+\pi_2+\pi_3=1$:

$$\left(\frac{22}{9}\right)\pi_3 + \left(\frac{16}{9}\right)\pi_3 + \pi_3 = 1 \quad \Rightarrow \quad \left(\frac{22}{9} + \frac{16}{9} + \frac{9}{9}\right)\pi_3 = 1 \quad \Rightarrow \quad \left(\frac{47}{9}\right)\pi_3 = 1,$$

so

$$\pi_3 = \frac{9}{47}.$$

Finally,

$$\pi_2 = \left(\frac{16}{9}\right)\left(\frac{9}{47}\right) = \frac{16}{47}, q \quad \pi_1 = \left(\frac{22}{9}\right)\left(\frac{9}{47}\right) = \frac{22}{47}.$$

Therefore the stationary distribution is

$$\boldsymbol{\pi} = [\pi_1, \pi_2, \pi_3] = \left[\frac{22}{47}, \frac{16}{47}, \frac{9}{47}\right].$$

∎

Proof of Example 2.13:

Let $0 < p < \frac{1}{2}$ and $q = 1 - p$. Under $\mathbb{P}_p$, up/down steps have probabilities p, q. Under $\mathbb{P}_q$, they have probabilities q, p. Both walks start at 0.

Fix a path $\omega = (0, x_1, \ldots, x_n)$ that **first reaches a at time n**. Its u up steps and d down steps satisfy $u - d = a$. By independence,

$$\frac{\mathbb{P}_p(\omega)}{\mathbb{P}_q(\omega)} = \frac{p^u q^d}{q^u p^d} = \left(\frac{p}{q}\right)^u \left(\frac{q}{p}\right)^d$$
$$= \left(\frac{p}{q}\right)^u \left(\frac{p}{q}\right)^{-d} = \left(\frac{p}{q}\right)^{u-d} = \left(\frac{p}{q}\right)^a.$$

This ratio is constant over all such paths. Summing over paths and first-hit times gives

$$\mathbb{P}_p(\tau_a < \infty) = \left(\frac{p}{q}\right)^a \mathbb{P}_q(\tau_a < \infty) = \left(\frac{p}{q}\right)^a.$$

Since $q > \frac{1}{2}$, the upward-drift result gives $\mathbb{P}_q(\tau_a < \infty) = 1$. ∎

Proof of Proposition 3.2:

Choose disjoint intervals $I_1, \cdots, I_n \subset (0, T)$ in chronological order, with lengths $\ell_1, \cdots, \ell_n$.

Given $N(T) = n$, the event $T_i \in I_i$ for every i requires **one arrival in each interval and none elsewhere**. Independent increments give

$$\begin{aligned}\mathbb{P}(T_i \in I_i \text{ for every } i \mid N(T) = n) &= \frac{\exp(-\lambda T)\prod_{i=1}^{n}(\lambda \ell_i)}{\exp(-\lambda T)\frac{(\lambda T)^n}{n!}} \\ &= n! \prod_{i=1}^{n}\left(\frac{\ell_i}{T}\right).\end{aligned}$$

For n independent $\text{Uniform}(0, T)$ points, each assignment of points to intervals has probability $\prod_{i=1}^{n}\left(\frac{\ell_i}{T}\right)$. There are $n!$ such assignments, so sorting gives the same interval probabilities.

These interval probabilities determine the joint distribution. ∎

Proof of Lemma 4.1:

1. Positivity. By right-continuity and $G(0) = 1$, $G\left(\frac{1}{n}\right) > 0$ for all sufficiently large integers n. Multiplicativity gives $G(1) = G\left(\frac{1}{n}\right)^n > 0$. Since $G(1) \leq 1$, set $r = -\ln(G(1)) \geq 0$; this rate is finite.

2. Rational times. For every integer $n \geq 1$, $G\left(\frac{1}{n}\right)^n = G(1) = e^{-r}$. Taking the nonnegative nth root and then repeated products gives, for every integer $m \geq 0$,

$$G\left(\frac{m}{n}\right) = G\left(\frac{1}{n}\right)^m = e^{-r\frac{m}{n}}.$$

3. All times and uniqueness. For any $t \geq 0$, choose nonnegative rational numbers a_k decreasing to t. Right-continuity of G and continuity of the exponential give

$$G(t) = \lim_{k\to\infty} G(a_k) = \lim_{k\to\infty} e^{-ra_k} = e^{-rt}.$$

The rate is unique because evaluating at $t = 1$ forces $r = -\ln(G(1))$. ∎

B Further DTMC Calculations

Optional reading: formulas that extend the first-step reasoning from Section 2.7.

When the gambler has an advantage

In Example 2.4, replace target 4 by N. Win 1 with probability p and lose 1 with probability $q = 1 - p$, where $0 < p < 1$.

For $h_i = \mathbb{P}_i(\tau_N < \tau_0)$, the same equations give

$$h_i = \begin{cases} \frac{i}{N} & \text{if } p = q = \frac{1}{2} \\ \frac{1-\left(\frac{q}{p}\right)^i}{1-\left(\frac{q}{p}\right)^N} & \text{if } p \neq q. \end{cases}$$

Check the direction ($0 < i < N$): a fair game gives $\frac{i}{N}$. Upward drift ($p > \frac{1}{2}$) raises the chance above $\frac{i}{N}$; downward drift lowers it.

Absorption: add up expected visits

Consider a **finite** chain with transient states followed by absorbing states. Assume absorption occurs with probability 1 from every transient state.

$$P = \begin{bmatrix} Q & R \\ \mathbf{0} & I \end{bmatrix}.$$

Q records transitions among transient states; R records transitions into absorbing states.

Fundamental matrix

$$F = (I - Q)^{-1} = I + Q + Q^2 + \cdots.$$

F_{ij} is the expected number of visits to transient state j before absorption, starting from i. The I term counts the visit at **time 0**.

The powers of Q track visits at times $1, 2, \ldots$; summing them counts all visits before stopping.

The machine: one failure and repair cycle

Reuse failure probability 0.2 and repair probability 0.4. **Stop at the first completed repair.** Use states U (before the first failure), D (awaiting repair), and absorbing A (repair complete).

$$Q = \begin{bmatrix} 0.8 & 0.2 \\ 0 & 0.6 \end{bmatrix}, \quad R = \begin{bmatrix} 0 \\ 0.4 \end{bmatrix}, \quad F = \begin{bmatrix} 5 & 2.5 \\ 0 & 2.5 \end{bmatrix}.$$

Adding expected visits gives expected time until absorption:

$$\boldsymbol{t} = F\mathbf{1} = \begin{bmatrix} 7.5 \\ 2.5 \end{bmatrix}.$$

Starting Up: **5 days until failure + 2.5 days until repair = 7.5 days.** Starting Down: only the repair wait remains.

Multiplying by the probabilities of leaving for A gives

$$B = FR = \begin{bmatrix} 1 \\ 1 \end{bmatrix} : \quad \text{repair eventually completes from either start.}$$

Reward over a fixed number of days

The forecast $\boldsymbol{\nu}^{(0)} P^k$ from Section 2.4 tells us where the chain is at time k. Attach reward r_i to a period spent in state i, with column vector $\boldsymbol{r}$.

For rewards counted at times $0, 1, \ldots, n-1$,

$$G_n = \sum_{k=0}^{n-1} r_{X_k}, \quad \mathbb{E}[G_n] = \sum_{k=0}^{n-1} \boldsymbol{\nu}^{(0)} P^k \boldsymbol{r}.$$

Machine example. Start Up on day 0; earn \$1200 when Up and $-$\$300 when Down. For **days 0 and 1**,

$$\mathbb{E}[G_2] = 1200 + (0.8 \cdot 1200 + 0.2 \cdot (-300)) = \$2100.$$

Each day's forecast gives that day's expected reward. Add across the chosen horizon. For long-run performance, the stationary weights give the average reward per day.

C Detailed Balance and Reversibility

Pairwise balance implies total balance

Continue from Page 130. Let $\boldsymbol{\pi}$ be a probability vector: $\pi_i \geq 0$ and $\sum_i \pi_i = 1$.

Stationarity	$\pi_j = \sum_i \pi_i P_{ij}$ for every j.
Global balance	$\sum_{i \neq j} \pi_i P_{ij} = \sum_{i \neq j} \pi_j P_{ji}$ for every j.
Detailed balance	$\pi_i P_{ij} = \pi_j P_{ji}$ for every pair i, j.

Global balance is stationarity with the self-loop removed. Detailed balance matches each opposing pair, so summing gives

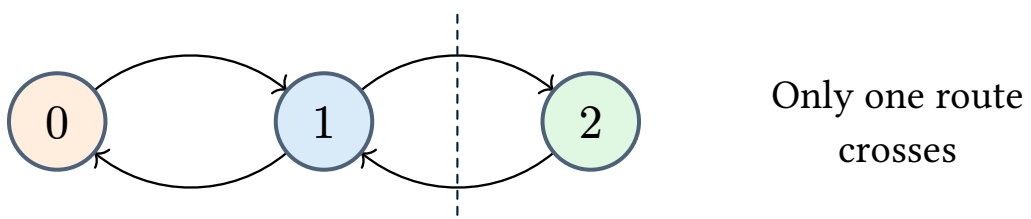
$$\sum_i \pi_i P_{ij} = \sum_i \pi_j P_{ji} = \pi_j \underbrace{\sum_i P_{ji}}_{\text{row j sums to 1}} = \pi_j.$$

Detailed balance $\Rightarrow$ **global balance** $\Leftrightarrow$ **stationarity**.

A single route forces pairwise balance

The machine ($\alpha, \beta > 0$): the only route out of Up leads to Down. Global balance gives $\pi_U \alpha = \pi_D \beta$, which is already detailed balance. The **weighted flows** match even if $\alpha \neq \beta$.

The same idea on a line: consider a finite irreducible chain that moves only between neighbors, with self-loops allowed.



At stationarity, the total probability on either side of the cut stays fixed. Only one route crosses, so its flows must match. Apply this at every cut:

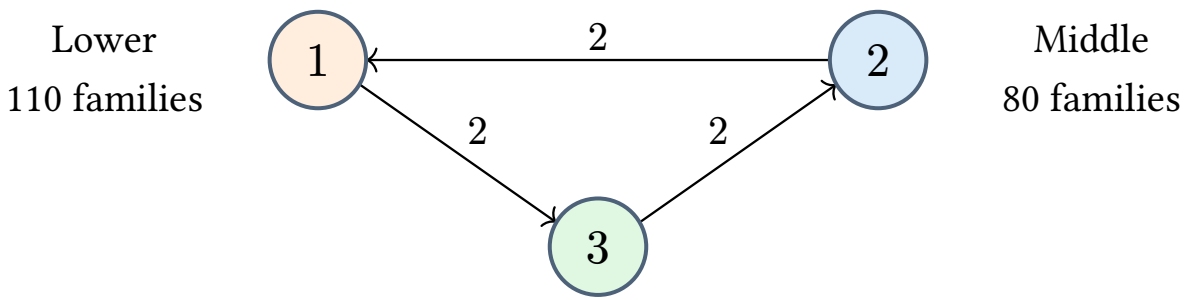
$$\pi_k P_{k,k+1} = \pi_{k+1} P_{k+1,k}.$$

Use these equations to find all weights, then normalize them to sum to 1.

Stationary populations can still circulate

Return to the 235 families on Page 130. Subtract the opposing expected moves on each pair:

$$1 \to 3: \quad 11 - 9 = 2; \quad 3 \to 2: \quad 18 - 16 = 2; \quad 2 \to 1: \quad 24 - 22 = 2.$$



These arrows show **net expected flow**: 2 families per generation around $1 \to 3 \to 2 \to 1$. Every class receives 2 and loses 2 after the opposing flows cancel.

Global balance keeps the class proportions fixed. **Detailed balance** would make every net arrow zero, although individual families could still travel around cycles.

Detailed balance means time reversibility

Imagine recording the chain **in stationarity**. The chance of seeing $i \to j$ is $\pi_i P_{ij}$. Detailed balance makes it equal to the chance $\pi_j P_{ji}$ of seeing the reversed pair.

For a finite irreducible chain, $\pi_i > 0$. Bayes' rule gives the backward transition probability:

$$P^{\mathrm{rev}}_{ij} = \mathbb{P}(X_{n-1} = j \mid X_n = i) = \frac{\pi_j P_{ji}}{\pi_i}.$$

Thus **detailed balance holds exactly when** $P^{\mathrm{rev}} = P$. This property is called **reversibility**: every finite sequence has the same probability as its reversal.

The social-mobility circulation changes direction when the recording is reversed.

Reversibility does not guarantee convergence. The alternating machine is reversible at $\boldsymbol{\pi} = \left[\frac{1}{2}, \frac{1}{2}\right]$ but periodic.

A cycle test for detailed balance

For a **finite irreducible chain**, detailed balance is possible exactly when the product of transition probabilities around **every closed cycle** equals the product in reverse.

For a three-state cycle, the test is

$$P_{12}P_{23}P_{31} = P_{13}P_{32}P_{21}.$$

Why necessary? Multiply the detailed-balance equations around the cycle. The positive π_i factors cancel. The full cycle criterion is also sufficient.

In our social-mobility model,

$$0.2 \times 0.2 \times 0.2 = 0.008 \quad \neq \quad 0.1 \times 0.4 \times 0.3 = 0.012.$$

One failed cycle rules out detailed balance, even though every arrow has a reverse arrow.

For calculations: solve pairwise equations and normalize when they are consistent. Check every pair. If they conflict, use global balance $\boldsymbol{\pi}P = \boldsymbol{\pi}$ instead.

Further reading: Levin and Peres, §1.6 Kelly, §1.5.

D CTMC Constructions and Calculations

Two equivalent CTMC constructions

Start in state i with total outgoing rate $q_i > 0$. Let H be the time until the next change.

Independent outgoing clocks

1. For each $j \neq i$ with $q_{ij} > 0$, independently draw $E_j \sim \mathrm{Exp}(q_{ij})$.
2. Wait $H = \min_j E_j$.
3. Jump to the destination whose clock rings first.

One holding time and one destination

1. Draw $H \sim \mathrm{Exp}(q_i)$.
2. Independently choose $j \neq i$ with probability $\frac{q_{ij}}{q_i}$.
3. After time H, jump to j.

The finite-clock case follows from Lemma 4.2. For $j \neq i$, both give the same joint law conditional on starting in state i:

$$\mathbb{P}(H > t, \text{next state} = j) = e^{-q_i t} \frac{q_{ij}}{q_i}.$$

Repeat using the new state's rates. If $q_i = 0$, stay there forever. Memorylessness allows surviving clocks with unchanged rates to be redrawn.

The embedded jump chain and holding rates

Let T_n be the nth state-change time, with $T_0 = 0$, and let $Y_n = X(T_n)$. The visited states $Y_0, Y_1, \cdots$ form the **embedded jump chain**, a DTMC.

At a nonabsorbing state i, its transition probabilities are

$$R_{ij} = \frac{q_{ij}}{q_i} \quad (j \neq i), \quad R_{ii} = 0.$$

For the two-machine count process, in state order $0, 1, 2$,

$$R = \begin{pmatrix} 0 & 1 & 0 \\ \frac{\alpha}{\alpha+\beta} & 0 & \frac{\beta}{\alpha+\beta} \\ 0 & 1 & 0 \end{pmatrix}.$$

Holding rates and destination probabilities recover the generator:

$$q_{ij} = q_i R_{ij} \quad (j \neq i), \quad q_{ii} = -q_i.$$

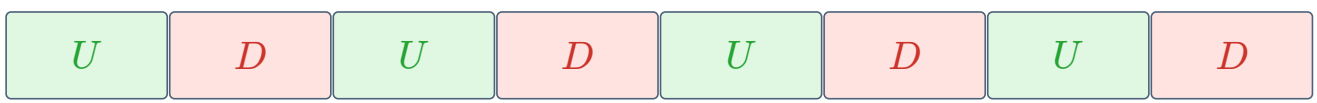
An absorbing state has no further jumps; its jump-chain row may be completed with $R_{ii} = 1$.

Visit frequencies and time proportions

For one Up/Down machine, the jump chain alternates $U, D, U, D, \cdots$. Use $\alpha = 0.2$, $\beta = 0.4$ per day.

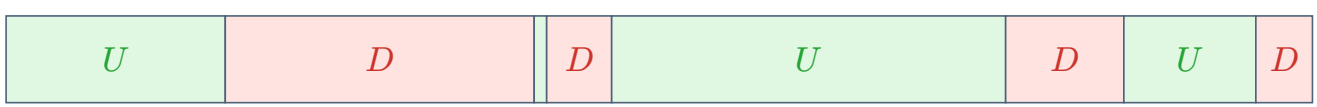
Same eight completed visits

Equal visit weights



Up visits: 4/8 = 50%

Actual holding durations



Up time: 16.77/28.91 = 58.0%

A simulated sample; long-run Up time approaches 2/3.

Both rows show the same visits: the upper row counts each once; the lower row retains its duration. Longer visits contribute more to the time proportion defined on Page 273.

Interactive supplement: **ctmc-visits-time.html**.

Time proportions from Up/Down cycles

An Up/Down cycle consists of one Up visit followed by one Down visit.

State	Visits per cycle	Mean duration
Up	1	$\frac{1}{\alpha} = 5$ days
Down	1	$\frac{1}{\beta} = 2.5$ days

Half the visits are Up, but Up visits last longer on average. Over repeated independent cycles, the time fraction approaches mean Up time divided by mean cycle duration:

$$\text{long-run Up-time proportion} = \frac{\frac{1}{\alpha}}{\frac{1}{\alpha} + \frac{1}{\beta}} = \frac{2}{3}.$$

This is the same duration-weighting idea as the bus example: clock-time observations give longer visits more chance to be observed.

Stationary distributions of the CTMC and its jump chain

For a finite irreducible CTMC with at least two states, let $\boldsymbol{\pi}^J$ be the stationary distribution of its embedded jump chain, with visit frequencies π_i^J.

$$\pi_i = \frac{\frac{\pi_i^J}{q_i}}{\sum_j \frac{\pi_j^J}{q_j}}.$$

Each visit to state i contributes mean duration $\frac{1}{q_i}$. Weight visit frequencies by these durations, then normalize.

$$\pi_i^J = \frac{\pi_i q_i}{\sum_j \pi_j q_j}.$$

Periodicity of the jump chain does not prevent convergence of its empirical visit frequencies.

Transient probabilities for one machine

One machine fails at rate α and is repaired at rate β. Let $\nu_U(t) = \mathbb{P}(X(t) = U)$ and $\nu_D(t) = 1 - \nu_U(t)$.

The Up probability changes at repair inflow rate minus failure outflow rate:

$$\nu_U'(t) = \beta\nu_D(t) - \alpha\nu_U(t) = \beta - (\alpha + \beta)\nu_U(t).$$

The flow rates balance at $\nu_U = \frac{\beta}{\alpha+\beta}$. Rewrite the equation as

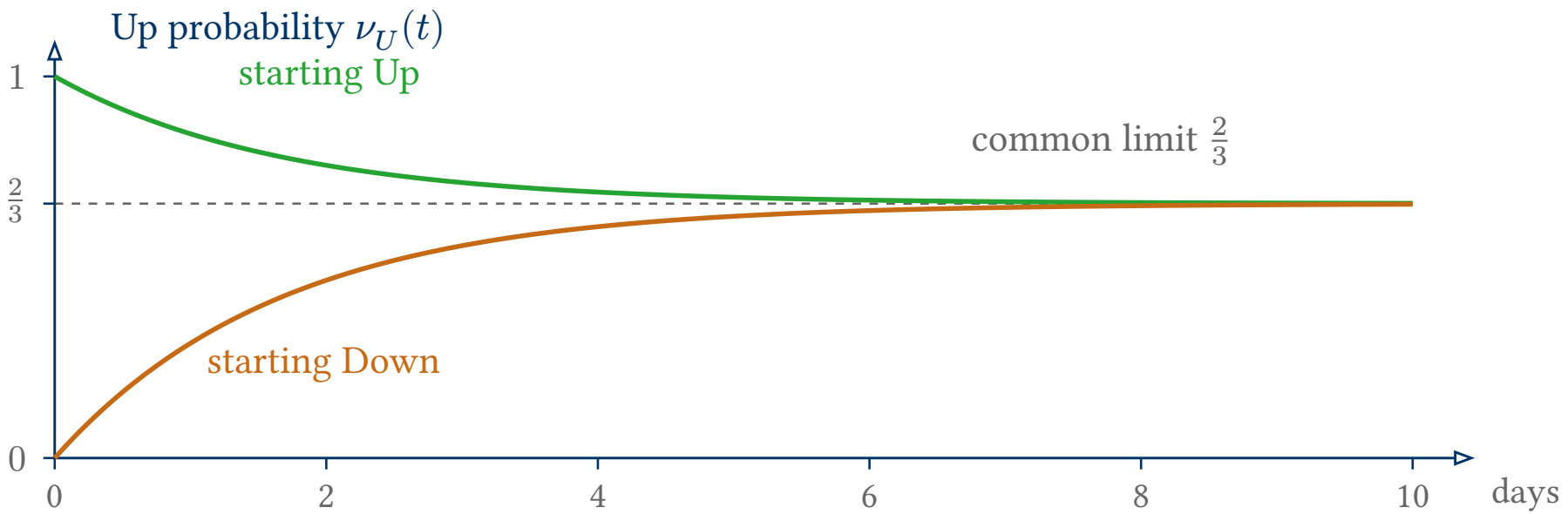
$$\nu_U'(t) = -(\alpha + \beta)\left[\nu_U(t) - \frac{\beta}{\alpha + \beta}\right].$$

The gap from balance decays exponentially. Its initial value is $\nu_U(0) - \frac{\beta}{\alpha+\beta}$, so

$$\nu_U(t) = \frac{\beta}{\alpha + \beta} + \left[\nu_U(0) - \frac{\beta}{\alpha + \beta}\right]e^{-(\alpha+\beta)t}.$$

Convergence from different initial states

With $\alpha = 0.2$ and $\beta = 0.4$ per day, the balanced Up probability is $\frac{2}{3}$.



For the machine that **starts Up**, the one-day forecast is

$$\nu_U(1) = \frac{2}{3} + \frac{1}{3}e^{-0.6} \approx 0.8496, \quad \nu_D(1) \approx 0.1504.$$

The matrix exponential and fixed-interval sampling

For a finite-state CTMC, the transition matrix is

$$P(t) = e^{tQ} = I + tQ + \frac{(tQ)^2}{2!} + \frac{(tQ)^3}{3!} + \cdots.$$

For a fixed interval Δ, the sampled sequence $X(0), X(\Delta), X(2\Delta), \cdots$ is a DTMC with transition matrix $P(\Delta)$.

For larger models, a numerical matrix exponential or an ODE solver provides transient forecasts. The reservoir equations describe probability flow rates between states.

Detailed balance: an optional sufficient condition

For a finite-state CTMC, a probability row vector $\boldsymbol{\pi}$ satisfies **detailed balance** if

$$\pi_i q_{ij} = \pi_j q_{ji} \quad (i \neq j).$$

These equations balance probability flow rates separately between every pair of states. Summing over $i \neq j$ gives global balance at state j, hence $\boldsymbol{\pi} Q = \mathbf{0}$.

The two-machine calculation on Page 272 satisfies detailed balance:

$$\beta \pi_0 = \alpha \pi_1, \quad \beta \pi_1 = 2\alpha \pi_2.$$

There are no transitions between 0 and 2, so their two flow rates are both zero.



Sufficient, but not necessary

Detailed balance implies stationarity. A stationary CTMC need not satisfy detailed balance. The general condition is global balance, as on Page 270.

Stationary probability flow can circulate

On states $0, 1, 2$, consider a clockwise cycle with rate one on each edge:

$$Q = \begin{pmatrix} -1 & 1 & 0 \\ 0 & -1 & 1 \\ 1 & 0 & -1 \end{pmatrix}, \quad \boldsymbol{\pi} = \left(\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right).$$

Each state's probability inflow and outflow rates are $\frac{1}{3}$ per unit time, so $\boldsymbol{\pi} Q = \mathbf{0}$. This is a stationary distribution.

However, the opposing probability flow rates between states 0 and 1 are

$$\pi_0 q_{0,1} = \frac{1}{3}, \quad \pi_1 q_{1,0} = 0.$$

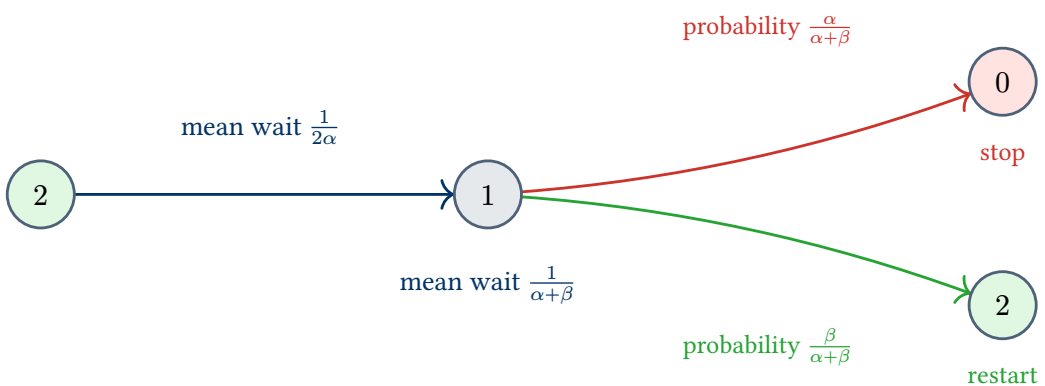
Thus detailed balance fails, even though global balance holds at every state.

How long until both machines are Down?

Two machines fail independently at rate α each. One technician repairs at rate β. Failure and repair times are independent exponential variables.

Suppose **both machines are working today**. How long, on average, until the first complete outage?

Let m_i be the mean time to first reach 0, starting with i machines Up. We stop on reaching 0, so $\boldsymbol{m_0 = 0}$.



Repairs can return the system to state 2, so an outage may require repeated attempts.

First-event equations for the outage time

Starting in state 2: the first failure takes mean time $\frac{1}{2\alpha}$. One machine then remains, with expected time m_1 still to go.

$$m_2 = \frac{1}{2\alpha} + m_1.$$

Starting in state 1: wait a mean $\frac{1}{\alpha+\beta}$ days for a failure or repair. A failure reaches the target. A repair restarts the problem from state 2.

$$m_1 = \frac{1}{\alpha+\beta} + \frac{\alpha}{\alpha+\beta}\underbrace{m_0}_{=0} + \frac{\beta}{\alpha+\beta}m_2.$$

First-event conditioning

Expected time = mean time to the next change + expected time remaining after that change.

This is the continuous-time version of DTMC first-step analysis (Proposition 2.5).

The expected time to the first outage

With failure rate $\alpha = 0.2$ per working machine and repair rate $\beta = 0.4$ per day,

$$m_2 = 2.5 + m_1, \quad m_1 = \frac{5}{3} + \frac{2}{3} m_2.$$

Substitution gives

$$m_2 = 2.5 + \frac{5}{3} + \frac{2}{3} m_2 \quad \rightarrow \quad m_2 = 12.5 \ \text{ days}, \quad m_1 = 10 \ \text{ days}.$$

Engineering question	Answer
Fraction of time at least one machine works	$\frac{4}{5}$
Mean time to the first outage, starting with both Up	12.5 days

Repairs prolong the time to the first outage by returning the system to state 2. The answer depends on the starting state and on when we stop observing.

General target-set equations follow on Page 510.

First-event equations for a target set

For a **finite irreducible CTMC** and a nonempty target set A, define

$$\tau_A = \inf\{t \geq 0 : X(t) \in A\}, \quad m_i = \mathbb{E}_{i[\tau_A]}.$$

These means are finite, with $m_i = 0$ on A. Outside A, condition on the next change:

$$m_i = \frac{1}{q_i} + \sum_{j \neq i} \frac{q_{ij}}{q_i} m_j \quad (i \notin A).$$

The first term is the mean holding time. The sum averages the remaining time over the next destination. For the **column vector $\boldsymbol{m}$**, this is

$$(Q\boldsymbol{m})_i = -1 \quad (i \notin A), \quad m_i = 0 \quad (i \in A).$$

Running cost: replace $\frac{1}{q_i}$ by $\frac{r_i}{q_i}$ for cost rate r_i, with zero cost remaining at A.

Hit A before B: for disjoint nonempty targets, use contribution 0 and boundaries 1 on A, 0 on B.

Birth-and-death stationary weights

On $\{0, 1, 2, \cdots\}$, assume positive neighbor rates, $\mu_0 = 0$, and **non-explosion**. The equations derived on Page 281 are the **detailed balance equations** for this process. They are equivalent to global balance here because only one edge crosses each finite-set boundary $\{0, \cdots, n\}$.

Write $\pi_n = \pi_0 w_n$. Repeating the probability recursion (Page 283) gives the coefficients:

$$w_0 = 1, \quad w_n = \prod_{k=0}^{n-1} \frac{\lambda_k}{\mu_{k+1}} \quad (n \geq 1).$$

Let $Z = \sum_{k=0}^{\infty} w_k$. A stationary probability distribution exists exactly when $Z < \infty$:

$$\pi_n = \frac{w_n}{Z} \quad (n \geq 0).$$

Finite-capacity variant: on $\{0, 1, \cdots, K\}$, set $\lambda_K = 0$ and keep positive rates between neighbors. Use the same weights through K and normalize by $Z_K = \sum_{k=0}^{K} w_k$. This finite sum always gives $\pi_n = \frac{w_n}{Z_K}$ for $0 \leq n \leq K$.

Further reading and modeling assumptions

- Karl Sigman, **Continuous-Time Markov Chains**, Columbia IEOR 6711 (2009). Sections 1.1–1.6 cover rates, the embedded chain, and non-explosion. Section 1.13 gives birth-and-death stationary laws.
- Robert Wolpert, **Birth and Death**, Duke University (2016). Section 3.1 derives the Poisson stationary law for immigration and linear death.
- Gonzalo Mateos, **Continuous-time Markov Chains**, Rochester ECE 440. Slides 48–50 connect birth-and-death steady state to a single-server queue.

Read alongside the worked examples. Finite models assume time homogeneity and finite transition rates. Irreducibility gives a common long-run law. Expected-time and accumulated-cost equations require the stated target-hitting assumption.

For infinite-state models, check non-explosion and the existence of a normalizable stationary distribution separately.

For general durations, revisit state sufficiency: elapsed age may carry predictive information.

E Queueing Reference

Sojourn distribution: an alternative derivation

We can also derive the sojourn result on Page 303 by summing conditional tails. For a stationary FCFS M/M/1 queue, PASTA gives $\mathbb{P}(N=n)=(1-\rho)\rho^n$, where $\rho=\frac{\lambda}{\mu}<1$. Given $N=n$, W is the sum of $n+1$ independent $\mathrm{Exp}(\mu)$ stages.

$$\mathbb{P}(W>t \mid N=n)=e^{-\mu t}\sum_{k=0}^{n}\frac{(\mu t)^k}{k!}.$$

Interchange nonnegative summands and use the geometric tail:

$$\begin{aligned}\mathbb{P}(W>t)&=e^{-\mu t}\sum_{k=0}^{\infty}\frac{(\mu t)^k}{k!}\sum_{n=k}^{\infty}(1-\rho)\rho^n\\&=e^{-\mu t}\sum_{k=0}^{\infty}\frac{(\rho\mu t)^k}{k!}=e^{-(\mu-\lambda)t}.\end{aligned}$$

Waiting distribution: an alternative derivation

The waiting-time result on Page 304 follows from the same summation:

$$\mathbb{P}(W_q = 0) = 1 - \rho, \quad \mathbb{P}(W_q > t) = \rho e^{-(\mu-\lambda)t}, \quad t \geq 0.$$

If $N = 0$, service begins immediately. If $N = n \geq 1$, waiting requires n exponential stages. The same summation as for W now starts at $n = k + 1$.

For $0 < \rho < 1$, conditional on having to wait,

$$\mathbb{P}(W_q > t \mid W_q > 0) = e^{-(\mu-\lambda)t}, \quad \mathbb{E}[W_q \mid W_q > 0] = \frac{1}{\mu - \lambda}.$$

At the gateway, 20% do not wait; among those who wait, the mean wait is 20 seconds. Across all arrivals, the mean wait is $0.8 \times 20 = 16$ seconds.

Poisson output needs a theorem

Rate conservation gives the **mean departure rate**. It does not identify the departure process. Even exponential marginal interdeparture times do not establish independent gaps.

Burke's theorem: M/M/1 output

For an unlimited, stationary M/M/1 queue with Poisson rate $\lambda < \mu$ and independent exponential service, the departure process is a **Poisson process of rate λ**.

For a tandem of two such stations, with independent service sequences and no feedback, the output of the first supplies Poisson input to the second. Check $\lambda < \mu_i$ at both stations.

This is a statement about **stationary output**. Starting an empty queue at time zero does not immediately give stationary Poisson departures.

Single-station formulas and their conditions

All models below are stationary and FCFS, with the independence assumptions stated in Chapter 5. Mean system time includes mean service time: $w = w_q + \mathbb{E}[S]$.

Model	Parameters and stability	Mean waiting time
M/M/1	$\rho = \frac{\lambda}{\mu} < 1$	$w_q = \frac{\rho}{\mu - \lambda}$
M/M/c	$\rho = \frac{\lambda}{c\mu} < 1$	$w_q = \frac{p_{\text{wait}}}{c\mu - \lambda}$
M/G/1	$\rho = \lambda \mathbb{E}[S] < 1$; finite $\mathbb{E}[S^2]$	$w_q = \frac{\lambda \mathbb{E}[S^2]}{2(1-\rho)}$

For M/M/c, define $a = \frac{\lambda}{\mu}$ and $p_{\text{wait}} = \mathbb{P}(W_q > 0) = \sum_{n=c}^{\infty} \pi_n$. Compute

$$\pi_0 = \left(\sum_{n=0}^{c-1} \frac{a^n}{n!} + \frac{a^c}{c!(1-\rho)} \right)^{-1}, \quad p_{\text{wait}} = \pi_0 \frac{a^c}{c!(1-\rho)}.$$

For these lossless models, $\ell_q = \lambda w_q$ and $\ell = \lambda w$.

Finite admission and admitted-customer time

For **M/M/1/K**, K includes service. Set $a = \frac{\lambda}{\mu}$; this offered load may exceed one.

$$\pi_n = \frac{a^n}{\sum_{j=0}^{K} a^j}, \quad n = 0, \cdots, K, \quad \lambda_{\text{eff}} = \lambda(1 - \pi_K).$$

$$\ell = \sum_{n=0}^{K} n\pi_n, \quad w = \frac{\ell}{\lambda_{\text{eff}}}, \quad w_q = w - \frac{1}{\mu}.$$

The rejection probability is π_K by PASTA for offered Poisson arrivals. Actual utilization is $\frac{\lambda_{\text{eff}}}{\mu} = 1 - \pi_0$.

The population counts admitted requests in the system; the time average concerns those same requests. Offered arrivals that are rejected contribute to neither population nor system time.

Worked example: which cashier should we hire?

Apply the customer-time cost rule to a shop receiving **25 customers/hour**. Model each choice as a stationary FCFS M/M/1 queue with **unlimited waiting space and everyone admitted**:

Quantity	Cashier A	Cashier B
Mean service time	1 minute	2 minutes
Hourly wage	\$10/hour	\$5/hour

Customer time costs **\$0.02 per customer-minute**, including service. With mean population ℓ, one hour contains **60ℓ customer-minutes on average**. The customer-time cost is $0.02 \times 60\ell = 1.20\ell$ dollars/hour. Adding the wage gives the total hourly cost:

$$\text{wage per hour} + 1.20\ell.$$

Both cashiers can keep up with average demand. **Does the cheaper cashier minimize total cost?**

Worked solution: a lower wage can cost more overall

Quantity	Cashier A	Cashier B
Service rate μ per hour	60	30
Mean system time $w = \frac{1}{\mu - 25}$	$\frac{1}{35}$ h ≈ 1.71 min	$\frac{1}{5}$ h $= 12$ min
Mean number $\ell = 25w$	$\frac{5}{7}$	5
Customer-time cost per hour	\$0.86	\$6.00
Wage plus customer-time cost	**\$10.86/hour**	**\$11.00/hour**

Under the stated cost assumptions, **choose A**, narrowly. The additional wage is offset by the reduction in customer-time cost.

The M/M/1 mean-time theorem predicts w; Theorem 5.4 gives $\ell = 25w$ with w in hours. Return to the customer-time cost rule for the general accounting argument.

One fast processor or two slower processors?

Now allow the resource design to change. Keep demand at 24/min and total service capacity at 30/min. Use exponential services and a shared FCFS queue in both designs.

Measure	One at 30/min	Two at 15/min each
Mean service duration	2 s	4 s
Mean waiting time w_q	8 s	7.11 s
Mean system time w	10 s	11.11 s

Two slower processors give a shorter mean **wait to start**. One fast processor gives a shorter mean **time to finish**.

Resource pooling keeps the same two processors. This comparison changes individual service speeds. **State the service objective before declaring one design better.**

From the joint state to waiting time

The states $(2,0)$ and $(1,1)$ have the same total count, but completion rates μ and 2μ. The total count alone is **not a birth-and-death state**.

Compute $\pi_{ij} = \mathbb{P}(L_1 = i, L_2 = j)$ numerically from $\boldsymbol{\pi} Q = 0$ with $\sum_{i,j\geq 0} \pi_{ij} = 1$.

By PASTA, an arrival sees $(N_1, N_2) = (i, j)$ with probability π_{ij}. It joins behind $\min(i, j)$ requests. FCFS and exponential memorylessness give

$$\mathbb{E}\left[W_q \mid N_1 = i, N_2 = j\right] = \frac{\min(i,j)}{\mu}.$$

Average over the states seen at arrival:

$$p_{\text{wait}} = \sum_{i\geq 1, j\geq 1} \pi_{ij}, \quad w_q = \sum_{i,j\geq 0} \pi_{ij} \frac{\min(i,j)}{\mu}, \quad w = w_q + \frac{1}{\mu}.$$

As a check, $\ell_q = \sum_{i,j\geq 0} \pi_{ij}(\max(i-1,0) + \max(j-1,0)) = \lambda w_q$.

Why does round robin improve on random routing?

Alternate Poisson arrivals of rate λ between two processors. Each processor sees **two** independent $\mathrm{Exp}(\lambda)$ stages between arrivals: an **Erlang-2/M/1** queue. The two arrival streams are dependent.

For $\lambda < 2\mu$, the GI/M/1 calculation gives the non-unit root

$$\sigma = \left(\frac{\lambda}{\lambda + \mu(1-\sigma)}\right)^2 = \left(\frac{\sqrt{1+4\frac{\lambda}{\mu}}-1}{2}\right)^2.$$

The stationary waiting-time tail and mean are

$$\mathbb{P}\big(W_q > t\big) = \sigma e^{-\mu(1-\sigma)t}, \quad w_q = \frac{\sigma}{\mu(1-\sigma)}, \quad t \geq 0.$$

At $\lambda = 24$/min and $\mu = 15$/min, $\sigma \approx 0.73985$, $\boldsymbol{w_q = 11.38}$ **s** and $\boldsymbol{w = 15.38}$ **s**.

Why it helps: alternation regularizes arrivals. **Why it falls short:** it ignores current queues, so an idle processor can coexist with waiting requests at the other.

Does a smaller mean order every waiting time?

For the same two-processor example, the mean-wait ordering is **pooling, JSQ, RR, random routing**. Other measures need not have that ranking.

Measure	Random	RR	JSQ	Pooled
Waiting probability p_{wait}	80%	73.99%	69.42%	71.11%
Mean wait among those who wait	20 s	15.38 s	11.27 s	10 s
Probability wait exceeds 10 s	48.52%	38.61%	28.72%	26.16%

JSQ has **fewer positive waits** than pooling, but **longer waits among those who wait**. Its waiting-time tail starts lower, then crosses the pooled tail. Thus the mean ordering does not imply $\mathbb{P}\left(W_q^{\text{pool}} > t\right) \leq \mathbb{P}\left(W_q^{\text{JSQ}} > t\right)$ for every $t \geq 0$.

JSQ optimality: Winston (1977). Round-robin mean performance: Liu and Towsley (1994).

Reading: the single-queue principles

- **Existing course material:** Section 4.5 gives the birth-and-death method. Section 5.1 derives the gateway stationary probabilities. Chapter 1 supplies the canteen and bus examples. The inherited *Queueing* note supplies the cashier, lead-time, finite-capacity, pooling and faster-server questions.
- **Ross, Introduction to Probability Models**, Chapter 8: M/M/1 and finite capacity (pp. 502–517) and the M/G/1 workload argument (pp. 538–540). Available in the course's local Books folder.
- **Sampling:** Wolff, Poisson Arrivals See Time Averages (1982). The relevant state is just before arrival; nonanticipation is essential.
- **Customer-time:** Sigman, Notes on Little's Law (2009). A sample-path treatment and boundary/population interpretation.

Gateway parameter changes and numerical comparisons in Chapter 5 are calculations for the models stated on the slides.

Reading: service variability and Poisson output

- **Residual work:** Modiano, MIT M/G/1 notes, slides 2–6. Residual-service triangles and P–K.
- **Poisson output:** Modiano, MIT Burke's theorem notes, slides 2–4. The stationary output theorem is stronger than equality of rates or a marginal gap calculation.

F Markov Decision Process Reference

Optional: calendar time and time remaining

The opening lecture uses $V_h(i)$ for maximum output with h shifts left. This optional reference retains the **inventory cost** formulation for later study, using $W_n(x)$ with n days remaining.

Let $V_t(x)$ be the smallest expected remaining cost **conditional on $X_t = x$**, including the terminal cost $H(X_T)$.

For a time-homogeneous model with the same terminal cost,

$$V_t(x) = W_{T-t}(x), \quad V_T(x) = H(x).$$

If costs or transitions vary with calendar time, write

$$V_t(x) = \min_{a \in \mathcal{A}_t(x)} \{c_t(x,a) + \sum_{x'} P_t(x'|x,a) V_{t+1}(x')\}.$$

Both conventions solve the final decision first and work backward.

Optional: terminal disposal cost

Suppose closing the shop costs **4 per leftover unit**, in addition to the final day's holding cost. Set $W_0(x) = H(x) = 4x$.

Starting empty with one day left gives

$$Q_1(0,1) = 2 + 3.5 + 4(0.3) = 6.7,$$

$$Q_1(0,2) = 2 + 1.8 + 4(1.0) = 7.8.$$

Order q	0	1	2	3	4	5
Total cost	8.8	**6.7**	7.8	11.5	16.5	21.5

Order one instead of two. The Bellman recursion is unchanged. The terminal cost changes its starting values.

Why discounted value iteration converges

Define $(Tv)(x) = \min_q \{c(x,q) + \gamma \sum_{x'} P(x'|x,q)v(x')\}$. For two value vectors u, v, each action's expected future values differ by at most $\gamma \max_{x'} |u(x') - v(x')|$.

Taking a minimum cannot increase this bound. Therefore

$$\|Tu - Tv\|_\infty \leq \gamma \|u - v\|_\infty.$$

Since $\gamma < 1$, the operator has a unique fixed point V^* and $v_k = T^k v_0$ converges to it. With $\delta_k = \|v_{k+1} - v_k\|_\infty$, summing the remaining geometric differences gives

$$\|v_{k+1} - V^*\|_\infty \leq \frac{\gamma}{1-\gamma} \delta_k.$$

At $\gamma = 0.9$, $\delta_k \leq 0.001$ bounds the value error by 0.009. This bounds value estimates, not the performance loss of a policy selected from them.

Discounted policy iteration

Start with any stationary deterministic policy π.

1. **Evaluate:** solve $(I - \gamma P^{\pi})V^{\pi} = c^{\pi}$.
2. **Improve:** at each state, choose an action minimizing $c(x, q) + \gamma \sum_{x'} P(x'|x, q)V^{\pi}(x')$.
3. **Repeat** until the policy no longer changes. Retain the current action whenever it is tied for the minimum.

An improved policy has value no greater than V^{π} in every state. For a finite discounted MDP, this procedure terminates at an optimal policy.

At termination, the evaluated value satisfies the **optimality equation** because its own actions attain every minimum.

A sufficient condition for average-cost optimality

Define the expected average-cost criterion

$$g^{\pi}(x) = \lim_{N\to\infty} \sup \frac{1}{N} \mathbb{E}_x^{\pi} \left[\sum_{t=0}^{N-1} c(X_t, q_t) \right].$$

Suppose we find a constant g and a vector h satisfying

$$g + h(x) = \min_{q} \{ c(x,q) + \sum_{x'} P(x'|x,q) h(x') \}$$

at every state. Then g is a lower bound on every policy's average cost. A stationary policy attaining every minimum achieves g.

To see why, sum the Bellman inequalities over time. The intermediate h terms cancel, leaving $h(X_0) - \mathbb{E}[h(X_N)]$. Dividing by N makes this term vanish because the state space is finite.

An average-cost certificate for inventory

For this inventory model, take

$$g = \frac{303}{110}, \quad h = \left(0, 0, -\frac{15}{11}, -2, -\frac{102}{77}, \frac{61}{539}\right).$$

Entries give current cost plus expected next relative value, $c(x, q) + P_q h$.

Stock	Order	Selected cost	Best alternative	$g + h(x)$
0	3	2.7545	3.3909	2.7545
1	2	2.7545	3.3909	2.7545
2	0	1.3909	2.7545	1.3909
3	0	0.7545	3.4299	0.7545
4	0	1.4299	4.8677	1.4299
5	0	2.8677	Unavailable	2.8677

The selected orders attain $g + h(x)$ in every state. Thus $(1, 3)$ also minimizes average cost, including from stocks 4 and 5.

Policy structure and model assumptions

The inventory calculation optimizes over **all feasible quantities** $q \in \{0, \ldots, 5 - x\}$. The resulting (s, S) pattern is a conclusion for the stated parameters.

A fixed order charge can favor batching. With other costs or demand assumptions, the reorder point and target may change. Delivery delays require more state information.

The full action set also permits policies with several recurrent classes. For example, orders $(0, 0, 3, 2, 1, 0)$ leave state 0 absorbing while $\{2, 3, 4, 5\}$ is another closed class. A single stationary average then need not describe every initial state.

Our comparison policies each have one recurrent class. The constant-g Bellman certificate above proves optimality for every initial state without assuming that every feasible policy is irreducible.

F.1 Queue Admission Control

Revenue now, congestion later

Admitting a job earns revenue now but can create future holding costs. Use the finite-horizon Bellman recursion to maximize expected discounted net revenue over two slots, with $\gamma = \frac{1}{2}$. After the second slot, no further revenue or holding costs are counted.

A service system with two places — Example F.1

The system can hold **two unfinished jobs**, including any job in service. At the beginning of each slot, observe the number of jobs in the system, n. One new job arrives for possible admission.

- **Accept** brings in 2 units of revenue and **Reject** brings in 0. At $n = 2$, rejection is compulsory.
- Jobs already present incur a holding cost of $4n$ for this slot.

At the end of the slot, one job completes with probability $\frac{1}{2}$ if the system is nonempty. A newly admitted job can complete in that slot if it is the only job.

Formulate the admission decision



Write $a = 1$ for Accept and $a = 0$ for Reject. A newly admitted job first incurs holding cost next slot if it remains unfinished.

? Build the model

1. What states can we observe? Which actions are feasible in each?
2. Which jobs incur holding cost in the current slot?
3. With one job present, which next states can follow Accept? Which can follow Reject?

State, action, reward, and transition

State n counts jobs before admission. The action sets are $\mathcal{A}(0) = \mathcal{A}(1) = \{0, 1\}$ and $\mathcal{A}(2) = \{0\}$.

$$r(n, a) = 2a - 4n, \quad n' = n + a - D.$$

When $n + a > 0$, independent trials give $D = 0$ or 1 with equal probability in each such slot. If the system is empty after admission, $D = 0$.

State n	Action	Reward	Next state
0	Reject	0	0 with probability 1
0	Accept	2	0 or 1, each with probability $\frac{1}{2}$
1	Reject	-4	0 or 1, each with probability $\frac{1}{2}$
1	Accept	-2	1 or 2, each with probability $\frac{1}{2}$
2	Reject	-8	1 or 2, each with probability $\frac{1}{2}$

One slot remaining

Let $w^{(k)}(n)$ be the optimal discounted value for the next k slots. Set $w^{(0)}(n) = 0$ because unfinished jobs incur no terminal cost.

State	Reject	Accept	$w^{(1)}(n)$
0	0	2	2
1	-4	-2	-2
2	-8	Unavailable	-8

With one slot left, accepting is better whenever there is room. The next update will also count the congestion left for a later slot.

Two slots remaining (exercise)

? Start with one job

Use the final-slot values $\left(w^{(1)}(0), w^{(1)}(1), w^{(1)}(2)\right) = (2, -2, -8)$.

At $n = 1$, which states can follow each action? Add today's reward to $\frac{1}{2}$ times the expected final-slot value. Should we accept or reject?

Complete the rule. Calculate both totals at $n = 0$ and the only feasible total at $n = 2$.

Initial jobs	Reject now	Accept now
0	$0 + \frac{1}{2}(2)$	$2 + \frac{1}{2}\left(\frac{2-2}{2}\right)$
2	$-8 + \frac{1}{2}\left(\frac{-2-8}{2}\right)$	Unavailable

The optimal two-slot admission rule

At $n = 1$, Reject leaves 0 or 1 jobs and Accept leaves 1 or 2, with equal chances in each pair.

Jobs	Reject now	Accept now	Two slots left
0	1	2	Accept
1	$-4 + \frac{1}{2}\left(\frac{2-2}{2}\right) = -4$	$-2 + \frac{1}{2}\left(\frac{-2-8}{2}\right) = -4.5$	Reject
2	-10.5	Unavailable	Must reject



Admission revenue and future congestion

At $n = 1$, accepting gains 2 now but loses 2.5 in discounted future value. Reject with two slots left.

In the final slot, accept whenever there is room. With two slots left, $w^{(2)} = (2, -4, -10.5)$.

At $n = 1$, adding a slot of holding costs lowers the value from -2 to -4. Each value is optimal for its own horizon.