Minimizing Delay in Supermarket-Checkout Systems

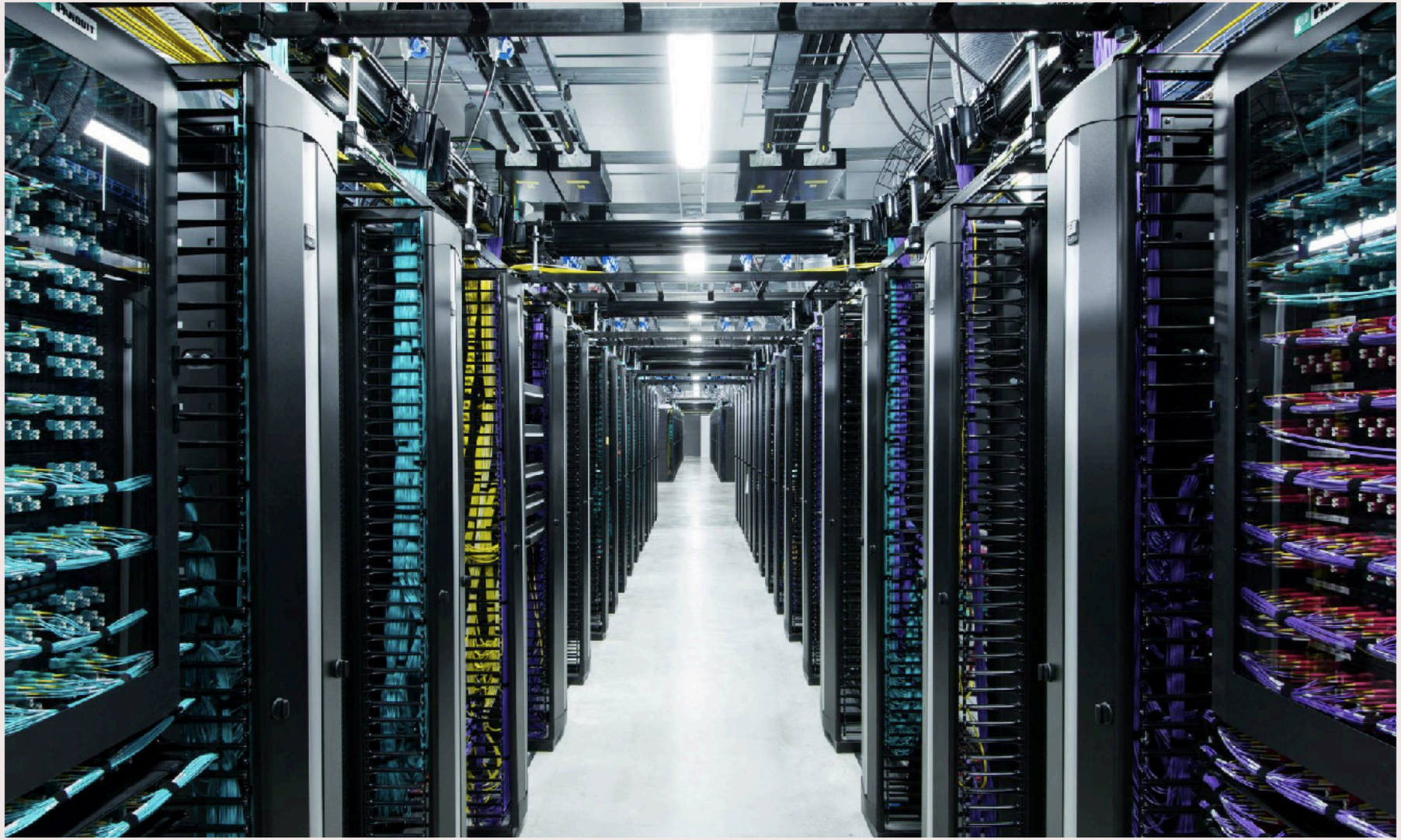
Daniela Hurtado-Lange
William and Mary
Joint work with Siva Theja Maguluri

Mathematics Colloquium, April 1st, 2022

Contact information: dahurtadolange@wm.edu

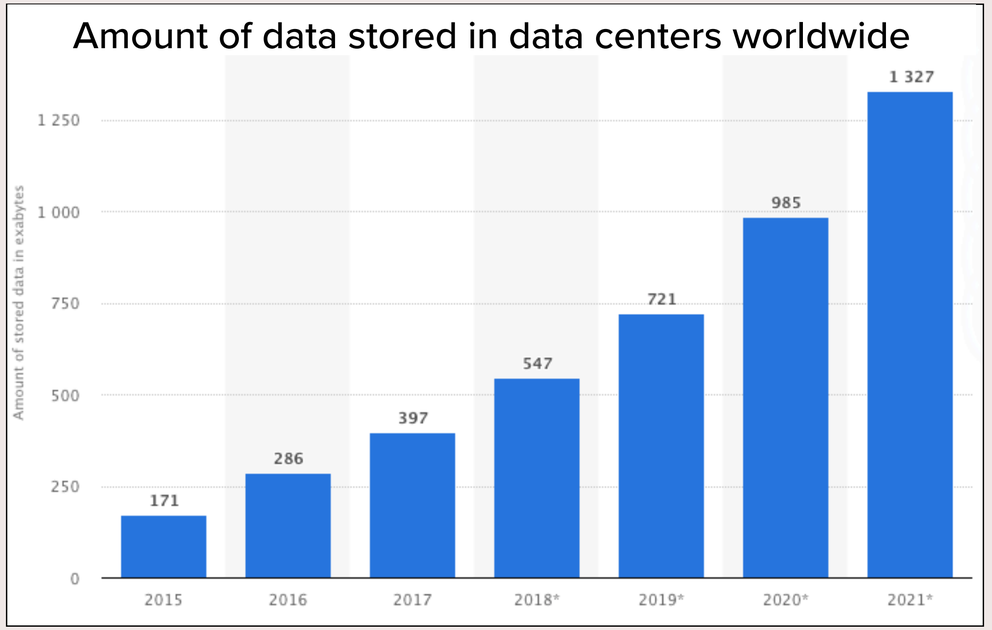
Motivation

Data Centers



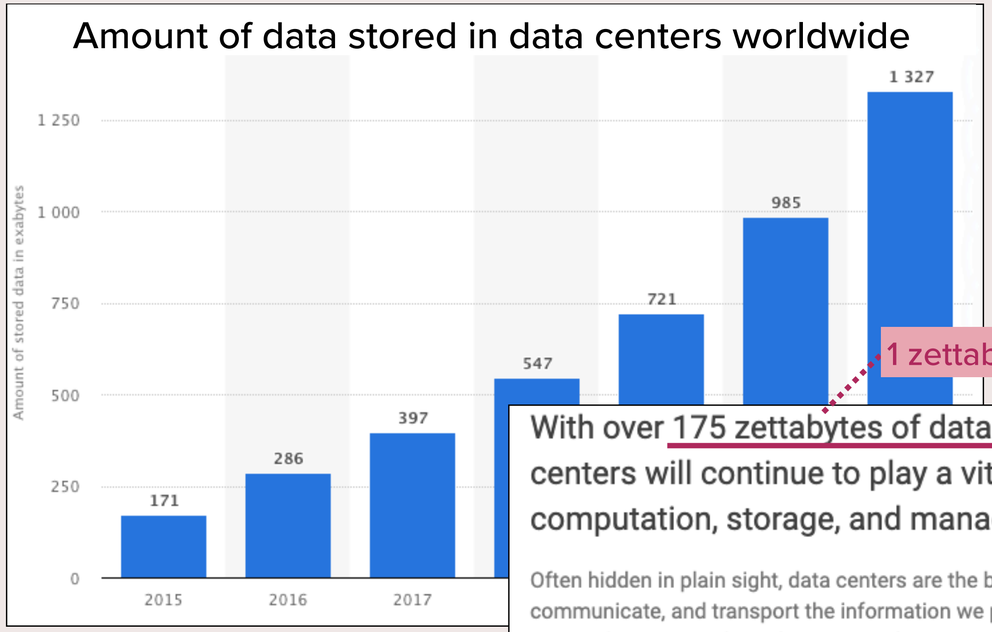
Data Storage and Production

Data Storage and Production



Source: https://www.statista.com/statistics/638613/worldwide-data-center-storage-

Data Storage and Production



Source: https://www.statista.com/statistics/

1 zettabyte = 1B terabytes

With over 175 zettabytes of data expected by 2025, data centers will continue to play a vital role in the ingestion, computation, storage, and management of information.

Often hidden in plain sight, data centers are the backbone of our internet. They store, communicate, and transport the information we produce every single day. The more data we create, the more vital our data centers become.

But many of today's data centers are clunky, inefficient, and outdated. To keep them running, data center operators, from FAMGA to colocation providers, are working on upgrading them to fit our ever-changing world.

Source: The future of data center, CB insights, available at https://www.cbinsights.com/research/future-of-data-centers/

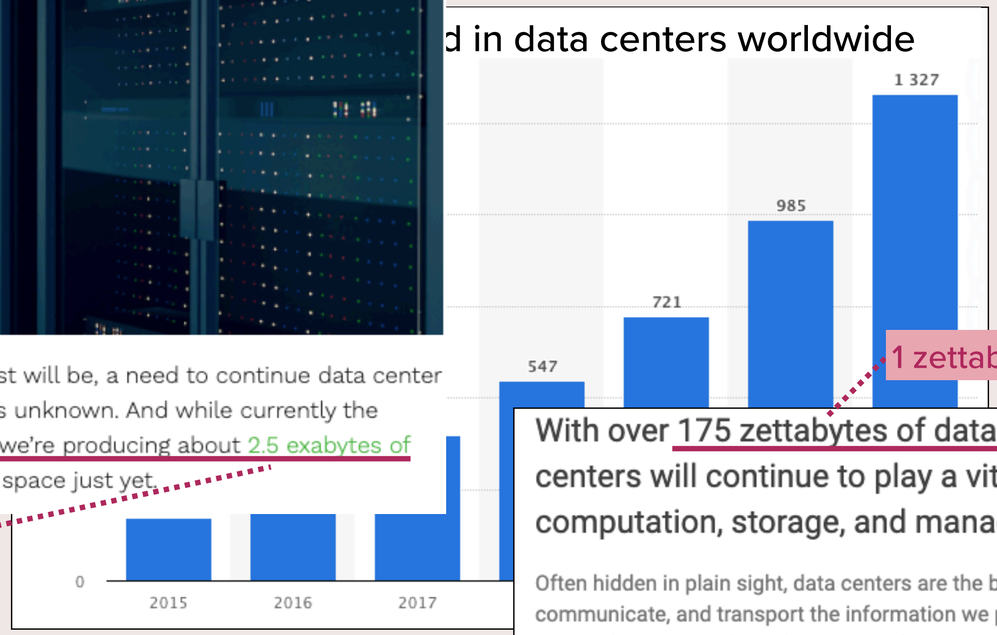
Data Storage and Production

Are There Data Center Storage Capacity Constraints?

There's no argument in the data world that there is, or at least will be, a need to continue data center and data storage expansion, but what that future looks like is unknown. And while currently the world's data centers store about 1,327 exabytes of data, and we're producing about 2.5 exabytes of data daily, we're not in danger of running out of data storage space just yet.

Source: Are data centers running out of storage?, VXCHNGE, available at https://www.vxchnge.com/blog/are-data-centers-running-out-of-storage

1 exabyte = 1M terabytes



Source: https://www.statista.com/statistics/

1 zettabyte = 1B terabytes

With over 175 zettabytes of data expected by 2025, data centers will continue to play a vital role in the ingestion, computation, storage, and management of information.

Often hidden in plain sight, data centers are the backbone of our internet. They store, communicate, and transport the information we produce every single day. The more data we create, the more vital our data centers become.

But many of today's data centers are clunky, inefficient, and outdated. To keep them running, data center operators, from FAMGA to colocation providers, are working on upgrading them to fit our ever-changing world.

Source: The future of data center, CB insights, available at https://www.cbinsights.com/research/future-of-data-centers/

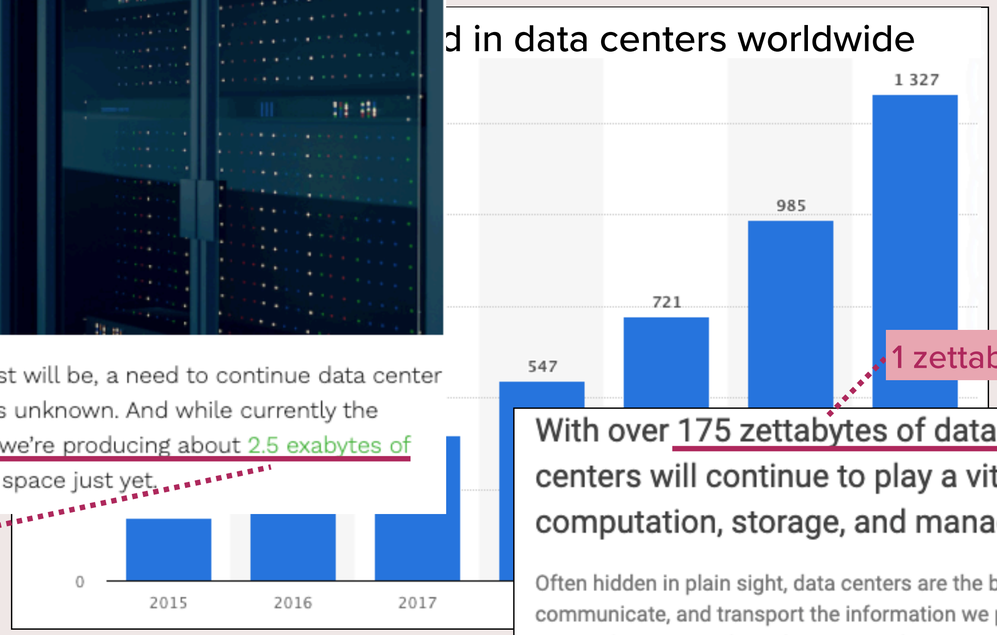
Data Storage and Production

Are There Data Center Storage Capacity Constraints?

There's no argument in the data world that there is, or at least will be, a need to continue data center and data storage expansion, but what that future looks like is unknown. And while currently the world's data centers store about 1,327 exabytes of data, and we're producing about 2.5 exabytes of data daily, we're not in danger of running out of data storage space just yet.

Source: Are data centers running out of storage?, VXCHNGE, available at https://www.vxchnge.com/blog/are-data-centers-running-out-of-storage

1 exabyte = 1M terabytes



Source: https://www.statista.com/statistics/

1 zettabyte = 1B terabytes

With over 175 zettabytes of data expected by 2025, data centers will continue to play a vital role in the ingestion, computation, storage, and management of information.

Often hidden in plain sight, data centers are the backbone of our internet. They store, communicate, and transport the information we produce every single day. The more data we create, the more vital our data centers become.

But many of today's data centers are clunky, inefficient, and outdated. To keep them running, data center operators, from FAMGA to colocation providers, are working on upgrading them to fit our ever-changing world.

Source: The future of data center, CB insights, available at https://www.cbinsights.com/research/future-of-data-centers/

29 terabytes per second

Significance of Delay

Significance of Delay

400 ms delay implies:

- ↓0.59% # searches per user
- ↓20% traffic

Significance of Delay

Google

400 ms delay implies:

- ↓0.59% # searches per user
- ↓20% traffic

Bing

Additional 1.5 s delay implies:

- ↓1.8% queries per user
- ↓4.3% revenues per user
- ↓3.8% overall satisfaction

Significance of Delay

Google

400 ms delay implies:

- ↓0.59% # searches per user
- ↓20% traffic

Bing

Additional 1.5 s delay implies:

- ↓1.8% queries per user
- ↓4.3% revenues per user
- ↓3.8% overall satisfaction

amazon

100 ms delay implies:

↓1% revenues ≈ ↓$745 million/year

Significance of Delay

Google

400 ms delay implies:

- ↓0.59% # searches per user
- ↓20% traffic

Bing

Additional 1.5 s delay implies:

- ↓1.8% queries per user
- ↓4.3% revenues per user
- ↓3.8% overall satisfaction

amazon

100 ms delay implies:

↓1% revenues ≈ ↓$745 million/year

Goal: Minimize delay

Significance of Delay

Google

400 ms delay implies:

- ↓0.59% # searches per user
- ↓20% traffic

Bing

Additional 1.5 s delay implies:

- ↓1.8% queries per user
- ↓4.3% revenues per user
- ↓3.8% overall satisfaction

amazon

100 ms delay implies:

↓1% revenues ≈ ↓$745 million/year

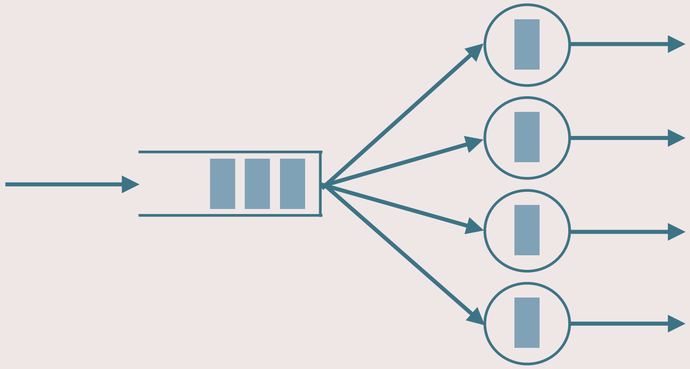
Goal: Minimize delay

How? Queueing theory

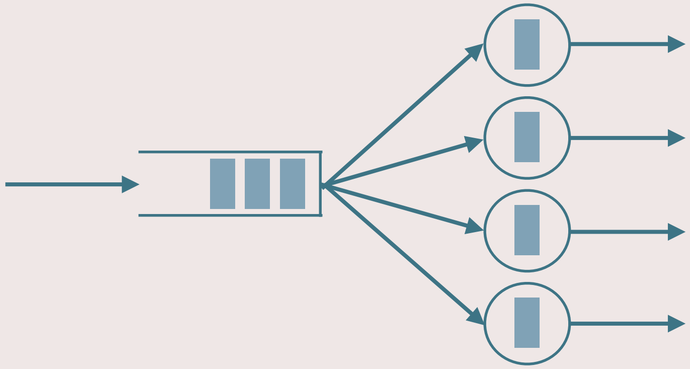
Stochastic Processing Networks (a.k.a. Queues)

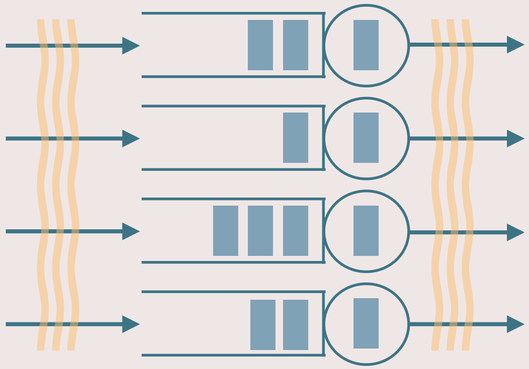
Stochastic Processing Networks (a.k.a. Queues)

Stochastic Processing Networks (a.k.a. Queues)

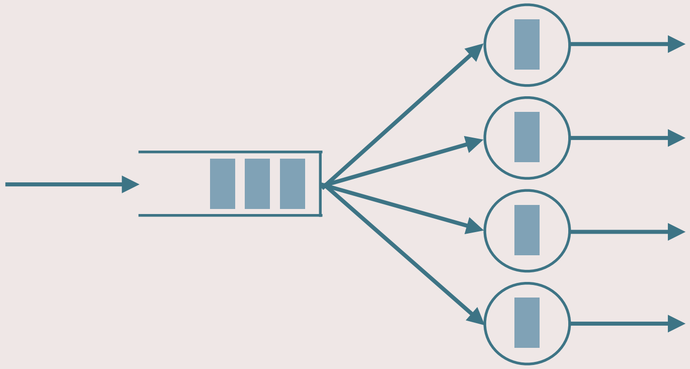


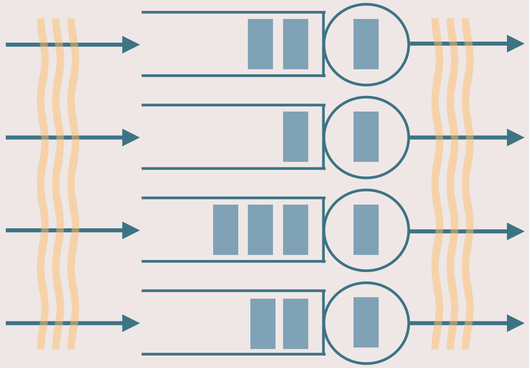
Stochastic Processing Networks (a.k.a. Queues)

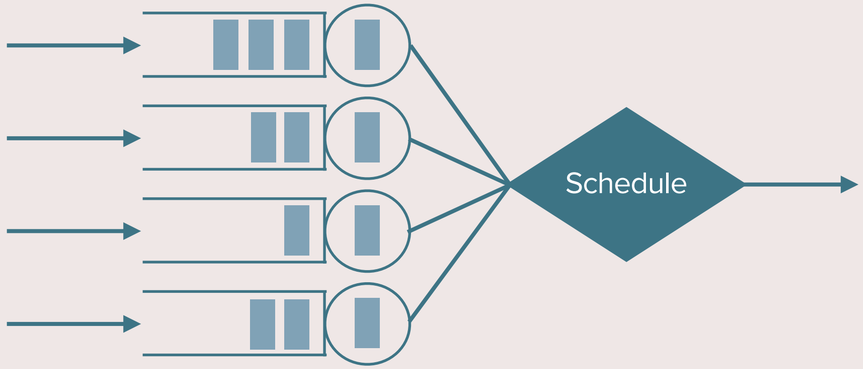




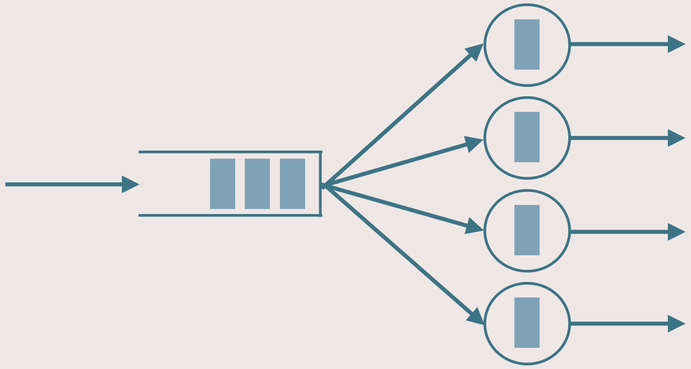
Stochastic Processing Networks (a.k.a. Queues)

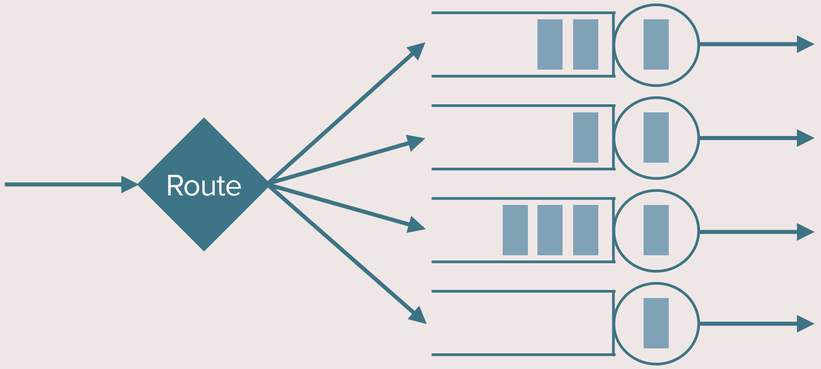


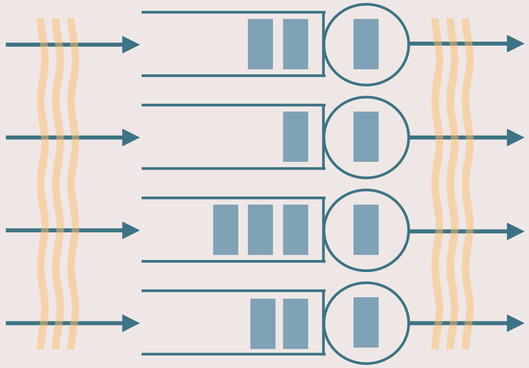


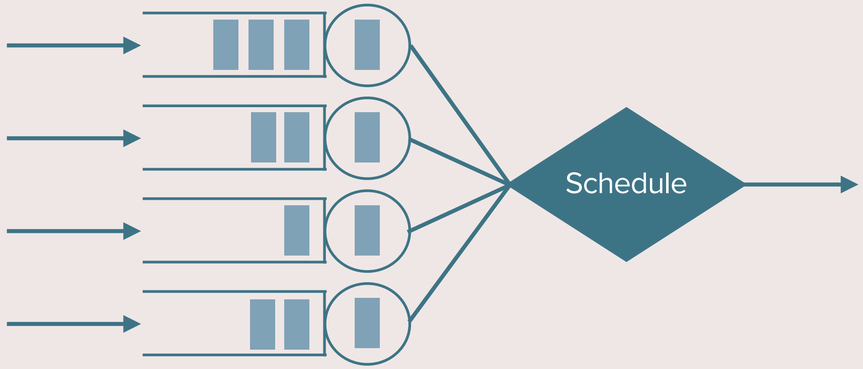


Stochastic Processing Networks (a.k.a. Queues)

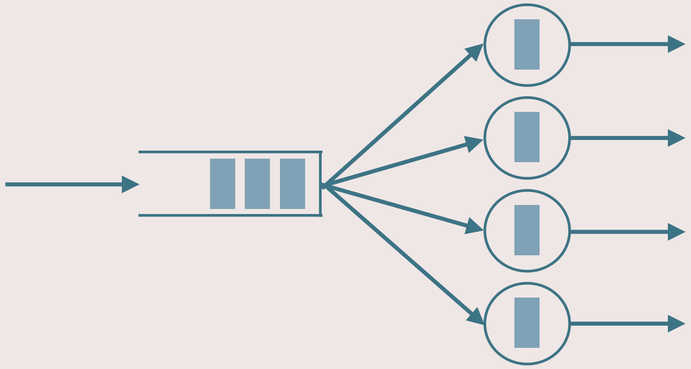


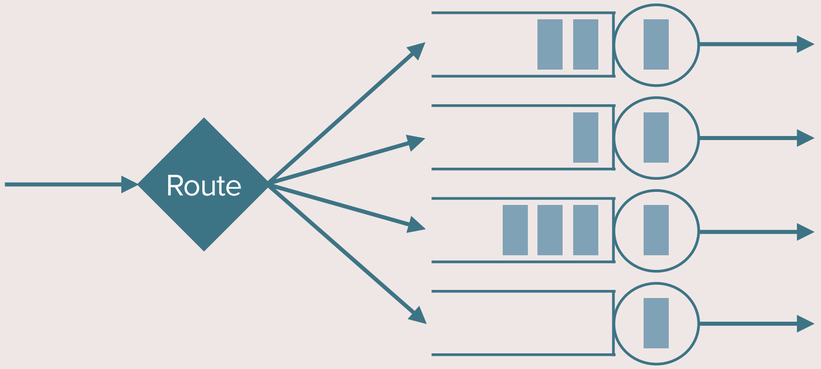


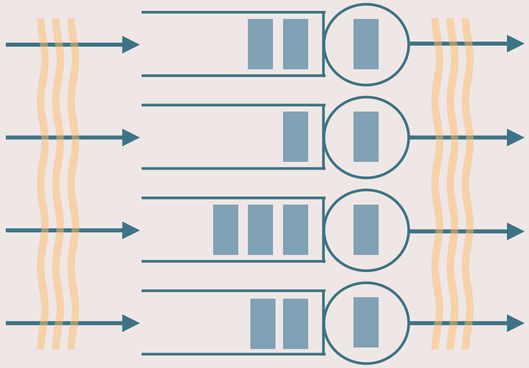


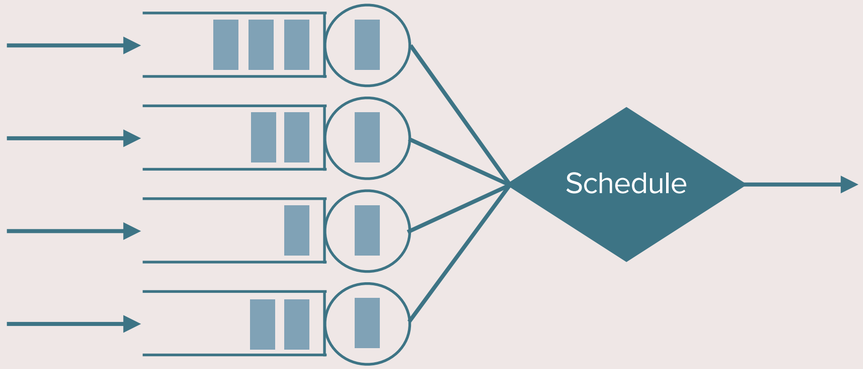


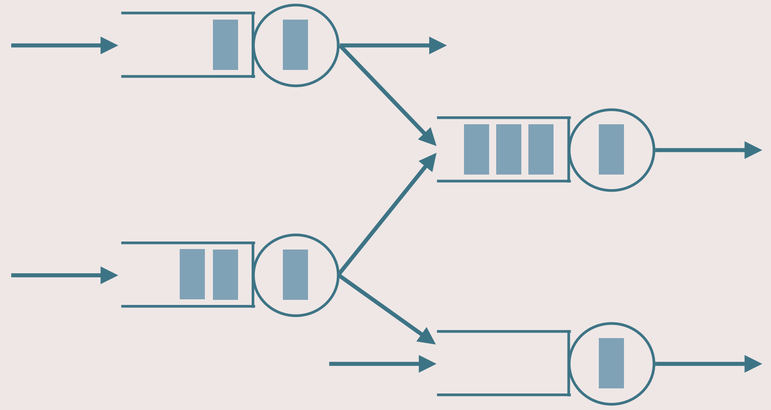
Stochastic Processing Networks (a.k.a. Queues)



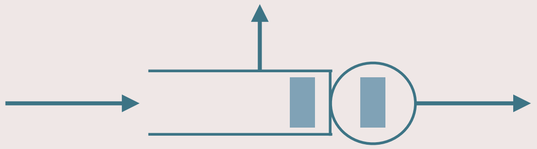


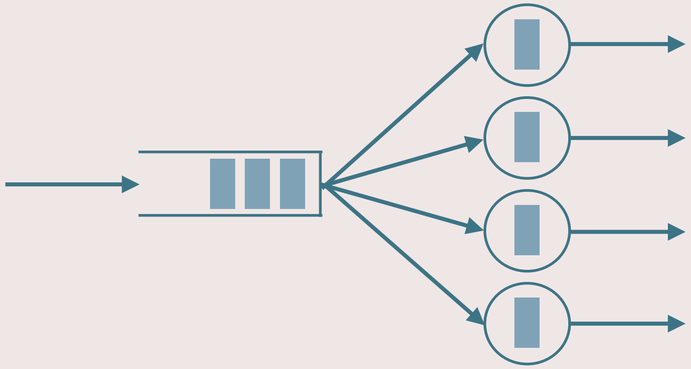


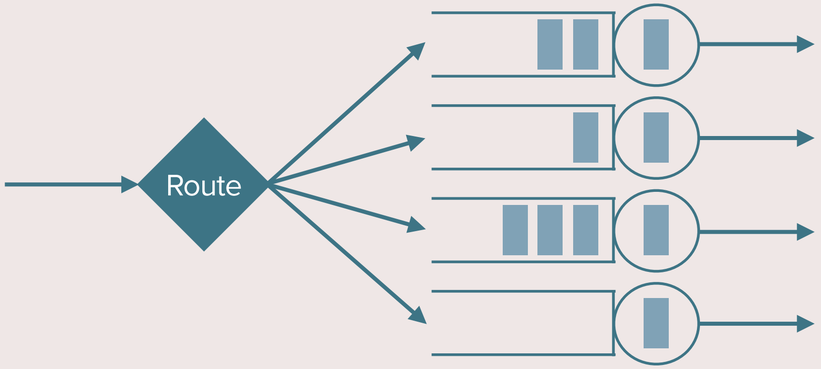


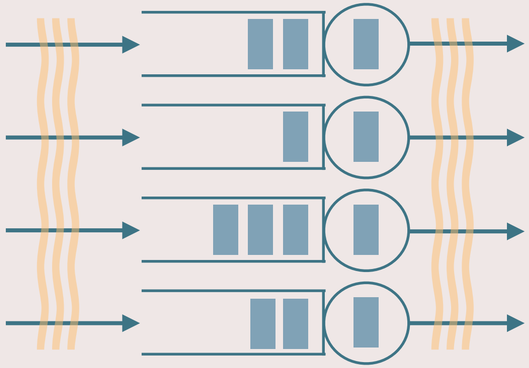


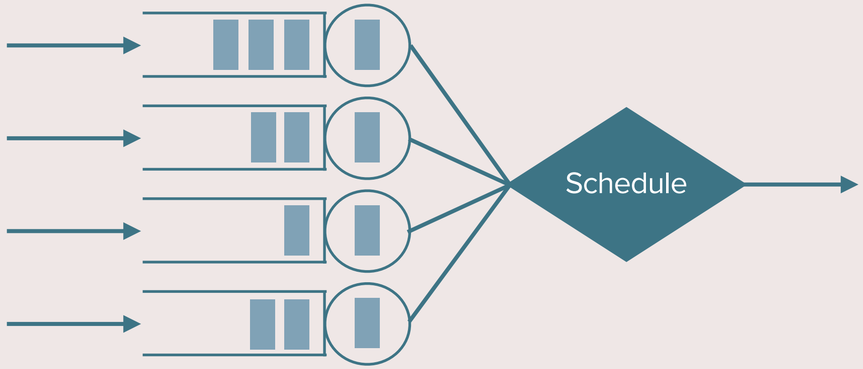
Stochastic Processing Networks (a.k.a. Queues)

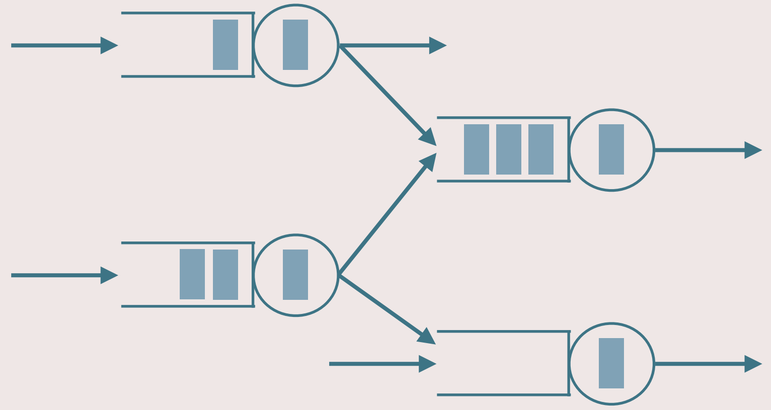




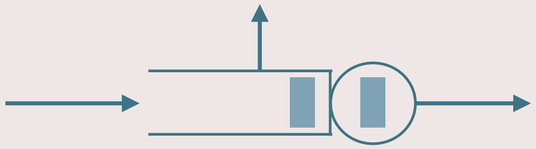


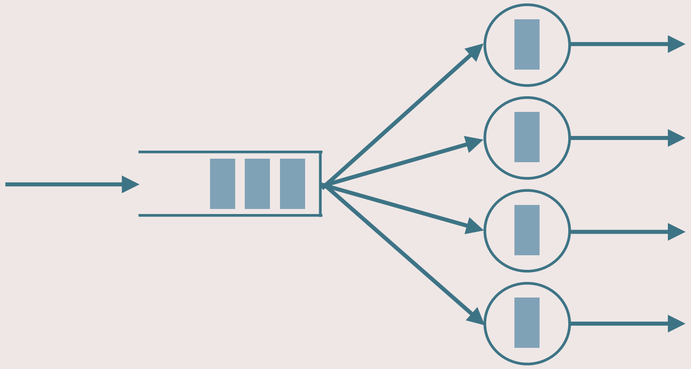


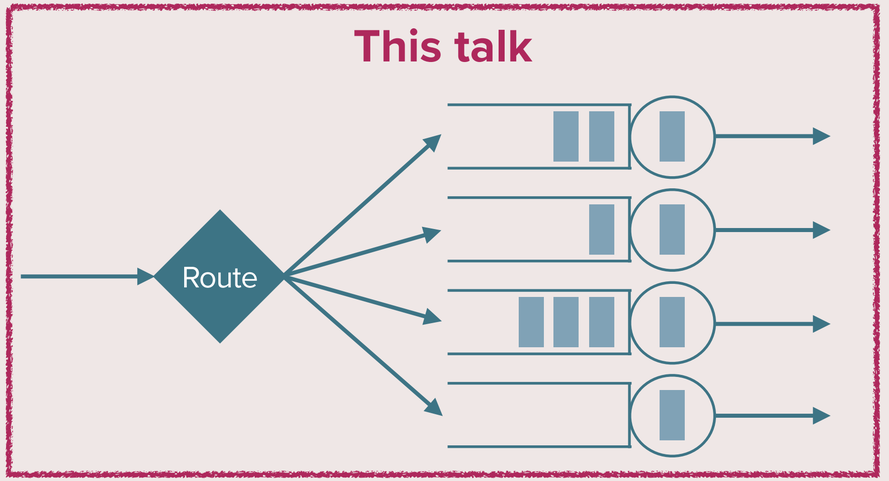


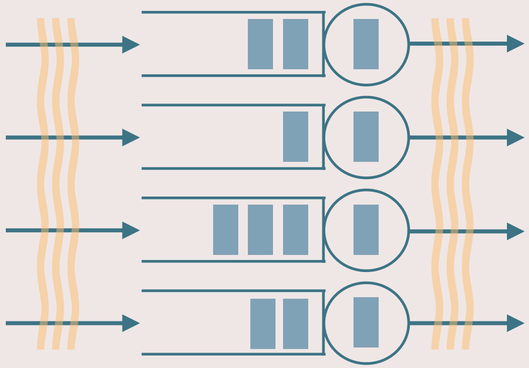


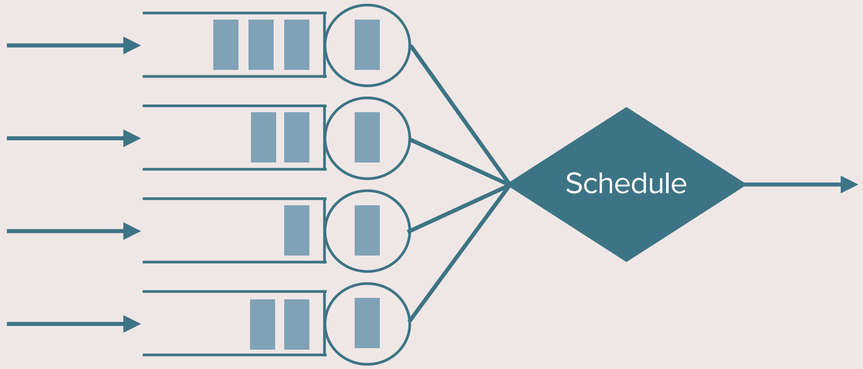
Stochastic Processing Networks (a.k.a. Queues)

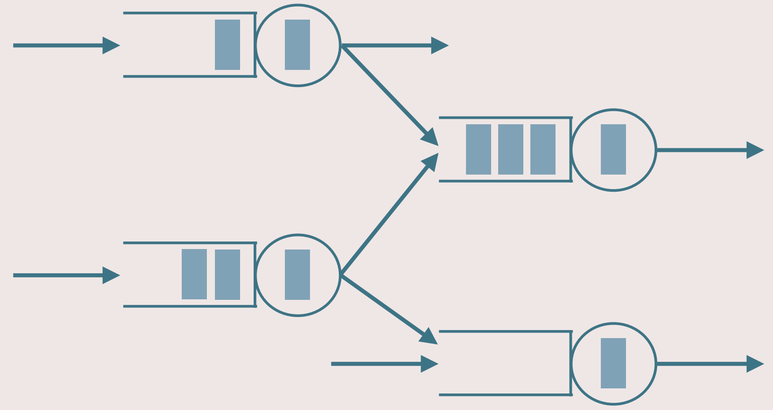












Outline

Outline

Part 1: **The Model**	• Important parts of a queueing model • Supermarket-checkout model • Some routing algorithms

Outline

Part 1: **The Model**	• Important parts of a queueing model • Supermarket-checkout model • Some routing algorithms
Part 2: **How to Solve?**	• Stability • Heavy-Traffic: The CRP condition • Analysis of a single-server queue • The MGF method

Outline

Part 1: **The Model**	• Important parts of a queueing model • Supermarket-checkout model • Some routing algorithms
Part 2: **How to Solve?**	• Stability • Heavy-Traffic: The CRP condition • Analysis of a single-server queue • The MGF method
Part 3: **An Open Question**	• Many-server heavy-traffic regime • Comparison of routing algorithms open question

Outline

Part 1: The Model	• Important parts of a queueing model • Supermarket-checkout model • Some routing algorithms
Part 2: How to Solve?	• Stability • Heavy-Traffic: The CRP condition • Analysis of a single-server queue • The MGF method
Part 3: An Open Question	• Many-server heavy-traffic regime • Comparison of routing algorithms open question
Conclusion and future work	

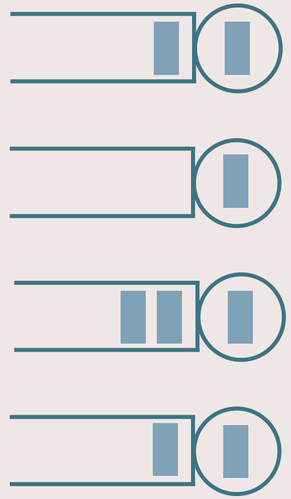
Outline

Part 1: The Model	• Important parts of a queueing model • Supermarket-checkout model • Some routing algorithms
Part 2: How to Solve?	• Stability • Heavy-Traffic: The CRP condition • Analysis of a single-server queue • The MGF method
Part 3: An Open Question	• Many-server heavy-traffic regime • Comparison of routing algorithms open question
Conclusion and future work	

Supermarket Checkout System

Supermarket Checkout System

- Discrete time model
- n servers with an infinite buffer

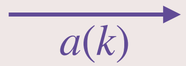


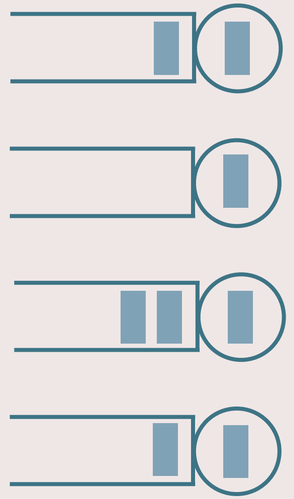
Supermarket Checkout System

- Discrete time model
- n servers with an infinite buffer

Arrivals $a(k)$:

- Single stream of arrivals
  - Sequence of i.i.d. random variables



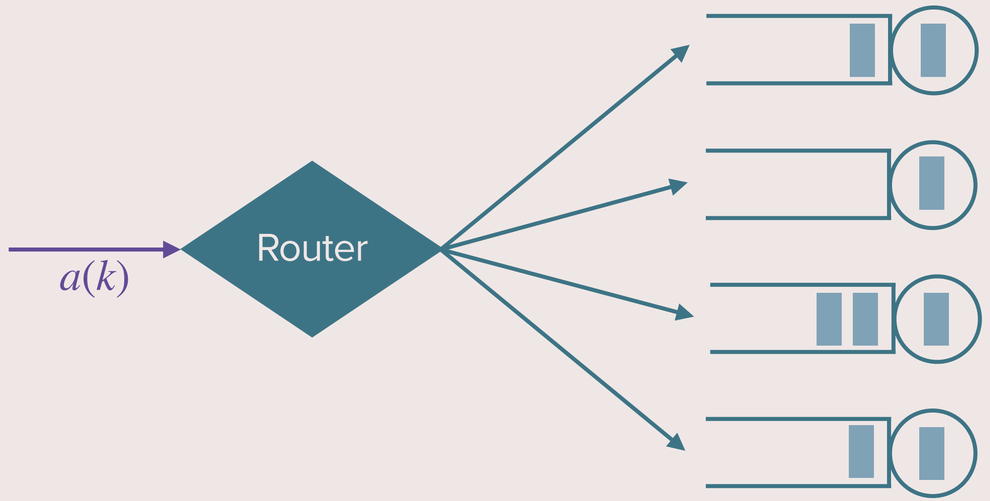


Supermarket Checkout System

- Discrete time model
- n servers with an infinite buffer

Arrivals $a(k)$:

- Single stream of arrivals
  - Sequence of i.i.d. random variables
- Upon arrival, jobs are routed to the queues



Supermarket Checkout System

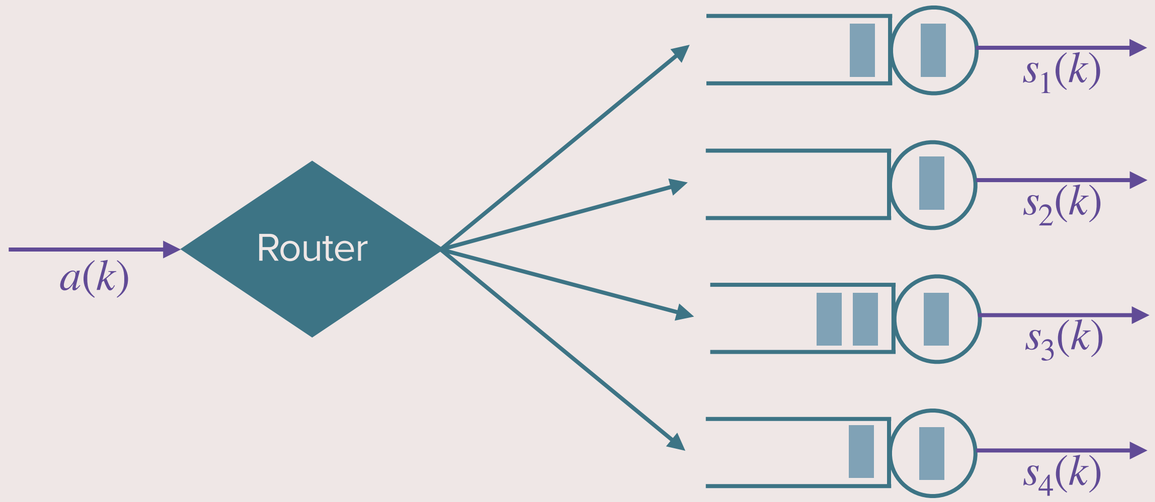
- Discrete time model
- n servers with an infinite buffer

Arrivals $a(k)$:

- Single stream of arrivals
  - Sequence of i.i.d. random variables
- Upon arrival, jobs are routed to the queues

Service $s_1(k), s_2(k), \ldots, s_n(k)$:

- Potential service is a sequence of i.i.d. random variables
- Service processes to different queues are correlated



Routing Algorithms

Routing Algorithms

Random routing:
Route arrivals to each queue uniformly at random

Routing Algorithms

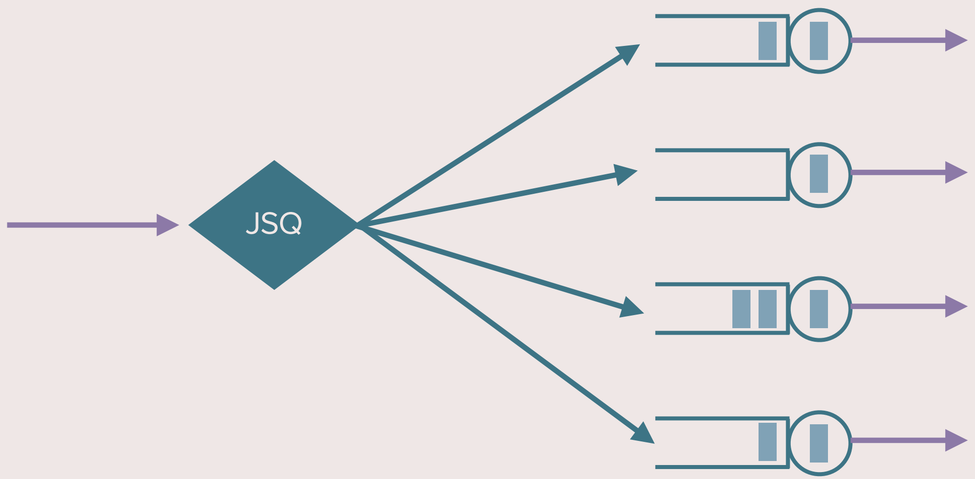
Random routing:
Route arrivals to each queue uniformly at random

Join the Shortest Queue (JSQ):
Route arrivals to the queue with the smallest number of jobs.

Routing Algorithms

Random routing:
Route arrivals to each queue uniformly at random

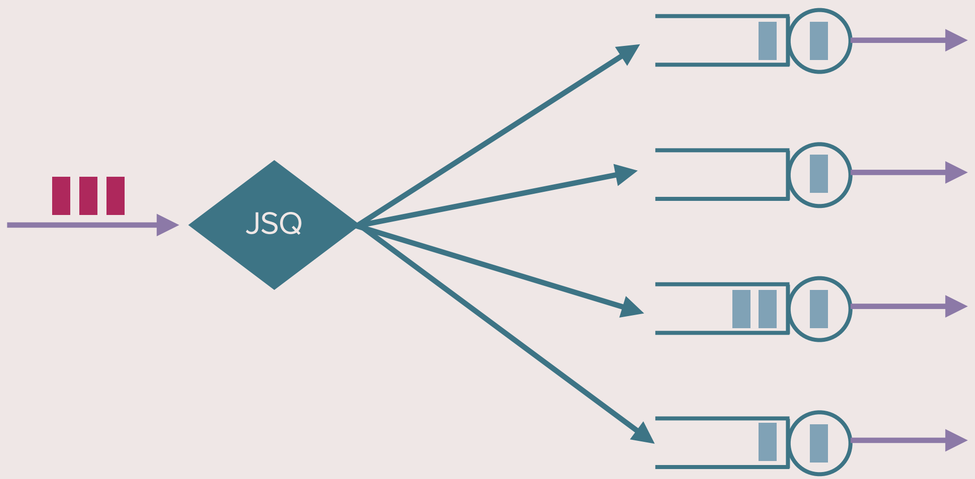
Join the Shortest Queue (JSQ):
Route arrivals to the queue with the smallest number of jobs.



Routing Algorithms

Random routing:
Route arrivals to each queue uniformly at random

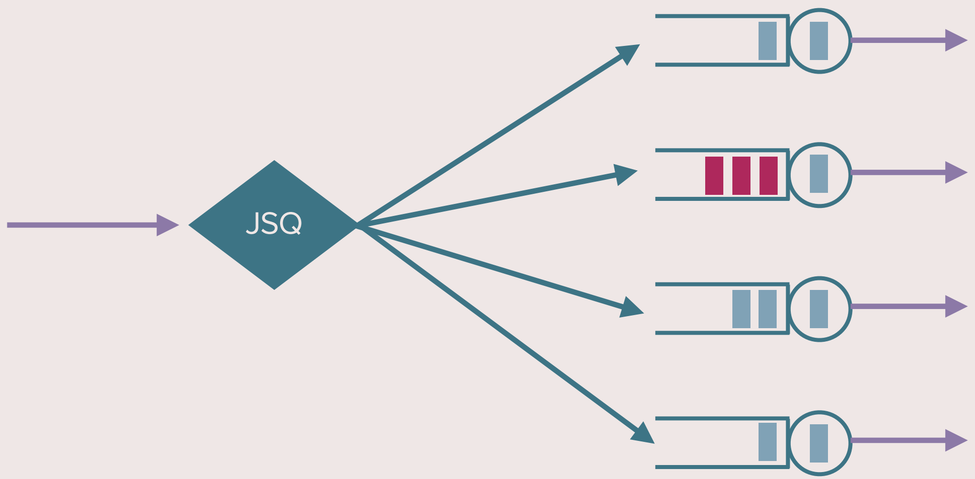
Join the Shortest Queue (JSQ):
Route arrivals to the queue with the smallest number of jobs.



Routing Algorithms

Random routing:
Route arrivals to each queue uniformly at random

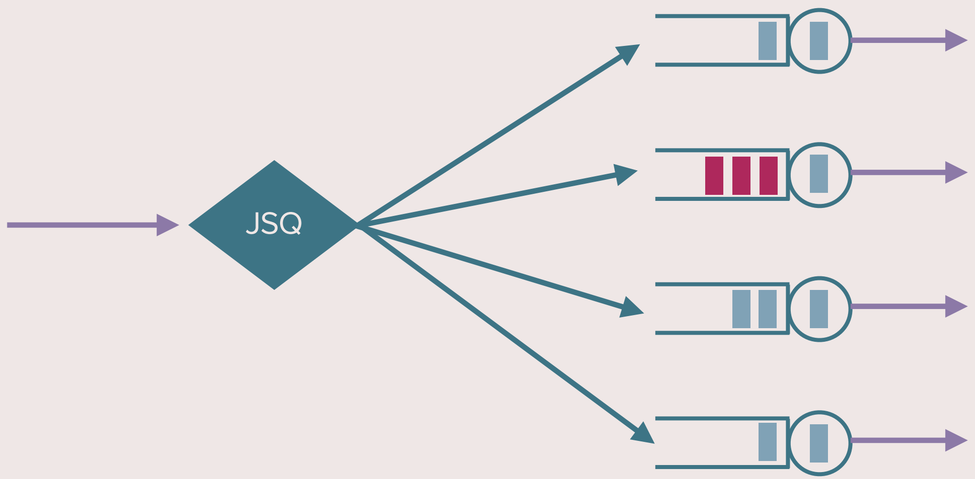
Join the Shortest Queue (JSQ):
Route arrivals to the queue with the smallest number of jobs.



Routing Algorithms

Random routing:
Route arrivals to each queue uniformly at random

Join the Shortest Queue (JSQ):
Route arrivals to the queue with the smallest number of jobs.



Power of d choices, or JSQ(d):
Given an integer $d \in [1,n] \cap \mathbb{Z}$, in each time slot:

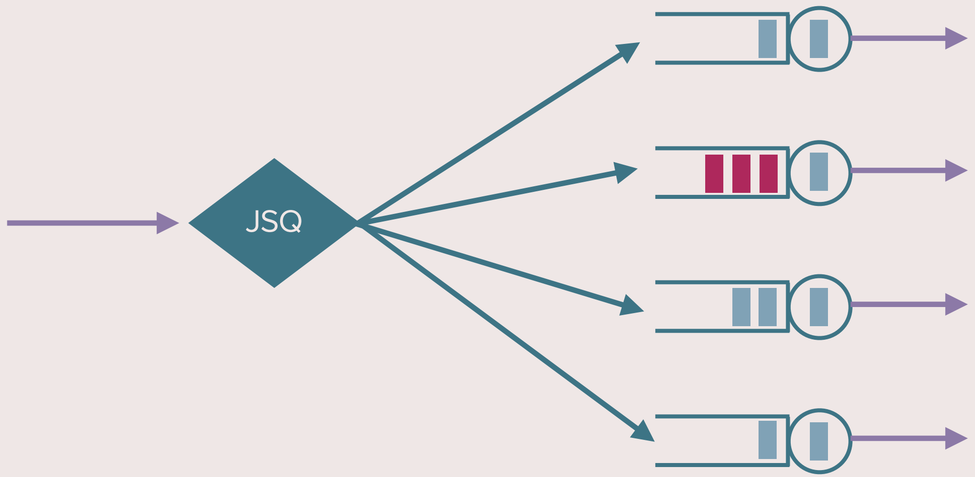
(1) Select d servers uniformly at random

(2) Route arrivals to the shortest queue among those d

Routing Algorithms

Random routing:
Route arrivals to each queue uniformly at random

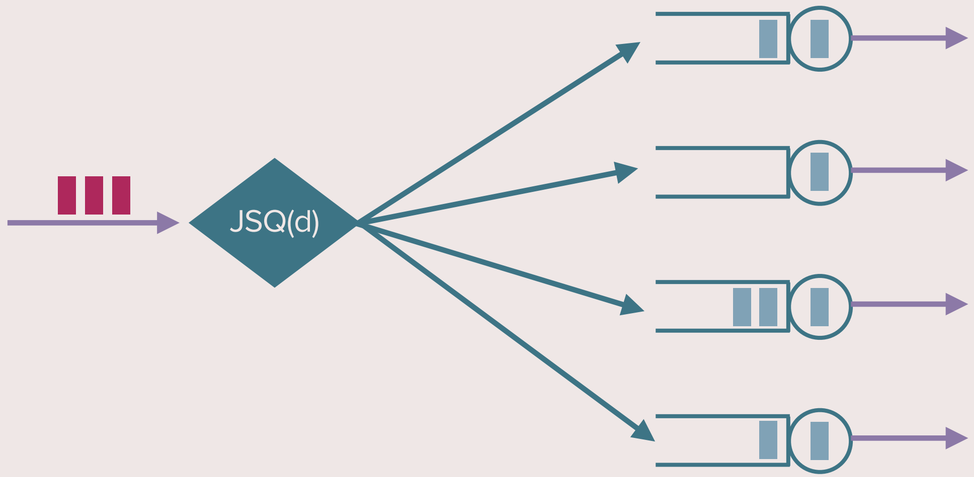
Join the Shortest Queue (JSQ):
Route arrivals to the queue with the smallest number of jobs.



Power of d choices, or JSQ(d):
Given an integer $d \in [1,n] \cap \mathbb{Z}$, in each time slot:

(1) Select d servers uniformly at random

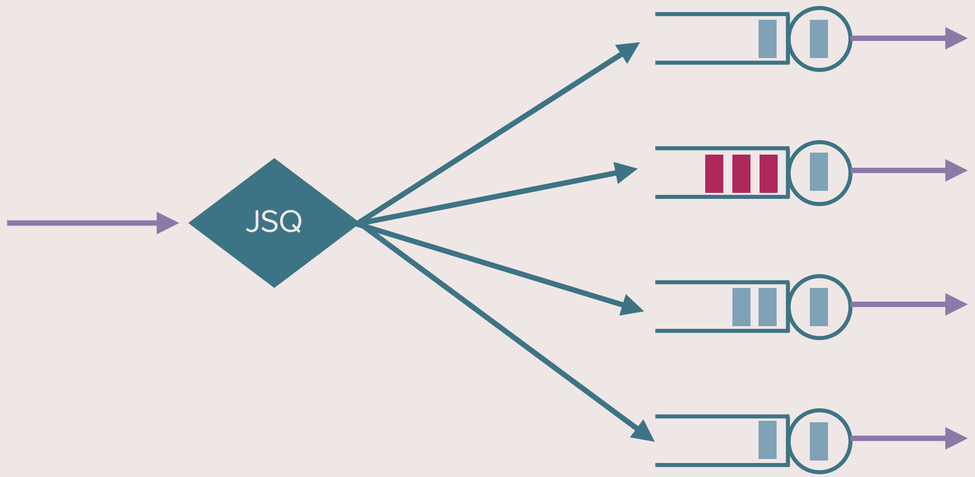
(2) Route arrivals to the shortest queue among those d



Routing Algorithms

Random routing:
Route arrivals to each queue uniformly at random

Join the Shortest Queue (JSQ):
Route arrivals to the queue with the smallest number of jobs.

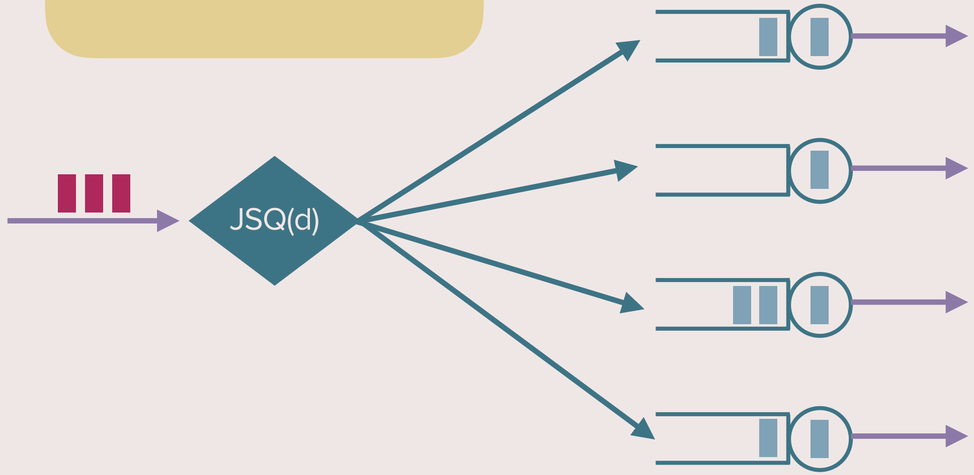


Power of d choices, or JSQ(d):
Given an integer $d \in [1,n] \cap \mathbb{Z}$, in each time slot:

(1) Select d servers uniformly at random

(2) Route arrivals to the shortest queue among those d

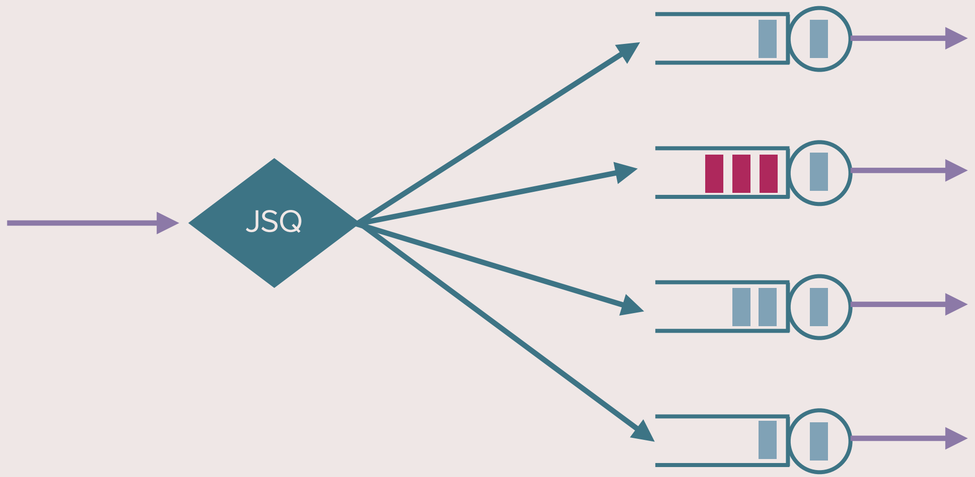
Example: $d = 2$



Routing Algorithms

Random routing:
Route arrivals to each queue uniformly at random

Join the Shortest Queue (JSQ):
Route arrivals to the queue with the smallest number of jobs.

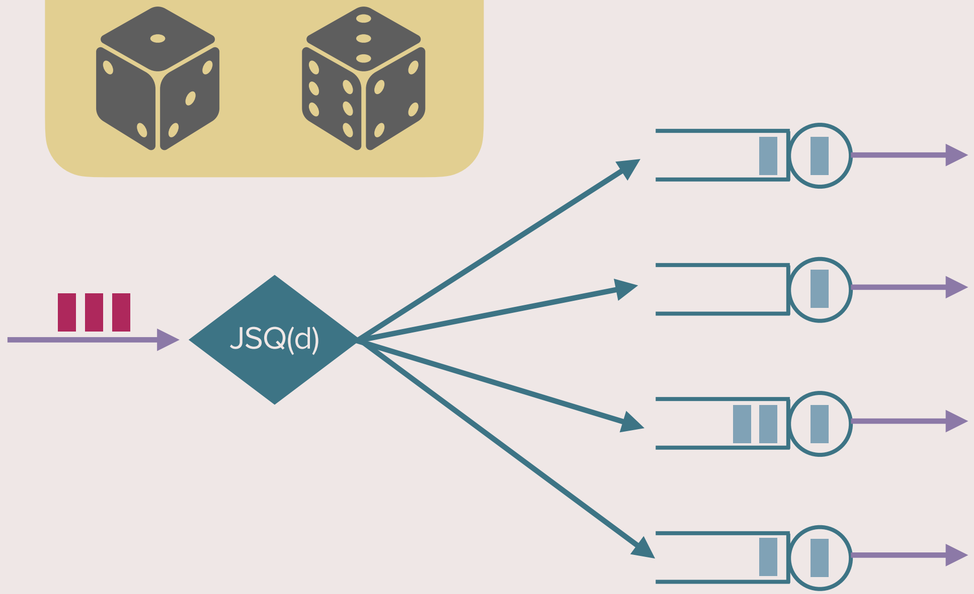


Power of d choices, or JSQ(d):
Given an integer $d \in [1,n] \cap \mathbb{Z}$, in each time slot:

(1) Select d servers uniformly at random

(2) Route arrivals to the shortest queue among those d

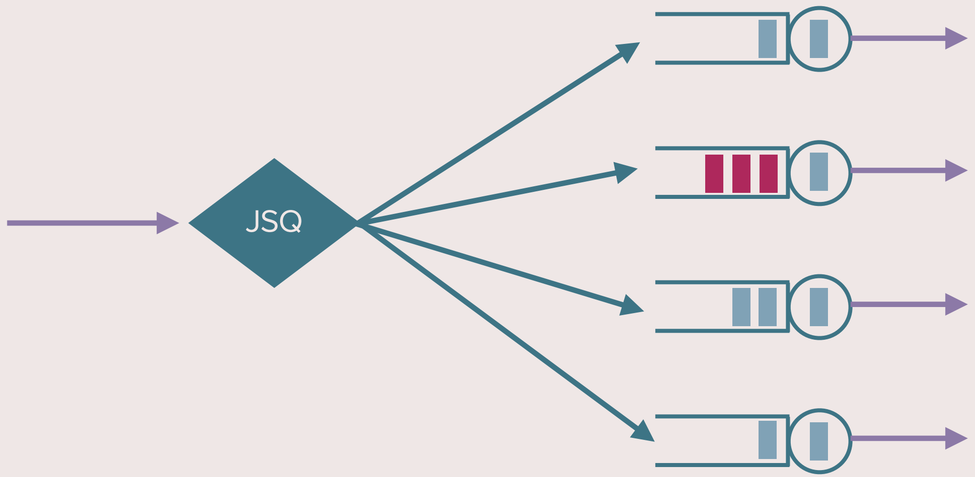
Example: $d = 2$



Routing Algorithms

Random routing:
Route arrivals to each queue uniformly at random

Join the Shortest Queue (JSQ):
Route arrivals to the queue with the smallest number of jobs.

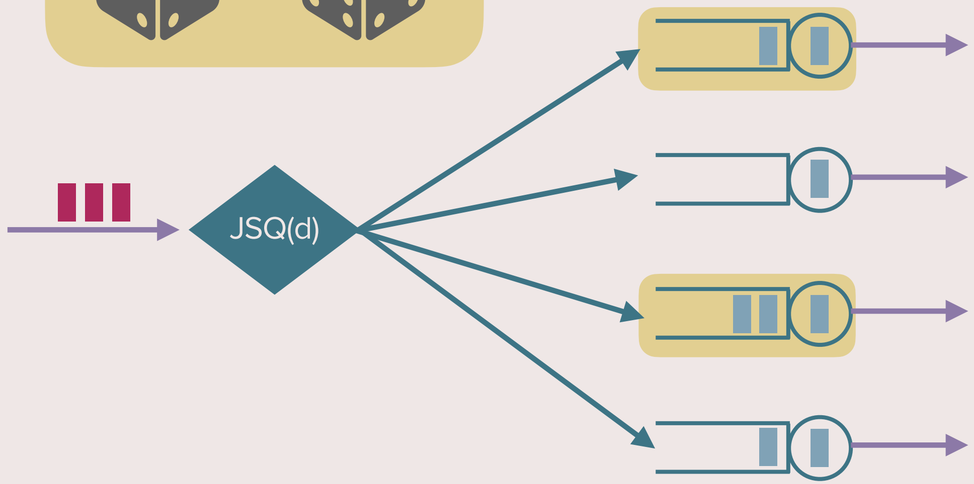


Power of d choices, or JSQ(d):
Given an integer $d \in [1,n] \cap \mathbb{Z}$, in each time slot:

(1) Select d servers uniformly at random

(2) Route arrivals to the shortest queue among those d

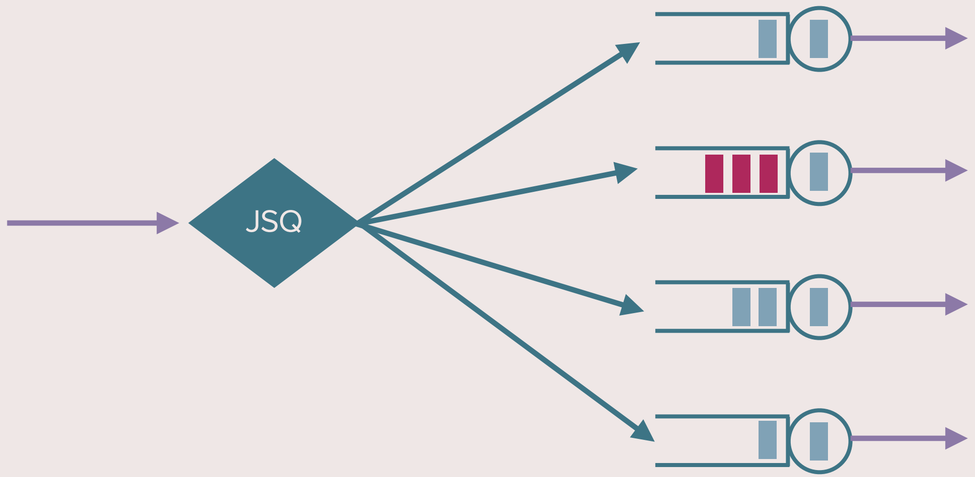
Example: $d = 2$



Routing Algorithms

Random routing:
Route arrivals to each queue uniformly at random

Join the Shortest Queue (JSQ):
Route arrivals to the queue with the smallest number of jobs.

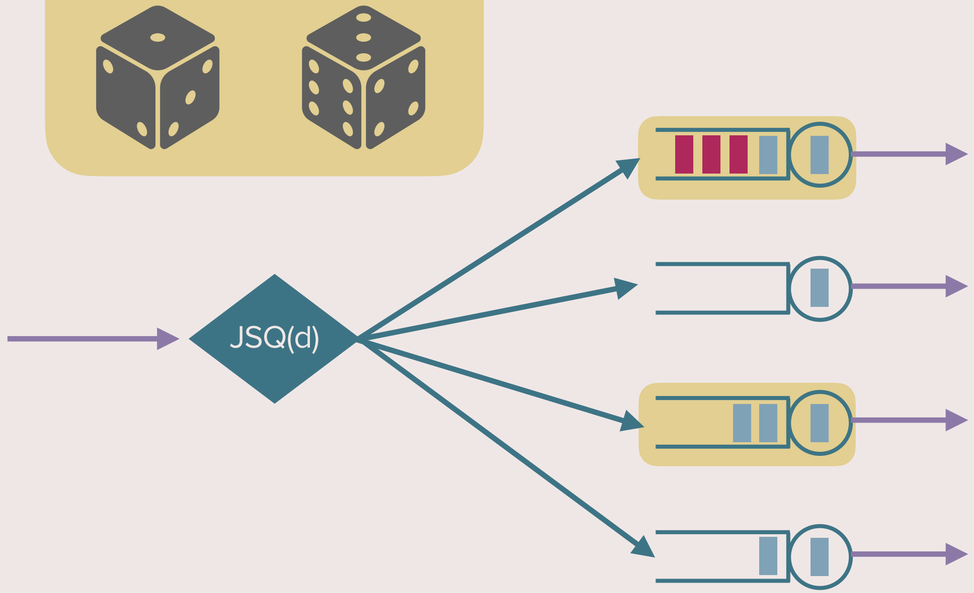


Power of d choices, or JSQ(d):
Given an integer $d \in [1,n] \cap \mathbb{Z}$, in each time slot:

(1) Select d servers uniformly at random

(2) Route arrivals to the shortest queue among those d

Example: $d = 2$



Routing Algorithms

Random routing:
Route arrivals to each queue uniformly at random

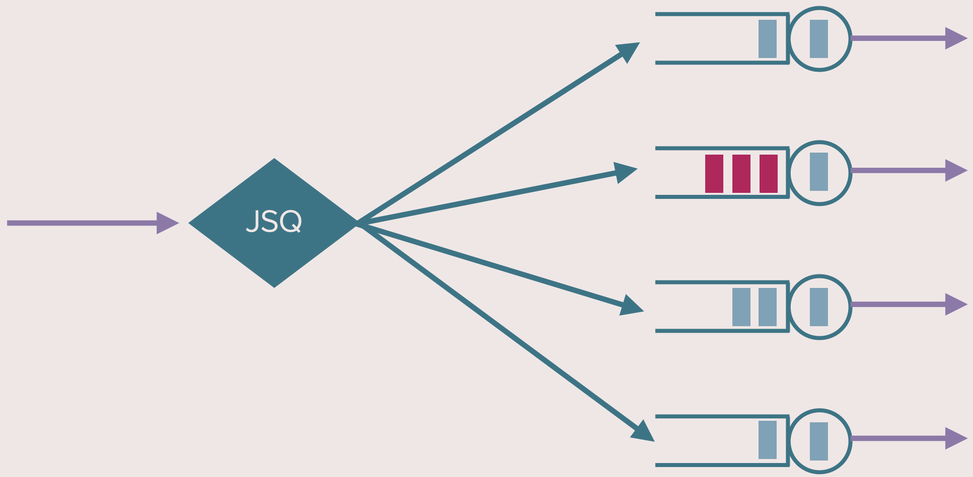
$d = 1$

Power of d choices, or JSQ(d):
Given an integer $d \in [1,n] \cap \mathbb{Z}$, in each time slot:

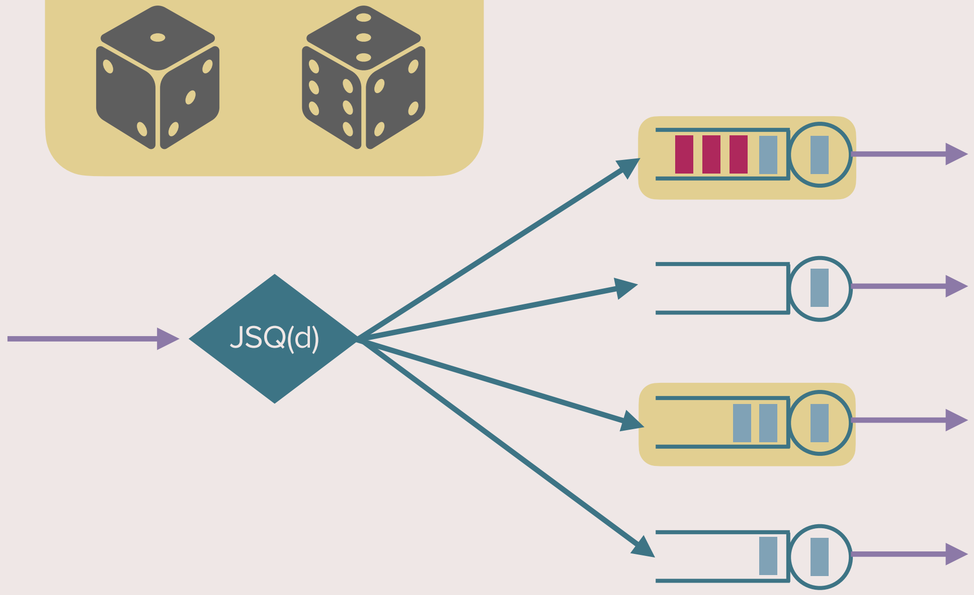
(1) Select d servers uniformly at random

(2) Route arrivals to the shortest queue among those d

Join the Shortest Queue (JSQ):
Route arrivals to the queue with the smallest number of jobs.



Example: $d = 2$



Routing Algorithms

Random routing:
Route arrivals to each queue uniformly at random

$d = 1$

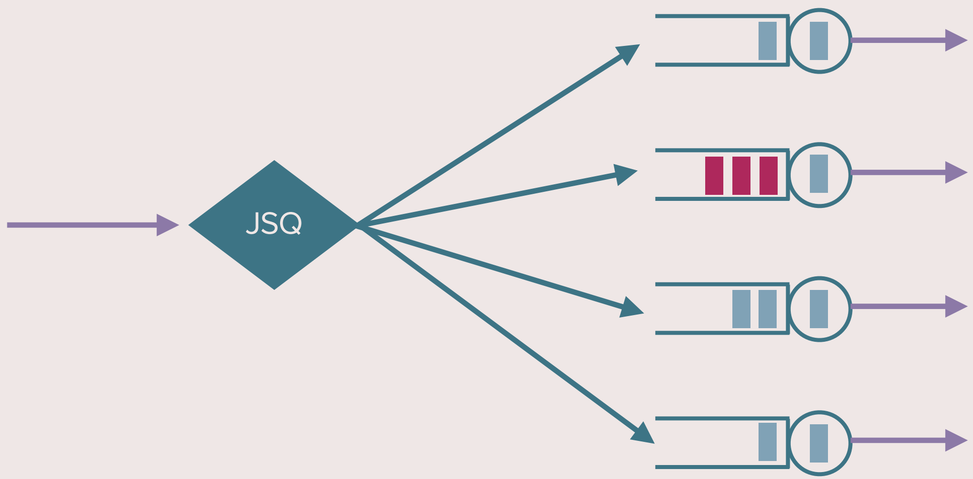
$d = n$

Join the Shortest Queue (JSQ):
Route arrivals to the queue with the smallest number of jobs.

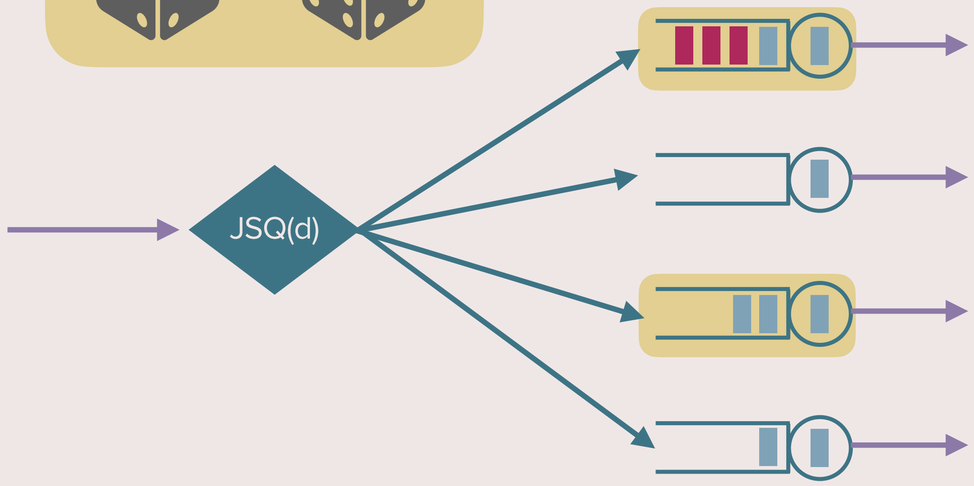
Power of d choices, or JSQ(d):
Given an integer $d \in [1,n] \cap \mathbb{Z}$, in each time slot:

(1) Select d servers uniformly at random

(2) Route arrivals to the shortest queue among those d

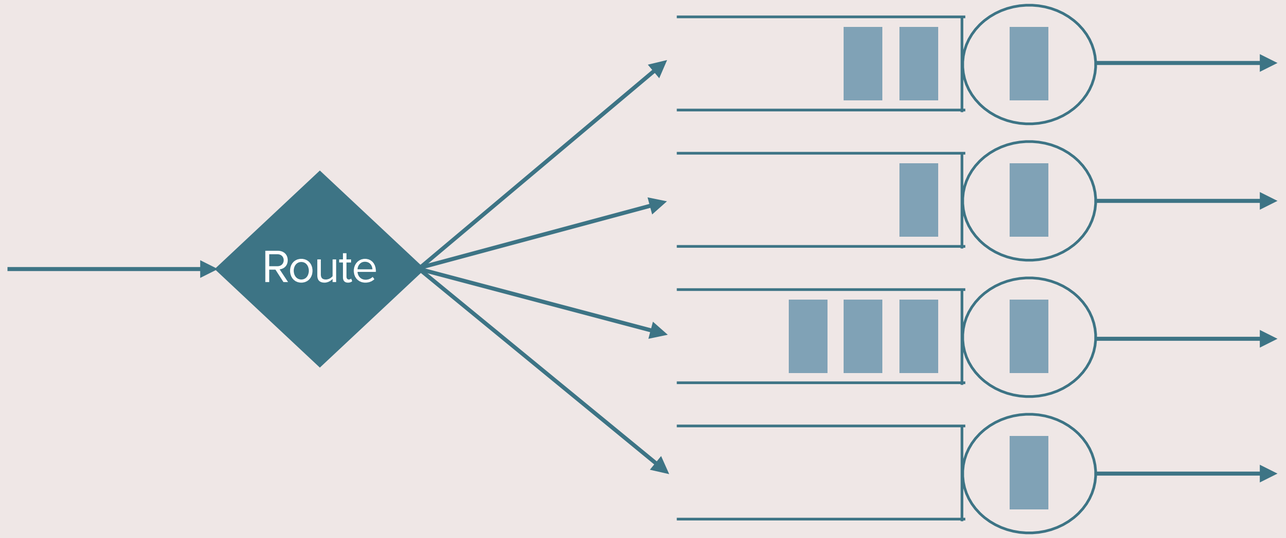


Example: $d = 2$

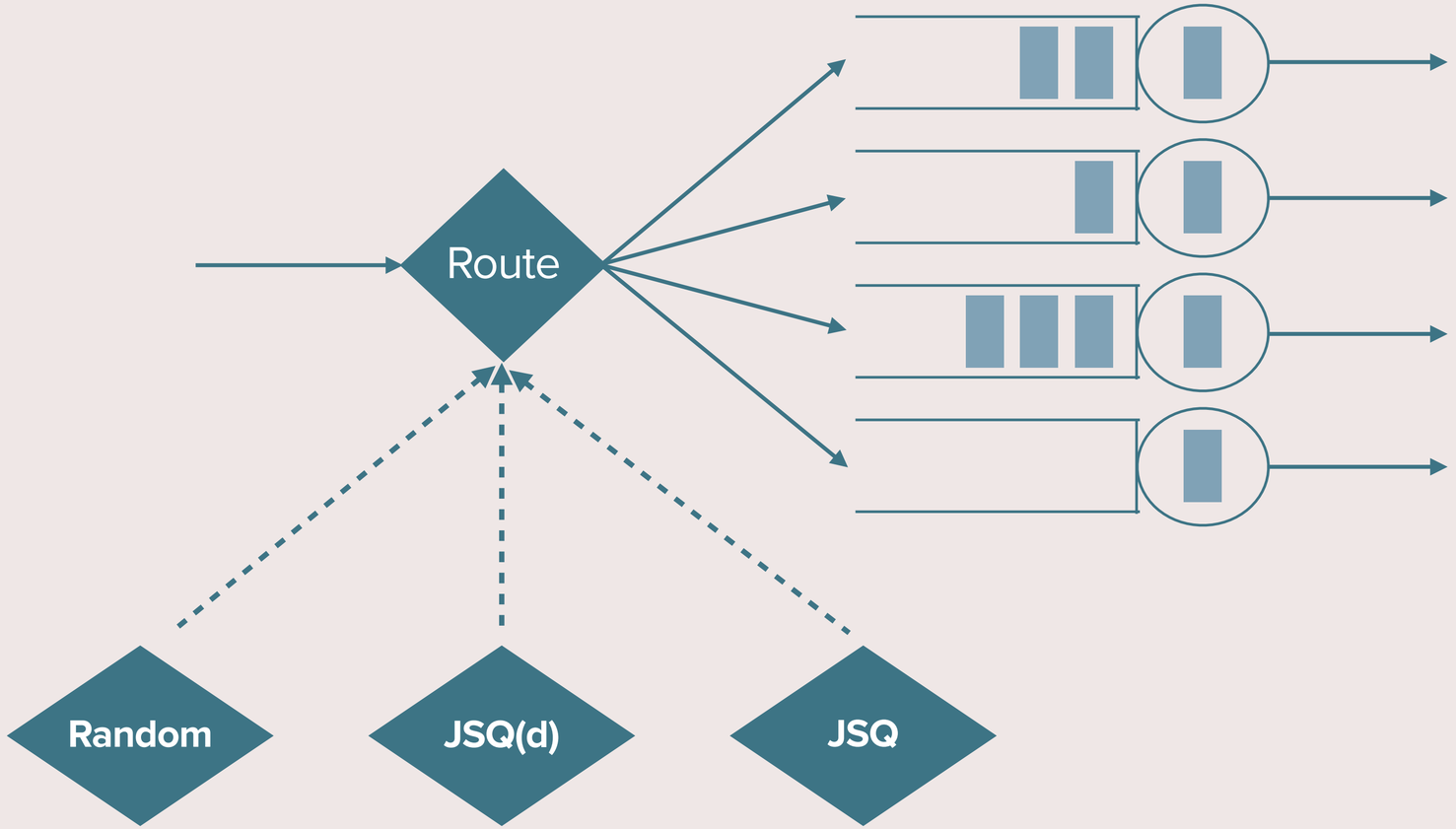


Goal Today

Goal Today

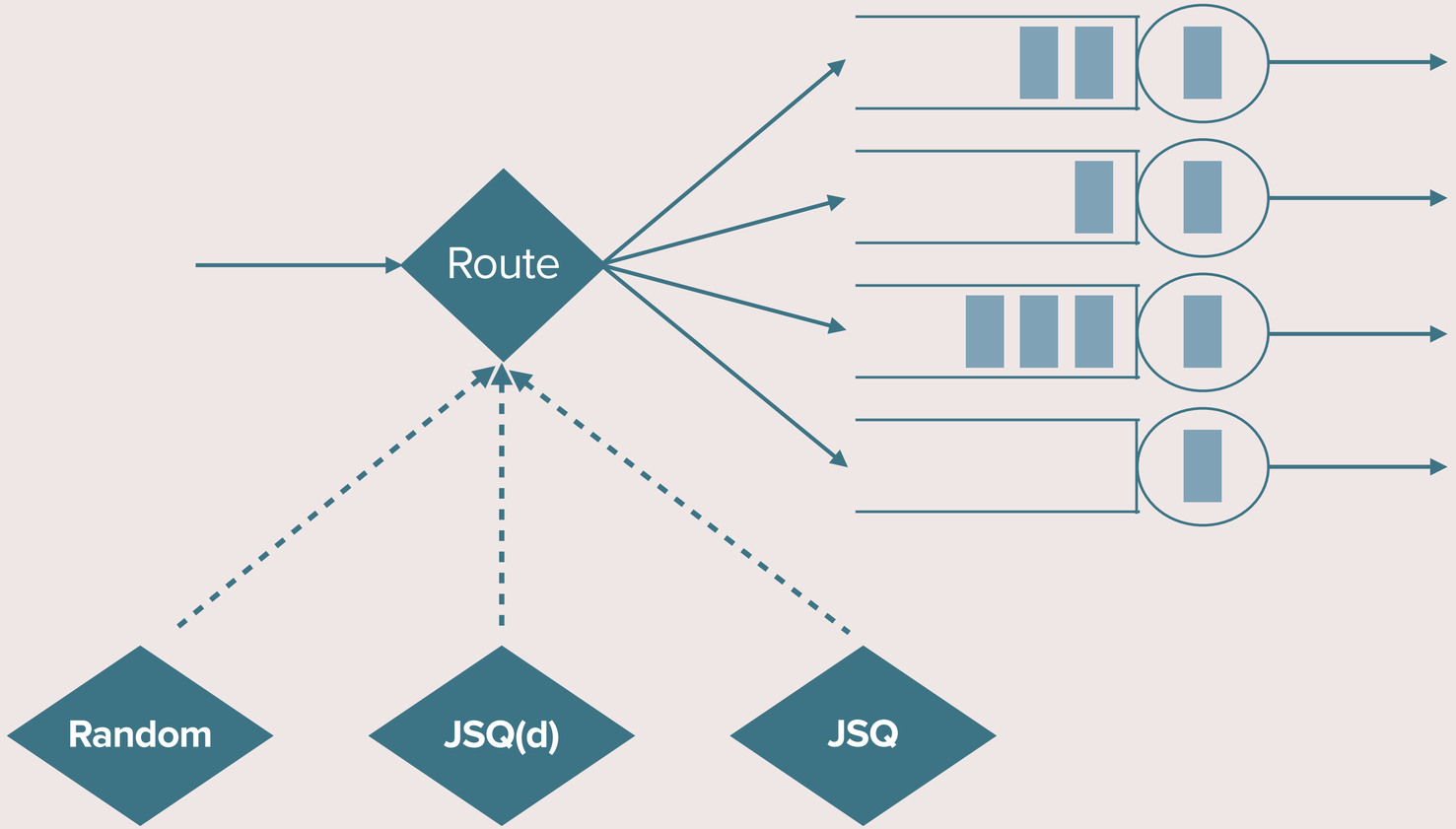


Goal Today



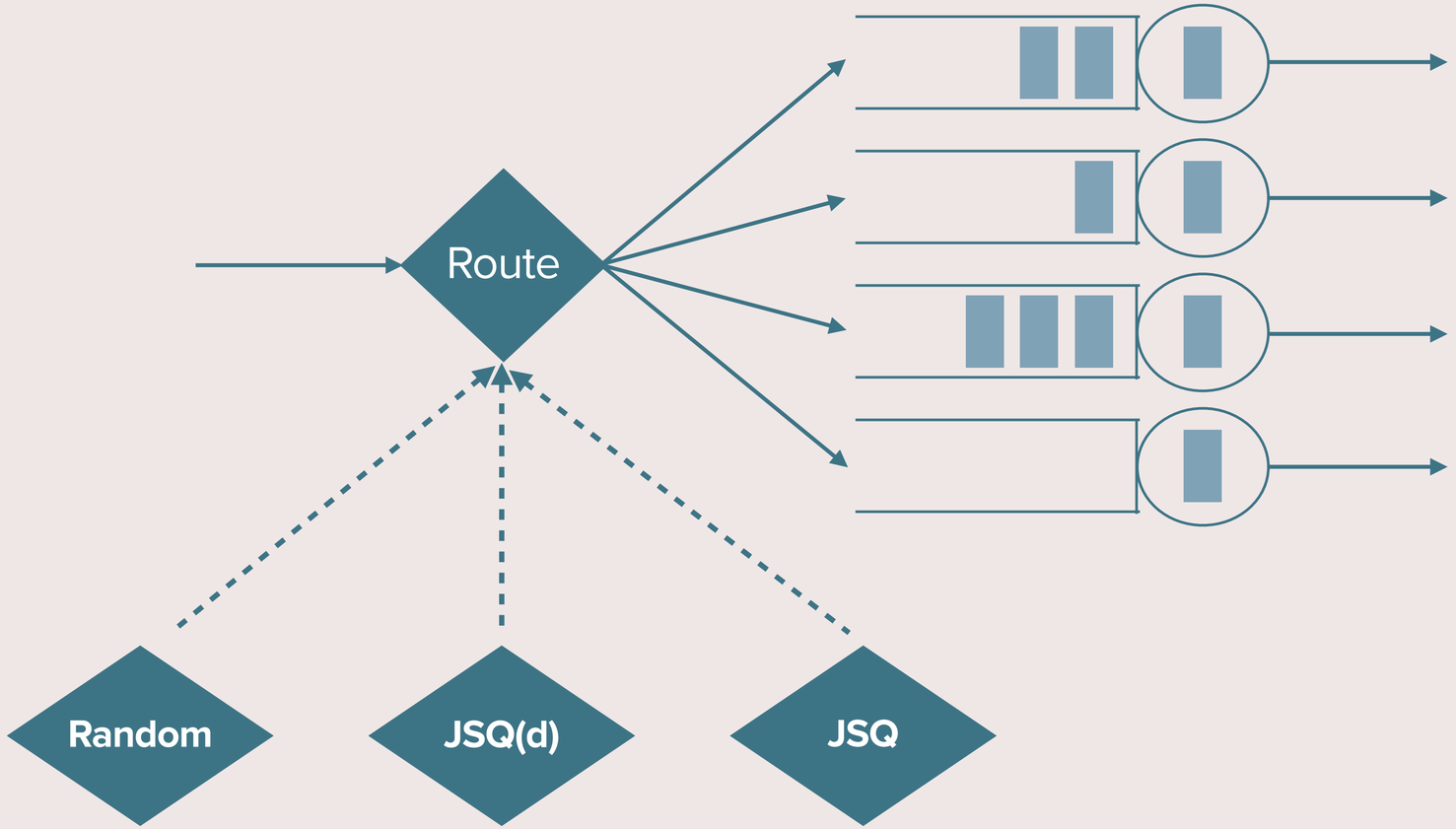
Goal Today

- Which one is better?



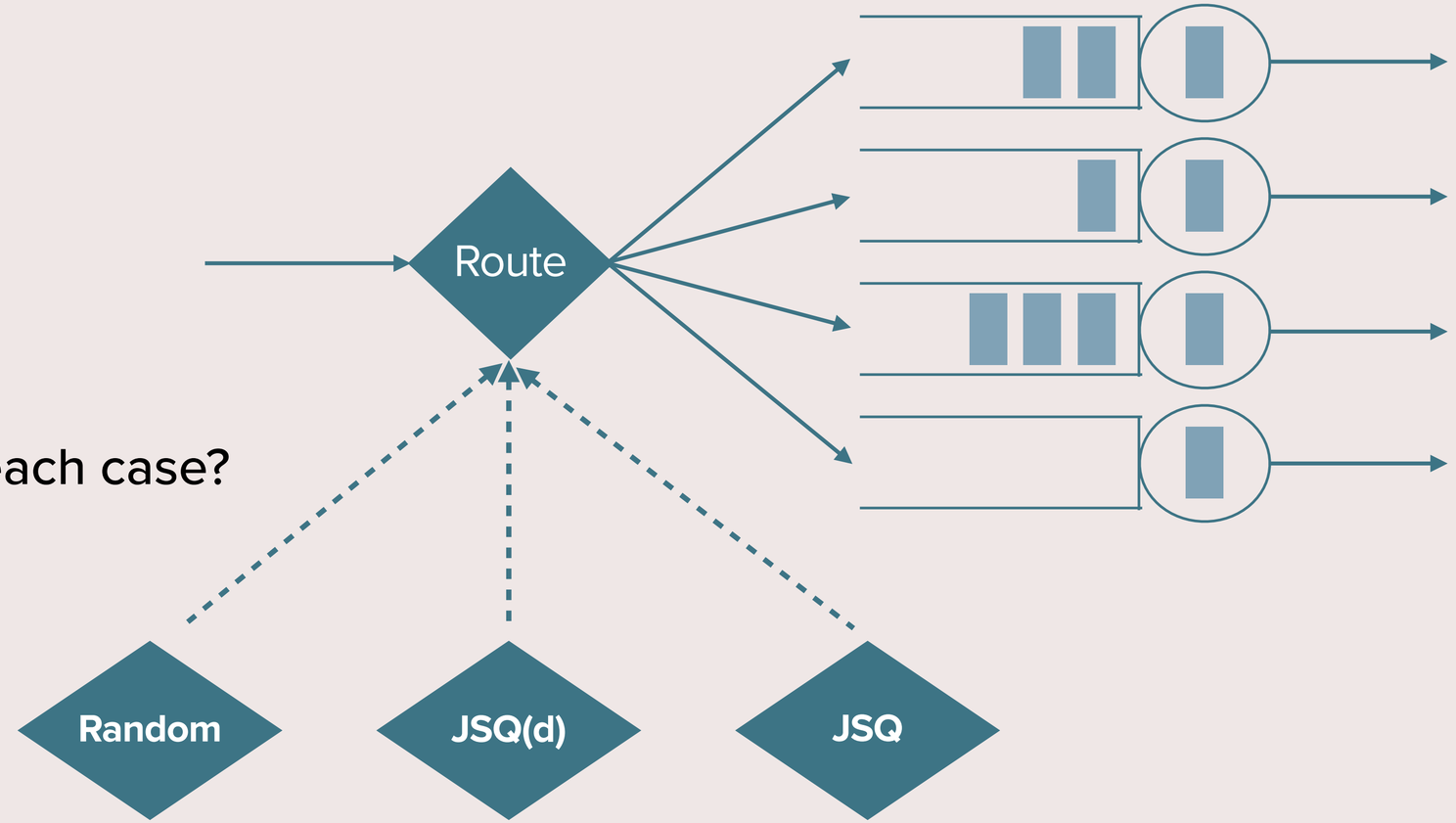
Goal Today

- Which one is better?
- Is any of these optimal?



Goal Today

- Which one is better?
- Is any of these optimal?
- Distribution of queue lengths in each case?



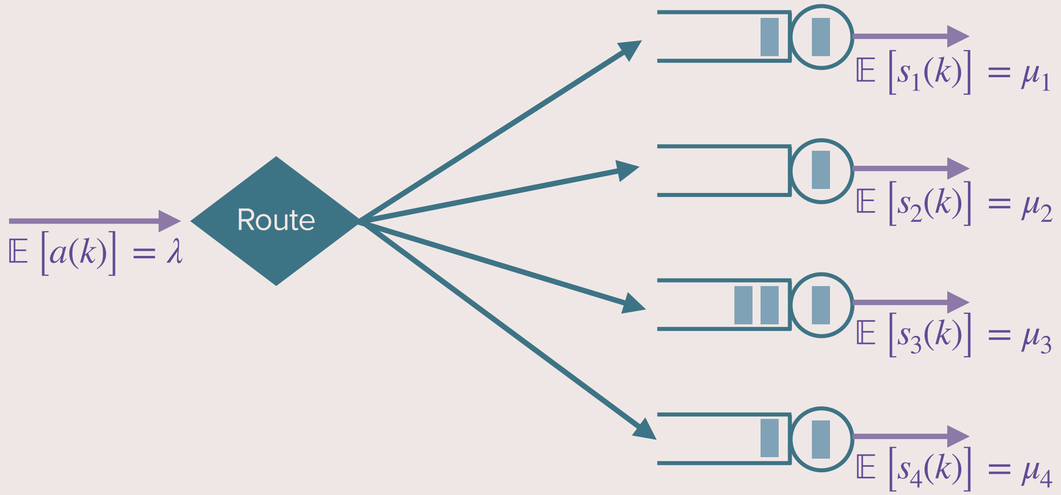
Outline

Part 1: The Model	• Important parts of a queueing model • Supermarket-checkout model • Some routing algorithms
Part 2: How to Solve?	• Stability • Heavy-Traffic: The CRP condition • Analysis of a single-server queue • The MGF method
Part 3: An Open Question	• Many-server heavy-traffic regime • Comparison of routing algorithms open question
Conclusion and future work	

Outline

Part 1: **The Model**	• Important parts of a queueing model • Supermarket-checkout model • Some routing algorithms
Part 2: **How to Solve?**	• Stability • Heavy-Traffic: The CRP condition • Analysis of a single-server queue • The MGF method
Part 3: **An Open Question**	• Many-server heavy-traffic regime • Comparison of routing algorithms open question
Conclusion and future work	

Stability



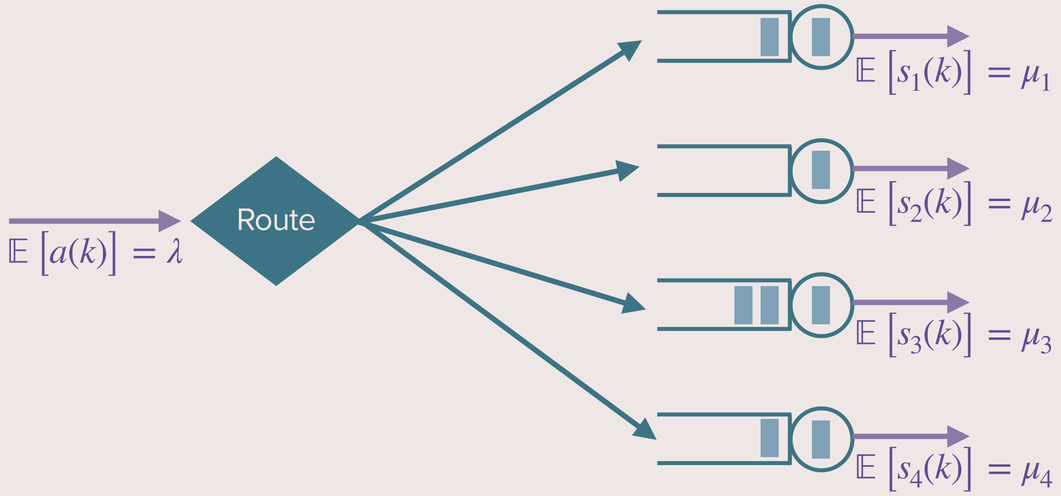
$$\mu_\Sigma = \sum_{i=1}^{N} \mu_i$$

Stability

Definition: Capacity region

Set of **arrival rates** so that there **exists** a routing algorithm such that the queue lengths are positive recurrent (stable), i.e., so that

$$\mathbb{E}\left[\sum_i q_i(k)\right] < \infty$$



$$\mu_\Sigma = \sum_{i=1}^{N} \mu_i$$

Stability

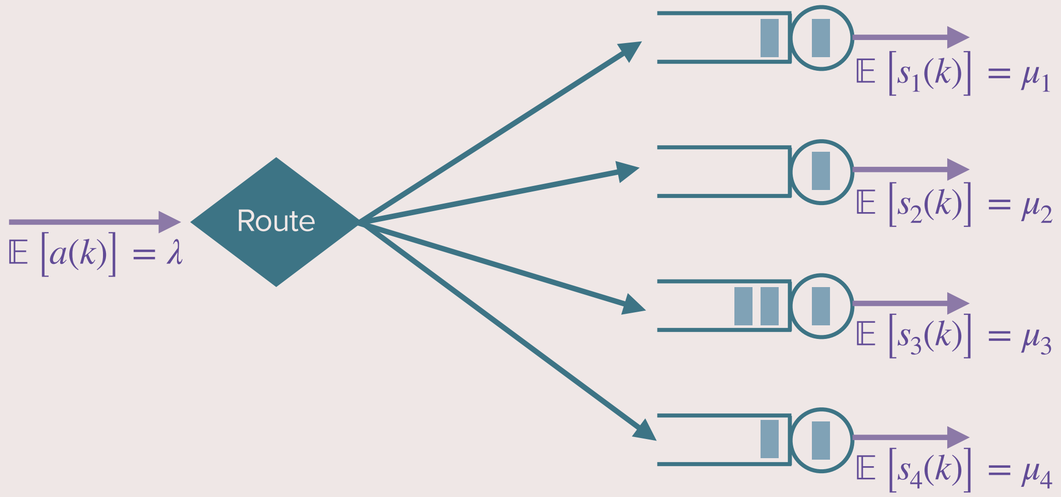
Definition: Capacity region

Set of **arrival rates** so that there **exists** a routing algorithm such that the queue lengths are positive recurrent (stable), i.e., so that

$$\mathbb{E}\left[\sum_i q_i(k)\right] < \infty$$

Supermarket Checkout System:

$$\mathscr{C} = \left\{\lambda \in \mathbb{R}_+ : \lambda \leq \mu_\Sigma\right\}$$



$$\mu_\Sigma = \sum_{i=1}^{N} \mu_i$$

Stability

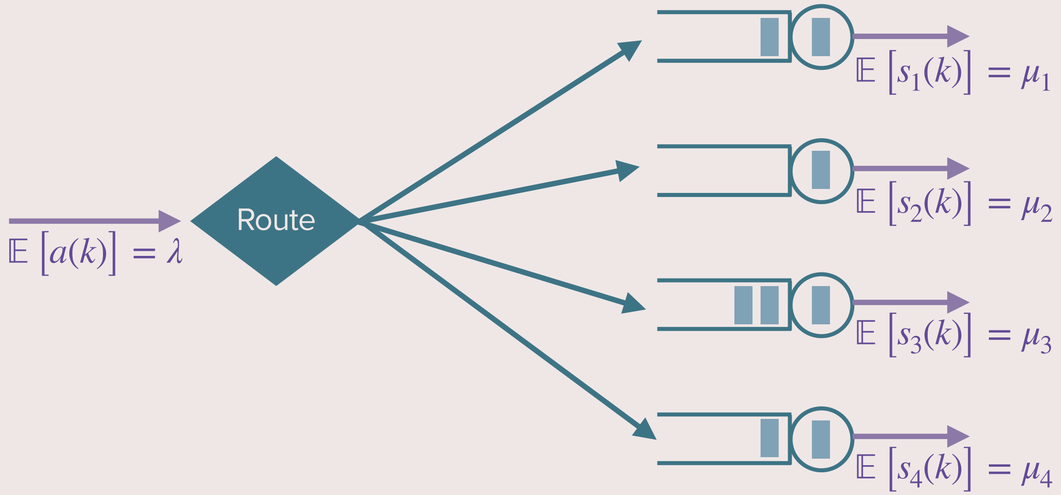
Definition: Capacity region

Set of **arrival rates** so that there **exists** a routing algorithm such that the queue lengths are positive recurrent (stable), i.e., so that

$$\mathbb{E}\left[\sum_i q_i(k)\right] < \infty$$

Supermarket Checkout System:

$$\mathscr{C} = \left\{\lambda \in \mathbb{R}_+ : \lambda \leq \mu_\Sigma\right\}$$



$$\mu_\Sigma = \sum_{i=1}^{N} \mu_i$$

Definition: Throughput optimal

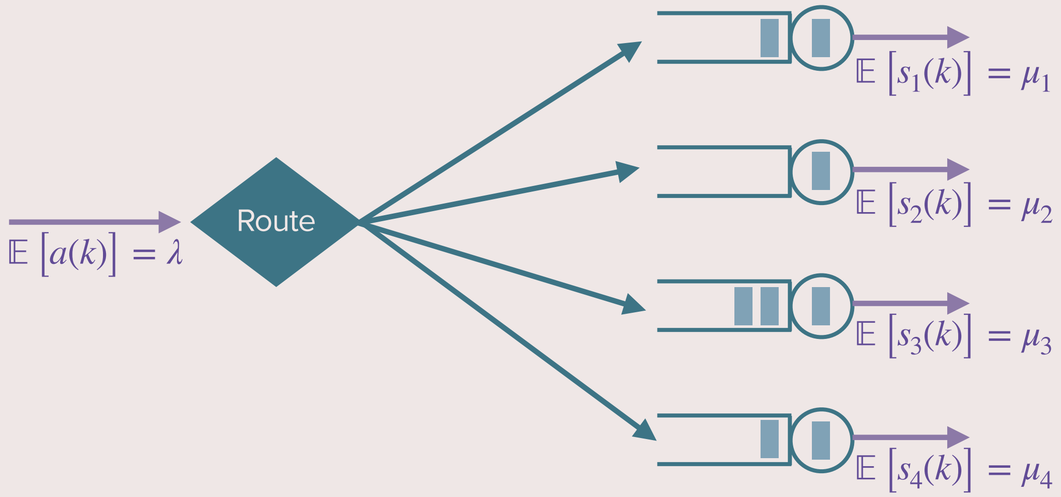
A **routing algorithm** $\mathscr{A}$ is throughput optimal, if the system operating under $\mathscr{A}$ is **stable for all** $\lambda \in Int(\mathscr{C})$

Stability

Definition: Capacity region

Set of **arrival rates** so that there **exists** a routing algorithm such that the queue lengths are positive recurrent (stable), i.e., so that

$$\mathbb{E}\left[\sum_i q_i(k)\right] < \infty$$



Supermarket Checkout System:

$$\mathscr{C} = \left\{\lambda \in \mathbb{R}_+ : \lambda \leq \mu_\Sigma\right\}$$

$$\mu_\Sigma = \sum_{i=1}^{N} \mu_i$$

Definition: Throughput optimal

A **routing algorithm** $\mathscr{A}$ is throughput optimal, if the system operating under $\mathscr{A}$ is **stable for all** $\lambda \in Int(\mathscr{C})$

Literature:

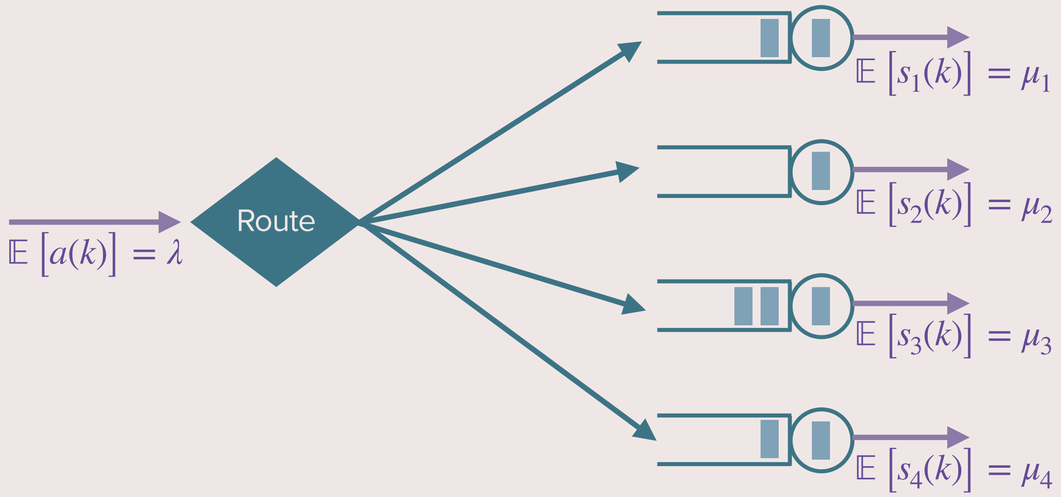
- **Random:** Only if all the servers are equal
- **JSQ:** Always
- **JSQ(d):** If all servers are equal

Stability

Definition: Capacity region

Set of **arrival rates** so that there **exists** a routing algorithm such that the queue lengths are positive recurrent (stable), i.e., so that

$$\mathbb{E}\left[\sum_i q_i(k)\right] < \infty$$



Supermarket Checkout System:

$$\mathscr{C} = \left\{\lambda \in \mathbb{R}_+ : \lambda \leq \mu_\Sigma\right\}$$

$$\mu_\Sigma = \sum_{i=1}^{N} \mu_i$$

Definition: Throughput optimal

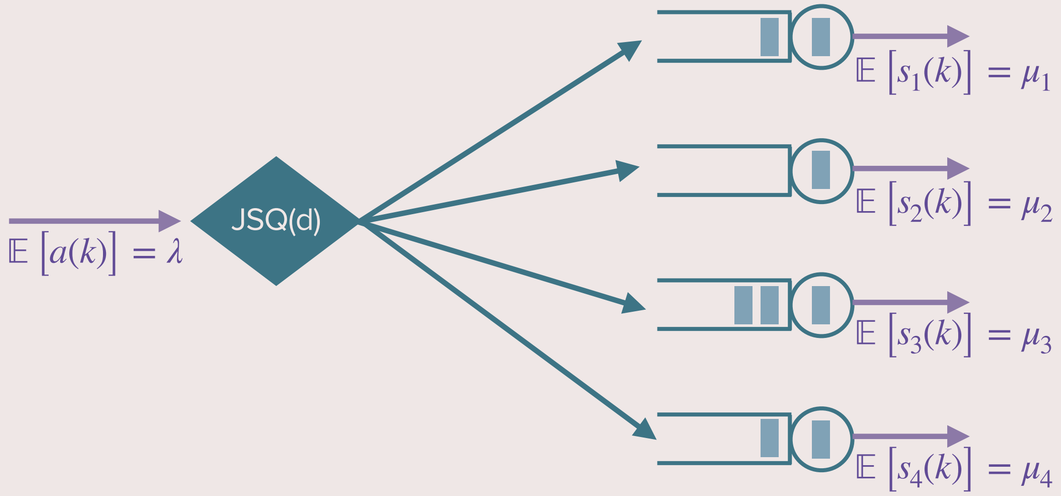
A **routing algorithm** $\mathscr{A}$ is throughput optimal, if the system operating under $\mathscr{A}$ is **stable for all** $\lambda \in Int(\mathscr{C})$

Literature:

- **Random:** Only if all the servers are equal
- **JSQ:** Always
- **JSQ(d):** If all servers are equal

Should it be better than random?

JSQ(d) When Servers are Different

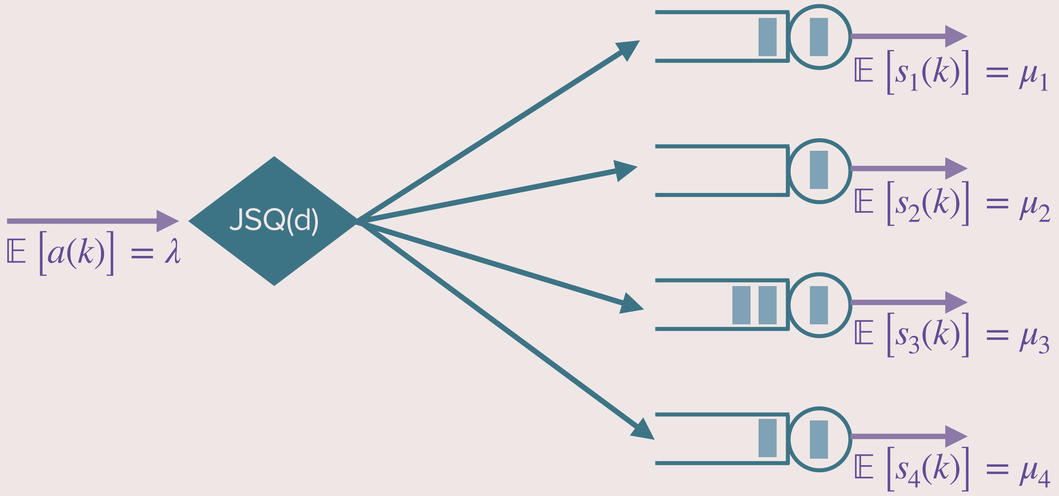


JSQ(d) When Servers are Different

Theorem: [HL, Maguluri 2020]

Let $1 \leq d \leq n-1$. Define $x_i = \dfrac{\mu_i}{\mu_\Sigma}$, and $y_i = \dfrac{\binom{i-1}{d-1}}{\binom{n}{d}}$.

Power-of-d is throughput optimal if and only if $\boldsymbol{x} \preccurlyeq \boldsymbol{y}$



JSQ(d) When Servers are Different

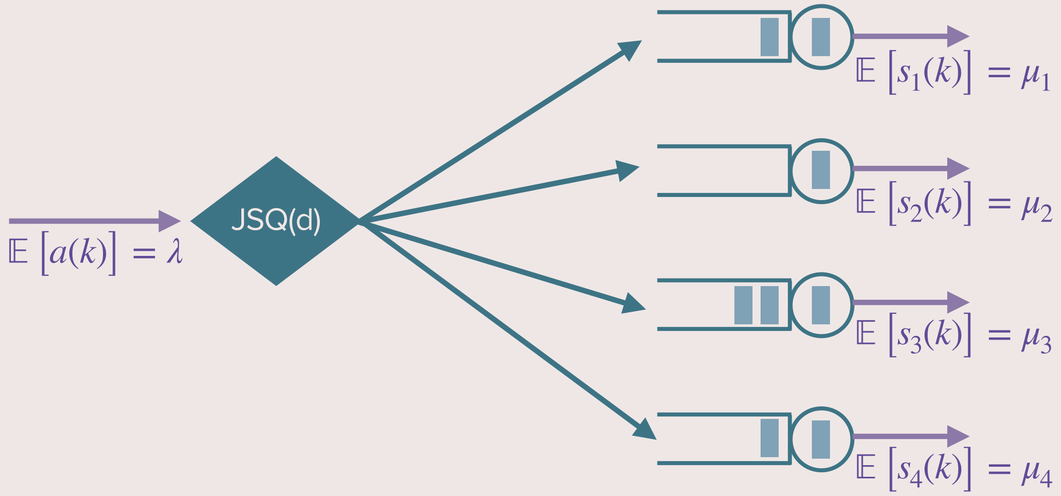
Theorem: [HL, Maguluri 2020]

Let $1 \leq d \leq n-1$. Define $x_i = \dfrac{\mu_i}{\mu_\Sigma}$, and $y_i = \dfrac{\binom{i-1}{d-1}}{\binom{n}{d}}$.

Power-of-d is throughput optimal if and only if $\boldsymbol{x} \preccurlyeq \boldsymbol{y}$

Majorization: Let $x_{(i)}$ be the i^{th} smallest component of $\boldsymbol{x} \in \mathbb{R}^n$. Then, the notation $\boldsymbol{x} \preccurlyeq \boldsymbol{y}$ means that

$$\sum_{i=j}^{n} x_{(i)} \leq \sum_{i=j}^{n} y_{(i)} \quad \forall j \in [n]$$



JSQ(d) When Servers are Different

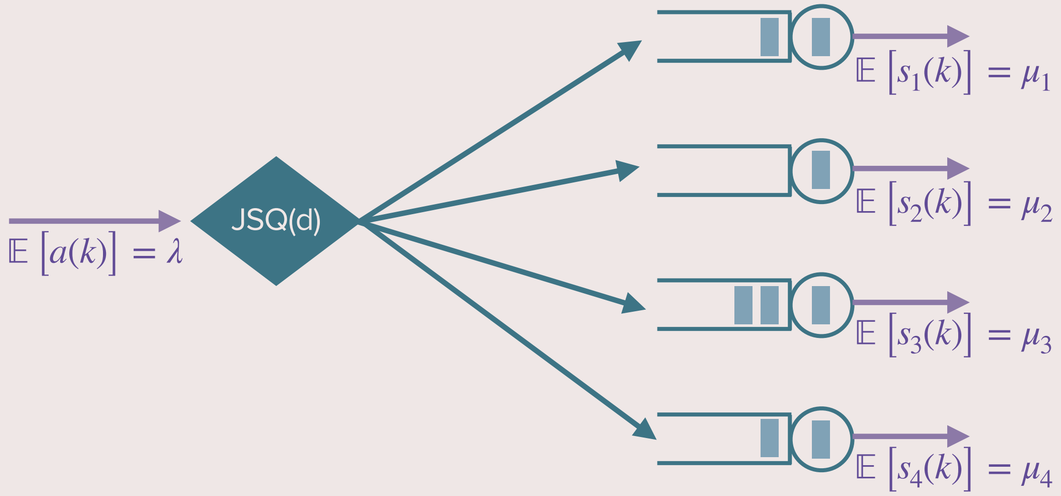
Theorem: [HL, Maguluri 2020]

Let $1 \leq d \leq n-1$. Define $x_i = \dfrac{\mu_i}{\mu_\Sigma}$, and $y_i = \dfrac{\binom{i-1}{d-1}}{\binom{n}{d}}$.

Power-of-d is throughput optimal if and only if $\boldsymbol{x} \preccurlyeq \boldsymbol{y}$

Majorization: Let $x_{(i)}$ be the i^{th} smallest component of $\boldsymbol{x} \in \mathbb{R}^n$. Then, the notation $\boldsymbol{x} \preccurlyeq \boldsymbol{y}$ means that

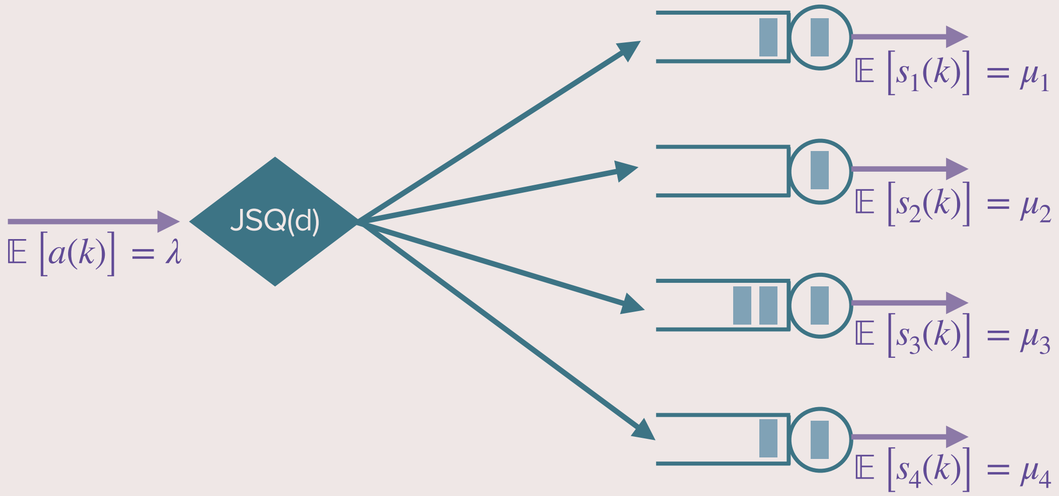
$$\sum_{i=j}^{n} x_{(i)} \leq \sum_{i=j}^{n} y_{(i)} \quad \forall j \in [n]$$



Then, the necessary and sufficient condition is that

$$\frac{\sum_{i=1}^{j} \mu_{(i)}}{\mu_\Sigma} \geq \frac{\binom{j}{d}}{\binom{n}{d}} \qquad \forall d \leq j \leq n-1$$

Interpretation of the Result



Theorem: JSQ(d) is throughput optimal if and only if

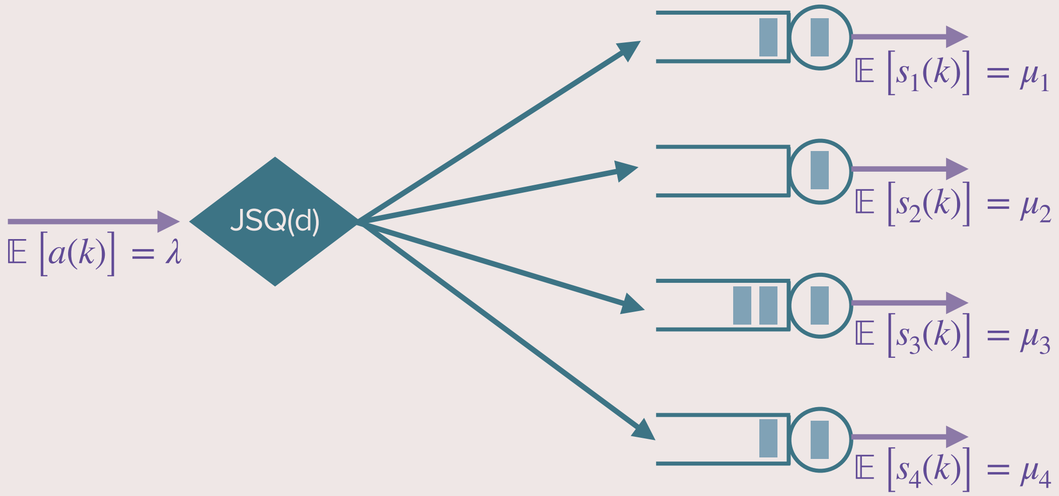
$$\frac{\sum_{i=1}^{j} \mu_{(i)}}{\mu_\Sigma} \geq \frac{\binom{j}{d}}{\binom{n}{d}} \qquad \forall d \leq j \leq n-1$$

Interpretation of the Result

- **If $d = 1$ (random)**

$$\implies RHS = \frac{j}{n}$$

$$\implies \mu_i = \mu_j \quad \forall i, j$$



Theorem: JSQ(d) is throughput optimal if and only if

$$\frac{\sum_{i=1}^{j} \mu_{(i)}}{\mu_\Sigma} \geq \frac{\binom{j}{d}}{\binom{n}{d}} \qquad \forall d \leq j \leq n-1$$

Interpretation of the Result

- **If $d = 1$ (random)**

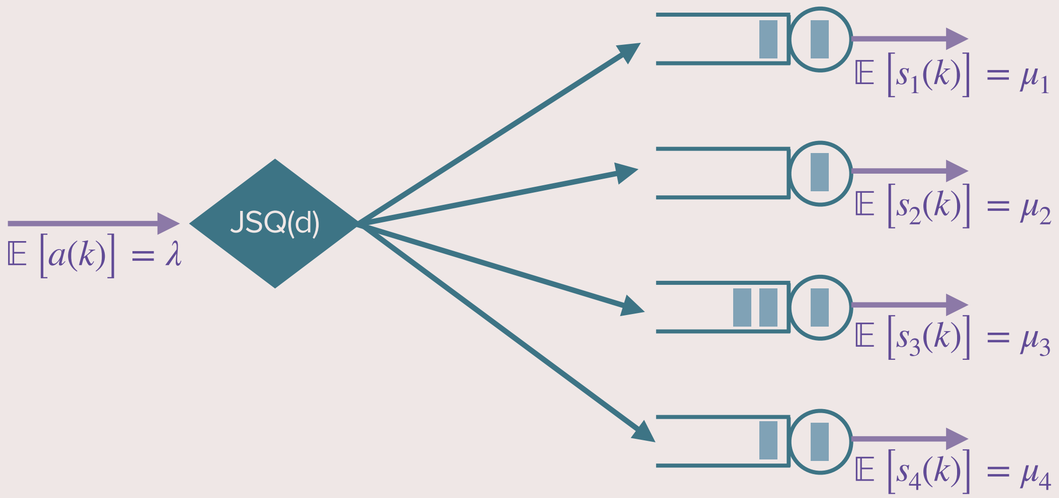
$$\implies RHS = \frac{j}{n}$$

$$\implies \mu_i = \mu_j \quad \forall i, j$$

- **If $d = n$ (JSQ)**

$$\implies RHS = 0$$

$$\implies \text{Any } \boldsymbol{\mu} \in \mathbb{R}_+^n$$



Theorem: JSQ(d) is throughput optimal if and only if

$$\frac{\sum_{i=1}^{j} \mu_{(i)}}{\mu_\Sigma} \geq \frac{\binom{j}{d}}{\binom{n}{d}} \qquad \forall d \leq j \leq n-1$$

Interpretation of the Result

- **If $d = 1$ (random)**

$$\implies RHS = \frac{j}{n}$$

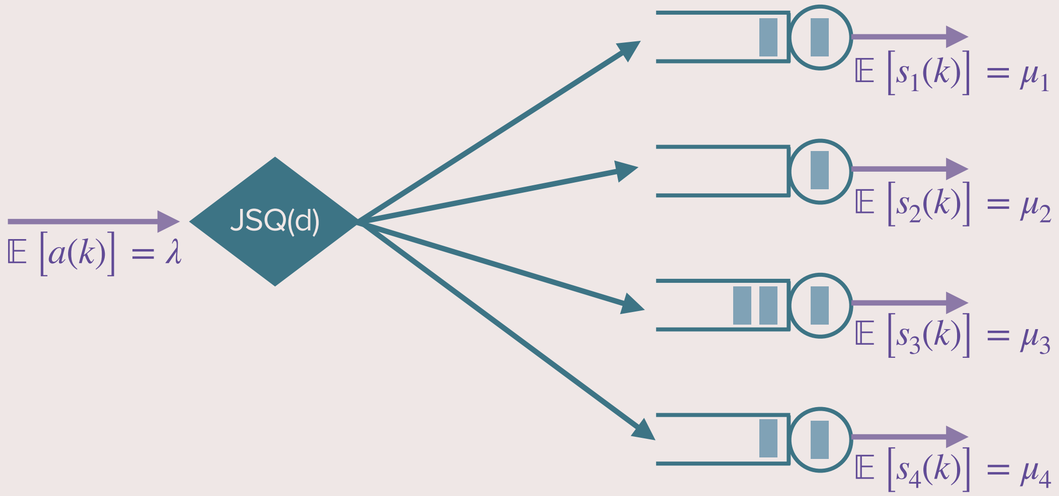
$$\implies \mu_i = \mu_j \quad \forall i, j$$

- **If $d = n$ (JSQ)**

$$\implies RHS = 0$$

$$\implies \text{Any } \boldsymbol{\mu} \in \mathbb{R}^n_+$$

- Faster servers should be sampled sufficiently often



Theorem: JSQ(d) is throughput optimal if and only if

$$\frac{\sum_{i=1}^{j} \mu_{(i)}}{\mu_\Sigma} \geq \frac{\binom{j}{d}}{\binom{n}{d}} \qquad \forall d \leq j \leq n-1$$

Interpretation of the Result

- **If $d = 1$ (random)**

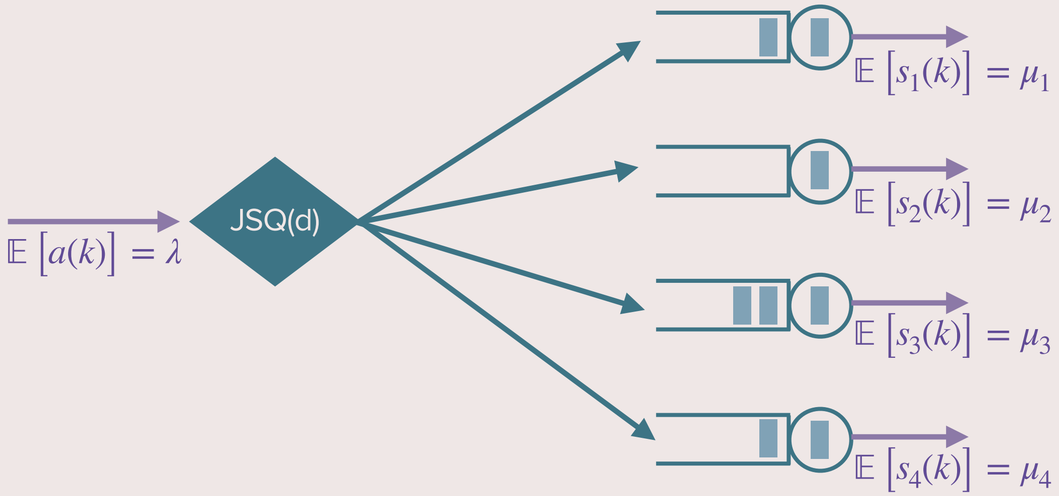
$$\implies RHS = \frac{j}{n}$$

$$\implies \mu_i = \mu_j \quad \forall i, j$$

- **If $d = n$ (JSQ)**

$$\implies RHS = 0$$

$$\implies \text{Any } \boldsymbol{\mu} \in \mathbb{R}^n_+$$



- Faster servers should be sampled sufficiently often
- Power-of-d samples **uniformly** at random, so we characterize the **amount of imbalance** that power-of-d can tolerate

Theorem: JSQ(d) is throughput optimal if and only if

$$\frac{\sum_{i=1}^{j} \mu_{(i)}}{\mu_\Sigma} \geq \frac{\binom{j}{d}}{\binom{n}{d}} \qquad \forall d \leq j \leq n-1$$

Interpretation of the Result

- **If $d = 1$ (random)**

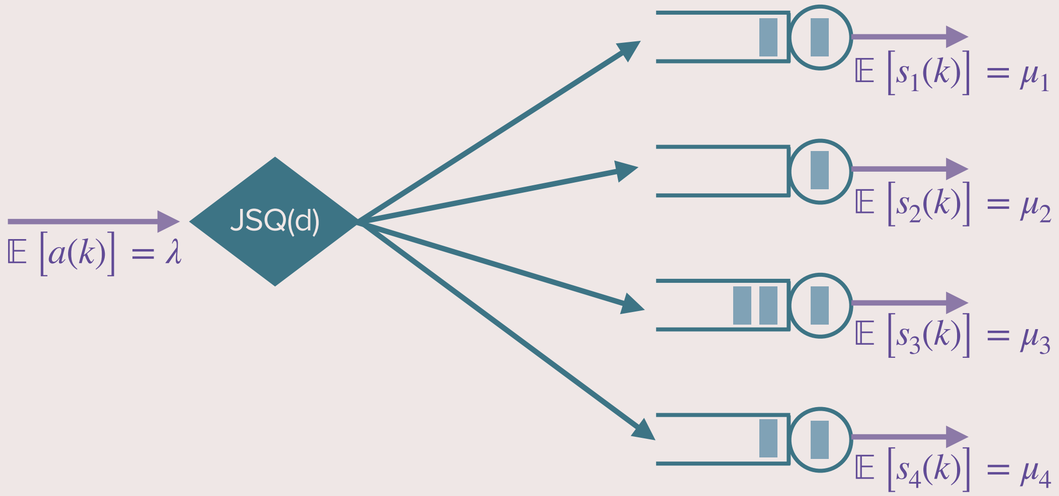
$$\implies RHS = \frac{j}{n}$$

$$\implies \mu_i = \mu_j \quad \forall i, j$$

- **If $d = n$ (JSQ)**

$$\implies RHS = 0$$

$$\implies \text{Any } \boldsymbol{\mu} \in \mathbb{R}_+^n$$

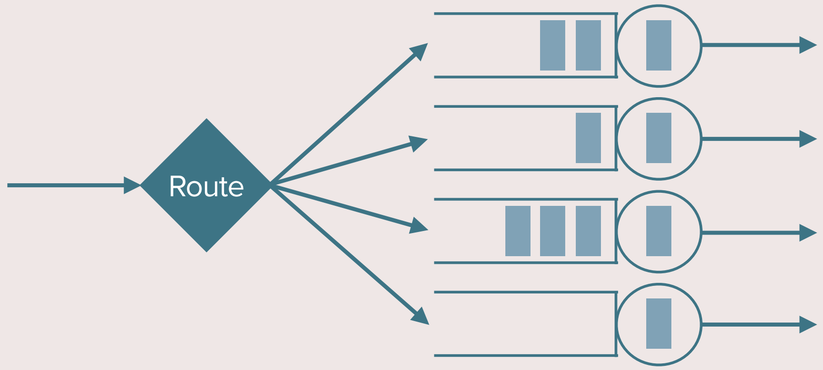


- Faster servers should be sampled sufficiently often
- Power-of-d samples **uniformly** at random, so we characterize the **amount of imbalance** that power-of-d can tolerate
- Fix n. As d gets larger, we can tolerate more imbalance.

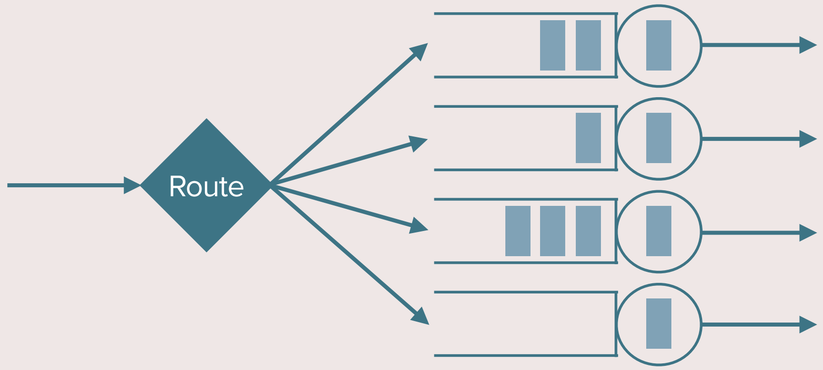
Theorem: JSQ(d) is throughput optimal if and only if

$$\frac{\sum_{i=1}^{j} \mu_{(i)}}{\mu_\Sigma} \geq \frac{\binom{j}{d}}{\binom{n}{d}} \qquad \forall d \leq j \leq n-1$$

Stability: Which One is Better?



Stability: Which One is Better?

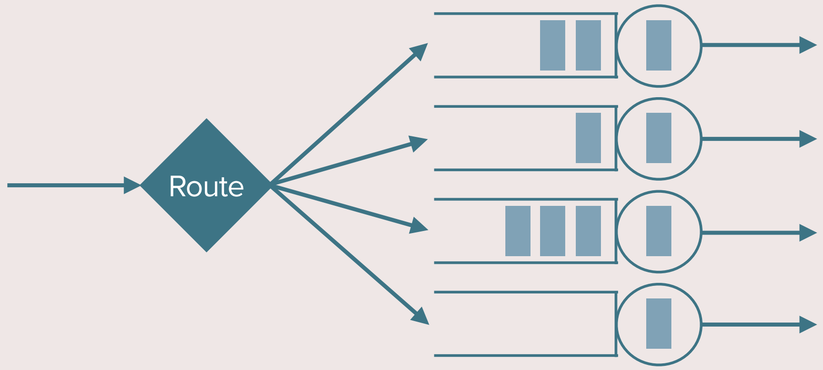


Random

JSQ(d)

JSQ

Stability: Which One is Better?



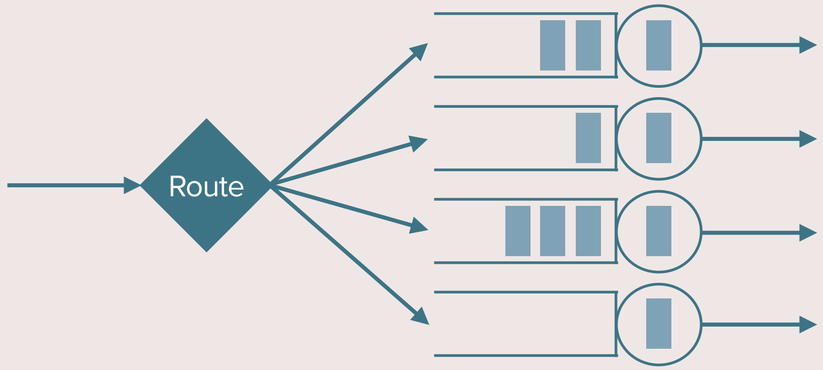
Random

Only if the service rates are equal

JSQ(d)

JSQ

Stability: Which One is Better?



Random

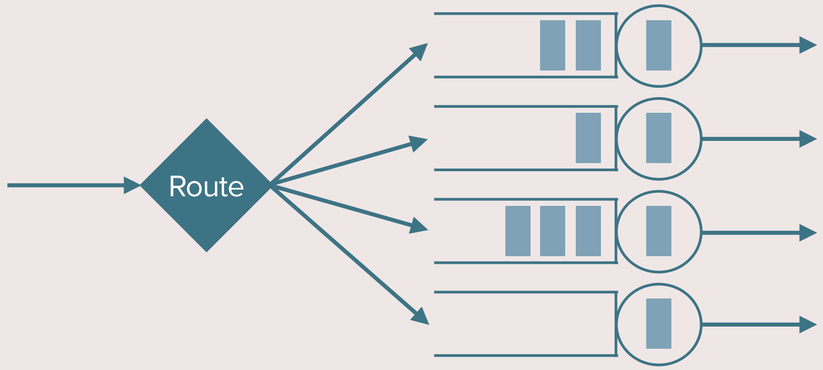
Only if the service rates are equal

JSQ(d)

JSQ

All service rates

Stability: Which One is Better?



Random

Only if the service rates are equal

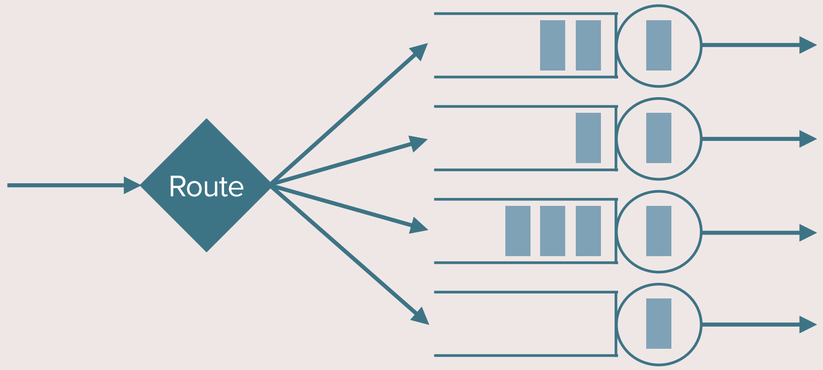
JSQ(d)

Majorization condition

JSQ

All service rates

Stability: Which One is Better?



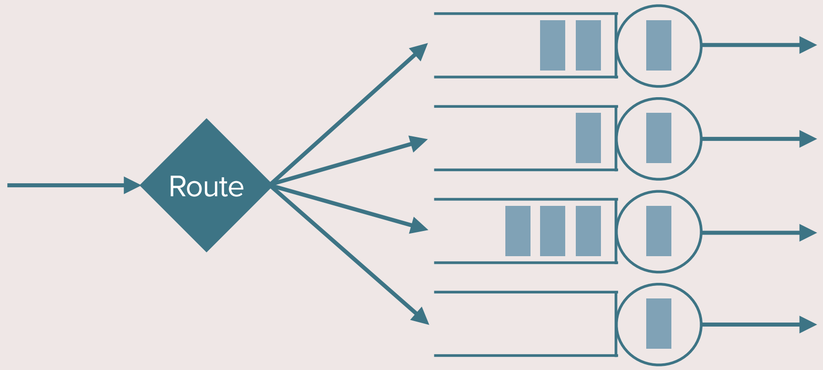
Random < **JSQ(d)** < **JSQ**

Only if the service rates are equal

Majorization condition

All service rates

Stability: Which One is Better?



Delay?

Asymptotic Analysis

Random < **JSQ(d)** < **JSQ**

Only if the service rates are equal

Majorization condition

All service rates

Heavy-Traffic Analysis

Heavy-Traffic Analysis

- Load the system close to maximum capacity:

Heavy-Traffic Analysis

- Load the system close to maximum capacity:

Arrival rate ≈ Service rate

Heavy-Traffic Analysis

- Load the system close to maximum capacity:

  Arrival rate ≈ Service rate

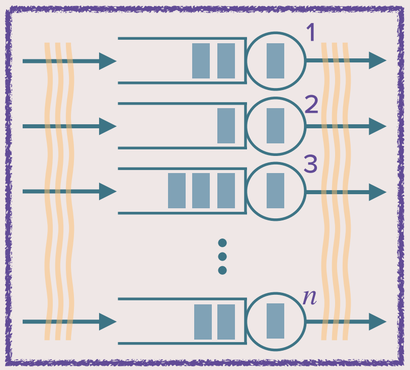
- State Space Collapse (SSC)

Heavy-Traffic Analysis

- Load the system close to maximum capacity:

  Arrival rate ≈ Service rate

- State Space Collapse (SSC)

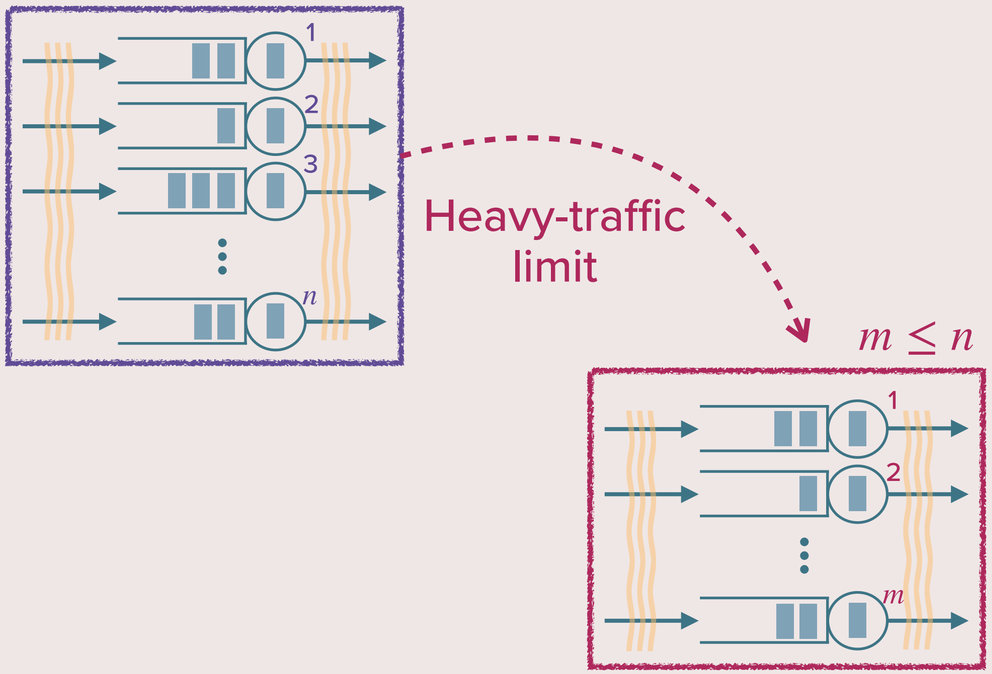


Heavy-Traffic Analysis

- Load the system close to maximum capacity:

Arrival rate ≈ Service rate

- State Space Collapse (SSC)

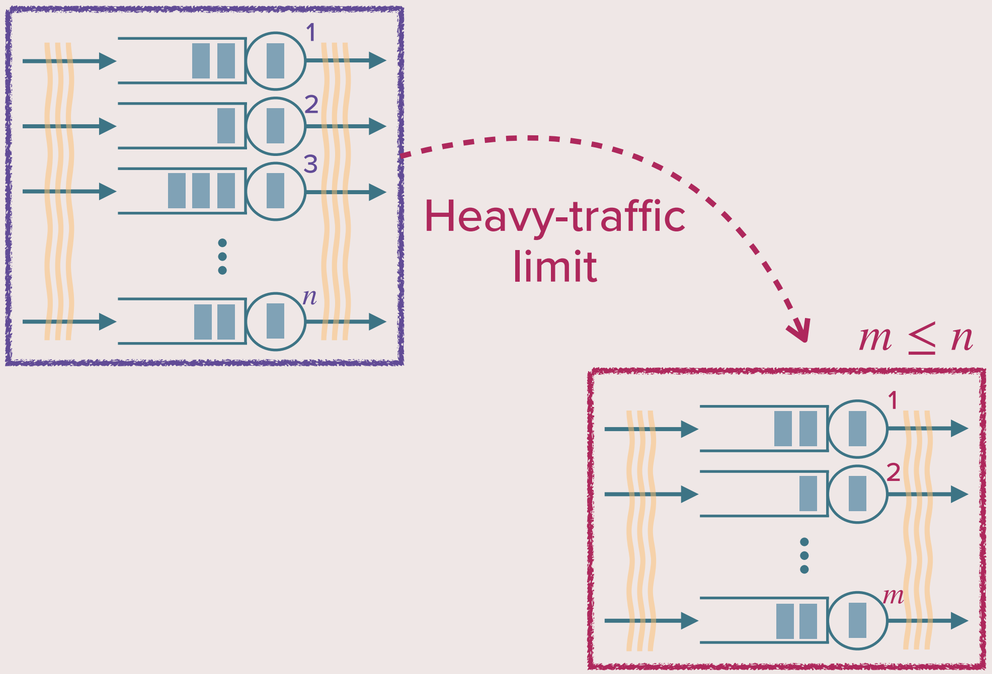


Heavy-Traffic Analysis

- Load the system close to maximum capacity:

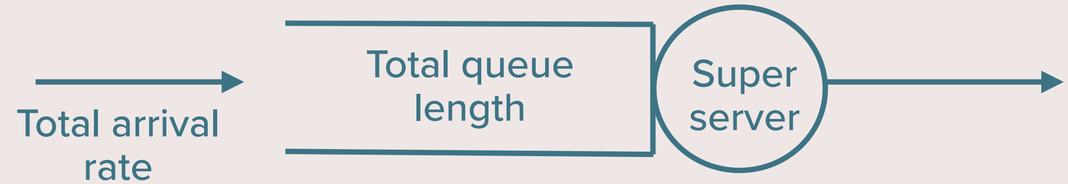
Arrival rate ≈ Service rate

- State Space Collapse (SSC)



Complete Resource Pooling (CRP):

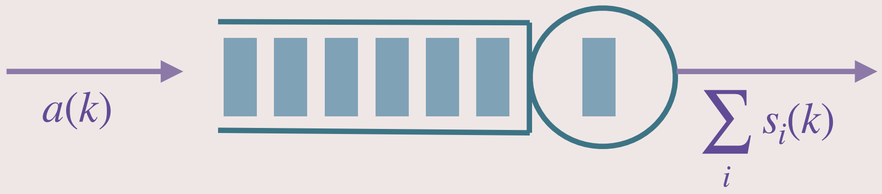
- SSC to a one-dimensional subspace
- System behaves as a single server queue
- All servers “pool” together and behave as a super server



Importance of CRP

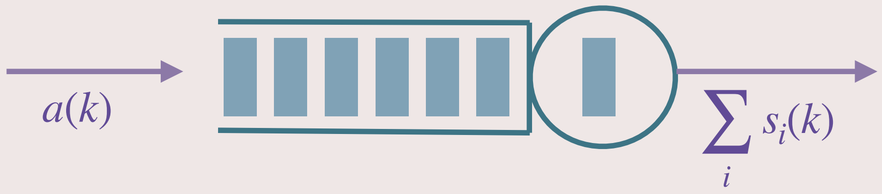
Importance of CRP

CRP

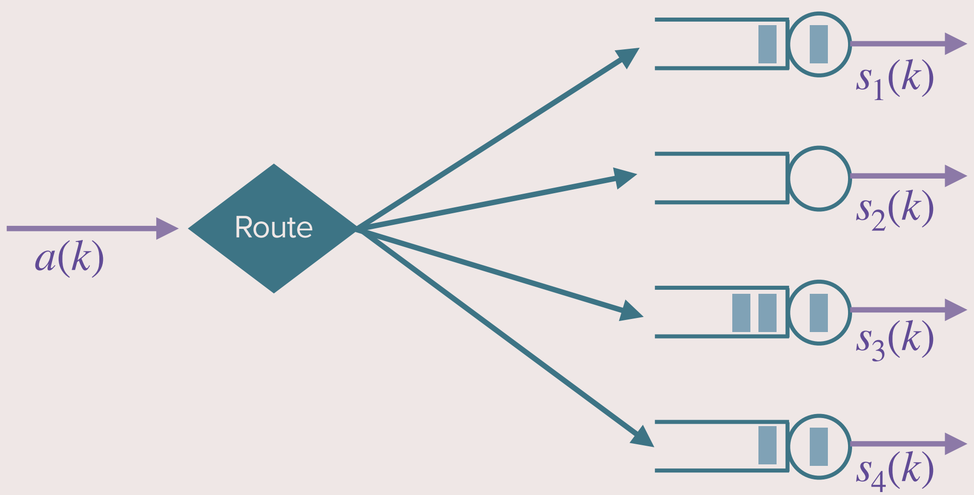


Importance of CRP

CRP

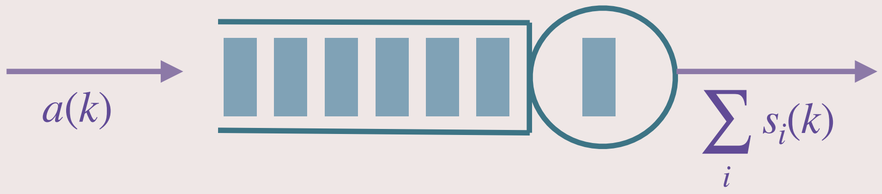


No CRP

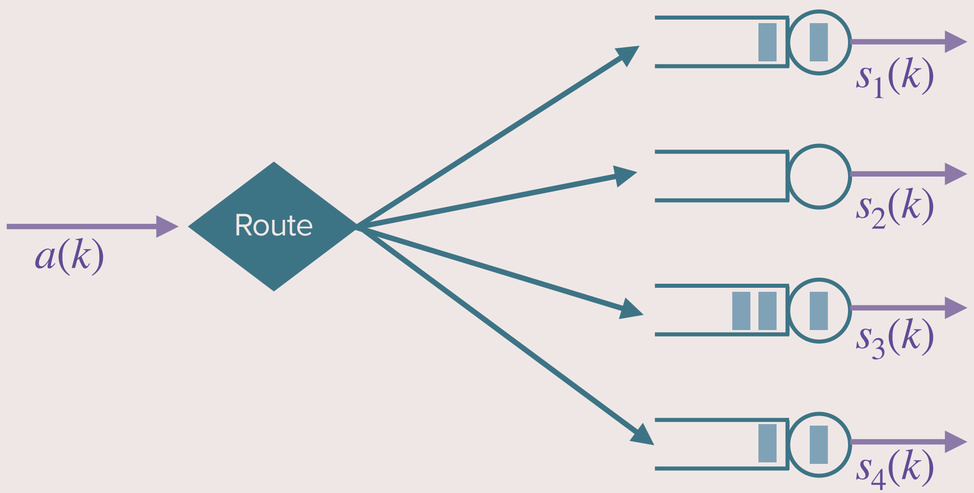


Importance of CRP

CRP



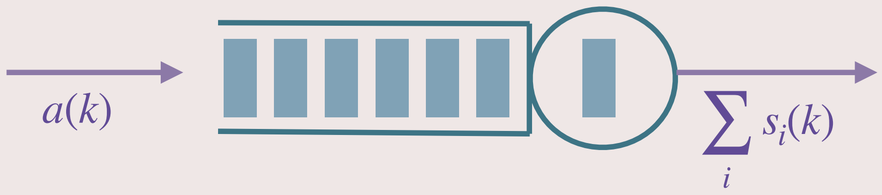
No CRP



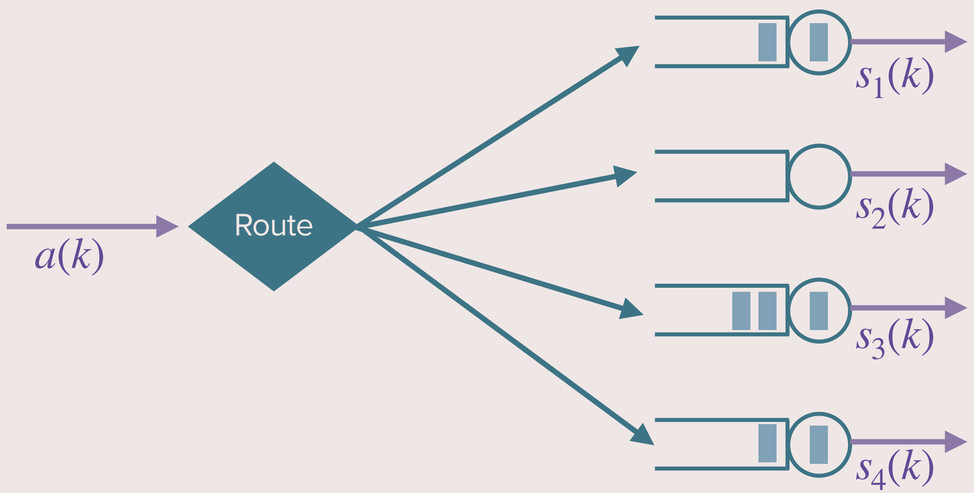
- To do: 7 jobs

Importance of CRP

CRP



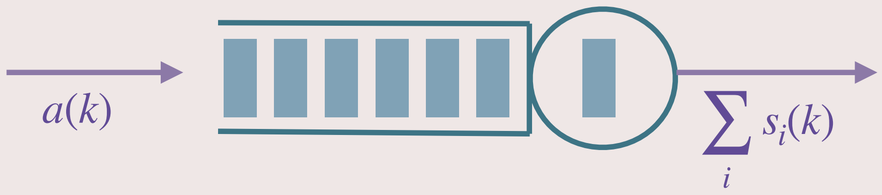
No CRP



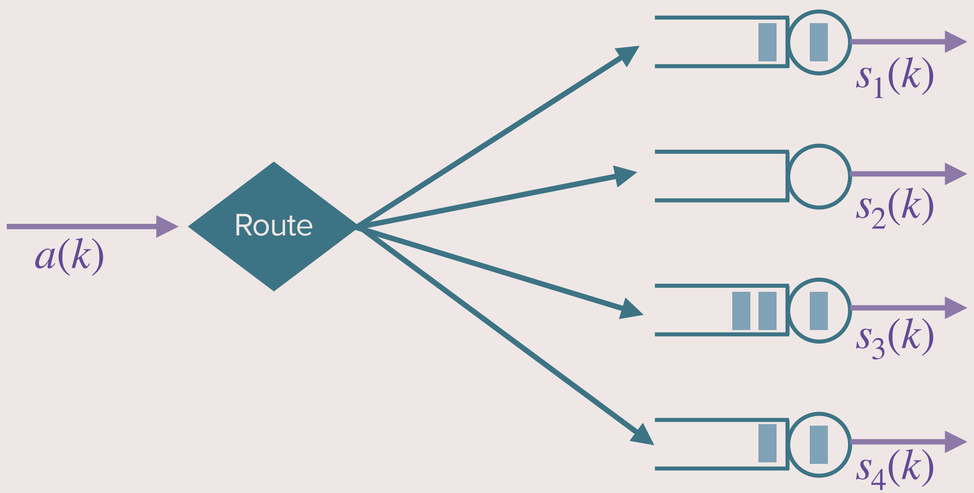
- To do: 7 jobs
- Same total service capacity

Importance of CRP

CRP



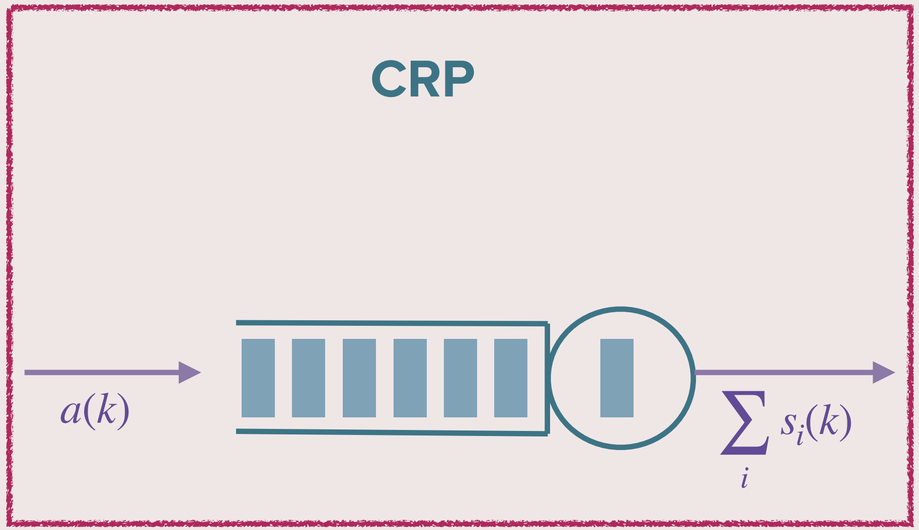
No CRP

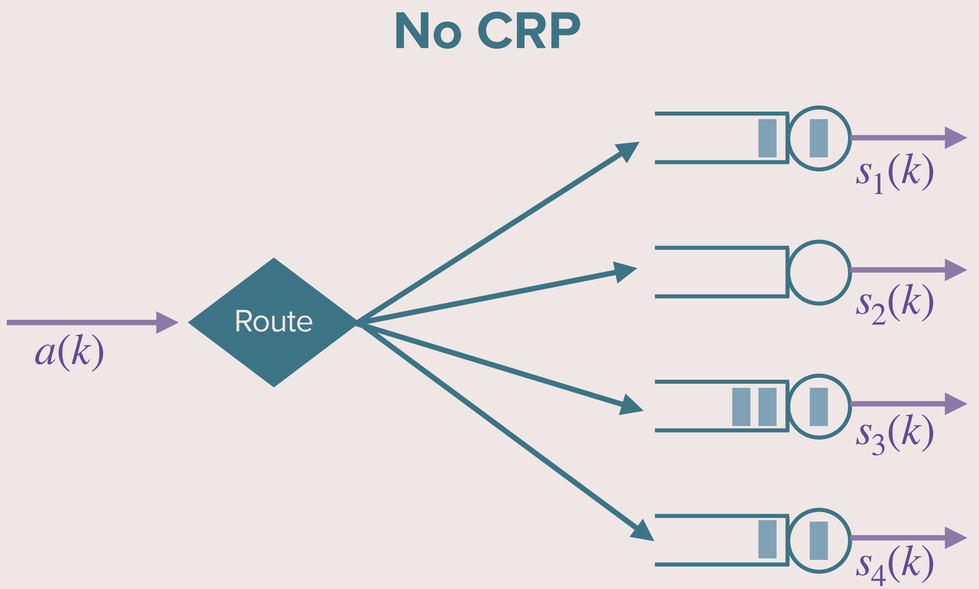


- To do: 7 jobs
- Same total service capacity

Which one finishes first?

Importance of CRP

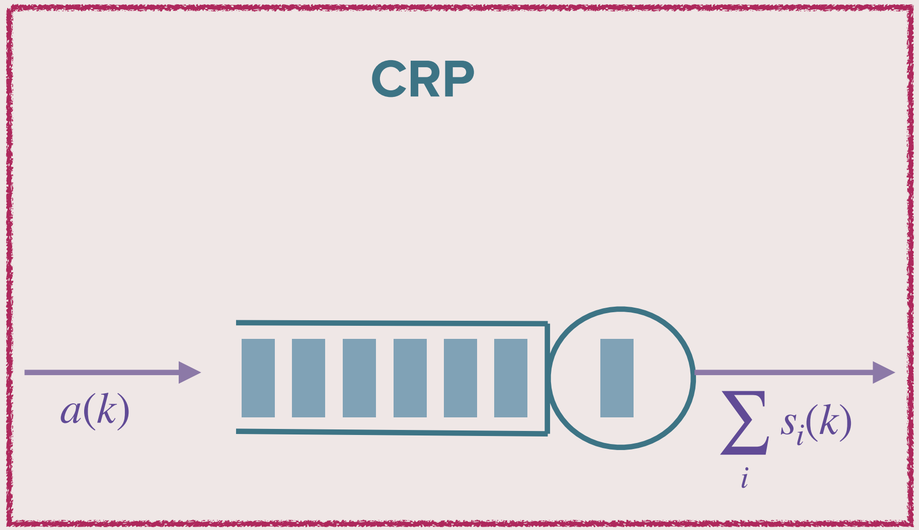


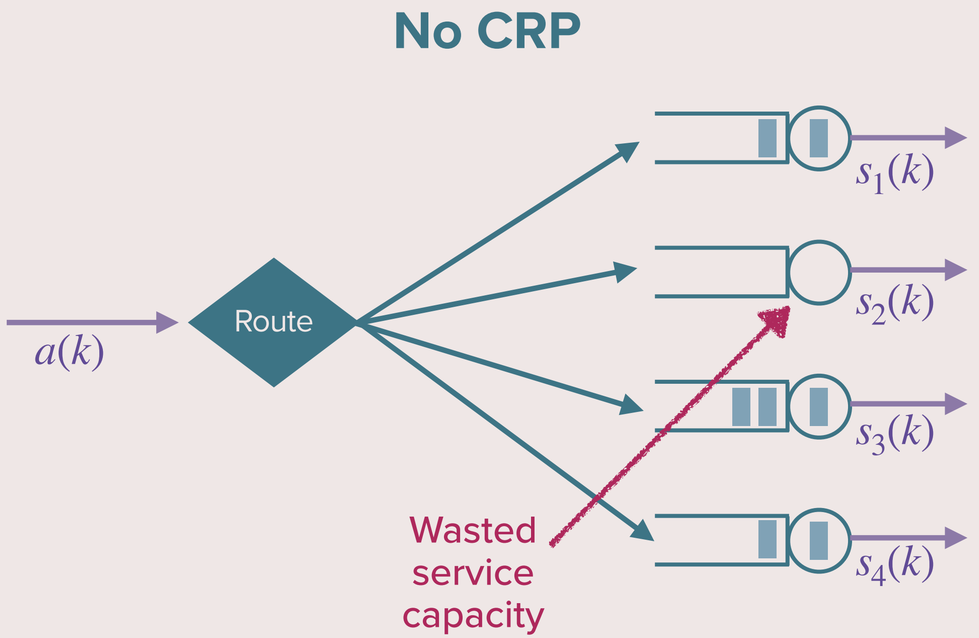


- To do: 7 jobs
- Same total service capacity

Which one finishes first?

Importance of CRP

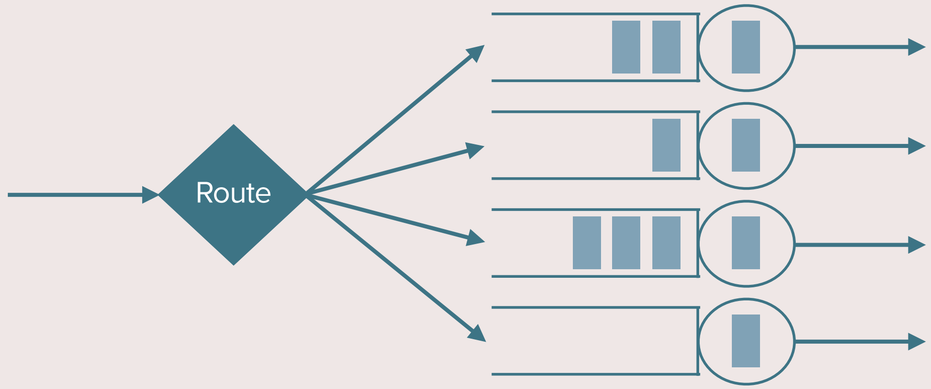




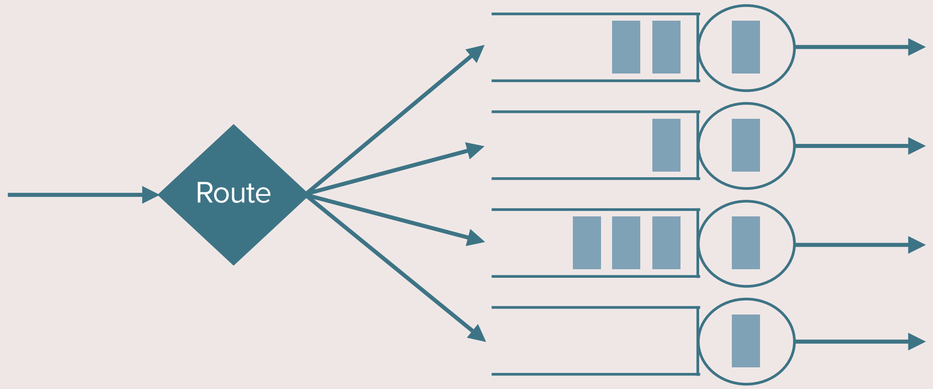
- To do: 7 jobs
- Same total service capacity

Which one finishes first?

CRP in Our Routing Algorithms



CRP in Our Routing Algorithms

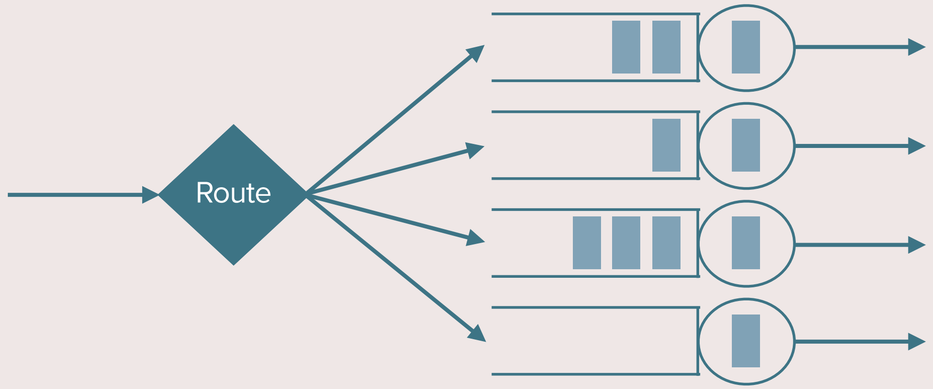


Random

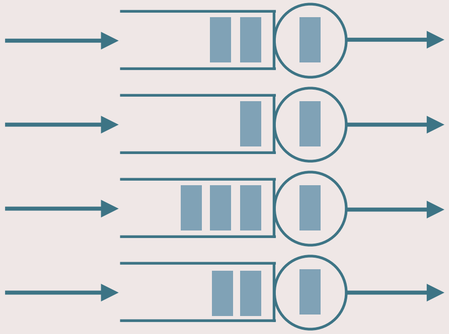
JSQ(d)

JSQ

CRP in Our Routing Algorithms



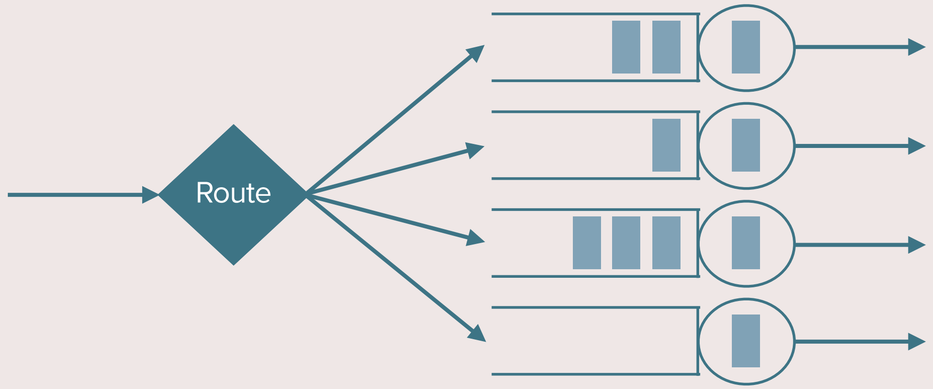
Random



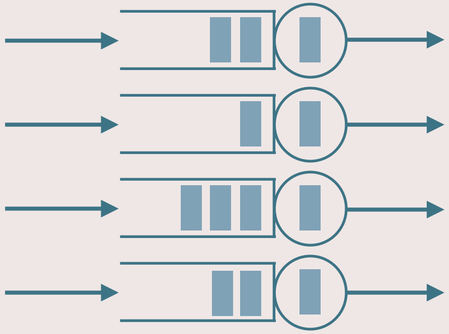
JSQ(d)

JSQ

CRP in Our Routing Algorithms



Random

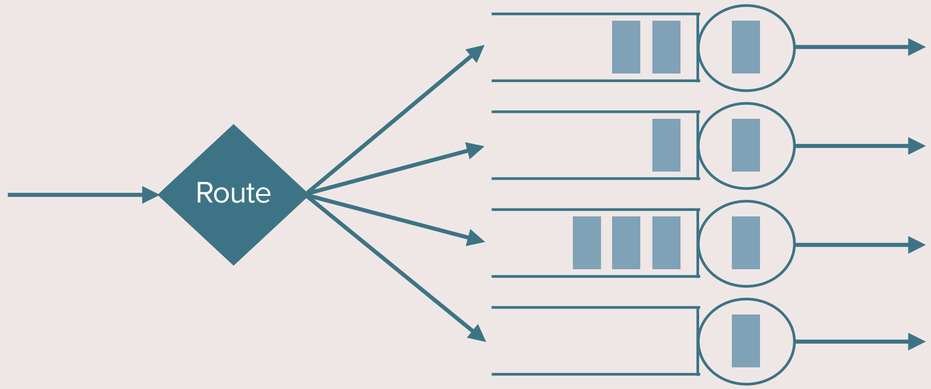


No CRP

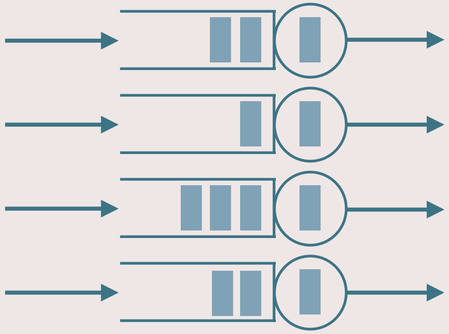
JSQ(d)

JSQ

CRP in Our Routing Algorithms



Random



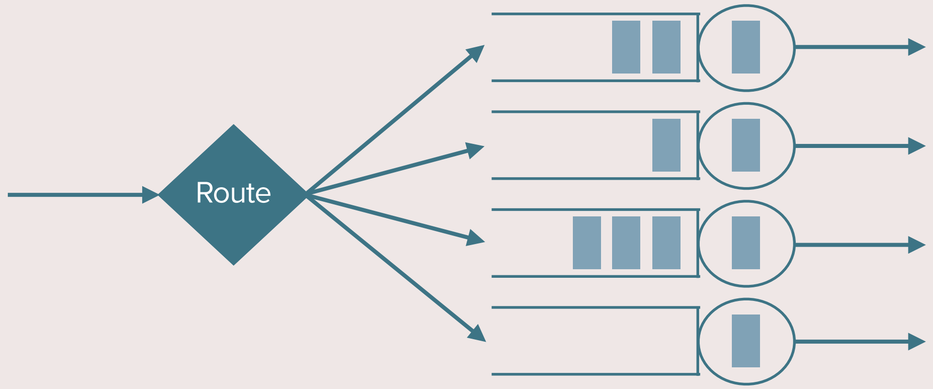
No CRP

JSQ(d)

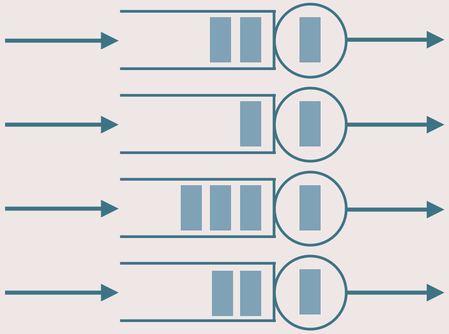
JSQ

Tries to make all queues equal

CRP in Our Routing Algorithms



Random



No CRP

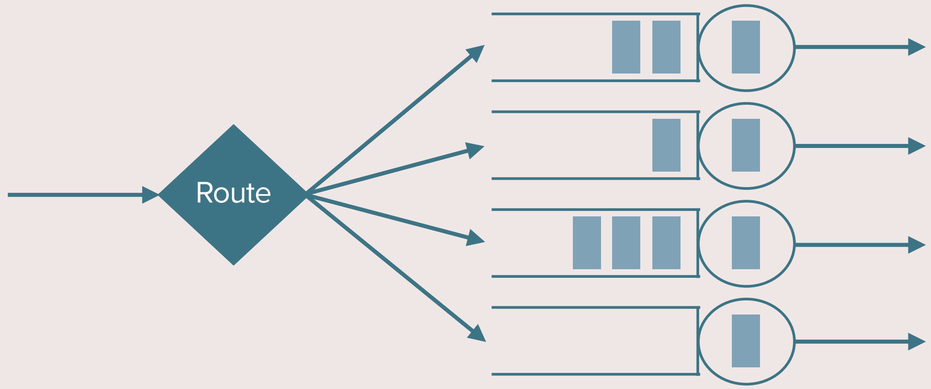
JSQ(d)

JSQ

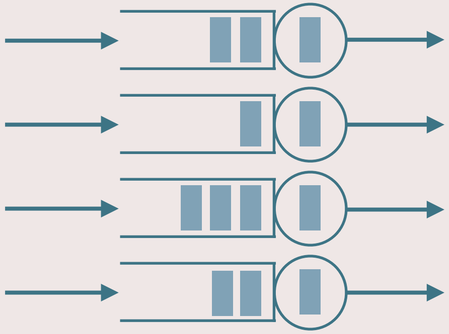
Tries to make all queues equal

CRP

CRP in Our Routing Algorithms



Random



No CRP

JSQ(d)

Select queues at random

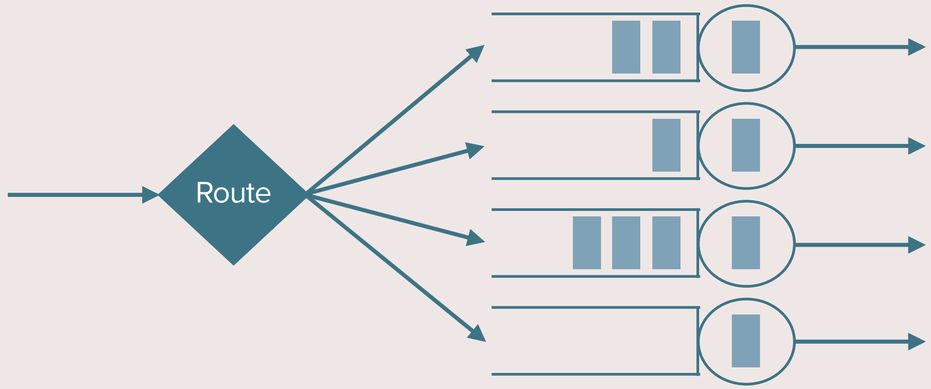
Route JSQ

JSQ

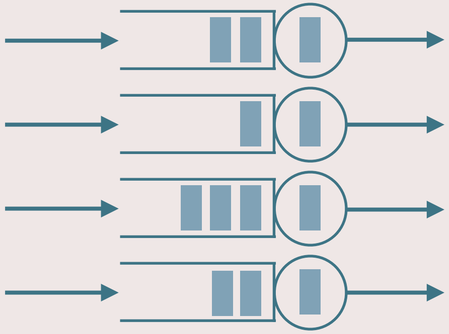
Tries to make all queues equal

CRP

CRP in Our Routing Algorithms



Random



No CRP

JSQ(d)

Select queues at random

Route JSQ

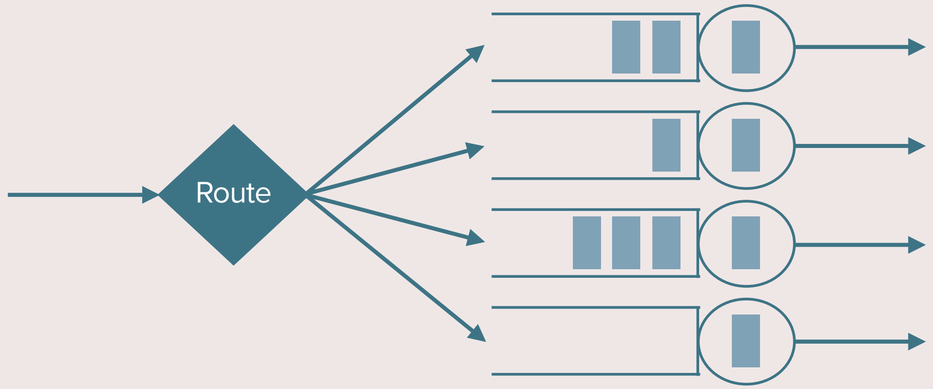
CRP

JSQ

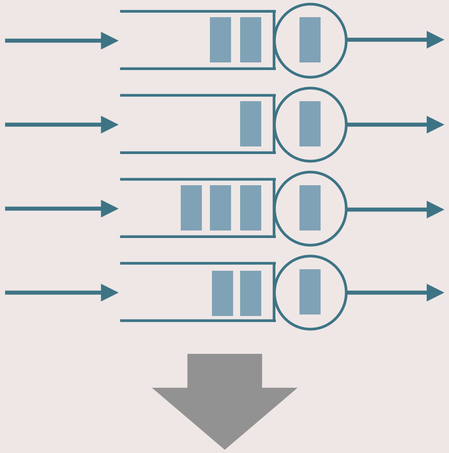
Tries to make all queues equal

CRP

CRP in Our Routing Algorithms



Random



No CRP

<

JSQ(d)

Select queues at random

Route JSQ

CRP

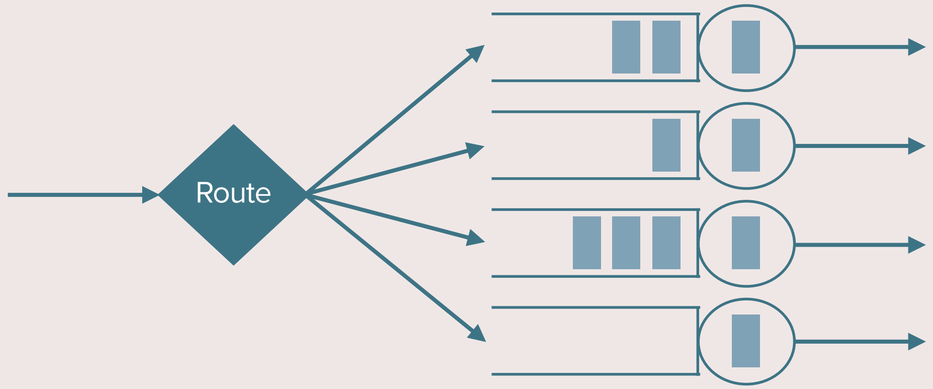
=

JSQ

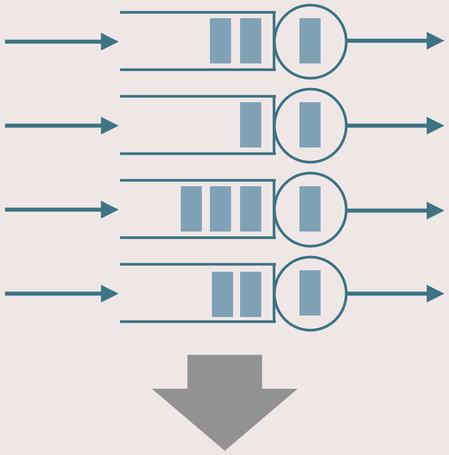
Tries to make all queues equal

CRP

CRP in Our Routing Algorithms



Random



No CRP

<

JSQ(d)

Select queues at random

Route JSQ

CRP

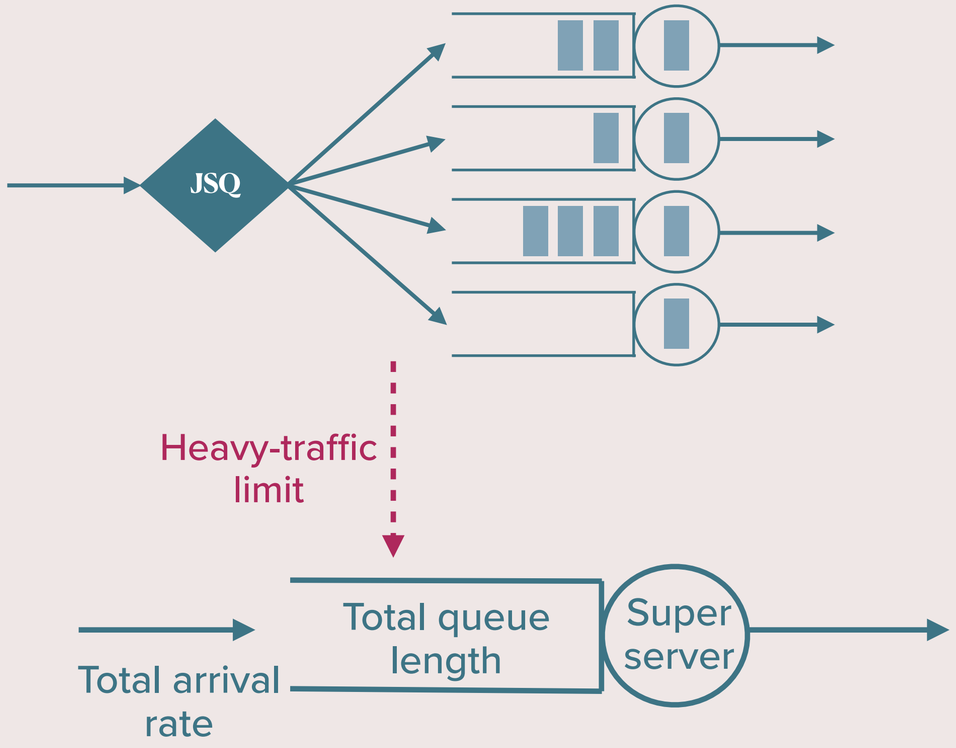
=

JSQ

Tries to make all queues equal

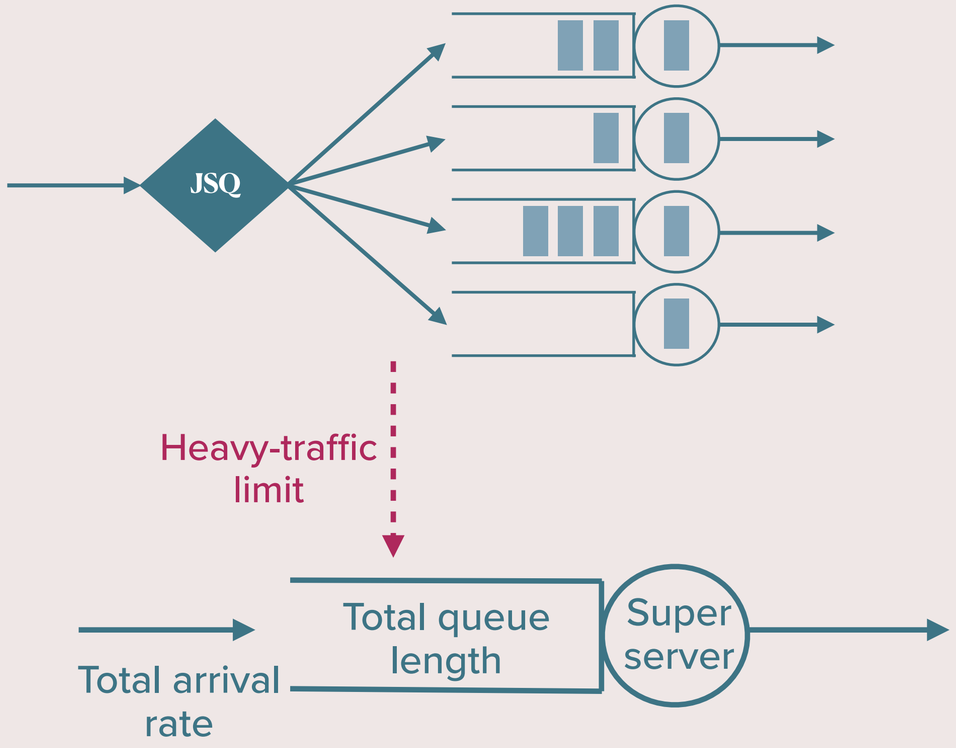
CRP

In the Literature



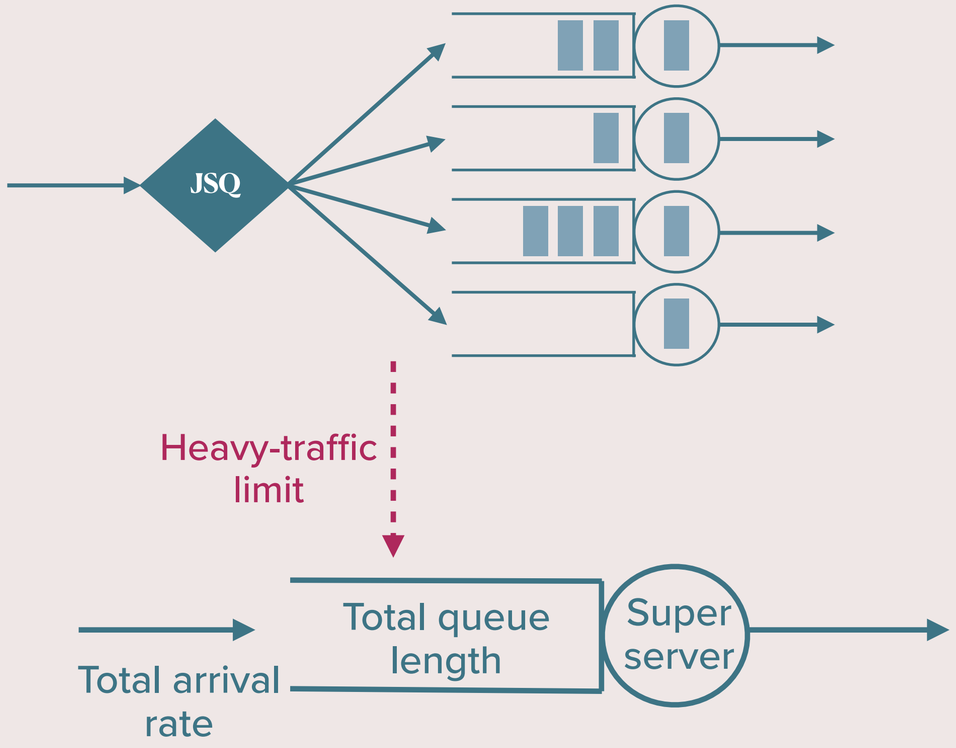
In the Literature

- The supermarket checkout system under JSQ satisfies the **CRP condition**
  - Diffusion limits: Foschini and Salz (1978)
  - Drift method: Eryilmaz and Srikant (2013)



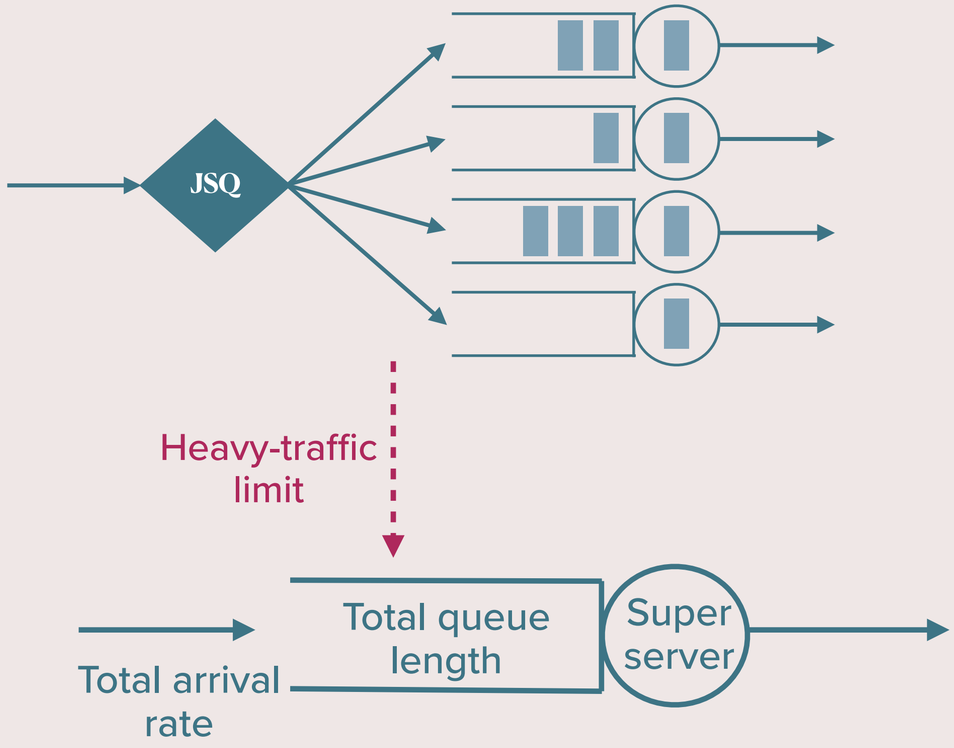
In the Literature

- The supermarket checkout system under JSQ satisfies the **CRP condition**
  - Diffusion limits: Foschini and Salz (1978)
  - Drift method: Eryilmaz and Srikant (2013)
- **Optimality** of JSQ:
  Winston (1977); Weber (1978); Ephremides, Varaiya and Walrand (1980)



In the Literature

- The supermarket checkout system under JSQ satisfies the **CRP condition**
  - Diffusion limits: Foschini and Salz (1978)
  - Drift method: Eryilmaz and Srikant (2013)
- **Optimality** of JSQ:
  Winston (1977); Weber (1978); Ephremides, Varaiya and Walrand (1980)
- Improving **performance** and decreasing **complexity** of routing algorithm:
  Chen and Ye (2012); Li, Kong and Wang (2018); Braverman (2020); Zhou, Tan and Schroff (2018); Ying (2016); Ying (2017); Eschenfeldt and Gamarnik (2018); Lu, Xie, Kliot, Geller, Larus and Greenberg (2011); Stolyar (2017); Ying, Srikant and Kang (2017)...

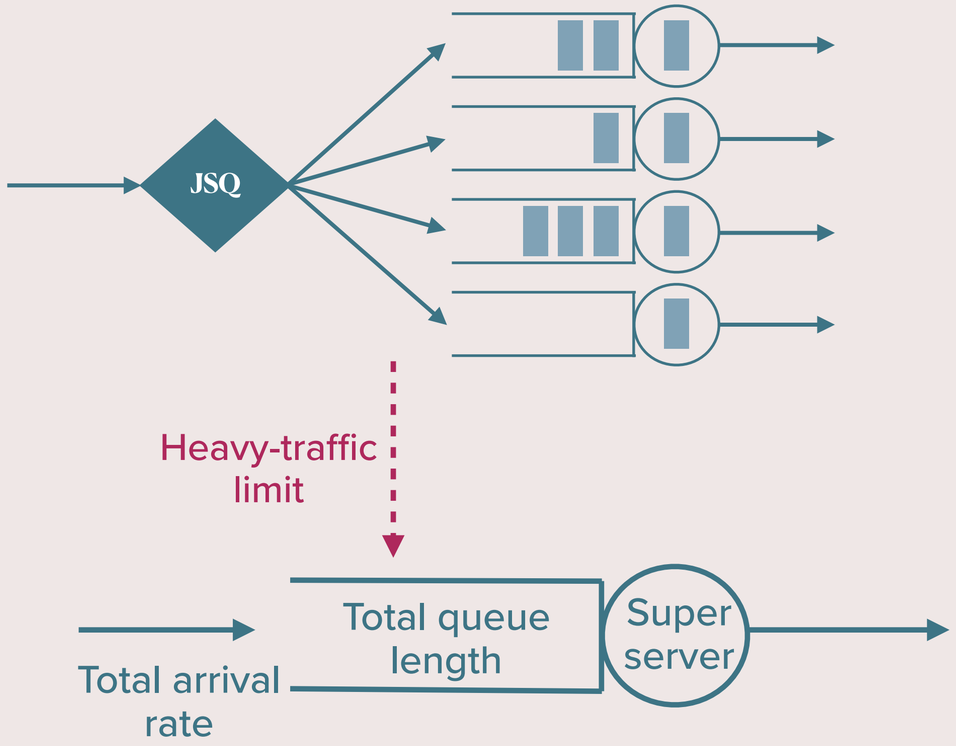


In the Literature

- The supermarket checkout system under JSQ satisfies the **CRP condition**
  - Diffusion limits: Foschini and Salz (1978)
  - Drift method: Eryilmaz and Srikant (2013)

- **Optimality** of JSQ:
  Winston (1977); Weber (1978); Ephremides, Varaiya and Walrand (1980)

- Improving **performance** and decreasing **complexity** of routing algorithm:
  Chen and Ye (2012); Li, Kong and Wang (2018); Braverman (2020); Zhou, Tan and Schroff (2018); Ying (2016); Ying (2017); Eschenfeldt and Gamarnik (2018); Lu, Xie, Kliot, Geller, Larus and Greenberg (2011); Stolyar (2017); Ying, Srikant and Kang (2017)...



Diffusion limits approach

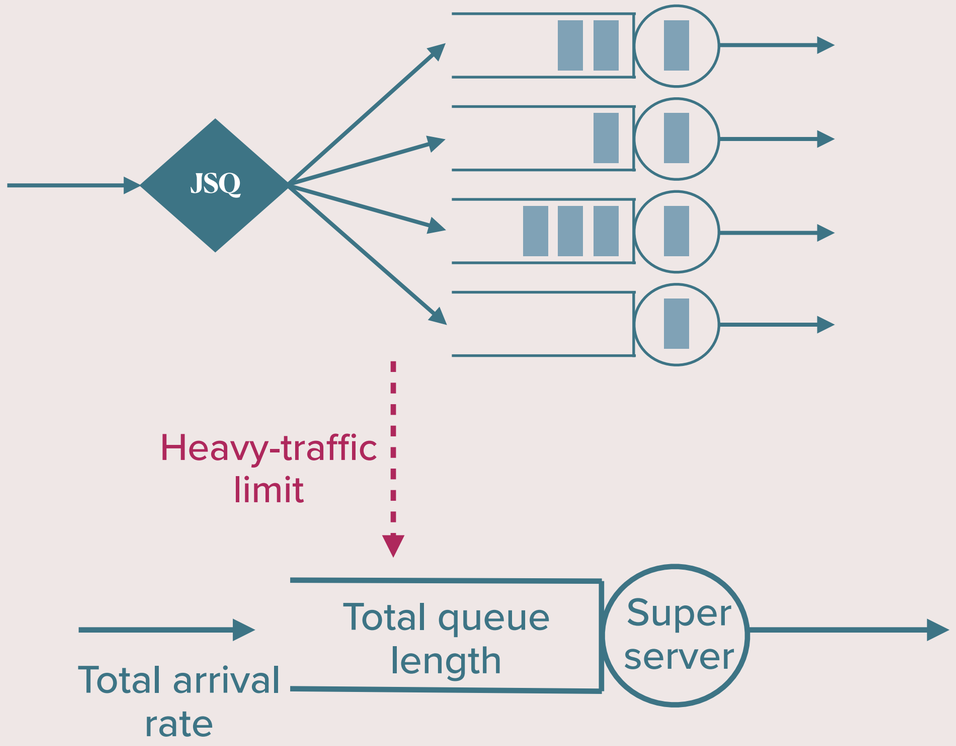
- Most popular
- Introduced by Kingman (1962)
- Show convergence in distribution of queue length scaled heavy-traffic parameter ϵ

In the Literature

- The supermarket checkout system under JSQ satisfies the **CRP condition**
  - Diffusion limits: Foschini and Salz (1978)
  - Drift method: Eryilmaz and Srikant (2013)

- **Optimality** of JSQ:
  Winston (1977); Weber (1978); Ephremides, Varaiya and Walrand (1980)

- Improving **performance** and decreasing **complexity** of routing algorithm:
  Chen and Ye (2012); Li, Kong and Wang (2018); Braverman (2020); Zhou, Tan and Schroff (2018); Ying (2016); Ying (2017); Eschenfeldt and Gamarnik (2018); Lu, Xie, Kliot, Geller, Larus and Greenberg (2011); Stolyar (2017); Ying, Srikant and Kang (2017)...



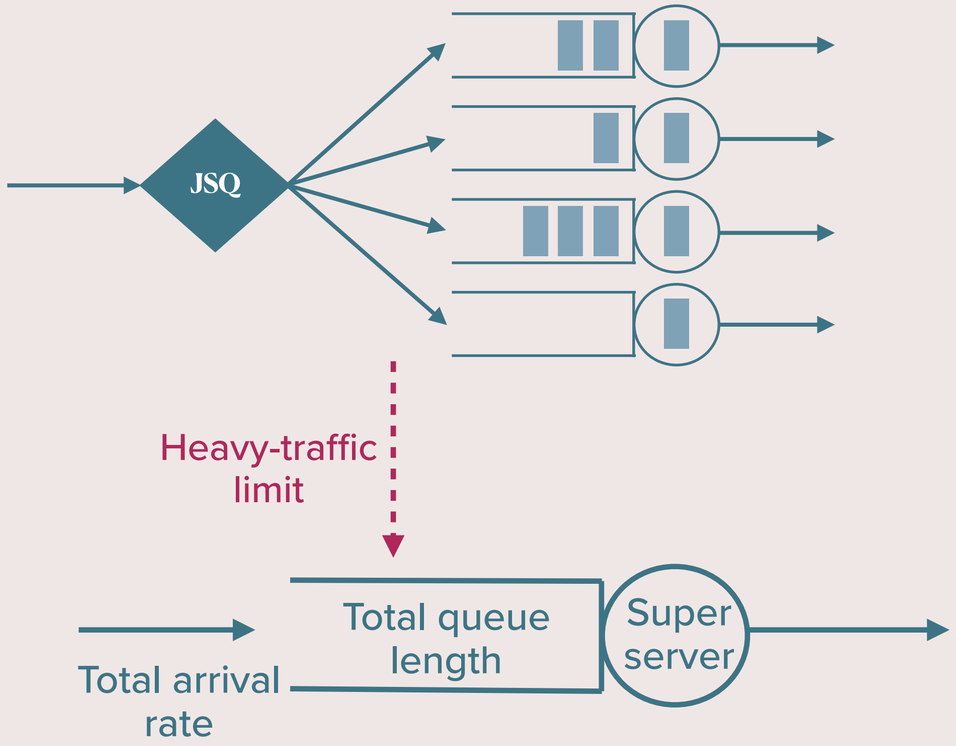
Diffusion limits approach

- Most popular
- Introduced by Kingman (1962)
- Show convergence in distribution of queue length scaled heavy-traffic parameter ϵ

$$\epsilon q^{(\epsilon)}(t)$$

In the Literature

- The supermarket checkout system under JSQ satisfies the **CRP condition**
  - Diffusion limits: Foschini and Salz (1978)
  - Drift method: Eryilmaz and Srikant (2013)
- **Optimality** of JSQ:
  Winston (1977); Weber (1978); Ephremides, Varaiya and Walrand (1980)
- Improving **performance** and decreasing **complexity** of routing algorithm:
  Chen and Ye (2012); Li, Kong and Wang (2018); Braverman (2020); Zhou, Tan and Schroff (2018); Ying (2016); Ying (2017); Eschenfeldt and Gamarnik (2018); Lu, Xie, Kliot, Geller, Larus and Greenberg (2011); Stolyar (2017); Ying, Srikant and Kang (2017)...



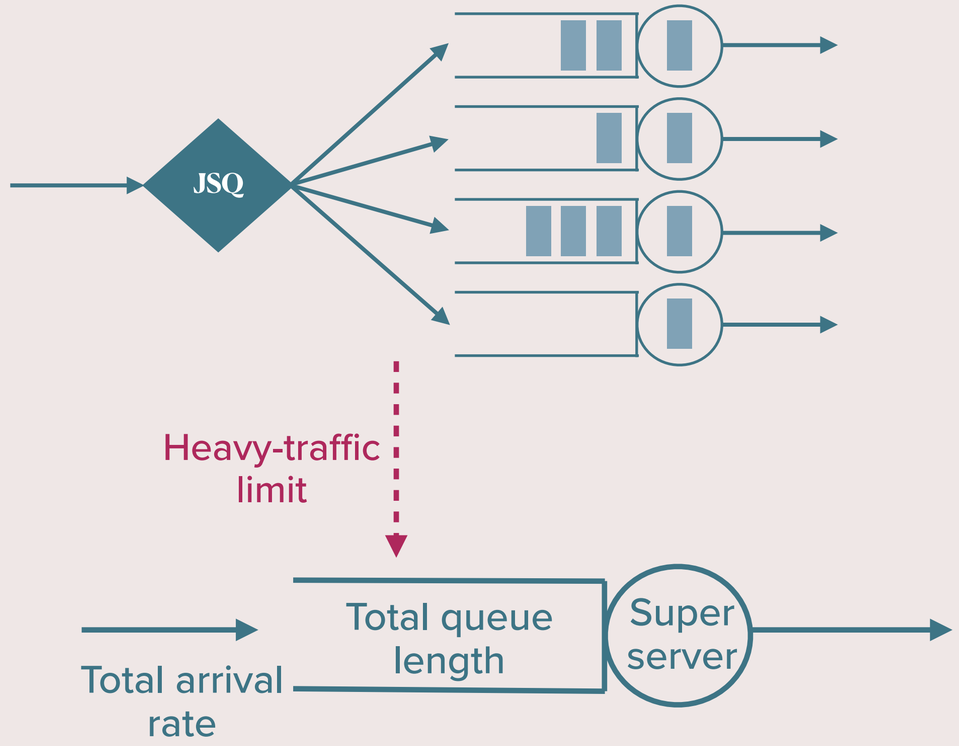
Diffusion limits approach

- Most popular
- Introduced by Kingman (1962)
- Show convergence in distribution of queue length scaled heavy-traffic parameter ϵ

$$\epsilon q^{(\epsilon)}(t) \xrightarrow{t \to \infty} \epsilon q^{(\epsilon)}(\infty)$$

In the Literature

- The supermarket checkout system under JSQ satisfies the **CRP condition**
  - Diffusion limits: Foschini and Salz (1978)
  - Drift method: Eryilmaz and Srikant (2013)
- **Optimality** of JSQ:
  Winston (1977); Weber (1978); Ephremides, Varaiya and Walrand (1980)
- Improving **performance** and decreasing **complexity** of routing algorithm:
  Chen and Ye (2012); Li, Kong and Wang (2018); Braverman (2020); Zhou, Tan and Schroff (2018); Ying (2016); Ying (2017); Eschenfeldt and Gamarnik (2018); Lu, Xie, Kliot, Geller, Larus and Greenberg (2011); Stolyar (2017); Ying, Srikant and Kang (2017)...



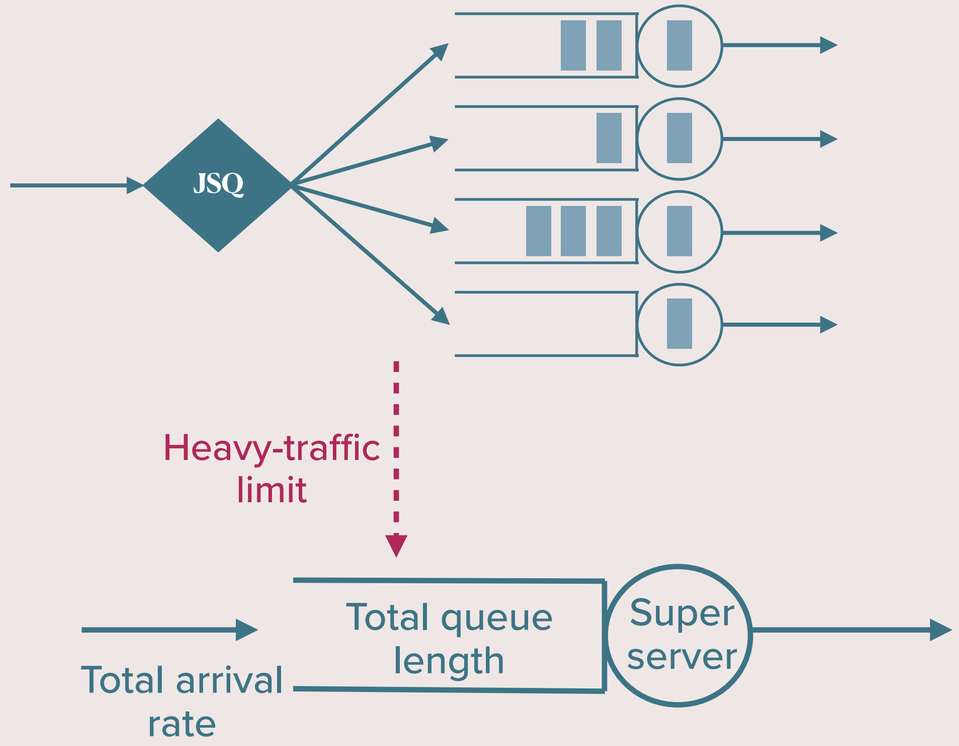
Diffusion limits approach

- Most popular
- Introduced by Kingman (1962)
- Show convergence in distribution of queue length scaled heavy-traffic parameter ϵ

$$\epsilon q^{(\epsilon)}(t) \xrightarrow{t \to \infty} \epsilon q^{(\epsilon)}(\infty)$$
$$\epsilon \downarrow 0 \Big\downarrow$$
$$\tilde{q}(t)$$
RBM

In the Literature

- The supermarket checkout system under JSQ satisfies the **CRP condition**
  - Diffusion limits: Foschini and Salz (1978)
  - Drift method: Eryilmaz and Srikant (2013)
- **Optimality** of JSQ:
  Winston (1977); Weber (1978); Ephremides, Varaiya and Walrand (1980)
- Improving **performance** and decreasing **complexity** of routing algorithm:
  Chen and Ye (2012); Li, Kong and Wang (2018); Braverman (2020); Zhou, Tan and Schroff (2018); Ying (2016); Ying (2017); Eschenfeldt and Gamarnik (2018); Lu, Xie, Kliot, Geller, Larus and Greenberg (2011); Stolyar (2017); Ying, Srikant and Kang (2017)...



Diffusion limits approach

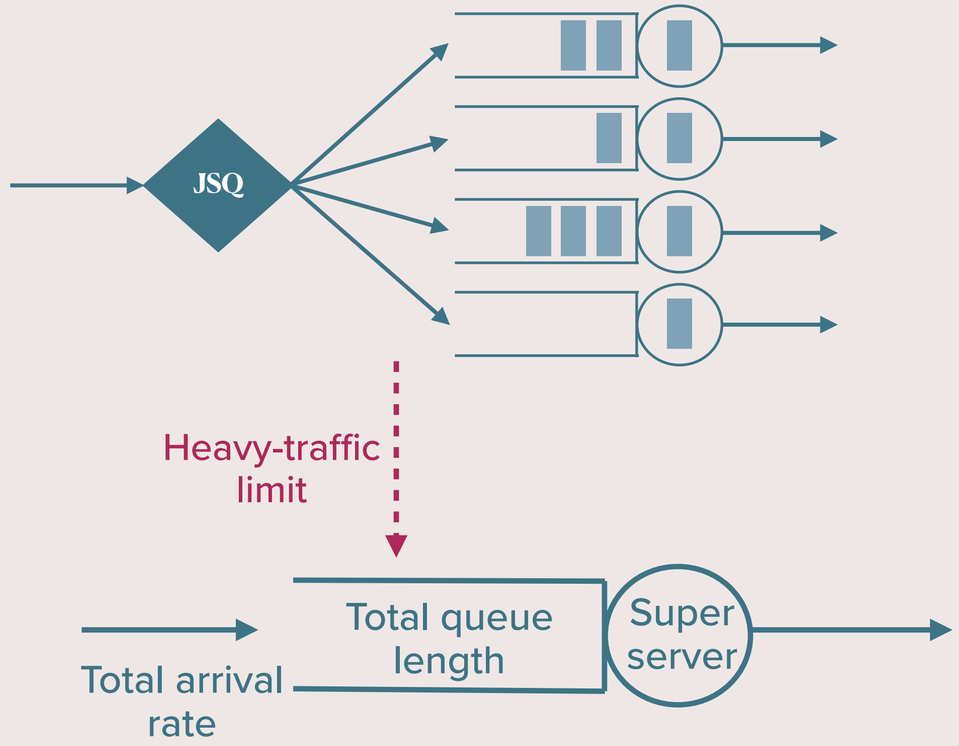
- Most popular
- Introduced by Kingman (1962)
- Show convergence in distribution of queue length scaled heavy-traffic parameter ϵ

$$\epsilon q^{(\epsilon)}(t) \xrightarrow{t \to \infty} \epsilon q^{(\epsilon)}(\infty)$$
$$\epsilon \downarrow 0 \Big\downarrow$$
$$\widetilde{q}(t) \xrightarrow{t \to \infty} q^*$$

RBM

In the Literature

- The supermarket checkout system under JSQ satisfies the **CRP condition**
  - Diffusion limits: Foschini and Salz (1978)
  - Drift method: Eryilmaz and Srikant (2013)
- **Optimality** of JSQ:
  Winston (1977); Weber (1978); Ephremides, Varaiya and Walrand (1980)
- Improving **performance** and decreasing **complexity** of routing algorithm:
  Chen and Ye (2012); Li, Kong and Wang (2018); Braverman (2020); Zhou, Tan and Schroff (2018); Ying (2016); Ying (2017); Eschenfeldt and Gamarnik (2018); Lu, Xie, Kliot, Geller, Larus and Greenberg (2011); Stolyar (2017); Ying, Srikant and Kang (2017)...



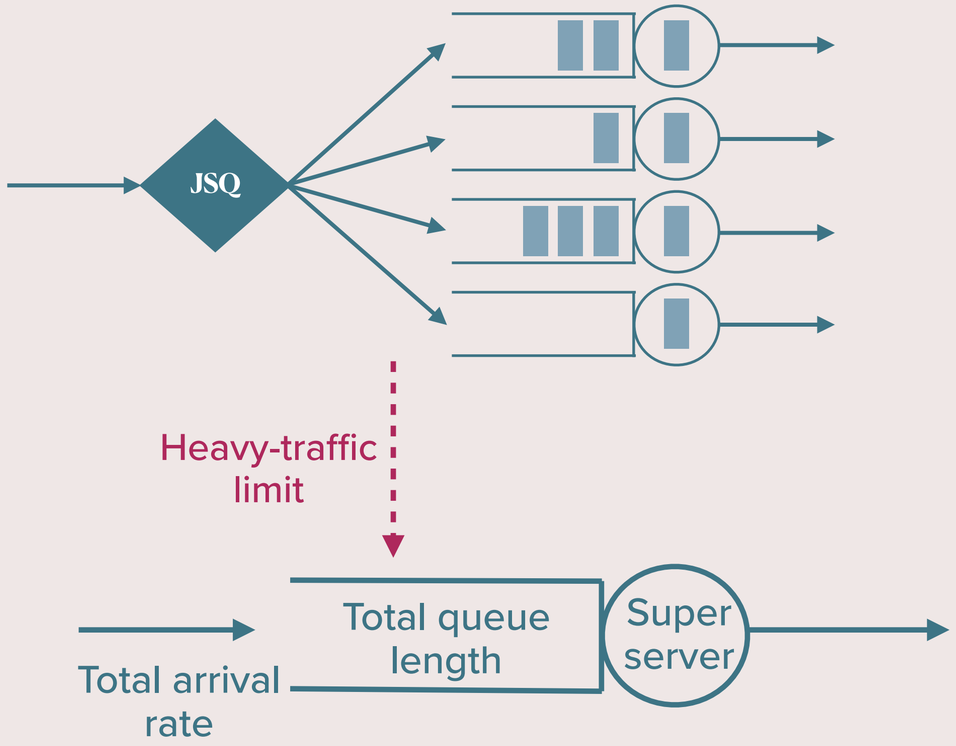
Diffusion limits approach

- Most popular
- Introduced by Kingman (1962)
- Show convergence in distribution of queue length scaled heavy-traffic parameter ϵ

$$\epsilon q^{(\epsilon)}(t) \xrightarrow{t \to \infty} \epsilon q^{(\epsilon)}(\infty)$$
$$\epsilon \downarrow 0 \qquad \epsilon \downarrow 0 \text{ Interchange of limits}$$
$$\widetilde{q}(t) \xrightarrow{t \to \infty} q^*$$
RBM

In the Literature

- The supermarket checkout system under JSQ satisfies the **CRP condition**
  - Diffusion limits: Foschini and Salz (1978)
  - Drift method: Eryilmaz and Srikant (2013)
- **Optimality** of JSQ:
  Winston (1977); Weber (1978); Ephremides, Varaiya and Walrand (1980)
- Improving **performance** and decreasing **complexity** of routing algorithm:
  Chen and Ye (2012); Li, Kong and Wang (2018); Braverman (2020); Zhou, Tan and Schroff (2018); Ying (2016); Ying (2017); Eschenfeldt and Gamarnik (2018); Lu, Xie, Kliot, Geller, Larus and Greenberg (2011); Stolyar (2017); Ying, Srikant and Kang (2017)...



Diffusion limits approach

- Most popular
- Introduced by Kingman (1962)
- Show convergence in distribution of queue length scaled heavy-traffic parameter ϵ

$$\begin{array}{ccc} \epsilon q^{(\epsilon)}(t) & \xrightarrow{t \to \infty} & \epsilon q^{(\epsilon)}(\infty) \\ \epsilon \downarrow 0 \big\downarrow & & \epsilon \downarrow 0 \big\downarrow \text{ Interchange of limits} \\ \widetilde{q}(t) & \xrightarrow{t \to \infty} & q^* \end{array}$$

RBM

We propose a proof in 4 lines

The Single-Server Queue

The Single-Server Queue

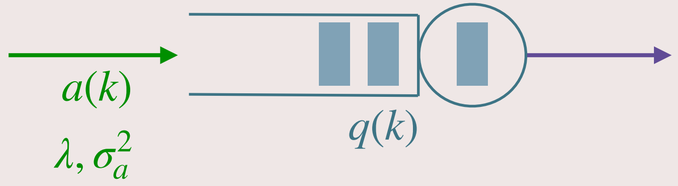
- Discrete-time model: k indexes time

The Single-Server Queue

- Discrete-time model: k indexes time
- $q(k)$: # jobs in the system at the beginning of time slot k

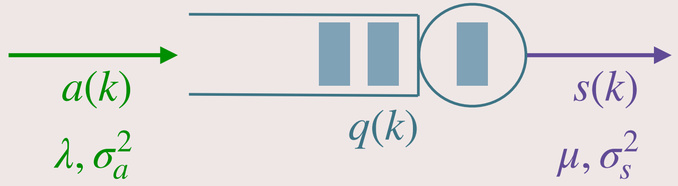
The Single-Server Queue

- Discrete-time model: k indexes time
- $q(k)$: # jobs in the system at the beginning of time slot k
- $a(k)$: # arrivals in time slot k
  - $\mathbb{E}[a(k)] = \lambda, \mathsf{Var}[a(k)] = \sigma_a^2$



The Single-Server Queue

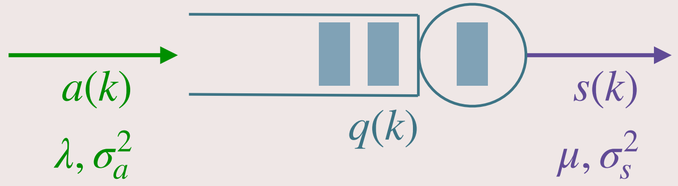
- Discrete-time model: k indexes time
- $q(k)$: # jobs in the system at the beginning of time slot k
- $a(k)$: # arrivals in time slot k
  - $\mathbb{E}[a(k)] = \lambda, \mathsf{Var}[a(k)] = \sigma_a^2$
- $s(k)$: potential service in time slot k
  - $\mathbb{E}[s(k)] = \mu, \mathsf{Var}[s(k)] = \sigma_s^2$



The Single-Server Queue

- Discrete-time model: k indexes time
- $q(k)$: # jobs in the system at the beginning of time slot k
- $a(k)$: # arrivals in time slot k
  - $\mathbb{E}[a(k)] = \lambda, \mathsf{Var}[a(k)] = \sigma_a^2$
- $s(k)$: potential service in time slot k
  - $\mathbb{E}[s(k)] = \mu, \mathsf{Var}[s(k)] = \sigma_s^2$
- Dynamics of the queues:

$$\begin{aligned} q(k+1) &= \max\left\{q(k) + a(k) - s(k), 0\right\} \\ &= q(k) + a(k) - s(k) + u(k) \end{aligned}$$

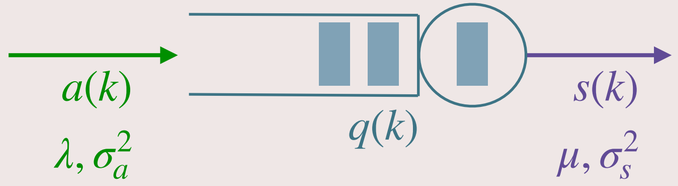


The Single-Server Queue

- Discrete-time model: k indexes time
- $q(k)$: # jobs in the system at the beginning of time slot k
- $a(k)$: # arrivals in time slot k
  - $\mathbb{E}[a(k)] = \lambda, \mathsf{Var}[a(k)] = \sigma_a^2$
- $s(k)$: potential service in time slot k
  - $\mathbb{E}[s(k)] = \mu, \mathsf{Var}[s(k)] = \sigma_s^2$
- Dynamics of the queues:

$$q(k+1) = \max\{q(k) + a(k) - s(k), 0\}$$
$$= q(k) + a(k) - s(k) + u(k)$$

Unused service



The Single-Server Queue

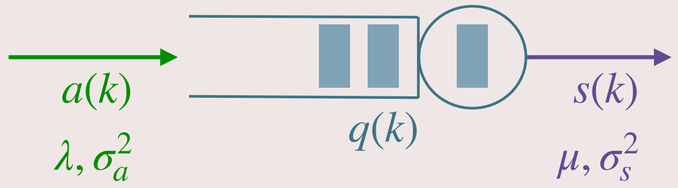
- Discrete-time model: k indexes time
- $q(k)$: # jobs in the system at the beginning of time slot k
- $a(k)$: # arrivals in time slot k
  - $\mathbb{E}[a(k)] = \lambda, \mathsf{Var}[a(k)] = \sigma_a^2$
- $s(k)$: potential service in time slot k
  - $\mathbb{E}[s(k)] = \mu, \mathsf{Var}[s(k)] = \sigma_s^2$
- Dynamics of the queues:

$$q(k+1) = \max\{q(k) + a(k) - s(k), 0\}$$
$$= q(k) + a(k) - s(k) + u(k)$$

Unused service

$$q(k+1)u(k) = 0$$

Key property



The Single-Server Queue

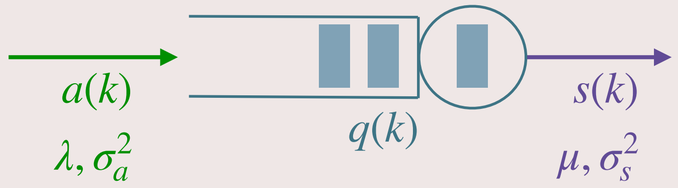
- Discrete-time model: k indexes time
- $q(k)$: # jobs in the system at the beginning of time slot k
- $a(k)$: # arrivals in time slot k
  - $\mathbb{E}[a(k)] = \lambda, \mathsf{Var}[a(k)] = \sigma_a^2$
- $s(k)$: potential service in time slot k
  - $\mathbb{E}[s(k)] = \mu, \mathsf{Var}[s(k)] = \sigma_s^2$
- Dynamics of the queues:

$$q(k+1) = \max\{q(k) + a(k) - s(k), 0\}$$
$$= q(k) + a(k) - s(k) + u(k)$$

Unused service

$$q(k+1)u(k) = 0$$

Key property



Heavy-traffic: $\epsilon \triangleq \mu - \lambda$

$\implies$ $\epsilon \downarrow 0$ is the heavy-traffic limit

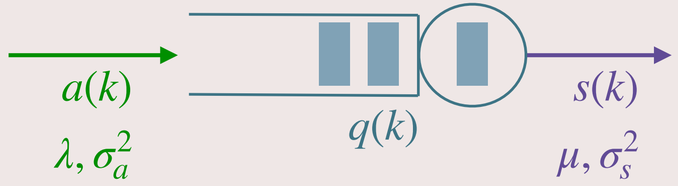
The Single-Server Queue

- Discrete-time model: k indexes time
- $q(k)$: # jobs in the system at the beginning of time slot k
- $a(k)$: # arrivals in time slot k
  - $\mathbb{E}[a(k)] = \lambda, \mathsf{Var}[a(k)] = \sigma_a^2$
- $s(k)$: potential service in time slot k
  - $\mathbb{E}[s(k)] = \mu, \mathsf{Var}[s(k)] = \sigma_s^2$
- Dynamics of the queues:

$$\begin{aligned} q(k+1) &= \max\{q(k) + a(k) - s(k), 0\} \\ &= q(k) + a(k) - s(k) + u(k) \end{aligned}$$

$u(k)$: Unused service

$q(k+1)u(k) = 0$ Key property



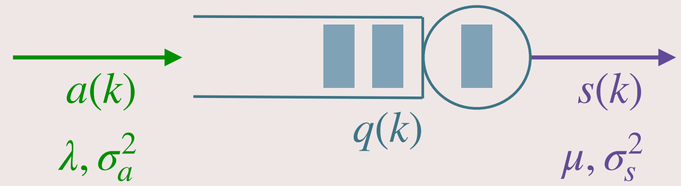
Heavy-traffic: $\epsilon \triangleq \mu - \lambda$

$\Longrightarrow$ $\epsilon \downarrow 0$ is the heavy-traffic limit

We wish to show:

$$\epsilon q \Rightarrow Expo\left(\frac{2}{\sigma_a^2 + \sigma_s^2}\right)$$

The MGF Method [HL, Maguluri '20]



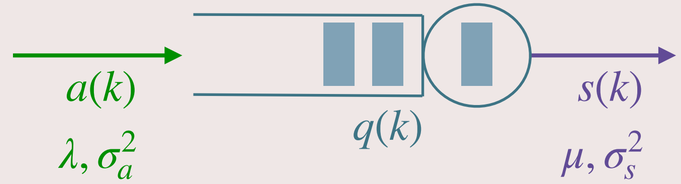
Dynamics of the queues:

$$q(k+1) = q(k) + a(k) - s(k) + u(k)$$

$$\implies q(k+1)u(k) = 0$$

The MGF Method [HL, Maguluri '20]

- Key Lemma:



Dynamics of the queues:

$$q(k+1) = q(k) + a(k) - s(k) + u(k)$$

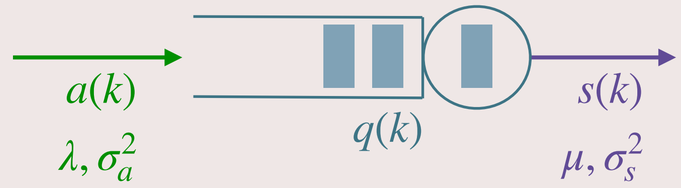
$$\implies q(k+1)u(k) = 0$$

The MGF Method [HL, Maguluri '20]

- Key Lemma:

Step 1:

$$\left(e^{\theta\epsilon q(k+1)} - 1\right)\left(e^{-\theta\epsilon u(k)} - 1\right) = 0$$



Dynamics of the queues:

$$q(k+1) = q(k) + a(k) - s(k) + u(k)$$

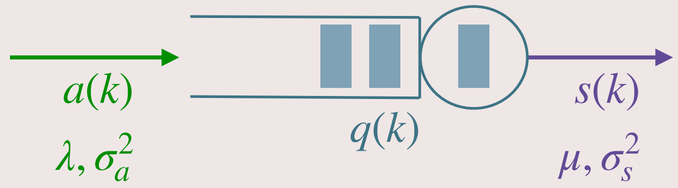
$$\implies q(k+1)u(k) = 0$$

The MGF Method [HL, Maguluri '20]

- Key Lemma:

Step 1:

$$\left(e^{\theta\epsilon q(k+1)} - 1\right)\left(e^{-\theta\epsilon u(k)} - 1\right) = 0$$



Dynamics of the queues:

$$q(k+1) = q(k) + a(k) - s(k) + u(k)$$

$$\Longrightarrow q(k+1)u(k) = 0$$

The MGF Method [HL, Maguluri '20]

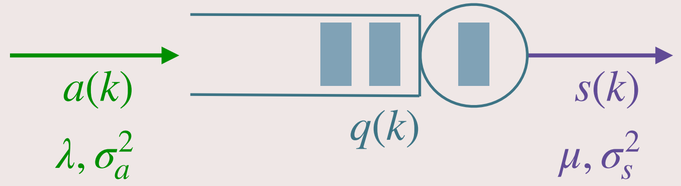
- Key Lemma:

Step 1:

$$\left(e^{\theta\epsilon q(k+1)} - 1\right)\left(e^{-\theta\epsilon u(k)} - 1\right) = 0$$

- From the lemma:

$$\mathbb{E}\left[e^{\theta\epsilon q(k+1)}\right] = \mathbb{E}\left[e^{\theta\epsilon(q(k)+a(k)-s(k))} - e^{-\theta\epsilon u(k)} + 1\right]$$



Dynamics of the queues:

$$q(k+1) = q(k) + a(k) - s(k) + u(k)$$

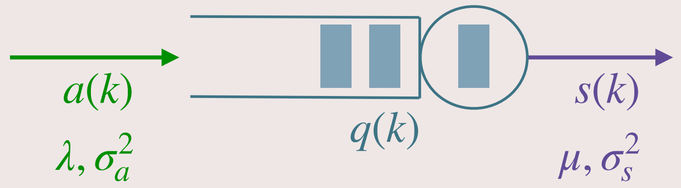
$$\Longrightarrow q(k+1)u(k) = 0$$

The MGF Method [HL, Maguluri '20]

- Key Lemma:

Step 1:

$$\left(e^{\theta\epsilon q(k+1)} - 1\right)\left(e^{-\theta\epsilon u(k)} - 1\right) = 0$$



Dynamics of the queues:

$$q(k+1) = q(k) + a(k) - s(k) + u(k)$$

$$\Longrightarrow q(k+1)u(k) = 0$$

- From the lemma:

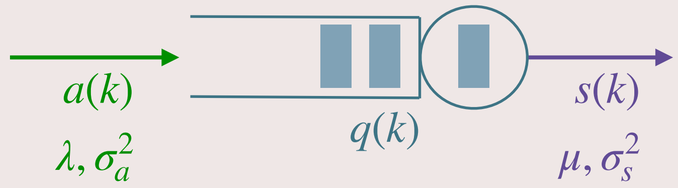
$$\mathbb{E}\left[e^{\theta\epsilon q(k+1)}\right] = \mathbb{E}\left[e^{\theta\epsilon(q(k)+a(k)-s(k))} - e^{-\theta\epsilon u(k)} + 1\right]$$

The MGF Method [HL, Maguluri '20]

- Key Lemma:

Step 1:

$$\left(e^{\theta\epsilon q(k+1)} - 1\right)\left(e^{-\theta\epsilon u(k)} - 1\right) = 0$$



Dynamics of the queues:

$$q(k+1) = q(k) + a(k) - s(k) + u(k)$$

$$\implies q(k+1)u(k) = 0$$

- From the lemma:

$$\mathbb{E}\left[e^{\theta\epsilon q(k+1)}\right] = \mathbb{E}\left[e^{\theta\epsilon(q(k)+a(k)-s(k))} - e^{-\theta\epsilon u(k)} + 1\right]$$

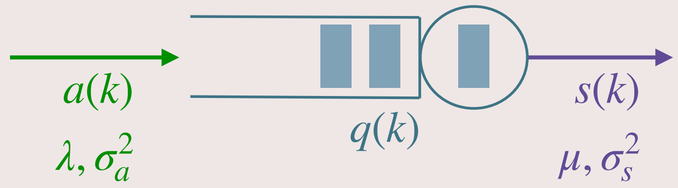
$$\implies \mathbb{E}\left[e^{\theta\epsilon q}\right] = \frac{1 - \mathbb{E}\left[e^{-\theta\epsilon u}\right]}{1 - \mathbb{E}\left[e^{\theta\epsilon(a-s)}\right]}$$

The MGF Method [HL, Maguluri '20]

- Key Lemma:

Step 1:

$$\left(e^{\theta\epsilon q(k+1)} - 1\right)\left(e^{-\theta\epsilon u(k)} - 1\right) = 0$$



Dynamics of the queues:

$$q(k+1) = q(k) + a(k) - s(k) + u(k)$$

$$\Longrightarrow q(k+1)u(k) = 0$$

- From the lemma:

$$\mathbb{E}\left[e^{\theta\epsilon q(k+1)}\right] = \mathbb{E}\left[e^{\theta\epsilon(q(k)+a(k)-s(k))} - e^{-\theta\epsilon u(k)} + 1\right]$$

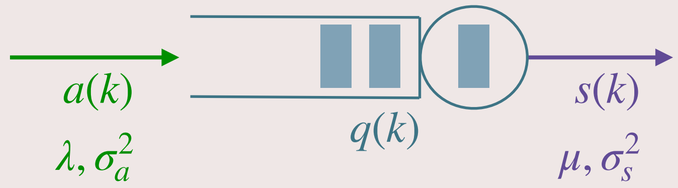
$$\Longrightarrow \mathbb{E}\left[e^{\theta\epsilon q}\right] = \frac{1 - \mathbb{E}\left[e^{-\theta\epsilon u}\right]}{1 - \mathbb{E}\left[e^{\theta\epsilon(a-s)}\right]}$$

Step 2: Bound unused service and take the heavy-traffic limit

The MGF Method [HL, Maguluri '20]

- Key Lemma:

Step 1:
$$\left(e^{\theta\epsilon q(k+1)} - 1\right)\left(e^{-\theta\epsilon u(k)} - 1\right) = 0$$



Dynamics of the queues:

$$q(k+1) = q(k) + a(k) - s(k) + u(k)$$
$$\Longrightarrow q(k+1)u(k) = 0$$

- From the lemma:

$$\mathbb{E}\left[e^{\theta\epsilon q(k+1)}\right] = \mathbb{E}\left[e^{\theta\epsilon(q(k)+a(k)-s(k))} - e^{-\theta\epsilon u(k)} + 1\right]$$

$$\Longrightarrow \mathbb{E}\left[e^{\theta\epsilon q}\right] = \frac{1 - \mathbb{E}\left[e^{-\theta\epsilon u}\right]}{1 - \mathbb{E}\left[e^{\theta\epsilon(a-s)}\right]}$$

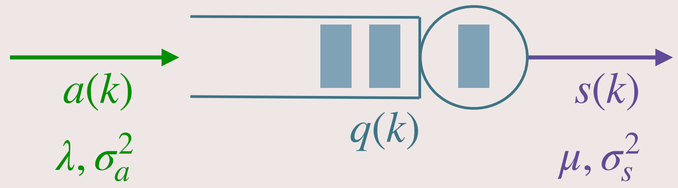
Step 2: Bound unused service and take the heavy-traffic limit

$$\Longrightarrow \lim_{\epsilon \downarrow 0} \mathbb{E}\left[e^{\theta\epsilon q}\right] = \frac{1}{1 - \theta\left(\frac{\sigma_a^2 + \sigma_s^2}{2}\right)}$$

The MGF Method [HL, Maguluri '20]

- Key Lemma:

Step 1:
$$\left(e^{\theta\epsilon q(k+1)} - 1\right)\left(e^{-\theta\epsilon u(k)} - 1\right) = 0$$



Dynamics of the queues:

$$q(k+1) = q(k) + a(k) - s(k) + u(k)$$
$$\Longrightarrow q(k+1)u(k) = 0$$

- From the lemma:

$$\mathbb{E}\left[e^{\theta\epsilon q(k+1)}\right] = \mathbb{E}\left[e^{\theta\epsilon(q(k)+a(k)-s(k))} - e^{-\theta\epsilon u(k)} + 1\right]$$

$$\Longrightarrow \mathbb{E}\left[e^{\theta\epsilon q}\right] = \frac{1 - \mathbb{E}\left[e^{-\theta\epsilon u}\right]}{1 - \mathbb{E}\left[e^{\theta\epsilon(a-s)}\right]}$$

Step 2: Bound unused service and take the heavy-traffic limit

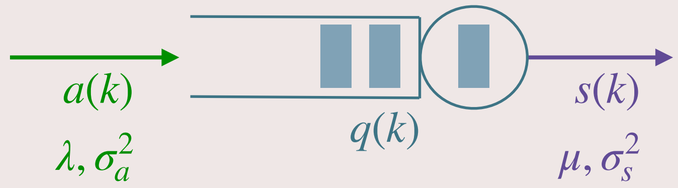
$$\Longrightarrow \lim_{\epsilon\downarrow 0} \mathbb{E}\left[e^{\theta\epsilon q}\right] = \frac{1}{1 - \theta\left(\frac{\sigma_a^2 + \sigma_s^2}{2}\right)}$$

MGF of expo. random variable with mean $\dfrac{\sigma_a^2 + \sigma_s^2}{2}$

The MGF Method [HL, Maguluri '20]

- Key Lemma:

Step 1:
$$\left(e^{\theta\epsilon q(k+1)} - 1\right)\left(e^{-\theta\epsilon u(k)} - 1\right) = 0$$



Dynamics of the queues:
$$q(k+1) = q(k) + a(k) - s(k) + u(k)$$
$$\Longrightarrow q(k+1)u(k) = 0$$

- From the lemma:

$$\mathbb{E}\left[e^{\theta\epsilon q(k+1)}\right] = \mathbb{E}\left[e^{\theta\epsilon(q(k)+a(k)-s(k))} - e^{-\theta\epsilon u(k)} + 1\right]$$

$$\Longrightarrow \mathbb{E}\left[e^{\theta\epsilon q}\right] = \frac{1 - \mathbb{E}\left[e^{-\theta\epsilon u}\right]}{1 - \mathbb{E}\left[e^{\theta\epsilon(a-s)}\right]}$$

Step 2: Bound unused service and take the heavy-traffic limit

$$\Longrightarrow \lim_{\epsilon\downarrow 0} \mathbb{E}\left[e^{\theta\epsilon q}\right] = \frac{1}{1 - \theta\left(\frac{\sigma_a^2 + \sigma_s^2}{2}\right)}$$

MGF of expo. random variable with mean $\dfrac{\sigma_a^2 + \sigma_s^2}{2}$

$$\epsilon q \Rightarrow Expo\left(\frac{2}{\sigma_a^2 + \sigma_s^2}\right)$$

Transform Techniques

Step 1: Prove an exponential version of $q(k+1)u(k) = 0$

Step 2: Bound unused service and take heavy-traffic limit

Transform Techniques

Step 1: Prove an exponential version of $q(k+1)u(k) = 0$

Step 2: Bound unused service and take heavy-traffic limit

Moment Generating Function

$$\mathbb{E}\left[e^{\theta \epsilon q}\right]$$

- $\theta \in \mathbb{R}$
- Two-sided Laplace transform of stationary distribution
- Must exist for θ in an interval around the origin

  ➡Must prove

Characteristic Function

$$\mathbb{E}\left[e^{i\theta \epsilon q}\right]$$

- $\theta \in \mathbb{R}, i = \sqrt{-1}$
- Fourier transform of stationary distribution
- Must use complex numbers

One-Sided Laplace Transform

$$\mathbb{E}\left[e^{\theta \epsilon q}\right]$$

- $\theta < 0$
- Always exists because $q \geq 0$
- Must show existence of moments

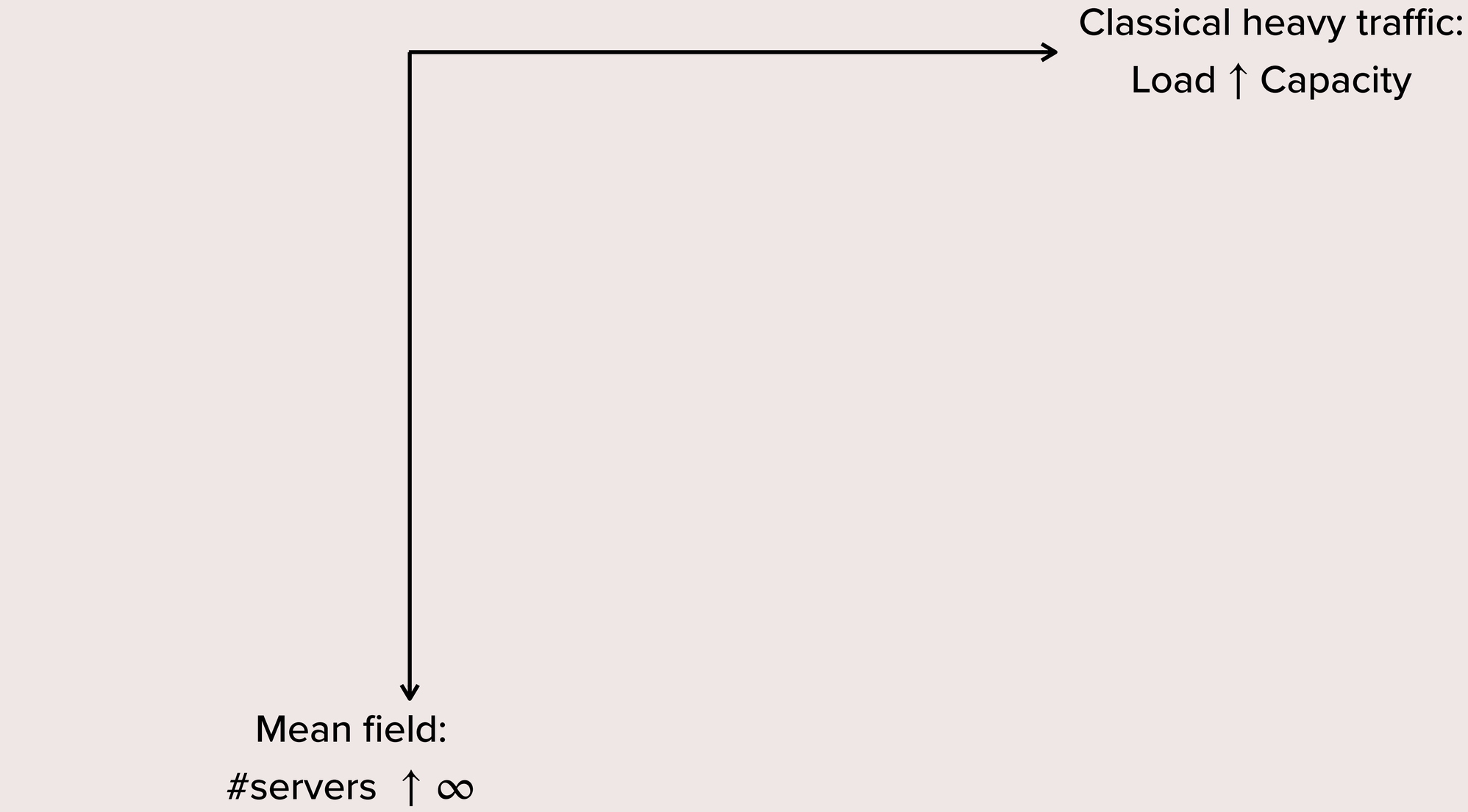
Outline

Part 1: The Model	• Important parts of a queueing model • Supermarket-checkout model • Some routing algorithms
Part 2: How to Solve?	• Stability • Heavy-Traffic: The CRP condition • Analysis of a single-server queue • The MGF method
Part 3: An Open Question	• Many-server heavy-traffic regime • Comparison of routing algorithms open question
Conclusion and future work	

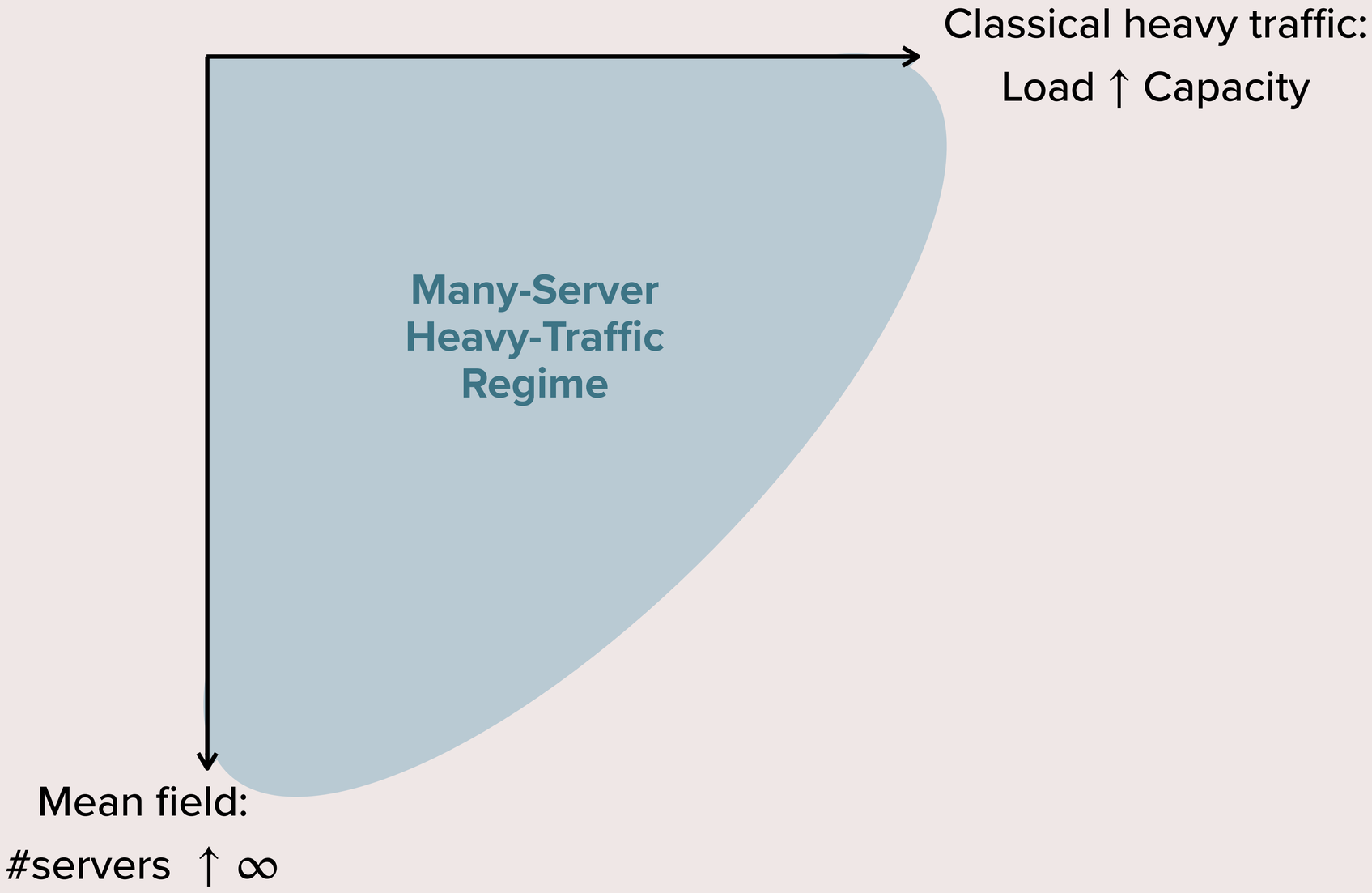
Outline

Part 1: **The Model**	• Important parts of a queueing model • Supermarket-checkout model • Some routing algorithms
Part 2: **How to Solve?**	• Stability • Heavy-Traffic: The CRP condition • Analysis of a single-server queue • The MGF method
Part 3: **An Open Question**	• Many-server heavy-traffic regime • Comparison of routing algorithms open question
Conclusion and future work	

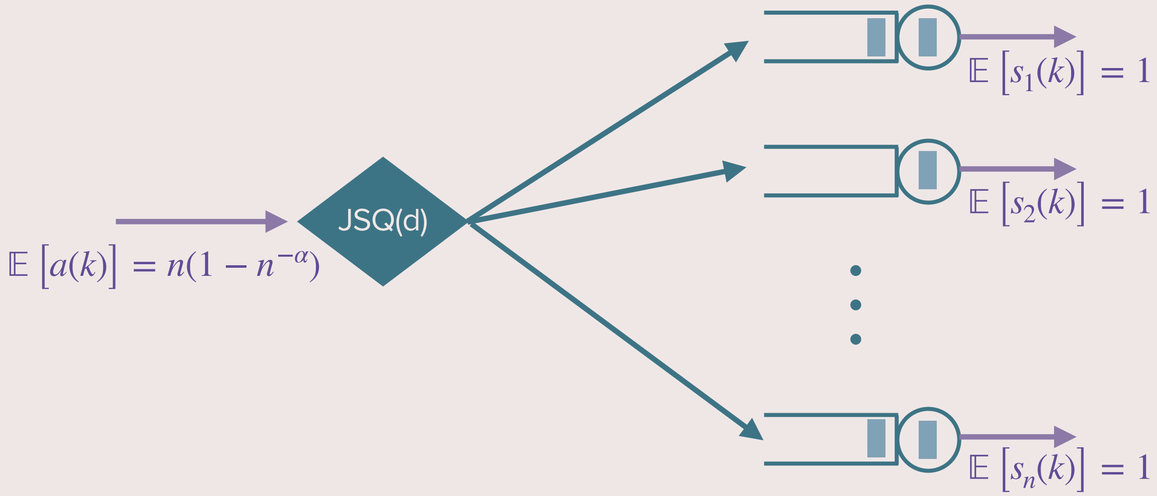
Many-Server Heavy-Traffic Regime



Many-Server Heavy-Traffic Regime



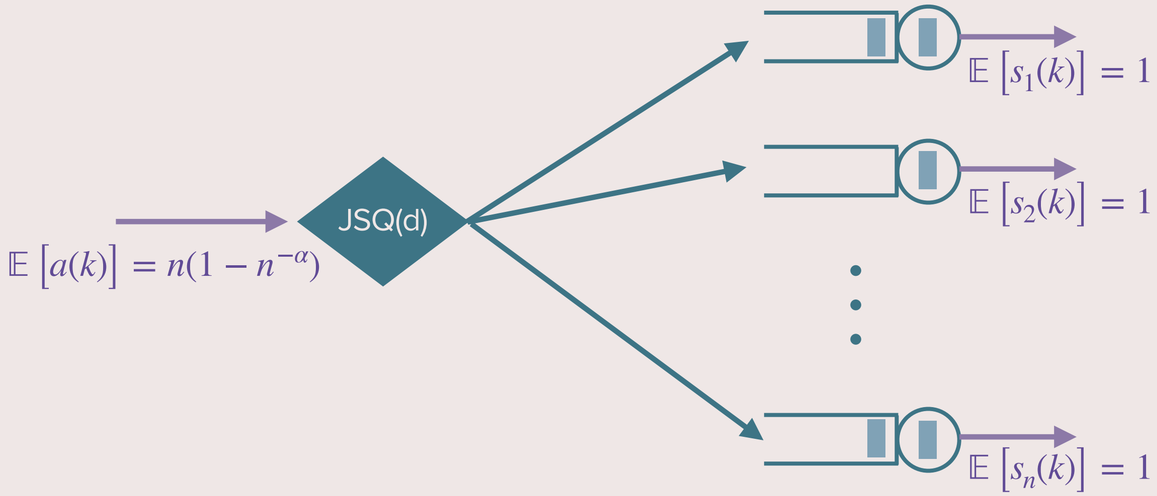
Many-Server Heavy-Traffic Parametrization



Many-Server Heavy-Traffic Parametrization

Service process:

- All servers are equal and independent of each other
- $\mathbb{E}\left[s_i(1)\right] = 1$



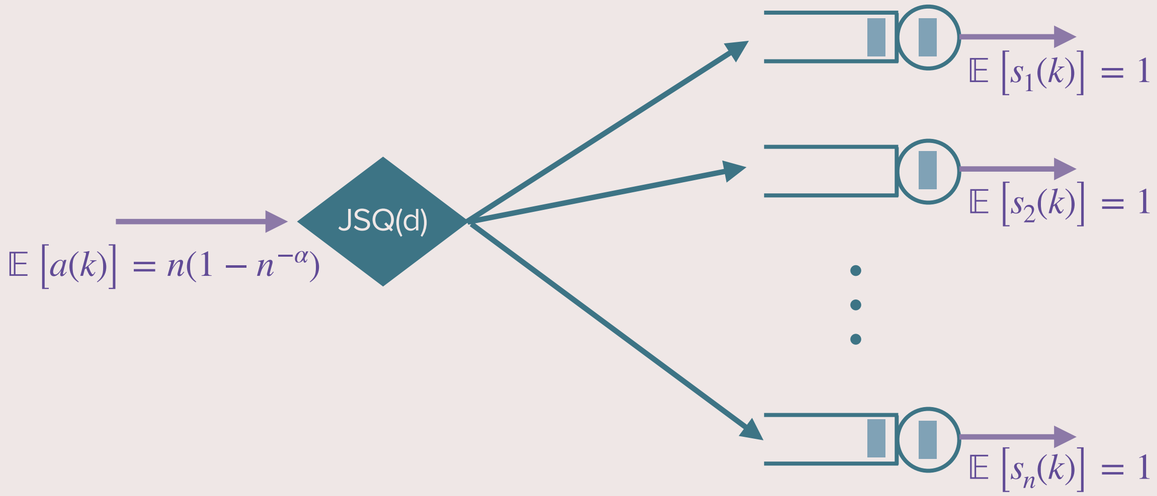
Many-Server Heavy-Traffic Parametrization

Service process:

- All servers are equal and independent of each other
- $\mathbb{E}\left[s_i(1)\right] = 1$

Arrival process:

- Consider $\alpha > 0$
- $\mathbb{E}\left[a(1)\right] = n(1 - n^{-\alpha})$
- *Arrivals "per server" have mean* $1 - n^{-\alpha}$



Many-Server Heavy-Traffic Parametrization

Service process:

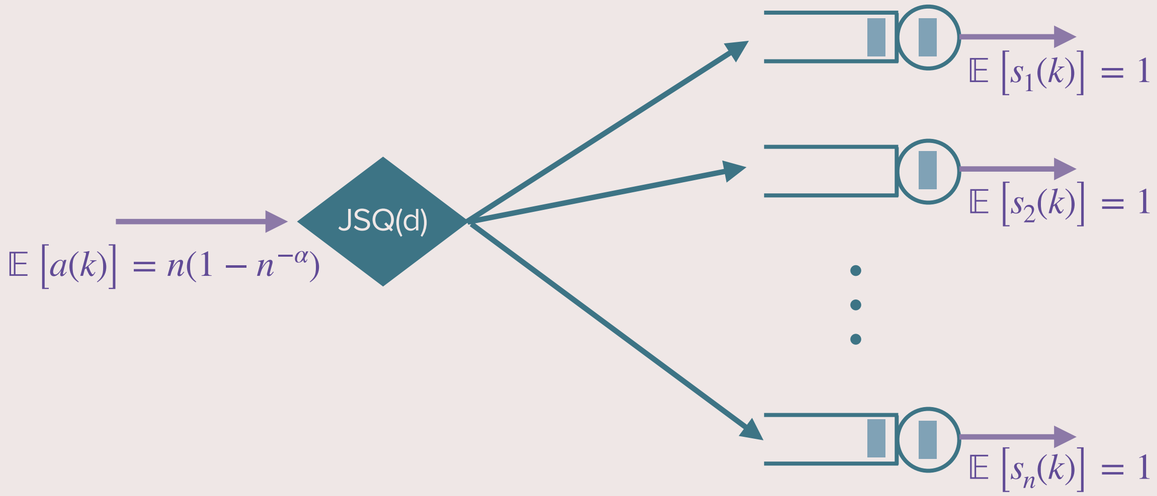
- All servers are equal and independent of each other
- $\mathbb{E}\left[s_i(1)\right] = 1$

Arrival process:

- Consider $\alpha > 0$
- $\mathbb{E}\left[a(1)\right] = n(1 - n^{-\alpha})$
- *Arrivals "per server" have mean* $1 - n^{-\alpha}$

Routing:

- Power-of-d choices



Many-Server Heavy-Traffic Parametrization

Service process:

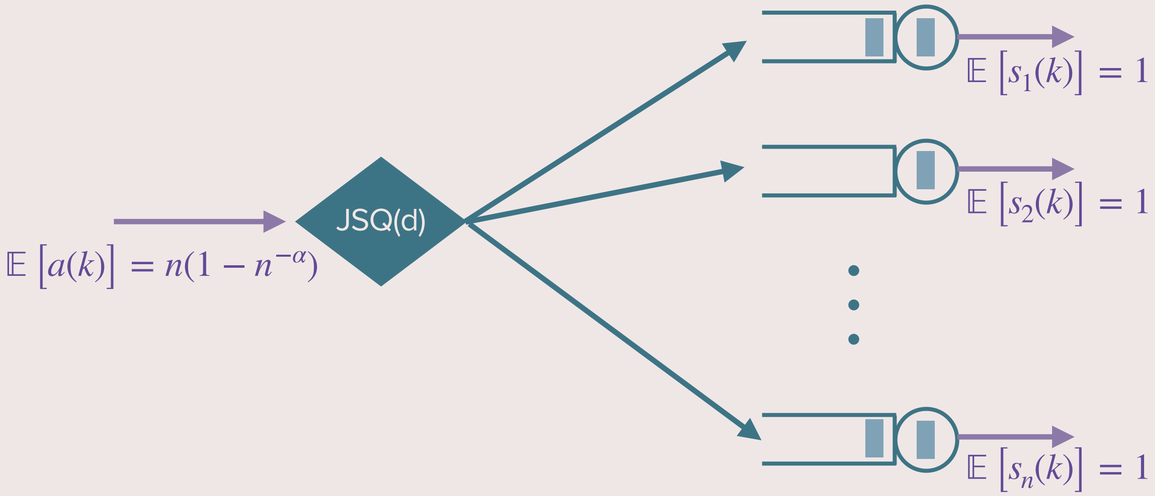
- All servers are equal and independent of each other
- $\mathbb{E}\left[s_i(1)\right] = 1$

Arrival process:

- Consider $\alpha > 0$
- $\mathbb{E}\left[a(1)\right] = n(1 - n^{-\alpha})$
- *Arrivals "per server" have mean* $1 - n^{-\alpha}$

Routing:

- Power-of-d choices



Many-server heavy-traffic limit:

Many-Server Heavy-Traffic Parametrization

Service process:

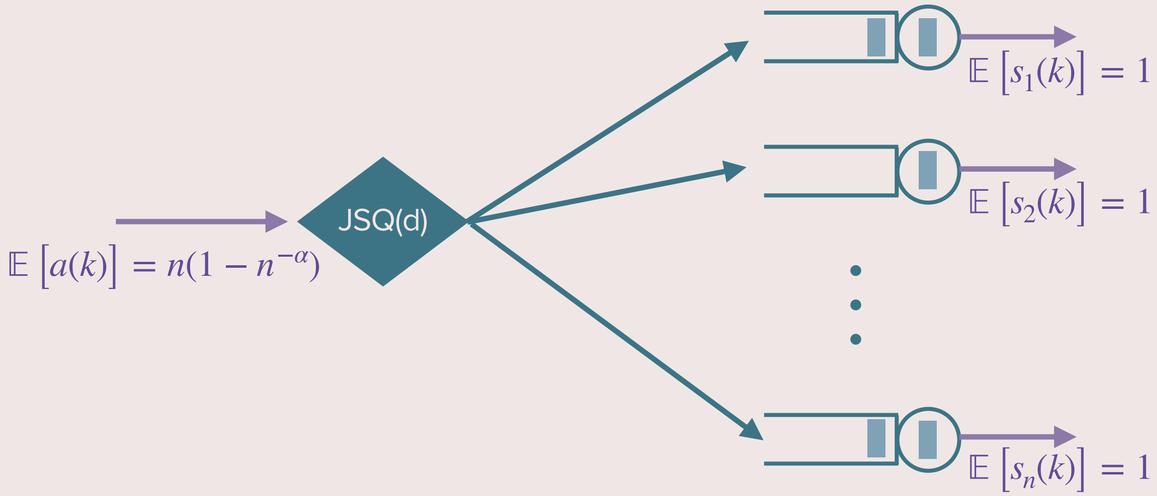
- All servers are equal and independent of each other
- $\mathbb{E}\left[s_i(1)\right] = 1$

Arrival process:

- Consider $\alpha > 0$
- $\mathbb{E}\left[a(1)\right] = n(1 - n^{-\alpha})$
- *Arrivals "per server" have mean* $1 - n^{-\alpha}$

Routing:

- Power-of-d choices



Many-server heavy-traffic limit:

- Take $n \uparrow \infty$

Many-Server Heavy-Traffic Parametrization

Service process:

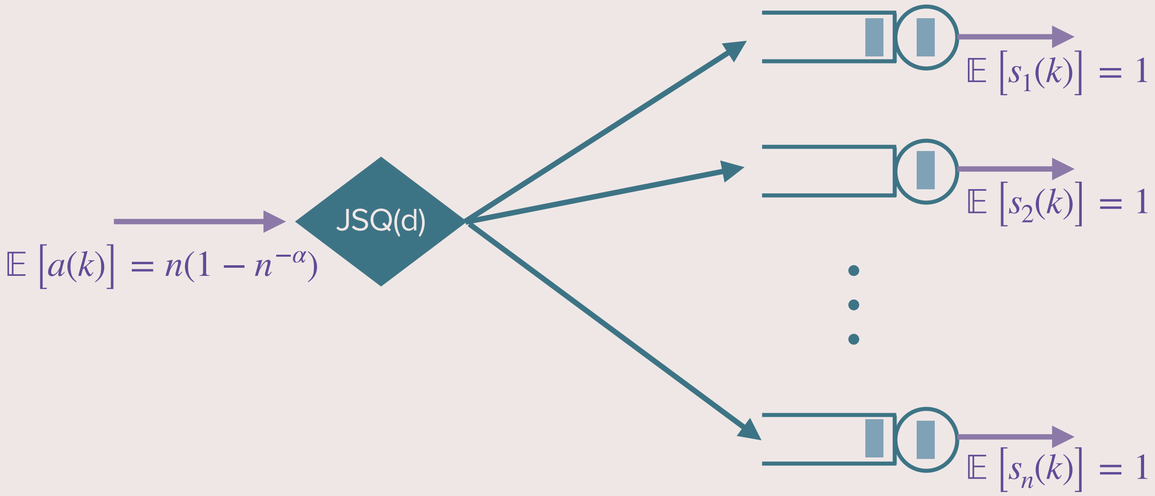
- All servers are equal and independent of each other
- $\mathbb{E}\left[s_i(1)\right] = 1$

Arrival process:

- Consider $\alpha > 0$
- $\mathbb{E}\left[a(1)\right] = n(1 - n^{-\alpha})$
- *Arrivals "per server" have mean* $1 - n^{-\alpha}$

Routing:

- Power-of-d choices



Many-server heavy-traffic limit:

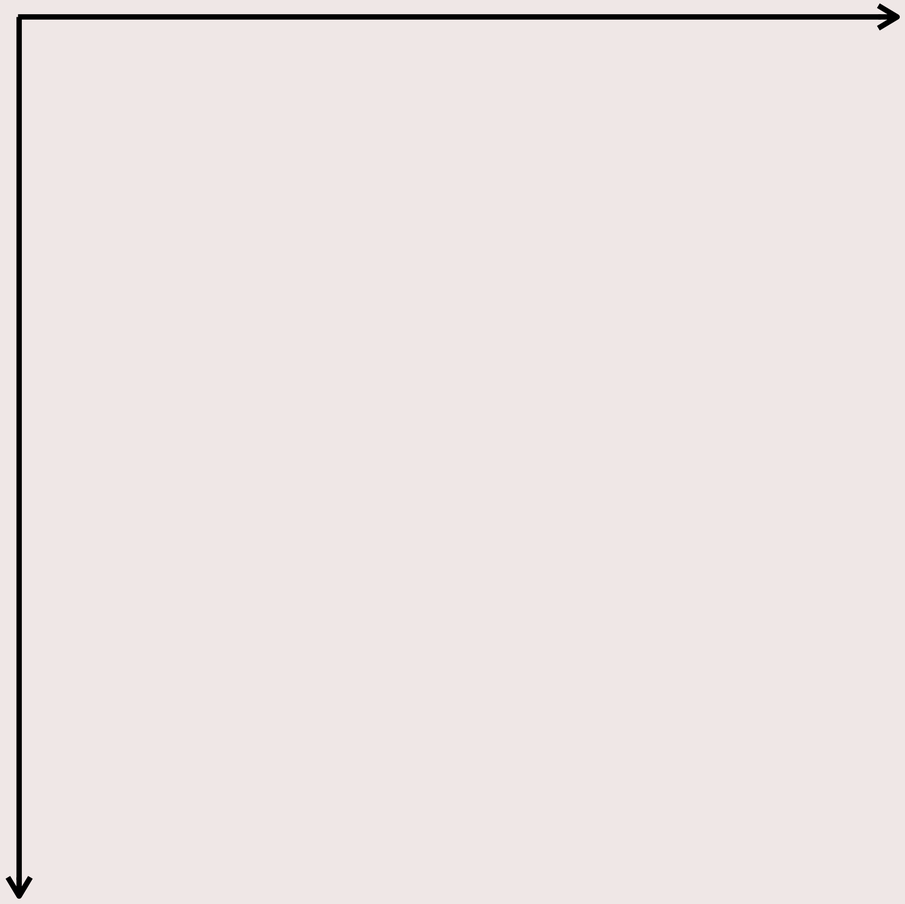
- Take $n \uparrow \infty$
- Then,
  - # servers $\uparrow \infty$
  - Load $\uparrow$ Max capacity

Many-Server Heavy-Traffic Regime

- Total service rate = n
- Arrival rate *per server* = $1 - n^{-\alpha}$

Classical heavy traffic:

"$\alpha \uparrow \infty$"

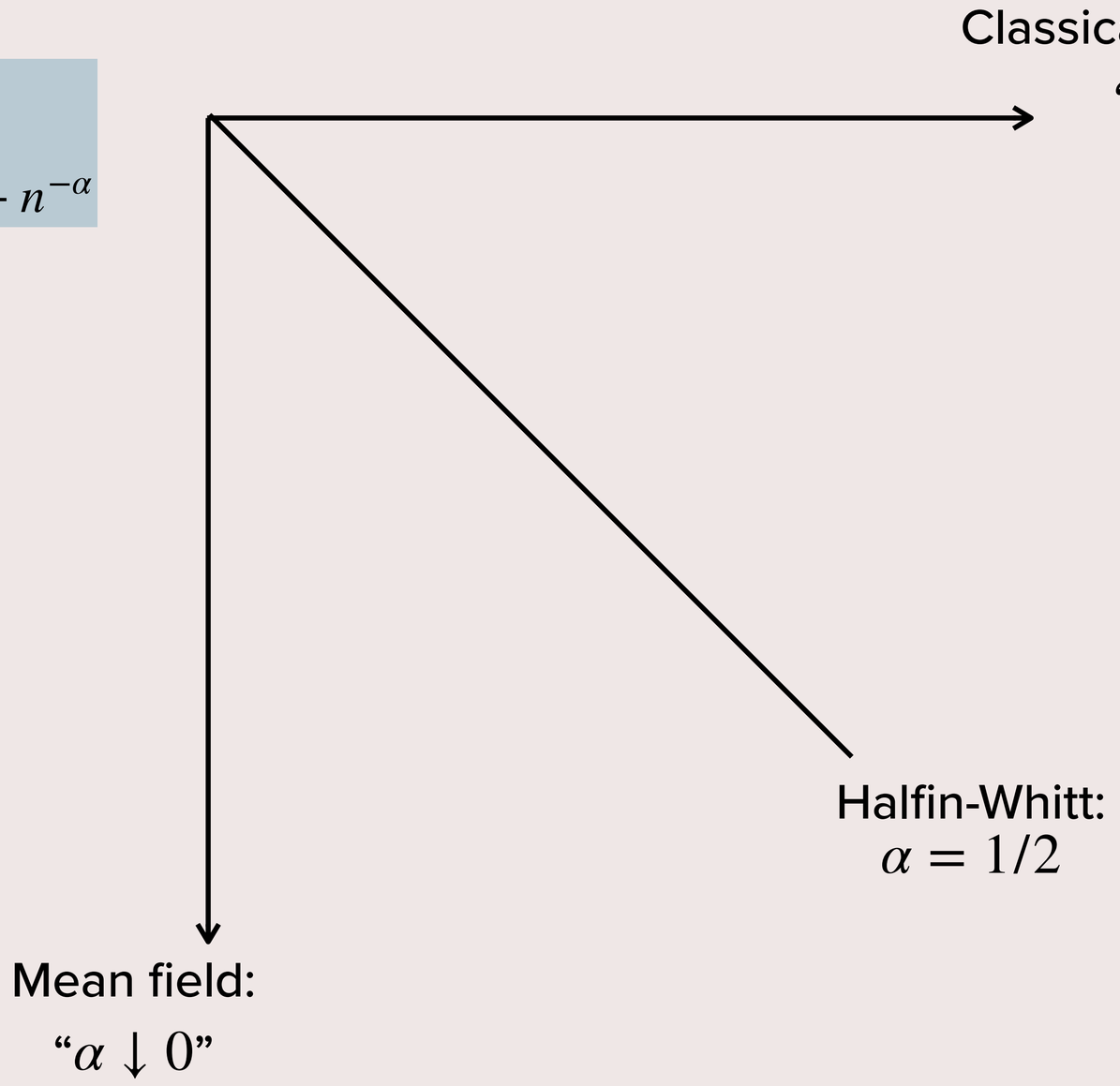


Mean field:

"$\alpha \downarrow 0$"

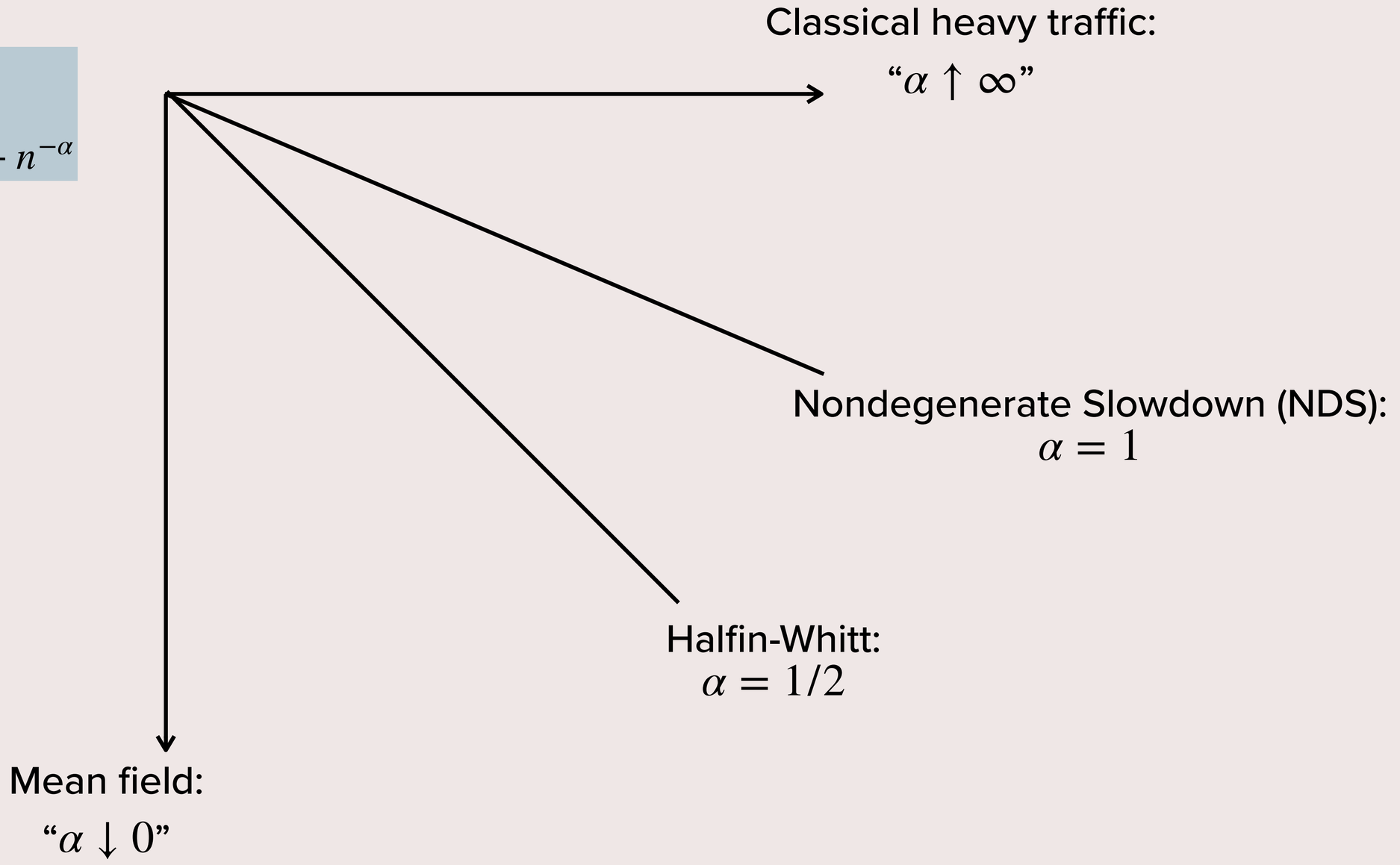
Many-Server Heavy-Traffic Regime

- Total service rate = n
- Arrival rate *per server* = $1 - n^{-\alpha}$



Many-Server Heavy-Traffic Regime

- Total service rate = n
- Arrival rate *per server* = $1 - n^{-\alpha}$



Many-Server Heavy-Traffic Regime

- Total service rate = n
- Arrival rate *per server* = $1 - n^{-\alpha}$

Foschini and Salz(1978); Harrison and López (1999); Eryilmaz and Srikant (2012); Braverman et.al. (2017); HL and Maguluri (2020) ...

Classical heavy traffic:

"$\alpha \uparrow \infty$"

Liu and Ying (2019); Zhao et. al. (2021)

Nondegenerate Slowdown (NDS):

$\alpha = 1$

Atar (2012); Gupta and Walton (2019)

Liu and Ying (2019), Weng and Wang (2020)

Vvedenskaya et al. (1996); Mitzenmacher (1996); Mitzenmacher (2001); Mukherjee et. al. (2018)...

Halfin-Whitt:

$\alpha = 1/2$

Halfin and Whitt (1981); Eschenfeldt and Gamarnik (2018); Banerjee and Mukherjee (2019,2020); Braverman (2020)...

Mean field:

"$\alpha \downarrow 0$"

Many-Server Heavy-Traffic Regime

- Total service rate = n
- Arrival rate *per server* = $1 - n^{-\alpha}$

Foschini and Salz(1978); Harrison and López (1999); Eryilmaz and Srikant (2012); Braverman et.al. (2017); HL and Maguluri (2020) ...

Classical heavy traffic:

"$\alpha \uparrow \infty$"

Next:
Value of $\alpha > 1$ to observe classical heavy-traffic behavior?

Liu and Ying (2019); Zhao et. al. (2021)

Nondegenerate Slowdown (NDS):

$\alpha = 1$

Atar (2012); Gupta and Walton (2019)

Liu and Ying (2019), Weng and Wang (2020)

Vvedenskaya et al. (1996); Mitzenmacher (1996); Mitzenmacher (2001); Mukherjee et. al. (2018)...

Halfin-Whitt:

$\alpha = 1/2$

Halfin and Whitt (1981); Eschenfeldt and Gamarnik (2018); Banerjee and Mukherjee (2019,2020); Braverman (2020)...

Mean field:

"$\alpha \downarrow 0$"

Many-Server Heavy-Traffic Regime

- Total service rate = n
- Arrival rate *per server* = $1 - n^{-\alpha}$

Foschini and Salz(1978); Harrison and López (1999); Eryilmaz and Srikant (2012); Braverman et.al. (2017); HL and Maguluri (2020) ...

Classical heavy traffic:

"$\alpha \uparrow \infty$"

Next:
Value of $\alpha > 1$ to observe classical heavy-traffic behavior?

$\alpha > 2$

Liu and Ying (2019); Zhao et. al. (2021)

Nondegenerate Slowdown (NDS):

$\alpha = 1$

Atar (2012); Gupta and Walton (2019)

Liu and Ying (2019), Weng and Wang (2020)

Vvedenskaya et al. (1996); Mitzenmacher (1996); Mitzenmacher (2001); Mukherjee et. al. (2018)...

Halfin-Whitt:

$\alpha = 1/2$

Halfin and Whitt (1981); Eschenfeldt and Gamarnik (2018); Banerjee and Mukherjee (2019,2020); Braverman (2020)...

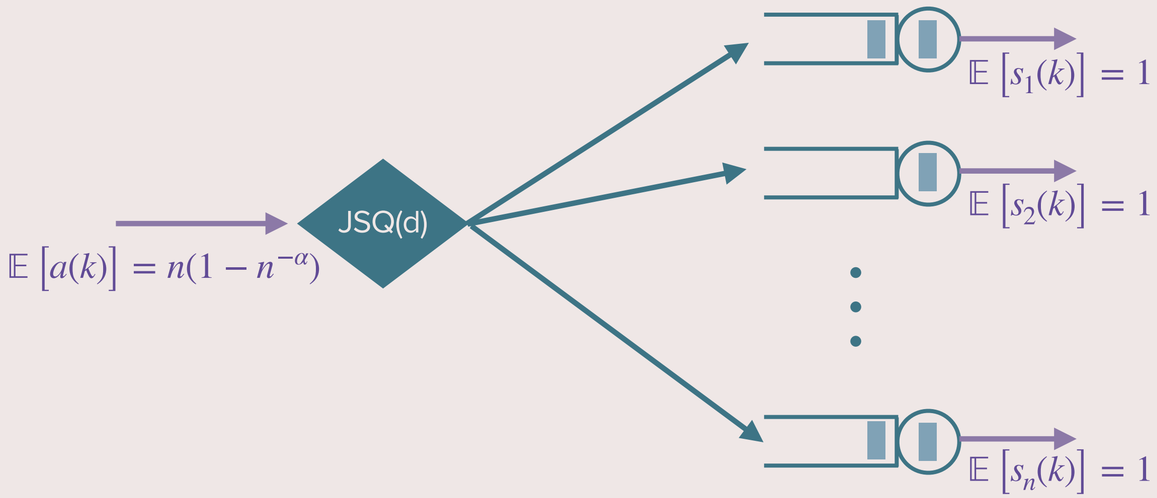
Mean field:

"$\alpha \downarrow 0$"

Before We Get to the Result...

Routing:

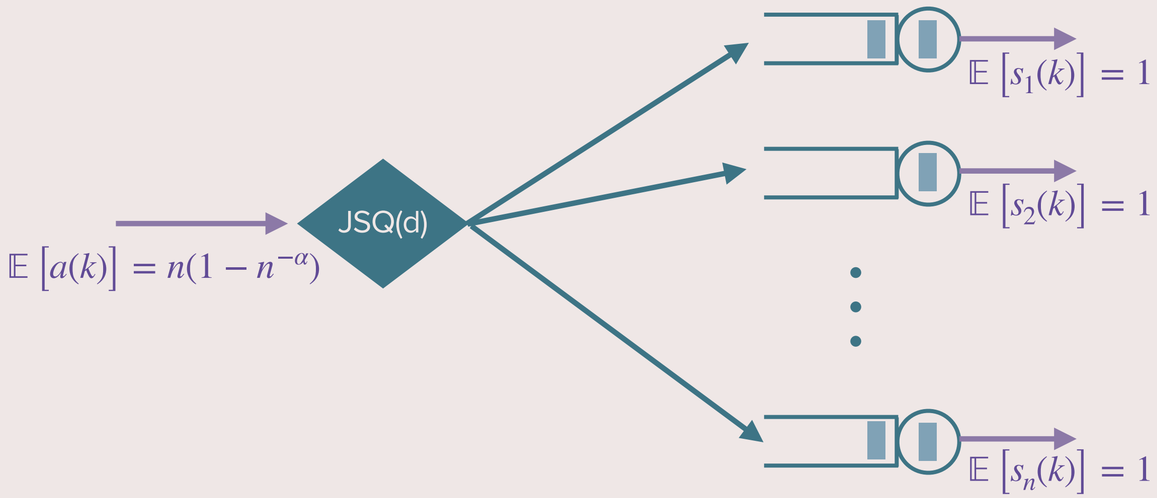
- Power-of-d choices
- In each time slot, route **all the arrivals to one server**



Before We Get to the Result...

Routing:

- Power-of-d choices
- In each time slot, route **all the arrivals to one server**

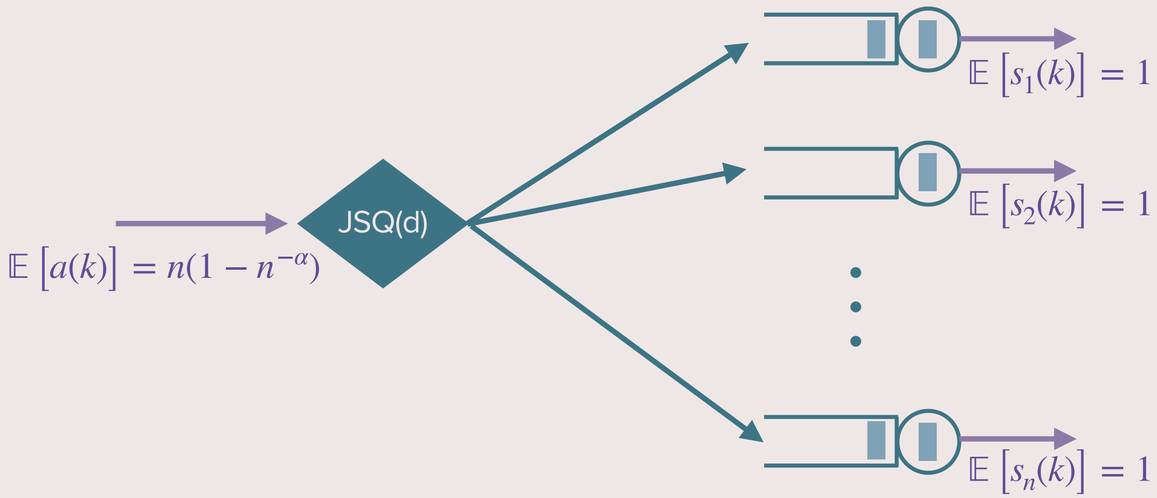


Warning!

Before We Get to the Result...

Routing:

- Power-of-d choices
- In each time slot, route **all the arrivals to one server**



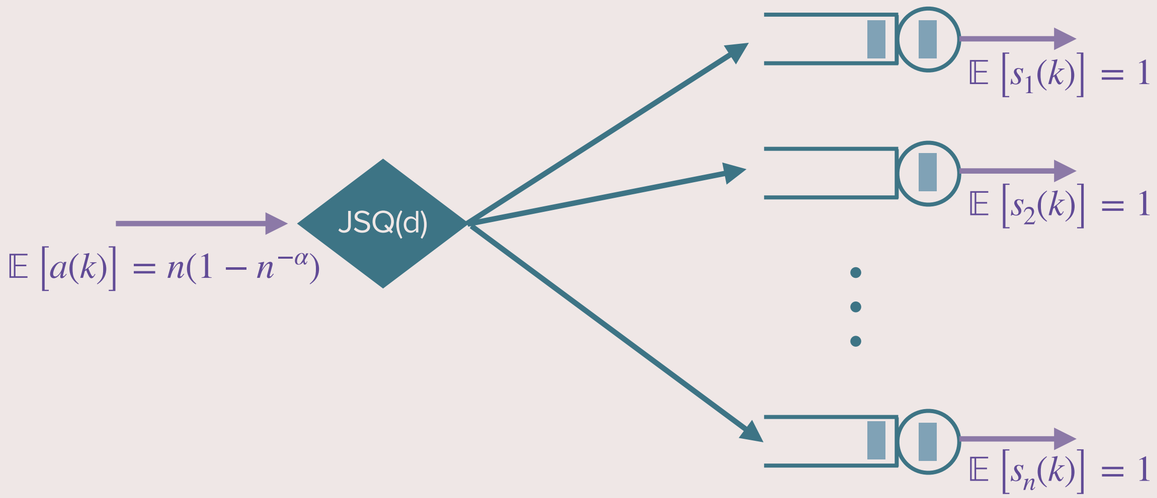
Warning!

- We want classical heavy-traffic behavior

Before We Get to the Result...

Routing:

- Power-of-d choices
- In each time slot, route **all the arrivals to one server**



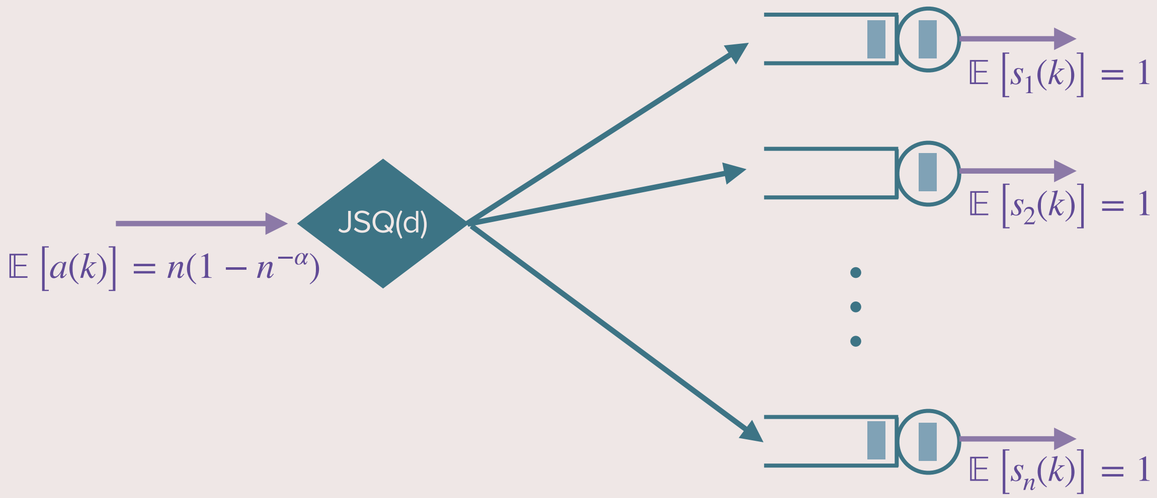
Warning!

- We want classical heavy-traffic behavior
- We expect CRP condition

Before We Get to the Result...

Routing:

- Power-of-d choices
- In each time slot, route **all the arrivals to one server**



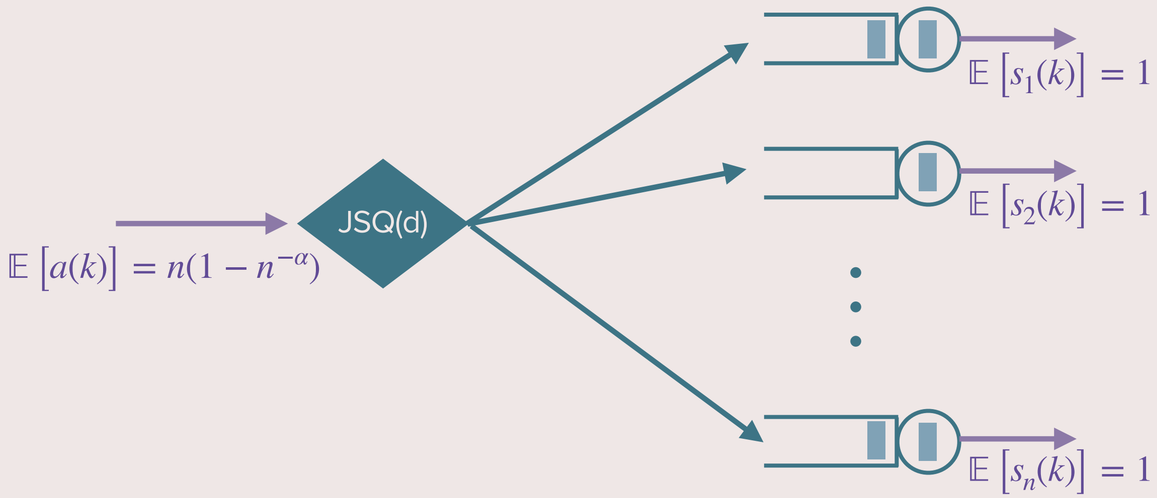
Warning!

- We want classical heavy-traffic behavior
- We expect CRP condition
- In every time slot we route $O(n)$ jobs

Before We Get to the Result...

Routing:

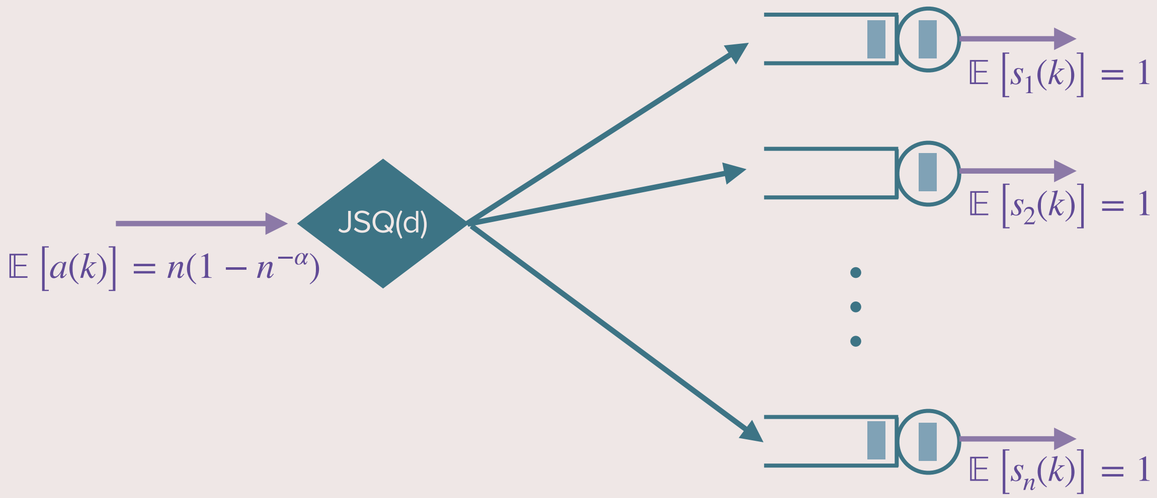
- Power-of-d choices
- In each time slot, route **all the arrivals to one server**



Warning!

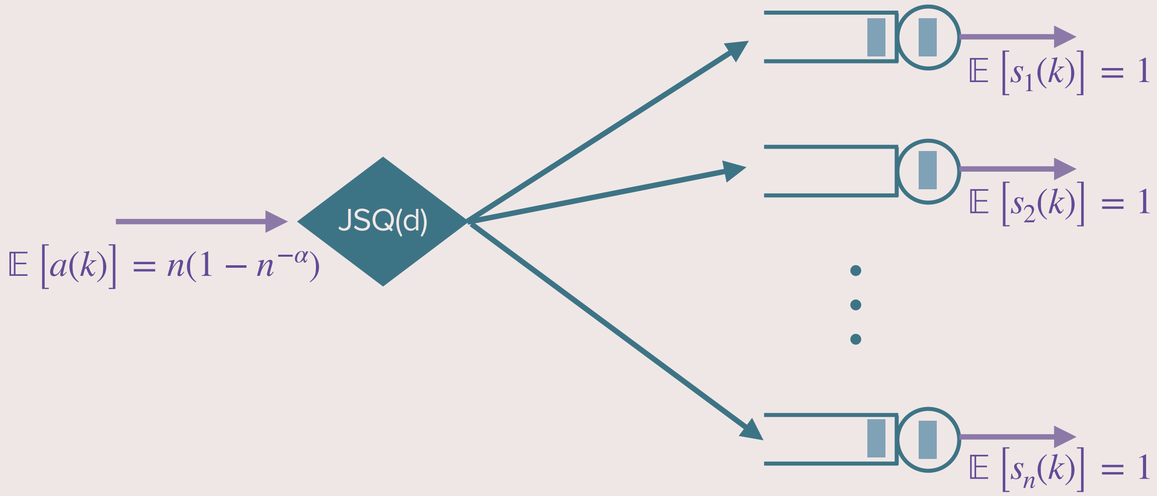
- We want classical heavy-traffic behavior
- We expect CRP condition
- In every time slot we route $O(n)$ jobs
- Then, convergence to CRP is weak

Supermarket Checkout Model in Continuous Time



Supermarket Checkout Model in Continuous Time

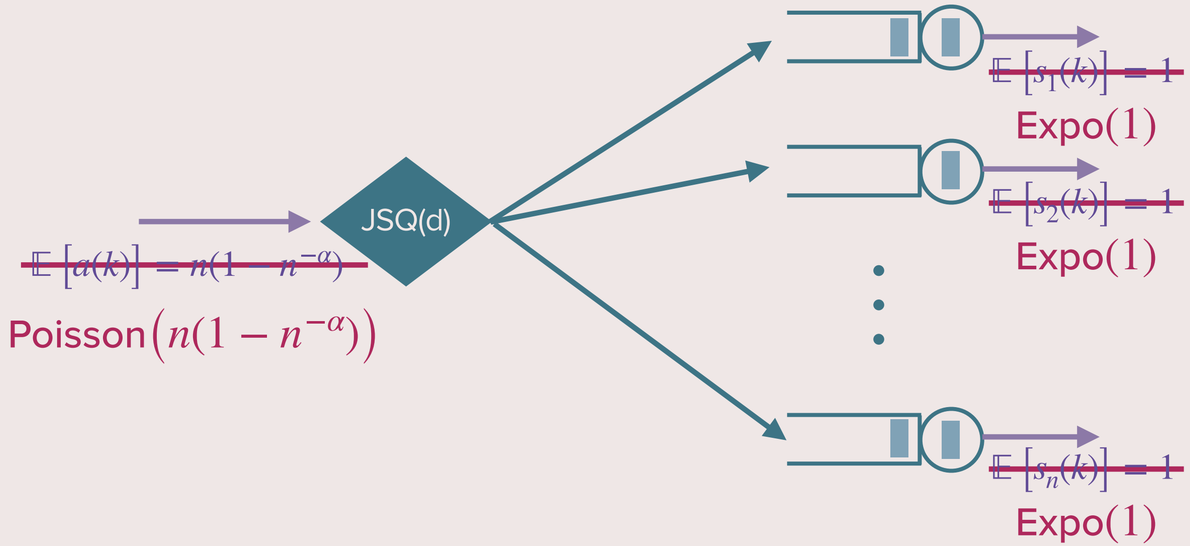
Model:



Supermarket Checkout Model in Continuous Time

Model:

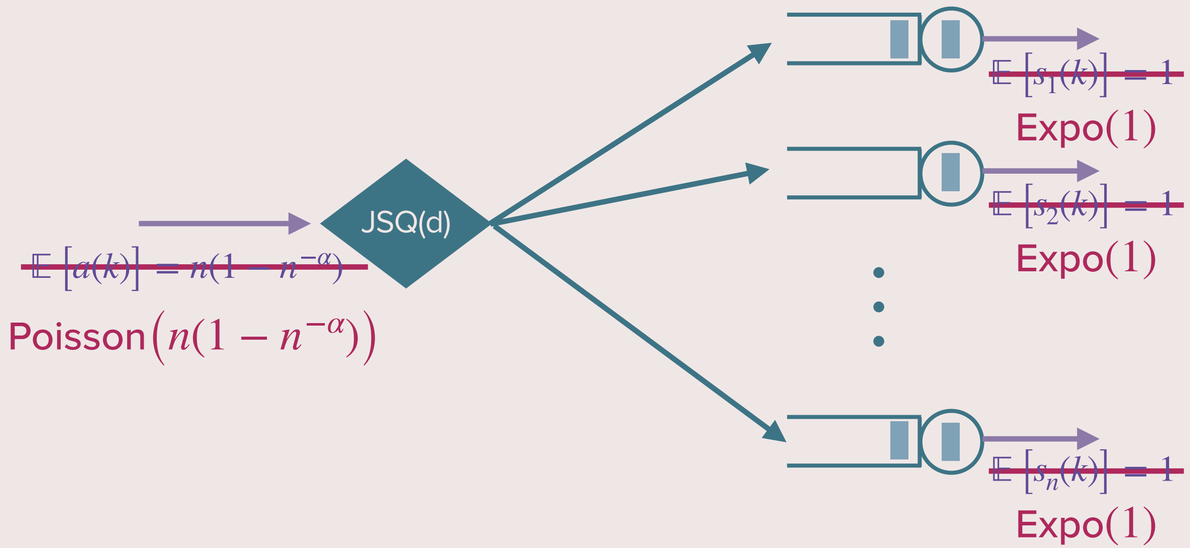
- Arrivals $\sim$ Poisson $\left(n(1-n^{-\alpha})\right)$
- Service time $\sim$ Expo(1)
  - All servers are equal



Supermarket Checkout Model in Continuous Time

Model:

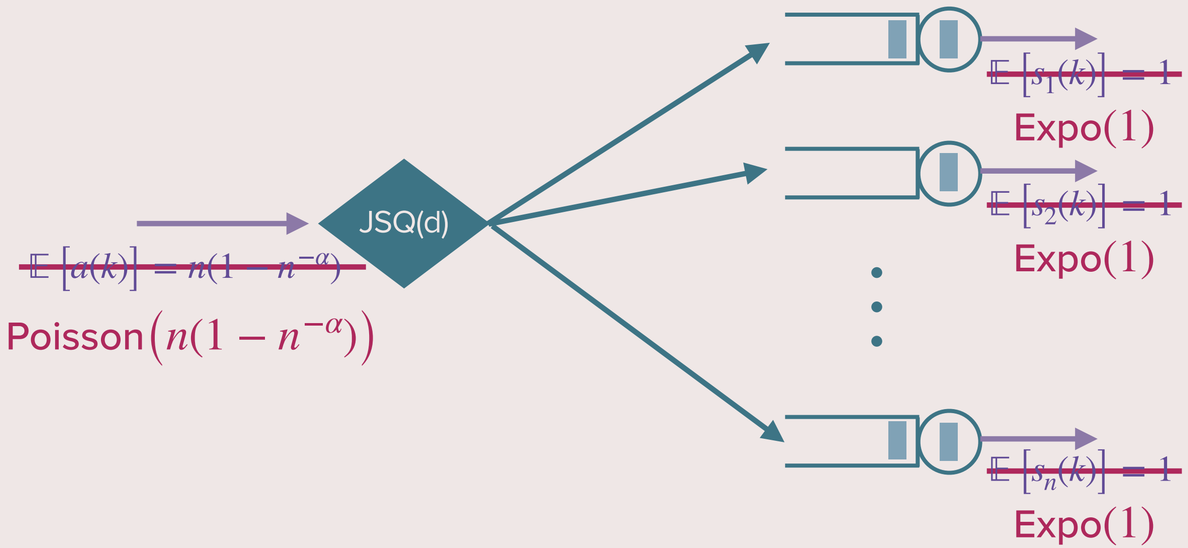
- Arrivals $\sim$ Poisson $\left(n(1-n^{-\alpha})\right)$
- Service time $\sim$ Expo(1)
  - All servers are equal
- Routing: JSQ(d)
  - $d = cn^{\beta}, c > 0, \beta \geq 0$



Supermarket Checkout Model in Continuous Time

Model:

- Arrivals $\sim$ Poisson $\left(n(1-n^{-\alpha})\right)$
- Service time $\sim$ Expo(1)
  - All servers are equal
- Routing: JSQ(d)
  - $d = cn^{\beta}, c > 0, \beta \geq 0$



Theorem: [HL, Maguluri '21]

Consider a supermarket checkout system as described above. If $\alpha + \beta > 3$, then as $n \uparrow \infty$

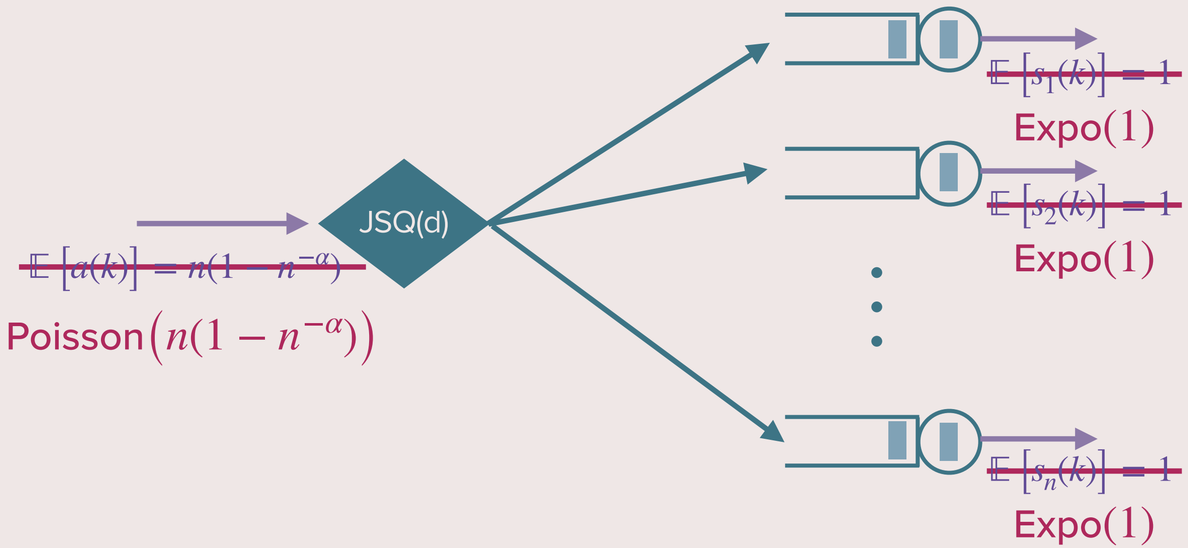
$$n^{-\alpha} \sum_i q_i \Rightarrow Expo(1)$$

and $$\lim_{n\uparrow\infty} n^{-\alpha}\mathbb{E}\left[\sum_i q_i\right] = 1$$

Supermarket Checkout Model in Continuous Time

Model:

- Arrivals $\sim$ Poisson $\left(n(1-n^{-\alpha})\right)$
- Service time $\sim$ Expo(1)
  - All servers are equal
- Routing: JSQ(d)
  - $d = cn^{\beta}, c > 0, \beta \geq 0$



Theorem: [HL, Maguluri '21]

Consider a supermarket checkout system as described above. If $\alpha + \beta > 3$, then as $n \uparrow \infty$

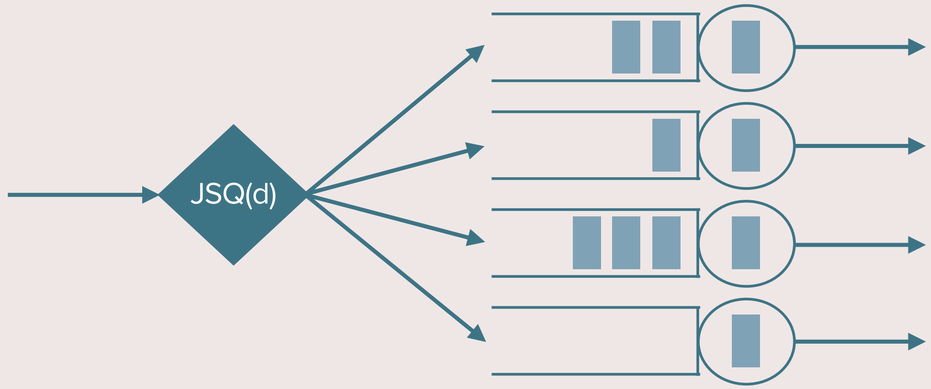
$$n^{-\alpha} \sum_i q_i \Rightarrow Expo(1)$$

$$\text{and } \lim_{n\uparrow\infty} n^{-\alpha}\mathbb{E}\left[\sum_i q_i\right] = 1$$

Proof:

1. Prove CRP
2. MGF method

Many-Server Heavy-Traffic: Which One is Better?

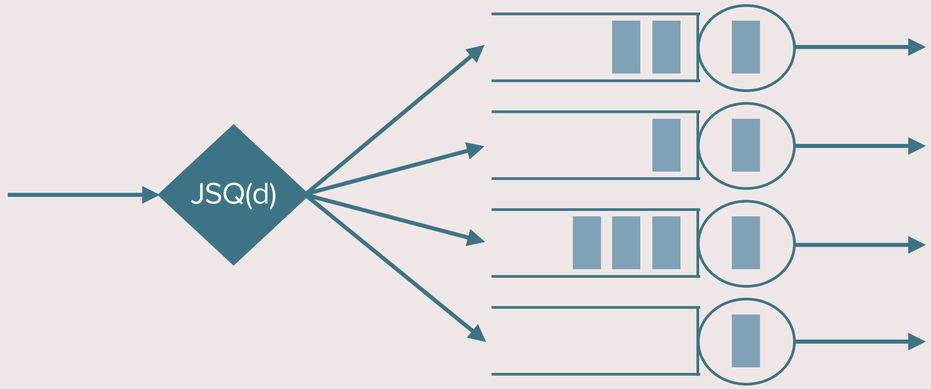


Theorem: [HL, Maguluri '21]

If $d = cn^{\beta}$ and $\alpha + \beta > 3$, then as $n \uparrow \infty$

$$n^{-\alpha} \sum_i q_i \Rightarrow Expo(1)$$

Many-Server Heavy-Traffic: Which One is Better?



Theorem: [HL, Maguluri '21]

If $d = cn^{\beta}$ and $\alpha + \beta > 3$, then as $n \uparrow \infty$

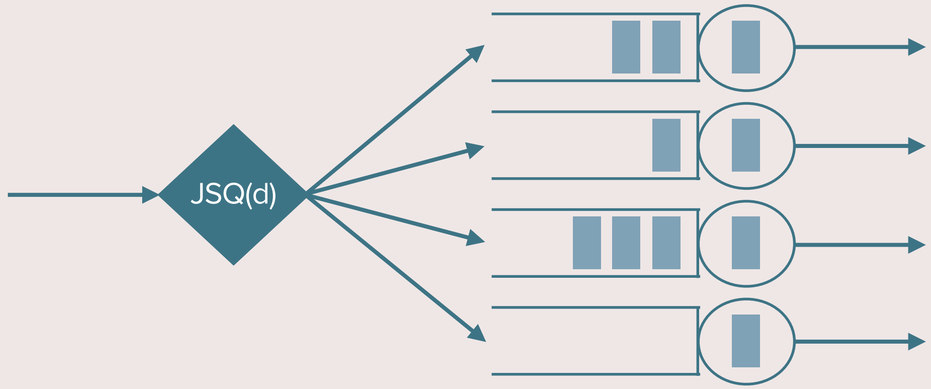
$$n^{-\alpha} \sum_i q_i \Rightarrow Expo(1)$$

Random

JSQ(d)

JSQ

Many-Server Heavy-Traffic: Which One is Better?



Theorem: [HL, Maguluri '21]

If $d = cn^{\beta}$ and $\alpha + \beta > 3$, then as $n \uparrow \infty$

$$n^{-\alpha} \sum_i q_i \Rightarrow Expo(1)$$

Random

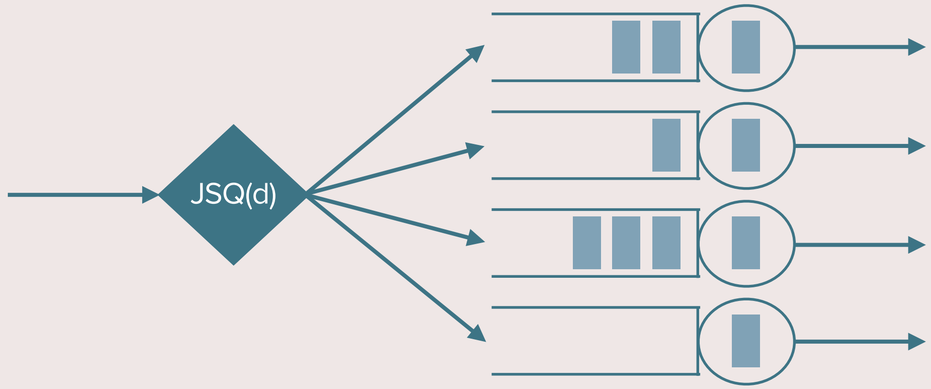
JSQ(d)

JSQ

$$d = cn^{\beta}$$

$$\alpha + \beta > 3$$

Many-Server Heavy-Traffic: Which One is Better?



Theorem: [HL, Maguluri '21]

If $d = cn^{\beta}$ and $\alpha + \beta > 3$, then as $n \uparrow \infty$

$$n^{-\alpha} \sum_i q_i \Rightarrow Expo(1)$$

Random

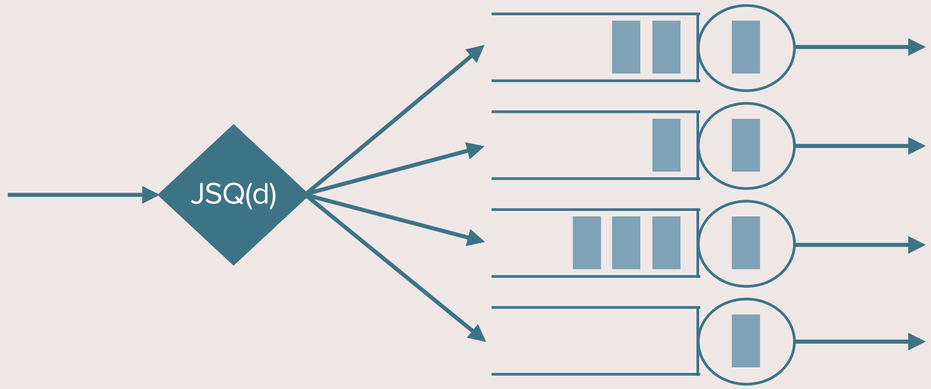
JSQ(d)

$$d = cn^{\beta}$$
$$\alpha + \beta > 3$$

JSQ

$$\alpha > 2$$

Many-Server Heavy-Traffic: Which One is Better?



Theorem: [HL, Maguluri '21]

If $d = cn^{\beta}$ and $\alpha + \beta > 3$, then as $n \uparrow \infty$

$$n^{-\alpha} \sum_i q_i \Rightarrow Expo(1)$$

Random

$\alpha > 3$

JSQ(d)

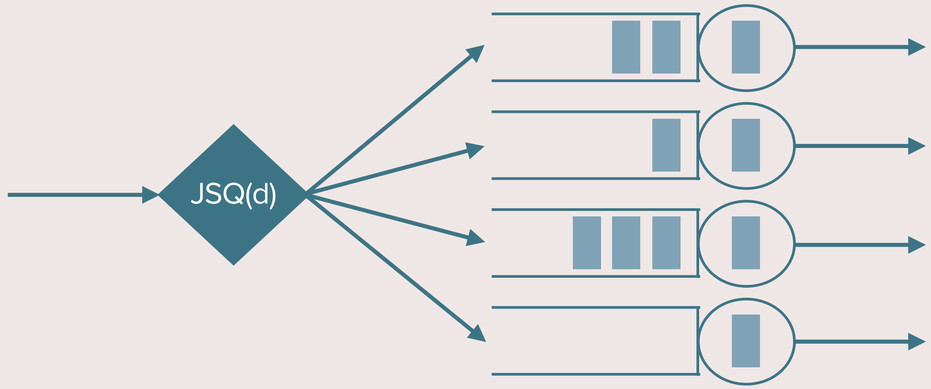
$d = cn^{\beta}$

$\alpha + \beta > 3$

JSQ

$\alpha > 2$

Many-Server Heavy-Traffic: Which One is Better?



Theorem: [HL, Maguluri '21]

If $d = cn^{\beta}$ and $\alpha + \beta > 3$, then as $n \uparrow \infty$

$$n^{-\alpha} \sum_i q_i \Rightarrow Expo(1)$$

Random $\leq$ **JSQ(d)** $<$ **JSQ**

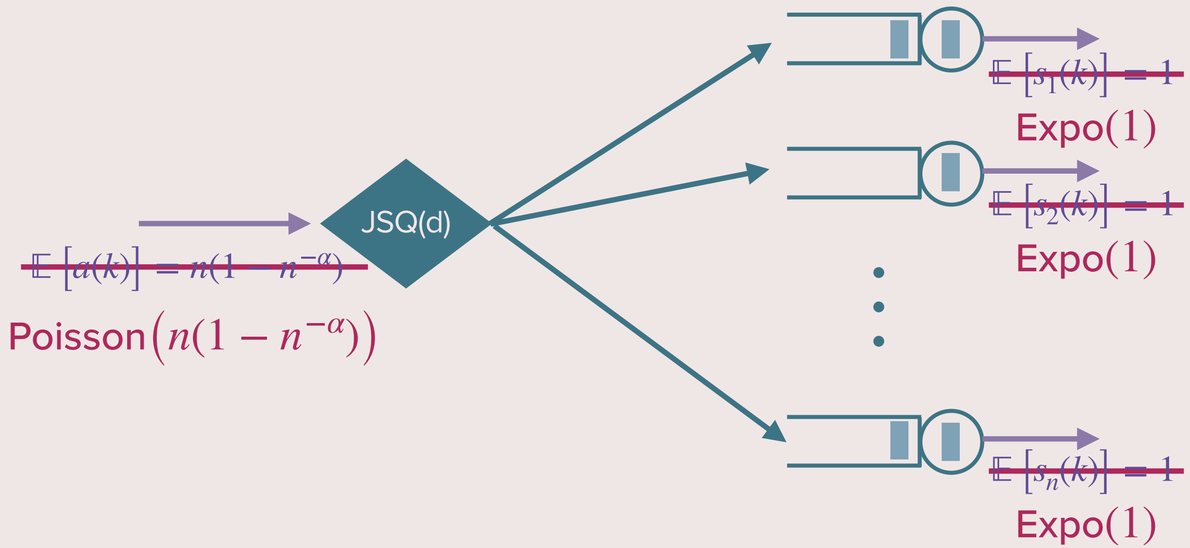
$\alpha > 3$

$d = cn^{\beta}$

$\alpha + \beta > 3$

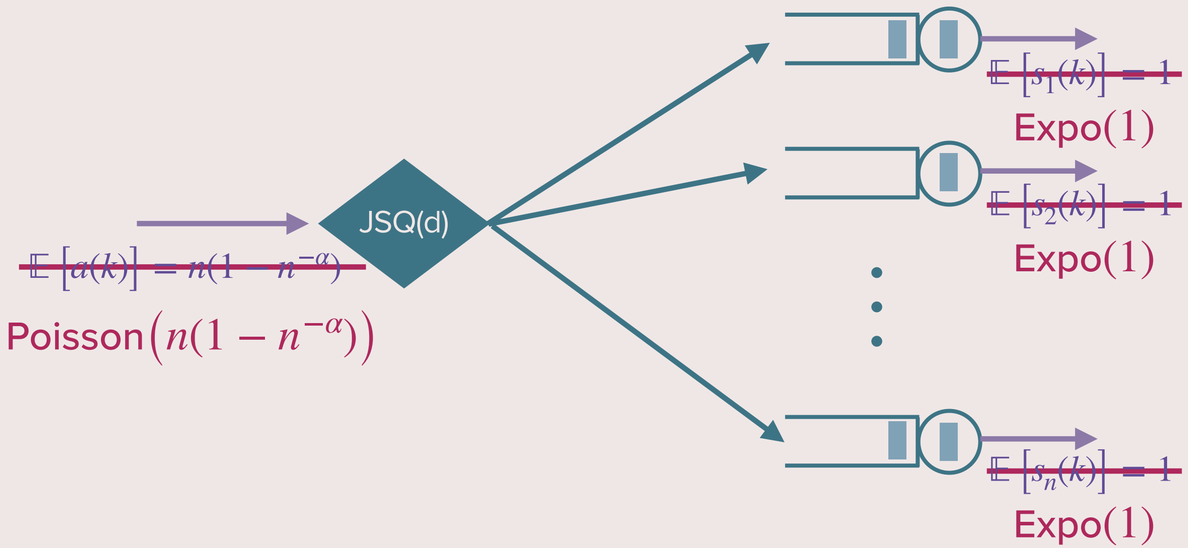
$\alpha > 2$

Why can't we get $\alpha > 1$?



Why can't we get $\alpha > 1$?

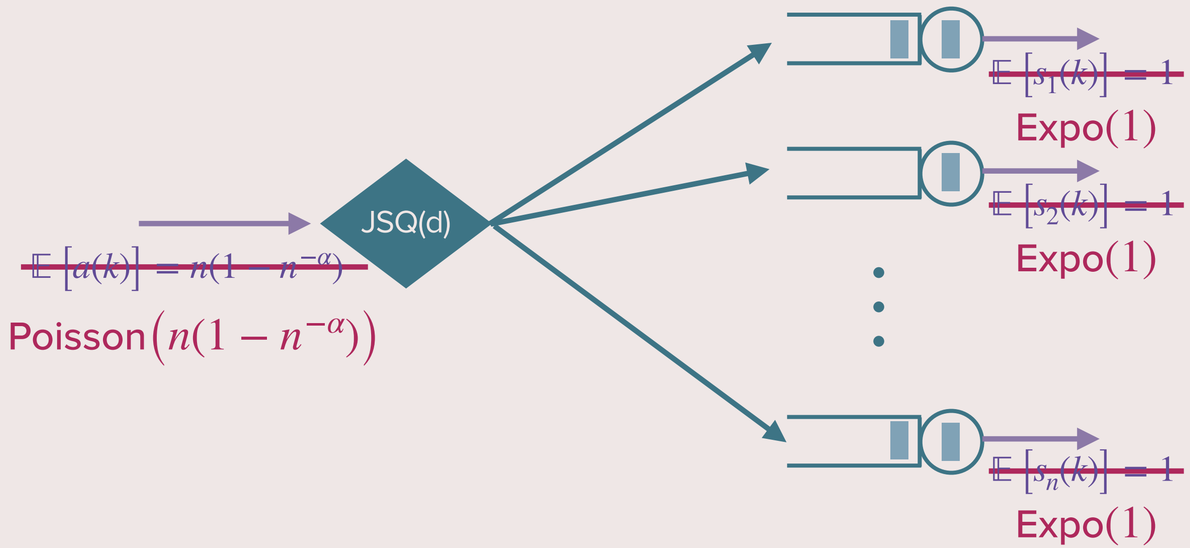
Bottleneck in the proof:



Why can't we get $\alpha > 1$?

Bottleneck in the proof:

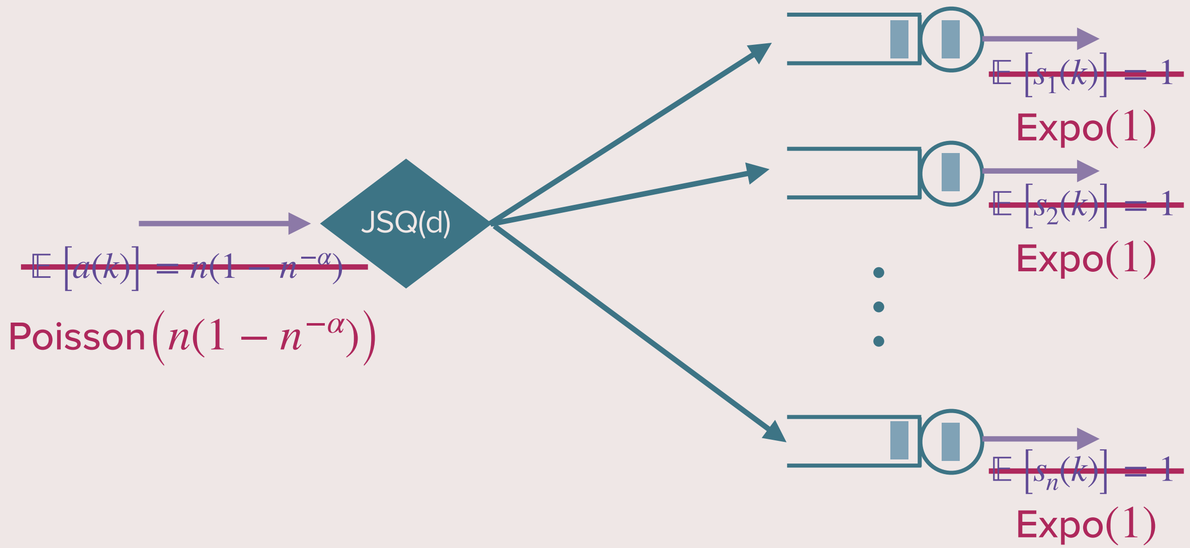
- Convergence to CRP



Why can't we get $\alpha > 1$?

Bottleneck in the proof:

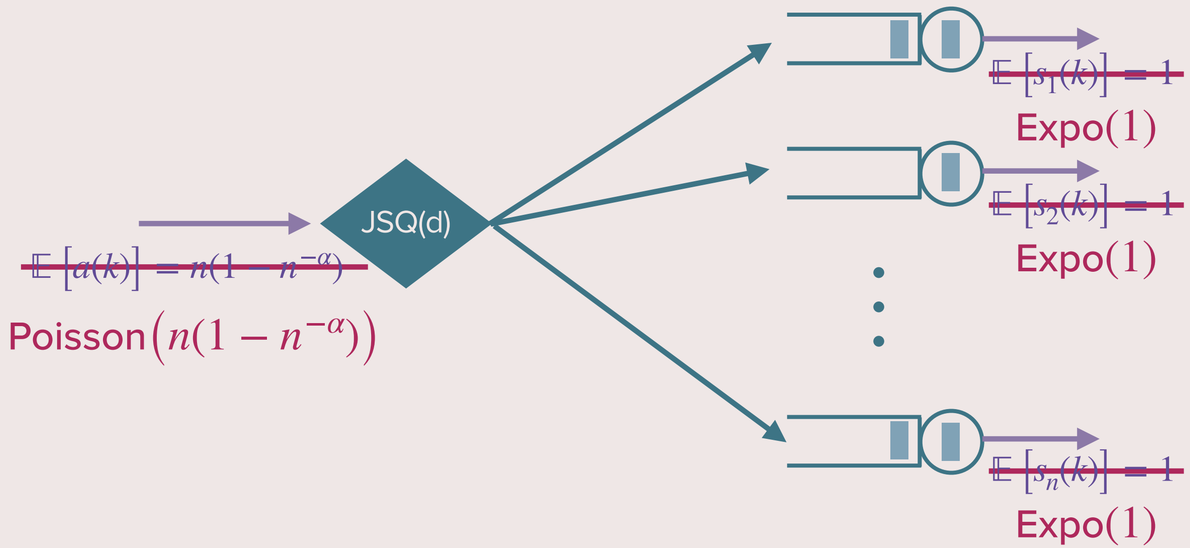
- Convergence to CRP
- Can we get faster convergence?



Why can't we get $\alpha > 1$?

Bottleneck in the proof:

- Convergence to CRP
- Can we get faster convergence?

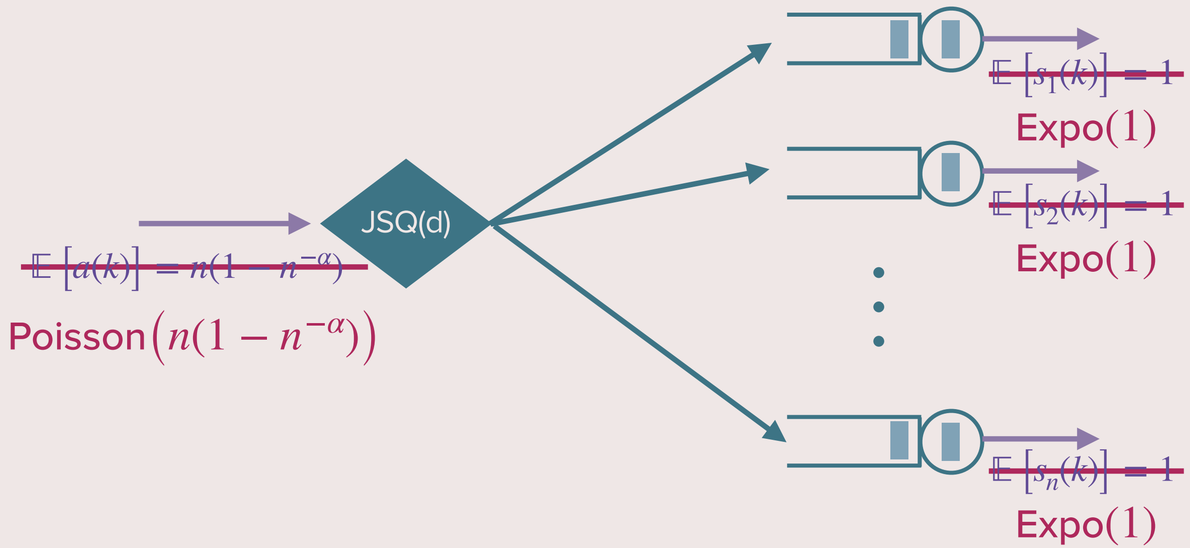


Back to discrete time:

Why can't we get $\alpha > 1$?

Bottleneck in the proof:

- Convergence to CRP
- Can we get faster convergence?



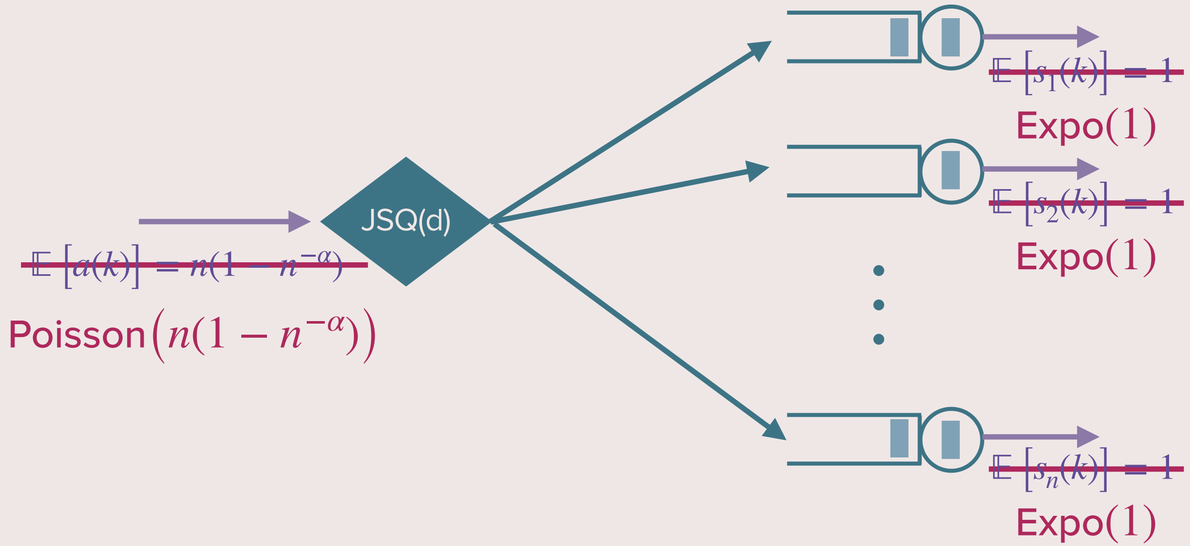
Back to discrete time:

- Current method: $\alpha > 11/2$

Why can't we get $\alpha > 1$?

Bottleneck in the proof:

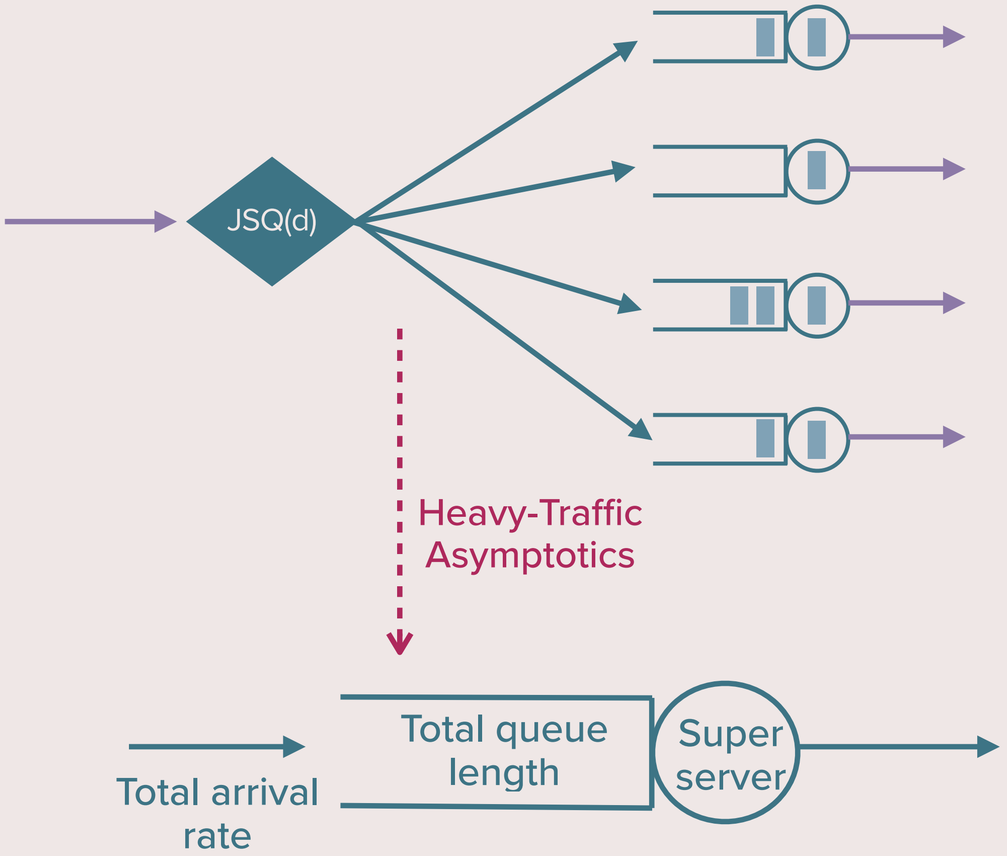
- Convergence to CRP
- Can we get faster convergence?



Back to discrete time:

- Current method: $\alpha > 11/2$
- What happens if we route one by one?
  - Can we reproduce continuous-time results?
  - Can we improve them?

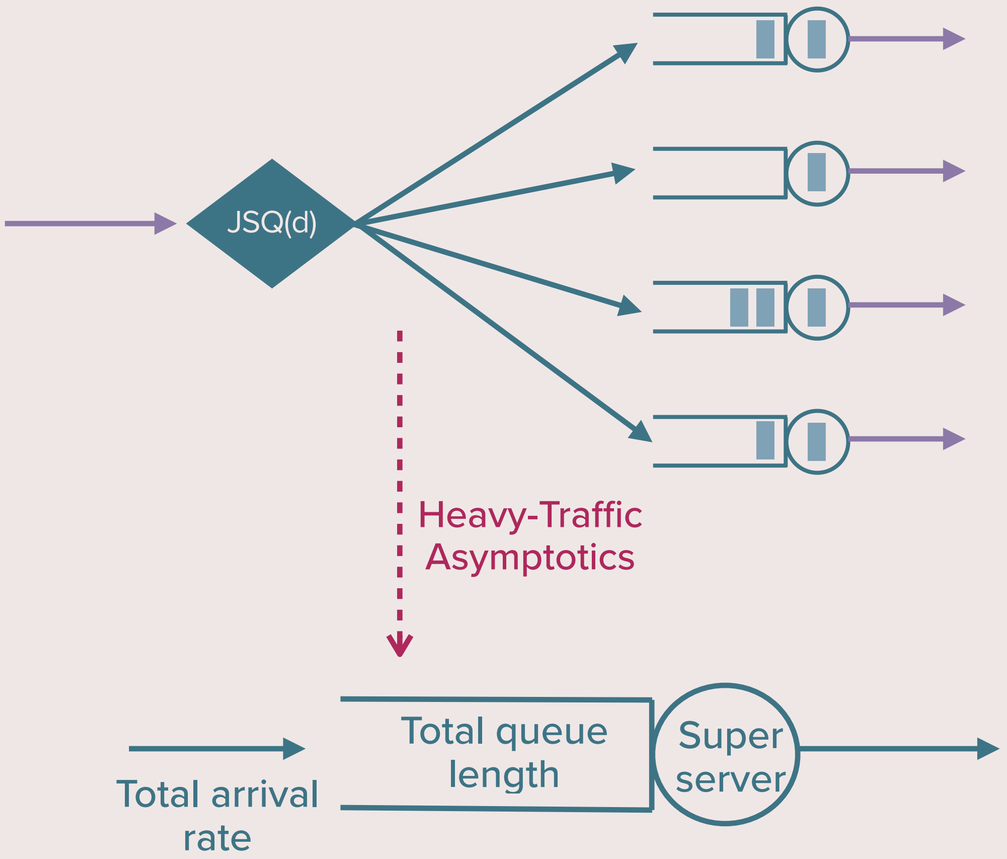
Conclusion and Future Work



Conclusion and Future Work

Classical Heavy-Traffic Regime:

- CRP condition = Optimality
- MGF method: Compute the distribution
  - Simple, flexible
  - More than just heavy-traffic results
  - What to do in non-CRP systems?



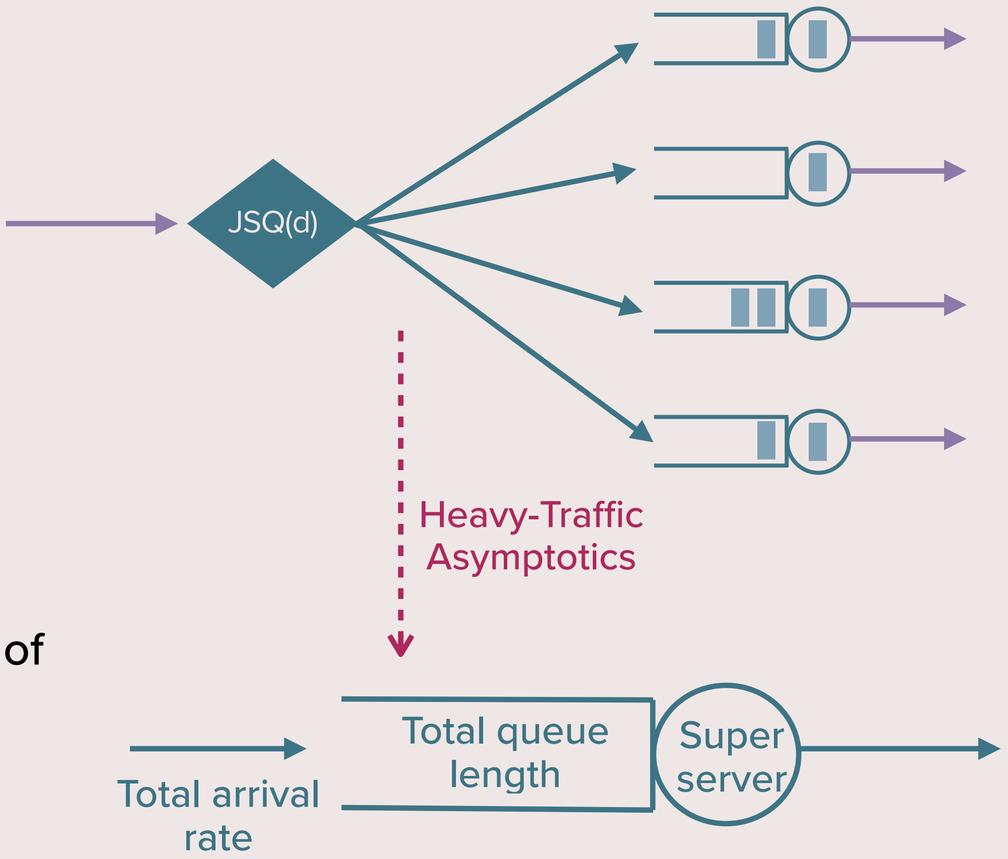
Conclusion and Future Work

Classical Heavy-Traffic Regime:

- CRP condition = Optimality
- MGF method: Compute the distribution
  - Simple, flexible
  - More than just heavy-traffic results
  - What to do in non-CRP systems?

Many-server heavy-traffic regime:

- How fast should the load grow with respect to the number of servers to observe classical heavy traffic?
- If $d = cn^{\beta}$, we need $\alpha + \beta > 3$
  - JSQ: $\alpha > 2$
  - What happens for $\alpha \in (1,2]$?
  - Stronger convergence to CRP? Another distribution?



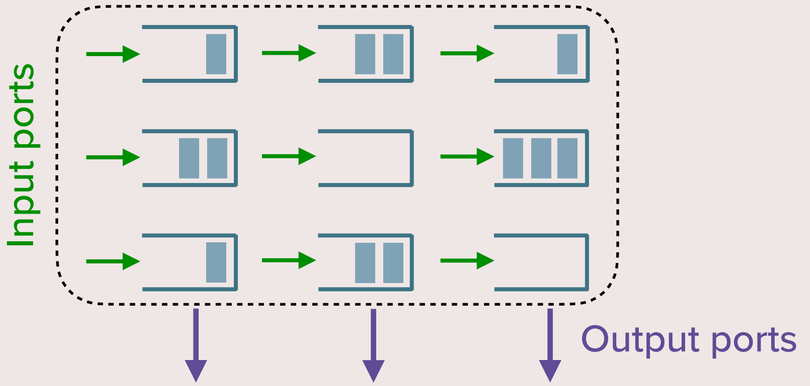
Minimizing Delay in Supermarket-Checkout Systems

Daniela Hurtado-Lange
William and Mary
Joint work with Siva Theja Maguluri

Mathematics Colloquium, April 1st, 2022

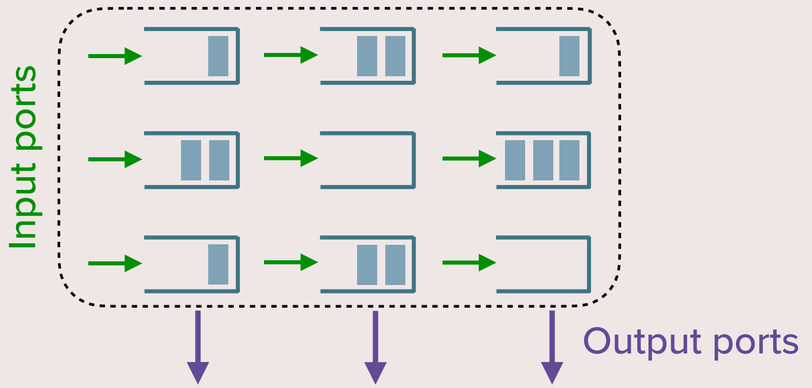
Contact information: dahurtadolange@wm.edu

The Input-Queued Switch Model



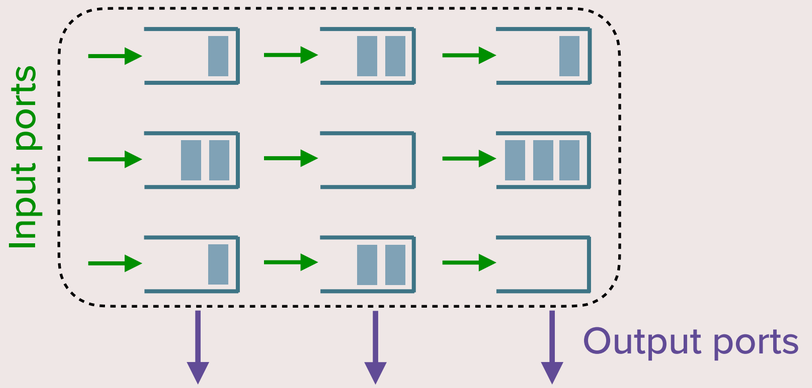
The Input-Queued Switch Model

- Discrete time model



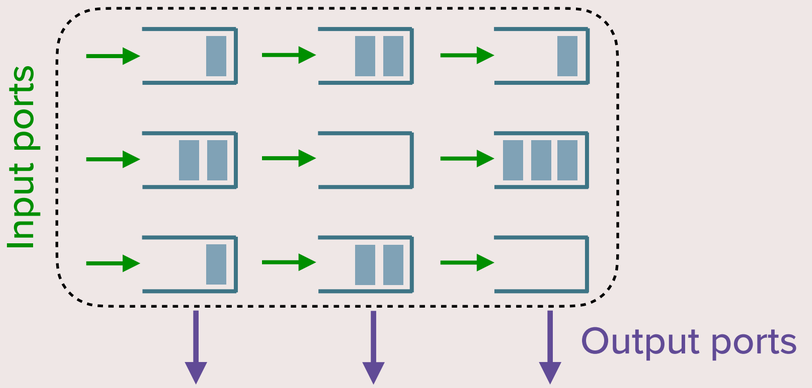
The Input-Queued Switch Model

- Discrete time model
- n input ports and n output ports
  - Jobs arrive at input ports, and go to the desired output port
  - Jobs are processed at output ports and take exactly one time slot



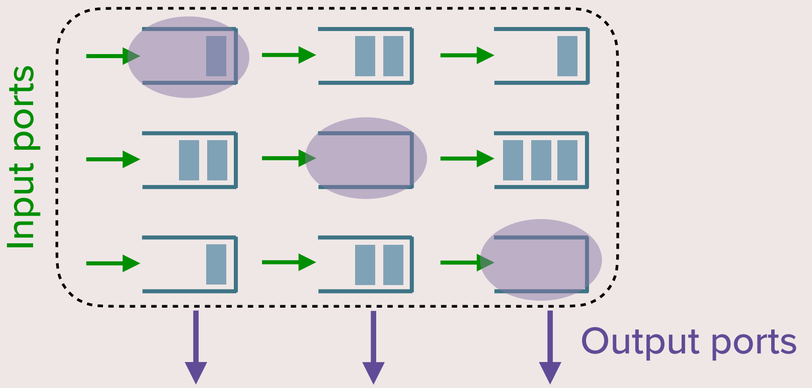
The Input-Queued Switch Model

- Discrete time model
- n input ports and n output ports
  - Jobs arrive at input ports, and go to the desired output port
  - Jobs are processed at output ports and take exactly one time slot
- **Constraint:** At most **one** job can be processed from each input port and at each output port



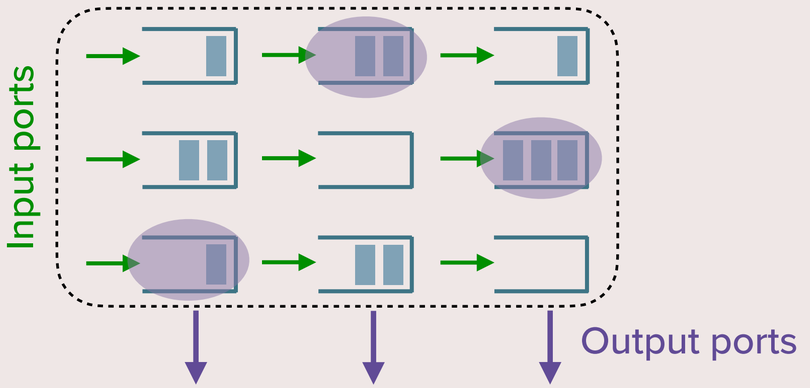
The Input-Queued Switch Model

- Discrete time model
- n input ports and n output ports
  - Jobs arrive at input ports, and go to the desired output port
  - Jobs are processed at output ports and take exactly one time slot
- **Constraint:** At most **one** job can be processed from each input port and at each output port



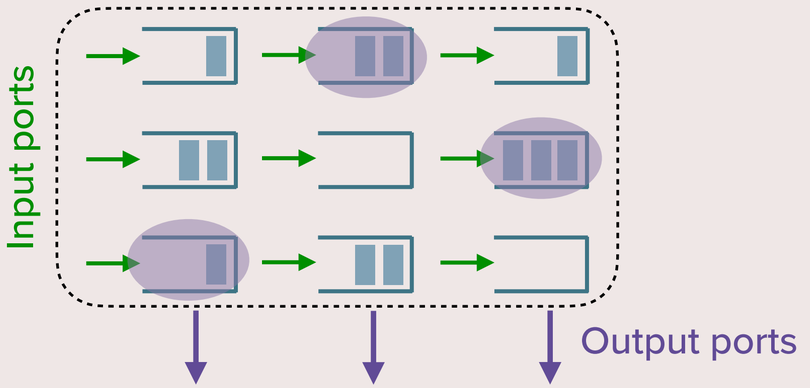
The Input-Queued Switch Model

- Discrete time model
- n input ports and n output ports
  - Jobs arrive at input ports, and go to the desired output port
  - Jobs are processed at output ports and take exactly one time slot
- **Constraint:** At most **one** job can be processed from each input port and at each output port



The Input-Queued Switch Model

- Discrete time model
- n input ports and n output ports
  - Jobs arrive at input ports, and go to the desired output port
  - Jobs are processed at output ports and take exactly one time slot
- **Constraint:** At most **one** job can be processed from each input port and at each output port
- **Scheduling:**
  - **Which queues to serve?** Largest total queue length

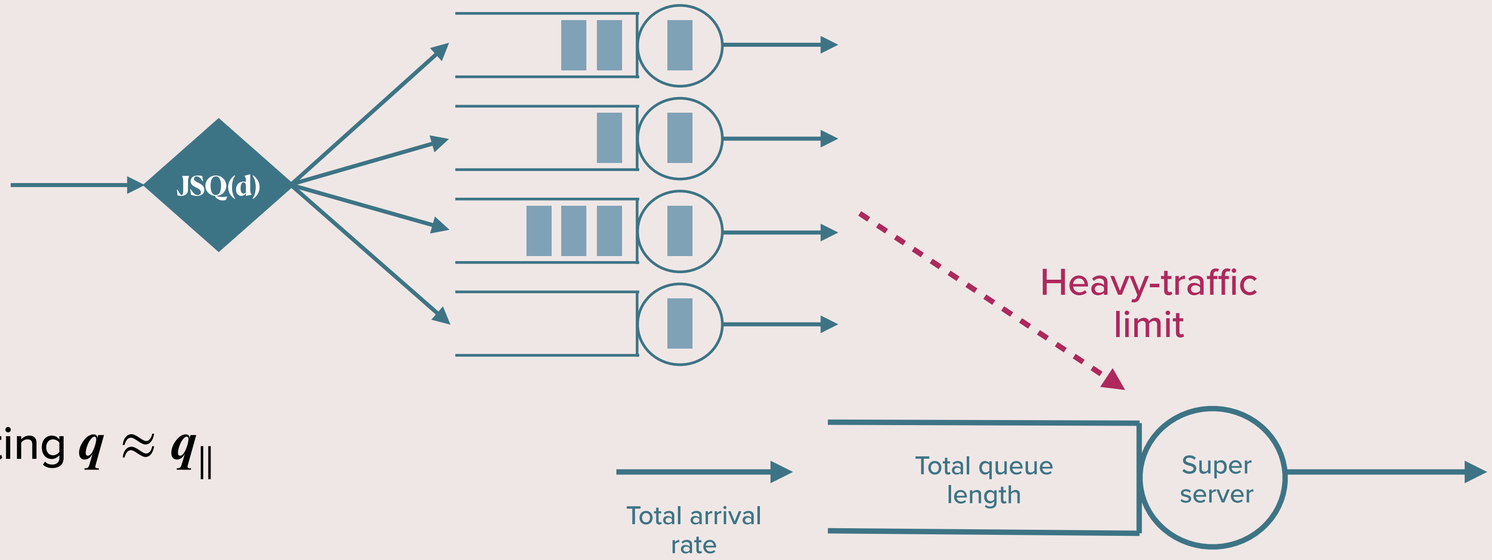


Formal Definition of CRP

Define:

$$\boldsymbol{q}_{\|} \triangleq \mathbf{1}\left(\frac{\sum q_i}{n}\right)$$

$\boldsymbol{q}_{\perp} \triangleq \boldsymbol{q} - \boldsymbol{q}_{\|}$ = Error of approximating $\boldsymbol{q} \approx \boldsymbol{q}_{\|}$

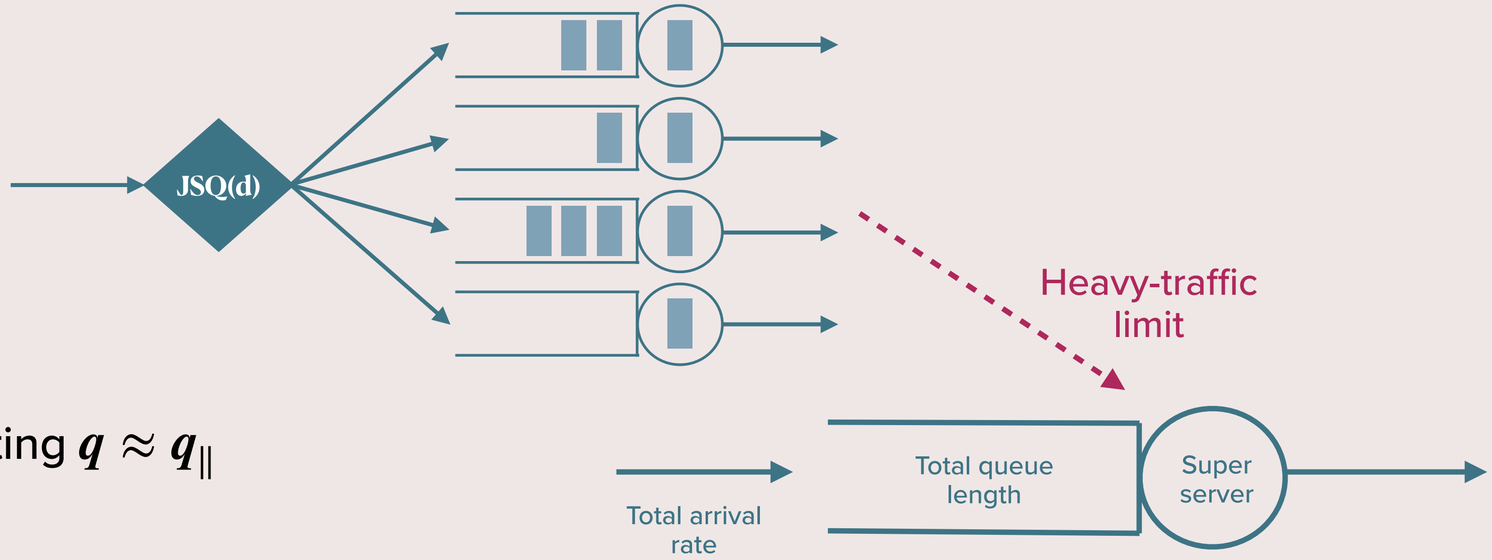


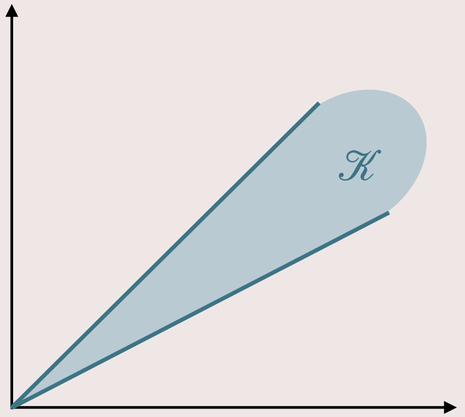
Formal Definition of CRP

Define:

$$\boldsymbol{q}_{\|} \triangleq \mathbf{1}\left(\frac{\sum q_i}{n}\right)$$

$\boldsymbol{q}_{\perp} \triangleq \boldsymbol{q} - \boldsymbol{q}_{\|}$ = Error of approximating $\boldsymbol{q} \approx \boldsymbol{q}_{\|}$



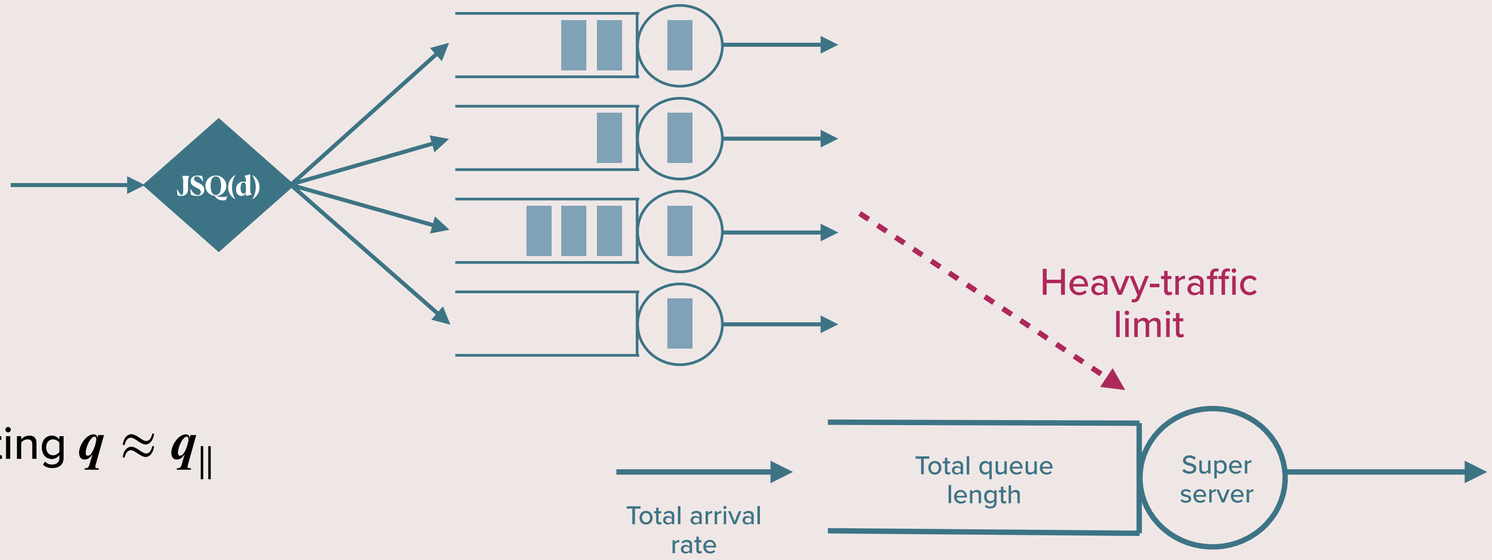


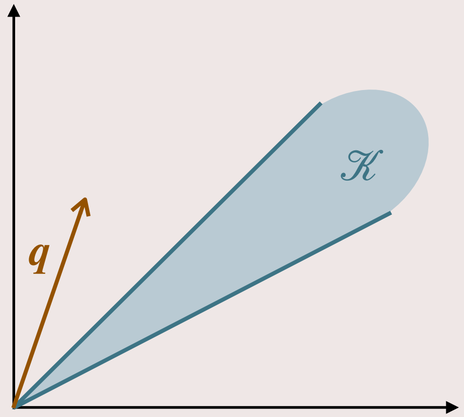
Formal Definition of CRP

Define:

$$\boldsymbol{q}_{\|} \triangleq \mathbf{1}\left(\frac{\sum q_i}{n}\right)$$

$\boldsymbol{q}_{\perp} \triangleq \boldsymbol{q} - \boldsymbol{q}_{\|}$ = Error of approximating $\boldsymbol{q} \approx \boldsymbol{q}_{\|}$



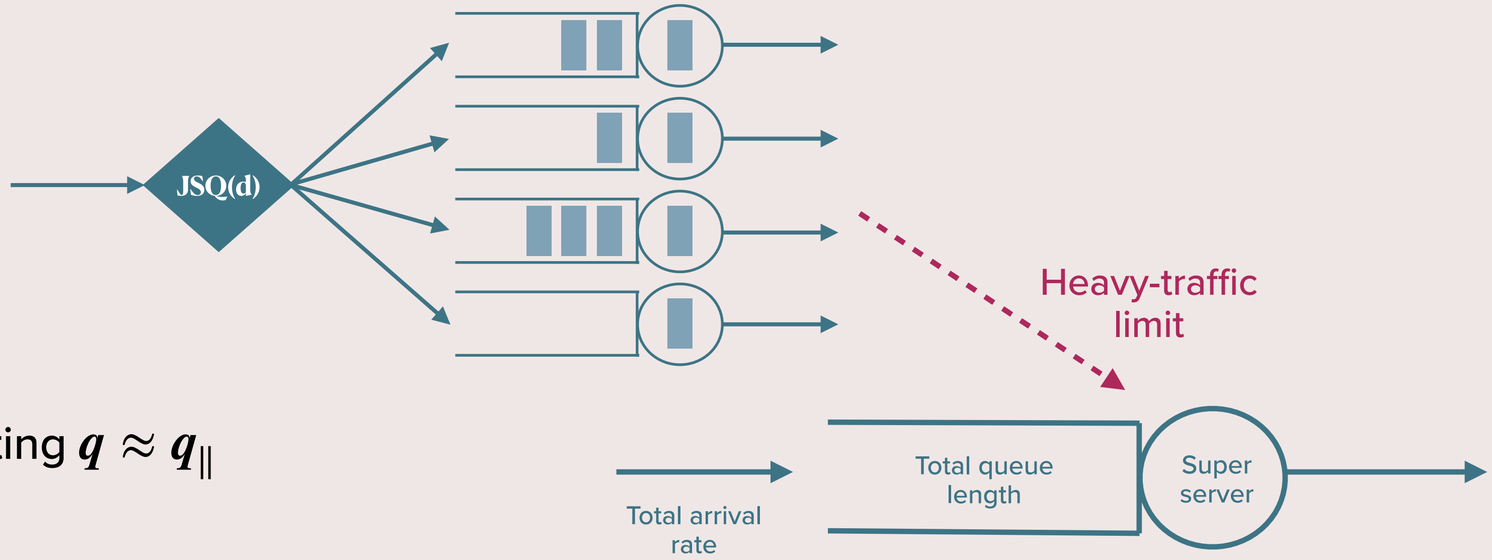


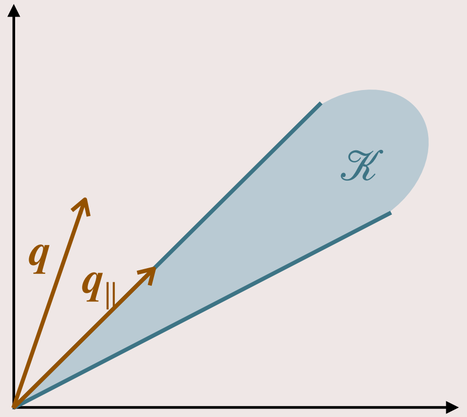
Formal Definition of CRP

Define:

$$\boldsymbol{q}_{\|} \triangleq \mathbf{1}\left(\frac{\sum q_i}{n}\right)$$

$\boldsymbol{q}_{\perp} \triangleq \boldsymbol{q} - \boldsymbol{q}_{\|}$ = Error of approximating $\boldsymbol{q} \approx \boldsymbol{q}_{\|}$



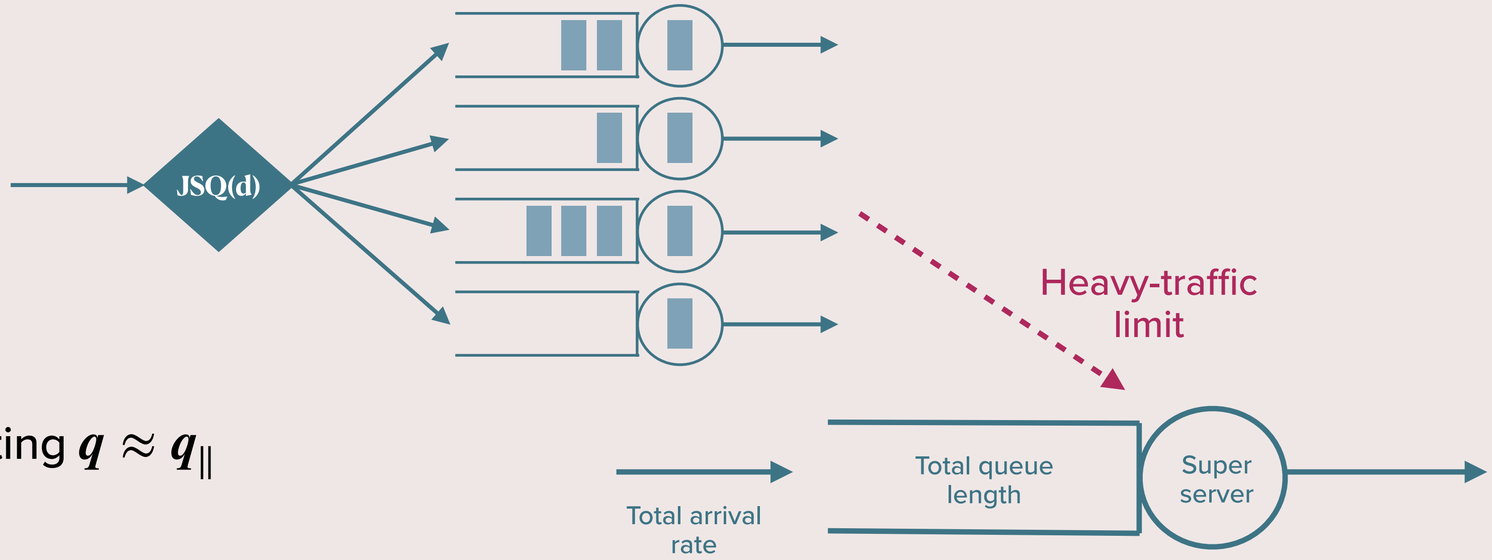


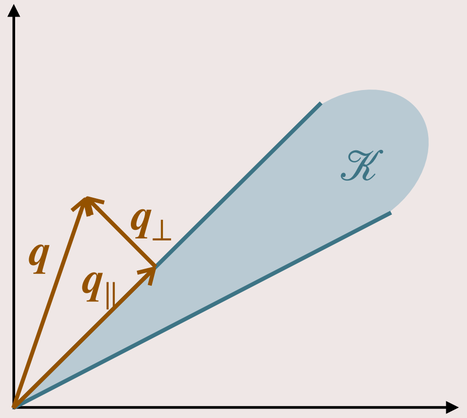
Formal Definition of CRP

Define:

$$\boldsymbol{q}_{\|} \triangleq \mathbf{1}\left(\frac{\sum q_i}{n}\right)$$

$\boldsymbol{q}_{\perp} \triangleq \boldsymbol{q} - \boldsymbol{q}_{\|}$ = Error of approximating $\boldsymbol{q} \approx \boldsymbol{q}_{\|}$



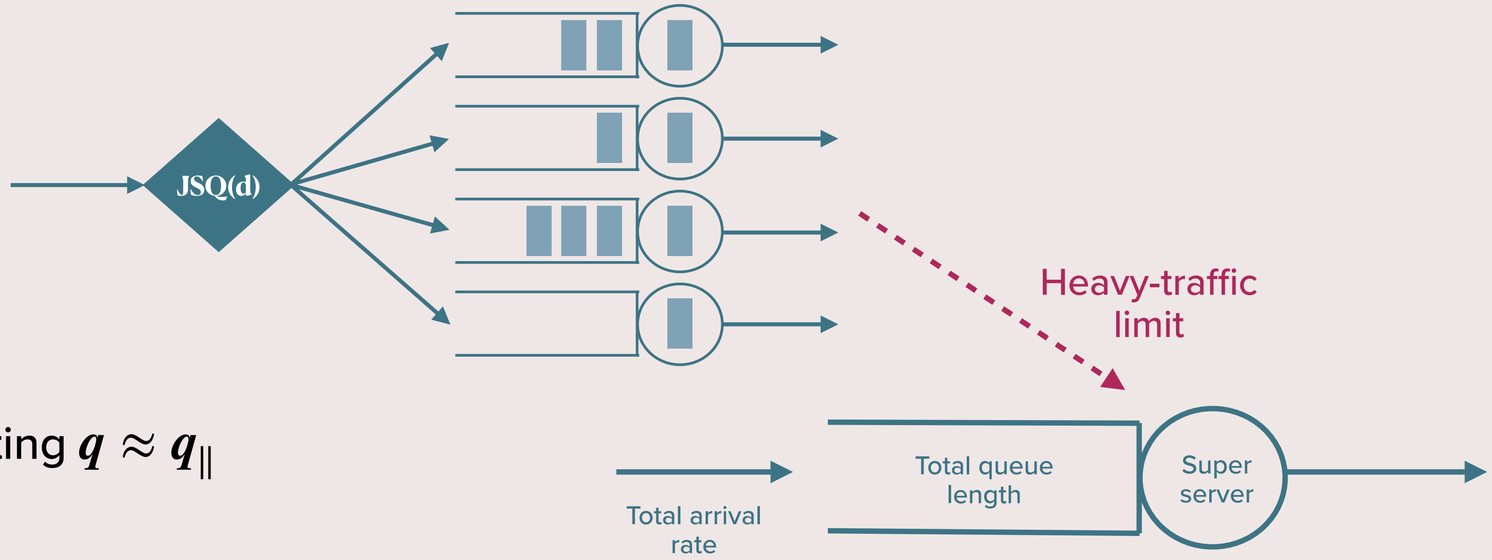


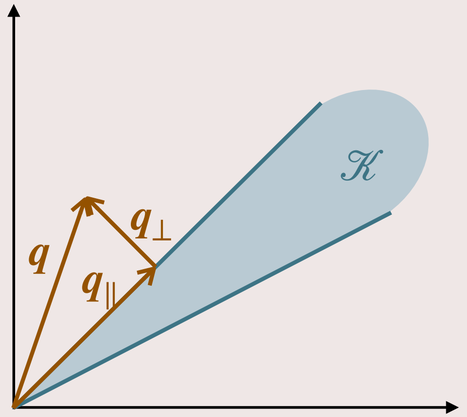
Formal Definition of CRP

Define:

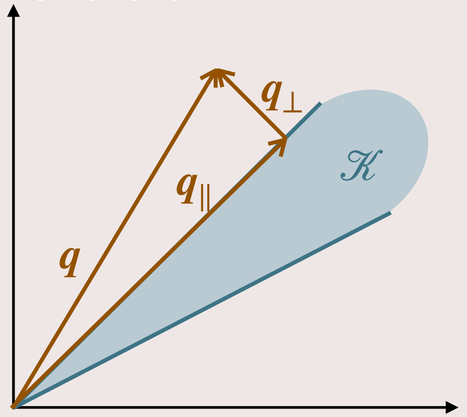
$$\boldsymbol{q}_{\|} \triangleq \mathbf{1}\left(\frac{\sum q_i}{n}\right)$$

$\boldsymbol{q}_{\perp} \triangleq \boldsymbol{q} - \boldsymbol{q}_{\|}$ = Error of approximating $\boldsymbol{q} \approx \boldsymbol{q}_{\|}$





Smaller ϵ

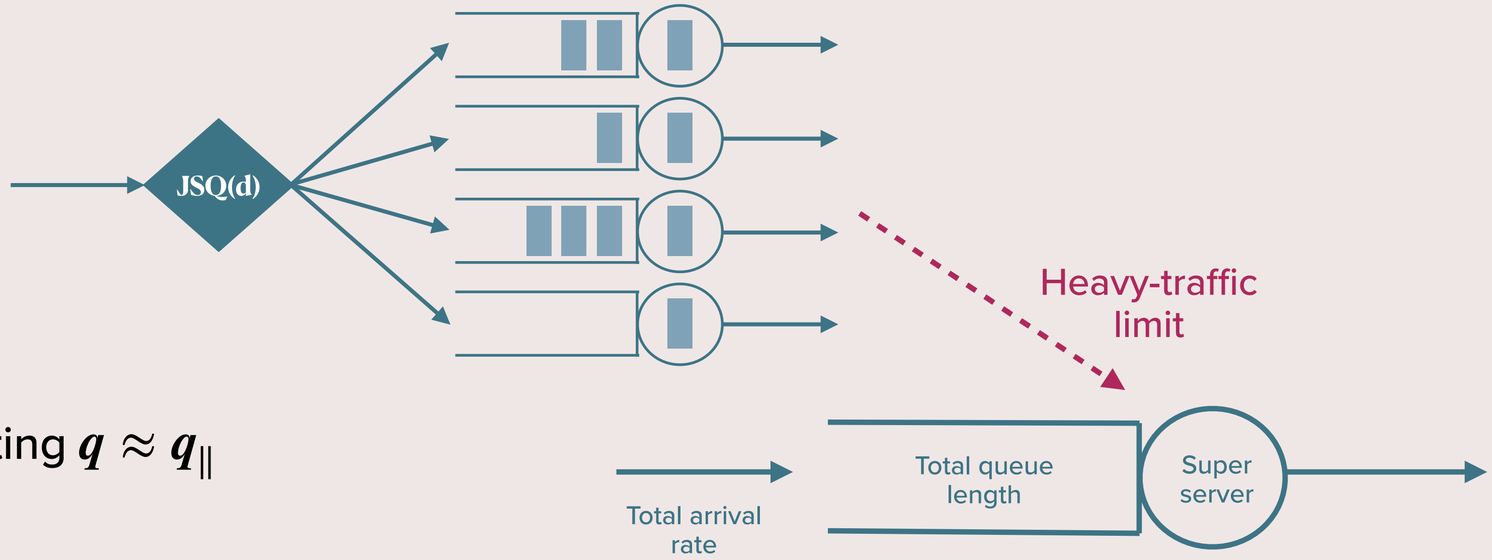


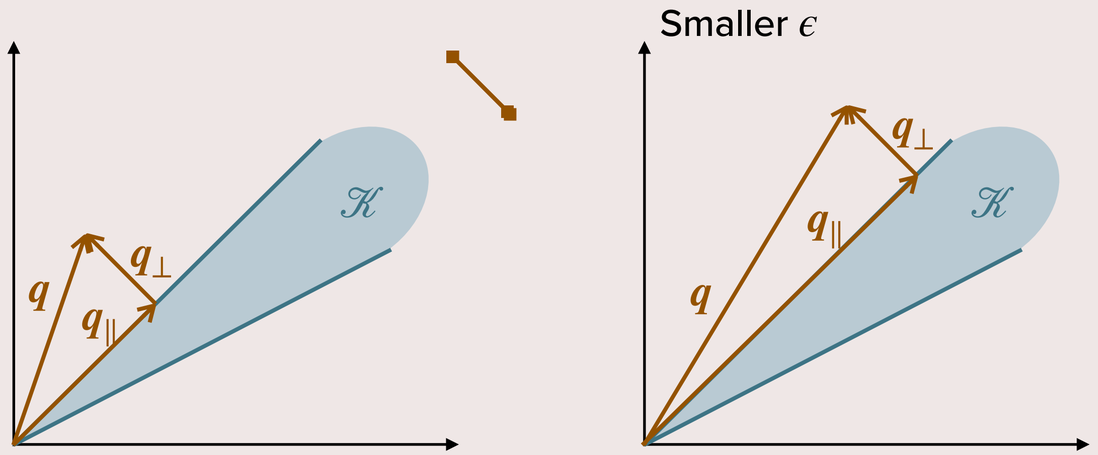
Formal Definition of CRP

Define:

$$\boldsymbol{q}_{\|} \triangleq \mathbf{1}\left(\frac{\sum q_i}{n}\right)$$

$\boldsymbol{q}_{\perp} \triangleq \boldsymbol{q} - \boldsymbol{q}_{\|}$ = Error of approximating $\boldsymbol{q} \approx \boldsymbol{q}_{\|}$



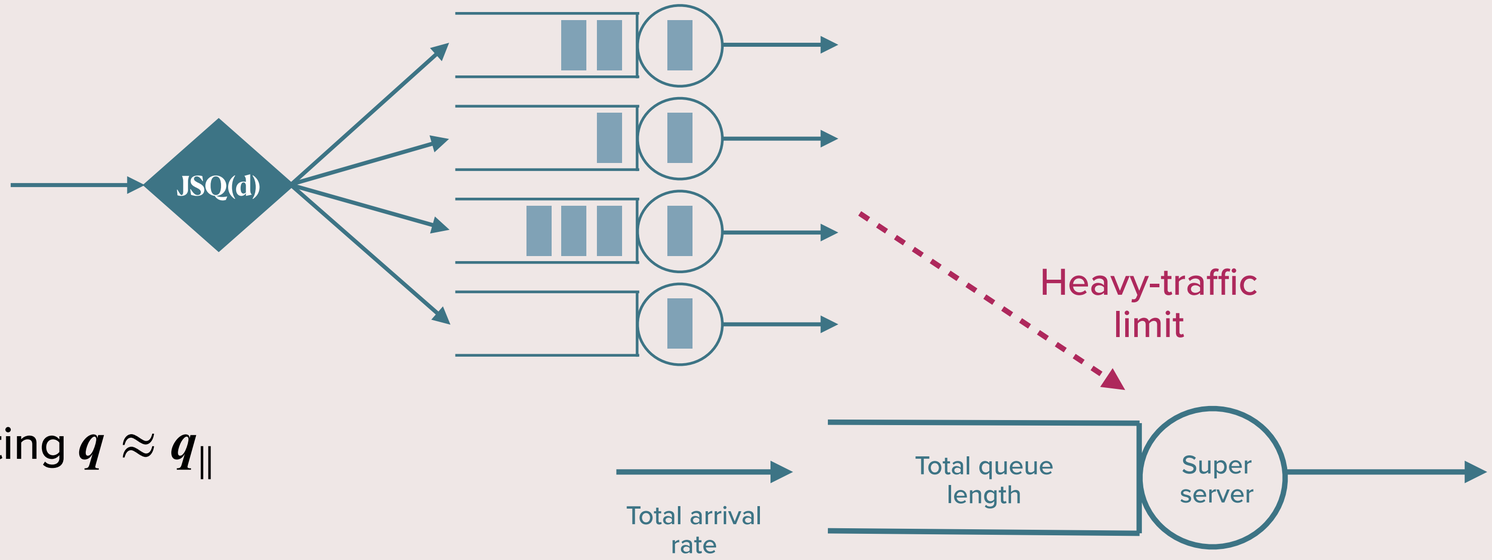


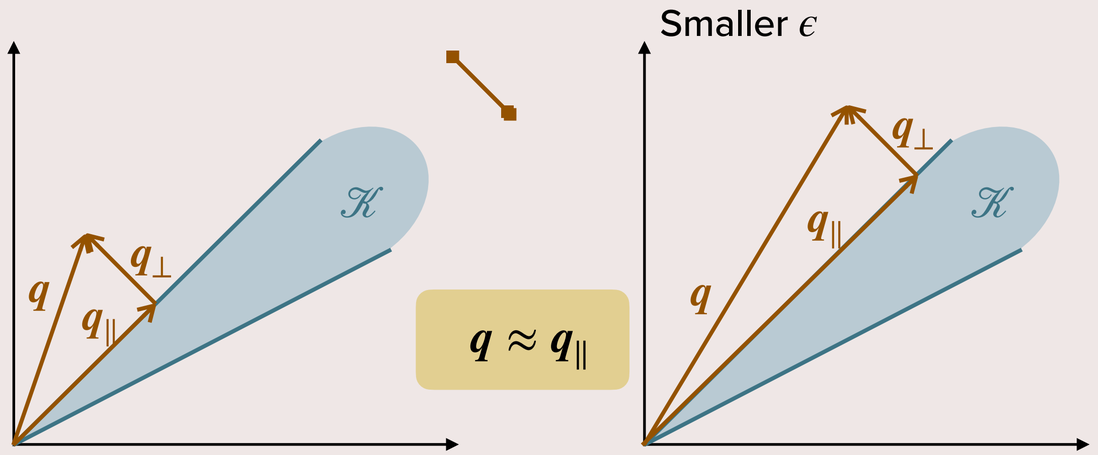
Formal Definition of CRP

Define:

$$\boldsymbol{q}_{\|} \triangleq \mathbf{1}\left(\frac{\sum q_i}{n}\right)$$

$\boldsymbol{q}_{\perp} \triangleq \boldsymbol{q} - \boldsymbol{q}_{\|}$ = Error of approximating $\boldsymbol{q} \approx \boldsymbol{q}_{\|}$



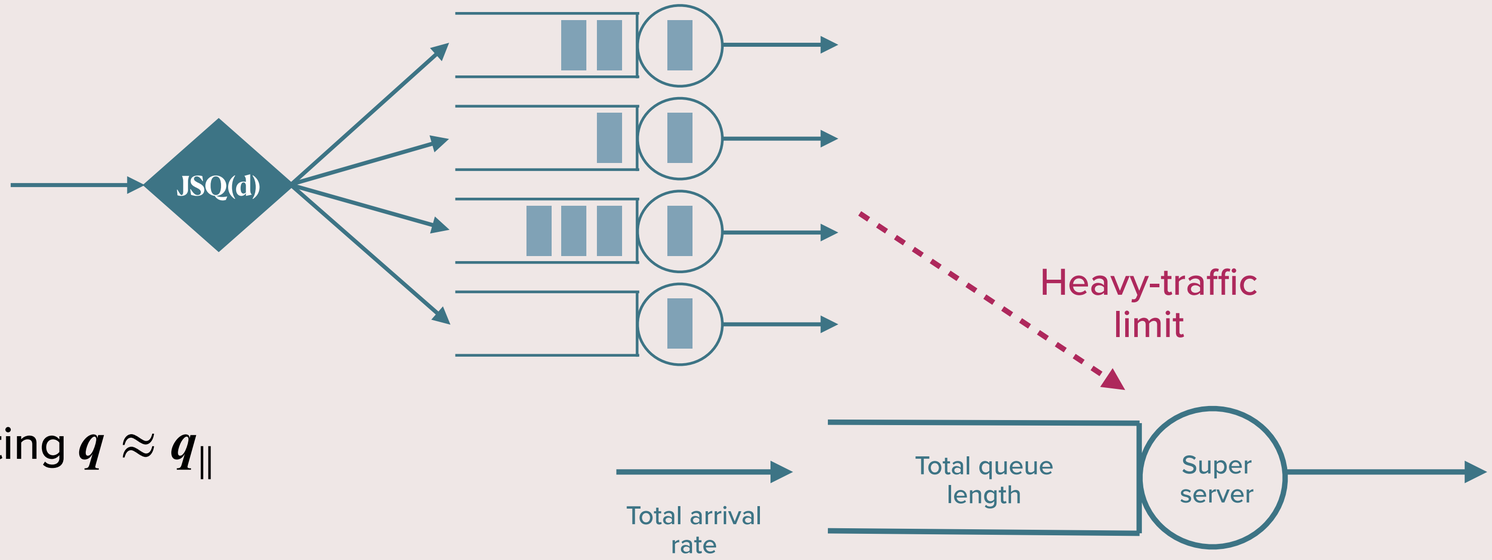


Formal Definition of CRP

Define:

$$\boldsymbol{q}_{\|} \triangleq \mathbf{1}\left(\frac{\sum q_i}{n}\right)$$

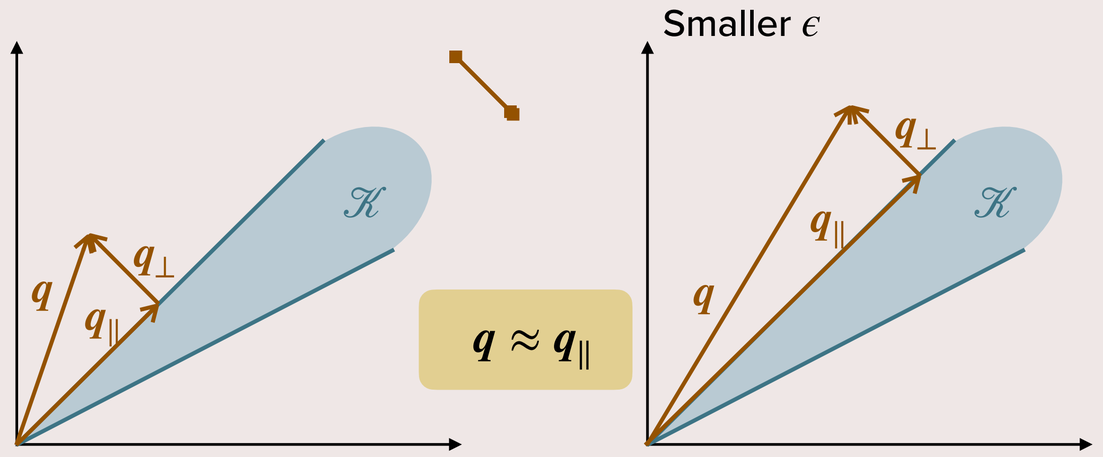
$\boldsymbol{q}_{\perp} \triangleq \boldsymbol{q} - \boldsymbol{q}_{\|}$ = Error of approximating $\boldsymbol{q} \approx \boldsymbol{q}_{\|}$



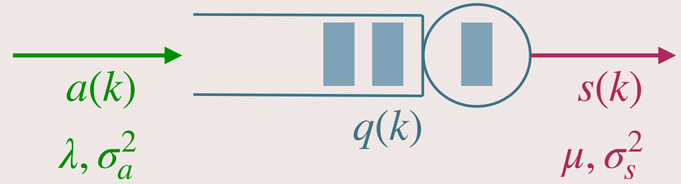
CRP:

$\boldsymbol{q}_{\perp}$ is negligible in heavy traffic

Formally, $\mathbb{E}\left[\|\boldsymbol{q}_{\perp}\|^m\right] \leq C_m$ for all $m = 1,2,3\ldots$



Single-Server Queue and Drift Method



Dynamics of the queues:

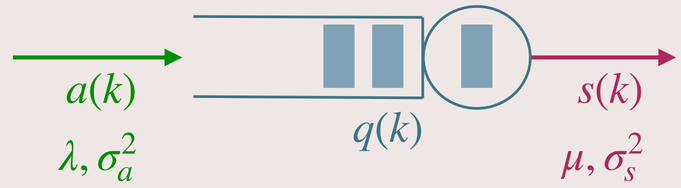
$$q(k+1) = q(k) + a(k) - s(k) + u(k)$$

$$\implies q(k+1)u(k) = 0$$

Single-Server Queue and Drift Method

- In steady-state ($k \to \infty$):

$$\mathbb{E}\left[q^2(k+1)\right] = \mathbb{E}\left[q^2(k)\right]$$



Dynamics of the queues:

$$q(k+1) = q(k) + a(k) - s(k) + u(k)$$

$$\implies q(k+1)u(k) = 0$$

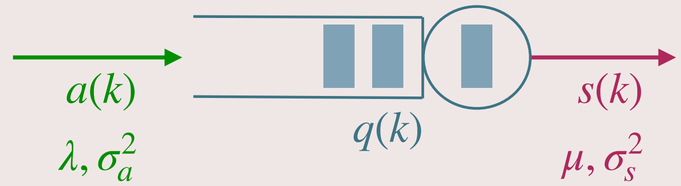
Single-Server Queue and Drift Method

- In steady-state ($k \to \infty$):

$$\mathbb{E}\left[q^2(k+1)\right] = \mathbb{E}\left[q^2(k)\right]$$

- Yields:

$$\mathbb{E}[q] = \frac{\mathbb{E}\left[(a-s)^2\right]}{2\epsilon} - \frac{\mathbb{E}\left[u^2\right]}{2\epsilon}$$



Dynamics of the queues:

$$q(k+1) = q(k) + a(k) - s(k) + u(k)$$

$$\implies q(k+1)u(k) = 0$$

Single-Server Queue and Drift Method

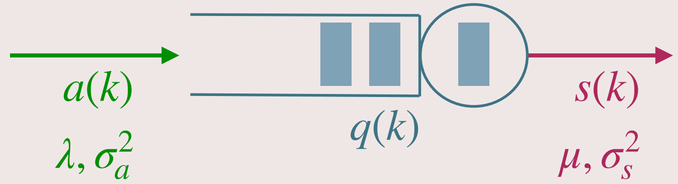
- In steady-state ($k \to \infty$):

$$\mathbb{E}\left[q^2(k+1)\right] = \mathbb{E}\left[q^2(k)\right]$$

- Yields:

$$\mathbb{E}[q] = \frac{\mathbb{E}\left[(a-s)^2\right]}{2\epsilon} - \frac{\mathbb{E}\left[u^2\right]}{2\epsilon}$$

Small compared to the first term



Dynamics of the queues:

$$q(k+1) = q(k) + a(k) - s(k) + u(k)$$

$$\implies q(k+1)u(k) = 0$$

Single-Server Queue and Drift Method

- In steady-state ($k \to \infty$):

$$\mathbb{E}\left[q^2(k+1)\right] = \mathbb{E}\left[q^2(k)\right]$$

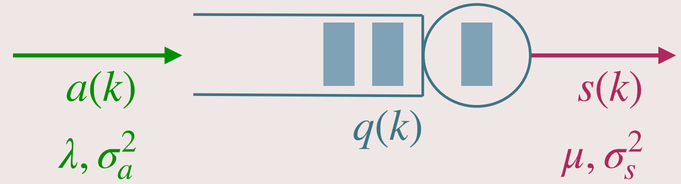
- Yields:

$$\mathbb{E}[q] = \frac{\mathbb{E}\left[(a-s)^2\right]}{2\epsilon} - \frac{\mathbb{E}\left[u^2\right]}{2\epsilon}$$

Small compared to the first term

- Therefore,

$$\lim_{\epsilon \downarrow 0} \mathbb{E}\left[\epsilon q\right] = \frac{\sigma_a^2 + \sigma_s^2}{2}$$



Dynamics of the queues:

$$q(k+1) = q(k) + a(k) - s(k) + u(k)$$

$$\implies q(k+1)u(k) = 0$$

Single-Server Queue and Drift Method

- In steady-state ($k \to \infty$):

$$\mathbb{E}\left[q^2(k+1)\right] = \mathbb{E}\left[q^2(k)\right]$$

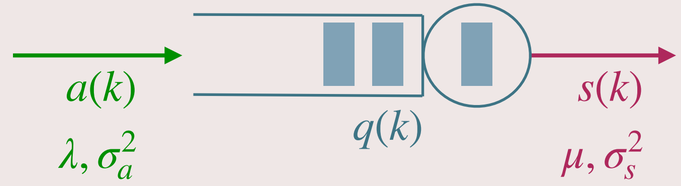
- Yields:

$$\mathbb{E}[q] = \frac{\mathbb{E}\left[(a-s)^2\right]}{2\epsilon} - \frac{\mathbb{E}\left[u^2\right]}{2\epsilon}$$

Small compared to the first term

- Therefore,

$$\lim_{\epsilon \downarrow 0} \mathbb{E}\left[\epsilon q\right] = \frac{\sigma_a^2 + \sigma_s^2}{2}$$



Dynamics of the queues:

$$q(k+1) = q(k) + a(k) - s(k) + u(k)$$

$$\implies q(k+1)u(k) = 0$$

Drift Method:

Test function:

Obtain:

Single-Server Queue and Drift Method

- In steady-state ($k \to \infty$):

$$\mathbb{E}\left[q^2(k+1)\right] = \mathbb{E}\left[q^2(k)\right]$$

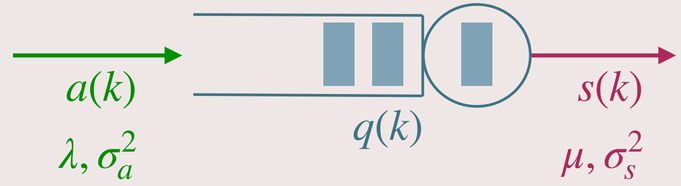
- Yields:

$$\mathbb{E}[q] = \frac{\mathbb{E}\left[(a-s)^2\right]}{2\epsilon} - \frac{\mathbb{E}\left[u^2\right]}{2\epsilon}$$

Small compared to the first term

- Therefore,

$$\lim_{\epsilon \downarrow 0} \mathbb{E}\left[\epsilon q\right] = \frac{\sigma_a^2 + \sigma_s^2}{2}$$



Dynamics of the queues:

$$q(k+1) = q(k) + a(k) - s(k) + u(k)$$

$$\implies q(k+1)u(k) = 0$$

Drift Method:

Test function: q^2

Obtain:

Single-Server Queue and Drift Method

- In steady-state ($k \to \infty$):

$$\mathbb{E}\left[q^2(k+1)\right] = \mathbb{E}\left[q^2(k)\right]$$

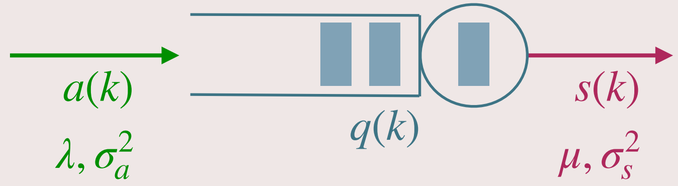
- Yields:

$$\mathbb{E}[q] = \frac{\mathbb{E}\left[(a-s)^2\right]}{2\epsilon} - \frac{\mathbb{E}\left[u^2\right]}{2\epsilon}$$

Small compared to the first term

- Therefore,

$$\lim_{\epsilon \downarrow 0} \mathbb{E}\left[\epsilon q\right] = \frac{\sigma_a^2 + \sigma_s^2}{2}$$



Dynamics of the queues:

$$q(k+1) = q(k) + a(k) - s(k) + u(k)$$

$$\implies q(k+1)u(k) = 0$$

Drift Method:

Test function: q^2

Obtain: $\mathbb{E}\left[\epsilon q\right]$

Single-Server Queue and Drift Method

- In steady-state ($k \to \infty$):

$$\mathbb{E}\left[q^2(k+1)\right] = \mathbb{E}\left[q^2(k)\right]$$

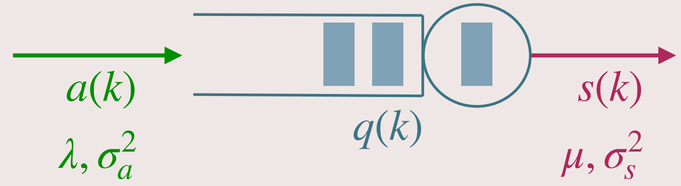
- Yields:

$$\mathbb{E}[q] = \frac{\mathbb{E}\left[(a-s)^2\right]}{2\epsilon} - \frac{\mathbb{E}\left[u^2\right]}{2\epsilon}$$

Small compared to the first term

- Therefore,

$$\lim_{\epsilon \downarrow 0} \mathbb{E}\left[\epsilon q\right] = \frac{\sigma_a^2 + \sigma_s^2}{2}$$



Dynamics of the queues:

$$q(k+1) = q(k) + a(k) - s(k) + u(k)$$

$$\implies q(k+1)u(k) = 0$$

Drift Method:

Test function: q^2

Obtain: $\mathbb{E}\left[\epsilon q\right]$

$$\lim_{\epsilon \downarrow 0} (\quad) = \frac{\sigma_a^2 + \sigma_s^2}{2}$$

Single-Server Queue and Drift Method

- In steady-state ($k \to \infty$):

$$\mathbb{E}\left[q^2(k+1)\right] = \mathbb{E}\left[q^2(k)\right]$$

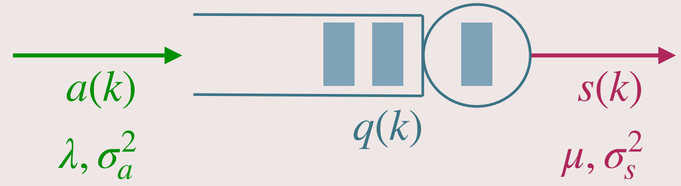
- Yields:

$$\mathbb{E}[q] = \frac{\mathbb{E}\left[(a-s)^2\right]}{2\epsilon} - \frac{\mathbb{E}\left[u^2\right]}{2\epsilon}$$

Small compared to the first term

- Therefore,

$$\lim_{\epsilon \downarrow 0} \mathbb{E}\left[\epsilon q\right] = \frac{\sigma_a^2 + \sigma_s^2}{2}$$



Dynamics of the queues:

$$q(k+1) = q(k) + a(k) - s(k) + u(k)$$

$$\implies q(k+1)u(k) = 0$$

Drift Method:

Test function: q^2 q^3

Obtain: $\mathbb{E}\left[\epsilon q\right]$

$$\lim_{\epsilon \downarrow 0} (\quad) = \frac{\sigma_a^2 + \sigma_s^2}{2}$$

Single-Server Queue and Drift Method

- In steady-state ($k \to \infty$):

$$\mathbb{E}\left[q^2(k+1)\right] = \mathbb{E}\left[q^2(k)\right]$$

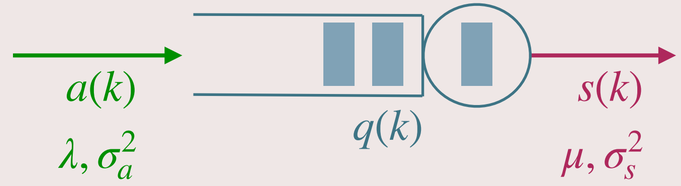
- Yields:

$$\mathbb{E}[q] = \frac{\mathbb{E}\left[(a-s)^2\right]}{2\epsilon} - \frac{\mathbb{E}\left[u^2\right]}{2\epsilon}$$

Small compared to the first term

- Therefore,

$$\lim_{\epsilon \downarrow 0} \mathbb{E}\left[\epsilon q\right] = \frac{\sigma_a^2 + \sigma_s^2}{2}$$



Dynamics of the queues:

$$q(k+1) = q(k) + a(k) - s(k) + u(k)$$

$$\implies q(k+1)u(k) = 0$$

Drift Method:

Test function: q^2, q^3

Obtain: $\mathbb{E}\left[\epsilon q\right]$, $\mathbb{E}\left[\epsilon^2 q^2\right]$

$$\lim_{\epsilon \downarrow 0} (\quad) = \frac{\sigma_a^2 + \sigma_s^2}{2}$$

Single-Server Queue and Drift Method

- In steady-state ($k \to \infty$):

$$\mathbb{E}\left[q^2(k+1)\right] = \mathbb{E}\left[q^2(k)\right]$$

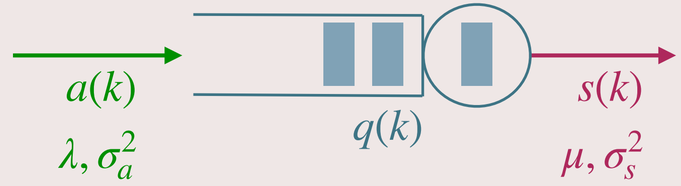
- Yields:

$$\mathbb{E}[q] = \frac{\mathbb{E}\left[(a-s)^2\right]}{2\epsilon} - \frac{\mathbb{E}\left[u^2\right]}{2\epsilon}$$

Small compared to the first term

- Therefore,

$$\lim_{\epsilon \downarrow 0} \mathbb{E}\left[\epsilon q\right] = \frac{\sigma_a^2 + \sigma_s^2}{2}$$



Dynamics of the queues:

$$q(k+1) = q(k) + a(k) - s(k) + u(k)$$
$$\implies q(k+1)u(k) = 0$$

Drift Method:

Test function: q^2, q^3

Obtain: $\mathbb{E}\left[\epsilon q\right]$, $\mathbb{E}\left[\epsilon^2 q^2\right]$

$$\lim_{\epsilon \downarrow 0} (\quad) = \frac{\sigma_a^2 + \sigma_s^2}{2} \qquad 2\left(\frac{\sigma_a^2 + \sigma_s^2}{2}\right)^2$$

Single-Server Queue and Drift Method

- In steady-state ($k \to \infty$):

$$\mathbb{E}\left[q^2(k+1)\right] = \mathbb{E}\left[q^2(k)\right]$$

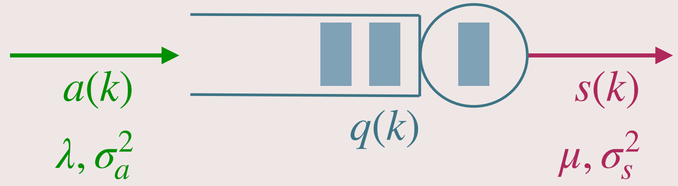
- Yields:

$$\mathbb{E}[q] = \frac{\mathbb{E}\left[(a-s)^2\right]}{2\epsilon} - \frac{\mathbb{E}\left[u^2\right]}{2\epsilon}$$

Small compared to the first term

- Therefore,

$$\lim_{\epsilon \downarrow 0} \mathbb{E}\left[\epsilon q\right] = \frac{\sigma_a^2 + \sigma_s^2}{2}$$



Dynamics of the queues:

$$q(k+1) = q(k) + a(k) - s(k) + u(k)$$
$$\implies q(k+1)u(k) = 0$$

Drift Method:

Test function: q^2, q^3, $\cdots$, q^m

Obtain:

$$\mathbb{E}\left[\epsilon q\right] \qquad \mathbb{E}\left[\epsilon^2 q^2\right]$$

$$\lim_{\epsilon \downarrow 0} (\quad) = \frac{\sigma_a^2 + \sigma_s^2}{2} \qquad 2\left(\frac{\sigma_a^2 + \sigma_s^2}{2}\right)^2$$

Single-Server Queue and Drift Method

- In steady-state ($k \to \infty$):

$$\mathbb{E}\left[q^2(k+1)\right] = \mathbb{E}\left[q^2(k)\right]$$

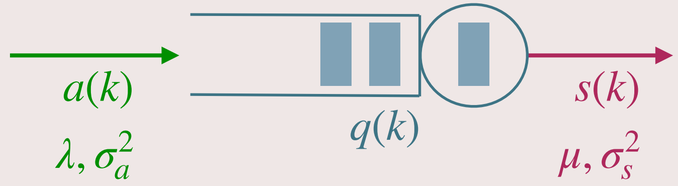
- Yields:

$$\mathbb{E}[q] = \frac{\mathbb{E}\left[(a-s)^2\right]}{2\epsilon} - \frac{\mathbb{E}\left[u^2\right]}{2\epsilon}$$

Small compared to the first term

- Therefore,

$$\lim_{\epsilon \downarrow 0} \mathbb{E}\left[\epsilon q\right] = \frac{\sigma_a^2 + \sigma_s^2}{2}$$



Dynamics of the queues:

$$q(k+1) = q(k) + a(k) - s(k) + u(k)$$
$$\implies q(k+1)u(k) = 0$$

Drift Method:

Test function: q^2, q^3, $\cdots$, q^m

Obtain: $\mathbb{E}\left[\epsilon q\right]$, $\mathbb{E}\left[\epsilon^2 q^2\right]$, $\cdots$, $\mathbb{E}\left[\epsilon^{m-1} q^{m-1}\right]$

$$\lim_{\epsilon \downarrow 0} (\quad) = \frac{\sigma_a^2 + \sigma_s^2}{2} \quad 2\left(\frac{\sigma_a^2 + \sigma_s^2}{2}\right)^2$$

Single-Server Queue and Drift Method

- In steady-state ($k \to \infty$):

$$\mathbb{E}\left[q^2(k+1)\right] = \mathbb{E}\left[q^2(k)\right]$$

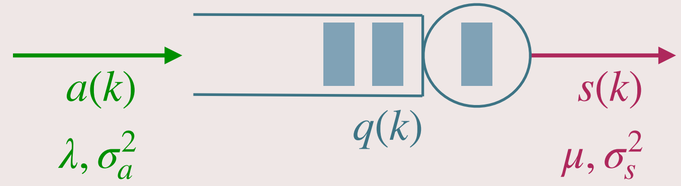
- Yields:

$$\mathbb{E}[q] = \frac{\mathbb{E}\left[(a-s)^2\right]}{2\epsilon} - \frac{\mathbb{E}\left[u^2\right]}{2\epsilon}$$

Small compared to the first term

- Therefore,

$$\lim_{\epsilon \downarrow 0} \mathbb{E}\left[\epsilon q\right] = \frac{\sigma_a^2 + \sigma_s^2}{2}$$



Dynamics of the queues:

$$q(k+1) = q(k) + a(k) - s(k) + u(k)$$
$$\implies q(k+1)u(k) = 0$$

Drift Method:

Test function: q^2, q^3, $\cdots$, q^m

Obtain: $\mathbb{E}\left[\epsilon q\right]$, $\mathbb{E}\left[\epsilon^2 q^2\right]$, $\cdots$, $\mathbb{E}\left[\epsilon^{m-1} q^{m-1}\right]$

$$\lim_{\epsilon \downarrow 0}(\quad) = \frac{\sigma_a^2 + \sigma_s^2}{2}, \quad 2\left(\frac{\sigma_a^2 + \sigma_s^2}{2}\right)^2, \quad \cdots, \quad (m-1)!\left(\frac{\sigma_a^2 + \sigma_s^2}{2}\right)^{m-1}$$

Single-Server Queue and Drift Method

- In steady-state ($k \to \infty$):

$$\mathbb{E}\left[q^2(k+1)\right] = \mathbb{E}\left[q^2(k)\right]$$

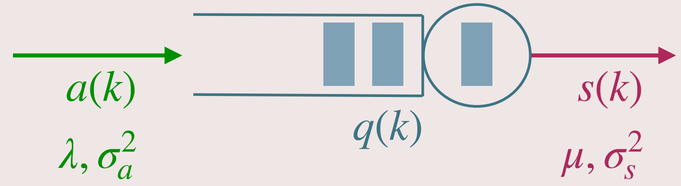
- Yields:

$$\mathbb{E}[q] = \frac{\mathbb{E}\left[(a-s)^2\right]}{2\epsilon} - \frac{\mathbb{E}\left[u^2\right]}{2\epsilon}$$

Small compared to the first term

- Therefore,

$$\lim_{\epsilon \downarrow 0} \mathbb{E}\left[\epsilon q\right] = \frac{\sigma_a^2 + \sigma_s^2}{2}$$



Dynamics of the queues:

$$q(k+1) = q(k) + a(k) - s(k) + u(k)$$
$$\implies q(k+1)u(k) = 0$$

Drift Method:

Test function: q^2 q^3 $\cdots$ q^m

Obtain: $\mathbb{E}\left[\epsilon q\right]$ $\mathbb{E}\left[\epsilon^2 q^2\right]$ $\cdots$ $\mathbb{E}\left[\epsilon^{m-1} q^{m-1}\right]$

$$\lim_{\epsilon \downarrow 0}(\quad) = \frac{\sigma_a^2 + \sigma_s^2}{2} \quad 2\left(\frac{\sigma_a^2 + \sigma_s^2}{2}\right)^2 \quad (m-1)!\left(\frac{\sigma_a^2 + \sigma_s^2}{2}\right)^{m-1}$$

$$\epsilon q \Rightarrow Expo\left(\frac{\sigma_a^2 + \sigma_s^2}{2}\right)$$

Single-Server Queue and Drift Method

- In steady-state ($k \to \infty$):

$$\mathbb{E}\left[q^2(k+1)\right] = \mathbb{E}\left[q^2(k)\right]$$

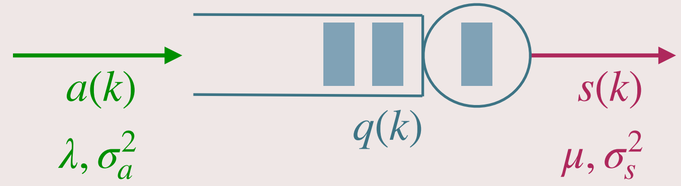
- Yields:

$$\mathbb{E}[q] = \frac{\mathbb{E}\left[(a-s)^2\right]}{2\epsilon} - \frac{\mathbb{E}\left[u^2\right]}{2\epsilon}$$

Small compared to the first term

- Therefore,

$$\lim_{\epsilon \downarrow 0} \mathbb{E}\left[\epsilon q\right] = \frac{\sigma_a^2 + \sigma_s^2}{2}$$



Dynamics of the queues:

$$q(k+1) = q(k) + a(k) - s(k) + u(k)$$
$$\implies q(k+1)u(k) = 0$$

Drift Method:

Test function: $1 + \epsilon q + \frac{1}{2}\epsilon^2 q^2 + \frac{1}{3!}\epsilon^3 q^3 + \cdots + \frac{1}{m!}\epsilon^m q^m + \cdots = e^{\epsilon q}$

Obtain: $1 + \mathbb{E}\left[\epsilon q\right] + \frac{1}{2}\mathbb{E}\left[\epsilon^2 q^2\right] + \cdots + \frac{1}{(m-1)!}\mathbb{E}\left[\epsilon^{m-1} q^{m-1}\right] + \cdots = \mathbb{E}\left[e^{\epsilon q}\right]$

$$\lim_{\epsilon \downarrow 0}(\quad) = \frac{\sigma_a^2 + \sigma_s^2}{2} \quad 2\left(\frac{\sigma_a^2 + \sigma_s^2}{2}\right)^2 \quad (m-1)!\left(\frac{\sigma_a^2 + \sigma_s^2}{2}\right)^{m-1}$$

$$\epsilon q \Rightarrow Expo\left(\frac{\sigma_a^2 + \sigma_s^2}{2}\right)$$

Single-Server Queue and Drift Method

- In steady-state ($k \to \infty$):

$$\mathbb{E}\left[q^2(k+1)\right] = \mathbb{E}\left[q^2(k)\right]$$

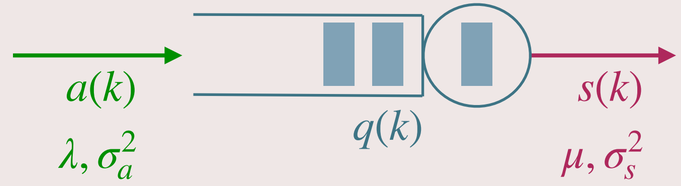
- Yields:

$$\mathbb{E}[q] = \frac{\mathbb{E}\left[(a-s)^2\right]}{2\epsilon} - \frac{\mathbb{E}\left[u^2\right]}{2\epsilon}$$

Small compared to the first term

- Therefore,

$$\lim_{\epsilon \downarrow 0} \mathbb{E}\left[\epsilon q\right] = \frac{\sigma_a^2 + \sigma_s^2}{2}$$



Dynamics of the queues:

$$q(k+1) = q(k) + a(k) - s(k) + u(k)$$
$$\implies q(k+1)u(k) = 0$$

Drift Method:

Test function: $1 + \epsilon q + \frac{1}{2}\epsilon^2 q^2 + \frac{1}{3!}\epsilon^3 q^3 + \cdots + \frac{1}{m!}\epsilon^m q^m + \cdots = e^{\epsilon q}$

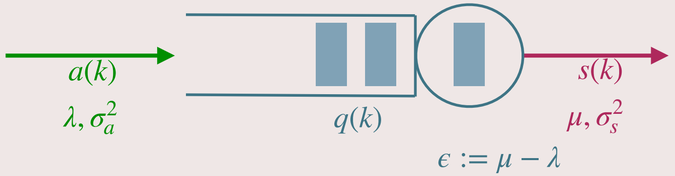
Obtain: $1 + \mathbb{E}\left[\epsilon q\right] + \frac{1}{2}\mathbb{E}\left[\epsilon^2 q^2\right] + \cdots + \frac{1}{(m-1)!}\mathbb{E}\left[\epsilon^{m-1} q^{m-1}\right] + \cdots = \mathbb{E}\left[e^{\epsilon q}\right]$

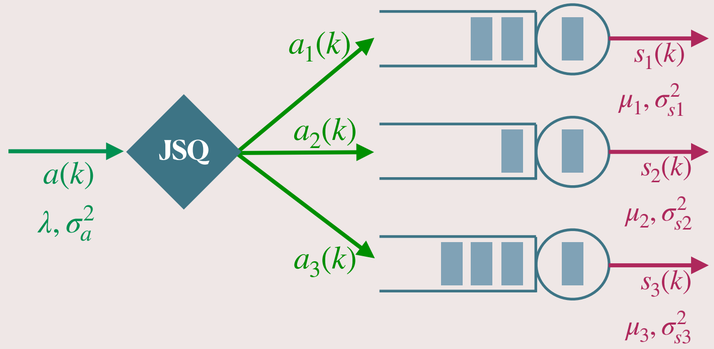
$$\lim_{\epsilon \downarrow 0}(\quad) = \frac{\sigma_a^2 + \sigma_s^2}{2} \quad 2\left(\frac{\sigma_a^2 + \sigma_s^2}{2}\right)^2 \quad (m-1)!\left(\frac{\sigma_a^2 + \sigma_s^2}{2}\right)^{m-1}$$

$$\epsilon q \Rightarrow Expo\left(\frac{\sigma_a^2 + \sigma_s^2}{2}\right)$$

Use $e^{\theta \epsilon q}$ as test function

The MGF Method [HL, Maguluri '20]



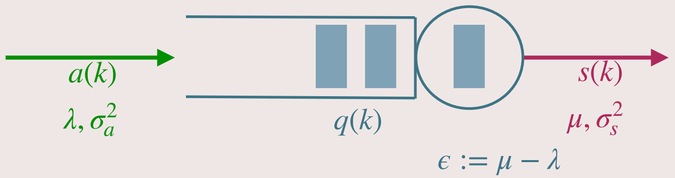


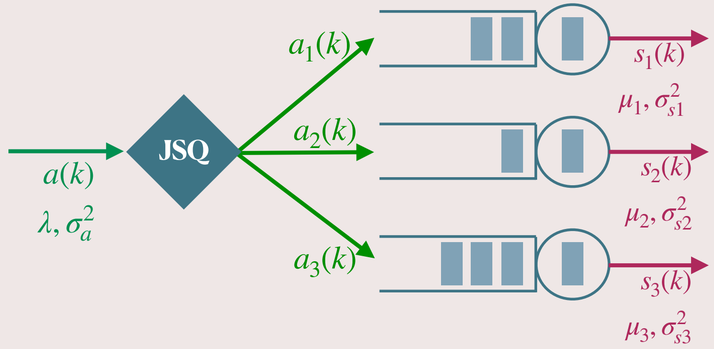
$$\epsilon := \sum_i \mu_i - \lambda$$

$$\boldsymbol{q}_{\|} := \mathbf{1}\left(\frac{\sum_j q_j}{N}\right)$$

$$\& \ \boldsymbol{q}_{\perp} := \boldsymbol{q} - \boldsymbol{q}_{\|}$$

The MGF Method [HL, Maguluri '20]





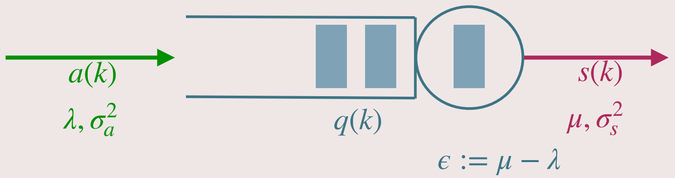
$$\epsilon := \sum_i \mu_i - \lambda$$

$$\boldsymbol{q}_{\|} := \mathbf{1}\left(\frac{\sum_j q_j}{N}\right)$$

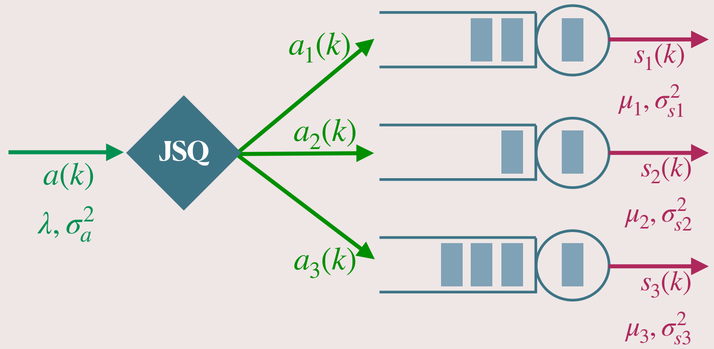
$$\& \ \boldsymbol{q}_{\perp} := \boldsymbol{q} - \boldsymbol{q}_{\|}$$

Set the drift of $V(q) = e^{\theta\epsilon q}$ to zero

The MGF Method [HL, Maguluri '20]



Set the drift of $V(q) = e^{\theta \epsilon q}$ to zero



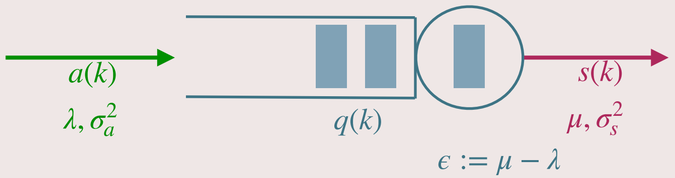
$$\epsilon := \sum_i \mu_i - \lambda$$

$$\boldsymbol{q}_{\|} := \mathbf{1}\left(\frac{\sum_j q_j}{N}\right)$$

$$\& \ \boldsymbol{q}_{\perp} := \boldsymbol{q} - \boldsymbol{q}_{\|}$$

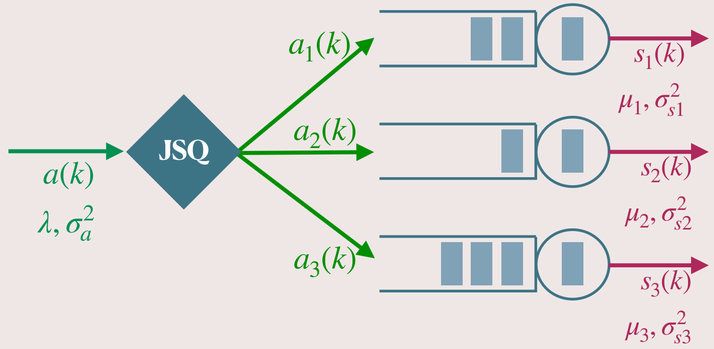
Set the drift of $V(\boldsymbol{q}) = e^{\theta \epsilon \sum_i q_i}$ to zero

The MGF Method [HL, Maguluri '20]



Set the drift of $V(q) = e^{\theta \epsilon q}$ to zero

Key Lemma: $\left(e^{\theta \epsilon q(k+1)} - 1\right)\left(e^{-\theta \epsilon u(k)} - 1\right) = 0$



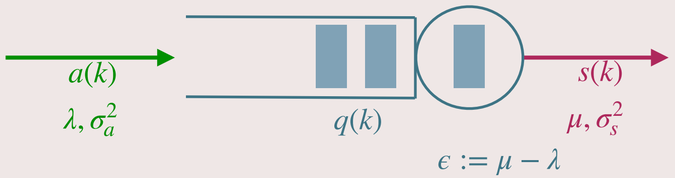
$$\epsilon := \sum_i \mu_i - \lambda$$

$$\boldsymbol{q}_{\|} := \mathbf{1}\left(\frac{\sum_j q_j}{N}\right)$$

$$\& \ \boldsymbol{q}_{\perp} := \boldsymbol{q} - \boldsymbol{q}_{\|}$$

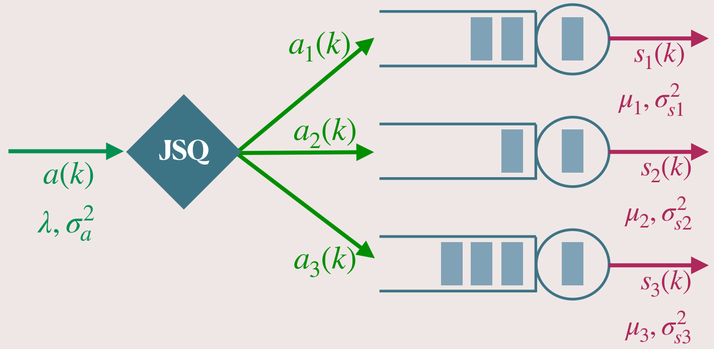
Set the drift of $V(\boldsymbol{q}) = e^{\theta \epsilon \sum_i q_i}$ to zero

The MGF Method [HL, Maguluri '20]



Set the drift of $V(q) = e^{\theta \epsilon q}$ to zero

Key Lemma: $\left(e^{\theta \epsilon q(k+1)} - 1\right)\left(e^{-\theta \epsilon u(k)} - 1\right) = 0$



$\epsilon := \sum_i \mu_i - \lambda$

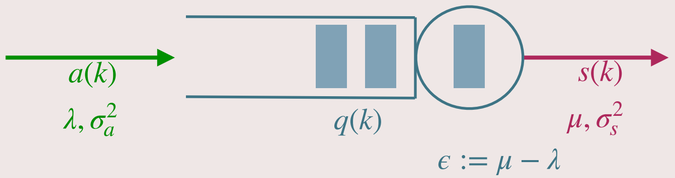
$\boldsymbol{q}_{\|} := \mathbf{1}\left(\frac{\sum_j q_j}{N}\right)$

& $\boldsymbol{q}_{\perp} := \boldsymbol{q} - \boldsymbol{q}_{\|}$

Set the drift of $V(\boldsymbol{q}) = e^{\theta \epsilon \sum_i q_i}$ to zero

Key Lemma: $\mathbb{E}\left[\left(e^{\theta \epsilon \sum_i q_i(k+1)} - 1\right)\left(e^{-\theta \epsilon \sum_i u_i(k)} - 1\right)\right]$ is $o(\epsilon^2)$

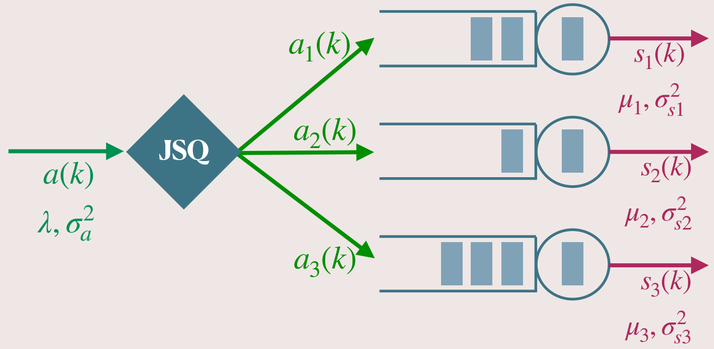
The MGF Method [HL, Maguluri '20]



Set the drift of $V(q) = e^{\theta\epsilon q}$ to zero

Key Lemma: $\left(e^{\theta\epsilon q(k+1)} - 1\right)\left(e^{-\theta\epsilon u(k)} - 1\right) = 0$

Yields: $\lim_{\epsilon \downarrow 0} \mathbb{E}\left[e^{\theta\epsilon q}\right] = \dfrac{1}{1 - \theta\left(\dfrac{\sigma_a^2 + \sigma_s^2}{2}\right)}$



$\epsilon := \sum_i \mu_i - \lambda$

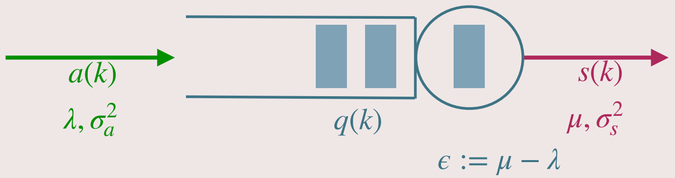
$\boldsymbol{q}_{\|} := \mathbf{1}\left(\dfrac{\sum_j q_j}{N}\right)$

& $\boldsymbol{q}_{\perp} := \boldsymbol{q} - \boldsymbol{q}_{\|}$

Set the drift of $V(\boldsymbol{q}) = e^{\theta\epsilon \sum_i q_i}$ to zero

Key Lemma: $\mathbb{E}\left[\left(e^{\theta\epsilon \sum_i q_i(k+1)} - 1\right)\left(e^{-\theta\epsilon \sum_i u_i(k)} - 1\right)\right]$ is $o(\epsilon^2)$

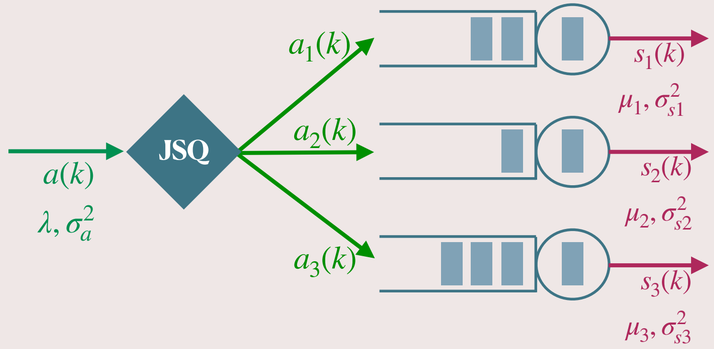
The MGF Method [HL, Maguluri '20]



Set the drift of $V(q) = e^{\theta\epsilon q}$ to zero

Key Lemma: $\left(e^{\theta\epsilon q(k+1)} - 1\right)\left(e^{-\theta\epsilon u(k)} - 1\right) = 0$

Yields: $\lim_{\epsilon \downarrow 0} \mathbb{E}\left[e^{\theta\epsilon q}\right] = \dfrac{1}{1 - \theta\left(\dfrac{\sigma_a^2 + \sigma_s^2}{2}\right)}$



$\epsilon := \sum_i \mu_i - \lambda$

$\boldsymbol{q}_{\|} := \mathbf{1}\left(\dfrac{\sum_j q_j}{N}\right)$

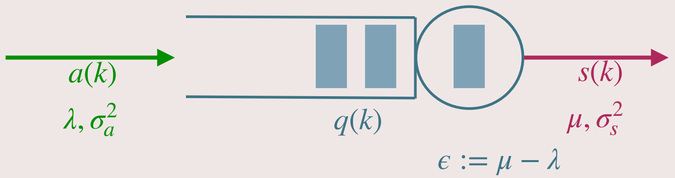
& $\boldsymbol{q}_{\perp} := \boldsymbol{q} - \boldsymbol{q}_{\|}$

Set the drift of $V(\boldsymbol{q}) = e^{\theta\epsilon \sum_i q_i}$ to zero

Key Lemma: $\mathbb{E}\left[\left(e^{\theta\epsilon \sum_i q_i(k+1)} - 1\right)\left(e^{-\theta\epsilon \sum_i u_i(k)} - 1\right)\right]$ is $o(\epsilon^2)$

Yields: $\lim_{\epsilon \downarrow 0} \mathbb{E}\left[e^{\theta\epsilon \sum_i q_i}\right] = \dfrac{1}{1 - \theta\left(\dfrac{\sigma_a^2 + \sum_i \sigma_{si}^2}{2}\right)}$

The MGF Method [HL, Maguluri '20]

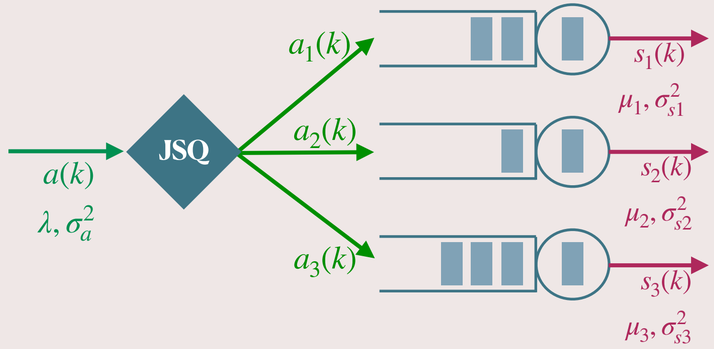


Set the drift of $V(q) = e^{\theta \epsilon q}$ to zero

Key Lemma: $\left(e^{\theta \epsilon q(k+1)} - 1\right)\left(e^{-\theta \epsilon u(k)} - 1\right) = 0$

Yields: $\lim_{\epsilon \downarrow 0} \mathbb{E}\left[e^{\theta \epsilon q}\right] = \dfrac{1}{1 - \theta\left(\dfrac{\sigma_a^2 + \sigma_s^2}{2}\right)}$

$\epsilon q \Rightarrow Expo\left(\dfrac{\sigma_a^2 + \sigma_s^2}{2}\right)$



$\epsilon := \sum_i \mu_i - \lambda$

$\boldsymbol{q}_{\|} := \mathbf{1}\left(\dfrac{\sum_j q_j}{N}\right)$

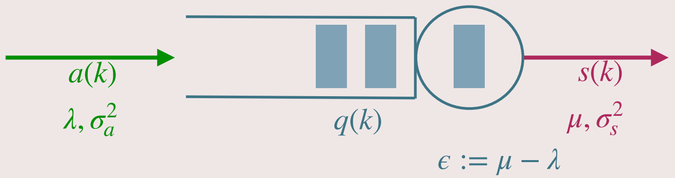
& $\boldsymbol{q}_{\perp} := \boldsymbol{q} - \boldsymbol{q}_{\|}$

Set the drift of $V(\boldsymbol{q}) = e^{\theta \epsilon \sum_i q_i}$ to zero

Key Lemma: $\mathbb{E}\left[\left(e^{\theta \epsilon \sum_i q_i(k+1)} - 1\right)\left(e^{-\theta \epsilon \sum_i u_i(k)} - 1\right)\right]$ is $o(\epsilon^2)$

Yields: $\lim_{\epsilon \downarrow 0} \mathbb{E}\left[e^{\theta \epsilon \sum_i q_i}\right] = \dfrac{1}{1 - \theta\left(\dfrac{\sigma_a^2 + \sum_i \sigma_{si}^2}{2}\right)}$

The MGF Method [HL, Maguluri '20]

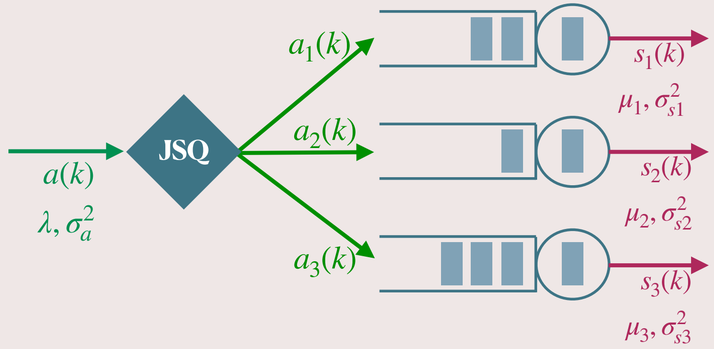


Set the drift of $V(q) = e^{\theta \epsilon q}$ to zero

Key Lemma: $\left(e^{\theta \epsilon q(k+1)} - 1\right)\left(e^{-\theta \epsilon u(k)} - 1\right) = 0$

Yields: $\lim_{\epsilon \downarrow 0} \mathbb{E}\left[e^{\theta \epsilon q}\right] = \dfrac{1}{1 - \theta\left(\dfrac{\sigma_a^2 + \sigma_s^2}{2}\right)}$

$\epsilon q \Rightarrow Expo\left(\dfrac{\sigma_a^2 + \sigma_s^2}{2}\right)$



$\epsilon := \sum_i \mu_i - \lambda$

$\boldsymbol{q}_{\|} := \mathbf{1}\left(\dfrac{\sum_j q_j}{N}\right)$

& $\boldsymbol{q}_{\perp} := \boldsymbol{q} - \boldsymbol{q}_{\|}$

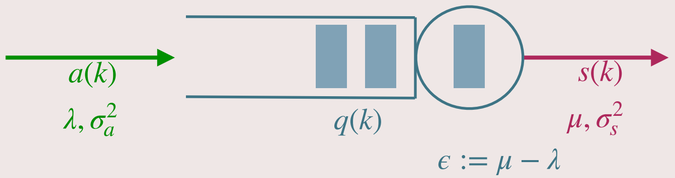
Set the drift of $V(\boldsymbol{q}) = e^{\theta \epsilon \sum_i q_i}$ to zero

Key Lemma: $\mathbb{E}\left[\left(e^{\theta \epsilon \sum_i q_i(k+1)} - 1\right)\left(e^{-\theta \epsilon \sum_i u_i(k)} - 1\right)\right]$ is $o(\epsilon^2)$

Yields: $\lim_{\epsilon \downarrow 0} \mathbb{E}\left[e^{\theta \epsilon \sum_i q_i}\right] = \dfrac{1}{1 - \theta\left(\dfrac{\sigma_a^2 + \sum_i \sigma_{si}^2}{2}\right)}$

Then, $\epsilon \boldsymbol{q}_{\|} \Rightarrow \dfrac{\mathbf{1}}{N} Expo\left(\dfrac{\sigma_a^2 + \sum_i \sigma_{si}^2}{2}\right)$

The MGF Method [HL, Maguluri '20]

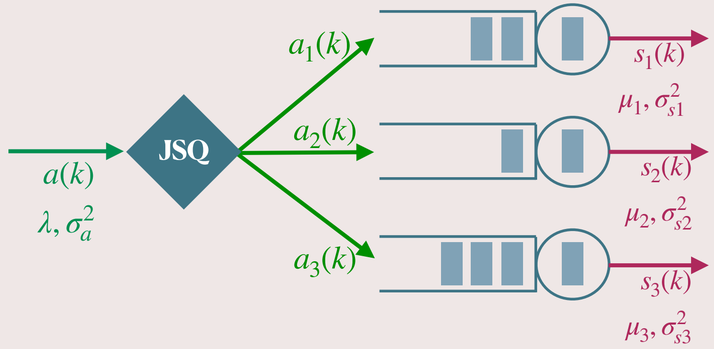


Set the drift of $V(q) = e^{\theta \epsilon q}$ to zero

Key Lemma: $\left(e^{\theta \epsilon q(k+1)} - 1\right)\left(e^{-\theta \epsilon u(k)} - 1\right) = 0$

Yields: $\lim_{\epsilon \downarrow 0} \mathbb{E}\left[e^{\theta \epsilon q}\right] = \dfrac{1}{1 - \theta\left(\dfrac{\sigma_a^2 + \sigma_s^2}{2}\right)}$

$\epsilon q \Rightarrow Expo\left(\dfrac{\sigma_a^2 + \sigma_s^2}{2}\right)$



$\epsilon := \sum_i \mu_i - \lambda$

$\boldsymbol{q}_{\|} := \mathbf{1}\left(\dfrac{\sum_j q_j}{N}\right)$

& $\boldsymbol{q}_{\perp} := \boldsymbol{q} - \boldsymbol{q}_{\|}$

Set the drift of $V(\boldsymbol{q}) = e^{\theta \epsilon \sum_i q_i}$ to zero

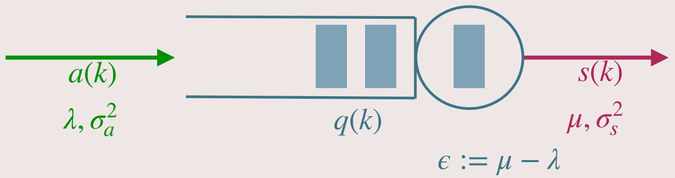
Key Lemma: $\mathbb{E}\left[\left(e^{\theta \epsilon \sum_i q_i(k+1)} - 1\right)\left(e^{-\theta \epsilon \sum_i u_i(k)} - 1\right)\right]$ is $o(\epsilon^2)$

Yields: $\lim_{\epsilon \downarrow 0} \mathbb{E}\left[e^{\theta \epsilon \sum_i q_i}\right] = \dfrac{1}{1 - \theta\left(\dfrac{\sigma_a^2 + \sum_i \sigma_{si}^2}{2}\right)}$

Then, $\epsilon \boldsymbol{q}_{\|} \Rightarrow \dfrac{\mathbf{1}}{N} Expo\left(\dfrac{\sigma_a^2 + \sum_i \sigma_{si}^2}{2}\right)$

SSC implies $\epsilon \boldsymbol{q}_{\perp} \Rightarrow \mathbf{0}$

The MGF Method [HL, Maguluri '20]

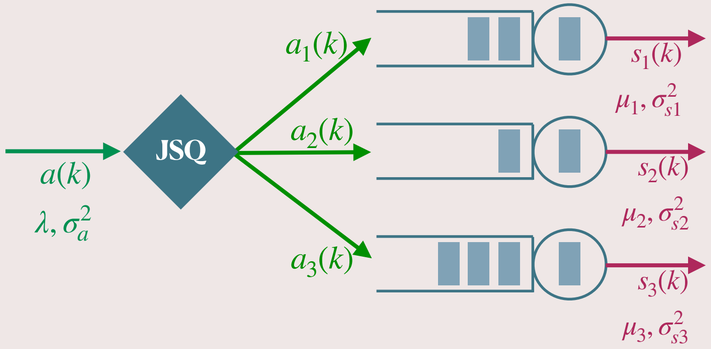


Set the drift of $V(q) = e^{\theta \epsilon q}$ to zero

Key Lemma: $\left(e^{\theta \epsilon q(k+1)} - 1\right)\left(e^{-\theta \epsilon u(k)} - 1\right) = 0$

Yields: $\lim_{\epsilon \downarrow 0} \mathbb{E}\left[e^{\theta \epsilon q}\right] = \dfrac{1}{1 - \theta\left(\dfrac{\sigma_a^2 + \sigma_s^2}{2}\right)}$

$\Rightarrow$ $\epsilon q \Rightarrow Expo\left(\dfrac{\sigma_a^2 + \sigma_s^2}{2}\right)$



$\epsilon := \sum_i \mu_i - \lambda$

$\boldsymbol{q}_{\|} := \mathbf{1}\left(\dfrac{\sum_j q_j}{N}\right)$

& $\boldsymbol{q}_{\perp} := \boldsymbol{q} - \boldsymbol{q}_{\|}$

Set the drift of $V(\boldsymbol{q}) = e^{\theta \epsilon \sum_i q_i}$ to zero

Key Lemma: $\mathbb{E}\left[\left(e^{\theta \epsilon \sum_i q_i(k+1)} - 1\right)\left(e^{-\theta \epsilon \sum_i u_i(k)} - 1\right)\right]$ is $o(\epsilon^2)$

Yields: $\lim_{\epsilon \downarrow 0} \mathbb{E}\left[e^{\theta \epsilon \sum_i q_i}\right] = \dfrac{1}{1 - \theta\left(\dfrac{\sigma_a^2 + \sum_i \sigma_{si}^2}{2}\right)}$

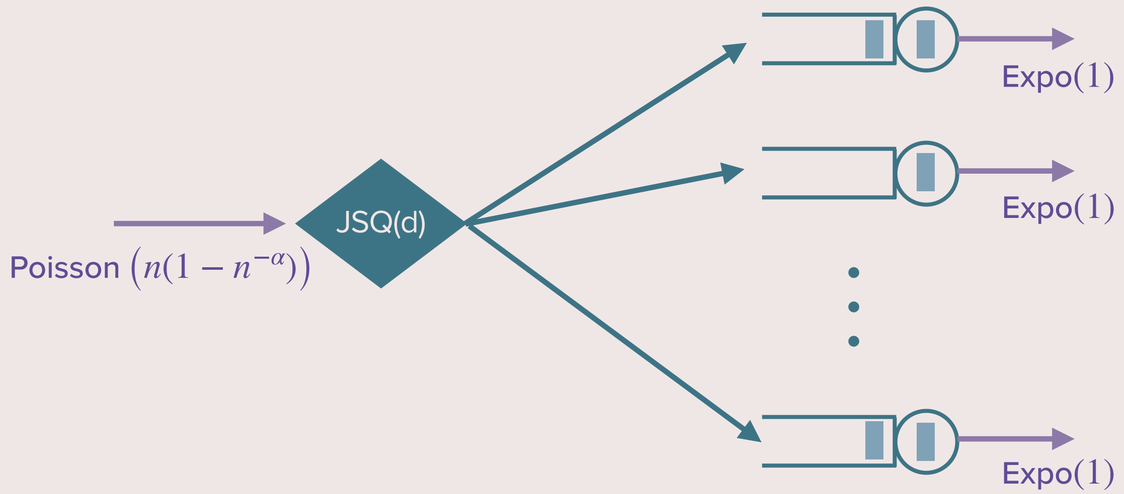
Then, $\epsilon \boldsymbol{q}_{\|} \Rightarrow \dfrac{\mathbf{1}}{N} Expo\left(\dfrac{\sigma_a^2 + \sum_i \sigma_{si}^2}{2}\right)$

SSC implies $\epsilon \boldsymbol{q}_{\perp} \Rightarrow \mathbf{0}$

$\Rightarrow$ $\epsilon \boldsymbol{q} \Rightarrow \dfrac{\mathbf{1}}{N} Expo\left(\dfrac{\sigma_a^2 + \sigma_s^2}{2}\right)$

Proof Sketch

Step 1: Multiplicative State Space Collapse



Proof Sketch

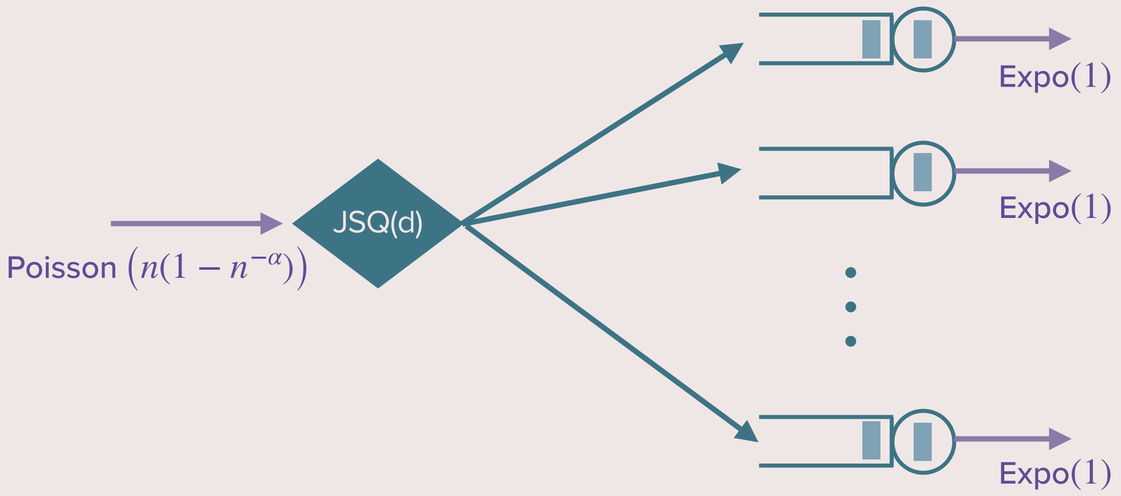
Step 1: Multiplicative State Space Collapse

Proposition (SSC): [HL, Maguluri '21]

$\boldsymbol{q}_{\|}(k) = \mathbf{1}\left(\frac{\sum_i q_i}{n}\right)$ = Projection of $\boldsymbol{q}$ on $\mathbf{1}$

$\boldsymbol{q}_{\perp}(k) = \boldsymbol{q}(k) - \boldsymbol{q}_{\|}(k)$ error of approximating $\boldsymbol{q} \approx \boldsymbol{q}_{\|}$

Then, for all $j = 1,2,3\ldots$ $\mathbb{E}\left[\|\boldsymbol{q}_{\perp}\|^j\right]^{\frac{1}{j}} \leq Cj\left(\frac{n^2}{d-1}\right)$



Proof Sketch

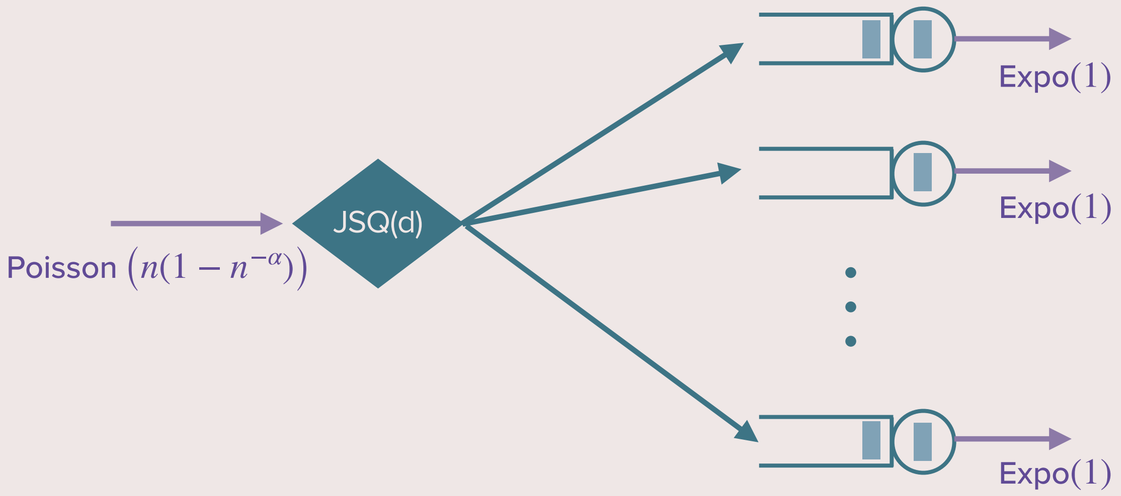
Step 1: Multiplicative State Space Collapse

Proposition (SSC): [HL, Maguluri '21]

$\boldsymbol{q}_{\|}(k) = \mathbf{1}\left(\frac{\sum_i q_i}{n}\right)$ = Projection of $\boldsymbol{q}$ on $\mathbf{1}$

$\boldsymbol{q}_{\perp}(k) = \boldsymbol{q}(k) - \boldsymbol{q}_{\|}(k)$ error of approximating $\boldsymbol{q} \approx \boldsymbol{q}_{\|}$

Then, for all $j = 1,2,3\ldots$ $\mathbb{E}\left[\|\boldsymbol{q}_{\perp}\|^j\right]^{\frac{1}{j}} \leq Cj\left(\frac{n^2}{d-1}\right)$



Step 2: Convergence in distribution

Proof Sketch

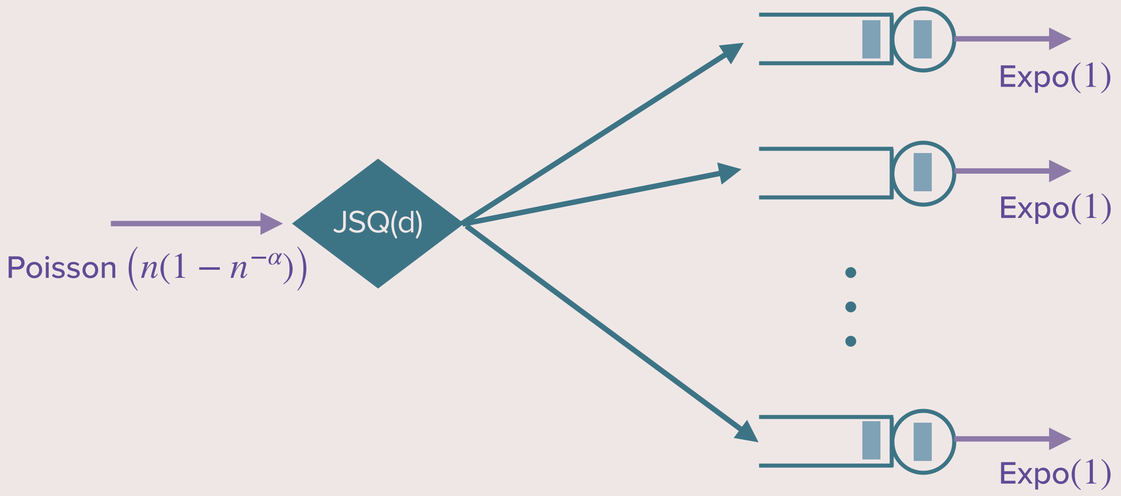
Step 1: Multiplicative State Space Collapse

Proposition (SSC): [HL, Maguluri '21]

$\boldsymbol{q}_{\parallel}(k) = \mathbf{1}\left(\frac{\sum_i q_i}{n}\right)$ = Projection of $\boldsymbol{q}$ on $\mathbf{1}$

$\boldsymbol{q}_{\perp}(k) = \boldsymbol{q}(k) - \boldsymbol{q}_{\parallel}(k)$ error of approximating $\boldsymbol{q} \approx \boldsymbol{q}_{\parallel}$

Then, for all $j = 1,2,3\ldots$ $\mathbb{E}\left[\|\boldsymbol{q}_{\perp}\|^j\right]^{\frac{1}{j}} \leq Cj\left(\frac{n^2}{d-1}\right)$



Step 2: Convergence in distribution

Proof #1: Transform method in continuous time

- Extend method to continuous time
- Translate unused service to $1\{q_i = 0\}$

Proof Sketch

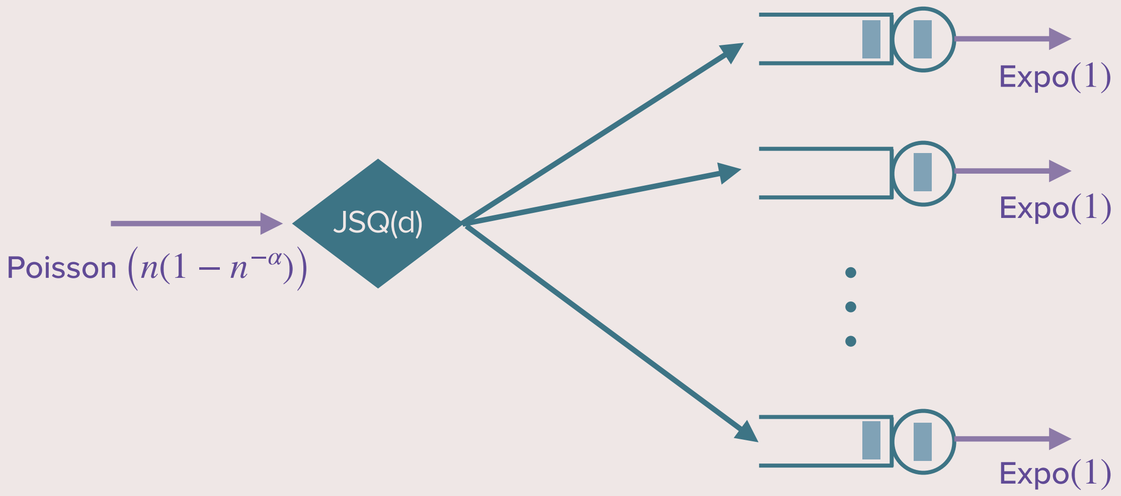
Step 1: Multiplicative State Space Collapse

Proposition (SSC): [HL, Maguluri '21]

$\boldsymbol{q}_{\parallel}(k) = \mathbf{1}\left(\frac{\sum_i q_i}{n}\right)$ = Projection of $\boldsymbol{q}$ on $\mathbf{1}$

$\boldsymbol{q}_{\perp}(k) = \boldsymbol{q}(k) - \boldsymbol{q}_{\parallel}(k)$ error of approximating $\boldsymbol{q} \approx \boldsymbol{q}_{\parallel}$

Then, for all $j = 1,2,3\ldots$ $\mathbb{E}\left[\|\boldsymbol{q}_{\perp}\|^j\right]^{\frac{1}{j}} \leq Cj\left(\frac{n^2}{d-1}\right)$



Step 2: Convergence in distribution

Proof #1: Transform method in continuous time

- Extend method to continuous time
- Translate unused service to $1\{q_i = 0\}$

Proof #2: Stein's method

- Bound Wasserstein's distance
- Obtain rate of convergence