Resource pooling in congested networks: proportional fairness and product form

Neil Walton

Joint work with:
Frank Kelly and Laurent Massoulié

Statistical Laboratory, University of Cambridge.

We are interested in studying proportional fairness as a way of sharing flow across different routes of a network

We review some recent results.

We are interested in studying proportional fairness as a way of sharing flow across different routes of a network

We review some recent results.

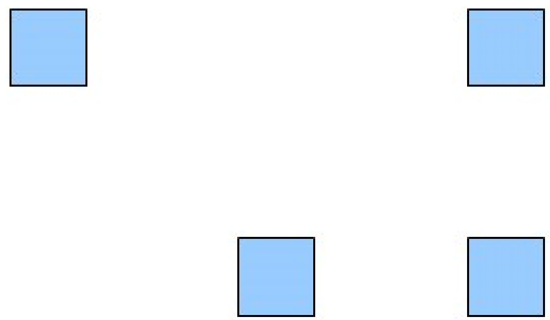
First we consider an equivalence between single-path and multi-path routing...

(Kang, Kelly, Lee, Williams '09)

A network (Kang, Kelly, Lee, Williams '09)

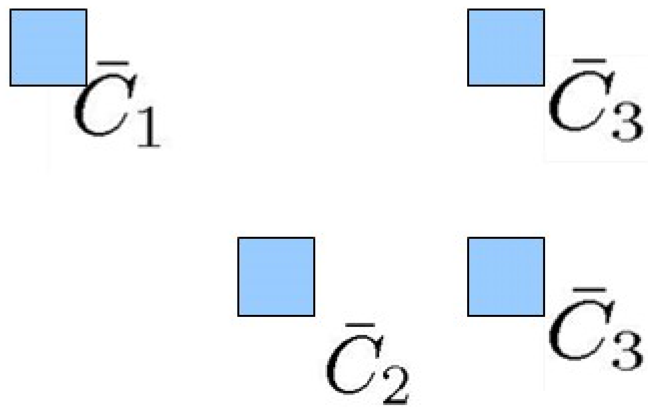
A network

(Kang, Kelly, Lee, Williams '09)



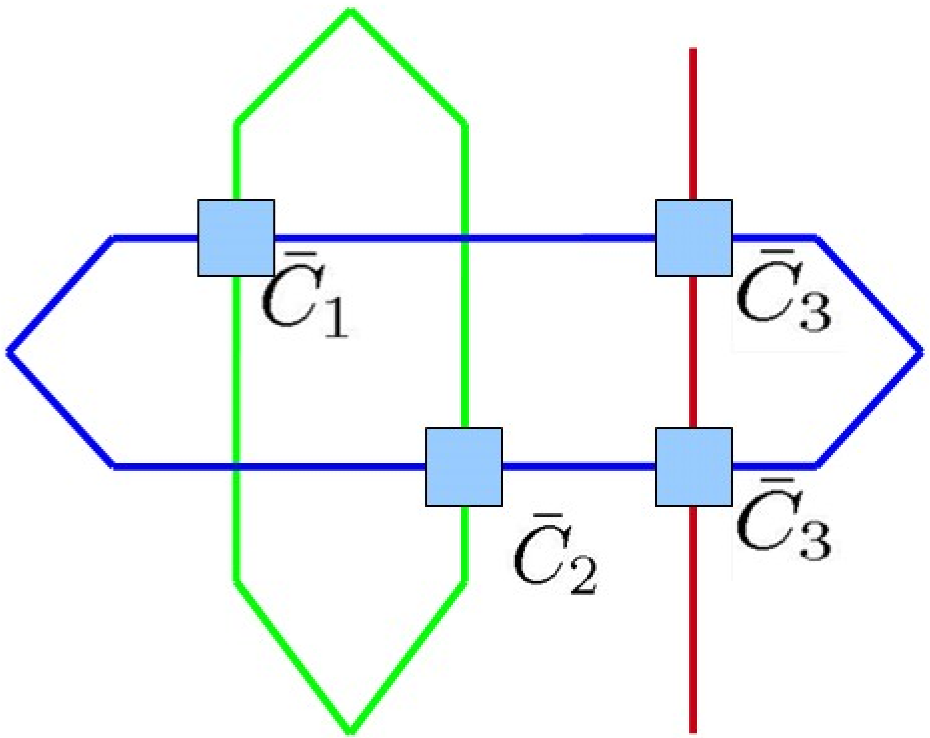
A network

(Kang, Kelly, Lee, Williams '09)



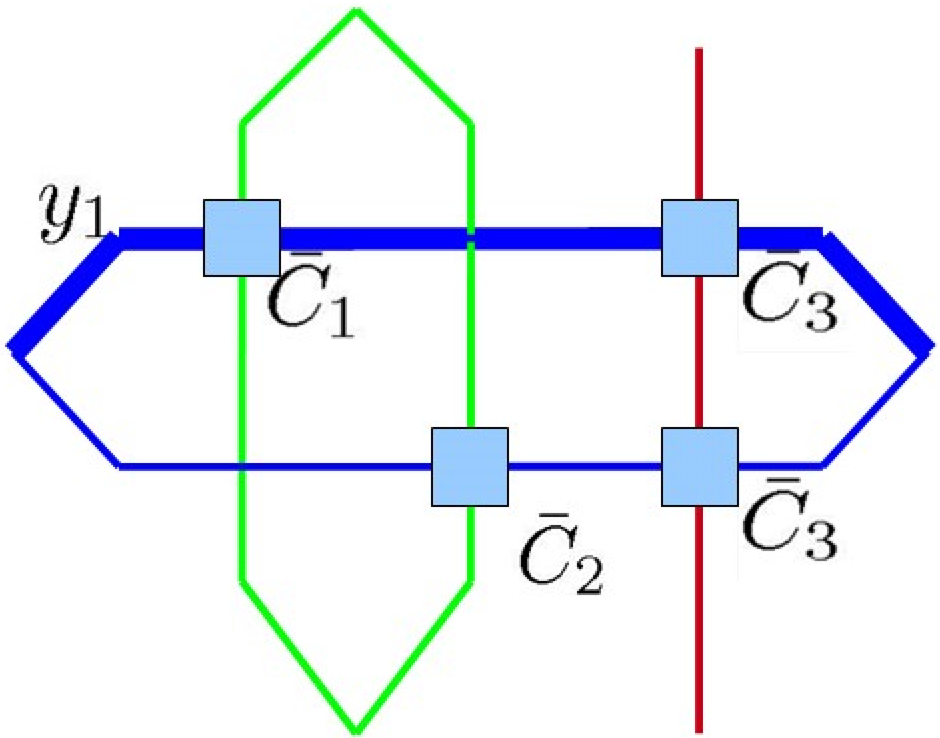
A network

(Kang, Kelly, Lee, Williams '09)



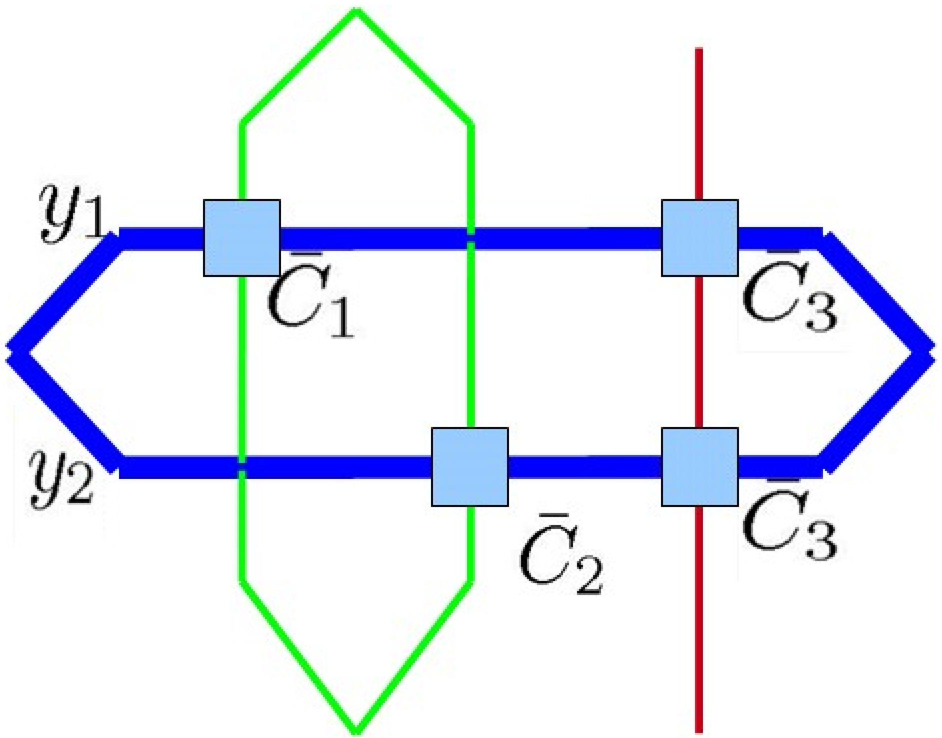
A network

(Kang, Kelly, Lee, Williams '09)



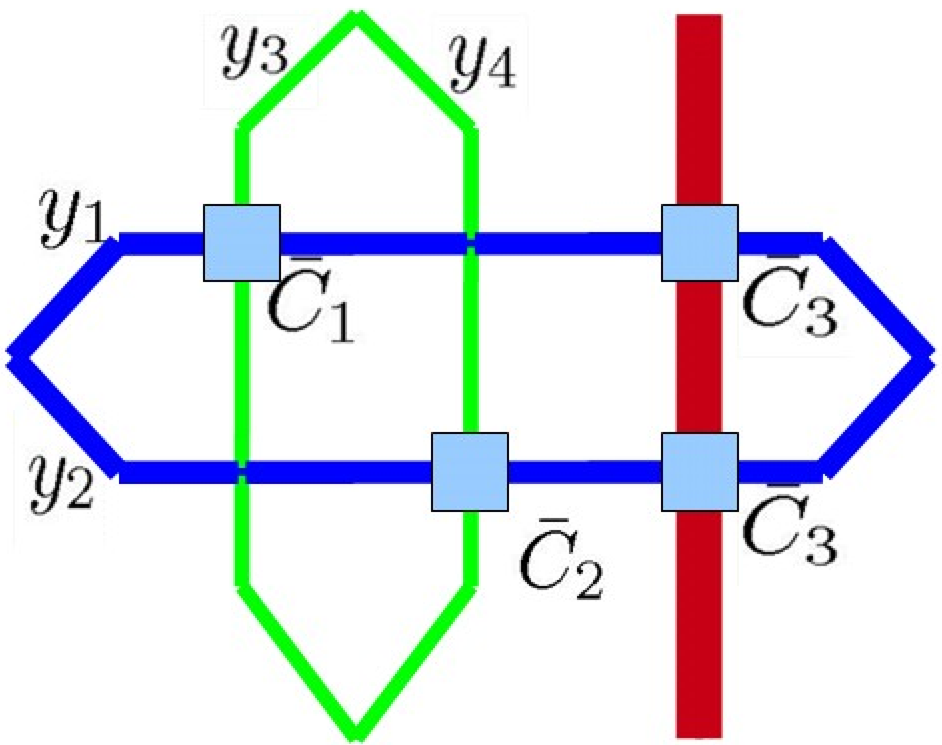
A network

(Kang, Kelly, Lee, Williams '09)



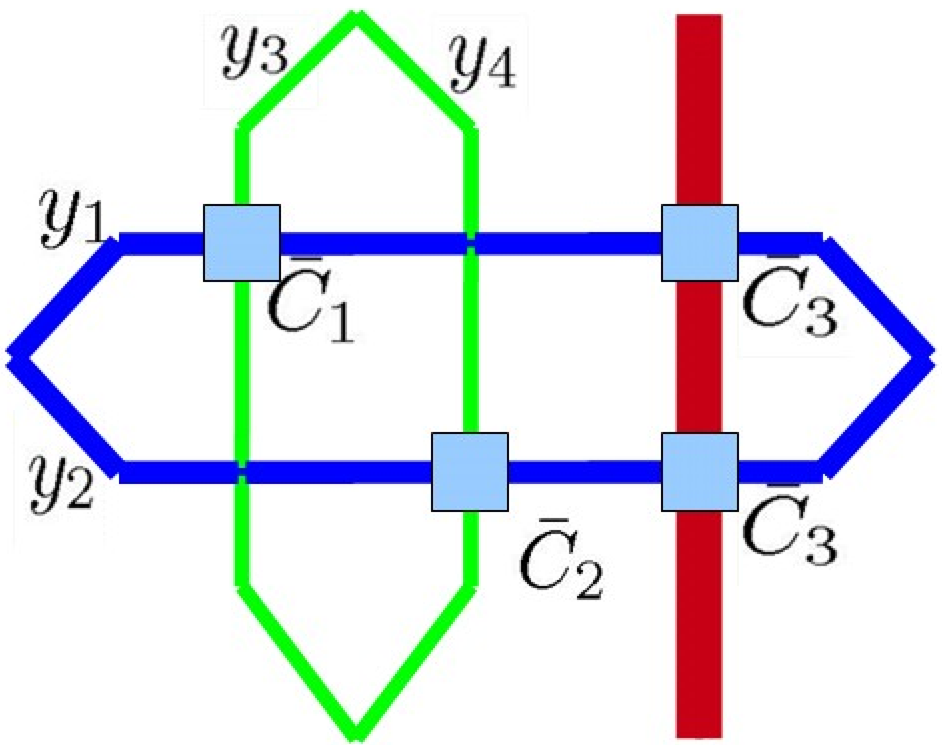
A network

(Kang, Kelly, Lee, Williams '09)



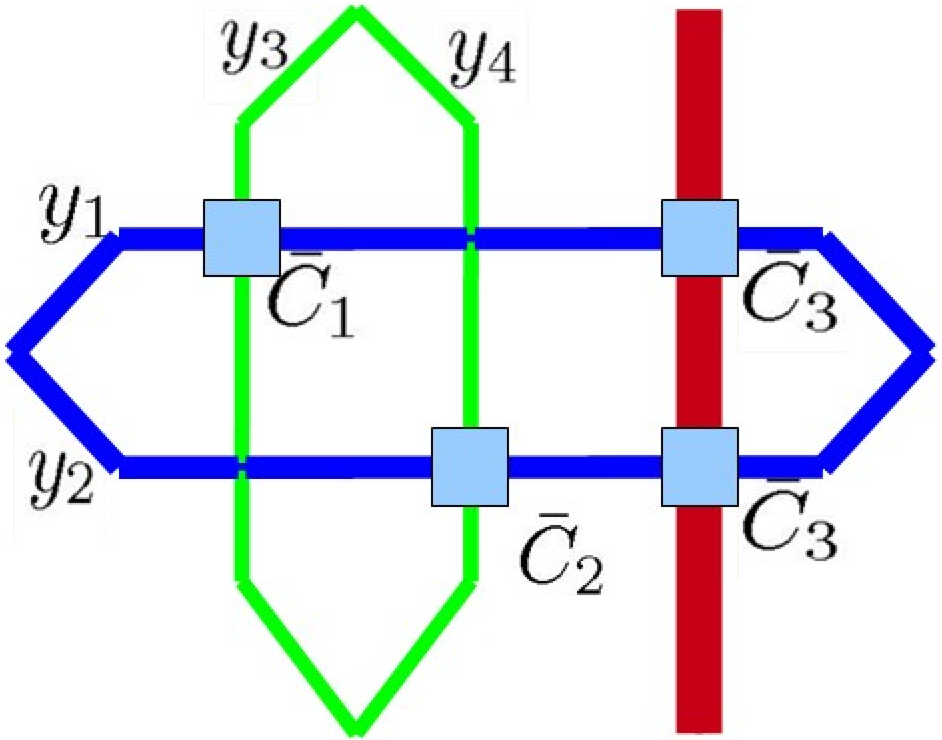
A network

(Kang, Kelly, Lee, Williams '09)



In general

A network (Kang, Kelly, Lee, Williams '09)

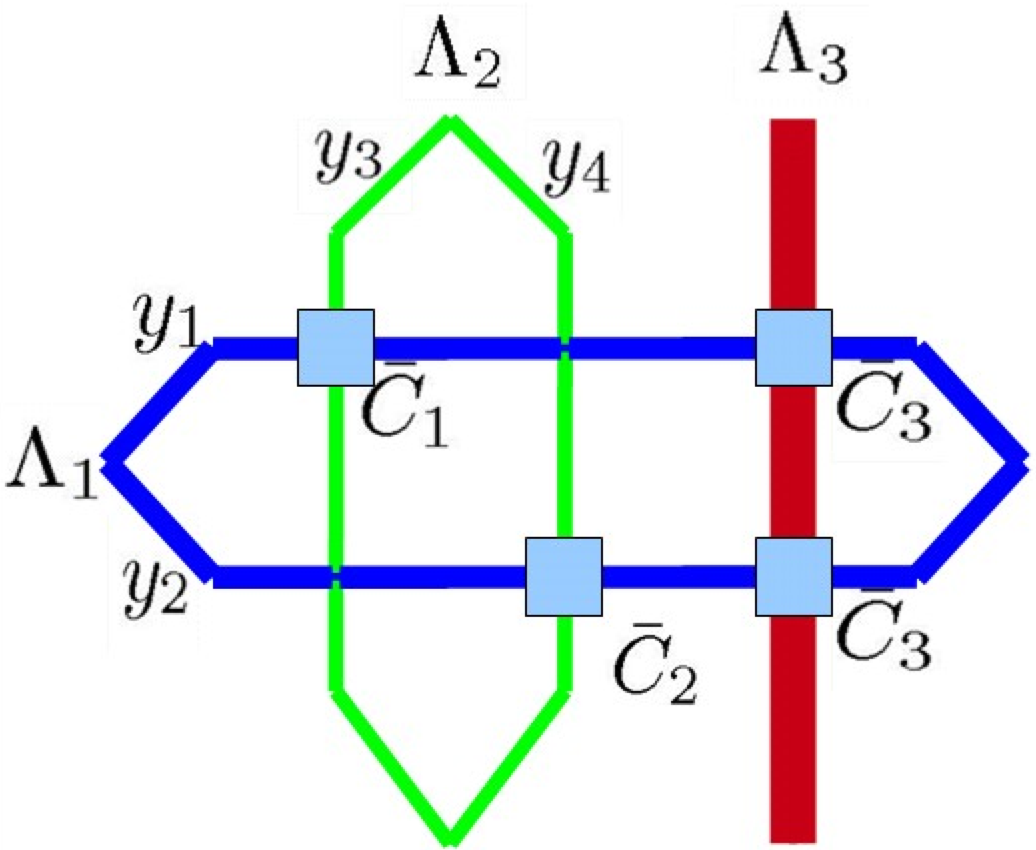


In general

$$\sum_{i:j\in i} y_i \le \bar{C}_j, \quad j \in \bar{\mathcal{J}}$$

A network

(Kang, Kelly, Lee, Williams '09)

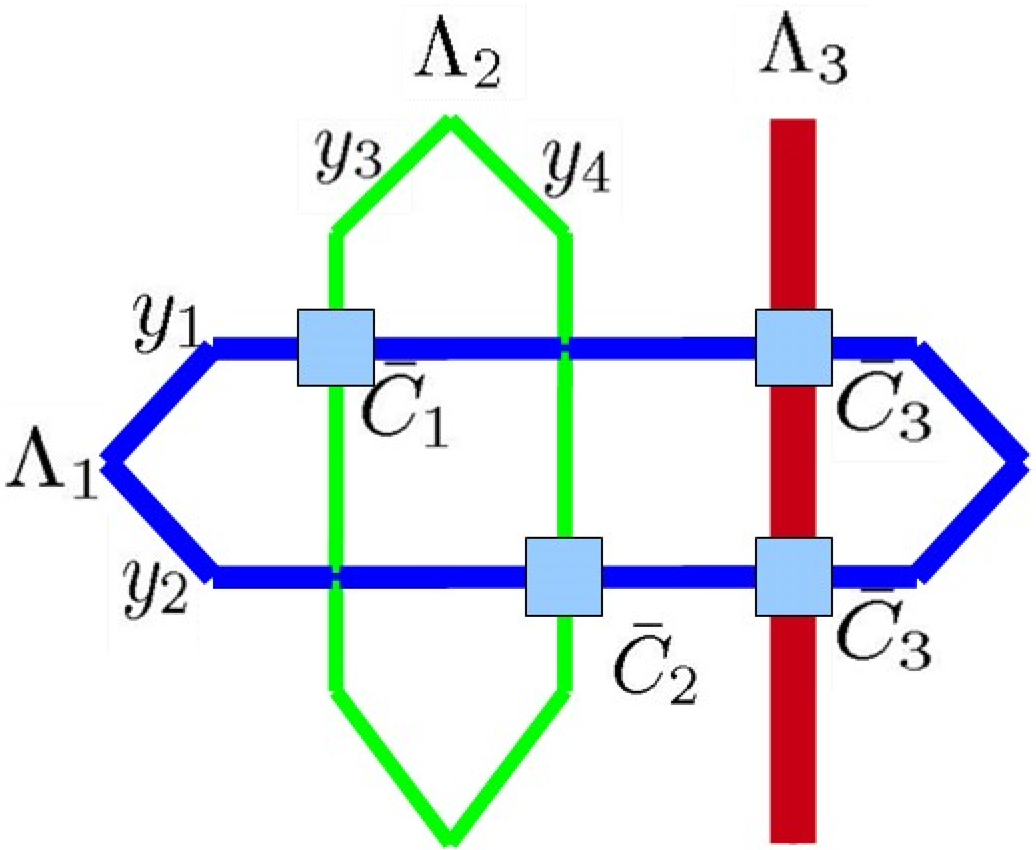


In general

$$\sum_{i:j\in i} y_i \leq \bar{C}_j, \quad j \in \bar{\mathcal{J}}$$

A network

(Kang, Kelly, Lee, Williams '09)



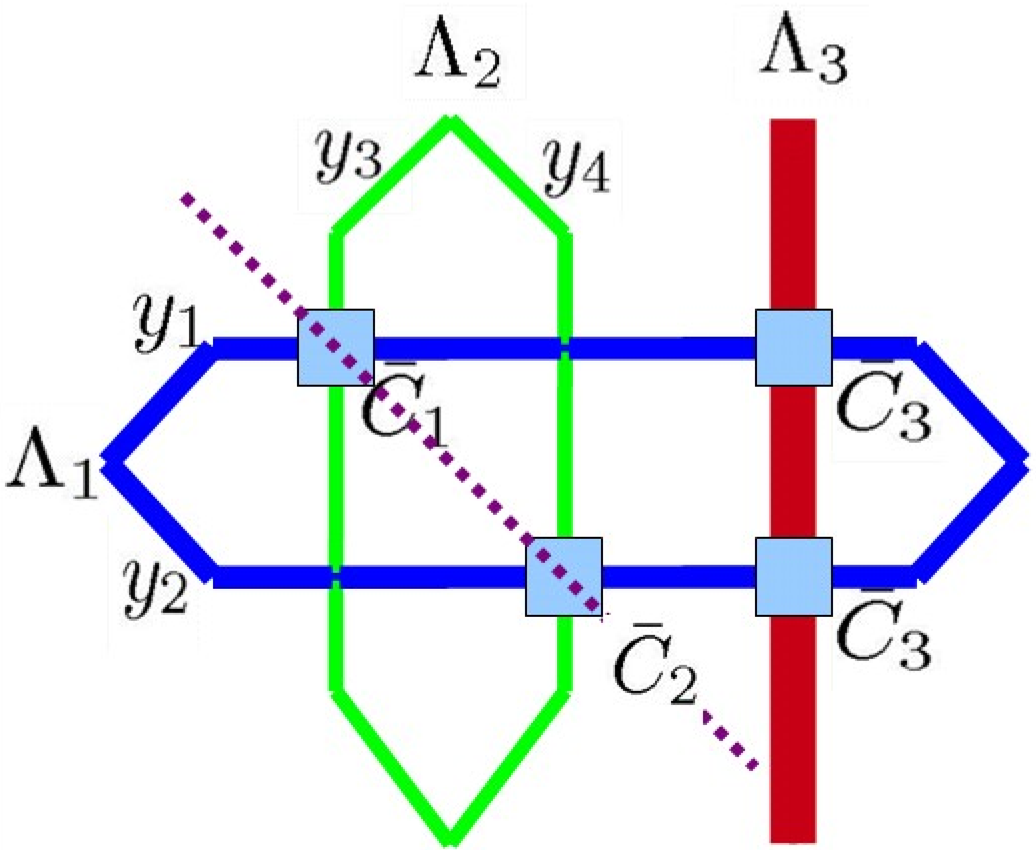
In general

$$\sum_{i:j\in i} y_i \le \bar{C}_j, \quad j \in \bar{\mathcal{J}}$$

$$\sum_{i\in s} y_i = \Lambda_s, \quad s \in \mathcal{S}$$

A network

(Kang, Kelly, Lee, Williams '09)



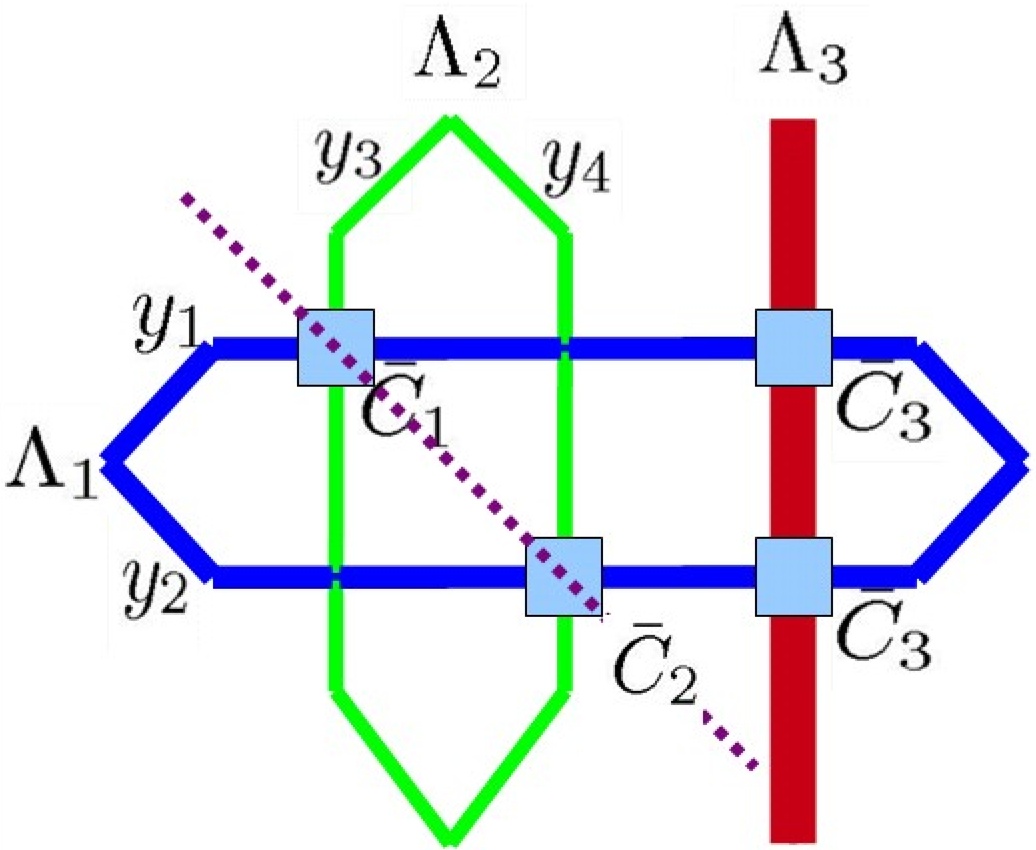
In general

$$\sum_{i:j\in i} y_i \leq \bar{C}_j, \quad j \in \bar{\mathcal{J}}$$

$$\sum_{i\in s} y_i = \Lambda_s, \quad s \in \mathcal{S}$$

A network

(Kang, Kelly, Lee, Williams '09)



$$\Lambda_1 + \Lambda_2 \leq \bar{C}_1 + \bar{C}_2$$

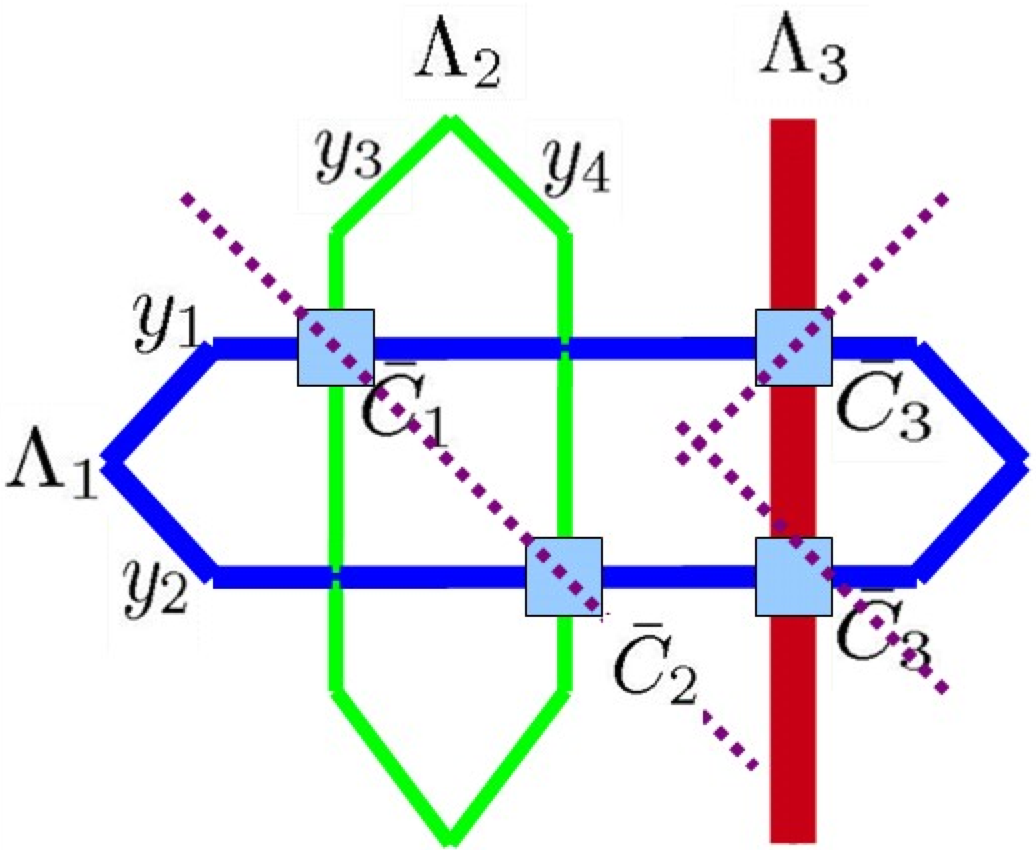
In general

$$\sum_{i:j\in i} y_i \leq \bar{C}_j, \quad j \in \bar{\mathcal{J}}$$

$$\sum_{i\in s} y_i = \Lambda_s, \quad s \in \mathcal{S}$$

A network

(Kang, Kelly, Lee, Williams '09)



$$\Lambda_1 + \Lambda_2 \leq \bar{C}_1 + \bar{C}_2$$

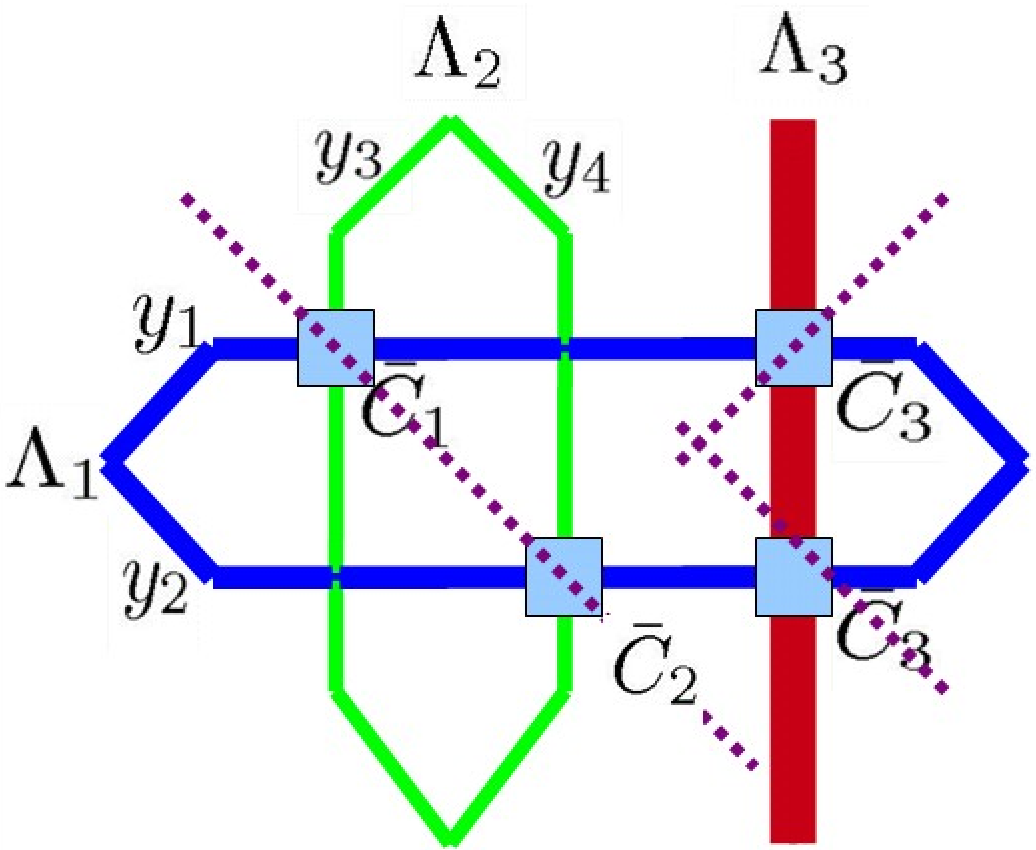
In general

$$\sum_{i:j\in i} y_i \leq \bar{C}_j, \quad j \in \bar{\mathcal{J}}$$

$$\sum_{i\in s} y_i = \Lambda_s, \quad s \in \mathcal{S}$$

A network

(Kang, Kelly, Lee, Williams '09)



$$\Lambda_1 + \Lambda_2 \leq \bar{C}_1 + \bar{C}_2$$

$$\frac{1}{2}\Lambda_1 + \Lambda_3 \leq \bar{C}_3$$

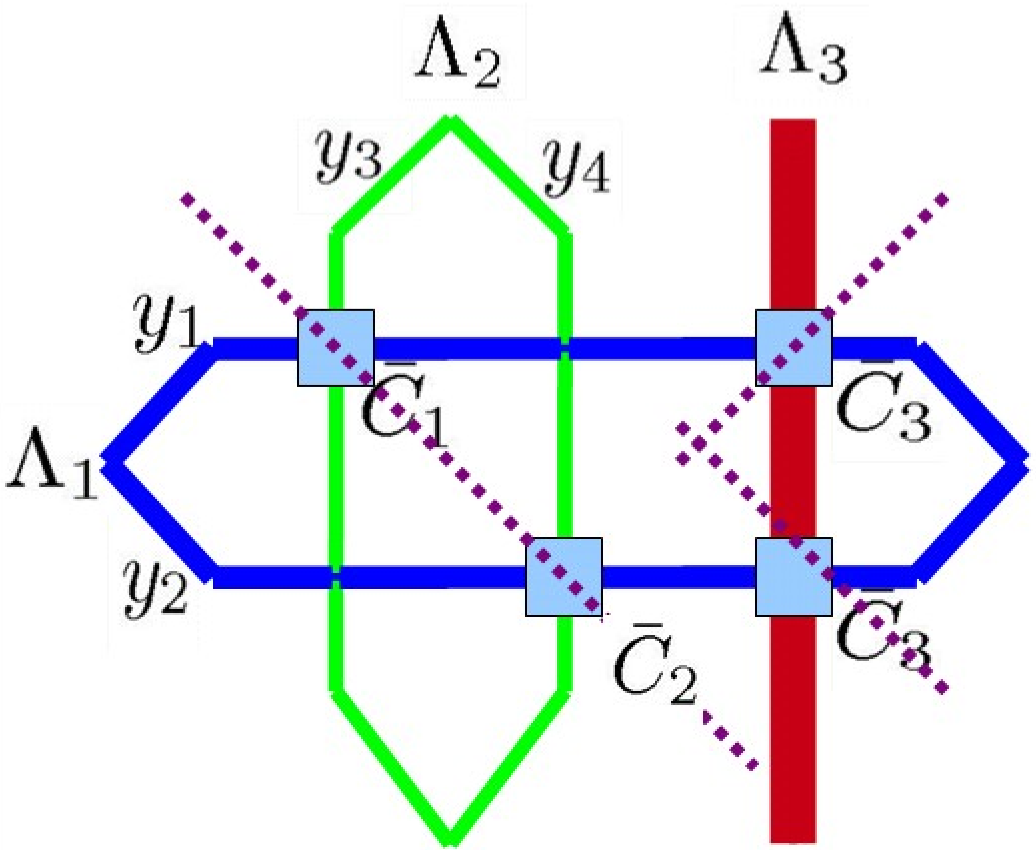
In general

$$\sum_{i:j\in i} y_i \leq \bar{C}_j, \quad j \in \bar{\mathcal{J}}$$

$$\sum_{i\in s} y_i = \Lambda_s, \quad s \in \mathcal{S}$$

A network

(Kang, Kelly, Lee, Williams '09)



$$\begin{pmatrix} 1 & 1 & 0 \\ \frac{1}{2} & 0 & 1 \end{pmatrix} \begin{pmatrix} \Lambda_1 \\ \Lambda_2 \\ \Lambda_3 \end{pmatrix} \leq \begin{pmatrix} \bar{C}_1 + \bar{C}_2 \\ \bar{C}_3 \end{pmatrix}$$

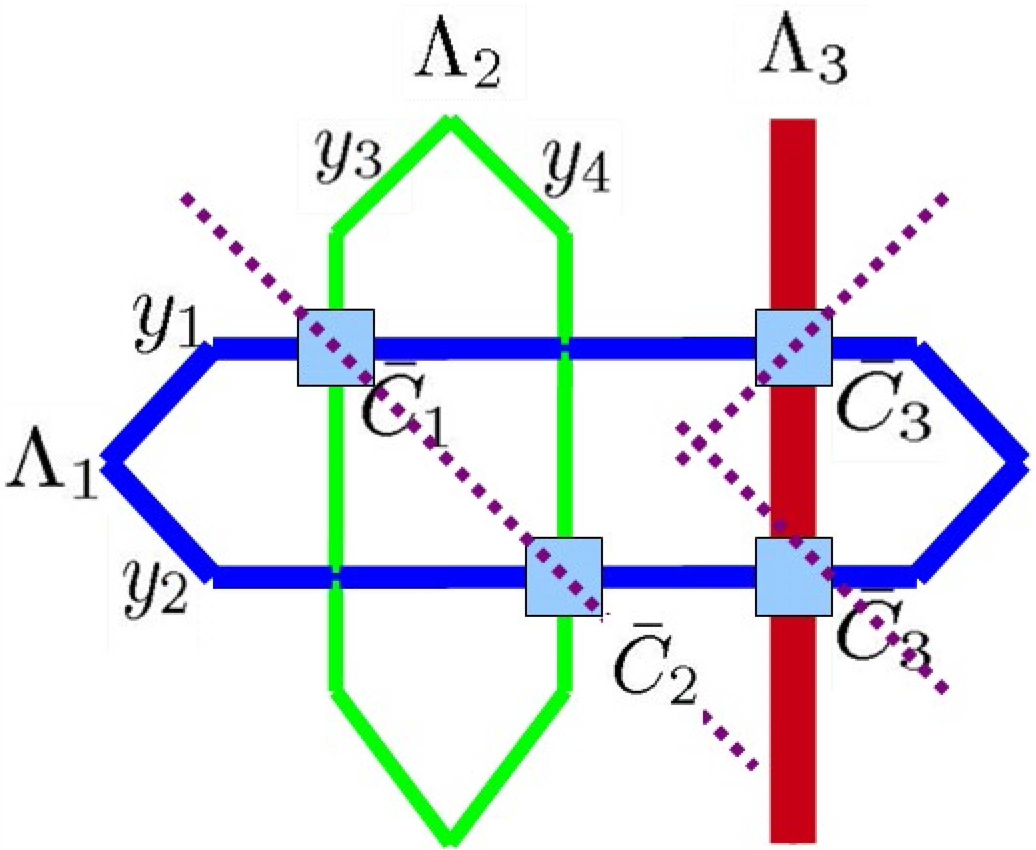
In general

$$\sum_{i:j\in i} y_i \leq \bar{C}_j, \quad j \in \bar{\mathcal{J}}$$

$$\sum_{i\in s} y_i = \Lambda_s, \quad s \in \mathcal{S}$$

A network

(Kang, Kelly, Lee, Williams '09)



$$\begin{pmatrix} 1 & 1 & 0 \\ \frac{1}{2} & 0 & 1 \end{pmatrix} \begin{pmatrix} \Lambda_1 \\ \Lambda_2 \\ \Lambda_3 \end{pmatrix} \leq \begin{pmatrix} \bar{C}_1 + \bar{C}_2 \\ \bar{C}_3 \end{pmatrix}$$

In general

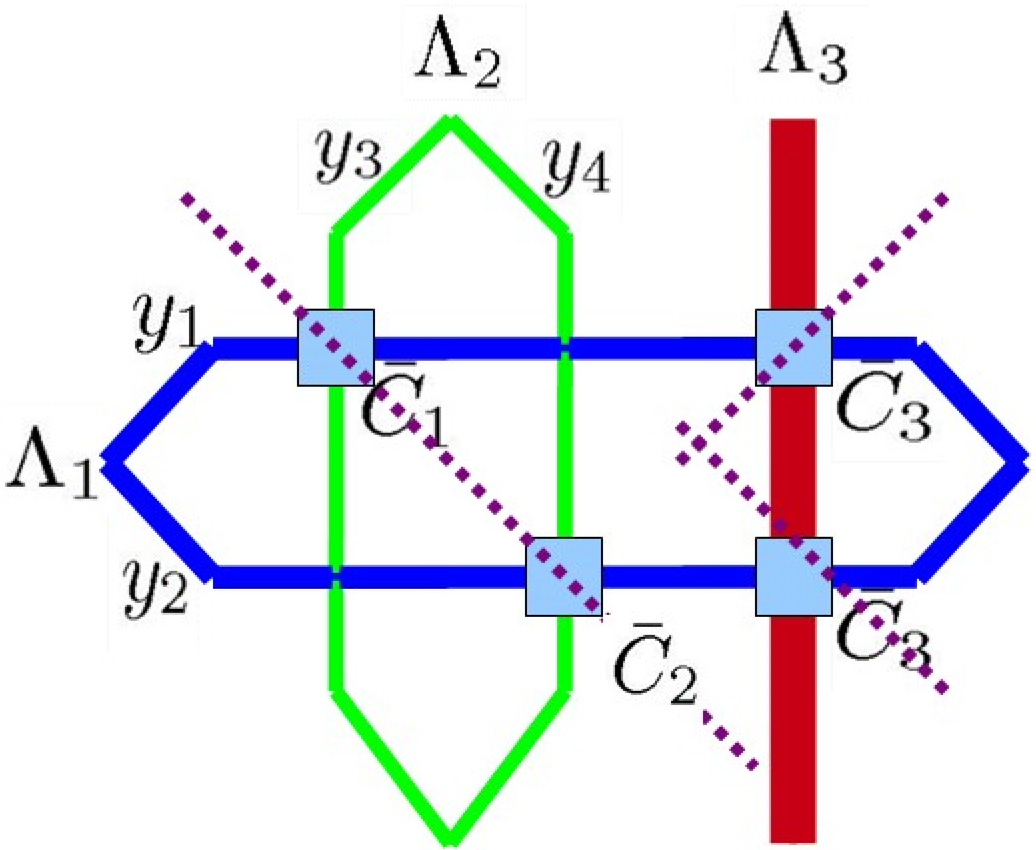
$$\sum_{i:j\in i} y_i \leq \bar{C}_j, \quad j \in \bar{\mathcal{J}}$$

$$\sum_{i\in s} y_i = \Lambda_s, \quad s \in \mathcal{S}$$

$$\Longleftrightarrow \quad \sum_i a_{ji}\Lambda_i \leq C_j, \quad j \in \mathcal{J}$$

A network

(Kang, Kelly, Lee, Williams '09)



$$\begin{pmatrix} 1 & 1 & 0 \\ \frac{1}{2} & 0 & 1 \end{pmatrix} \begin{pmatrix} \Lambda_1 \\ \Lambda_2 \\ \Lambda_3 \end{pmatrix} \le \begin{pmatrix} \bar{C}_1 + \bar{C}_2 \\ \bar{C}_3 \end{pmatrix}$$

In general

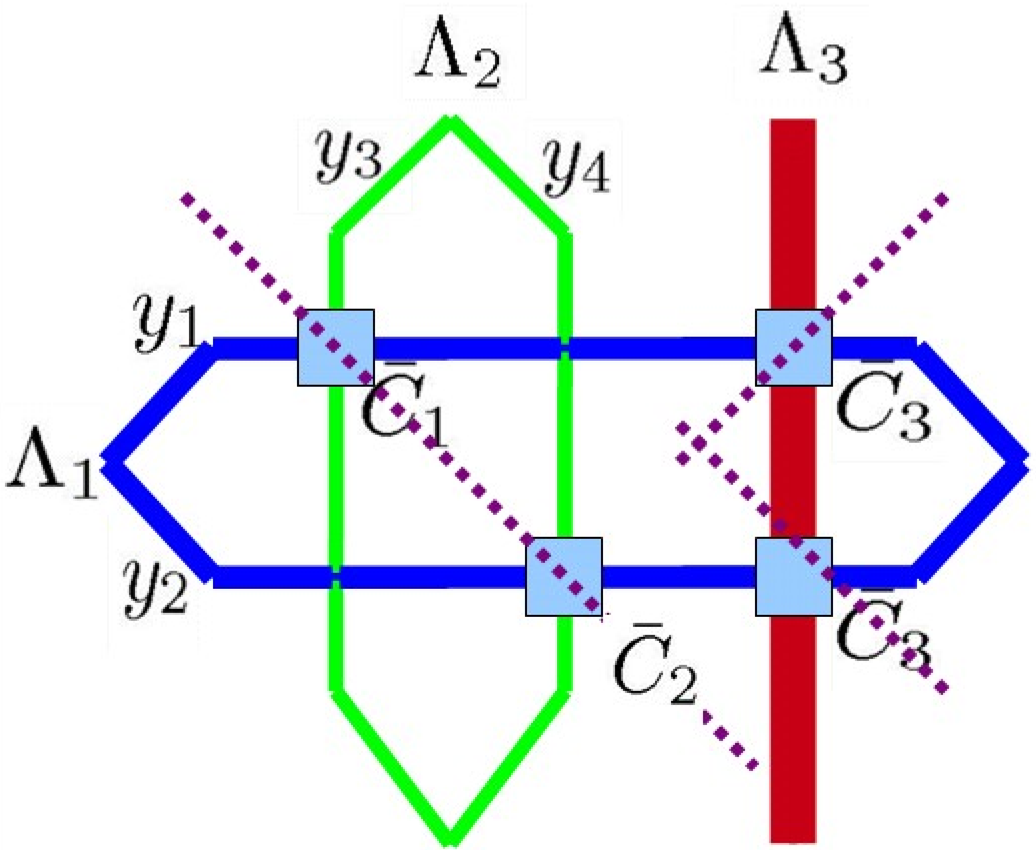
$$\sum_{i:j\in i} y_i \le \bar{C}_j, \quad j \in \bar{\mathcal{J}}$$

$$\sum_{i\in s} y_i = \Lambda_s, \quad s \in \mathcal{S}$$

$$\Longleftrightarrow \quad \sum_i a_{ji}\Lambda_i \le C_j, \quad j \in \mathcal{J}$$

A network

(Kang, Kelly, Lee, Williams '09)



$$\begin{pmatrix} 1 & 1 & 0 \\ \frac{1}{2} & 0 & 1 \end{pmatrix} \begin{pmatrix} \Lambda_1 \\ \Lambda_2 \\ \Lambda_3 \end{pmatrix} \le \begin{pmatrix} \bar{C}_1 + \bar{C}_2 \\ \bar{C}_3 \end{pmatrix}$$

In general

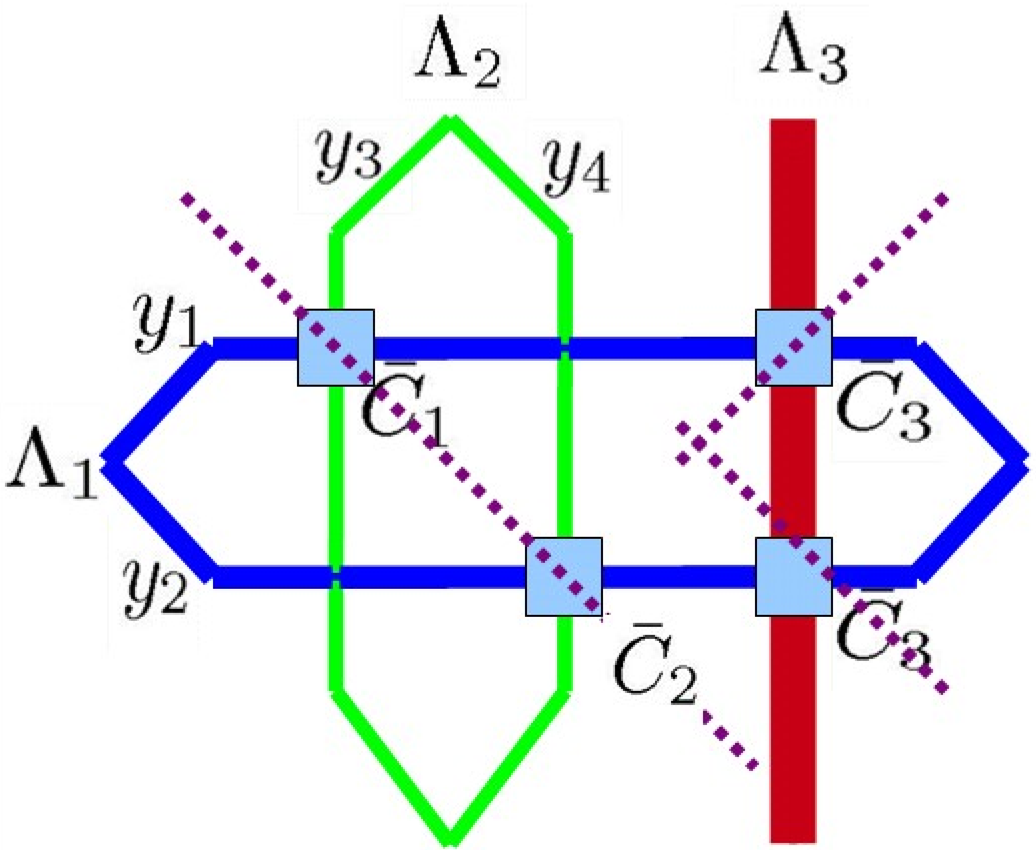
$$\sum_{i: j \in i} y_i \le \bar{C}_j, \quad j \in \bar{\mathcal{J}}$$

$$\sum_{i \in s} y_i = \Lambda_s, \quad s \in \mathcal{S}$$

$$\Longleftrightarrow \quad \sum_i a_{ji} \Lambda_i \le C_j, \quad j \in \mathcal{J}$$

A network

(Kang, Kelly, Lee, Williams '09)



$$\begin{pmatrix} 1 & 1 & 0 \\ \frac{1}{2} & 0 & 1 \end{pmatrix} \begin{pmatrix} \Lambda_1 \\ \Lambda_2 \\ \Lambda_3 \end{pmatrix} \le \begin{pmatrix} \bar{C}_1 + \bar{C}_2 \\ \bar{C}_3 \end{pmatrix}$$

In general

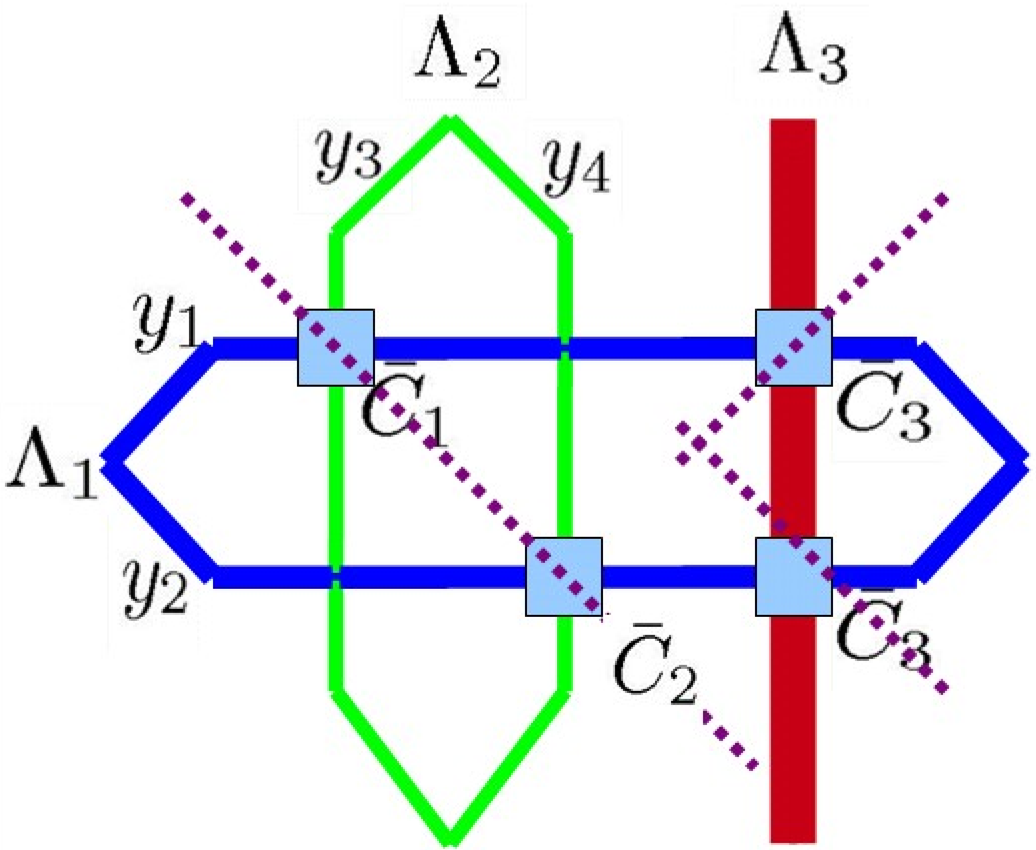
$$\sum_{i:j\in i} y_i \le \bar{C}_j, \quad j \in \bar{\mathcal{J}}$$

$$\sum_{i\in s} y_i = \Lambda_s, \quad s \in \mathcal{S}$$

$$\Longleftrightarrow \quad \sum_i a_{ji}\Lambda_i \le C_j, \quad j \in \mathcal{J}$$

Set of Resource pools

A network (Kang, Kelly, Lee, Williams '09)



In general

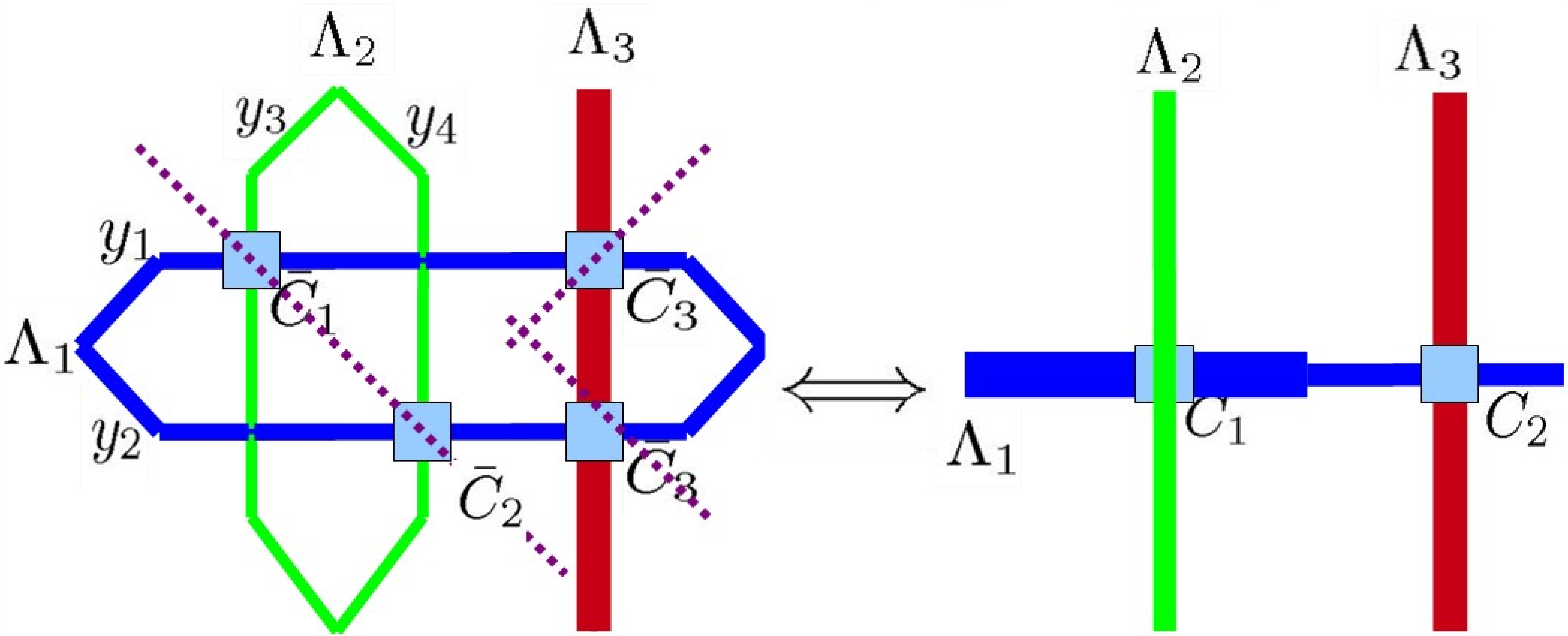
$$\sum_{i:j\in i} y_i \le \bar{C}_j, \quad j \in \bar{\mathcal{J}}$$

$$\sum_{i\in s} y_i = \Lambda_s, \quad s \in \mathcal{S}$$

$$\iff \sum_i a_{ji}\Lambda_i \le C_j, \quad j \in \mathcal{J}$$

A network

(Kang, Kelly, Lee, Williams '09)



In general

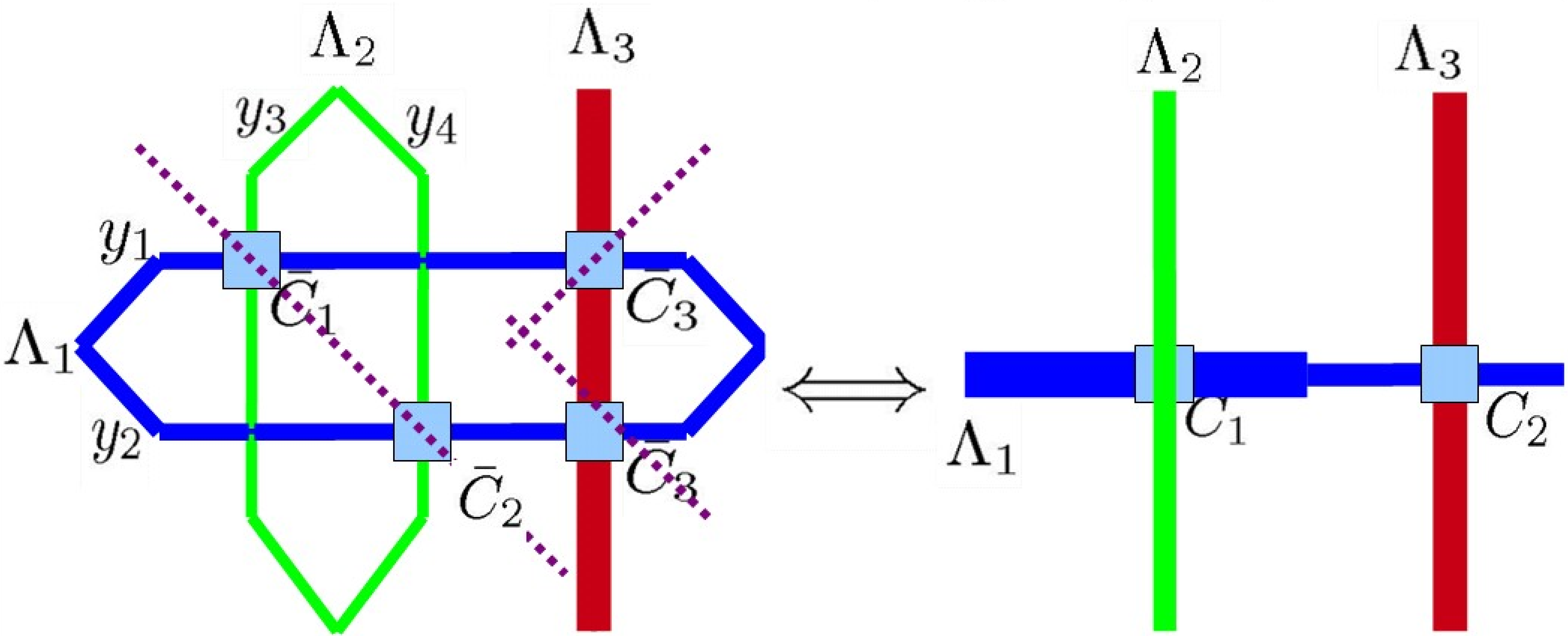
$$\sum_{i:j\in i} y_i \leq \bar{C}_j, \quad j \in \bar{\mathcal{J}}$$

$$\sum_{i\in s} y_i = \Lambda_s, \quad s \in \mathcal{S}$$

$$\Longleftrightarrow \quad \sum_i a_{ji}\Lambda_i \leq C_j, \quad j \in \mathcal{J}$$

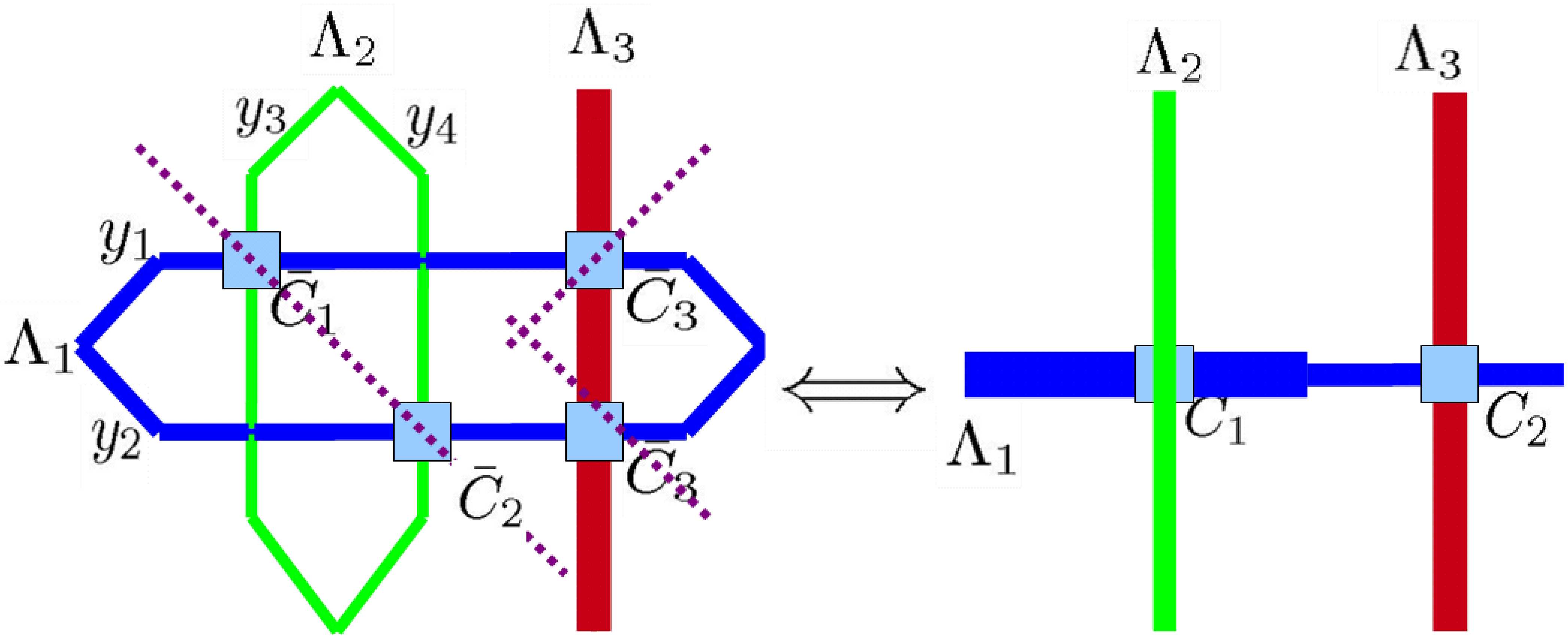
A network

(Kang, Kelly, Lee, Williams '09)



A network

(Kang, Kelly, Lee, Williams '09)



So multi-path routing is the same as single path routing when we pool resources

Proportional fairness

Proportional fairness

$$\text{subject to} \quad \sum_i a_{ji}\Lambda_i \leq C_j, \quad j \in \mathcal{J}$$

Proportional fairness

$$\max \quad \sum_i n_i \log \Lambda_i$$

$$\text{subject to} \quad \sum_i a_{ji} \Lambda_i \leq C_j, \quad j \in \mathcal{J}$$

Proportional fairness

$$\begin{array}{ll} \max & \displaystyle\sum_i n_i \log \Lambda_i \\ \text{subject to} & \displaystyle\sum_i a_{ji}\Lambda_i \le C_j, \quad j \in \mathcal{J} \\ \text{over} & \Lambda_i \ge 0,\ i \in \mathcal{I}. \end{array}$$

Proportional fairness

$$
\begin{aligned}
\text{max} \quad & \sum_i n_i \log \Lambda_i \\
\text{subject to} \quad & \sum_i a_{ji} \Lambda_i \le C_j, \quad j \in \mathcal{J} \\
\text{over} \quad & \Lambda_i \ge 0, \ i \in \mathcal{I}.
\end{aligned}
$$

The pooling of resources is not particular to proportional fairness

Proportional fairness

$$\begin{array}{ll} \max & \displaystyle\sum_i n_i \log \Lambda_i \\ \text{subject to} & \displaystyle\sum_i a_{ji}\Lambda_i \le C_j, \quad j \in \mathcal{J} \\ \text{over} & \Lambda_i \ge 0, \; i \in \mathcal{I}. \end{array}$$

The pooling of resources is not particular to proportional fairness

But

Proportional fairness does have some special properties…

A **very** imprecise thought:

A **very** imprecise thought:

"Proportional fairness is the network version of processor sharing"

A **very** imprecise thought:

"Proportional fairness is the network version of processor sharing"

The advantage of processor sharing queues:

A **very** imprecise thought:

"Proportional fairness is the network version of processor sharing"

The advantage of processor sharing queues:

$$\mathbb{E}\left[\, S \mid x \,\right] = \frac{x}{C - \rho}$$

A **very** imprecise thought:

"Proportional fairness is the network version of processor sharing"

The advantage of processor sharing queues:

Expected processing time of a job of size *x*

$$\mathbb{E}\left[\, S \mid x \,\right] = \frac{x}{C - \rho}$$

A **very** imprecise thought:

"Proportional fairness is the network version of processor sharing"

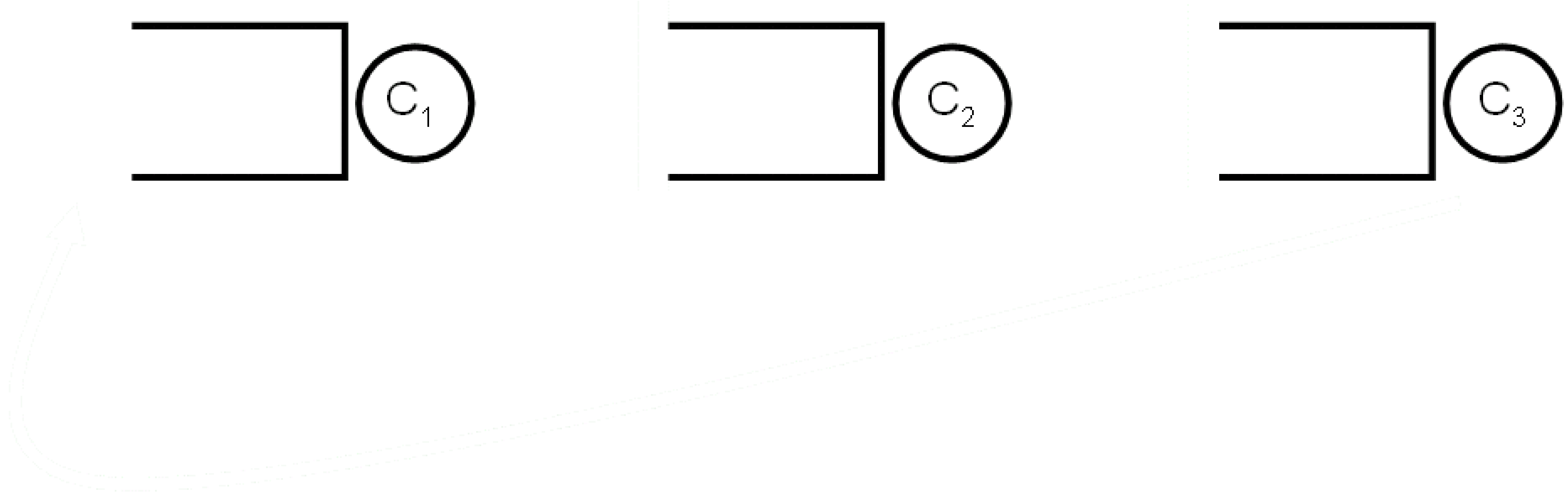
The advantage of processor sharing queues:

Expected processing time of a job of size *x*

Expected processing time of a job of size *x*

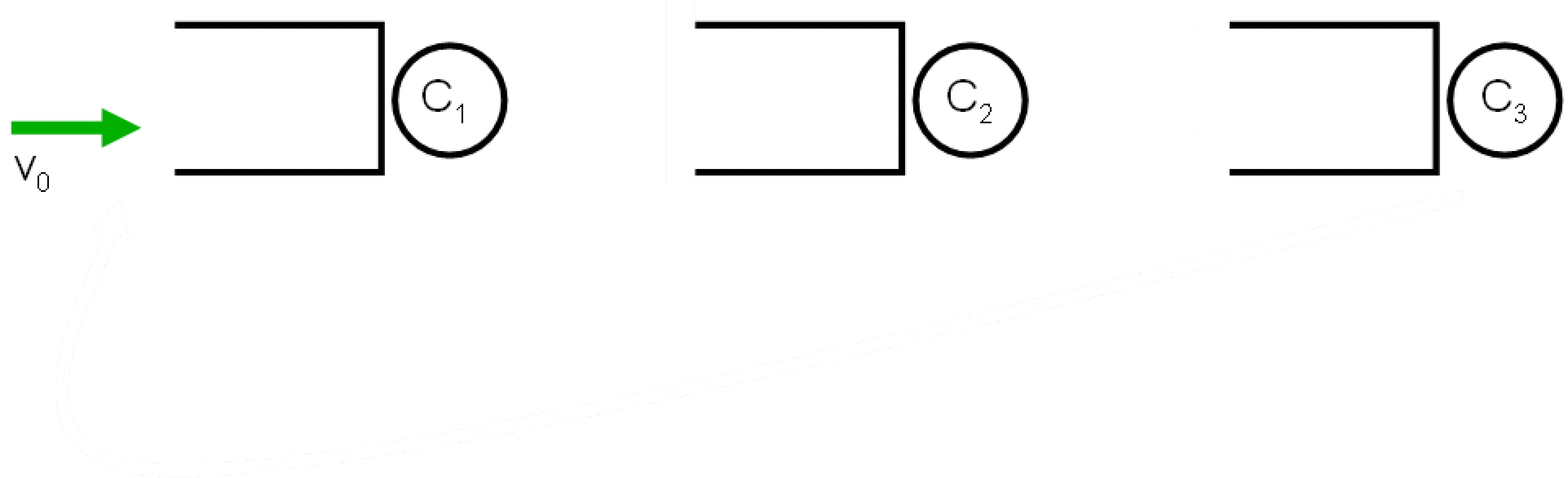
$$\mathbb{E}\left[\, S \mid x \,\right] = \frac{x}{C - \rho}$$

A Multi-class Network of Processor Sharing Queues



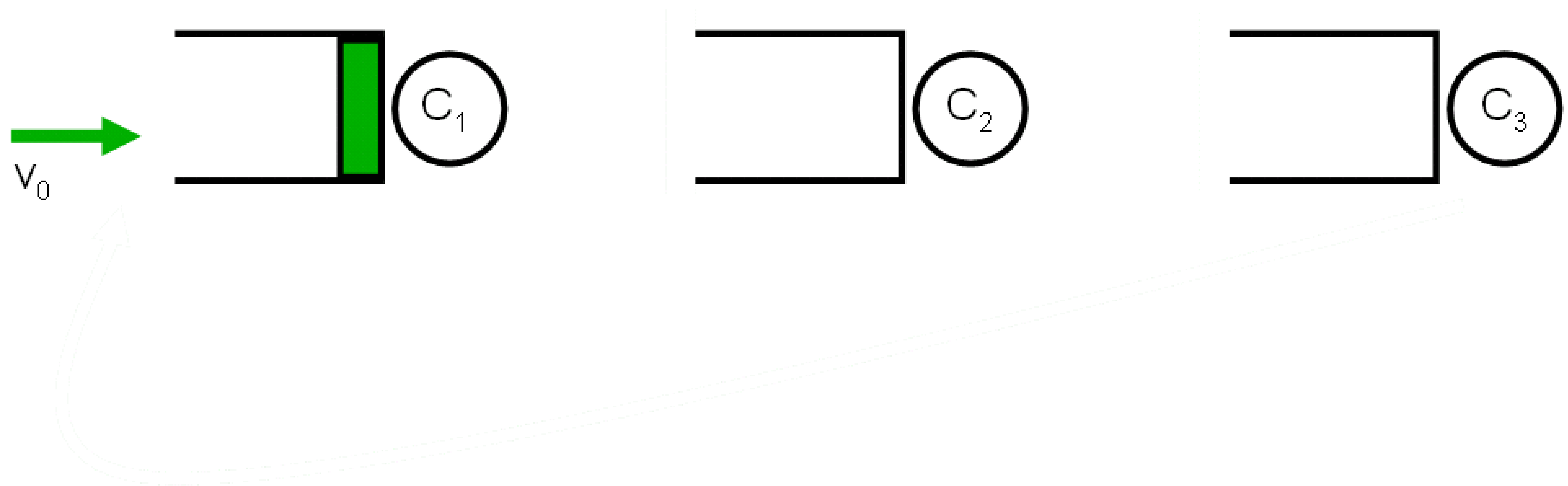
REF: Kelly '79, Massoulie '99, Proutiere '03, Bonald and Proutiere '04. W '09.

A Multi-class Network of Processor Sharing Queues



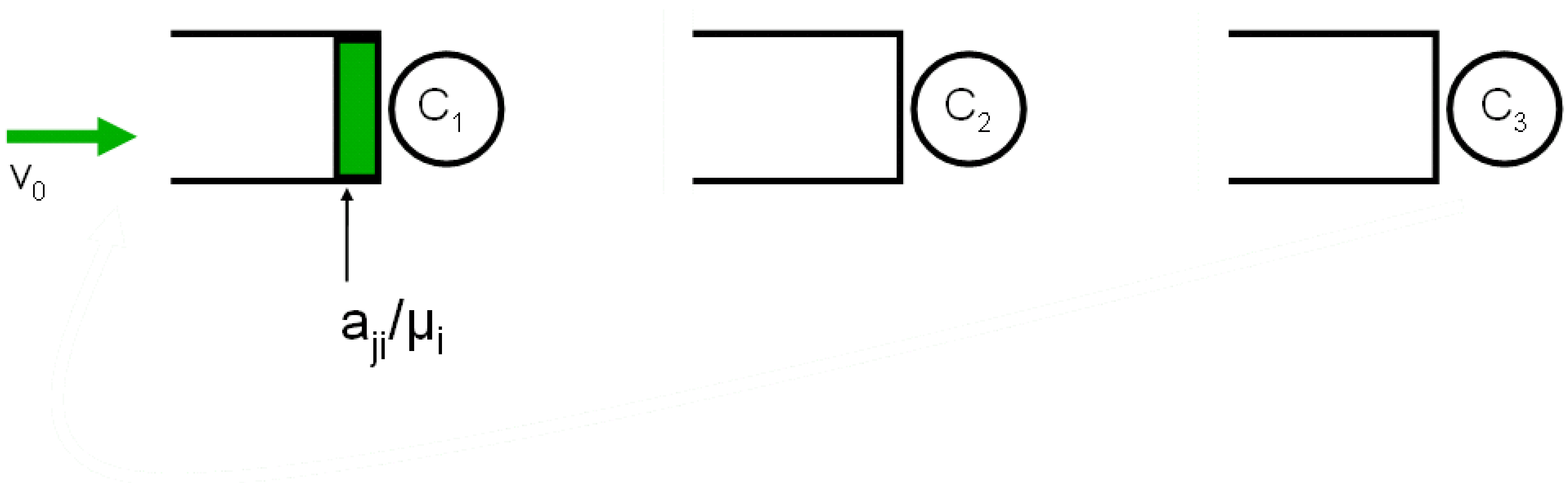
REF: Kelly '79, Massoulie '99, Proutiere '03, Bonald and Proutiere '04. W '09.

A Multi-class Network of Processor Sharing Queues



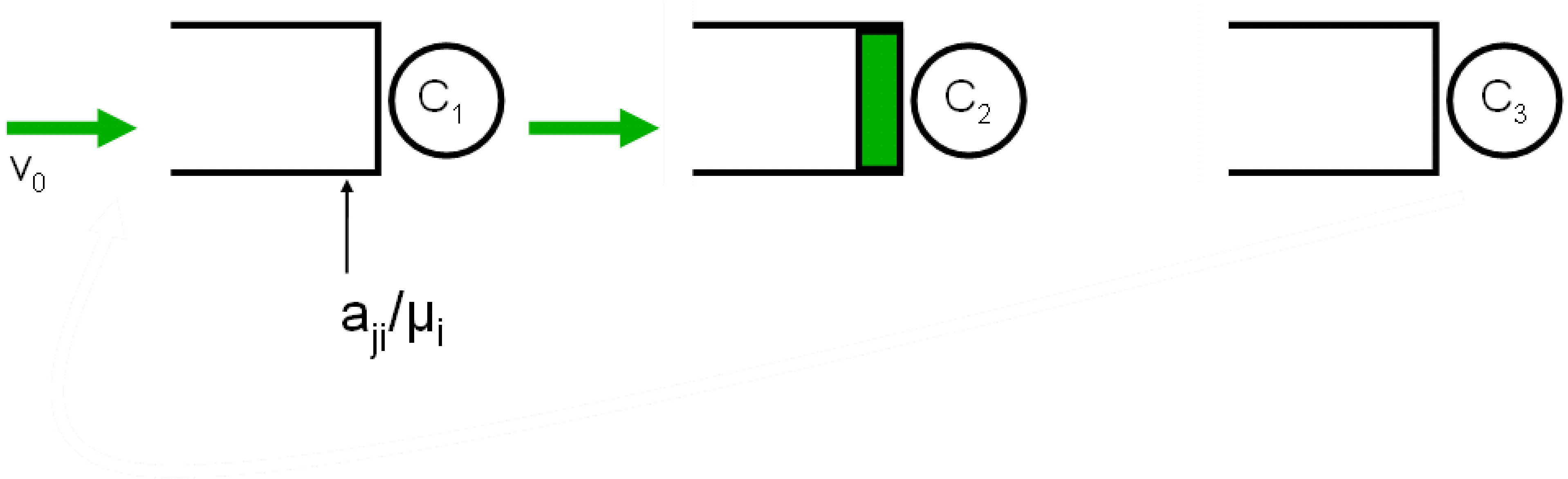
REF: Kelly '79, Massoulie '99, Proutiere '03, Bonald and Proutiere '04. W '09.

A Multi-class Network of Processor Sharing Queues



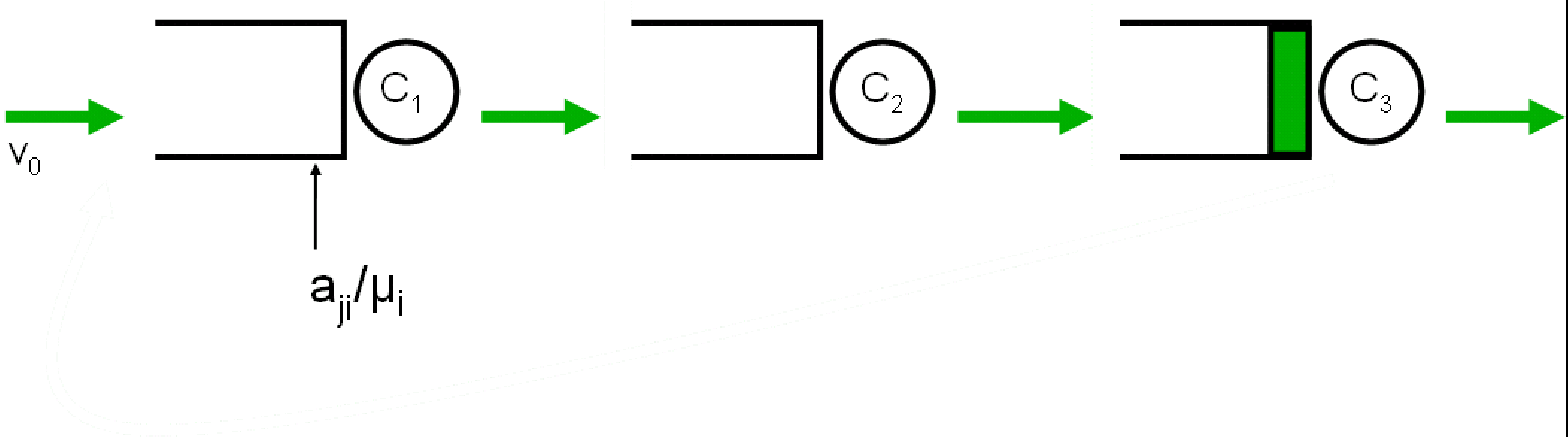
REF: Kelly '79, Massoulie '99, Proutiere '03, Bonald and Proutiere '04. W '09.

A Multi-class Network of Processor Sharing Queues



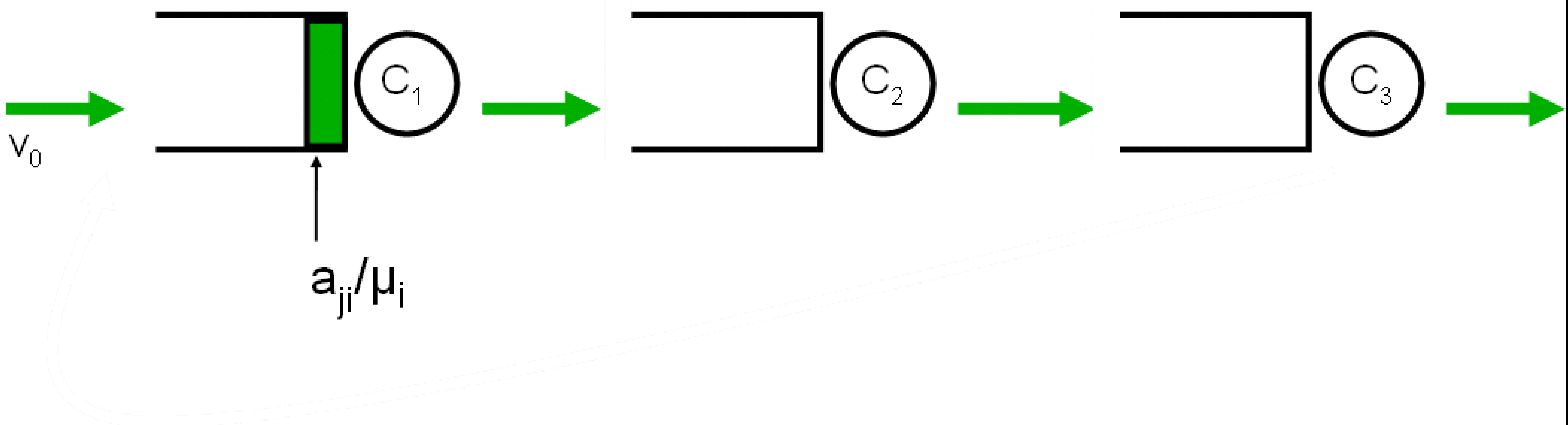
REF: Kelly '79, Massoulie '99, Proutiere '03, Bonald and Proutiere '04. W '09.

A Multi-class Network of Processor Sharing Queues



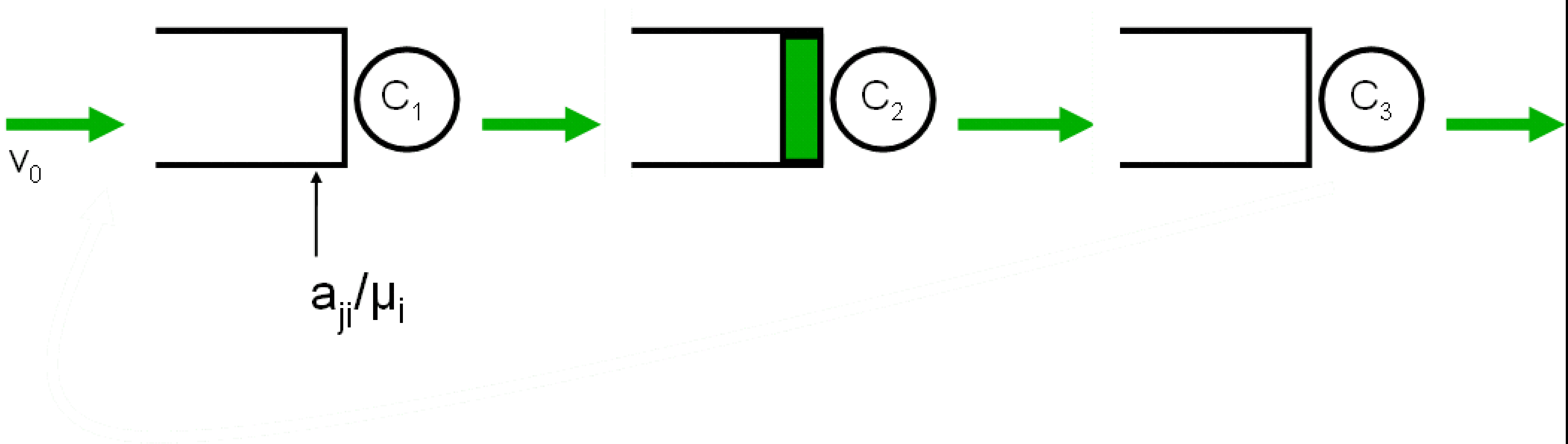
REF: Kelly '79, Massoulie '99, Proutiere '03, Bonald and Proutiere '04. W '09.

A Multi-class Network of Processor Sharing Queues



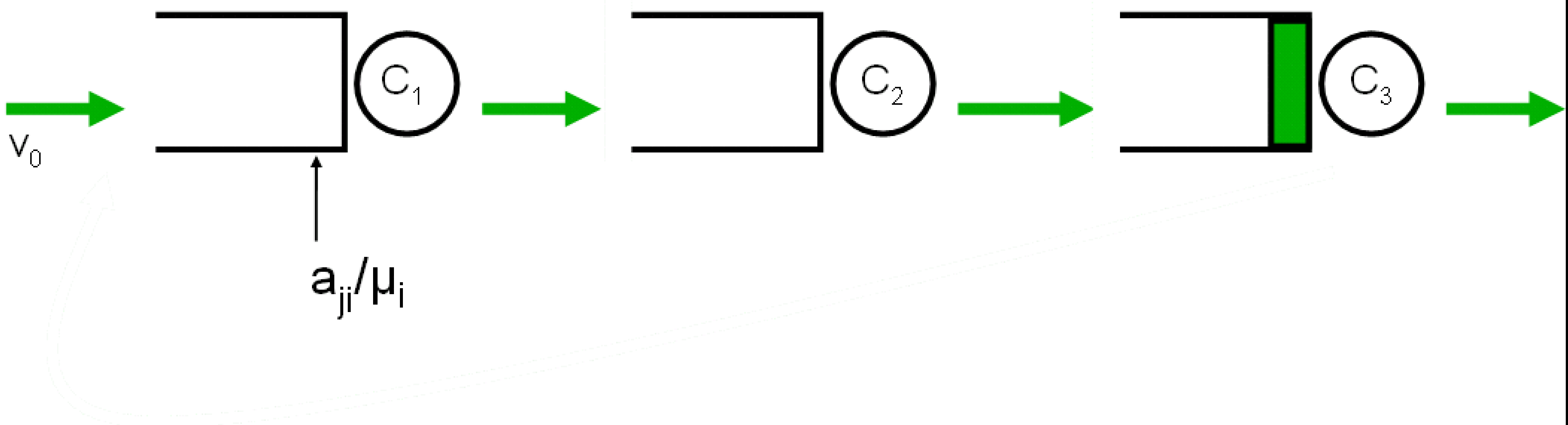
REF: Kelly '79, Massoulie '99, Proutiere '03, Bonald and Proutiere '04. W '09.

A Multi-class Network of Processor Sharing Queues



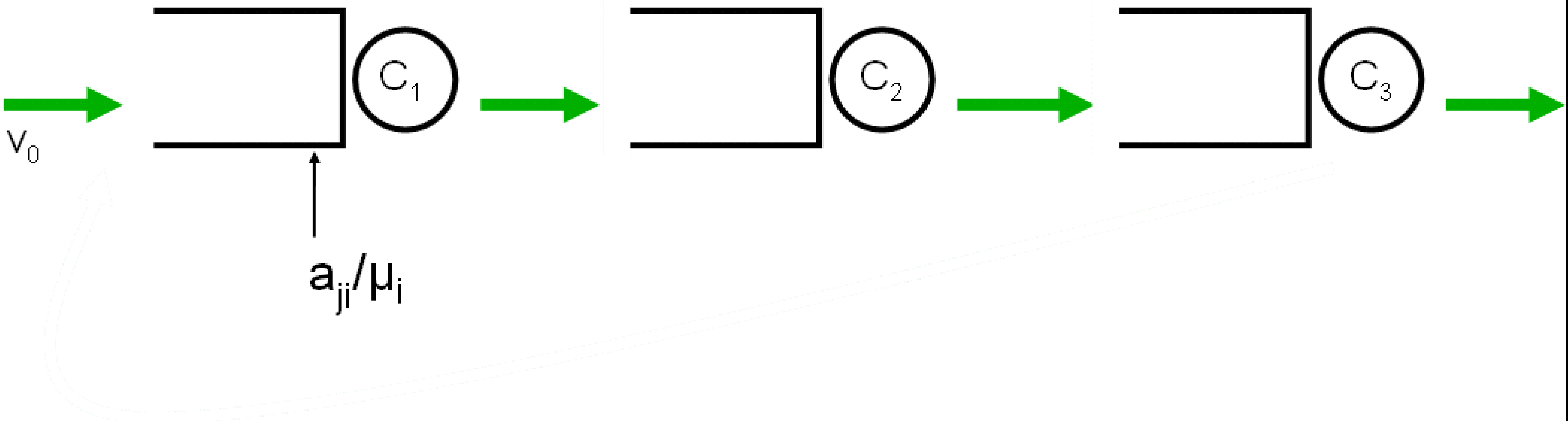
REF: Kelly '79, Massoulie '99, Proutiere '03, Bonald and Proutiere '04. W '09.

A Multi-class Network of Processor Sharing Queues



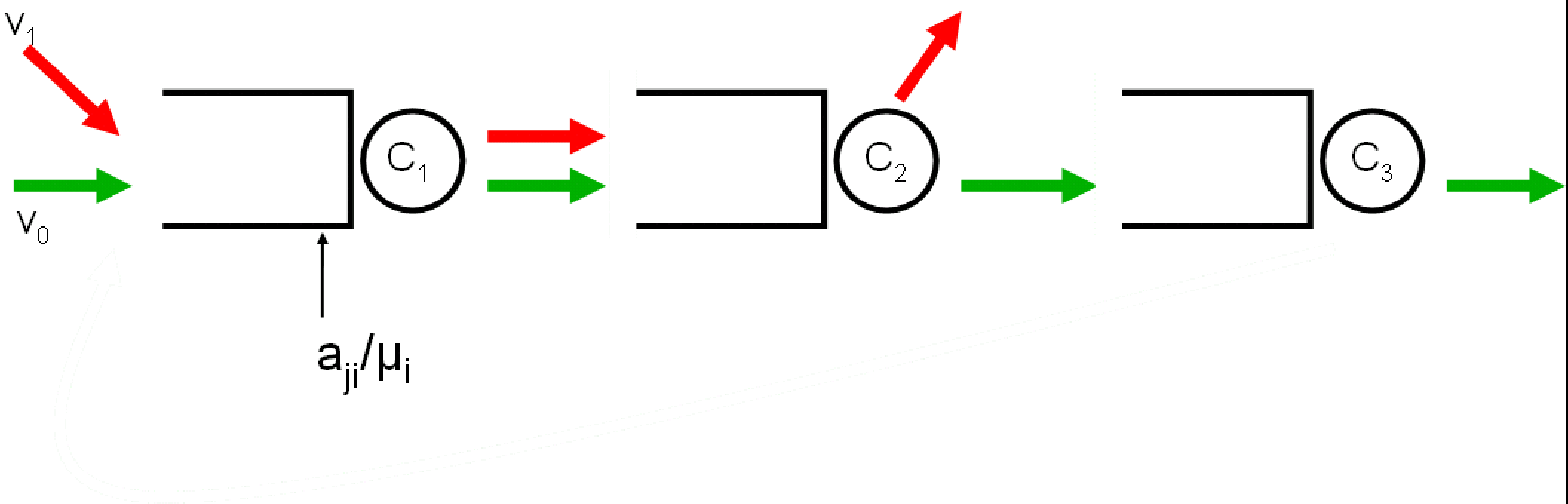
REF: Kelly '79, Massoulie '99, Proutiere '03, Bonald and Proutiere '04. W '09.

A Multi-class Network of Processor Sharing Queues



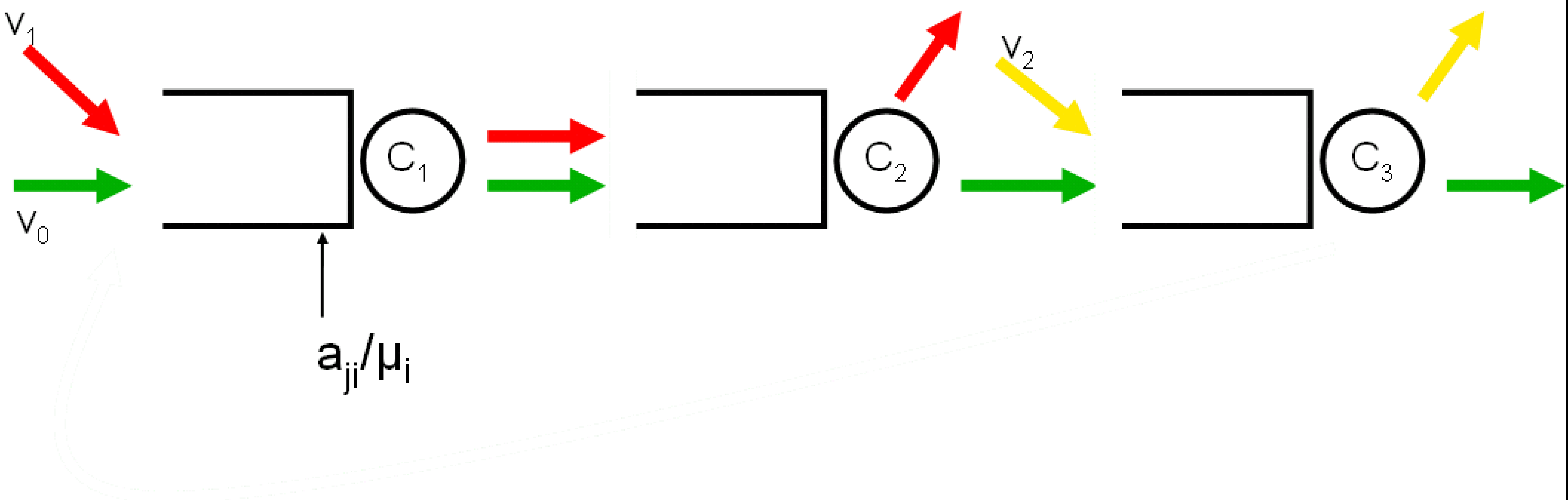
REF: Kelly '79, Massoulie '99, Proutiere '03, Bonald and Proutiere '04. W '09.

A Multi-class Network of Processor Sharing Queues



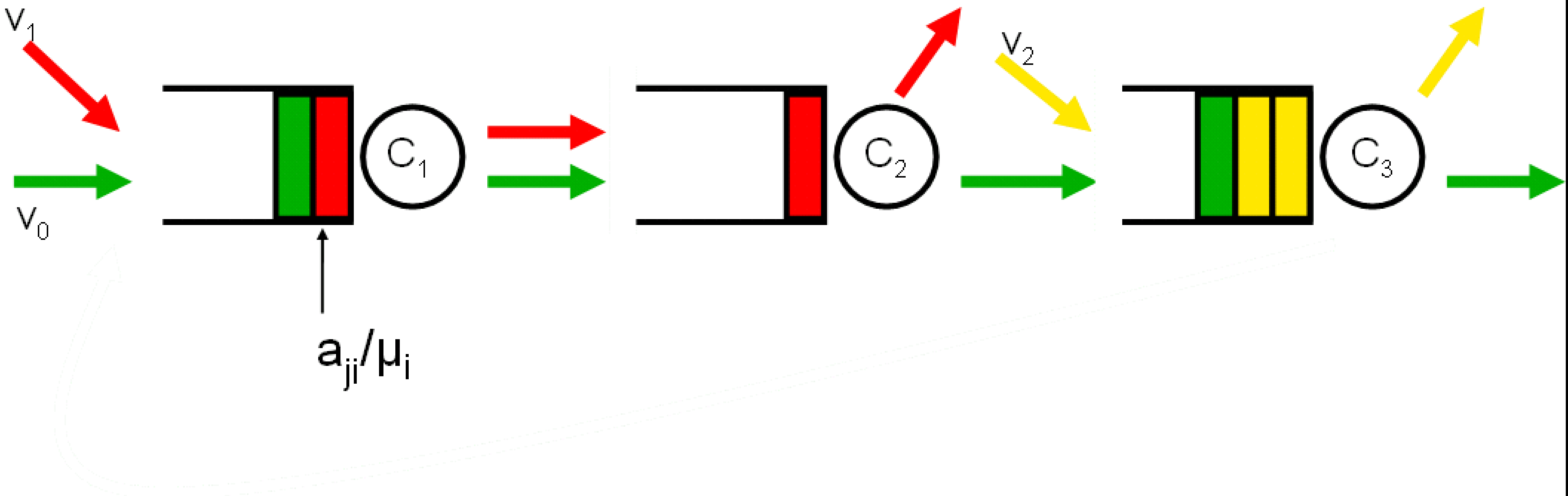
REF: Kelly '79, Massoulie '99, Proutiere '03, Bonald and Proutiere '04. W '09.

A Multi-class Network of Processor Sharing Queues



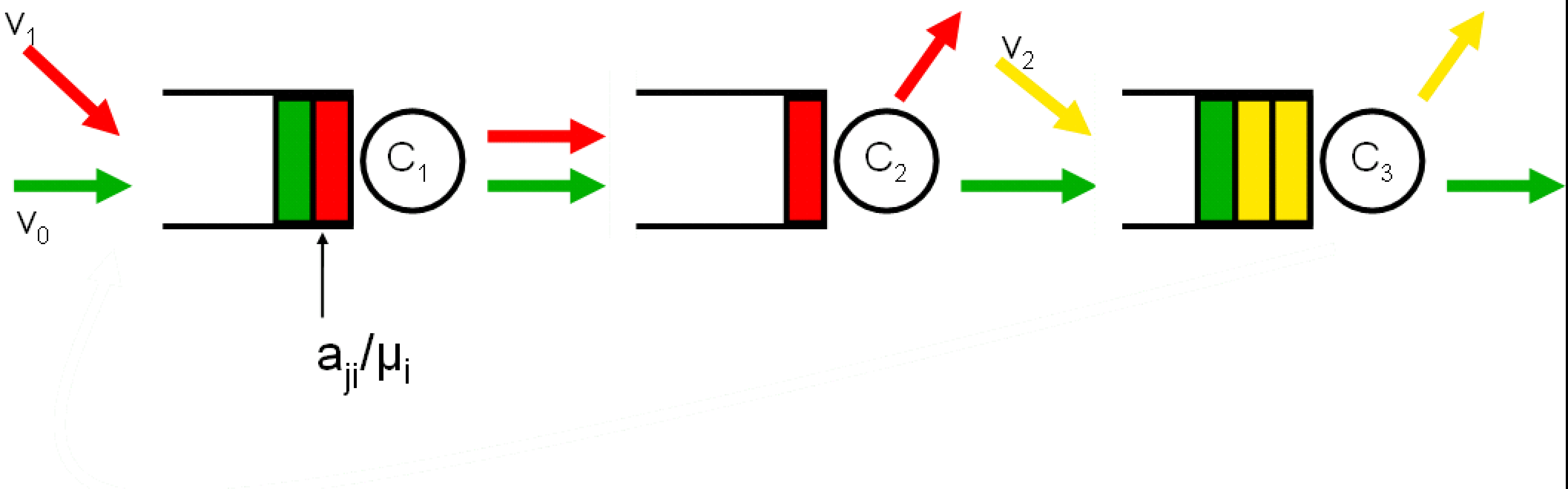
REF: Kelly '79, Massoulie '99, Proutiere '03, Bonald and Proutiere '04. W '09.

A Multi-class Network of Processor Sharing Queues



REF: Kelly '79, Massoulie '99, Proutiere '03, Bonald and Proutiere '04. W '09.

A Multi-class Network of Processor Sharing Queues



IMPORTANT POINT:

Queue sizes are independent Geometric Distributions

REF: Kelly '79, Massoulie '99, Proutiere '03, Bonald and Proutiere '04. W '09.

A Closed Multi-class Network of Processor Sharing Queues

A Closed Multi-class Network of Processor Sharing Queues

This argument is due to: Schweitzer ’79, Kelly ’89, Roberts and Massoulie ’99

A Closed Multi-class Network of Processor Sharing Queues

This argument is due to: Schweitzer ’79, Kelly ’89, Roberts and Massoulie ’99

By Little’s Law:

$$\Lambda_i q_j = \bar{m}_{ji}$$

A Closed Multi-class Network of Processor Sharing Queues

This argument is due to: Schweitzer '79, Kelly '89, Roberts and Massoulie '99

By Little's Law:

$$\Lambda_i q_j = \bar{m}_{ji}$$

arrival rate
route *i*

A Closed Multi-class Network of Processor Sharing Queues

This argument is due to: Schweitzer '79, Kelly '89, Roberts and Massoulie '99

By Little's Law:

$$\Lambda_i q_j = \bar{m}_{ji}$$

arrival rate route *i*

sojourn time

A Closed Multi-class Network of Processor Sharing Queues

This argument is due to: Schweitzer '79, Kelly '89, Roberts and Massoulie '99

By Little's Law:

$$\Lambda_i q_j = \bar{m}_{ji}$$

arrival rate route *i*

sojourn time

#route *i* packets at queue *j*

A Closed Multi-class Network of Processor Sharing Queues

This argument is due to: Schweitzer '79, Kelly '89, Roberts and Massoulie '99

By Little's Law:

$$\Lambda_i q_j = \bar{m}_{ji}$$

Summing over queues, *j*:

A Closed Multi-class Network of Processor Sharing Queues

This argument is due to: Schweitzer '79, Kelly '89, Roberts and Massoulie '99

By Little's Law:

$$\Lambda_i q_j = \bar{m}_{ji}$$

Summing over queues, *j*:

$$\frac{n_i}{\Lambda_i} - \sum_{j \in i} q_j = 0$$

A Closed Multi-class Network of Processor Sharing Queues

This argument is due to: Schweitzer '79, Kelly '89, Roberts and Massoulie '99

By Little's Law:

$$\Lambda_i q_j = \bar{m}_{ji}$$

Summing over queues, *j*:

$$\frac{n_i}{\Lambda_i} - \sum_{j \in i} q_j = 0$$

Since queues are stable therefore:

$$\sum_{i : j \in i} \Lambda_i \leq C_j$$

A Closed Multi-class Network of Processor Sharing Queues

This argument is due to: Schweitzer '79, Kelly '89, Roberts and Massoulie '99

By Little's Law:

$$\Lambda_i q_j = \bar{m}_{ji}$$

Summing over queues, *j*:

$$\frac{n_i}{\Lambda_i} - \sum_{j \in i} q_j = 0$$

Since queues are stable therefore:

$$\sum_{i: j \in i} \Lambda_i \leq C_j$$

If very stable then sojourn is small:

$$q_j \Big(C_j - \sum_{i: j \in i} \Lambda_i \Big) = 0$$

A Closed Multi-class Network of Processor Sharing Queues

This argument is due to: Schweitzer '79, Kelly '89, Roberts and Massoulie '99

By Little's Law:

$$\Lambda_i q_j = \bar{m}_{ji}$$

Summing over queues, *j*:

$$\frac{n_i}{\Lambda_i} - \sum_{j \in i} q_j = 0$$

Since queues are stable therefore:

$$\sum_{i : j \in i} \Lambda_i \leq C_j$$

If very stable then sojourn is small:

$$q_j \Big(C_j - \sum_{i : j \in i} \Lambda_i \Big) = 0$$

A Closed Multi-class Network of Processor Sharing Queues

This argument is due to: Schweitzer '79, Kelly '89, Roberts and Massoulie '99

By Little's Law:

$$\Lambda_i q_j = \bar{m}_{ji}$$

Summing over queues, *j*:

$$\frac{n_i}{\Lambda_i} - \sum_{j \in i} q_j = 0$$

Since queues are stable therefore:

$$\sum_{i: j \in i} \Lambda_i \leq C_j$$

If very stable then sojourn is small:

$$q_j \Big(C_j - \sum_{i: j \in i} \Lambda_i \Big) = 0$$

These are the Kuhn-Tucker conditions for proportionally fair optimization!

A Multi-class Network of Processor Sharing Queues

A Multi-class Network of Processor Sharing Queues

A large deviations analysis is sufficient to show [W '09]:

A Multi-class Network of Processor Sharing Queues

A large deviations analysis is sufficient to show [W '09]:

$$\Lambda_i^{SN}(\lfloor hn \rfloor) \xrightarrow[h\to\infty]{} \Lambda_i^{PF}(n)$$

A Multi-class Network of Processor Sharing Queues

A large deviations analysis is sufficient to show [W '09]:

$$\Lambda_i^{SN}(\lfloor hn \rfloor) \xrightarrow[h\to\infty]{} \Lambda_i^{PF}(n)$$

Stationary throughput
of closed queueing network

A Multi-class Network of Processor Sharing Queues

A large deviations analysis is sufficient to show [W '09]:

$$\Lambda_i^{SN}(\lfloor hn \rfloor) \xrightarrow[h\to\infty]{} \Lambda_i^{PF}(n)$$

Stationary throughput of closed queueing network

Proportionally fair allocation

A Multi-class Network of Processor Sharing Queues

A large deviations analysis is sufficient to show [W '09]:

$$\Lambda_i^{SN}(\lfloor hn \rfloor) \xrightarrow[h\to\infty]{} \Lambda_i^{PF}(n)$$

A Multi-class Network of Processor Sharing Queues

A large deviations analysis is sufficient to show [W '09]:

$$\Lambda_i^{SN}(\lfloor hn \rfloor) \xrightarrow[h\to\infty]{} \Lambda_i^{PF}(n)$$

Suggests product form results
associated with proportional fairness

A Multi-class Network of Processor Sharing Queues

A large deviations analysis is sufficient to show [W '09]:

$$\Lambda_i^{SN}(\lfloor hn \rfloor) \xrightarrow[h\to\infty]{} \Lambda_i^{PF}(n)$$

Suggests product form results associated with proportional fairness

This point had previously been considered for proportional fairness (Kang et al. '09)

A Multi-class Network of Processor Sharing Queues

A large deviations analysis is sufficient to show [W '09]:

$$\Lambda_i^{SN}(\lfloor hn \rfloor) \xrightarrow[h\to\infty]{} \Lambda_i^{PF}(n)$$

Suggests product form results associated with proportional fairness

This point had previously been considered for proportional fairness (Kang et al. '09)

Idea shadow prices q_j are like queue sizes and so are independent.

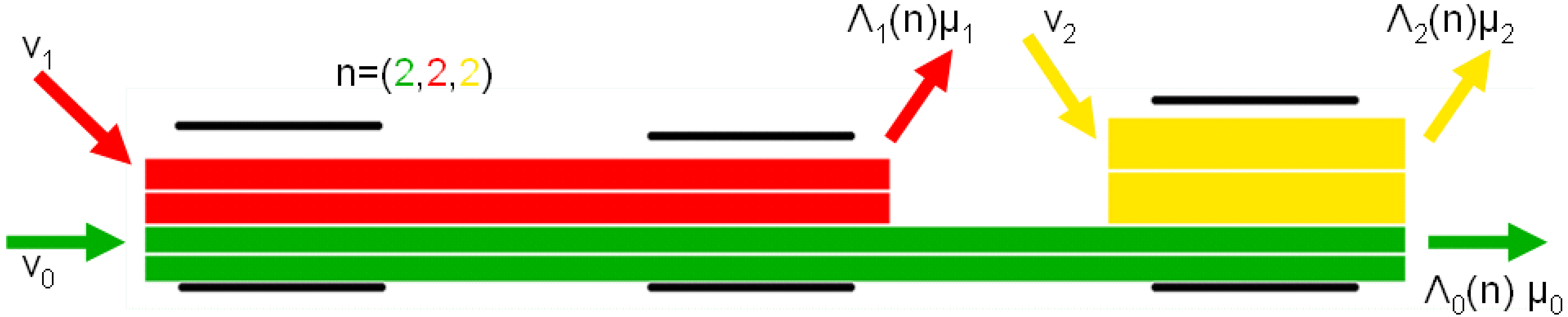
Proportional fairness considers flows and resources, not packets and queues
So we must use a different model: (Roberts and Massoulie '98)

$$\sum_i a_{ji} \frac{\nu_i}{\mu_i} \approx C_j, \quad j \in \mathcal{J}.$$

So are lagrange multipliers independent exponential distributions?

Well sometimes...we know Local Traffic condition (Kang et al.)

Proportional fairness considers flows and resources, not packets and queues
So we must use a different model: (Roberts and Massoulie '98)

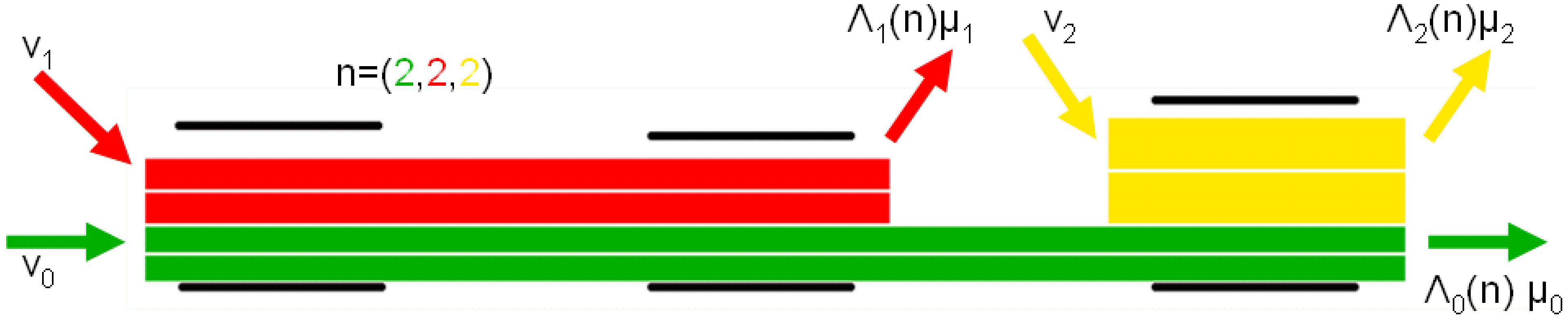


$$\sum_i a_{ji}\frac{\nu_i}{\mu_i} \approx C_j, \quad j \in \mathcal{J}.$$

So are lagrange multipliers independent exponential distributions?

Well sometimes...we know Local Traffic condition (Kang et al.)

Proportional fairness considers flows and resources, not packets and queues
So we must use a different model: (Roberts and Massoulie '98)



Proportional fair model is stable iff

De Veciana, Lee & Konstantopoulos 1999

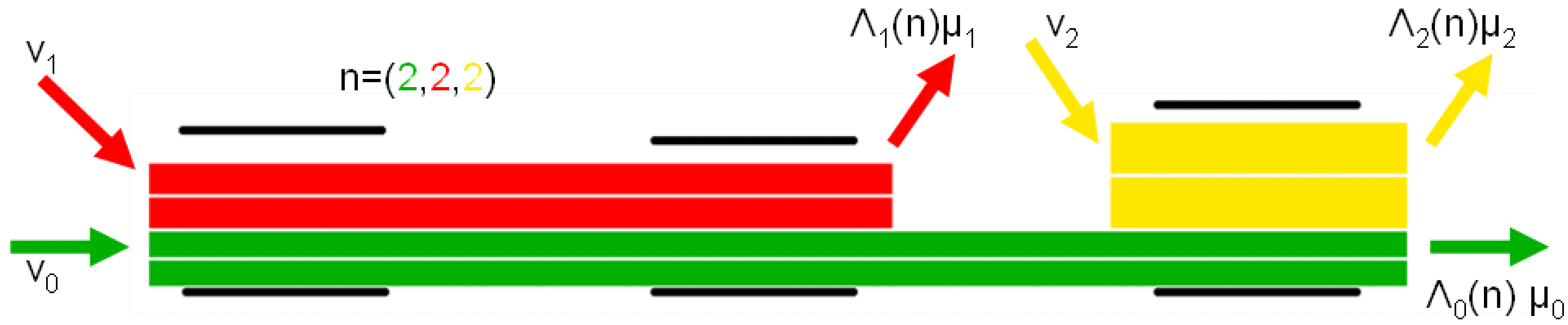
Bonald and Massoulié

$$\sum_i a_{ji} \frac{\nu_i}{\mu_i} \approx C_j, \quad j \in \mathcal{J}.$$

So are lagrange multipliers independent exponential distributions?

Well sometimes...we know Local Traffic condition (Kang et al.)

Proportional fairness considers flows and resources, not packets and queues
So we must use a different model: (Roberts and Massoulie '98)



Proportional fair model is stable iff

De Veciana, Lee & Konstantopoulos 1999
Bonald and Massoulié

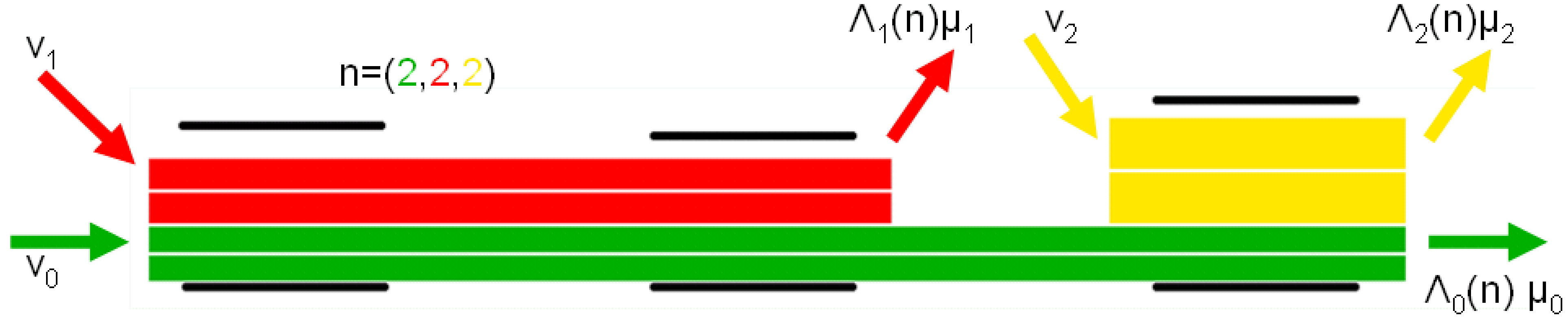
$$\sum_i a_{ji} \frac{\nu_i}{\mu_i} \approx C_j, \quad j \in \mathcal{J}.$$

So are lagrange multipliers independent exponential distributions?

Are lagrange multipliers independent geometric distributions?

Well sometimes…we know Local Traffic condition (Kang et al.)

Proportional fairness considers flows and resources, not packets and queues
So we must use a different model: (Roberts and Massoulie '98)



Proportional fair model is stable iff

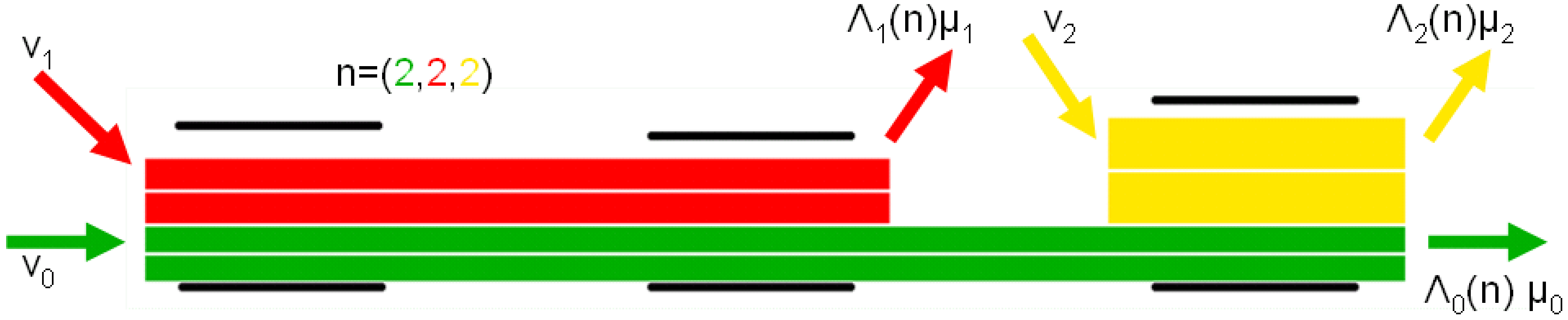
De Veciana, Lee & Konstantopoulos 1999
Bonald and Massoulié

$$\sum_i a_{ji} \frac{\nu_i}{\mu_i} \approx C_j, \quad j \in \mathcal{J}.$$

So are lagrange multipliers independent exponential distributions?

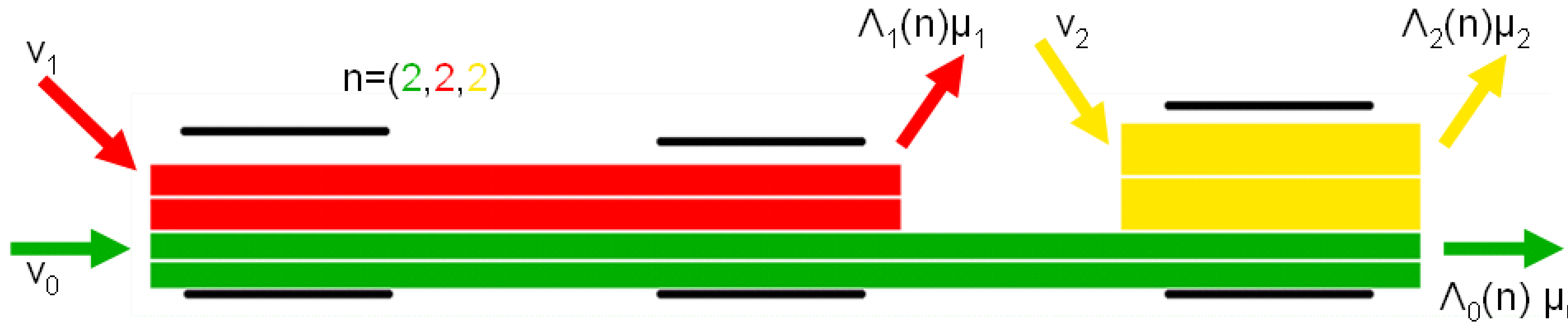
Are lagrange multipliers independent geometric distributions?
Well sometimes…we know Local Traffic condition (Kang et al.)
No

Proportional fairness considers flows and resources, not packets and queues
So we must use a different model: (Roberts and Massoulie '98)



$$\sum_i a_{ji} \frac{\nu_i}{\mu_i} \leq C_j, \quad j \in \mathcal{J}.$$

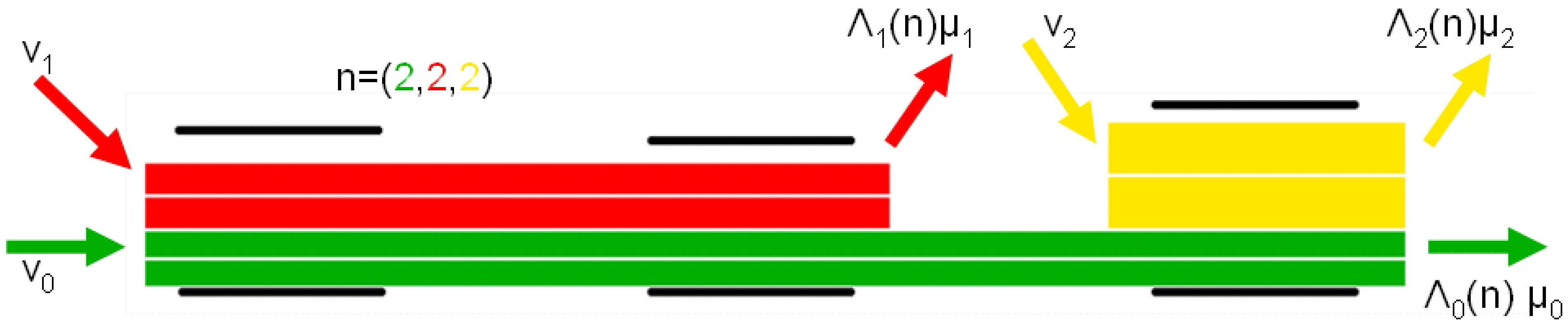
Proportional fairness considers flows and resources, not packets and queues
So we must use a different model: (Roberts and Massoulie '98)



What about Heavy Traffic?

$$\sum_i a_{ji} \frac{\nu_i}{\mu_i} \approx C_j, \quad j \in \mathcal{J}.$$

Proportional fairness considers flows and resources, not packets and queues
So we must use a different model: (Roberts and Massoulie '98)

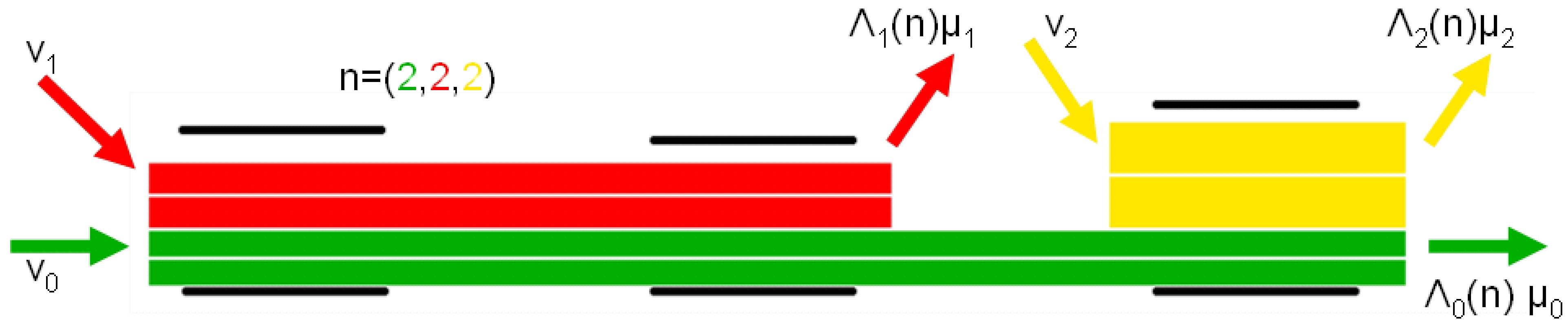


What about Heavy Traffic?

$$\sum_i a_{ji} \frac{\nu_i}{\mu_i} \approx C_j, \quad j \in \mathcal{J}.$$

So are lagrange multipliers independent exponential distributions?

Proportional fairness considers flows and resources, not packets and queues
So we must use a different model: (Roberts and Massoulie '98)



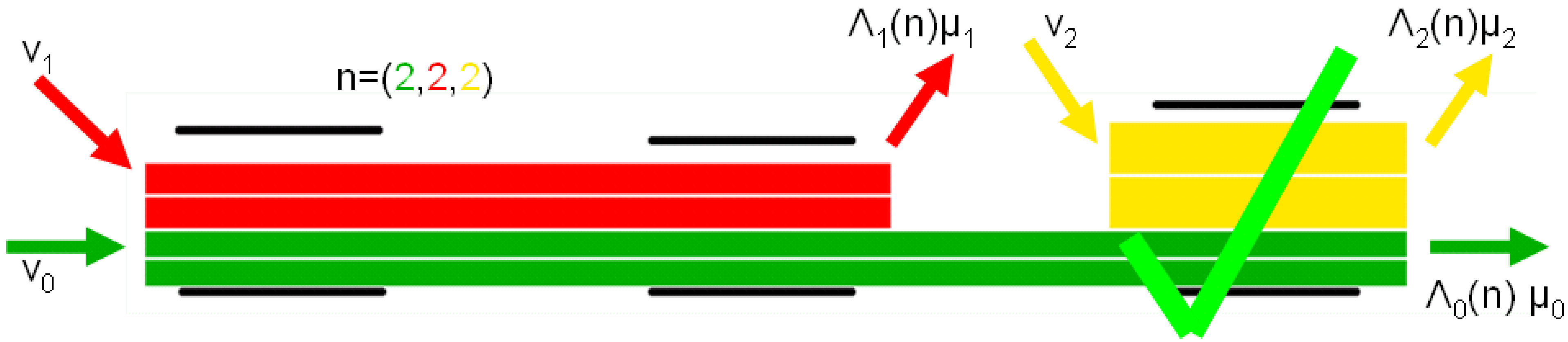
What about Heavy Traffic?

$$\sum_i a_{ji}\frac{\nu_i}{\mu_i} \approx C_j, \quad j \in \mathcal{J}.$$

So are lagrange multipliers independent exponential distributions?

Well sometimes...we know Local Traffic condition (Kang et al.)

Proportional fairness considers flows and resources, not packets and queues
So we must use a different model: (Roberts and Massoulie '98)



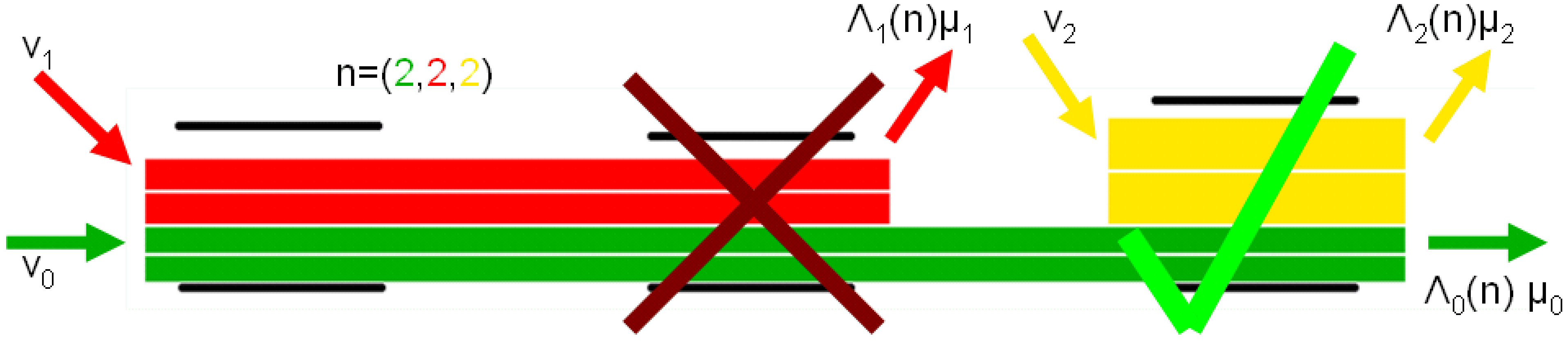
What about Heavy Traffic?

$$\sum_i a_{ji} \frac{\nu_i}{\mu_i} \approx C_j, \quad j \in \mathcal{J}.$$

So are lagrange multipliers independent exponential distributions?

Well sometimes...we know Local Traffic condition (Kang et al.)

Proportional fairness considers flows and resources, not packets and queues
So we must use a different model: (Roberts and Massoulie '98)



What about Heavy Traffic?

$$\sum_i a_{ji} \frac{\nu_i}{\mu_i} \approx C_j, \quad j \in \mathcal{J}.$$

So are lagrange multipliers independent exponential distributions?

Well sometimes...we know Local Traffic condition (Kang et al.)

What about in general for $\sum_i a_{ji} \frac{\nu_i}{\mu_i} \approx C_j, \quad j \in \mathcal{J}.$?

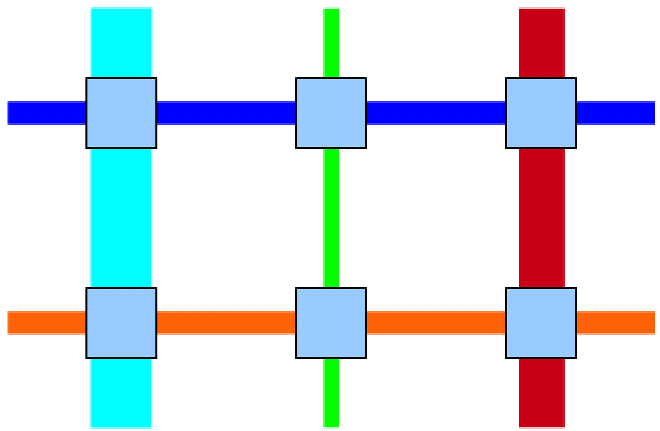
What about in general for $\sum_i a_{ji} \frac{\nu_i}{\mu_i} \approx C_j, \quad j \in \mathcal{J}.$?

Is not true independent exponentials in general...

What about in general for $\displaystyle\sum_i a_{ji} \frac{\nu_i}{\mu_i} \approx C_j, \quad j \in \mathcal{J}.$?

Is not true independent exponentials in general...

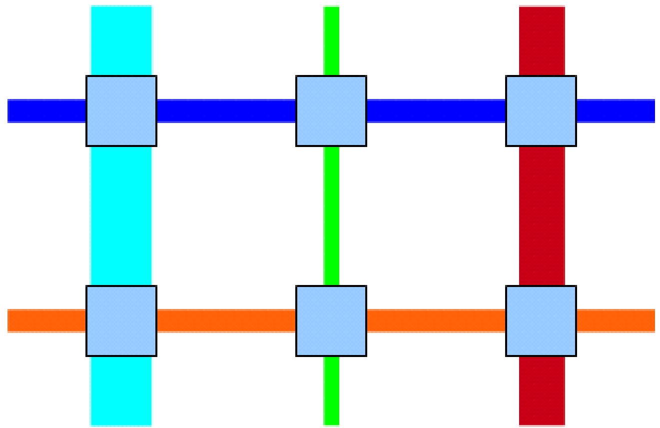
Grid networks



What about in general for $\sum_i a_{ji} \frac{\nu_i}{\mu_i} \approx C_j, \quad j \in \mathcal{J}.$?

Is not true independent exponentials in general...

Grid networks

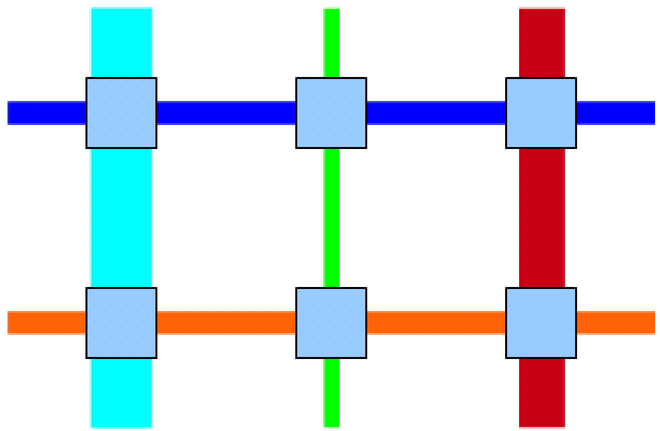


Total number in system **should** be Erlang(6)

What about in general for $\sum_i a_{ji} \frac{\nu_i}{\mu_i} \approx C_j, \quad j \in \mathcal{J}.$?

Is not true independent exponentials in general...

Grid networks



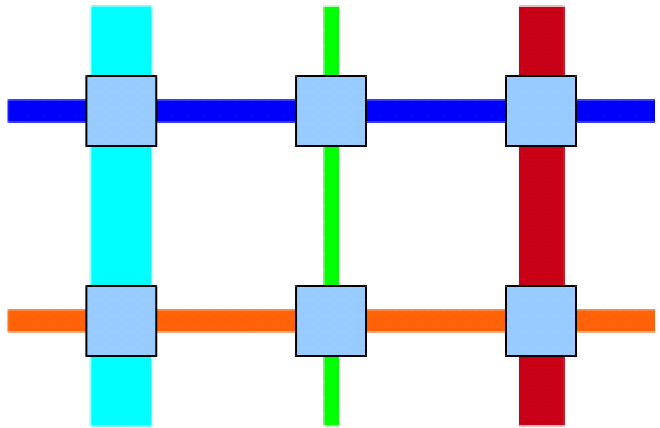
Total number in system **should** be Erlang(6)

Total number in system is **actually** Erlang(4)

What about in general for $\sum_i a_{ji} \frac{\nu_i}{\mu_i} \approx C_j, \quad j \in \mathcal{J}.$?

Is not true independent exponentials in general...

Grid networks



Total number in system **should** be Erlang(6)

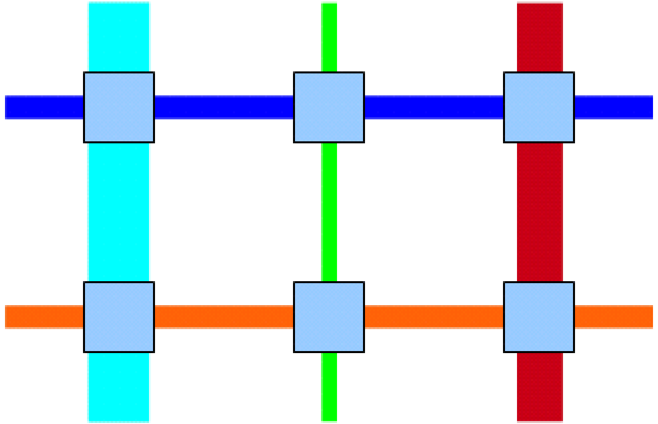
Total number in system is **actually** Erlang(4)

Suggests a simple structure

What about in general for $\sum_i a_{ji} \frac{\nu_i}{\mu_i} \approx C_j, \quad j \in \mathcal{J}.$?

Is not true independent exponentials in general...

Grid networks



Total number in system **should** be Erlang(6)

Total number in system is **actually** Erlang(4)

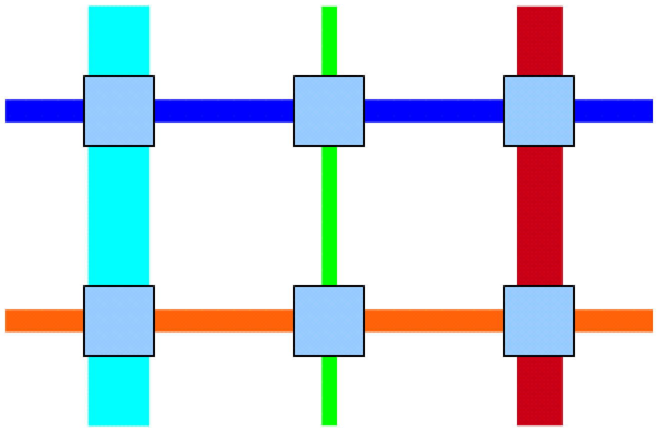
Suggests a simple structure

A conjecture: In Heavy traffic

What about in general for $\sum_i a_{ji} \frac{\nu_i}{\mu_i} \approx C_j, \quad j \in \mathcal{J}.$?

Is not true independent exponentials in general...

Grid networks



Total number in system **should** be Erlang(6)

Total number in system is **actually** Erlang(4)

Suggests a simple structure

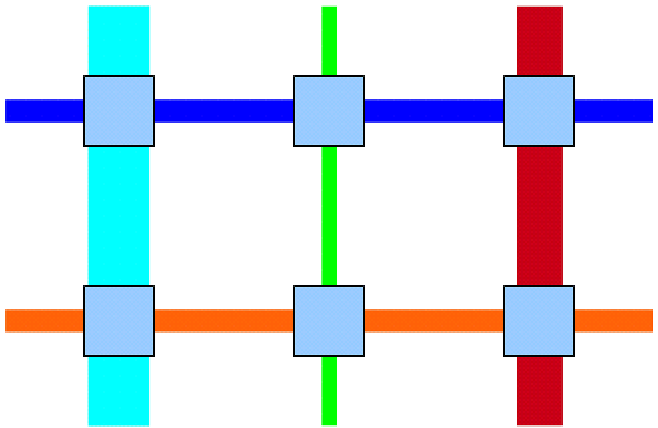
A conjecture: In Heavy traffic

$$\tilde{N}_s = \sum_{j \in \mathcal{J}^*} a_{js} \rho_s \tilde{Q}_j$$

What about in general for $\sum_i a_{ji} \frac{\nu_i}{\mu_i} \approx C_j, \quad j \in \mathcal{J}.$?

Is not true independent exponentials in general...

Grid networks



Total number in system **should** be Erlang(6)

Total number in system is **actually** Erlang(4)

Suggests a simple structure

A conjecture: In Heavy traffic

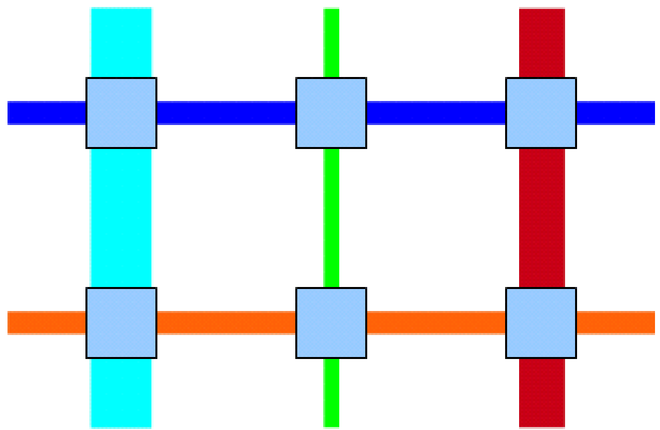
$$\tilde{N}_s = \sum_{j \in \mathcal{J}^*} a_{js} \rho_s \tilde{Q}_j$$

Where $\tilde{Q}_j$ has density,

What about in general for $\sum_i a_{ji} \frac{\nu_i}{\mu_i} \approx C_j, \quad j \in \mathcal{J}.$?

Is not true independent exponentials in general...

Grid networks



Total number in system **should** be Erlang(6)

Total number in system is **actually** Erlang(4)

Suggests a simple structure

A conjecture: In Heavy traffic

$$\tilde{N}_s = \sum_{j \in \mathcal{J}^*} a_{js} \rho_s \tilde{Q}_j$$

Where $\tilde{Q}_j$ has density,

$$p(q) = C' \mathbb{I}[q \in \mathcal{K}] e^{-\sum_{j \in \mathcal{J}^*} \sum_{s \in \mathcal{S}} q_j a_{js} \sigma_s}$$

Utility Optimization in Congested Queueing Networks

Neil Walton

Statistical Laboratory, University of Cambridge.

A Decomposition Theorem

A Decomposition Theorem

F.P. Kelly, “Charging and Rate Control of Elastic Traffic” (1997)

A Decomposition Theorem

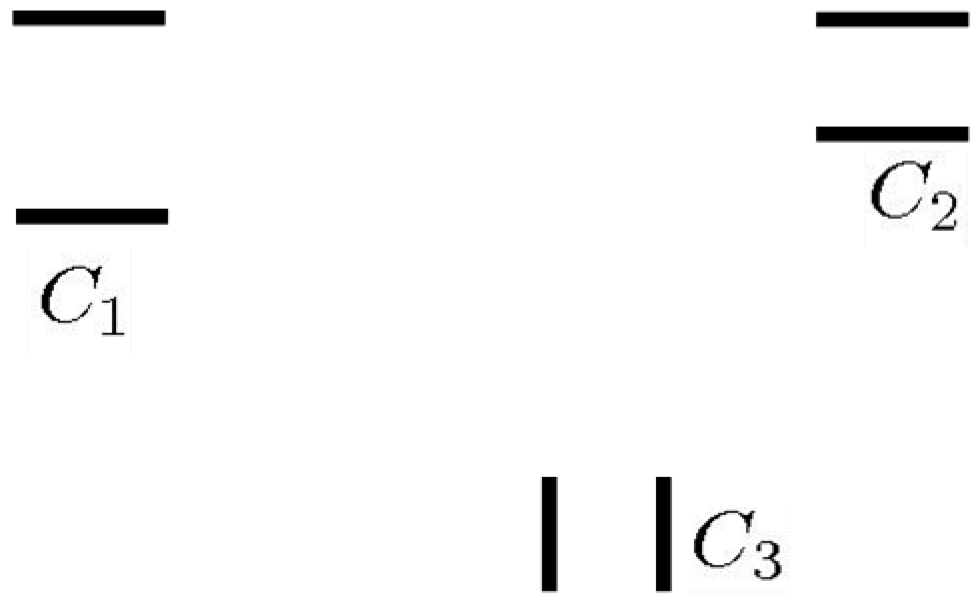
F.P. Kelly, “Charging and Rate Control of Elastic Traffic” (1997)

A NETWORK

A Decomposition Theorem

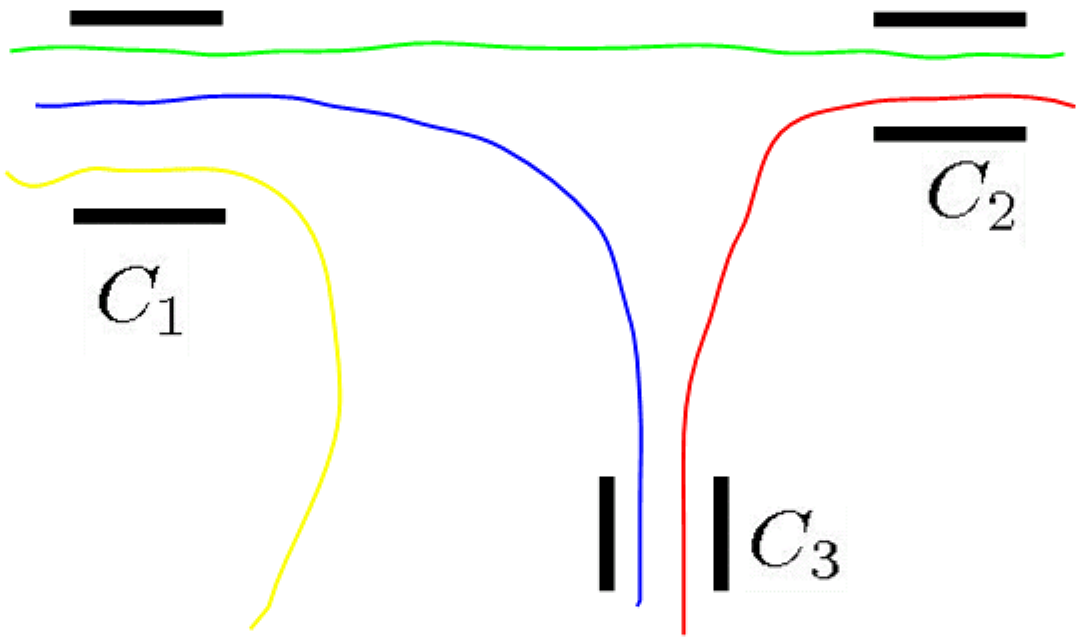
F.P. Kelly, “Charging and Rate Control of Elastic Traffic” (1997)

A NETWORK



A Decomposition Theorem

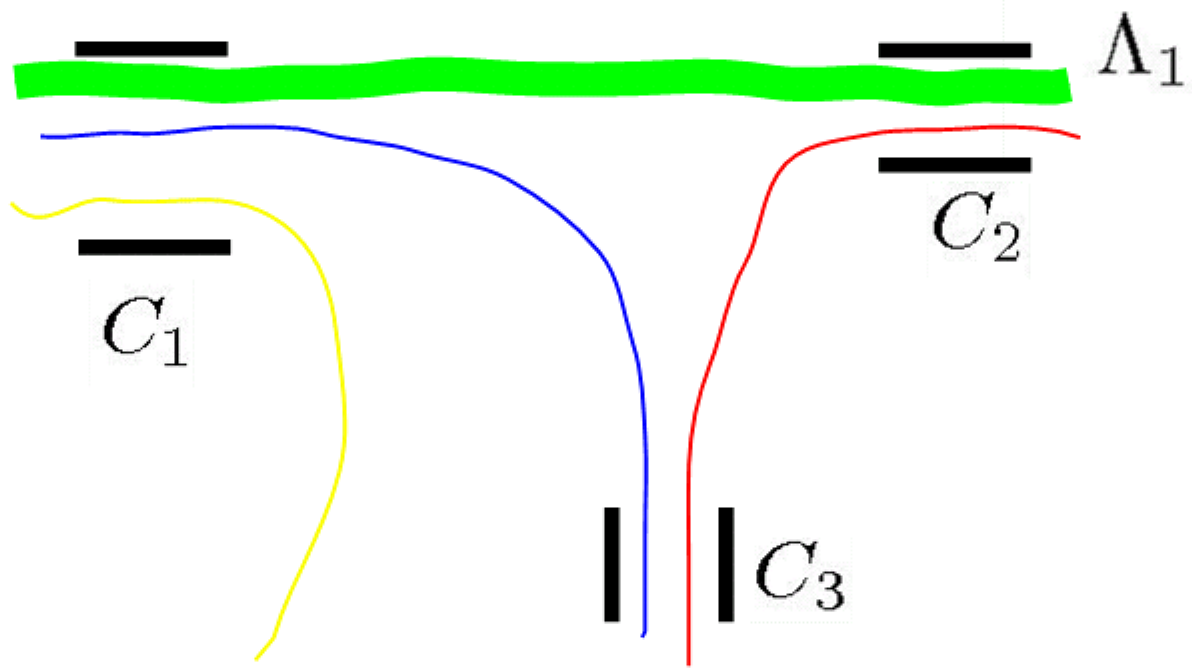
F.P. Kelly, “Charging and Rate Control of Elastic Traffic” (1997)



A NETWORK

A Decomposition Theorem

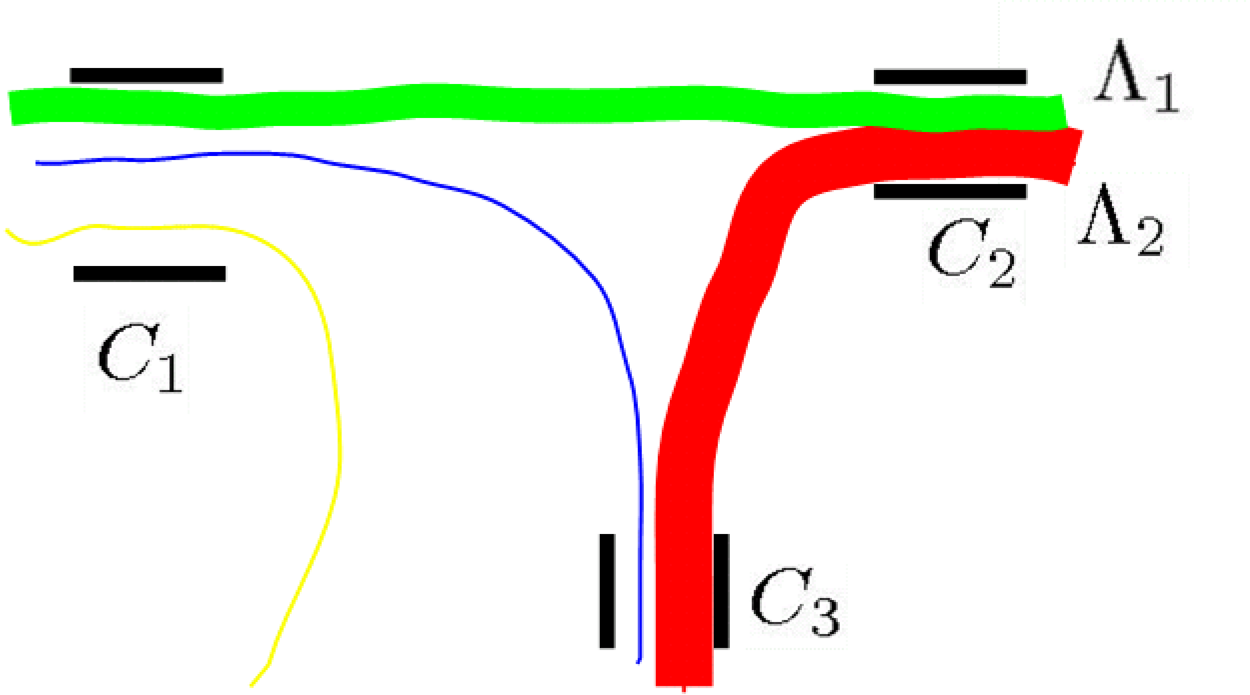
F.P. Kelly, “Charging and Rate Control of Elastic Traffic” (1997)



A NETWORK

A Decomposition Theorem

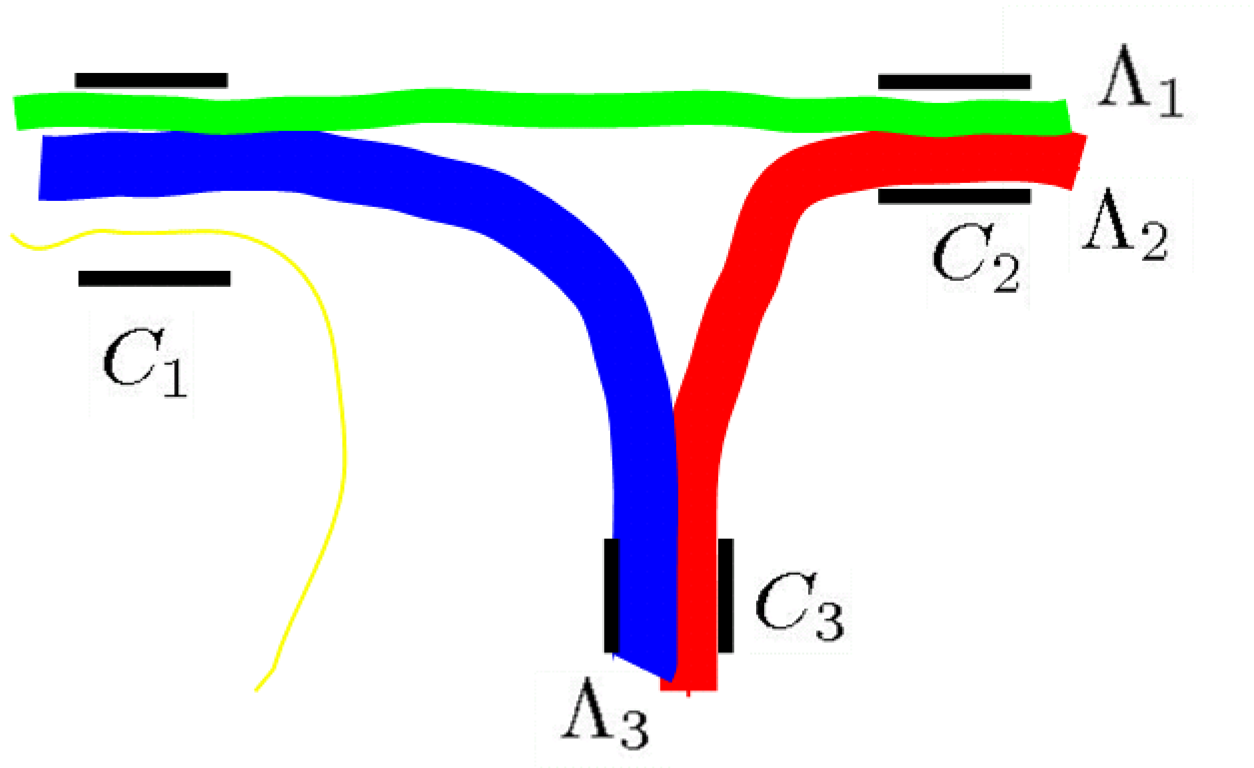
F.P. Kelly, “Charging and Rate Control of Elastic Traffic” (1997)



A NETWORK

A Decomposition Theorem

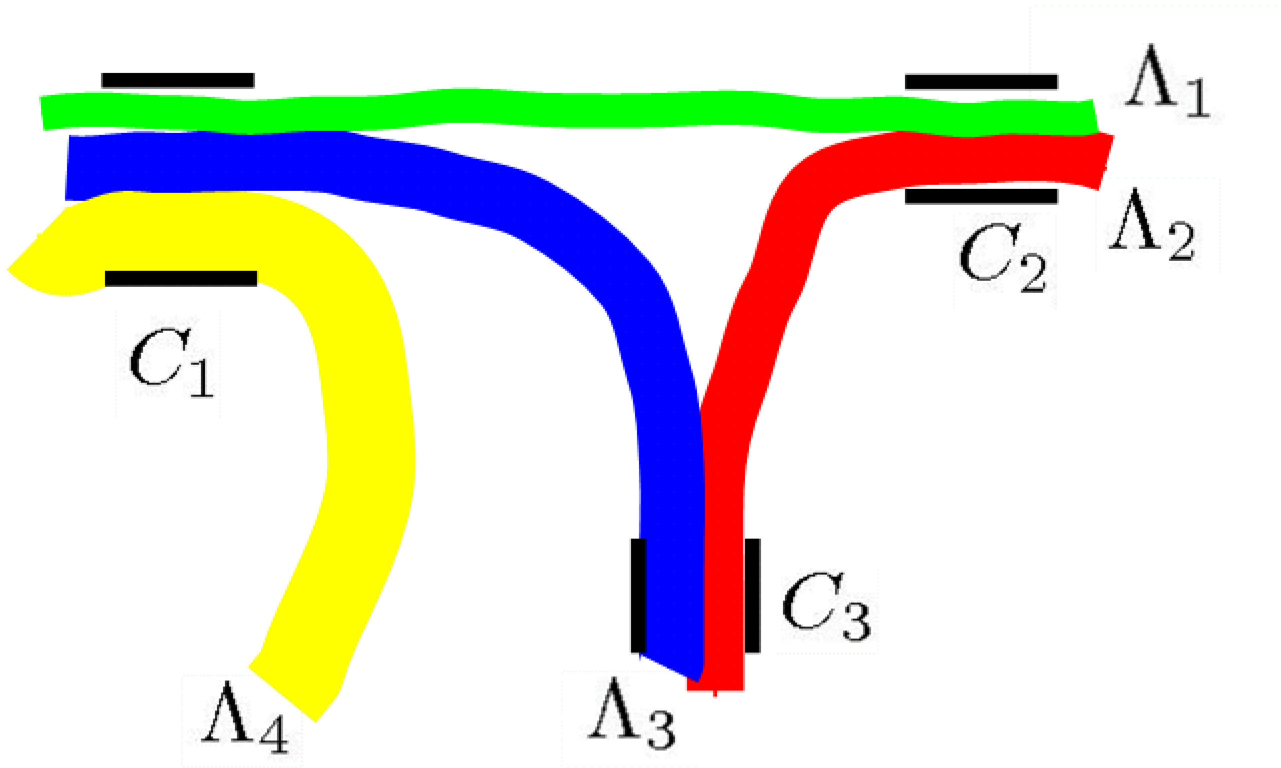
F.P. Kelly, "Charging and Rate Control of Elastic Traffic" (1997)



A NETWORK

A Decomposition Theorem

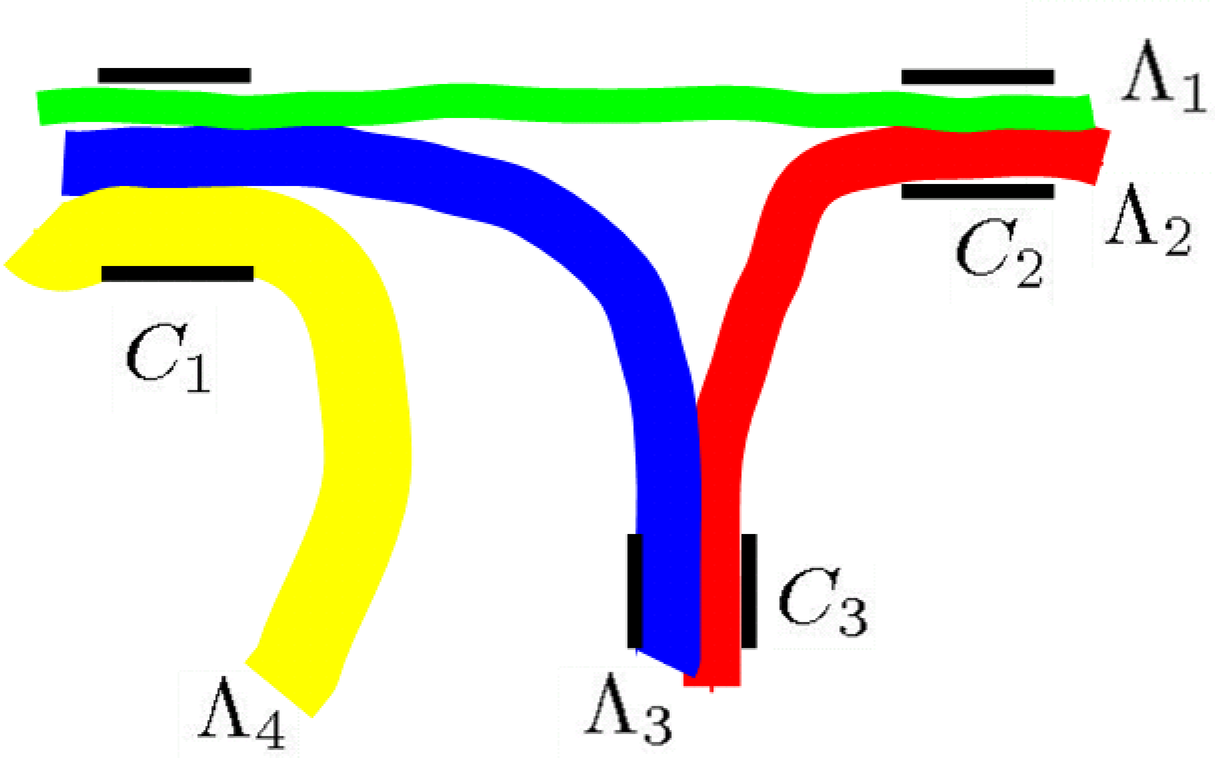
F.P. Kelly, “Charging and Rate Control of Elastic Traffic” (1997)



A NETWORK

A Decomposition Theorem

F.P. Kelly, “Charging and Rate Control of Elastic Traffic” (1997)

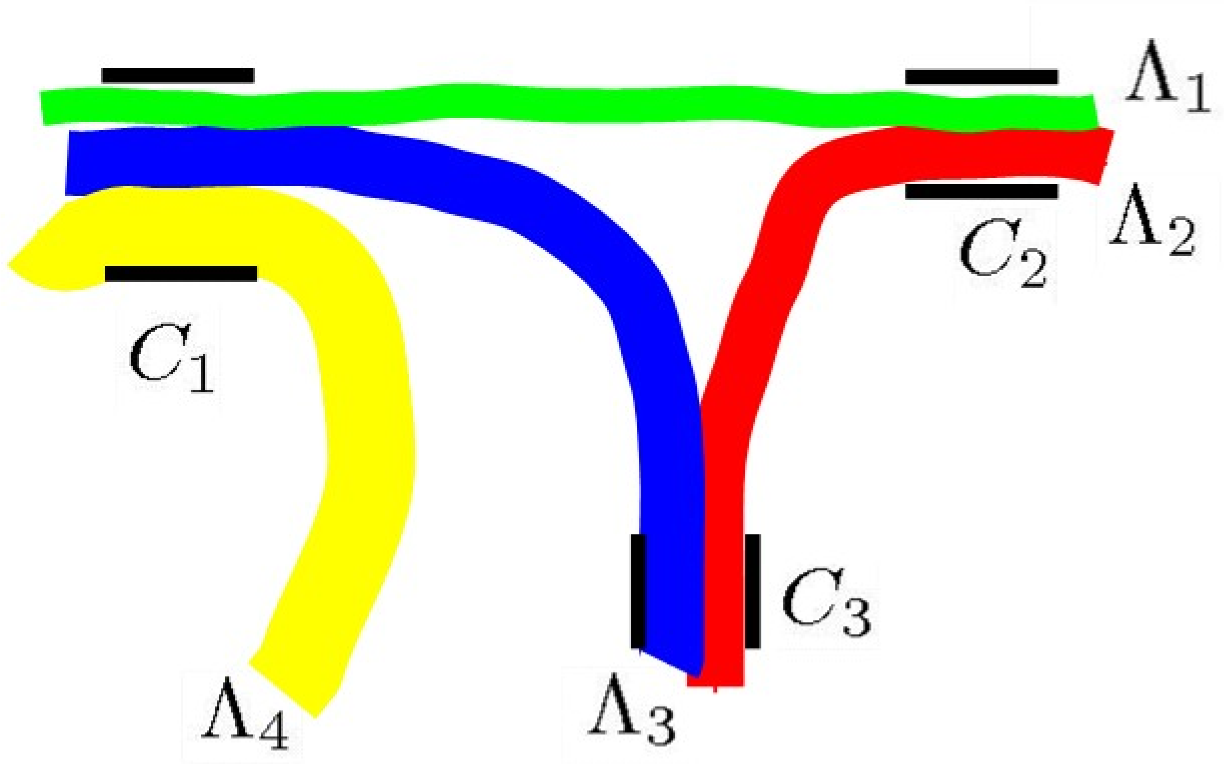


A NETWORK

$$\sum_{i:j\in i} \Lambda_i \le C_j,\ j \in \mathcal{J}$$

A Decomposition Theorem

F.P. Kelly, “Charging and Rate Control of Elastic Traffic” (1997)



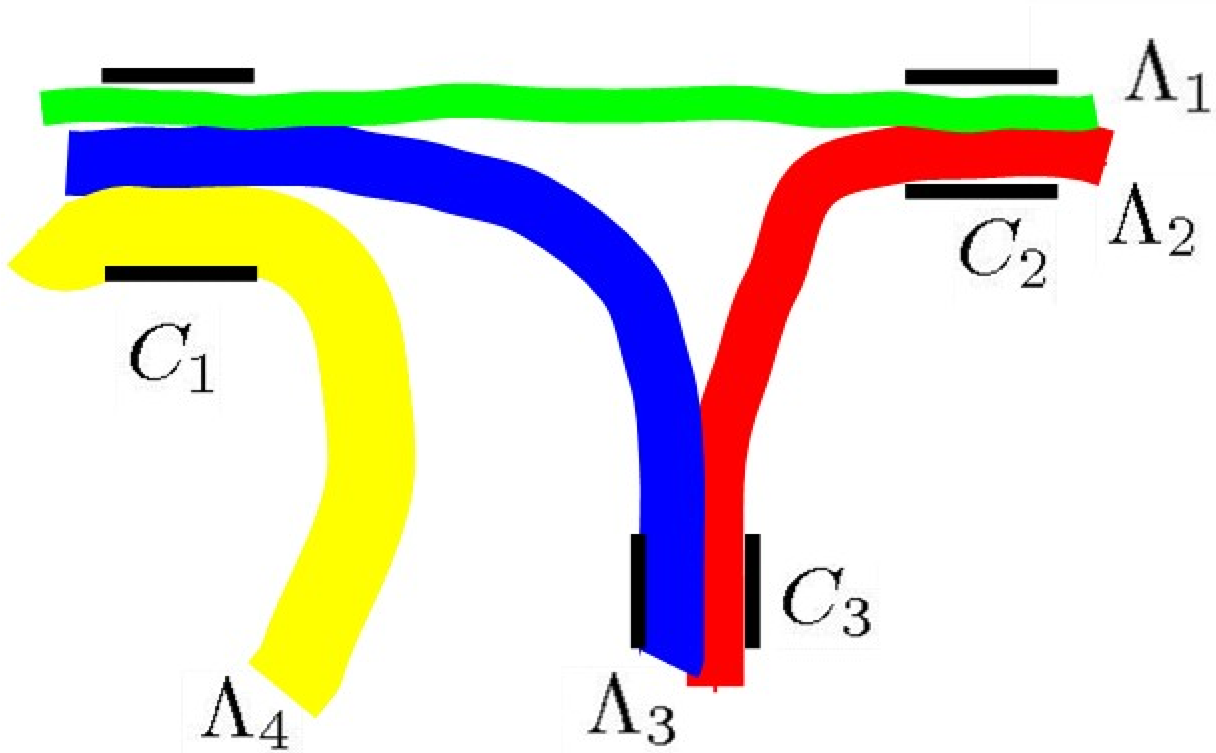
A NETWORK

$$U_i(\Lambda_i)$$

$$\sum_{i:j\in i} \Lambda_i \le C_j,\ j \in \mathcal{J}$$

A Decomposition Theorem

F.P. Kelly, "Charging and Rate Control of Elastic Traffic" (1997)



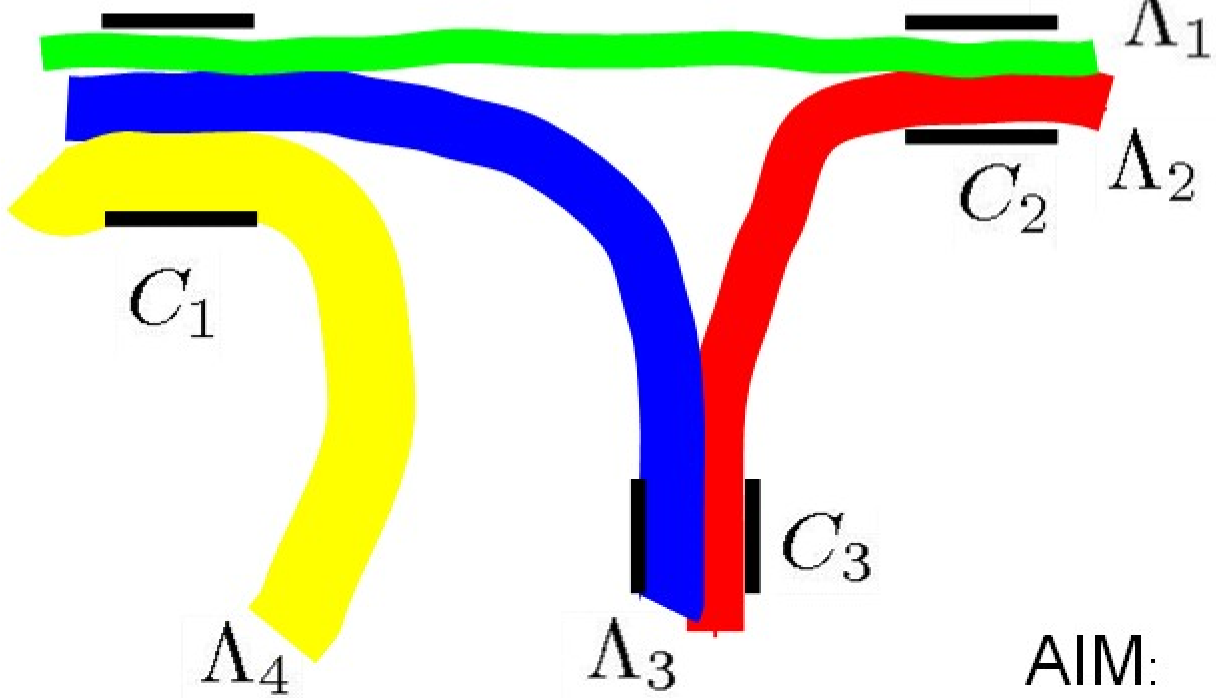
A NETWORK

$$\sum_{i \in \mathcal{I}} U_i(\Lambda_i)$$

$$\sum_{i:j \in i} \Lambda_i \leq C_j, \ j \in \mathcal{J}$$

A Decomposition Theorem

F.P. Kelly, “Charging and Rate Control of Elastic Traffic” (1997)



A NETWORK

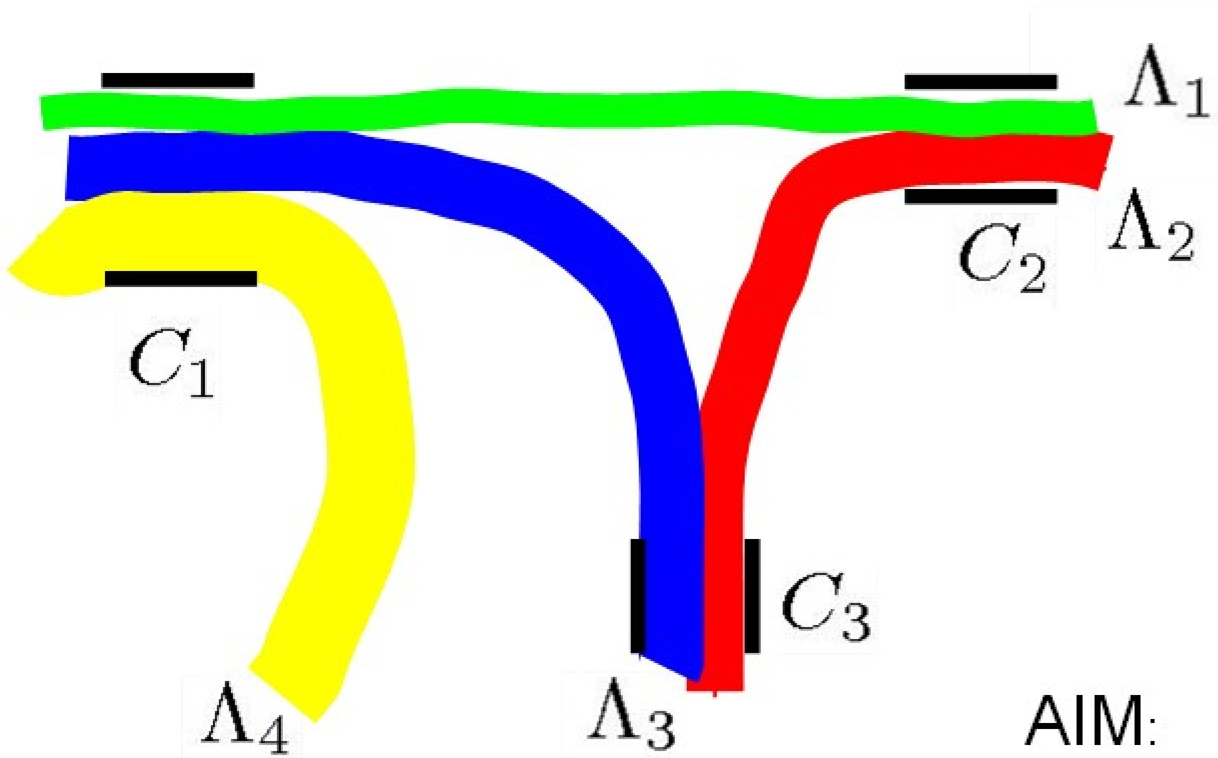
AIM:

$$\sum_{i \in \mathcal{I}} U_i(\Lambda_i)$$

$$\sum_{i: j \in i} \Lambda_i \leq C_j, \; j \in \mathcal{J}$$

A Decomposition Theorem

F.P. Kelly, “Charging and Rate Control of Elastic Traffic” (1997)



A NETWORK

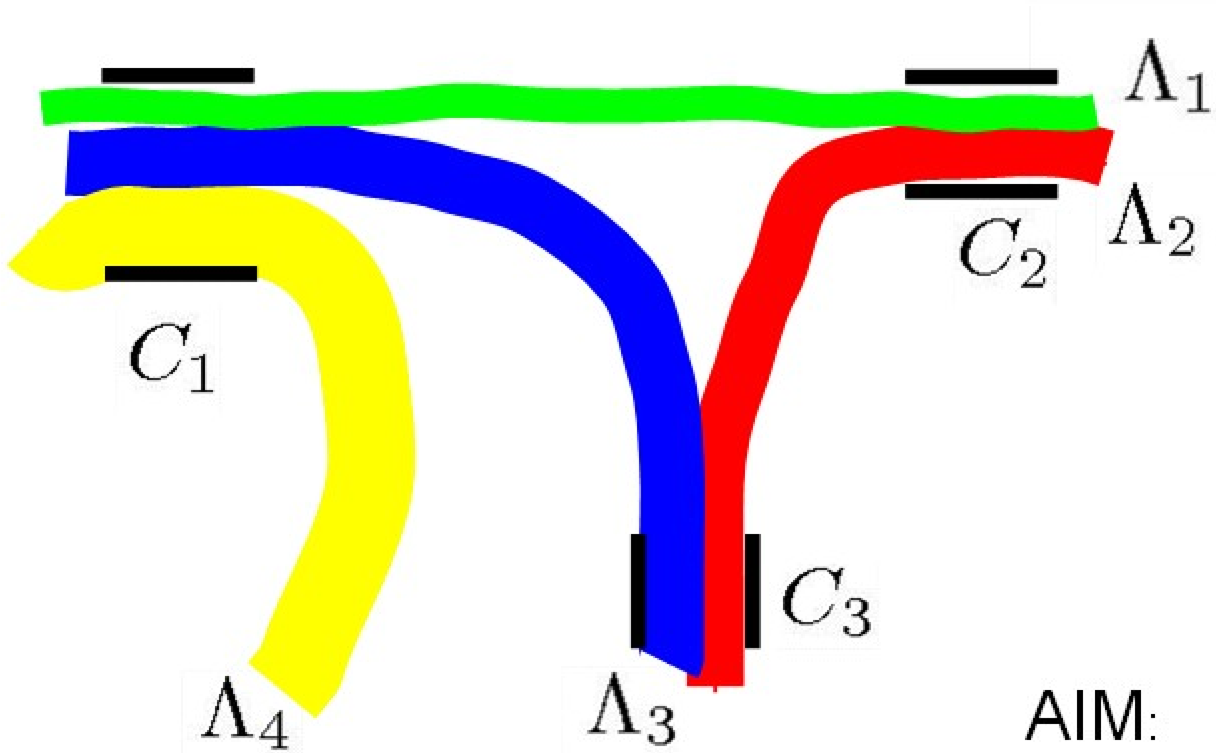
AIM: SYSTEM PROBLEM

$$\sum_{i \in \mathcal{I}} U_i(\Lambda_i)$$

$$\sum_{i : j \in i} \Lambda_i \leq C_j, \; j \in \mathcal{J}$$

A Decomposition Theorem

F.P. Kelly, “Charging and Rate Control of Elastic Traffic” (1997)



A NETWORK

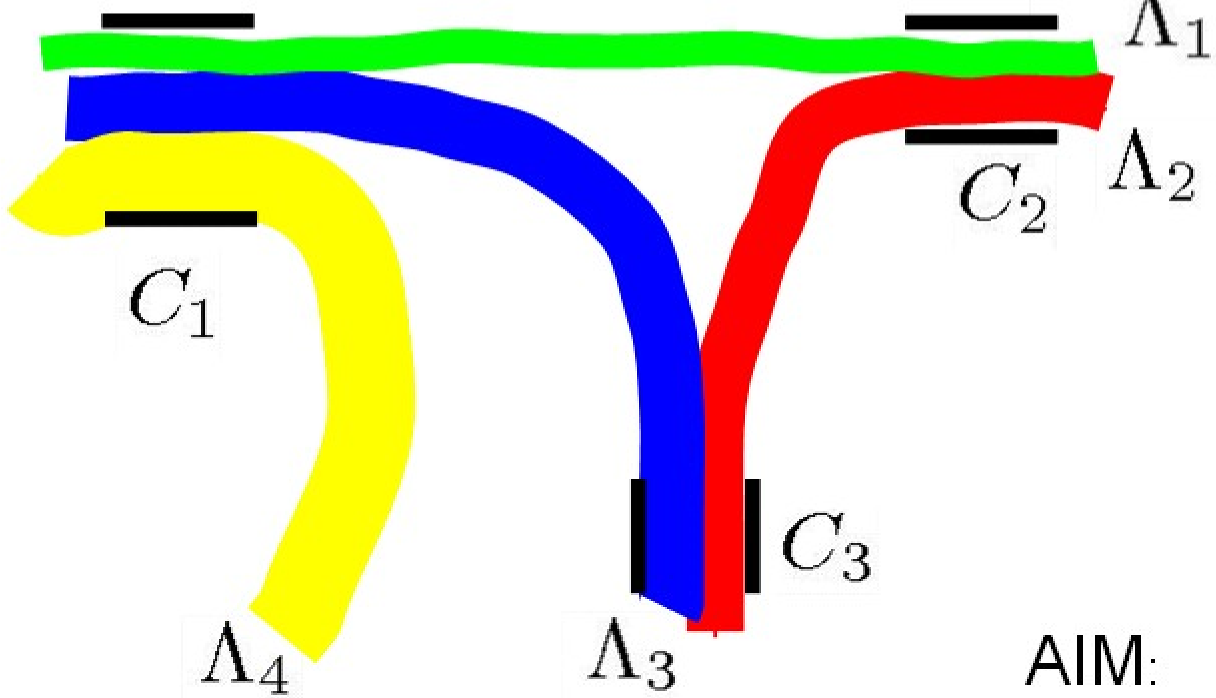
AIM: SYSTEM PROBLEM

$$\max \quad \sum_{i \in \mathcal{I}} U_i(\Lambda_i)$$

$$\sum_{i:j \in i} \Lambda_i \leq C_j, \; j \in \mathcal{J}$$

A Decomposition Theorem

F.P. Kelly, "Charging and Rate Control of Elastic Traffic" (1997)



A NETWORK

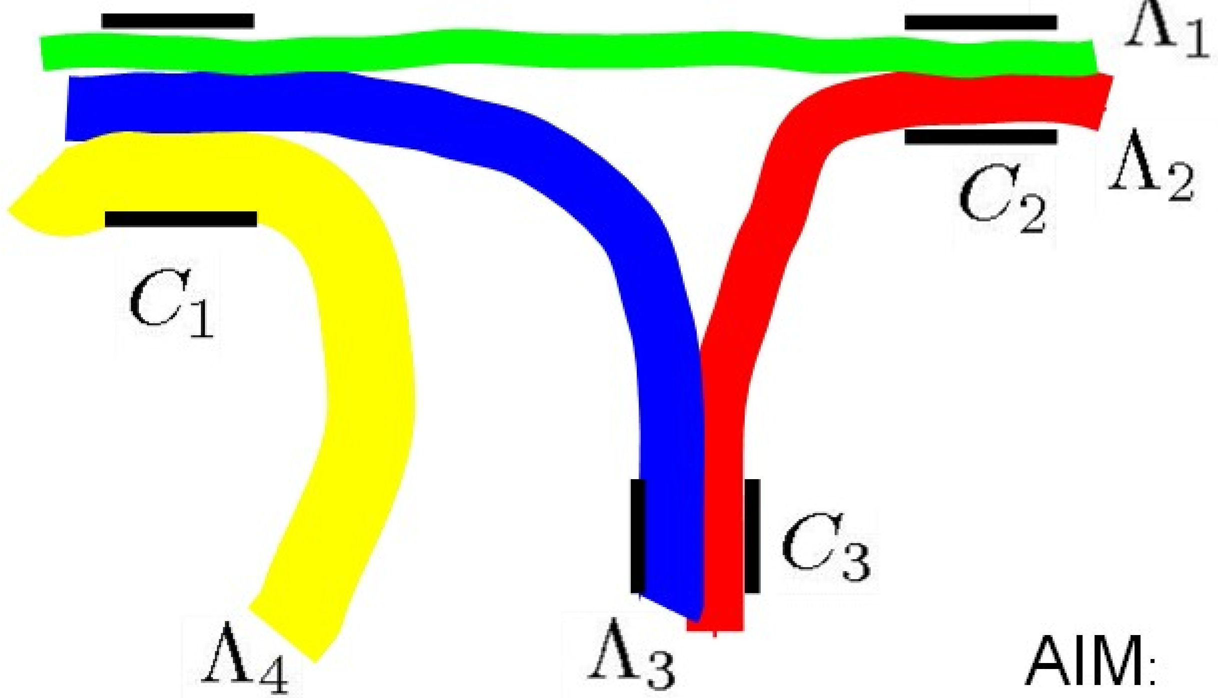
AIM: SYSTEM PROBLEM

$$\max \quad \sum_{i \in \mathcal{I}} U_i(\Lambda_i)$$

$$\text{subject to} \quad \sum_{i : j \in i} \Lambda_i \leq C_j, \; j \in \mathcal{J}$$

A Decomposition Theorem

F.P. Kelly, “Charging and Rate Control of Elastic Traffic” (1997)



A NETWORK

AIM: SYSTEM PROBLEM

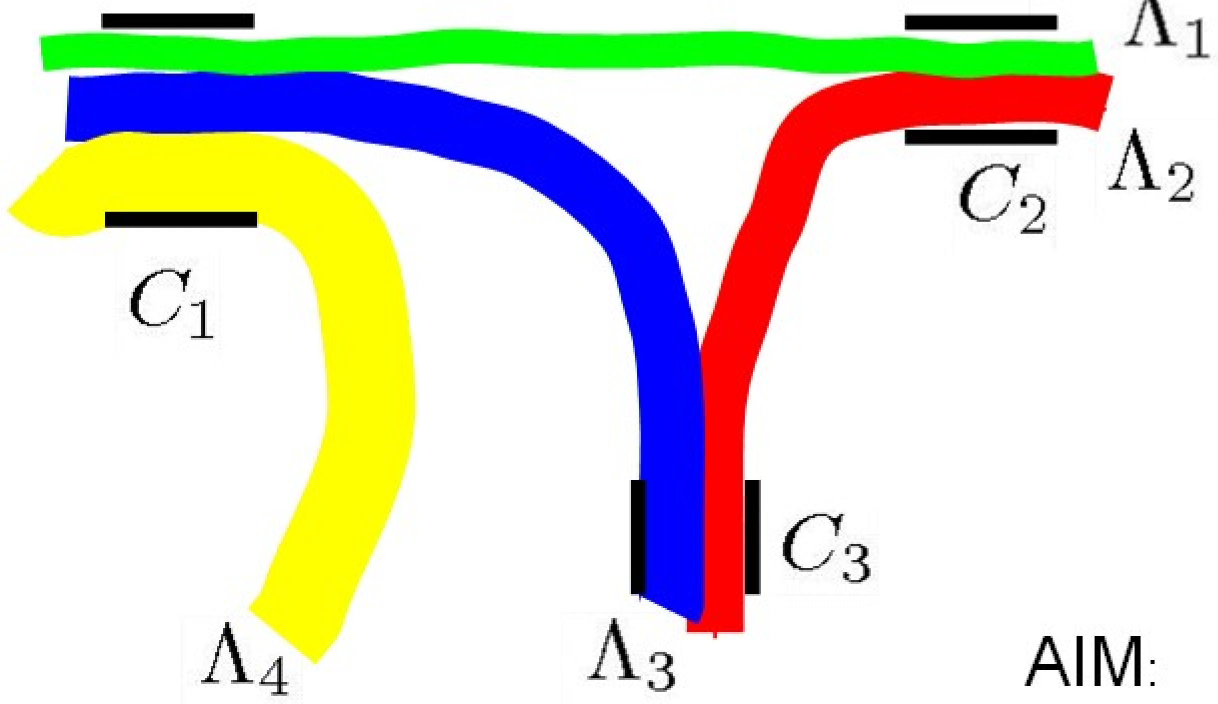
$$\max \quad \sum_{i \in \mathcal{I}} U_i(\Lambda_i)$$

$$\text{subject to} \quad \sum_{i : j \in i} \Lambda_i \leq C_j, \; j \in \mathcal{J}$$

$$\text{over} \quad \Lambda_i \geq 0, \quad i \in \mathcal{I}.$$

A Decomposition Theorem

F.P. Kelly, "Charging and Rate Control of Elastic Traffic" (1997)



A NETWORK PROBLEM

AIM: SYSTEM PROBLEM

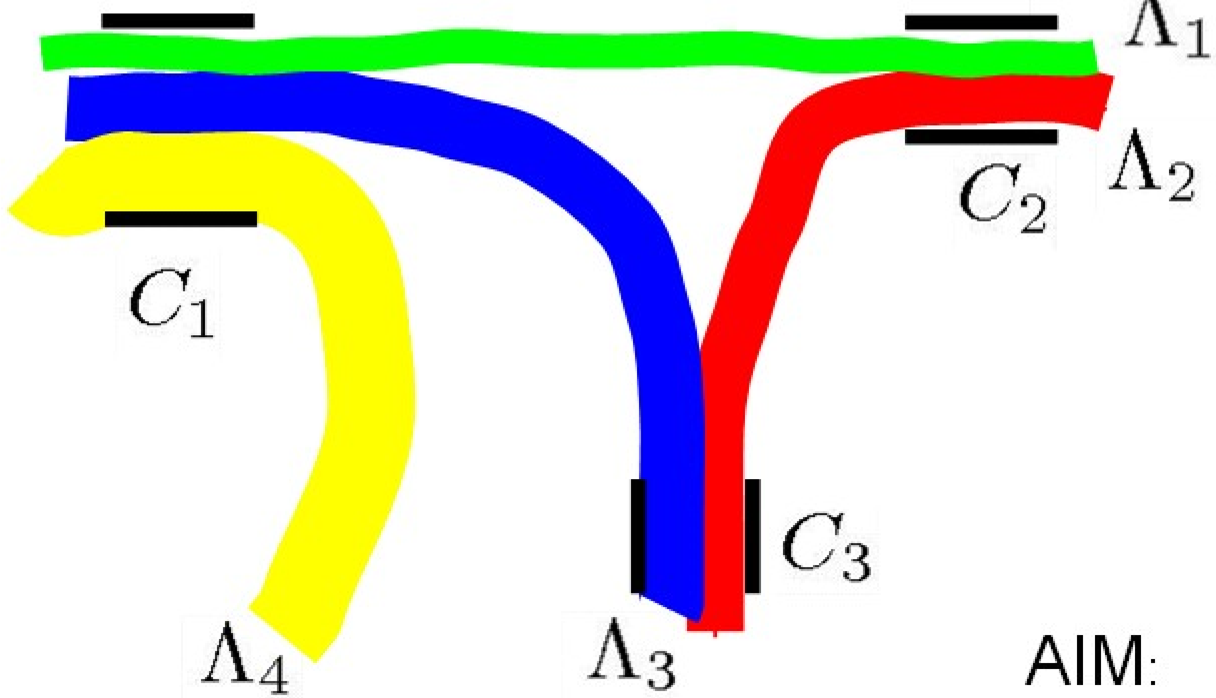
$$\max \quad \sum_{i \in \mathcal{I}} U_i(\Lambda_i)$$

$$\text{subject to} \quad \sum_{i: j \in i} \Lambda_i \le C_j, \ j \in \mathcal{J}$$

$$\text{over} \quad \Lambda_i \ge 0, \quad i \in \mathcal{I}.$$

A Decomposition Theorem

F.P. Kelly, "Charging and Rate Control of Elastic Traffic" (1997)



A NETWORK PROBLEM

$$\sum_{i \in \mathcal{I}} \bar{m}_i \log \Lambda_i$$

AIM: SYSTEM PROBLEM

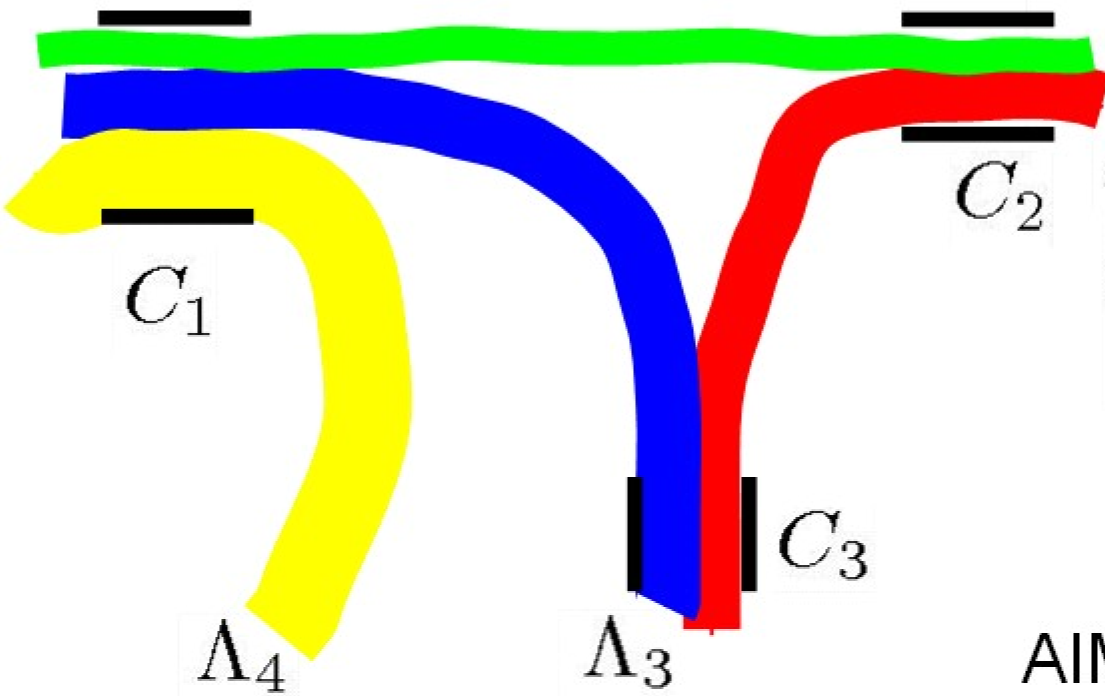
$$\max \quad \sum_{i \in \mathcal{I}} U_i(\Lambda_i)$$

$$\text{subject to} \quad \sum_{i:j \in i} \Lambda_i \leq C_j, \; j \in \mathcal{J}$$

$$\text{over} \quad \Lambda_i \geq 0, \quad i \in \mathcal{I}.$$

A Decomposition Theorem

F.P. Kelly, “Charging and Rate Control of Elastic Traffic” (1997)



A NETWORK PROBLEM

$$\max \sum_{i\in\mathcal{I}} \bar{m}_i \log \Lambda_i$$

$$\text{subject to} \quad \sum_{i:j\in i} \Lambda_i \le C_j,$$

AIM: SYSTEM PROBLEM

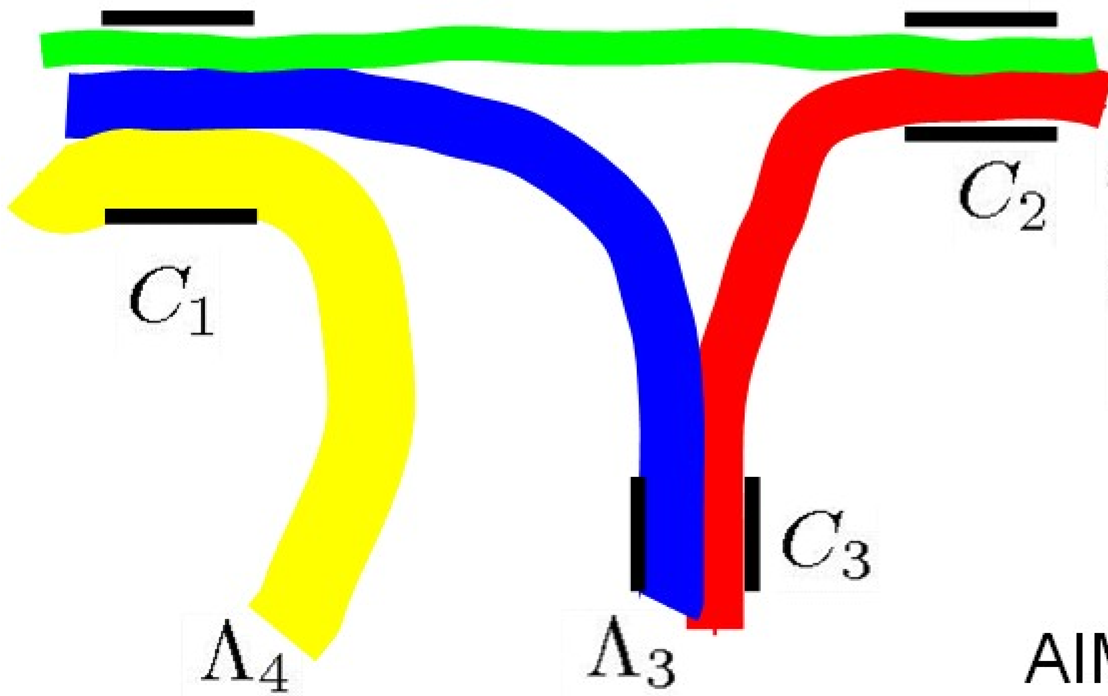
$$\max \quad \sum_{i\in\mathcal{I}} U_i(\Lambda_i)$$

$$\text{subject to} \quad \sum_{i:j\in i} \Lambda_i \le C_j,\ j \in \mathcal{J}$$

$$\text{over} \quad \Lambda_i \ge 0, \quad i \in \mathcal{I}.$$

A Decomposition Theorem

F.P. Kelly, "Charging and Rate Control of Elastic Traffic" (1997)



A NETWORK PROBLEM

$$\max \sum_{i \in \mathcal{I}} \bar{m}_i \log \Lambda_i$$

$$\text{subject to} \quad \sum_{i: j \in i} \Lambda_i \le C_j,$$

USER PROBLEM

AIM: SYSTEM PROBLEM

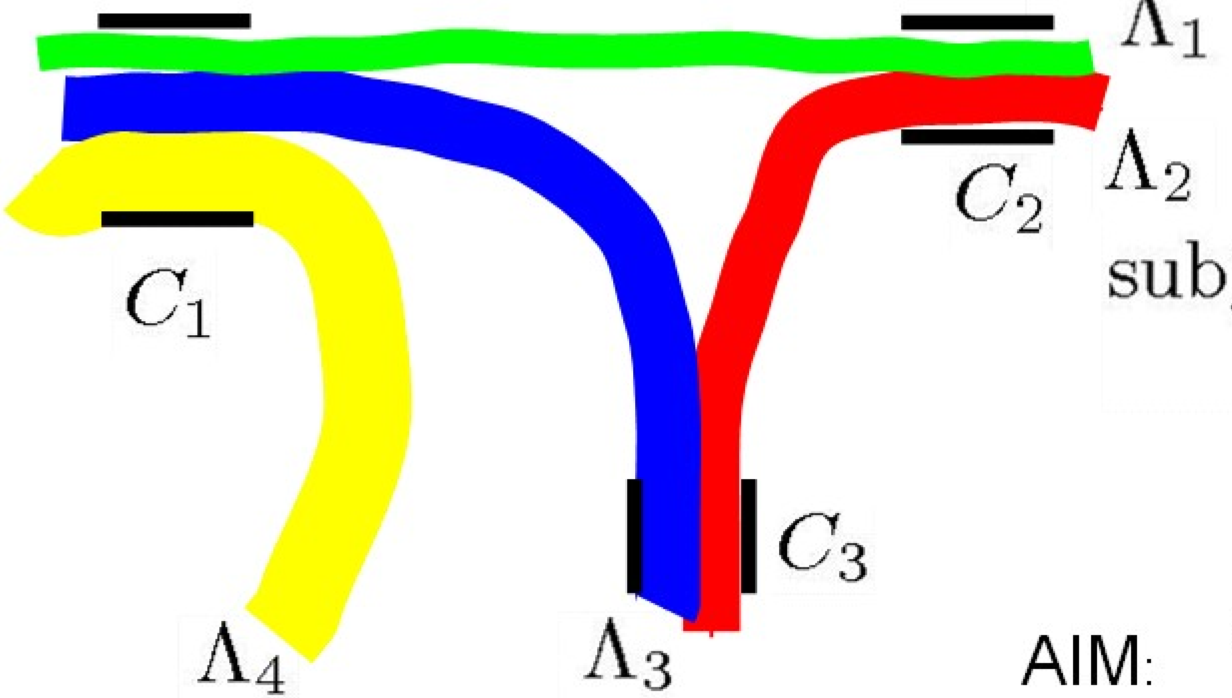
$$\max \quad \sum_{i \in \mathcal{I}} U_i(\Lambda_i)$$

$$\text{subject to} \quad \sum_{i: j \in i} \Lambda_i \le C_j, \; j \in \mathcal{J}$$

$$\text{over} \quad \Lambda_i \ge 0, \quad i \in \mathcal{I}.$$

A Decomposition Theorem

F.P. Kelly, "Charging and Rate Control of Elastic Traffic" (1997)



A NETWORK PROBLEM

$$\max \sum_{i \in \mathcal{I}} \bar{m}_i \log \Lambda_i$$

$$\text{subject to} \quad \sum_{i: j \in i} \Lambda_i \le C_j,$$

AIM: SYSTEM PROBLEM

$$\max \quad \sum_{i \in \mathcal{I}} U_i(\Lambda_i)$$

$$\text{subject to} \quad \sum_{i: j \in i} \Lambda_i \le C_j, \ j \in \mathcal{J}$$

$$\text{over} \quad \Lambda_i \ge 0, \quad i \in \mathcal{I}.$$

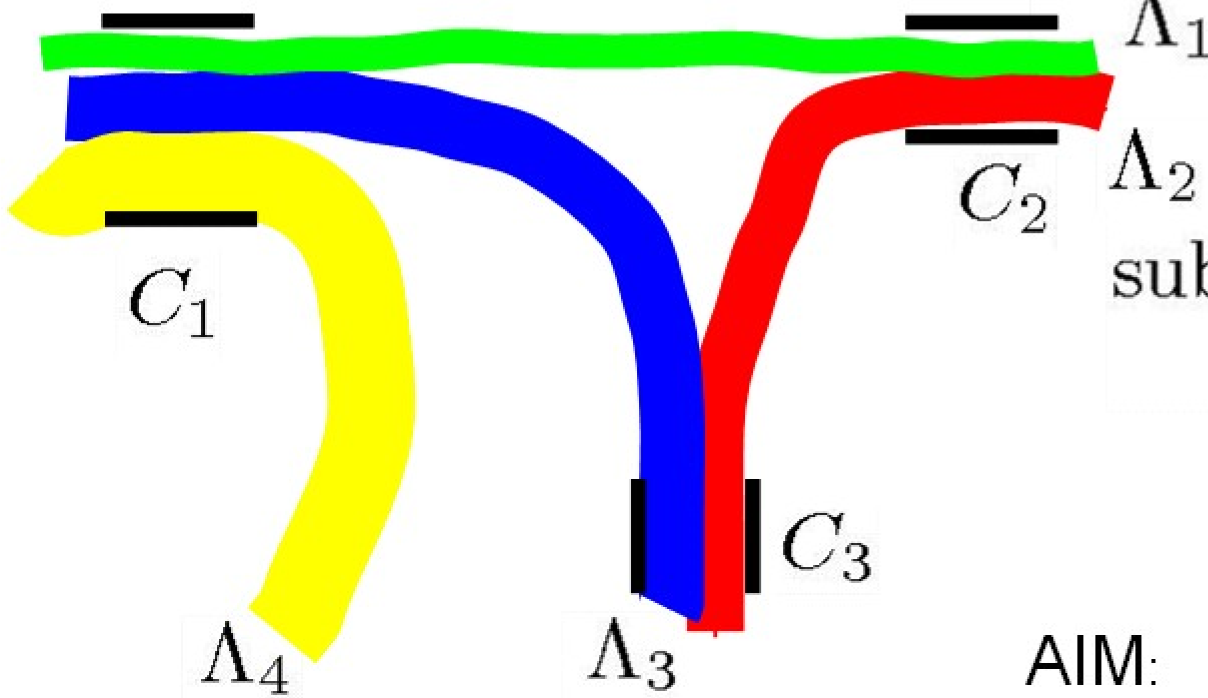
USER PROBLEM

$$\max \quad U_i\left(\frac{\bar{m}_i}{q_i}\right) - \bar{m}_i$$

$$\text{over} \quad \bar{m}_i \ge 0, \quad i \in \mathcal{I}.$$

A Decomposition Theorem

F.P. Kelly, “Charging and Rate Control of Elastic Traffic” (1997)



A NETWORK PROBLEM

$$\max \sum_{i \in \mathcal{I}} \bar{m}_i \log \Lambda_i$$

$$\text{subject to} \quad \sum_{i:j \in i} \Lambda_i \le C_j,$$

SYSTEM PROBLEM

AIM:

$$\max \quad \sum_{i \in \mathcal{I}} U_i(\Lambda_i)$$

$$\text{subject to} \quad \sum_{i:j \in i} \Lambda_i \le C_j, \; j \in \mathcal{J}$$

$$\text{over} \quad \Lambda_i \ge 0, \quad i \in \mathcal{I}.$$

USER PROBLEM

$$\max \quad U_i\left(\frac{\bar{m}_i}{q_i}\right) - \bar{m}_i$$

$$\text{over} \quad \bar{m}_i \ge 0, \quad i \in \mathcal{I}.$$

A Decomposition Theorem (Kelly)

A Decomposition Theorem (Kelly)

The Simultaneous Solution of,

A Decomposition Theorem (Kelly)

The Simultaneous Solution of,

USER PROBLEMS

$$\max \quad U_i\left(\frac{\bar{m}_i}{q_i}\right) - \bar{m}_i$$

$$\text{over} \quad \bar{m}_i \geq 0, \quad i \in \mathcal{I}.$$

A Decomposition Theorem (Kelly)

The Simultaneous Solution of,

USER PROBLEMS

$$\max \quad U_i\left(\frac{\bar{m}_i}{q_i}\right) - \bar{m}_i$$

$$\text{over} \quad \bar{m}_i \geq 0, \quad i \in \mathcal{I}.$$

NETWORK PROBLEM

$$\max \sum_{i \in \mathcal{I}} \bar{m}_i \log \Lambda_i$$

$$\text{subject to} \quad \sum_{i: j \in i} \Lambda_i \leq C_j,$$

A Decomposition Theorem (Kelly)

The Simultaneous Solution of,

USER PROBLEMS

$$\max \quad U_i\left(\frac{\bar{m}_i}{q_i}\right) - \bar{m}_i$$
$$\text{over} \quad \bar{m}_i \geq 0, \quad i \in \mathcal{I}.$$

NETWORK PROBLEM

$$\max \sum_{i \in \mathcal{I}} \bar{m}_i \log \Lambda_i$$
$$\text{subject to} \quad \sum_{i: j \in i} \Lambda_i \leq C_j,$$

AND

$$\bar{m}_i = \Lambda_i q_i$$

A Decomposition Theorem (Kelly)

The Simultaneous Solution of,

USER PROBLEMS

$$\max \quad U_i\left(\frac{\bar{m}_i}{q_i}\right) - \bar{m}_i$$

$$\text{over} \quad \bar{m}_i \geq 0, \quad i \in \mathcal{I}.$$

NETWORK PROBLEM

$$\max \sum_{i \in \mathcal{I}} \bar{m}_i \log \Lambda_i$$

$$\text{subject to} \quad \sum_{i:j \in i} \Lambda_i \leq C_j,$$

AND

$$\bar{m}_i = \Lambda_i q_i$$

Money per unit time

A Decomposition Theorem (Kelly)

The Simultaneous Solution of,

USER PROBLEMS

$$\max \quad U_i\left(\frac{\bar{m}_i}{q_i}\right) - \bar{m}_i$$

$$\text{over} \quad \bar{m}_i \geq 0, \quad i \in \mathcal{I}.$$

NETWORK PROBLEM

$$\max \sum_{i \in \mathcal{I}} \bar{m}_i \log \Lambda_i$$

$$\text{subject to} \quad \sum_{i: j \in i} \Lambda_i \leq C_j,$$

AND

$$\bar{m}_i = \Lambda_i q_i$$

Money per unit time

Price

A Decomposition Theorem (Kelly)

The Simultaneous Solution of,

USER PROBLEMS

$$\max \quad U_i\left(\frac{\bar{m}_i}{q_i}\right) - \bar{m}_i$$
$$\text{over} \quad \bar{m}_i \geq 0, \quad i \in \mathcal{I}.$$

NETWORK PROBLEM

$$\max \sum_{i \in \mathcal{I}} \bar{m}_i \log \Lambda_i$$
$$\text{subject to} \sum_{i:j \in i} \Lambda_i \leq C_j,$$

Network choses prices
with Lagrangian multipliers:

$$q_i = \sum_{j \in i} q_j$$

AND

$$\bar{m}_i = \Lambda_i q_i$$

Money per unit time

Price

A Decomposition Theorem (Kelly)

The Simultaneous Solution of,

USER PROBLEMS

$$\max \quad U_i\left(\frac{\bar{m}_i}{q_i}\right) - \bar{m}_i$$
$$\text{over} \quad \bar{m}_i \geq 0, \quad i \in \mathcal{I}.$$

User *i* choses wealth:

$$\bar{m}_i$$

NETWORK PROBLEM

$$\max \sum_{i \in \mathcal{I}} \bar{m}_i \log \Lambda_i$$
$$\text{subject to} \sum_{i:j \in i} \Lambda_i \leq C_j,$$

Network choses prices
with Lagrangian multipliers:

$$q_i = \sum_{j \in i} q_j$$

AND

$$\bar{m}_i = \Lambda_i q_i$$

Money per unit time

Price

A Decomposition Theorem (Kelly)

The Simultaneous Solution of,

USER PROBLEMS

$$\max \quad U_i\left(\frac{\bar{m}_i}{q_i}\right) - \bar{m}_i$$

$$\text{over} \quad \bar{m}_i \geq 0, \quad i \in \mathcal{I}.$$

NETWORK PROBLEM

$$\max \sum_{i \in \mathcal{I}} \bar{m}_i \log \Lambda_i$$

$$\text{subject to} \quad \sum_{i: j \in i} \Lambda_i \leq C_j,$$

AND

$$\bar{m}_i = \Lambda_i q_i$$

A Decomposition Theorem (Kelly)

The Simultaneous Solution of,

USER PROBLEMS

$$\max \quad U_i\left(\frac{\bar{m}_i}{q_i}\right) - \bar{m}_i$$

$$\text{over} \quad \bar{m}_i \geq 0, \quad i \in \mathcal{I}.$$

NETWORK PROBLEM

$$\max \sum_{i \in \mathcal{I}} \bar{m}_i \log \Lambda_i$$

$$\text{subject to} \quad \sum_{i:j \in i} \Lambda_i \leq C_j,$$

AND

$$\bar{m}_i = \Lambda_i q_i$$

Solves

A Decomposition Theorem (Kelly)

The Simultaneous Solution of,

USER PROBLEMS

$$\max \quad U_i\left(\frac{\bar{m}_i}{q_i}\right) - \bar{m}_i$$

$$\text{over} \quad \bar{m}_i \geq 0, \quad i \in \mathcal{I}.$$

NETWORK PROBLEM

$$\max \sum_{i \in \mathcal{I}} \bar{m}_i \log \Lambda_i$$

$$\text{subject to} \quad \sum_{i:j \in i} \Lambda_i \leq C_j,$$

AND

$$\bar{m}_i = \Lambda_i q_i$$

Solves

SYSTEM PROBLEM

$$\max \quad \sum_{i \in \mathcal{I}} U_i(\Lambda_i)$$

$$\text{subject to} \quad \sum_{i:j \in i} \Lambda_i \leq C_j, \; j \in \mathcal{J}$$

$$\text{over} \quad \Lambda_i \geq 0, \quad i \in \mathcal{I}.$$

Dynamics

Dynamics

We want a network solve these optimisation problems **implicitly**

Dynamics

We want a network solve these optimisation problems **implicitly**

...Differential Equations?

Dynamics

We want a network solve these optimisation problems **implicitly**
...Differential Equations?

Kelly, Maulloo, Tan (1998)
Kelly, Gibbens (1999)
Kelly, Key, Zachary (2000)
Johari, Tan (2001)
Raina, Towsley, Wischik (2005)
Strulo, Walker, Wennink (2007)
Yi, Chiang (2008)

Dynamics

We want a network solve these optimisation problems **implicitly**

...Differential Equations?

Dynamics

We want a network solve these optimisation problems **implicitly**

...Differential Equations?

A NETWORK ALGORITHM:

Dynamics

We want a network solve these optimisation problems **implicitly**

...Differential Equations?

A NETWORK ALGORITHM:

$$\frac{d}{dt}\Lambda_i(t) = \kappa\Big(m_i - \Lambda_i(t)\sum_{j\in i}\mu_j(t)\Big)$$

Dynamics

We want a network solve these optimisation problems **implicitly**

...Differential Equations?

A NETWORK ALGORITHM:

$$\frac{d}{dt}\Lambda_i(t) = \kappa\Big(m_i - \Lambda_i(t)\sum_{j\in i}\mu_j(t)\Big)$$

Increase Rule

Dynamics

We want a network solve these optimisation problems **implicitly**

...Differential Equations?

A NETWORK ALGORITHM:

$$\frac{d}{dt}\Lambda_i(t) = \kappa\Big(m_i - \Lambda_i(t)\sum_{j\in i}\mu_j(t)\Big)$$

Increase Rule

Decrease Rule

Dynamics

We want a network solve these optimisation problems **implicitly**

...Differential Equations?

A NETWORK ALGORITHM:

$$\frac{d}{dt}\Lambda_i(t) = \kappa\Big(m_i - \Lambda_i(t)\sum_{j\in i}\mu_j(t)\Big)$$

Dynamics

We want a network solve these optimisation problems **implicitly**

...Differential Equations?

A NETWORK ALGORITHM:

$$\frac{d}{dt}\Lambda_i(t) = \kappa\Big(m_i - \Lambda_i(t)\sum_{j\in i}\mu_j(t)\Big)$$

Where

$$\mu_j(t) = p_j\Big(\sum_{r:j\in r}\Lambda_r(t)\Big)$$

Dynamics

We want a network solve these optimisation problems **implicitly**

...Differential Equations?

A NETWORK ALGORITHM:

$$\frac{d}{dt}\Lambda_i(t) = \kappa\Big(m_i - \Lambda_i(t)\sum_{j\in i}\mu_j(t)\Big)$$

Where

$$\mu_j(t) = p_j(\sum_{r:j\in r}\Lambda_r(t))$$

Congestion

Dynamics

We want a network solve these optimisation problems **implicitly**

...Differential Equations?

A NETWORK ALGORITHM:

$$\frac{d}{dt}\Lambda_i(t) = \kappa\Big(m_i - \Lambda_i(t)\sum_{j\in i}\mu_j(t)\Big)$$

Where

$$\mu_j(t) = p_j\Big(\sum_{r:j\in r}\Lambda_r(t)\Big)$$

Price Congestion

Dynamics

We want a network solve these optimisation problems **implicitly**

...Differential Equations?

A NETWORK ALGORITHM:

$$\frac{d}{dt}\Lambda_i(t) = \kappa\Big(m_i - \Lambda_i(t)\sum_{j\in i}\mu_j(t)\Big)$$

Where

$$\mu_j(t) = p_j\Big(\sum_{r:j\in r}\Lambda_r(t)\Big)$$

Dynamics

We want a network solve these optimisation problems **implicitly**

...Differential Equations?

A NETWORK ALGORITHM:

$$\frac{d}{dt}\Lambda_i(t) = \kappa\Big(m_i - \Lambda_i(t)\sum_{j\in i}\mu_j(t)\Big)$$

Where

$$\mu_j(t) = p_j(\sum_{r:j\in r}\Lambda_r(t))$$

A Lyanpunov Function:

Dynamics

We want a network solve these optimisation problems **implicitly**

...Differential Equations?

A NETWORK ALGORITHM:

$$\frac{d}{dt}\Lambda_i(t) = \kappa\Big(m_i - \Lambda_i(t)\sum_{j\in i}\mu_j(t)\Big)$$

Where

$$\mu_j(t) = p_j\Big(\sum_{r:j\in r}\Lambda_r(t)\Big)$$

A Lyanpunov Function:

$$\mathcal{L}(\Lambda) = \sum_{i\in\mathcal{I}} m_i \log\Lambda_i - \sum_{j\in\mathcal{J}}\int_0^{\sum_{r:j\in r}\Lambda_r} p_j(y)\,dy$$

Dynamics

We want a network solve these optimisation problems **implicitly**

...Differential Equations?

A NETWORK ALGORITHM:

$$\frac{d}{dt}\Lambda_i(t) = \kappa\Big(m_i - \Lambda_i(t)\sum_{j\in i}\mu_j(t)\Big)$$

Where

$$\mu_j(t) = p_j\big(\sum_{r:j\in r}\Lambda_r(t)\big)$$

A Lyanpunov Function:

$$\mathcal{L}(\Lambda) = \sum_{i\in\mathcal{I}} m_i \log\Lambda_i - \sum_{j\in\mathcal{J}}\int_0^{\sum_{r:j\in r}\Lambda_r} p_j(y)dy$$

Network Problem
Utility

Dynamics

We want a network solve these optimisation problems **implicitly**

...Differential Equations?

A NETWORK ALGORITHM:

$$\frac{d}{dt}\Lambda_i(t) = \kappa\Big(m_i - \Lambda_i(t)\sum_{j\in i}\mu_j(t)\Big)$$

Where

$$\mu_j(t) = p_j\Big(\sum_{r:j\in r}\Lambda_r(t)\Big)$$

A Lyanpunov Function:

$$\mathcal{L}(\Lambda) = \sum_{i\in\mathcal{I}} m_i \log \Lambda_i - \sum_{j\in\mathcal{J}} \int_0^{\sum_{r:j\in r}\Lambda_r} p_j(y)\,dy$$

Network Problem Utility

Softer Capacity Constraints

Dynamics

We want a network solve these optimisation problems **implicitly**

...Differential Equations?

A NETWORK ALGORITHM:

$$\frac{d}{dt}\Lambda_i(t) = \kappa\Big(m_i - \Lambda_i(t)\sum_{j\in i}\mu_j(t)\Big)$$

Where

$$\mu_j(t) = p_j\Big(\sum_{r:j\in r}\Lambda_r(t)\Big)$$

A Lyanpunov Function:

$$\mathcal{L}(\Lambda) = \sum_{i\in\mathcal{I}} m_i \log\Lambda_i - \sum_{j\in\mathcal{J}}\int_0^{\sum_{r:j\in r}\Lambda_r} p_j(y)\,dy$$

Dynamics

We want a network solve these optimisation problems **implicitly**

...Differential Equations?

A NETWORK ALGORITHM:

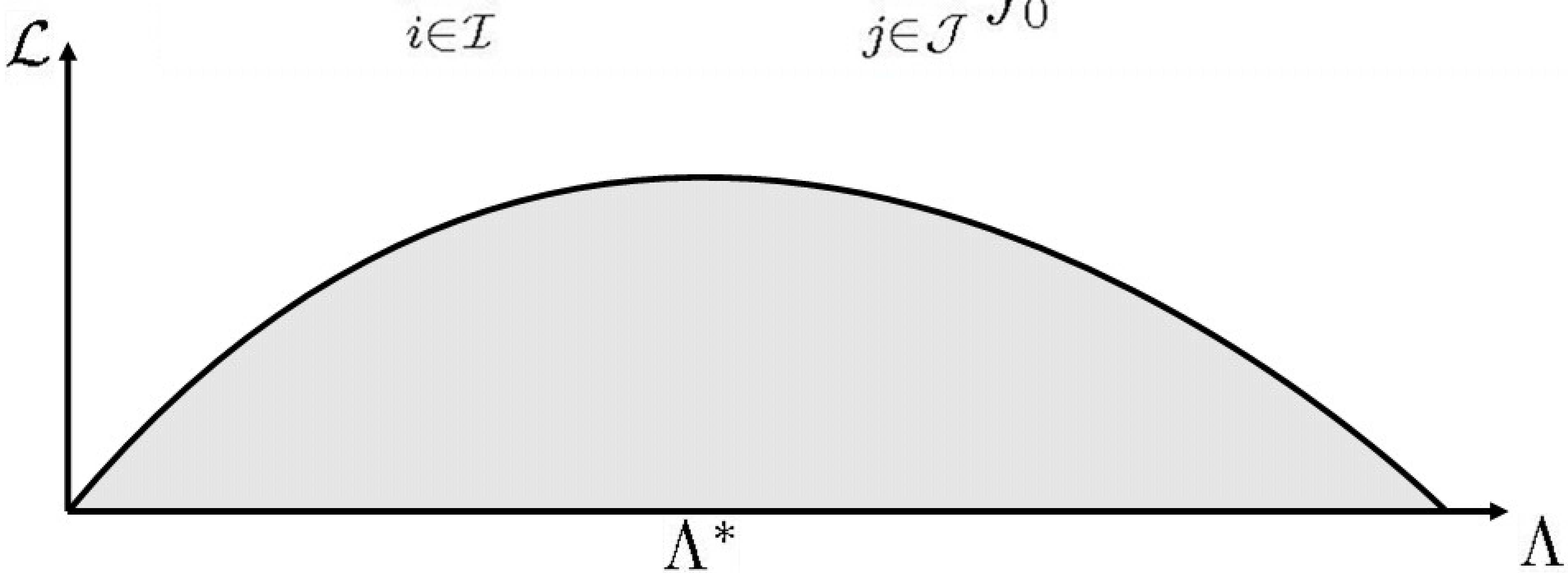
$$\frac{d}{dt}\Lambda_i(t) = \kappa\Big(m_i - \Lambda_i(t)\sum_{j\in i}\mu_j(t)\Big)$$

Where

$$\mu_j(t) = p_j(\sum_{r:j\in r}\Lambda_r(t))$$

A Lyanpunov Function:

$$\mathcal{L}(\Lambda) = \sum_{i\in\mathcal{I}} m_i \log\Lambda_i - \sum_{j\in\mathcal{J}}\int_0^{\sum_{r:j\in r}\Lambda_r} p_j(y)dy$$



Dynamics

We want a network solve these optimisation problems **implicitly**

...Differential Equations?

A NETWORK ALGORITHM:

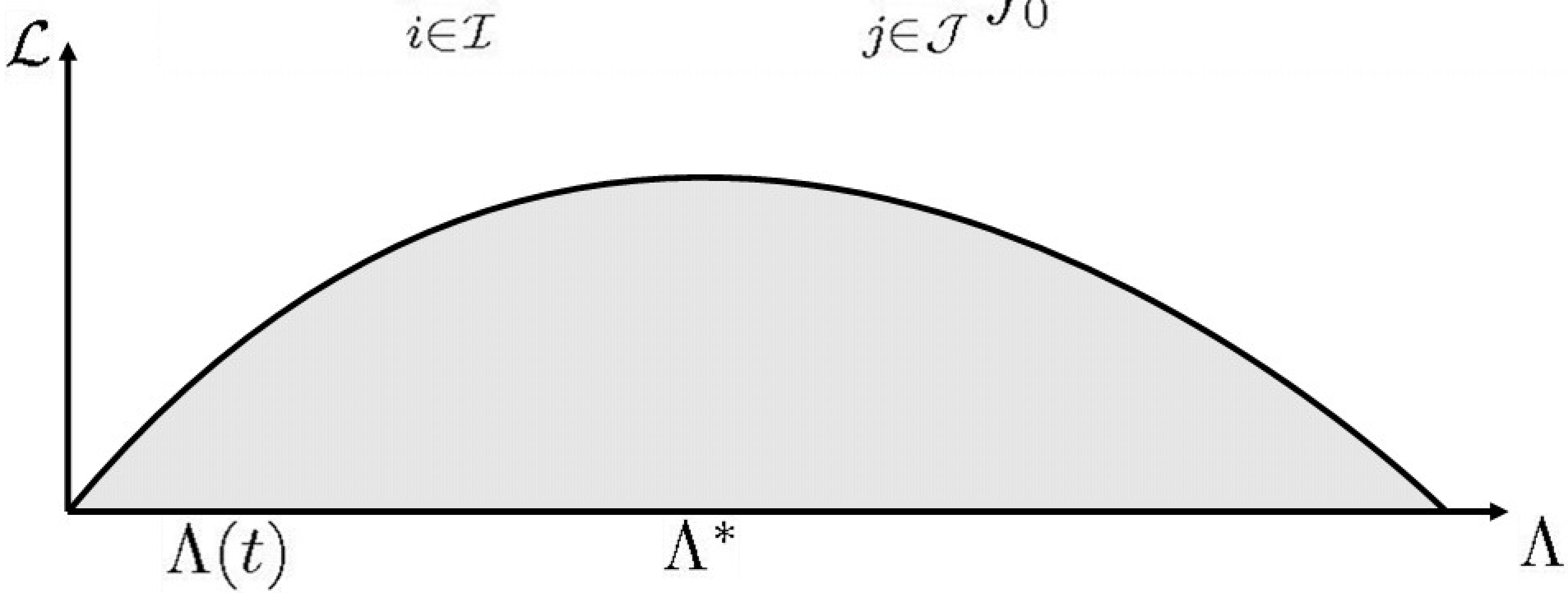
$$\frac{d}{dt}\Lambda_i(t) = \kappa\Big(m_i - \Lambda_i(t)\sum_{j\in i}\mu_j(t)\Big)$$

Where

$$\mu_j(t) = p_j\big(\sum_{r:j\in r}\Lambda_r(t)\big)$$

A Lyanpunov Function:

$$\mathcal{L}(\Lambda) = \sum_{i\in\mathcal{I}} m_i \log\Lambda_i - \sum_{j\in\mathcal{J}}\int_0^{\sum_{r:j\in r}\Lambda_r} p_j(y)dy$$



Dynamics

We want a network solve these optimisation problems **implicitly**

...Differential Equations?

A NETWORK ALGORITHM:

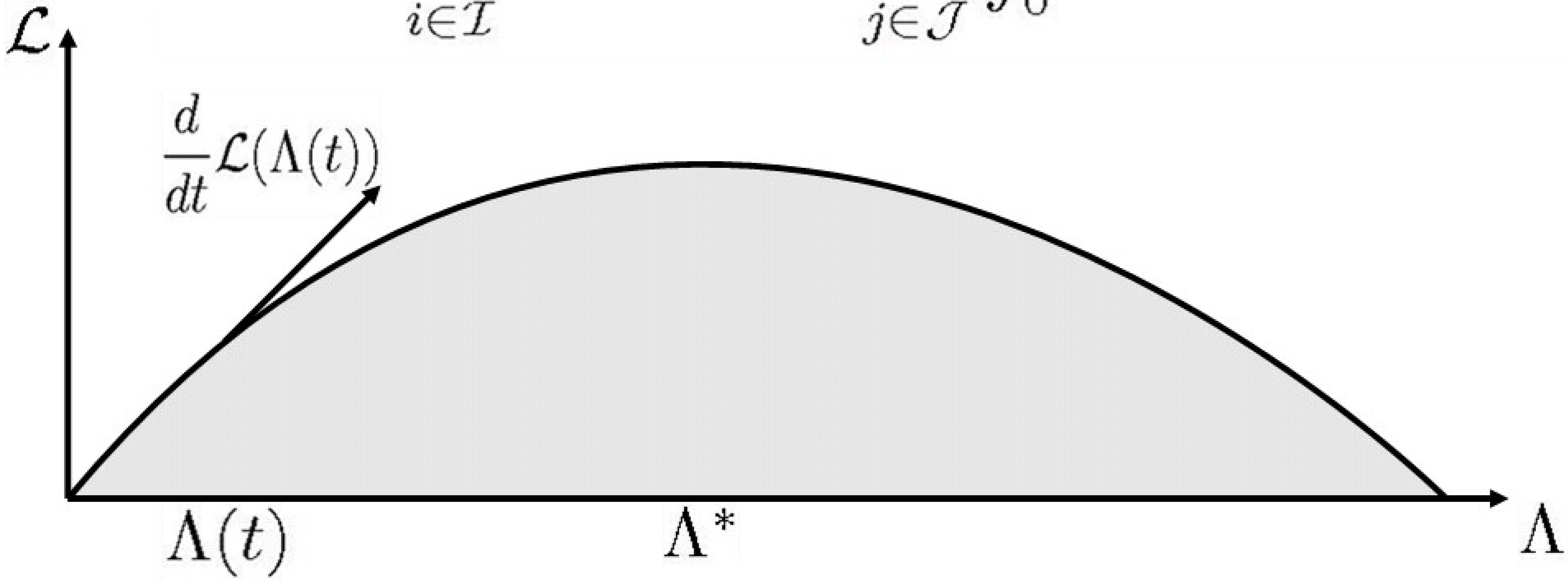
$$\frac{d}{dt}\Lambda_i(t) = \kappa\Big(m_i - \Lambda_i(t)\sum_{j\in i}\mu_j(t)\Big)$$

Where

$$\mu_j(t) = p_j\big(\sum_{r:j\in r}\Lambda_r(t)\big)$$

A Lyanpunov Function:

$$\mathcal{L}(\Lambda) = \sum_{i\in\mathcal{I}} m_i \log\Lambda_i - \sum_{j\in\mathcal{J}}\int_0^{\sum_{r:j\in r}\Lambda_r} p_j(y)dy$$



Dynamics

We want a network solve these optimisation problems **implicitly**

...Differential Equations?

A NETWORK ALGORITHM:

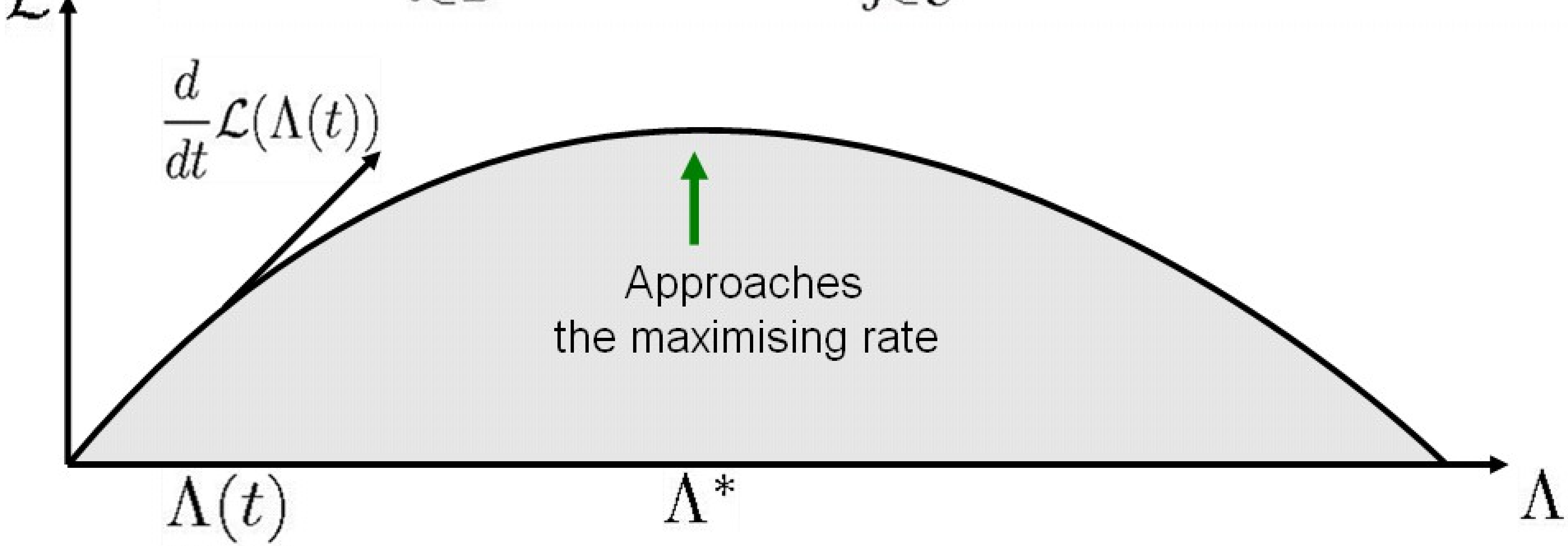
$$\frac{d}{dt}\Lambda_i(t) = \kappa\Big(m_i - \Lambda_i(t)\sum_{j\in i}\mu_j(t)\Big)$$

Where

$$\mu_j(t) = p_j\Big(\sum_{r:j\in r}\Lambda_r(t)\Big)$$

A Lyanpunov Function:

$$\mathcal{L}(\Lambda) = \sum_{i\in\mathcal{I}} m_i \log\Lambda_i - \sum_{j\in\mathcal{J}}\int_0^{\sum_{r:j\in r}\Lambda_r} p_j(y)\,dy$$



Dynamics

We want a network solve these optimisation problems **implicitly**

...Differential Equations?

A NETWORK ALGORITHM:

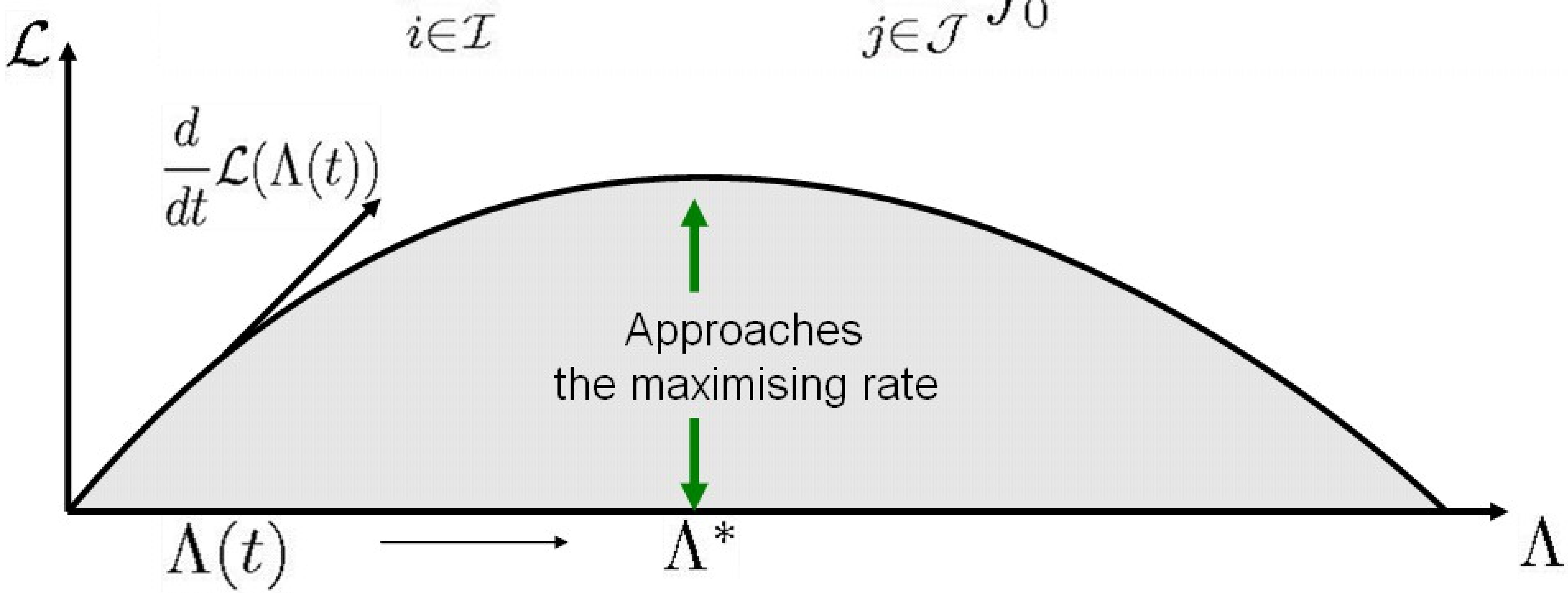
$$\frac{d}{dt}\Lambda_i(t) = \kappa\Big(m_i - \Lambda_i(t)\sum_{j\in i}\mu_j(t)\Big)$$

Where

$$\mu_j(t) = p_j\Big(\sum_{r:j\in r}\Lambda_r(t)\Big)$$

A Lyanpunov Function:

$$\mathcal{L}(\Lambda) = \sum_{i\in\mathcal{I}} m_i \log \Lambda_i - \sum_{j\in\mathcal{J}}\int_0^{\sum_{r:j\in r}\Lambda_r} p_j(y)\,dy$$



Dynamics

We want a network solve these optimisation problems **implicitly**

...Differential Equations?

A NETWORK ALGORITHM:

$$\frac{d}{dt}\Lambda_i(t) = \kappa\Big(m_i - \Lambda_i(t)\sum_{j\in i}\mu_j(t)\Big)$$

Where

$$\mu_j(t) = p_j\Big(\sum_{r:j\in r}\Lambda_r(t)\Big)$$

A Lyanpunov Function:

$$\mathcal{L}(\Lambda) = \sum_{i\in\mathcal{I}} m_i \log\Lambda_i - \sum_{j\in\mathcal{J}}\int_0^{\sum_{r:j\in r}\Lambda_r} p_j(y)\,dy$$

Dynamics

We want a network solve these optimisation problems **implicitly**

...Differential Equations?

A NETWORK ALGORITHM:

$$\frac{d}{dt}\Lambda_i(t) = \kappa\Big(m_i - \Lambda_i(t)\sum_{j\in i}\mu_j(t)\Big)$$

Where

$$\mu_j(t) = p_j\Big(\sum_{r:j\in r}\Lambda_r(t)\Big)$$

A Lyanpunov Function:

$$\mathcal{L}(\Lambda) = \sum_{i\in\mathcal{I}} m_i \log\Lambda_i - \sum_{j\in\mathcal{J}}\int_0^{\sum_{r:j\in r}\Lambda_r} p_j(y)dy$$

Therefore the NETWORK ALGORITHM solves NETWORK PROBLEM:

Dynamics

We want a network solve these optimisation problems **implicitly**

...Differential Equations?

A NETWORK ALGORITHM:

$$\frac{d}{dt}\Lambda_i(t) = \kappa\Big(m_i - \Lambda_i(t)\sum_{j\in i}\mu_j(t)\Big)$$

Where

$$\mu_j(t) = p_j\Big(\sum_{r:j\in r}\Lambda_r(t)\Big)$$

A Lyanpunov Function:

$$\mathcal{L}(\Lambda) = \sum_{i\in\mathcal{I}} m_i \log\Lambda_i - \sum_{j\in\mathcal{J}}\int_0^{\sum_{r:j\in r}\Lambda_r} p_j(y)dy$$

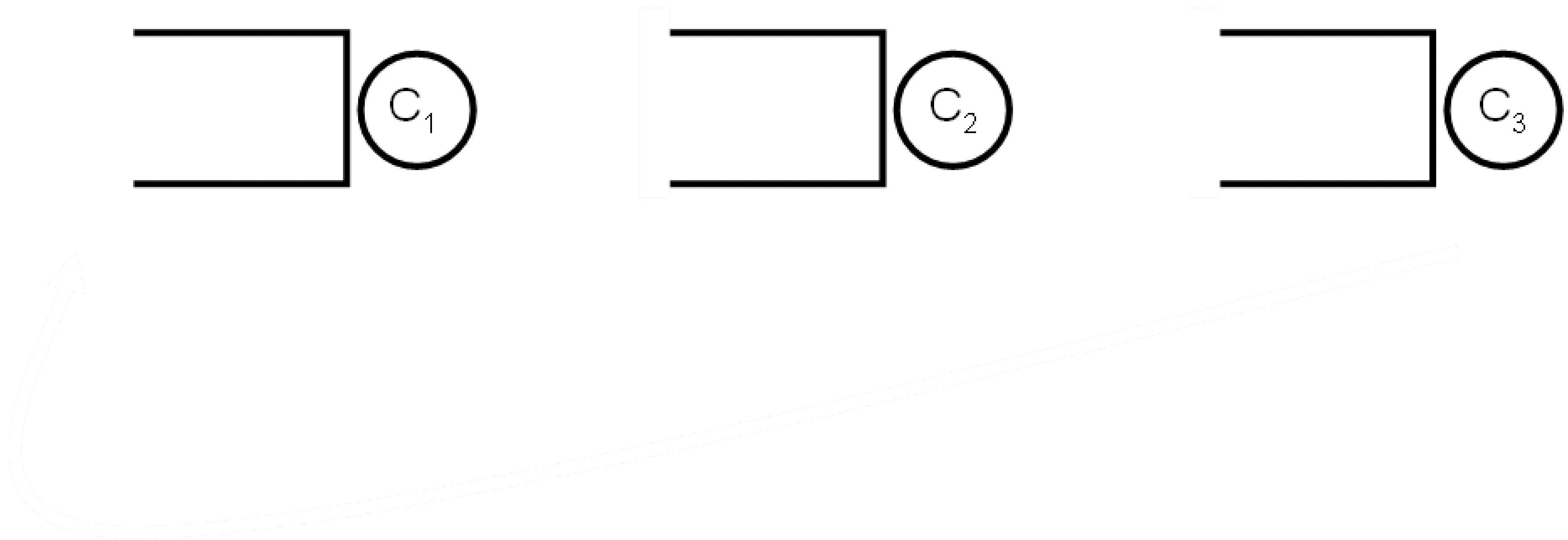
Therefore the NETWORK ALGORITHM solves NETWORK PROBLEM:

$$\max \quad \sum_{i\in\mathcal{I}} m_i \log\Lambda_i - \sum_{j\in\mathcal{J}}\int_0^{\sum_{r:j\in r}\Lambda_r} p_j(y)dy$$

$$\text{over} \quad \Lambda_i \geq 0, \quad i\in\mathcal{I}.$$

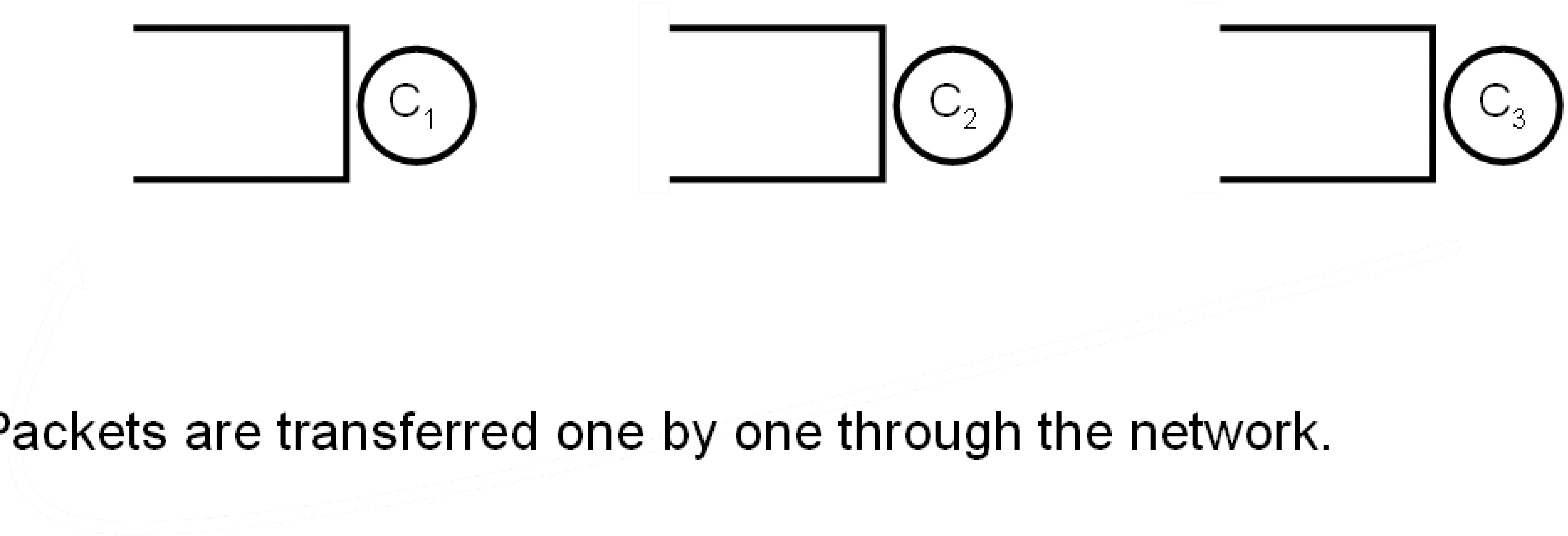
We can add queueing dynamics to the NETWORK PROBLEM…

A Closed Multi-class Queueing Network



REF: Kelly ‘79

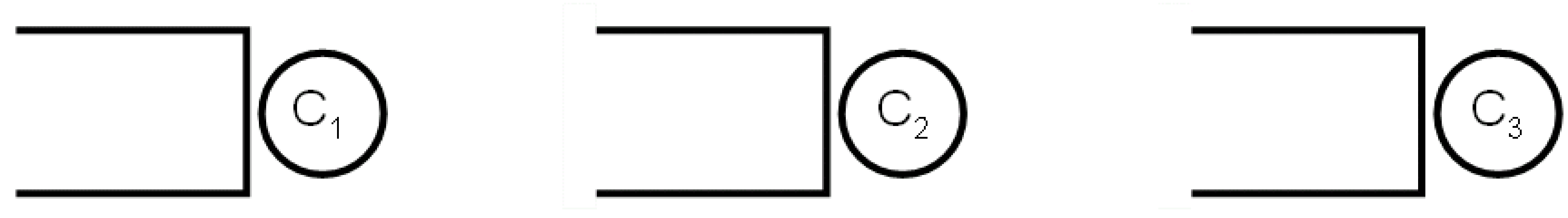
A Closed Multi-class Queueing Network



- Packets are transferred one by one through the network.

REF: Kelly '79

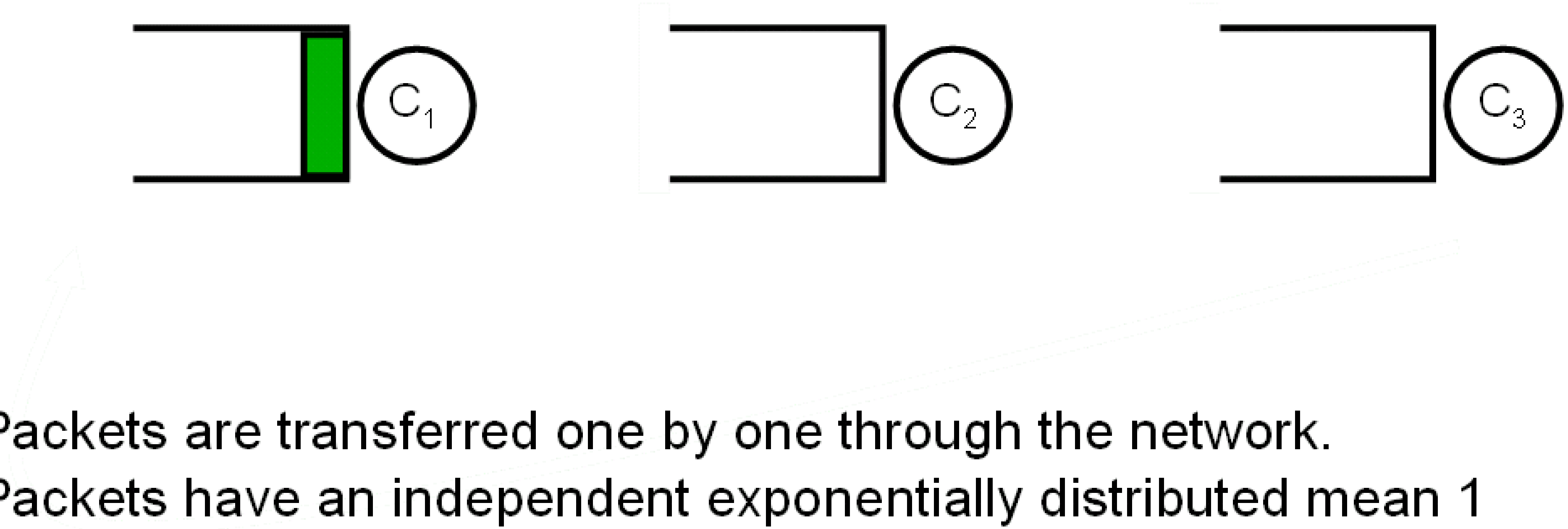
A Closed Multi-class Queueing Network



- Packets are transferred one by one through the network.
- Packets have an independent exponentially distributed mean 1 service requirement at each queue.

REF: Kelly ‘79

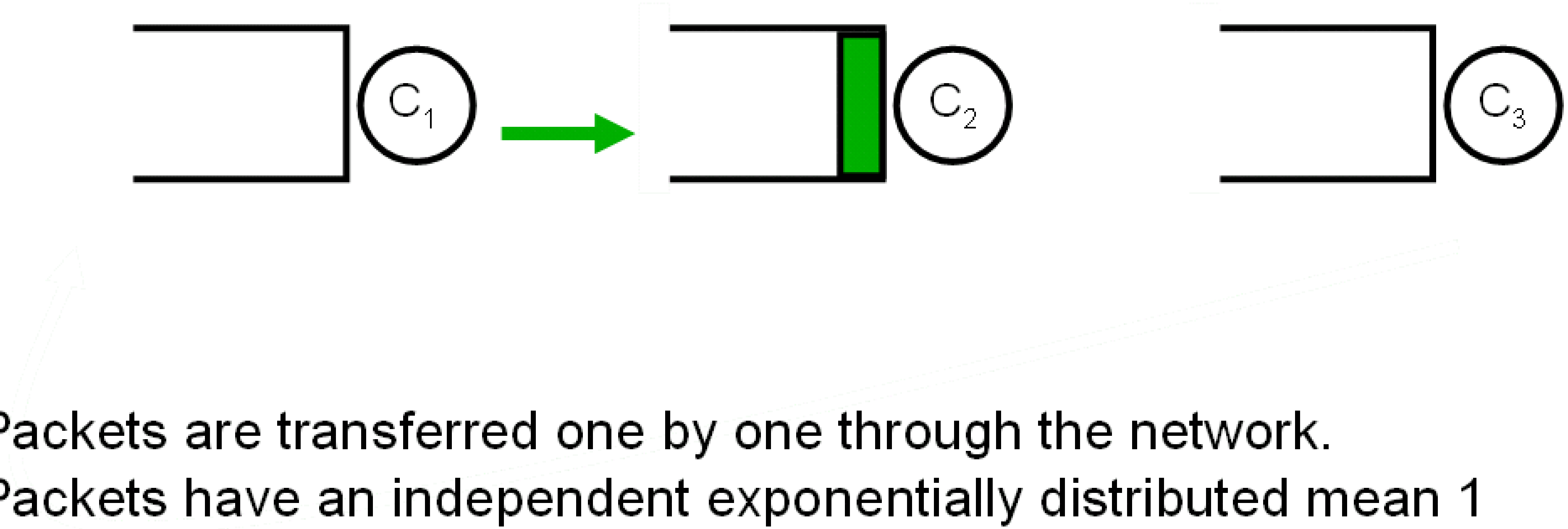
A Closed Multi-class Queueing Network



- Packets are transferred one by one through the network.
- Packets have an independent exponentially distributed mean 1 service requirement at each queue.

REF: Kelly '79

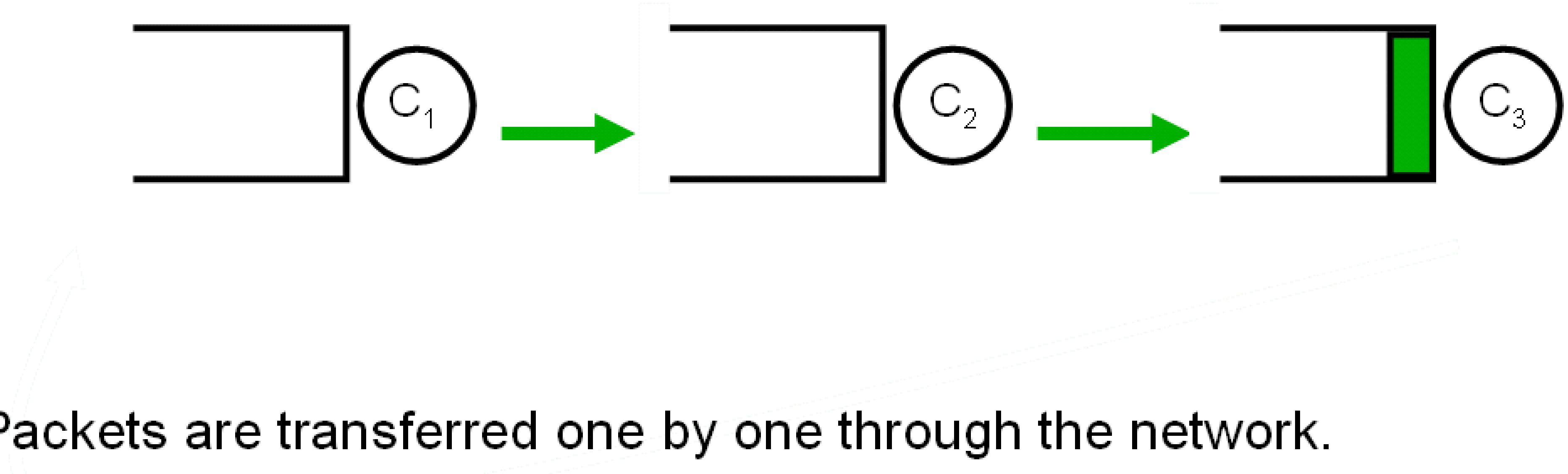
A Closed Multi-class Queueing Network



- Packets are transferred one by one through the network.
- Packets have an independent exponentially distributed mean 1 service requirement at each queue.

REF: Kelly '79

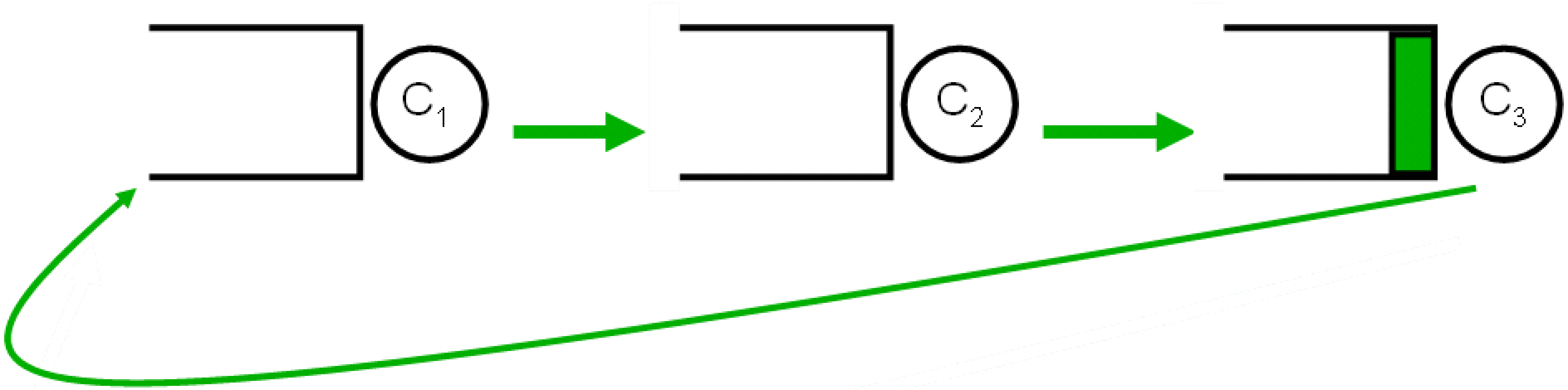
A Closed Multi-class Queueing Network



- Packets are transferred one by one through the network.
- Packets have an independent exponentially distributed mean 1 service requirement at each queue.

REF: Kelly ‘79

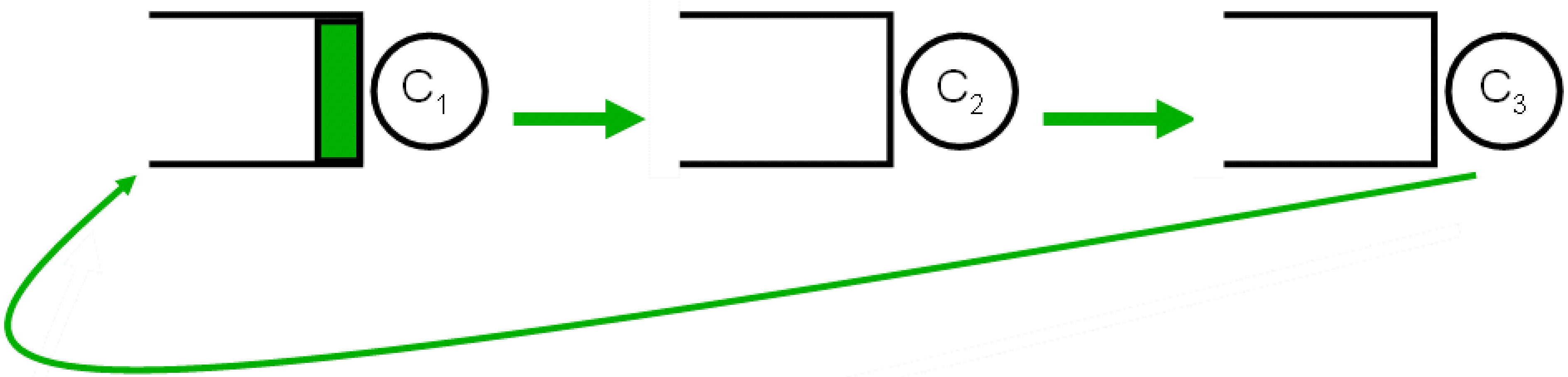
A Closed Multi-class Queueing Network



- Packets are transferred one by one through the network.
- Packets have an independent exponentially distributed mean 1 service requirement at each queue.

REF: Kelly '79

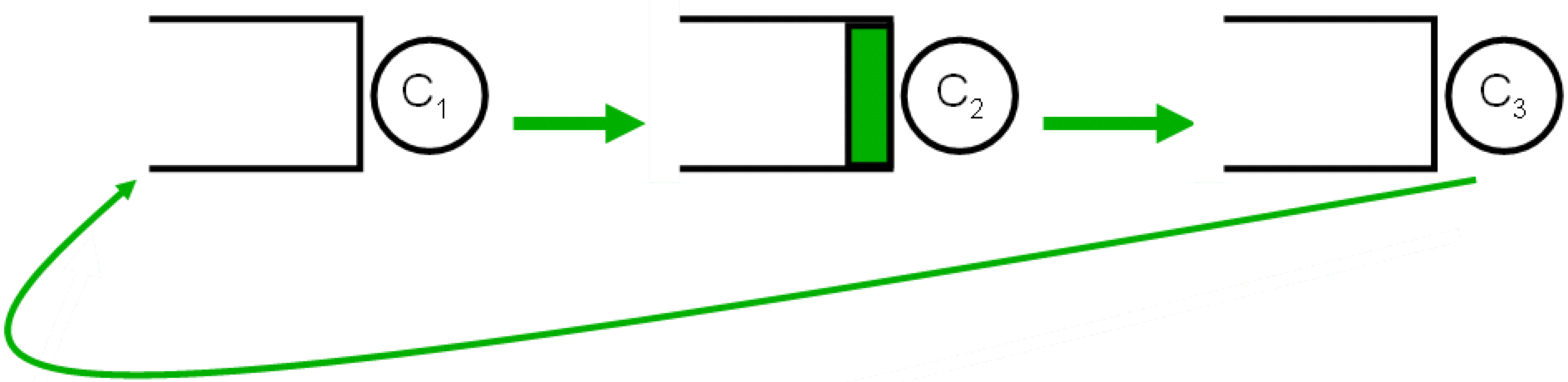
A Closed Multi-class Queueing Network



- Packets are transferred one by one through the network.
- Packets have an independent exponentially distributed mean 1 service requirement at each queue.

REF: Kelly '79

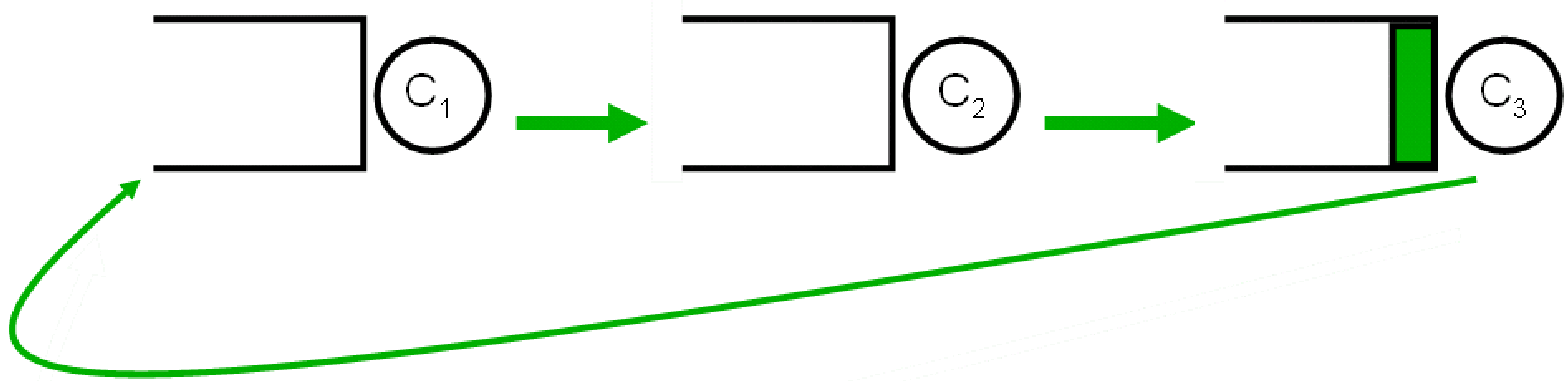
A Closed Multi-class Queueing Network



• Packets are transferred one by one through the network.
• Packets have an independent exponentially distributed mean 1 service requirement at each queue.

REF: Kelly '79

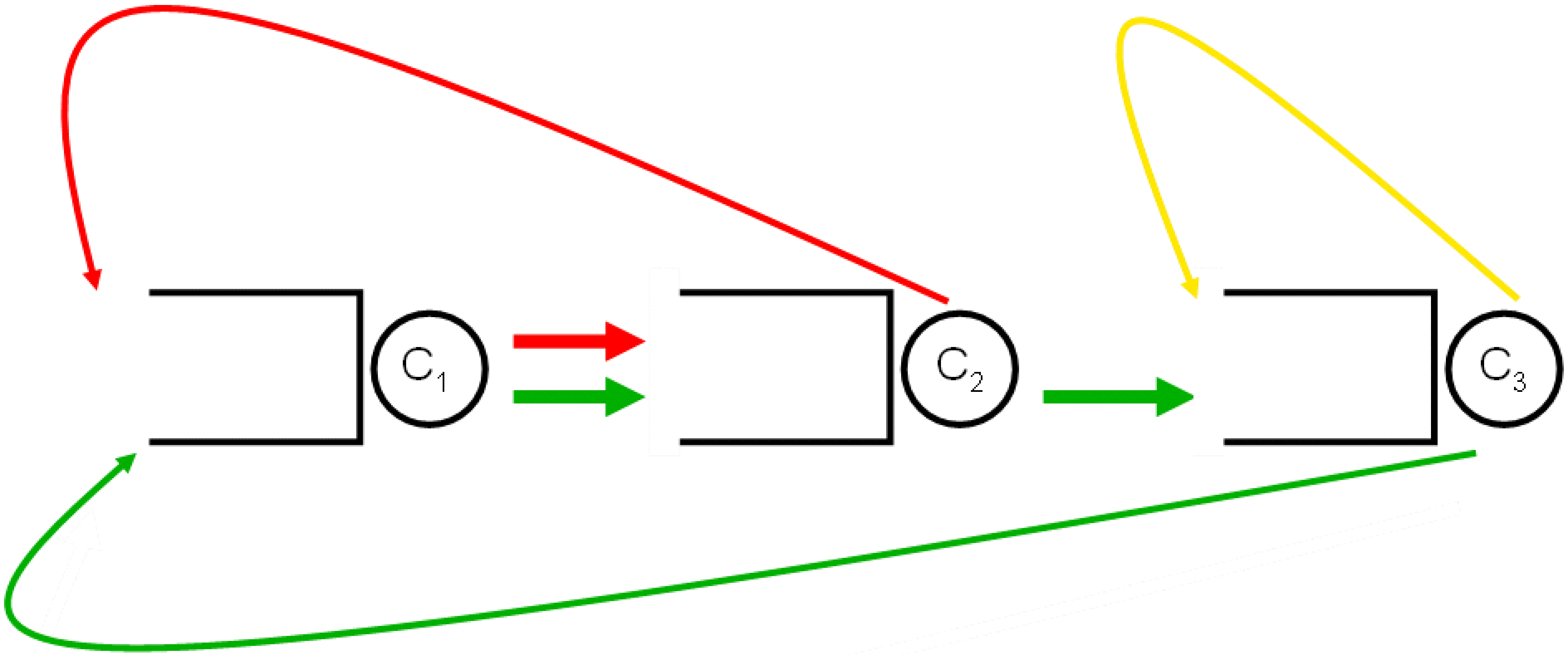
A Closed Multi-class Queueing Network



- Packets are transferred one by one through the network.
- Packets have an independent exponentially distributed mean 1 service requirement at each queue.

REF: Kelly '79

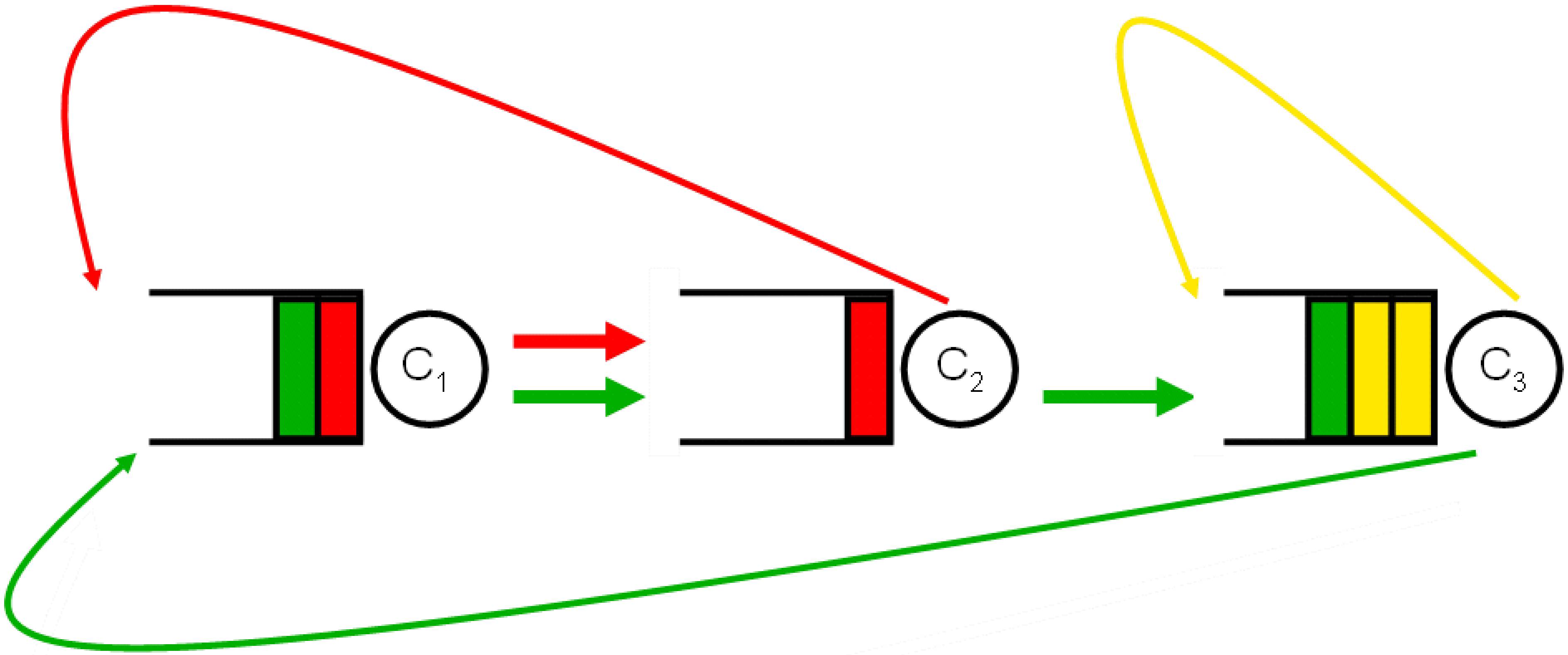
A Closed Multi-class Queueing Network



- Packets are transferred one by one through the network.
- Packets have an independent exponentially distributed mean 1 service requirement at each queue.

REF: Kelly '79

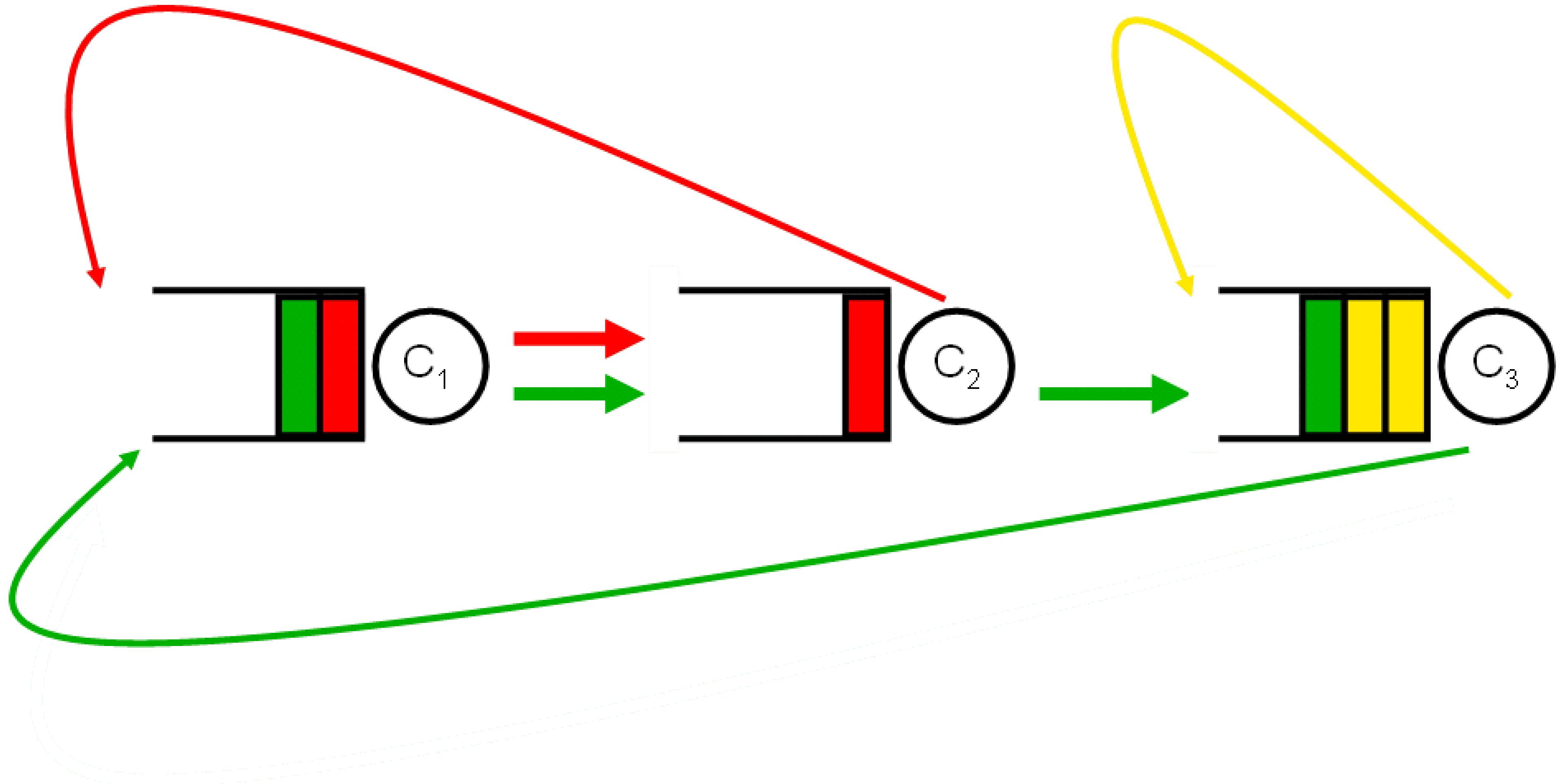
A Closed Multi-class Queueing Network



- Packets are transferred one by one through the network.
- Packets have an independent exponentially distributed mean 1 service requirement at each queue.

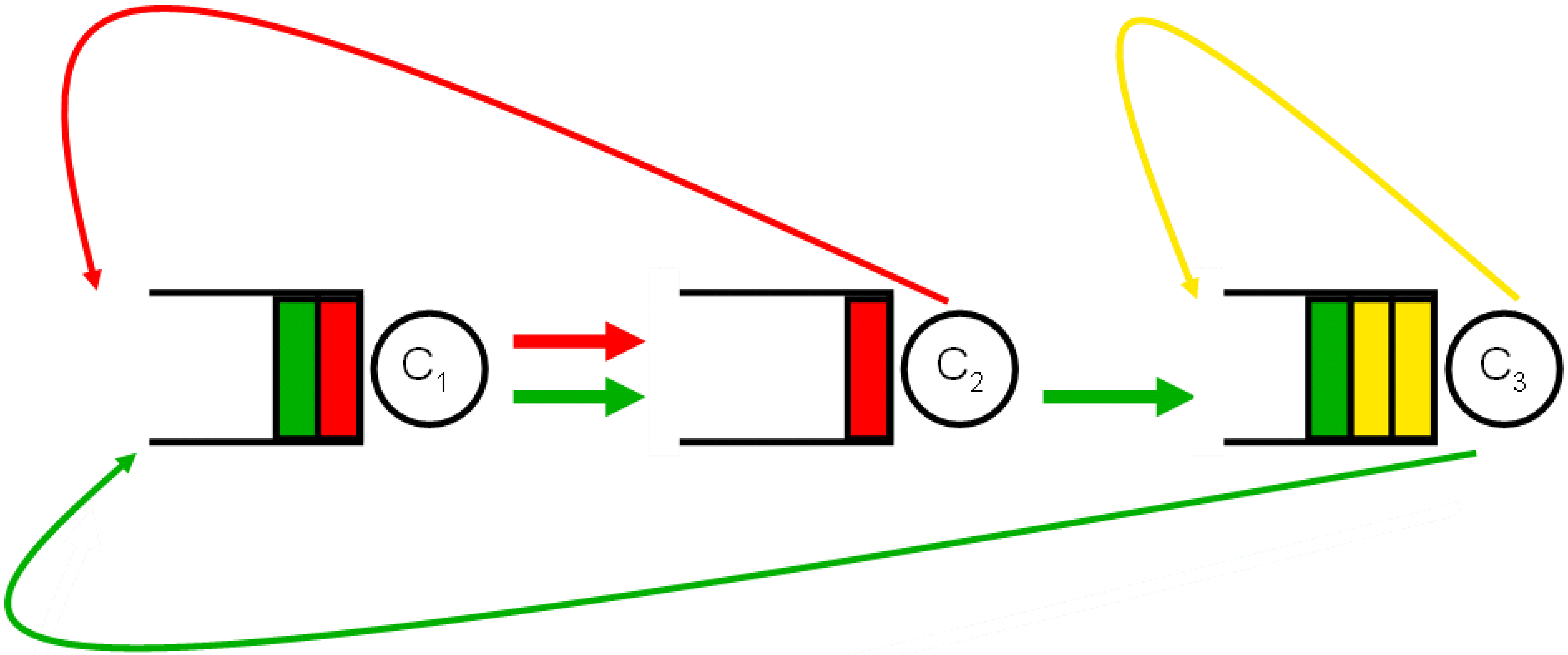
REF: Kelly '79

A Closed Multi-class Queueing Network



REF: Kelly '79

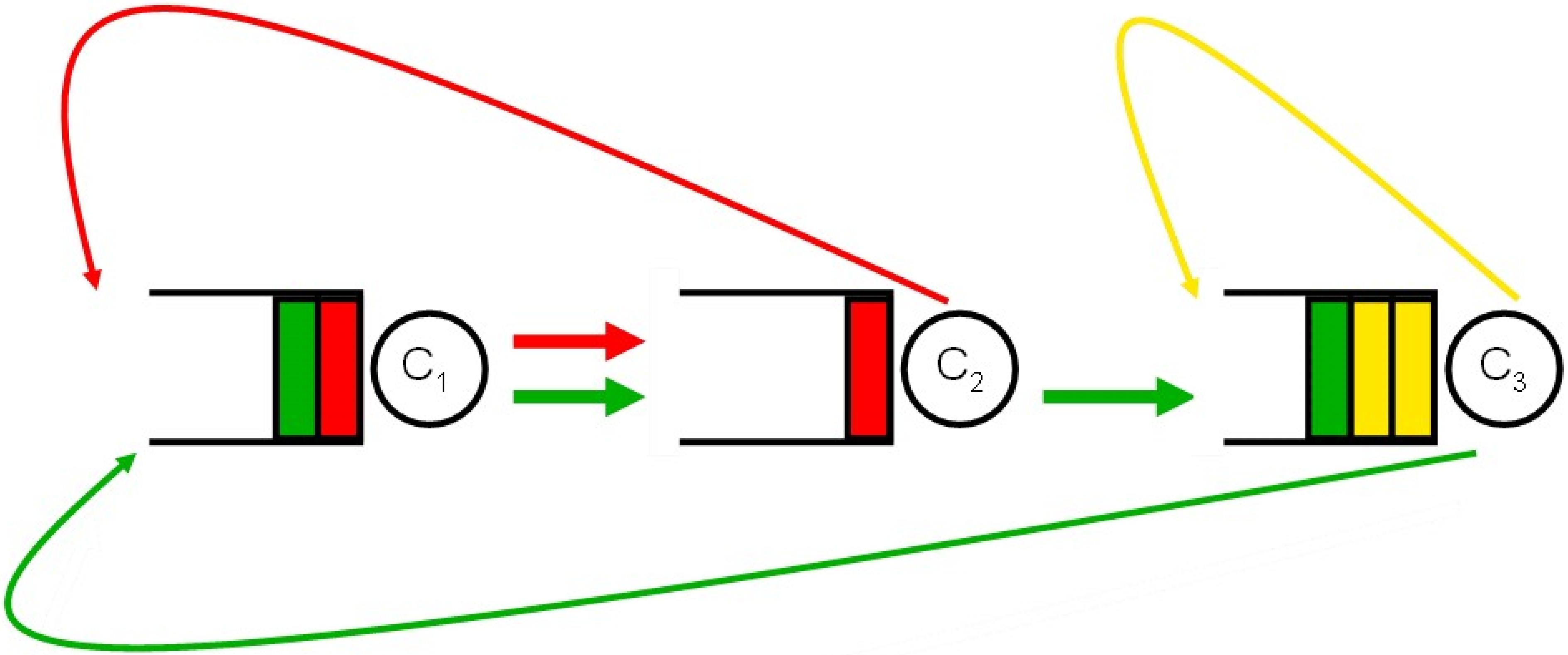
A Closed Multi-class Queueing Network



Stationary Distribution:

REF: Kelly '79

A Closed Multi-class Queueing Network

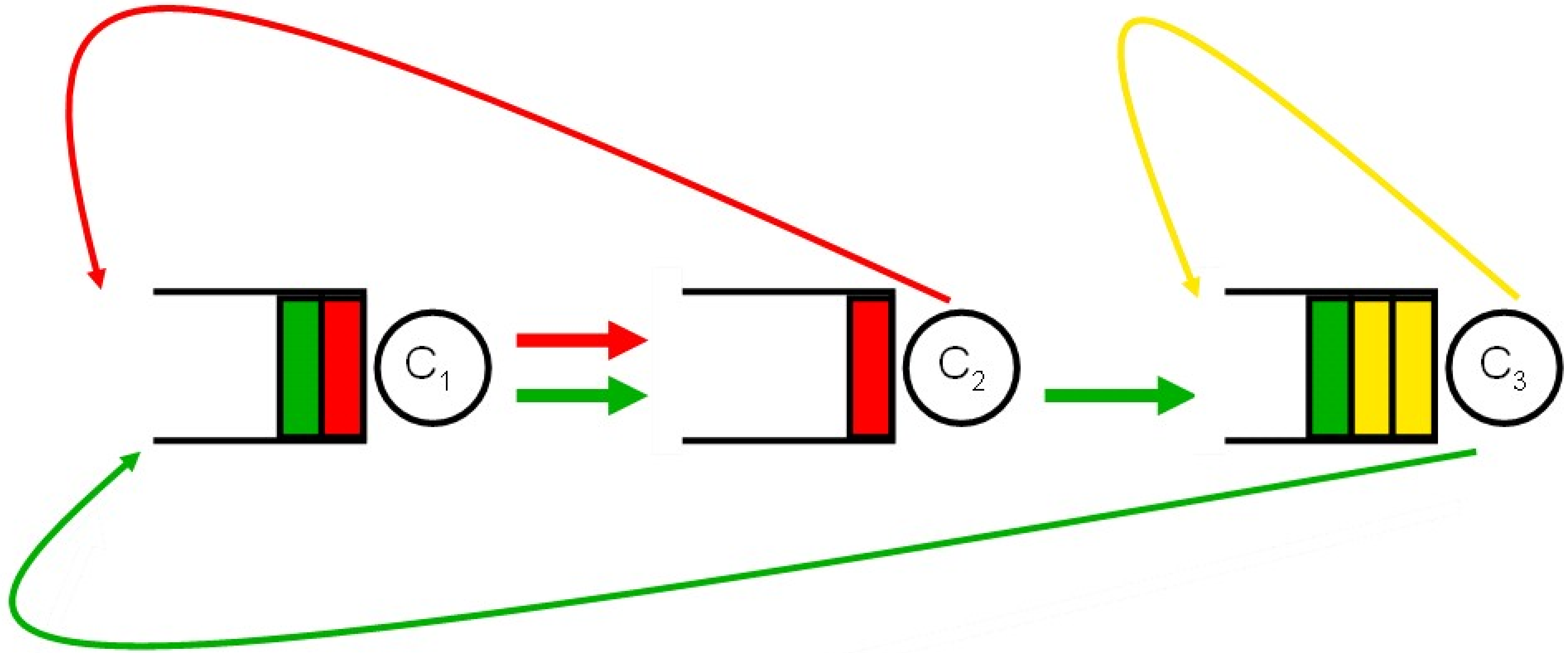


Stationary Distribution:

$$\mathbb{P}(M = m) = \frac{1}{B_n} \prod_{j \in \mathcal{J}} \left(\binom{m_j}{m_{ji} \; : i \ni j} \prod_{i : j \in i} \left(\frac{\rho_i}{C_j} \right)^{m_{ji}} \right)$$

REF: Kelly '79

A Closed Multi-class Queueing Network



Stationary Distribution:

$$\mathbb{P}(M = m) = \frac{1}{B_n} \prod_{j \in \mathcal{J}} \left(\binom{m_j}{m_{ji} \;:\; i \ni j} \prod_{i : j \in i} \left(\frac{\rho_i}{C_j} \right)^{m_{ji}} \right)$$

$$M_{ji} = \#\text{ route } i \text{ packets at queue } j.$$

REF: Kelly ‘79

A Closed Multi-class Queueing Network

How do we form a large deviations connection with proportional fairness:

REF: Pittel '79

A Closed Multi-class Queueing Network

How do we form a large deviations connection with proportional fairness:

- Consider the stationary distribution of the number of packets at each queue:

REF: Pittel '79

A Closed Multi-class Queueing Network

How do we form a large deviations connection with proportional fairness:

- Consider the stationary distribution of the number of packets at each queue:

$$\mathbb{P}(M = m) = B^{-1} \prod_{j \in \mathcal{J}} \left(\binom{m_j}{m_{ji} \ : i \ni j} \prod_{i \in I} \left(\frac{\rho_i}{C_j} \right)^{m_{ji}} \right),$$

REF: Pittel '79

A Closed Multi-class Queueing Network

How do we form a large deviations connection with proportional fairness:

- Consider the stationary distribution of the number of packets at each queue:

$$\mathbb{P}(M=m) = B^{-1} \prod_{j \in \mathcal{J}} \left(\binom{m_j}{m_{ji} \; : i \ni j} \prod_{i \in I} \left(\frac{\rho_i}{C_j} \right)^{m_{ji}} \right),$$

- Stirling's approximation:

REF: Pittel '79

A Closed Multi-class Queueing Network

How do we form a large deviations connection with proportional fairness:

- Consider the stationary distribution of the number of packets at each queue:

$$\mathbb{P}(M = m) = B^{-1} \prod_{j \in \mathcal{J}} \left(\binom{m_j}{m_{ji} \; : i \ni j} \prod_{i \in I} \left(\frac{\rho_i}{C_j} \right)^{m_{ji}} \right),$$

- Stirling's approximation:

$$\lim_{c \to \infty} \frac{1}{c} \log \mathbb{P}(M = cm) = - \sum_{\substack{(j,i) \in \mathcal{K}: \\ m_j > 0}} m_{ji} \log \frac{m_{ji} C_j}{m_j \rho_i}$$

REF: Pittel '79

A Closed Multi-class Queueing Network

How do we form a large deviations connection with proportional fairness:

- Consider the stationary distribution of the number of packets at each queue:

$$\mathbb{P}(M = m) = B^{-1} \prod_{j \in \mathcal{J}} \left(\binom{m_j}{m_{ji} \, : i \ni j} \prod_{i \in I} \left(\frac{\rho_i}{C_j} \right)^{m_{ji}} \right),$$

- Stirling's approximation:

$$\lim_{c \to \infty} \frac{1}{c} \log \mathbb{P}(M = cm) = - \sum_{\substack{(j,i) \in \mathcal{K}: \\ m_j > 0}} m_{ji} \log \frac{m_{ji} C_j}{m_j \rho_i}$$

- Thus for the equilibrium distribution for the number of packets in transfer:

REF: Pittel '79

A Closed Multi-class Queueing Network

How do we form a large deviations connection with proportional fairness:

- Consider the stationary distribution of the number of packets at each queue:

$$\mathbb{P}(M=m) = B^{-1} \prod_{j \in \mathcal{J}} \left(\binom{m_j}{m_{ji} \; : i \ni j} \prod_{i \in I} \left(\frac{\rho_i}{C_j} \right)^{m_{ji}} \right),$$

- Stirling's approximation:

$$\lim_{c \to \infty} \frac{1}{c} \log \mathbb{P}(M = cm) = - \sum_{\substack{(j,i) \in \mathcal{K}: \\ m_j > 0}} m_{ji} \log \frac{m_{ji} C_j}{m_j \rho_i}$$

- Thus for the equilibrium distribution for the number of packets in transfer:

$$\lim_{c \to \infty} \frac{1}{c} \log \mathbb{P}(M = m) = -\min_{m \geq 0} \sum_{j,i} m_{ji} \log \frac{m_{ji} C_j}{m_j \rho_i}$$

$$\text{subject to} \quad \sum_{j \in i} m_{ji} = \bar{m}_i, \quad i \in \mathcal{I}.$$

REF: Pittel '79

A Closed Multi-class Queueing Network

A Closed Multi-class Queueing Network

- Consider the rate function found:

A Closed Multi-class Queueing Network

- Consider the rate function found:

$$\min_{m \geq 0} \quad \sum_{j,i} m_{ji} \log \frac{m_{ji} C_j}{m_j \rho_i} \quad \text{subject to} \quad \sum_{j \in i} m_{ji} = \bar{m}_i, \quad i \in \mathcal{I}.$$

A Closed Multi-class Queueing Network

- Consider the rate function found:

$$\min_{m \geq 0} \quad \sum_{j,i} m_{ji} \log \frac{m_{ji} C_j}{m_j \rho_i} \quad \text{subject to} \quad \sum_{j \in i} m_{ji} = \bar{m}_i, \quad i \in \mathcal{I}.$$

- Calculate its dual:

A Closed Multi-class Queueing Network

- Consider the rate function found:

$$\min_{m\geq 0} \quad \sum_{j,i} m_{ji} \log \frac{m_{ji} C_j}{m_j \rho_i} \quad \text{subject to} \quad \sum_{j \in i} m_{ji} = \bar{m}_i, \quad i \in \mathcal{I}.$$

- Calculate its dual:

$$\max_{\Lambda \in \mathbb{R}_+^I} \quad \sum_{i \in \mathcal{I}} \bar{m}_i \log \Lambda_i \quad \text{subject to} \quad \sum_{i : j \in i} \Lambda_i \leq C_j, \quad j \in \mathcal{J}.$$

A Closed Multi-class Queueing Network

- Consider the rate function found:

$$\min_{m \geq 0} \quad \sum_{j,i} m_{ji} \log \frac{m_{ji} C_j}{m_j \rho_i} \quad \text{subject to} \quad \sum_{j \in i} m_{ji} = \bar{m}_i, \quad i \in \mathcal{I}.$$

- Calculate its dual:

$$\max_{\Lambda \in \mathbb{R}_+^I} \quad \sum_{i \in \mathcal{I}} \bar{m}_i \log \Lambda_i \quad \text{subject to} \quad \sum_{i : j \in i} \Lambda_i \leq C_j, \quad j \in \mathcal{J}.$$

The dual coincides with NETWORK problem.

A Closed Multi-class Queueing Network

- Consider the rate function found:

$$\min_{m\geq 0} \quad \sum_{j,i} m_{ji} \log \frac{m_{ji} C_j}{m_j \rho_i} \quad \text{subject to} \quad \sum_{j\in i} m_{ji} = \bar{m}_i, \quad i \in \mathcal{I}.$$

- Calculate its dual:

$$\max_{\Lambda\in\mathbb{R}_+^I} \quad \sum_{i\in\mathcal{I}} \bar{m}_i \log \Lambda_i \quad \text{subject to} \quad \sum_{i:j\in i} \Lambda_i \leq C_j, \quad j \in \mathcal{J}.$$

The dual coincides with NETWORK problem.

This is sufficient to show that throughput's converge:

A Closed Multi-class Queueing Network

- Consider the rate function found:

$$\min_{m \geq 0} \quad \sum_{j,i} m_{ji} \log \frac{m_{ji} C_j}{m_j \rho_i} \quad \text{subject to} \quad \sum_{j \in i} m_{ji} = \bar{m}_i, \quad i \in \mathcal{I}.$$

- Calculate its dual:

$$\max_{\Lambda \in \mathbb{R}^I_+} \quad \sum_{i \in \mathcal{I}} \bar{m}_i \log \Lambda_i \quad \text{subject to} \quad \sum_{i : j \in i} \Lambda_i \leq C_j, \quad j \in \mathcal{J}.$$

The dual coincides with NETWORK problem.

This is sufficient to show that throughput's converge:

$$\Lambda_i^{SN}(c\bar{m}) \xrightarrow[c \to \infty]{} \Lambda_i^{PF}(\bar{m})$$

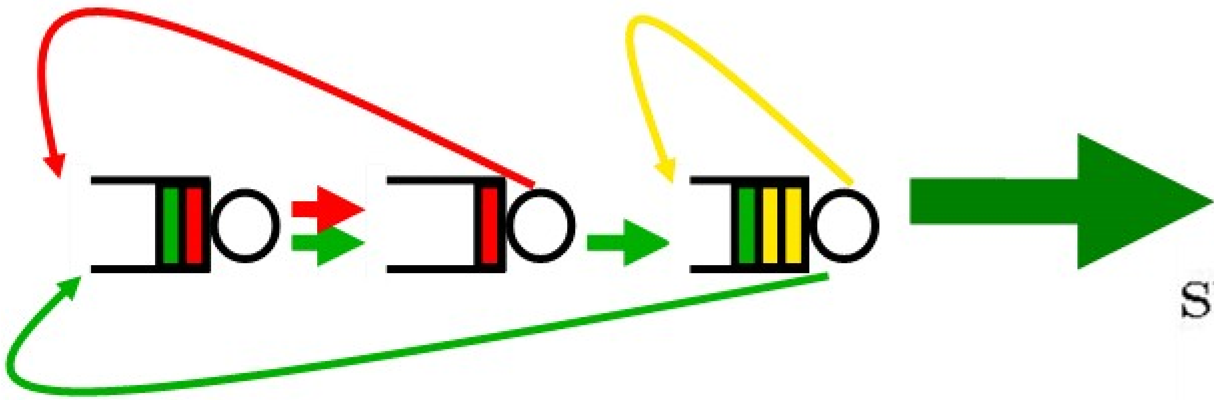
So…

So…

The NETWORK PROBLEM is solved by a Closed Queueing Network

So...

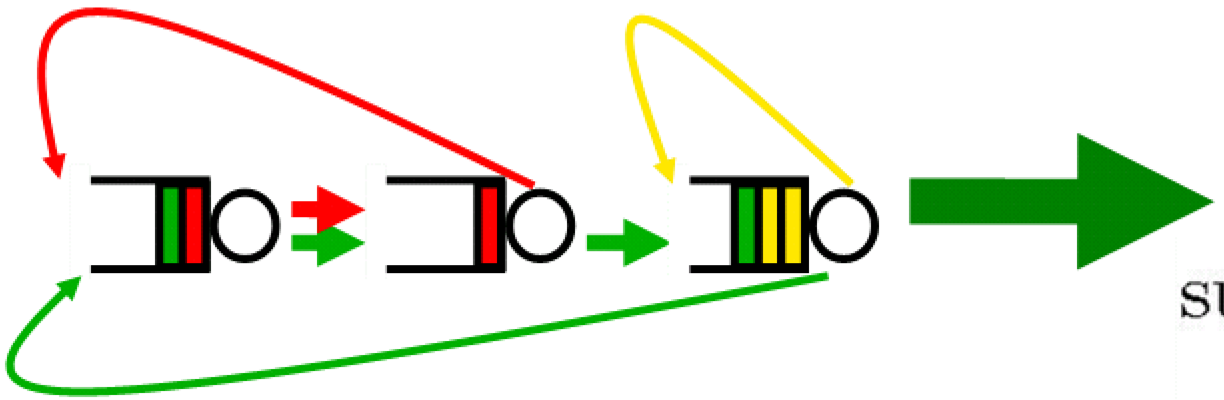
The NETWORK PROBLEM is solved by a Closed Queueing Network



$$\max \sum_{i \in \mathcal{I}} \bar{m}_i \log \Lambda_i$$
$$\text{subject to} \quad \sum_{i : j \in i} \Lambda_i \le C_j,$$

So...

The NETWORK PROBLEM is solved by a Closed Queueing Network

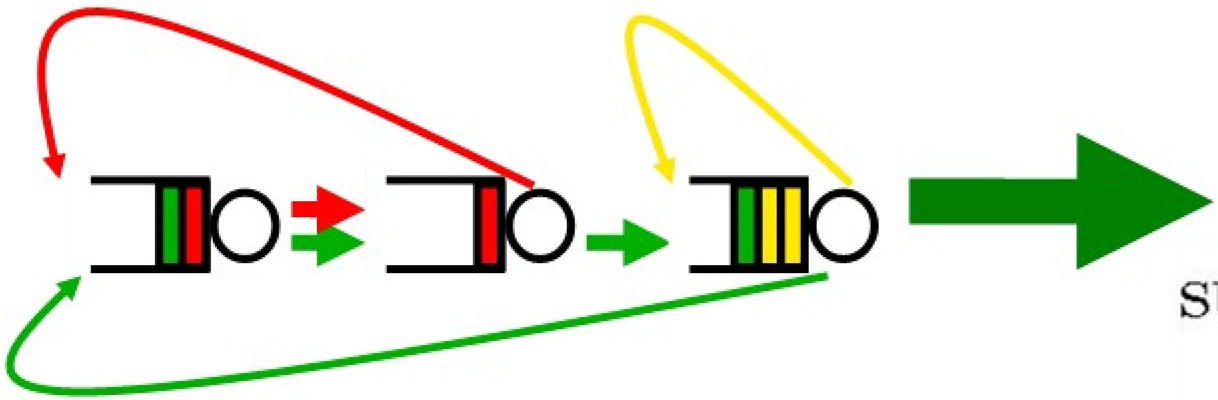


$$\max \sum_{i \in \mathcal{I}} \bar{m}_i \log \Lambda_i$$
$$\text{subject to} \quad \sum_{i : j \in i} \Lambda_i \leq C_j,$$

What about the USER PROBLEM?

So...

The NETWORK PROBLEM is solved by a Closed Queueing Network



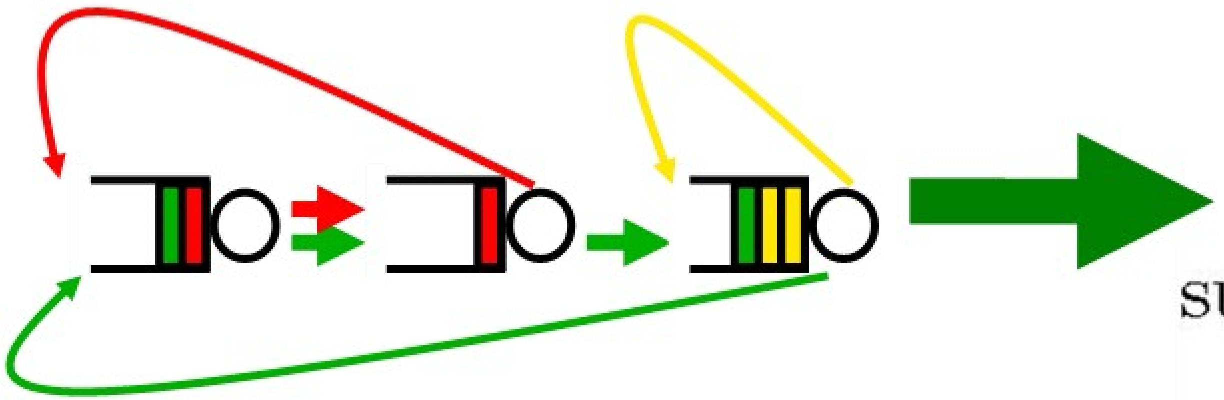
$$\max \sum_{i \in \mathcal{I}} \bar{m}_i \log \Lambda_i$$
$$\text{subject to} \quad \sum_{i: j \in i} \Lambda_i \le C_j,$$

What about the USER PROBLEM?

$$\max \quad U_i\left(\frac{\bar{m}_i}{q_i}\right) - \bar{m}_i$$
$$\text{over} \quad \bar{m}_i \ge 0, \quad i \in \mathcal{I}.$$

So...

The NETWORK PROBLEM is solved by a Closed Queueing Network



$$\max \quad \sum_{i \in \mathcal{I}} \bar{m}_i \log \Lambda_i$$
$$\text{subject to} \quad \sum_{i: j \in i} \Lambda_i \le C_j,$$

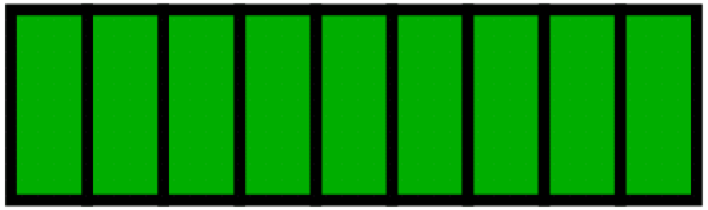
What about the USER PROBLEM?

$$\max \quad U_i\left(\frac{\bar{m}_i}{q_i}\right) - \bar{m}_i$$
$$\text{over} \quad \bar{m}_i \ge 0, \quad i \in \mathcal{I}.$$

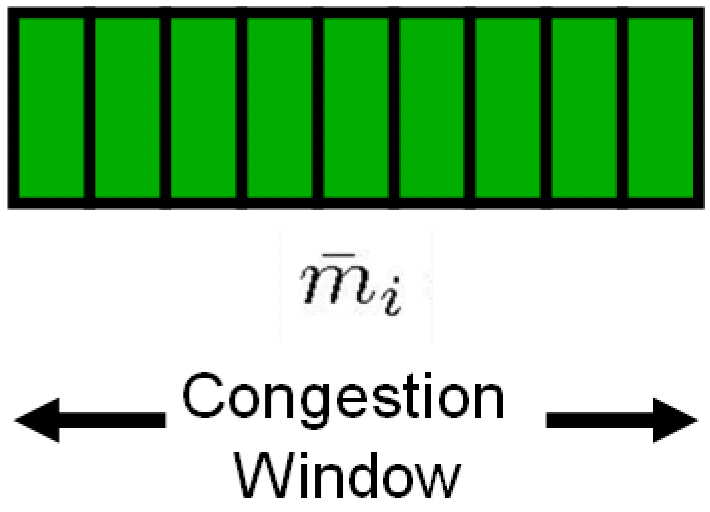
Looks like a Legendre-Fenchel Transform...

Congestion Windows

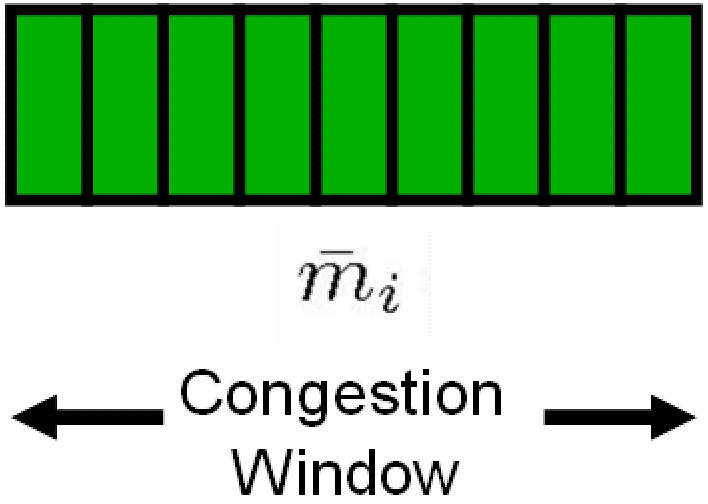
Congestion Windows



Congestion Windows



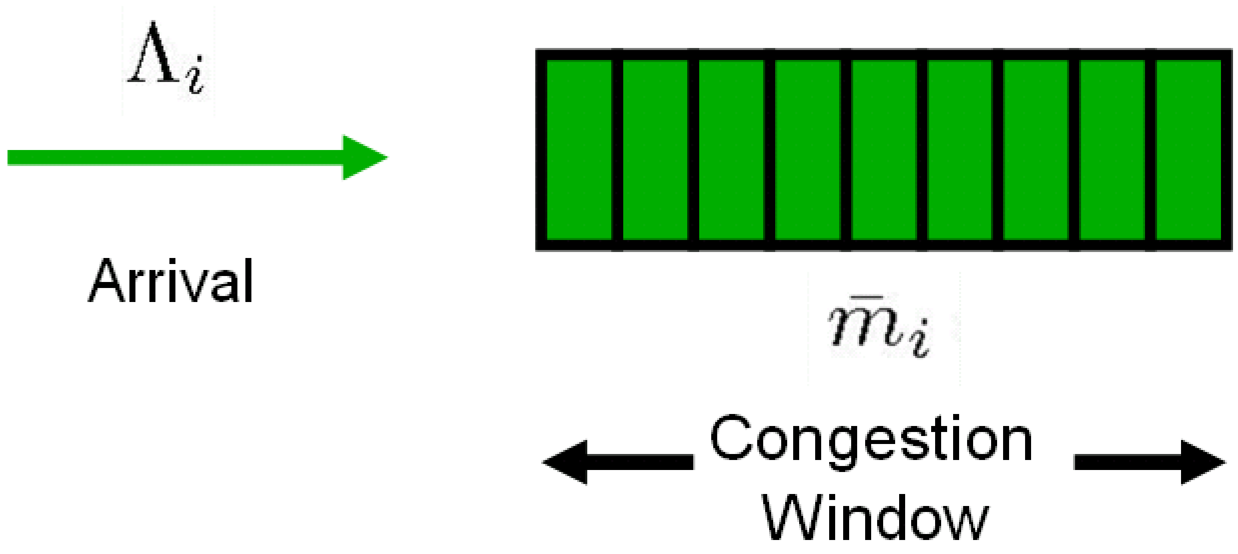
Congestion Windows



Arrivals acknowledges packets.

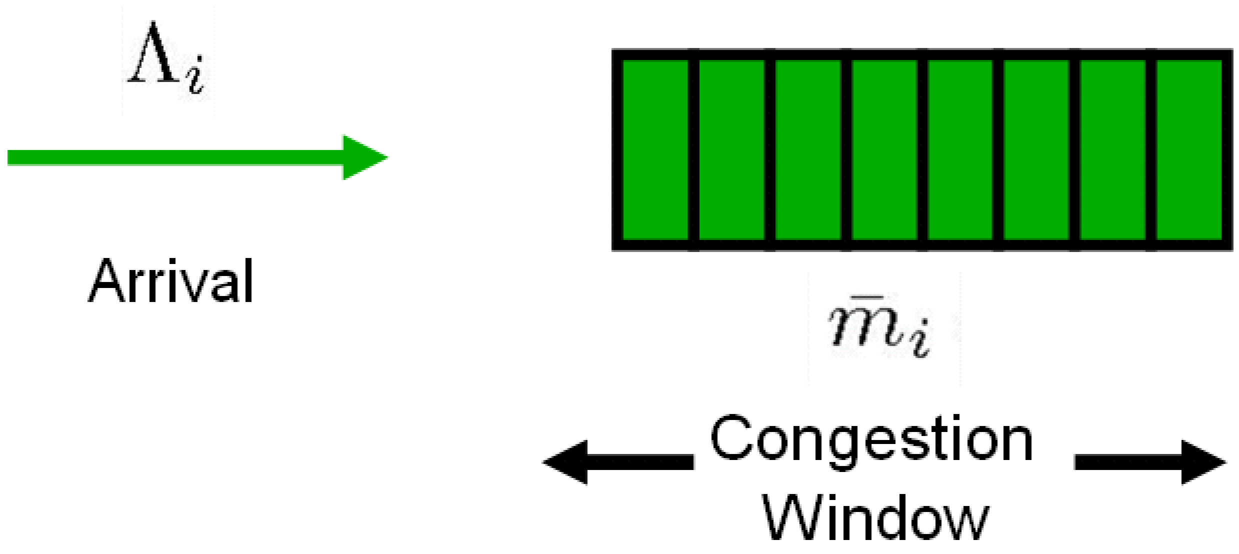
Decrease the congestion window size.

Congestion Windows



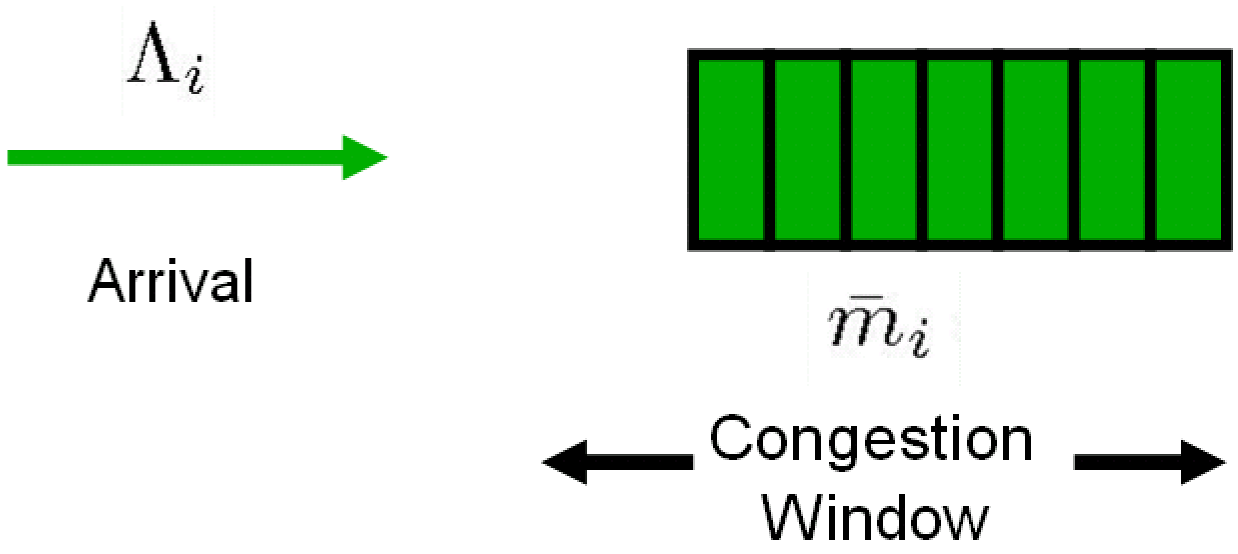
Arrivals acknowledges packets.
Decrease the congestion window size.

Congestion Windows



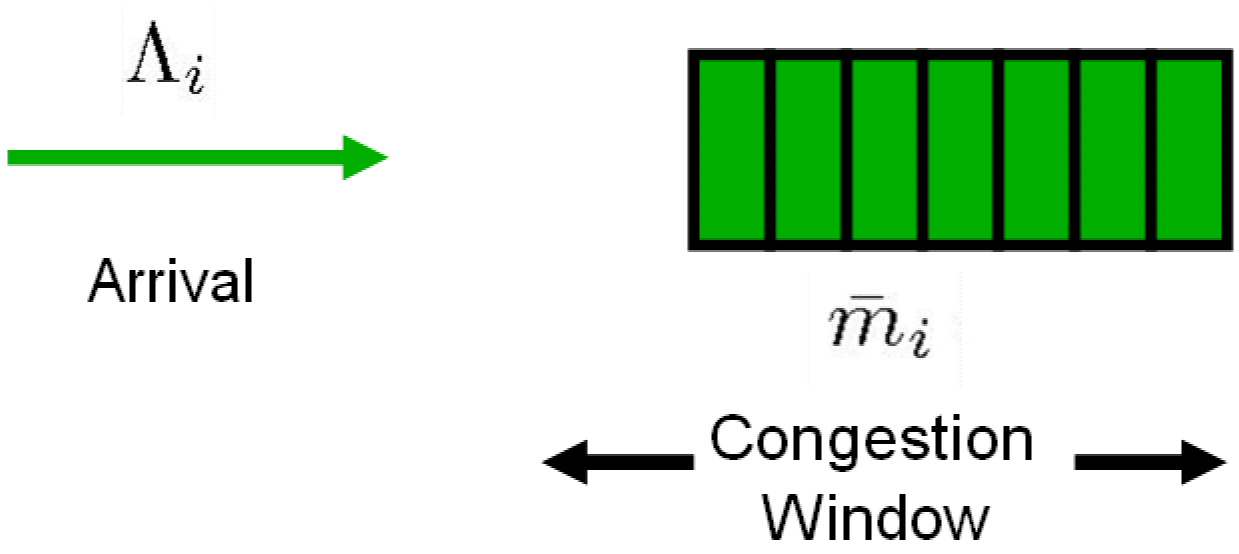
Arrivals acknowledges packets.
Decrease the congestion window size.

Congestion Windows



Arrivals acknowledges packets.
Decrease the congestion window size.

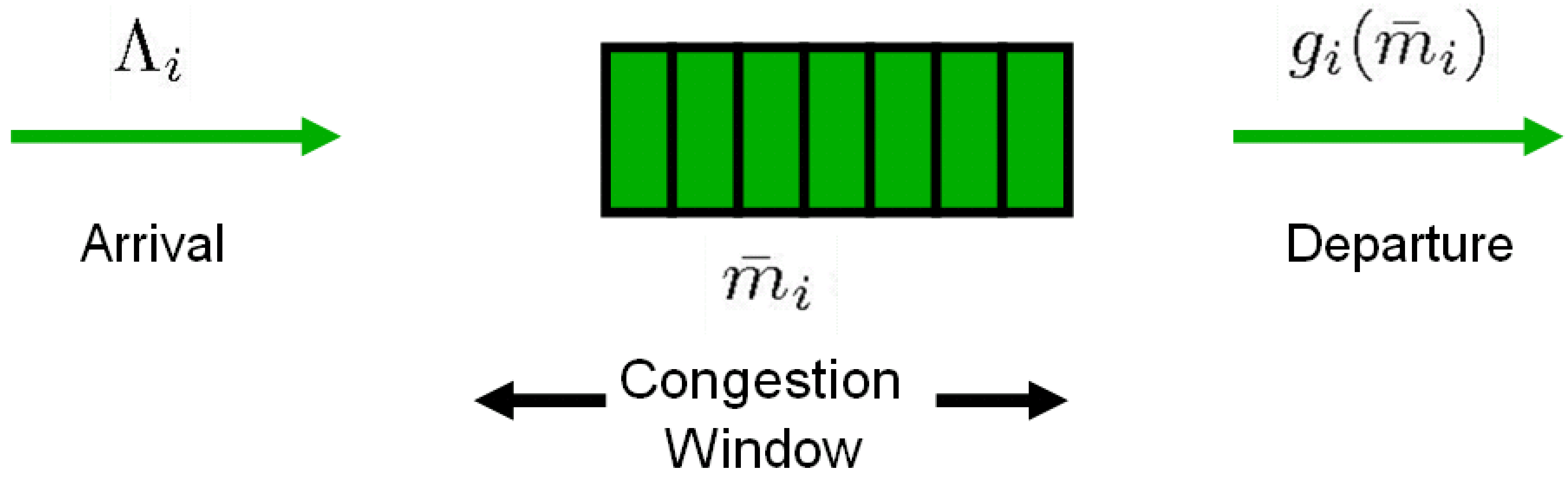
Congestion Windows



Arrivals acknowledges packets.
Decrease the congestion window size.

Departures send packets into the network.
Increase the congestion window.

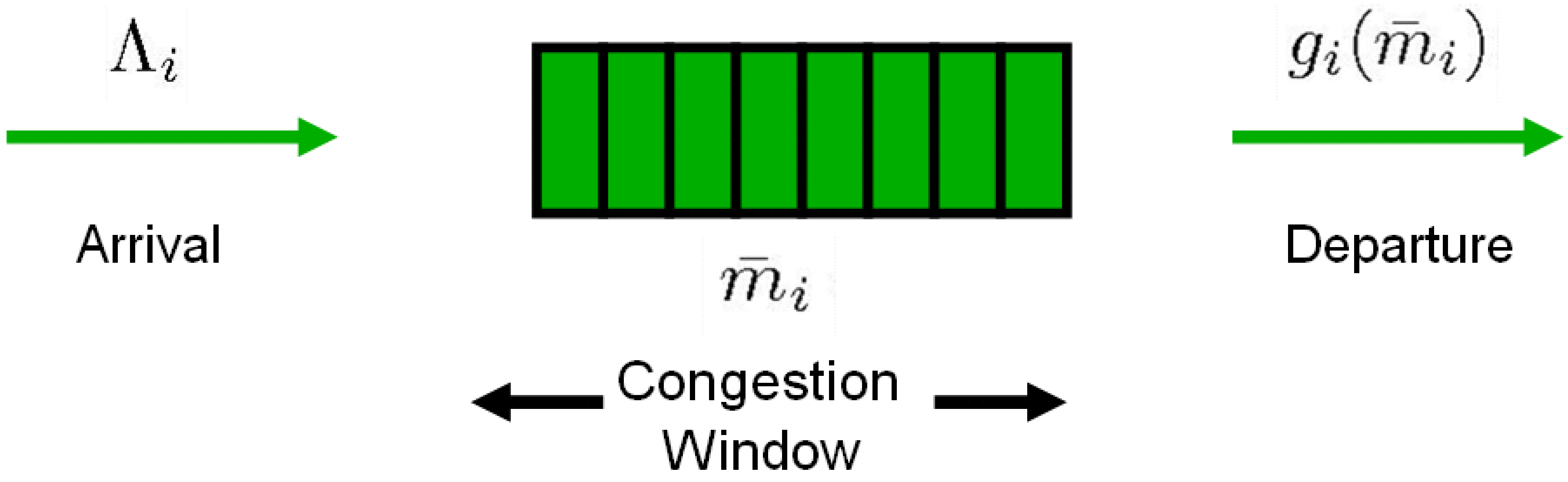
Congestion Windows



Arrivals acknowledges packets.
Decrease the congestion window size.

Departures send packets into the network.
Increase the congestion window.

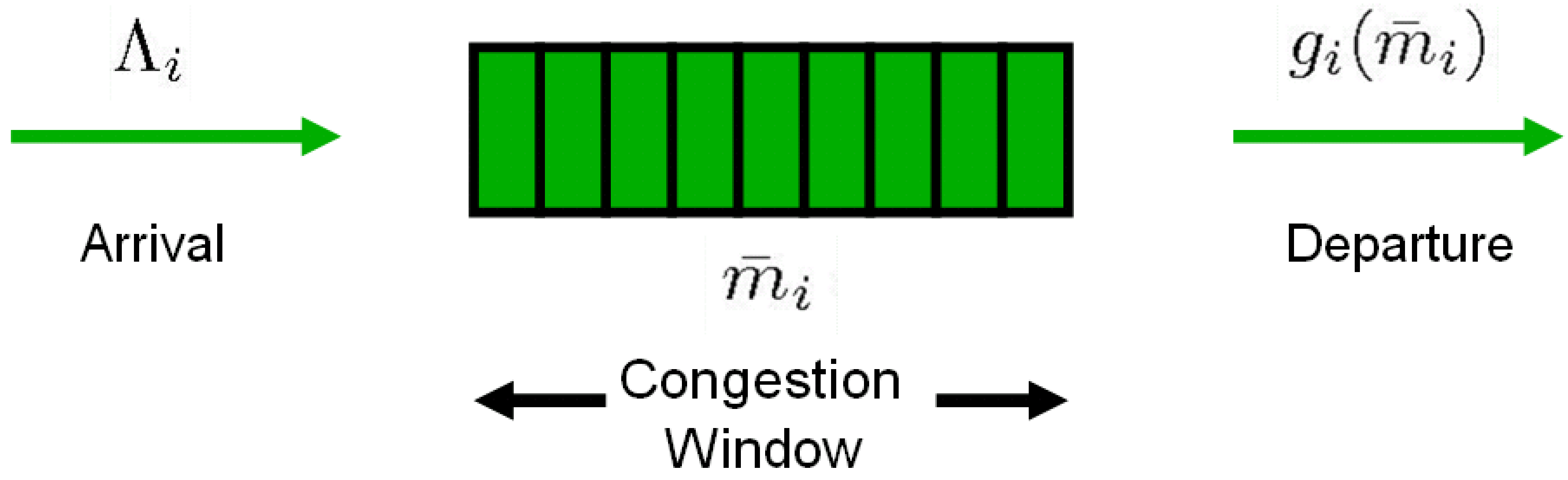
Congestion Windows



Arrivals acknowledges packets.
Decrease the congestion window size.

Departures send packets into the network.
Increase the congestion window.

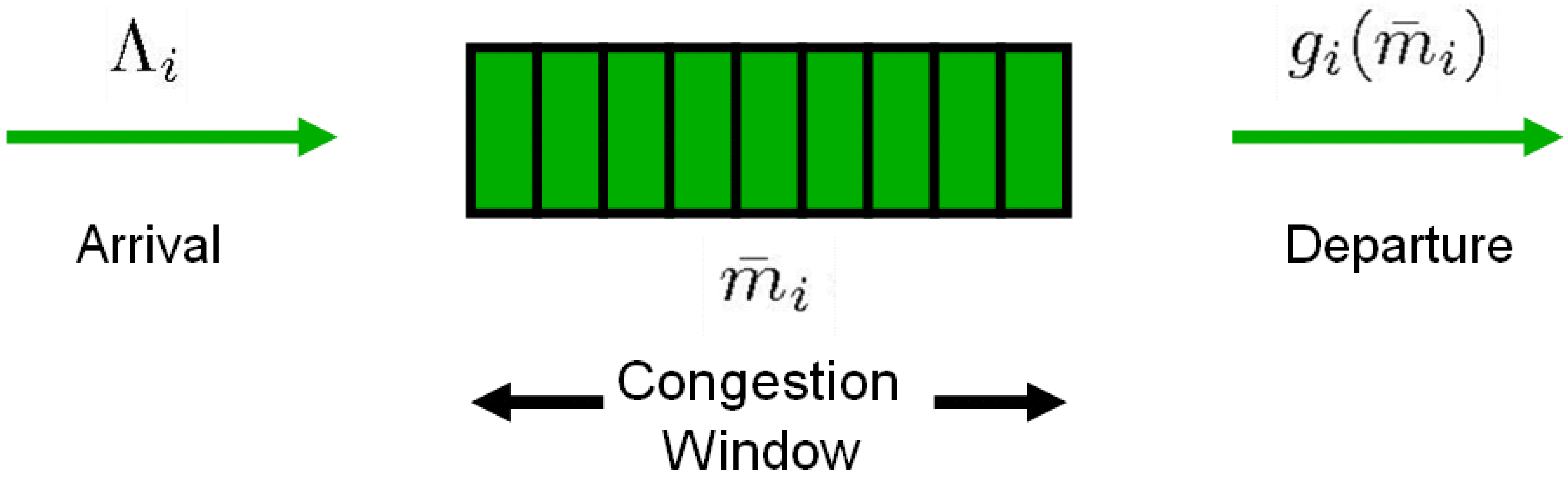
Congestion Windows



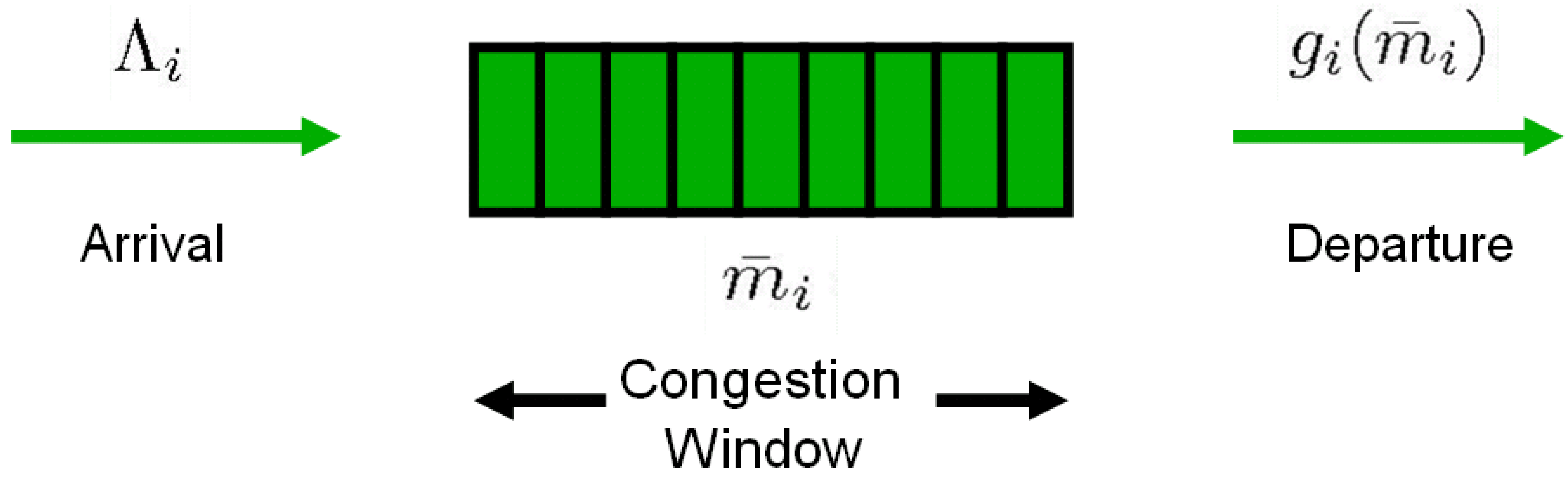
Arrivals acknowledges packets.
Decrease the congestion window size.

Departures send packets into the network.
Increase the congestion window.

Congestion Windows

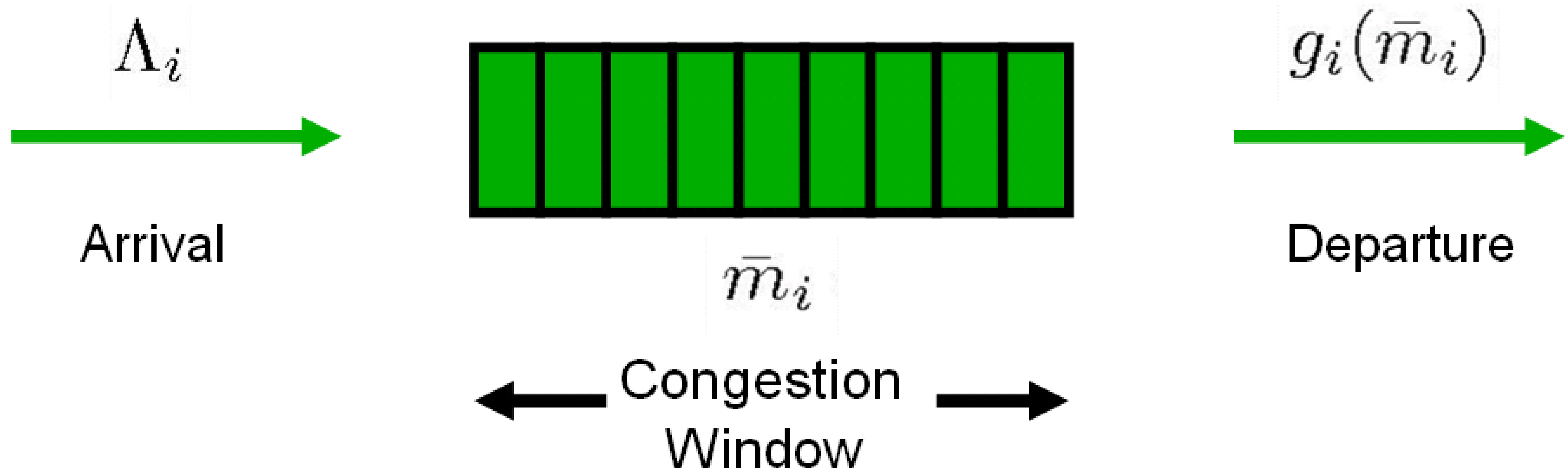


Congestion Windows



Reversible with Stationary Distribution:

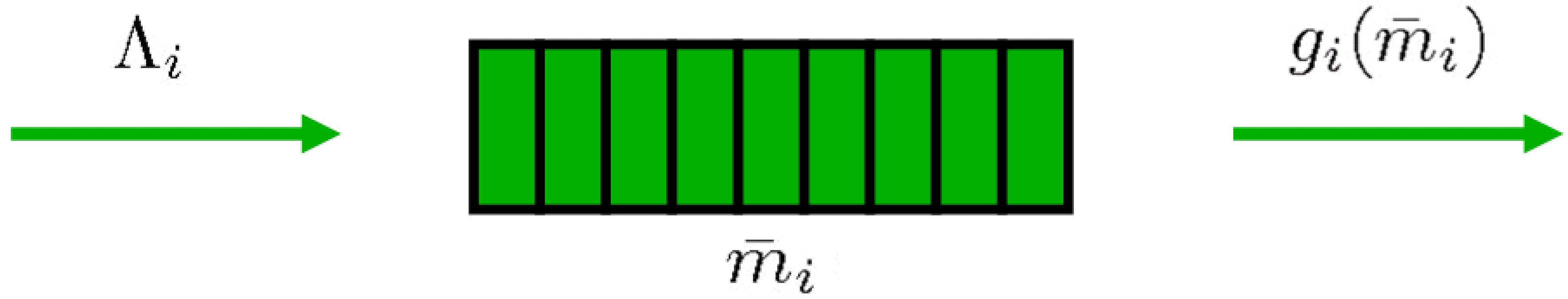
Congestion Windows



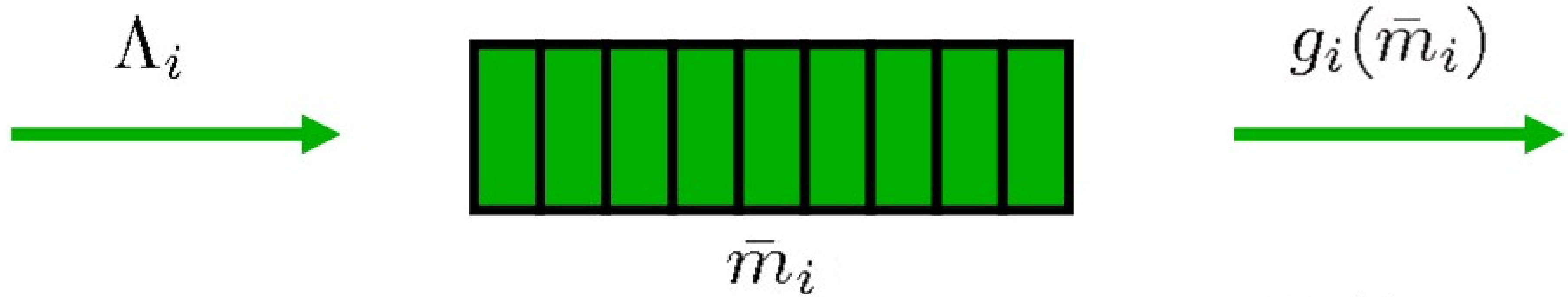
Reversible with Stationary Distribution:

$$\pi_i(\bar{m}_i) = \prod_{k=1}^{\bar{m}_i} \frac{g_i(k)}{\Lambda_i}$$

Congestion Windows

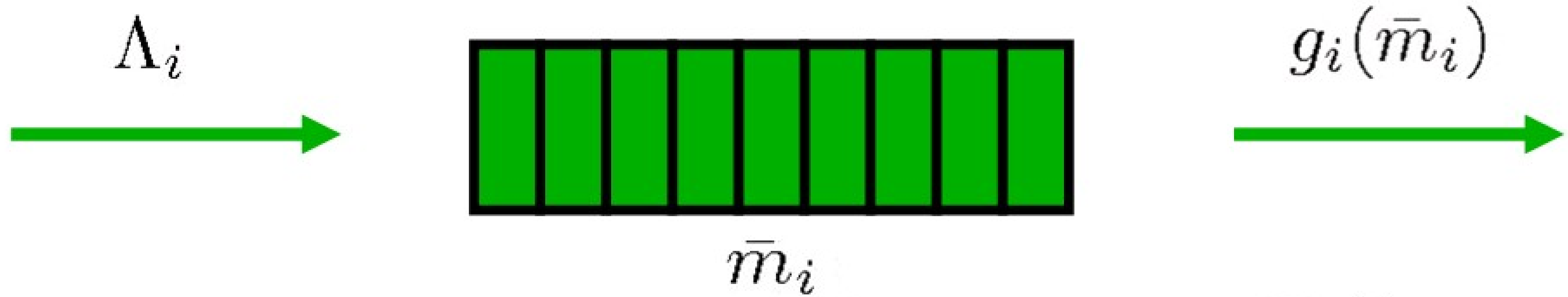


Congestion Windows



Let $\Lambda_i = e^{\lambda_i}$ and $g_i^{(c)}(\bar{m}_i) = e^{cG_i(\frac{\bar{m}_i+1}{c}) - cG_i(\frac{\bar{m}_i}{c})}$

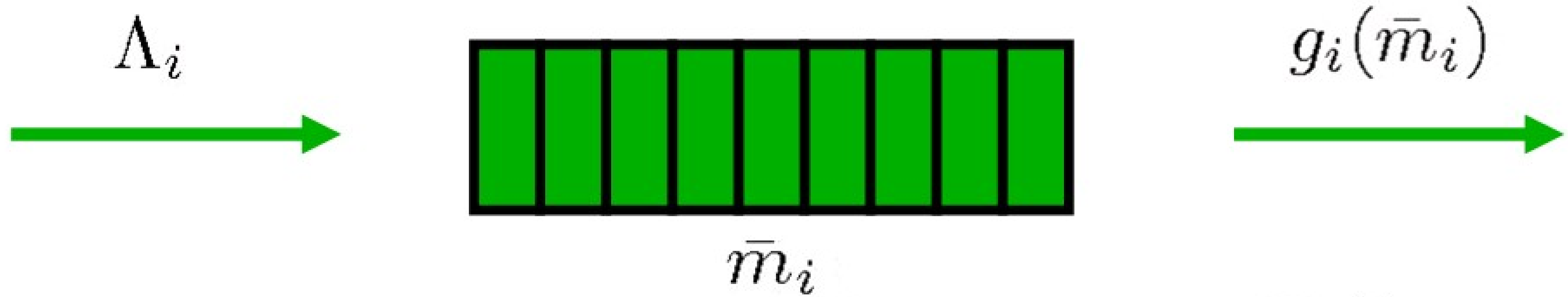
Congestion Windows



Let $\Lambda_i = e^{\lambda_i}$ and $g_i^{(c)}(\bar{m}_i) = e^{cG_i(\frac{\bar{m}_i+1}{c}) - cG_i(\frac{\bar{m}_i}{c})}$

Where we define G_i by a **NEW** USER PROBLEM:

Congestion Windows

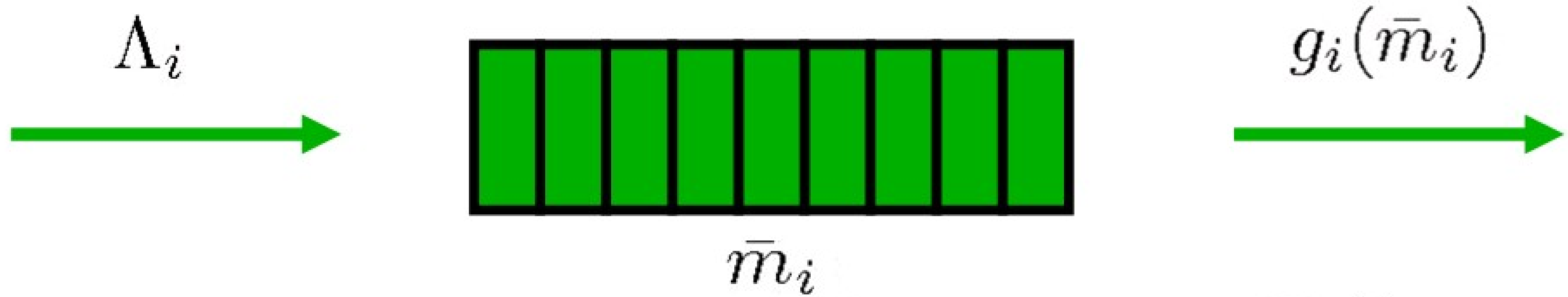


Let $\Lambda_i = e^{\lambda_i}$ and $g_i^{(c)}(\bar{m}_i) = e^{cG_i(\frac{\bar{m}_i+1}{c}) - cG_i(\frac{\bar{m}_i}{c})}$

Where we define G_i by a **NEW** USER PROBLEM:

$$-G_i(\bar{m}_i) = \max_{\lambda_i \in \mathbb{R}} \{U_i(e^{\lambda_i}) - \bar{m}_i \lambda_i\}$$

Congestion Windows



Let $\Lambda_i = e^{\lambda_i}$ and $g_i^{(c)}(\bar{m}_i) = e^{cG_i\left(\frac{\bar{m}_i+1}{c}\right) - cG_i\left(\frac{\bar{m}_i}{c}\right)}$

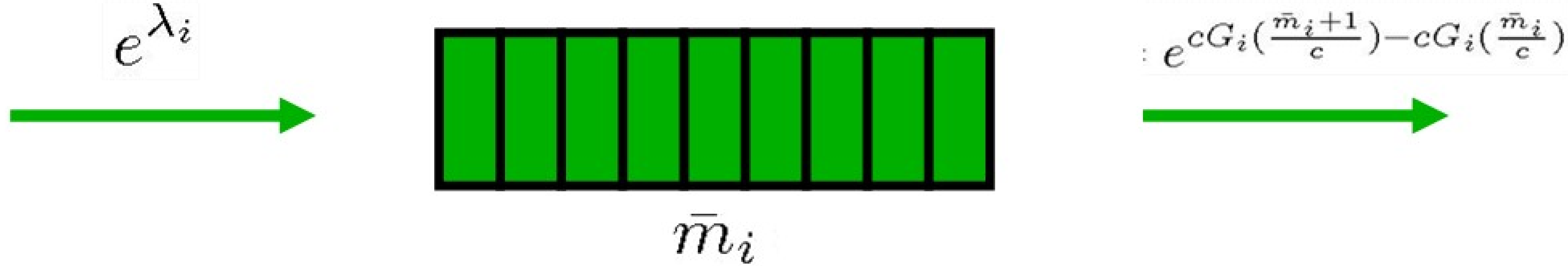
Where we define G_i by a **NEW** USER PROBLEM:

$$-G_i(\bar{m}_i) = \max_{\lambda_i \in \mathbb{R}} \{U_i(e^{\lambda_i}) - \bar{m}_i \lambda_i\}$$

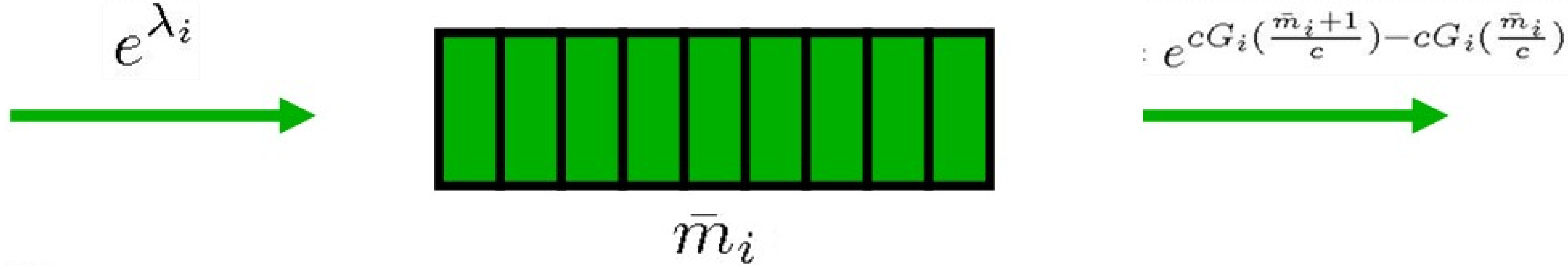
With stationary distribution

$$\pi_i^{(c)}(\bar{m}_i) = e^{cG_i\left(\frac{\bar{m}_i}{c}\right) - \lambda_i \bar{m}_i}$$

Congestion Windows



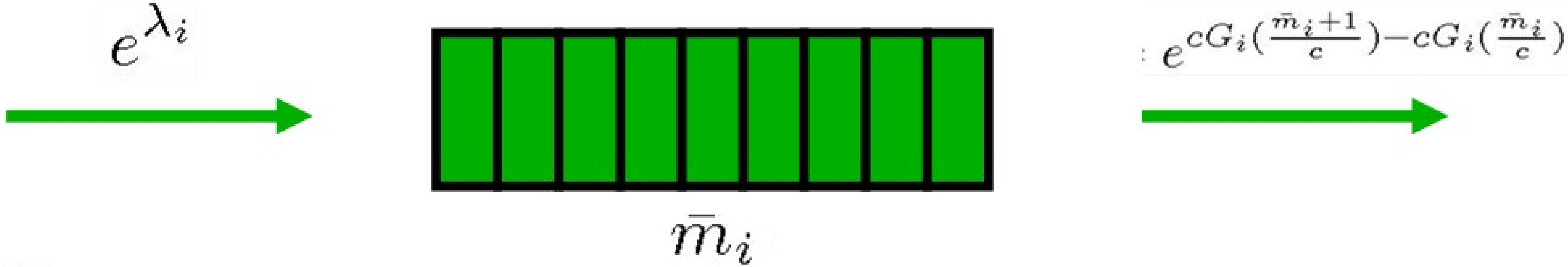
Congestion Windows



Thus

$$\lim_{c\to\infty}\frac{1}{c}\log\pi^{(c)}(c\bar{m}_i)=G_i(\bar{m}_i)-\lambda_i\bar{m}_i$$

Congestion Windows

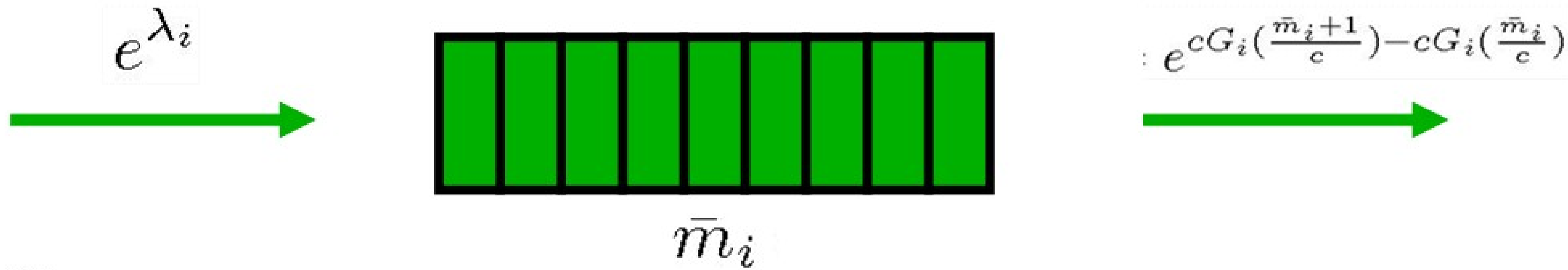


Thus

$$\lim_{c\to\infty}\frac{1}{c}\log\pi^{(c)}(c\bar{m}_i) = G_i(\bar{m}_i) - \lambda_i\bar{m}_i$$

The most likely state is

Congestion Windows



Thus

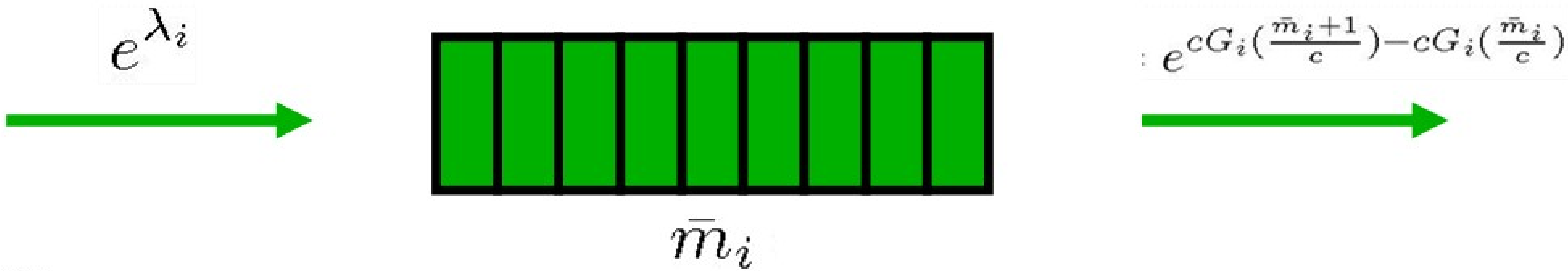
$$\lim_{c\to\infty}\frac{1}{c}\log\pi^{(c)}(c\bar{m}_i)=G_i(\bar{m}_i)-\lambda_i\bar{m}_i$$

The most likely state is

$$\lim_{c\to\infty}\frac{1}{c}\log\pi^{(c)}(c\bar{m}_i^*)=\max_{\bar{m}_i>0}\{G_i(\bar{m}_i)-\lambda_i\bar{m}_i\}$$

$$= -U_i(e^{\lambda_i})$$

Congestion Windows



Thus

$$\lim_{c\to\infty} \frac{1}{c} \log \pi^{(c)}(c\bar{m}_i) = G_i(\bar{m}_i) - \lambda_i \bar{m}_i$$

The most likely state is

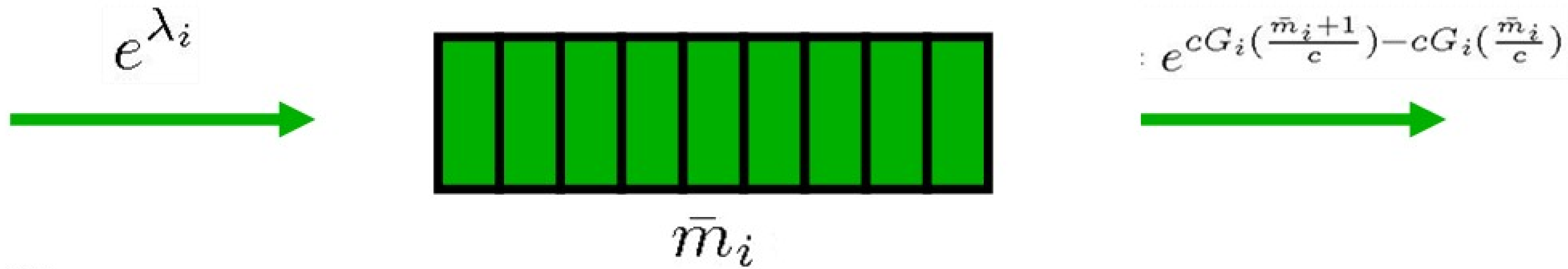
$$\lim_{c\to\infty} \frac{1}{c} \log \pi^{(c)}(c\bar{m}_i^*) = \max_{\bar{m}_i>0} \{G_i(\bar{m}_i) - \lambda_i \bar{m}_i\}$$

$$= \quad - U_i(e^{\lambda_i})$$

Provided

Assumption: $U_i(e^{\lambda_i})$ is concave.

Congestion Windows



Thus

$$\lim_{c\to\infty}\frac{1}{c}\log\pi^{(c)}(c\bar{m}_i)=G_i(\bar{m}_i)-\lambda_i\bar{m}_i$$

The most likely state is

$$\lim_{c\to\infty}\frac{1}{c}\log\pi^{(c)}(c\bar{m}_i^*)=\max_{\bar{m}_i>0}\{G_i(\bar{m}_i)-\lambda_i\bar{m}_i\}$$

$$=\quad -U_i(e^{\lambda_i})$$

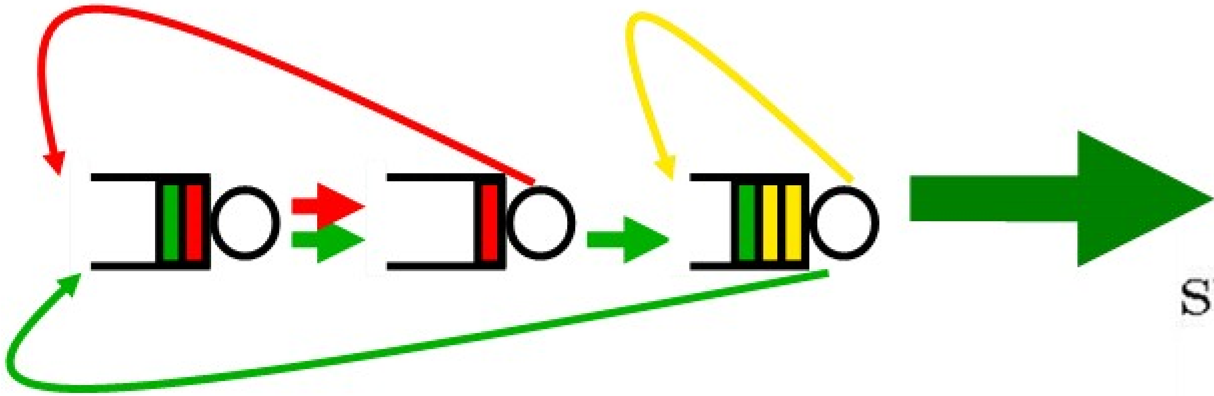
Provided

Assumption: $U_i(e^{\lambda_i})$ is concave.

For example: weighted alpha fair for $\alpha > 1$

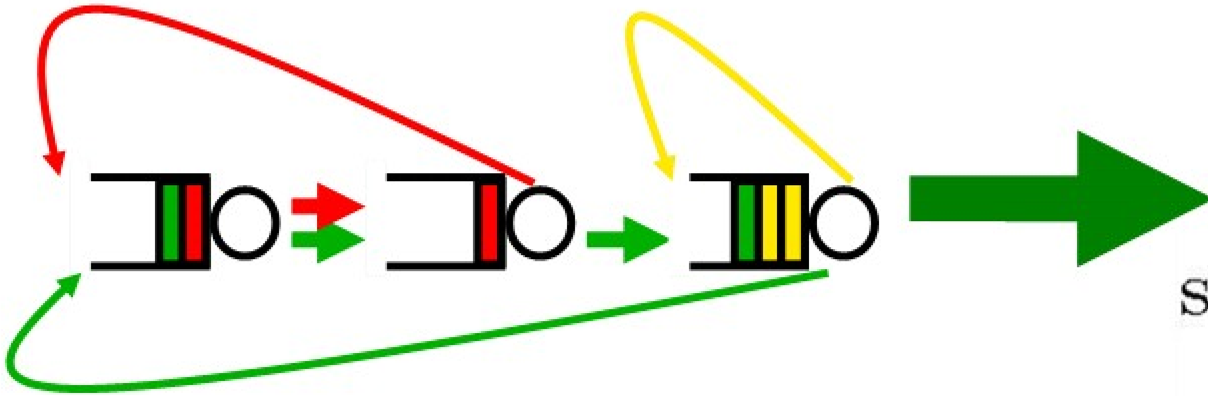
NETWORK PROBLEM is:

NETWORK PROBLEM is:



$$\max \sum_{i\in\mathcal{I}} \bar{m}_i \log \Lambda_i$$
$$\text{subject to} \quad \sum_{i:j\in i} \Lambda_i \le C_j,$$

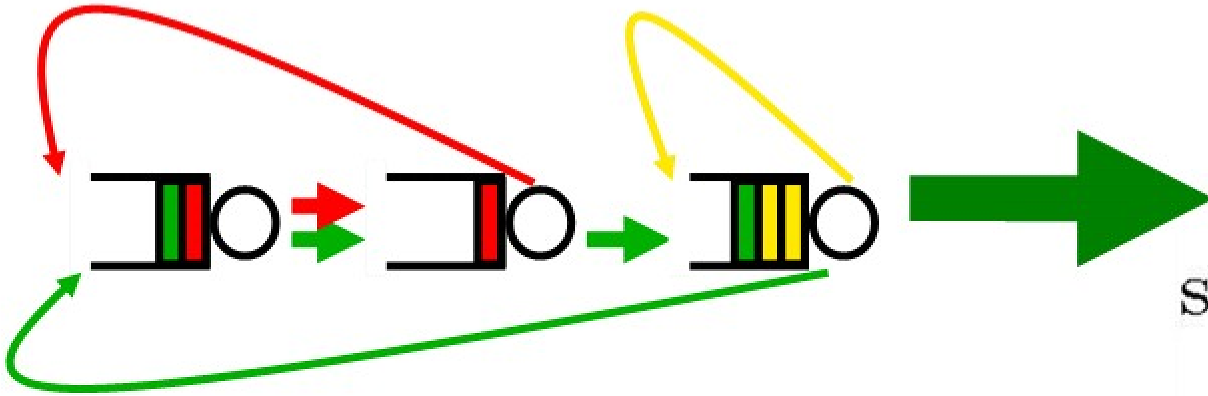
NETWORK PROBLEM is:



$$\max \sum_{i \in \mathcal{I}} \bar{m}_i \log \Lambda_i$$
$$\text{subject to} \sum_{i : j \in i} \Lambda_i \leq C_j,$$

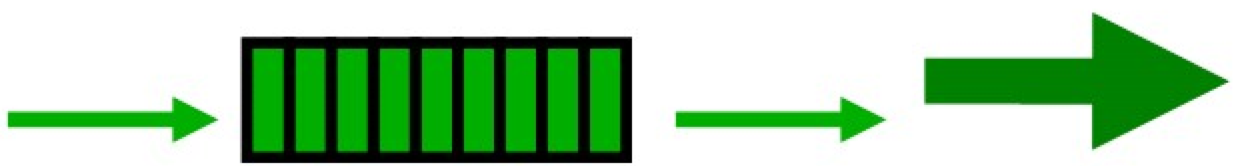
USER PROBLEM:

NETWORK PROBLEM is:



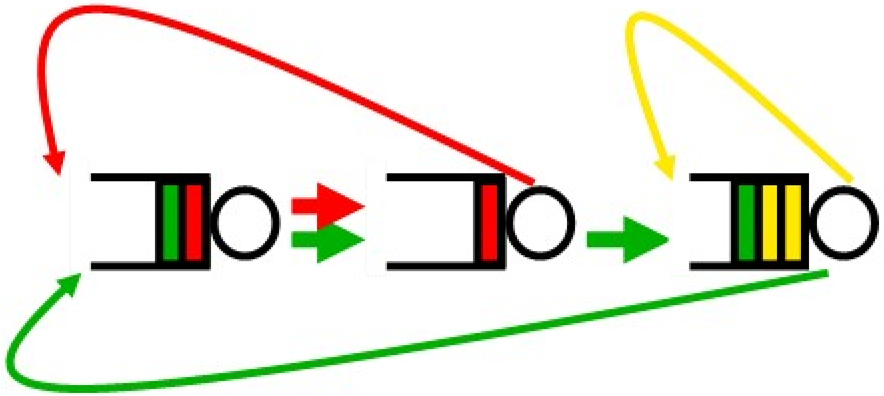
$$\max \sum_{i \in \mathcal{I}} \bar{m}_i \log \Lambda_i$$
$$\text{subject to} \quad \sum_{i : j \in i} \Lambda_i \leq C_j,$$

USER PROBLEM:



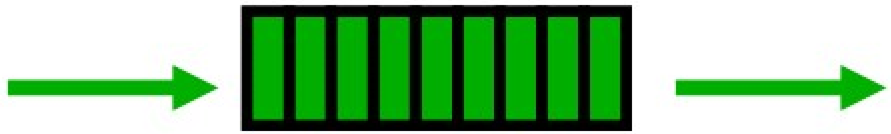
$$\max_{\lambda_i \in \mathbb{R}} \{ U_i(e^{\lambda_i}) - \bar{m}_i \lambda_i \}$$

NETWORK PROBLEM is:



$$\max \sum_{i \in \mathcal{I}} \bar{m}_i \log \Lambda_i$$
$$\text{subject to} \sum_{i : j \in i} \Lambda_i \leq C_j,$$

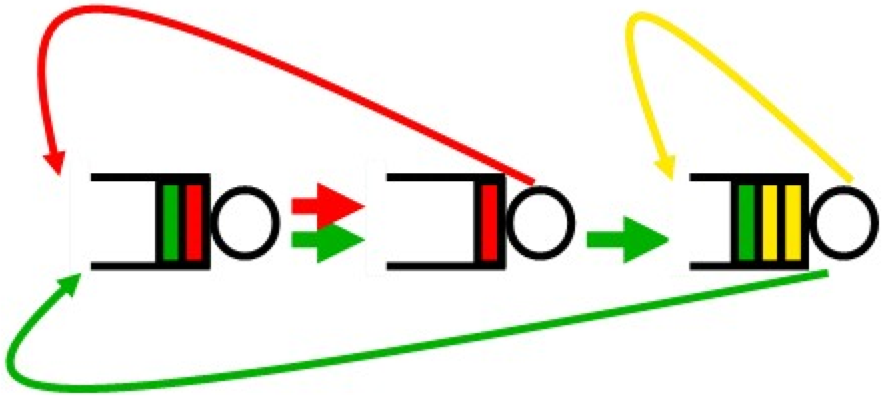
USER PROBLEM:



$$\max_{\lambda_i \in \mathbb{R}} \{U_i(e^{\lambda_i}) - \bar{m}_i \lambda_i\}$$

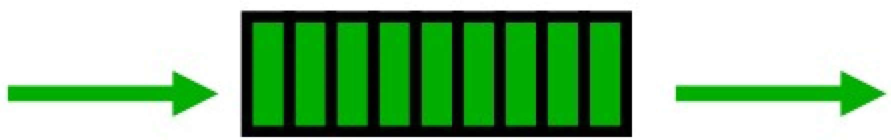
AND PRICES:

NETWORK PROBLEM is:



$$\max \sum_{i\in\mathcal{I}} \bar{m}_i \log \Lambda_i$$
$$\text{subject to} \sum_{i:j\in i} \Lambda_i \le C_j,$$

USER PROBLEM:

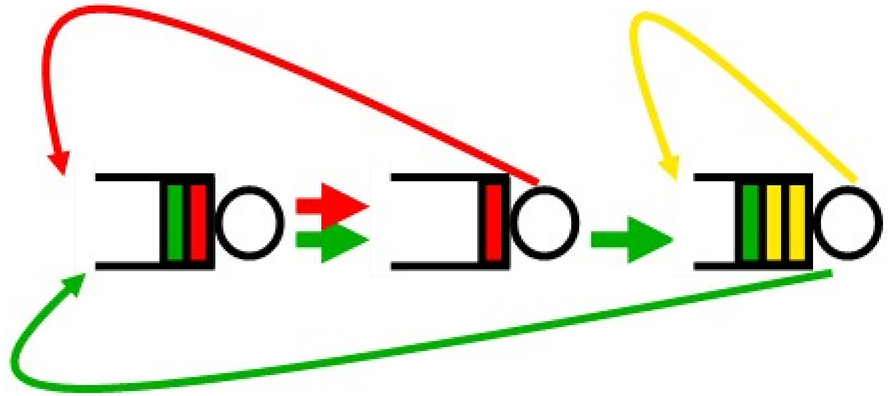


$$\max_{\lambda_i\in\mathbb{R}} \{U_i(e^{\lambda_i}) - \bar{m}_i\lambda_i\}$$

AND PRICES:

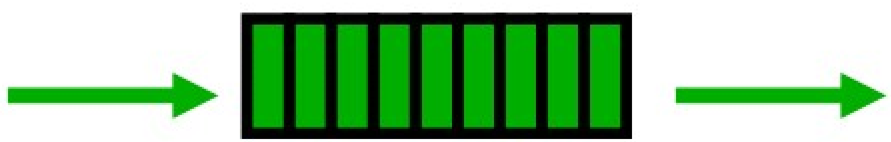
$$\bar{m}_i = \Lambda_i q_i$$

NETWORK PROBLEM is:



$$\max \sum_{i \in \mathcal{I}} \bar{m}_i \log \Lambda_i$$
$$\text{subject to} \quad \sum_{i : j \in i} \Lambda_i \le C_j,$$

USER PROBLEM:



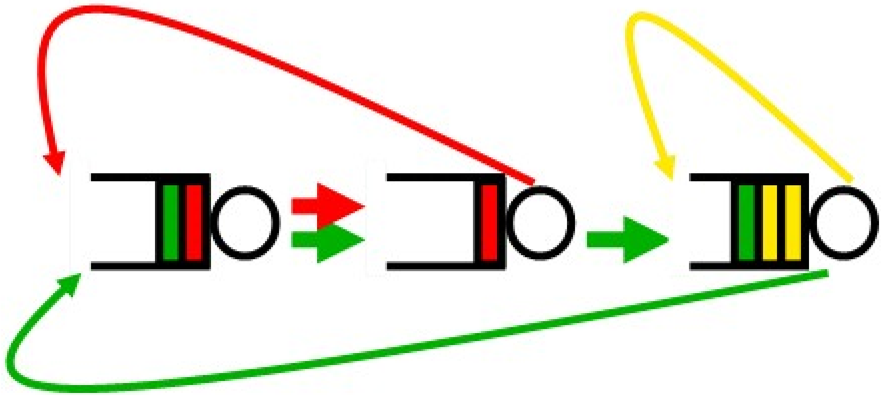
$$\max_{\lambda_i \in \mathbb{R}} \{ U_i(e^{\lambda_i}) - \bar{m}_i \lambda_i \}$$

AND PRICES:

Little's Law

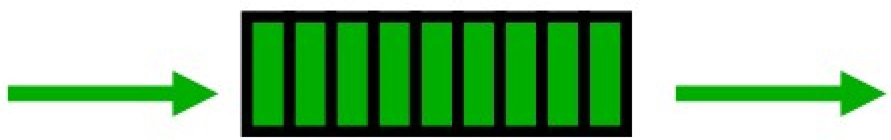
$$\bar{m}_i = \Lambda_i q_i$$

NETWORK PROBLEM is:



$$\max \sum_{i \in \mathcal{I}} \bar{m}_i \log \Lambda_i$$
$$\text{subject to} \quad \sum_{i : j \in i} \Lambda_i \leq C_j,$$

USER PROBLEM:



$$\max_{\lambda_i \in \mathbb{R}} \{ U_i(e^{\lambda_i}) - \bar{m}_i \lambda_i \}$$

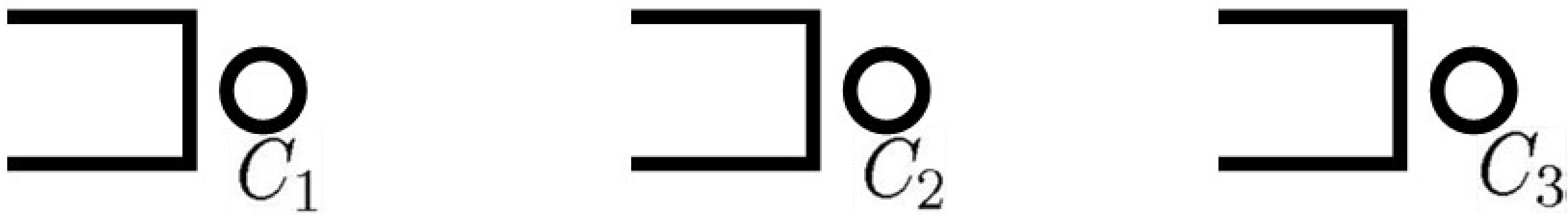
AND PRICES:

Little's Law

$$\bar{m}_i = \Lambda_i q_i$$

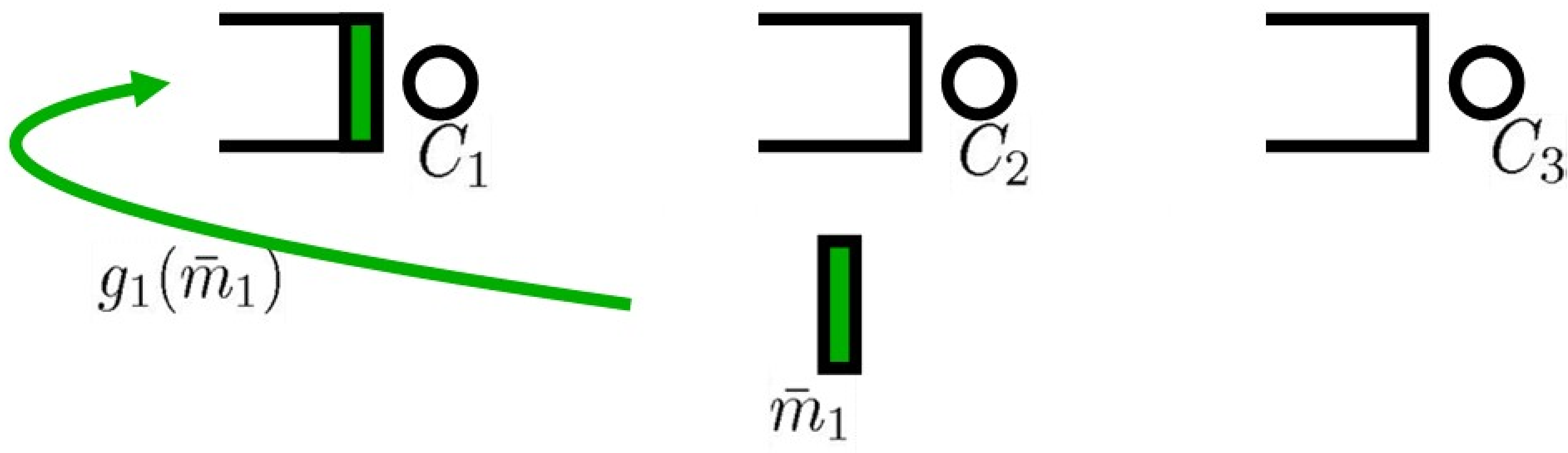
SYSTEM PROBLEM...

A System of Queues

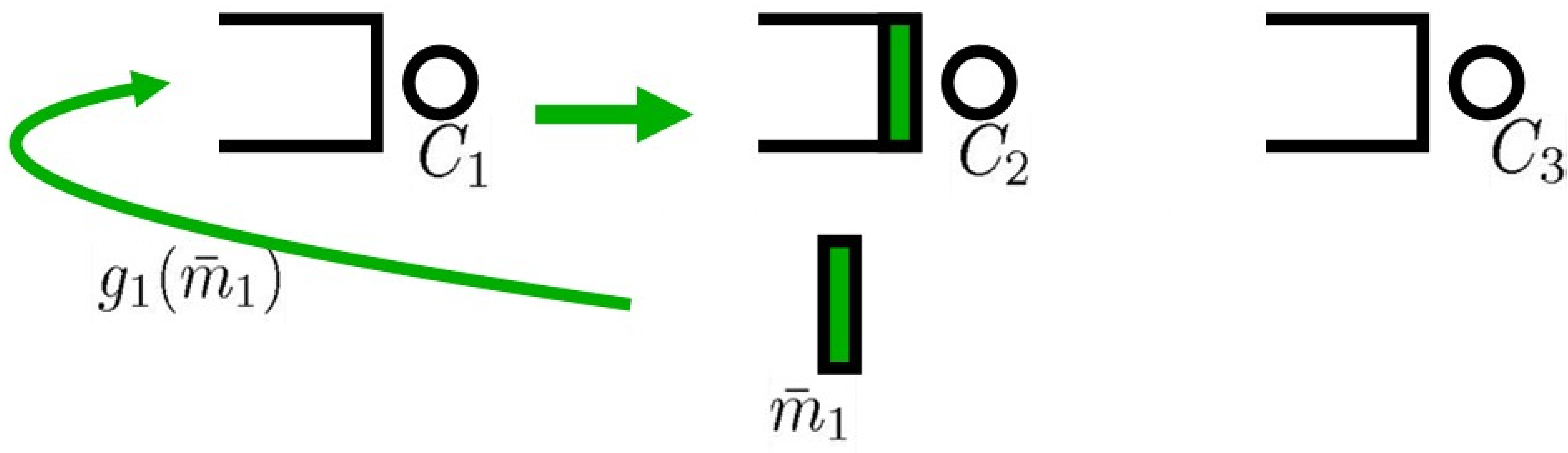


$\bar{m}_1$

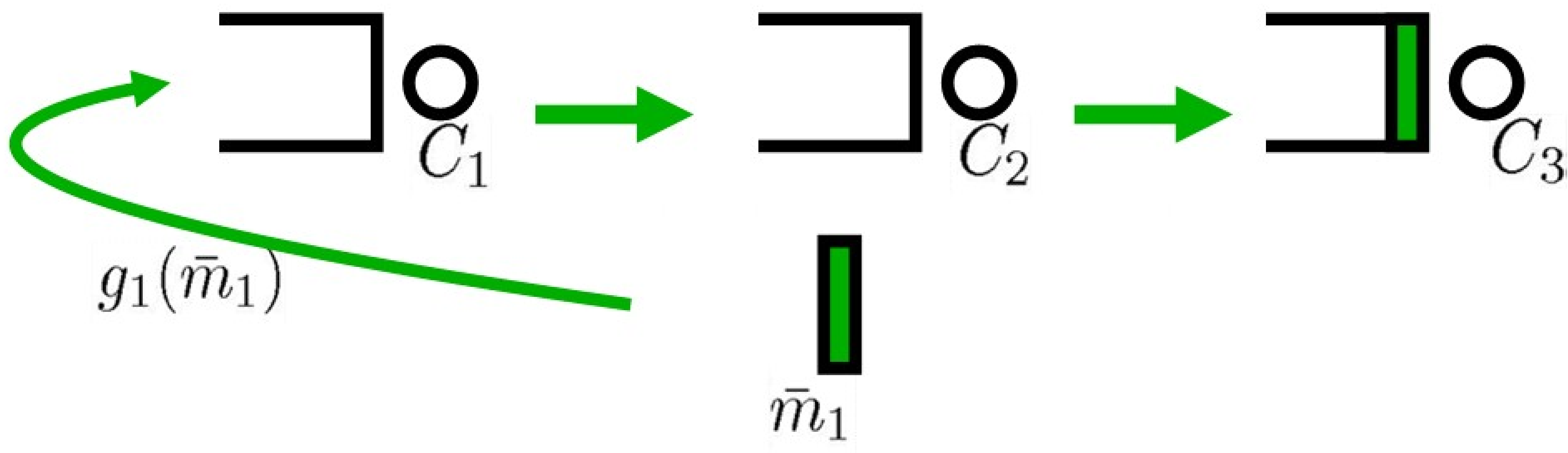
A System of Queues



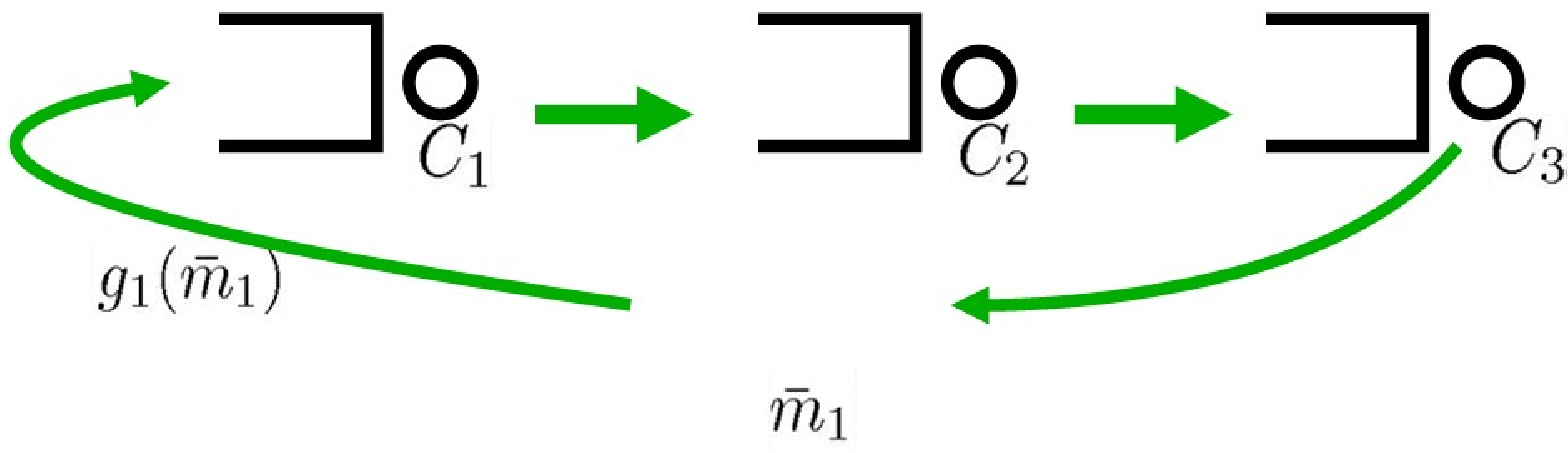
A System of Queues



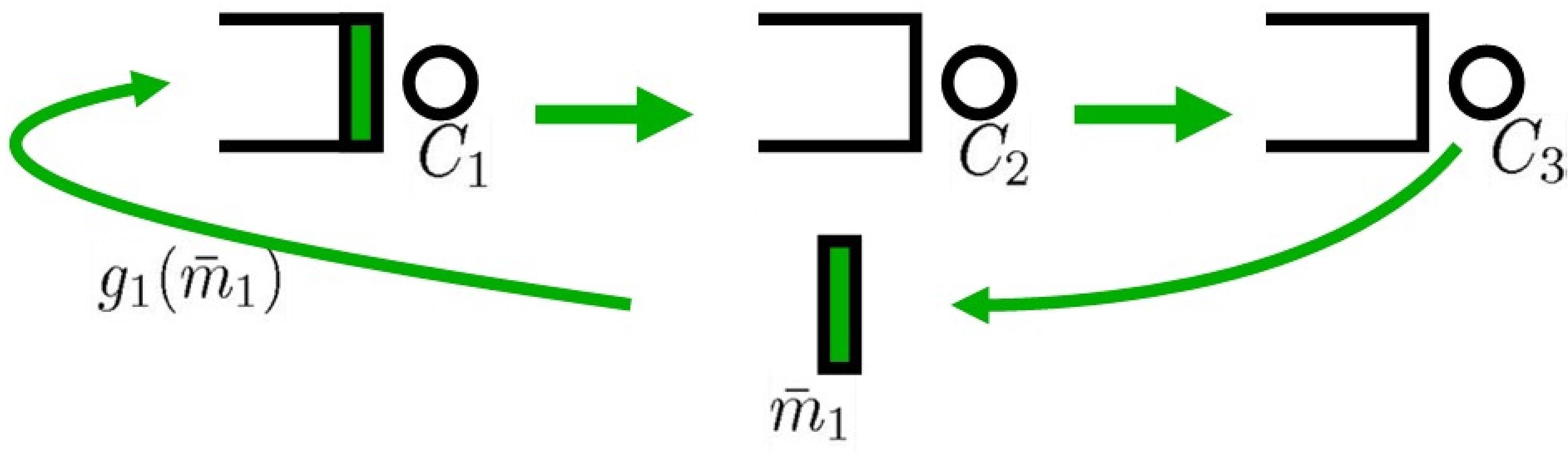
A System of Queues



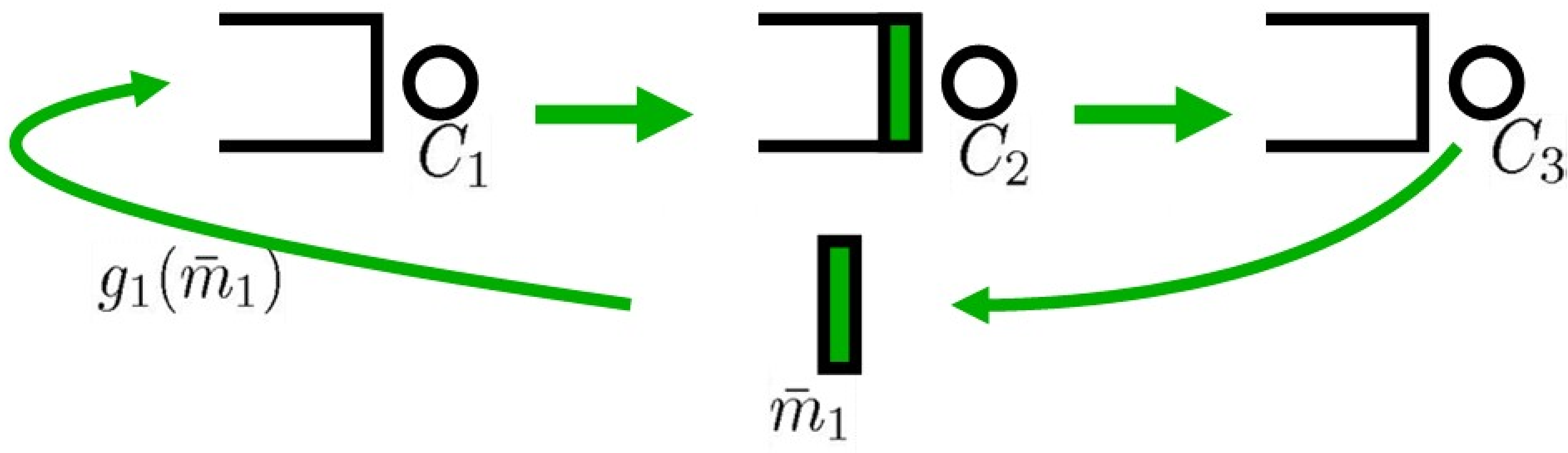
A System of Queues



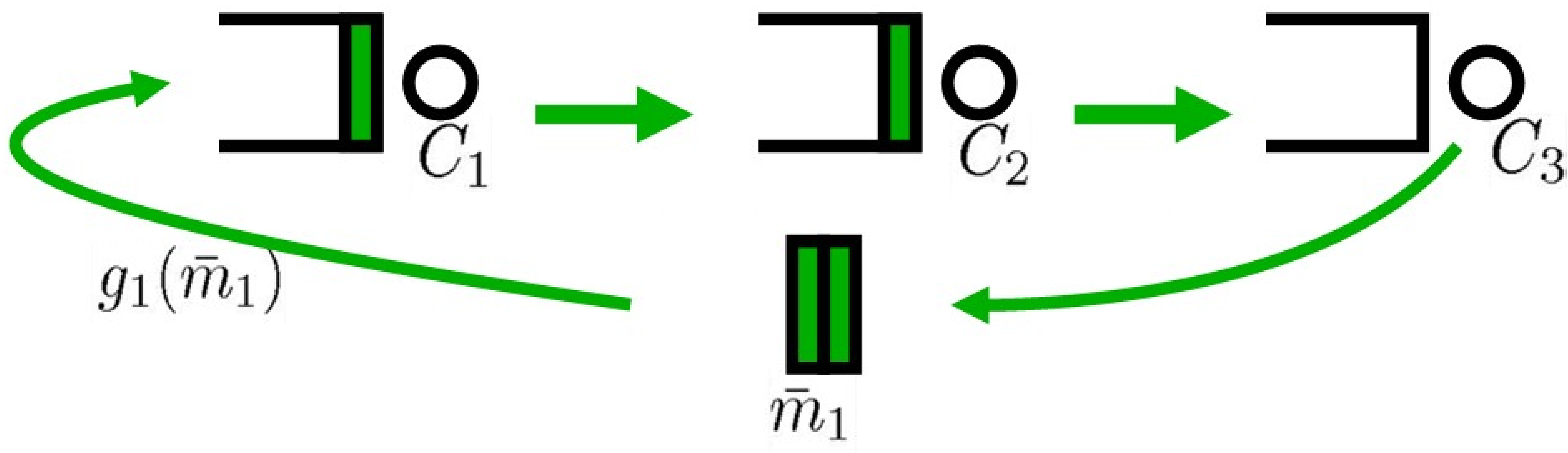
A System of Queues



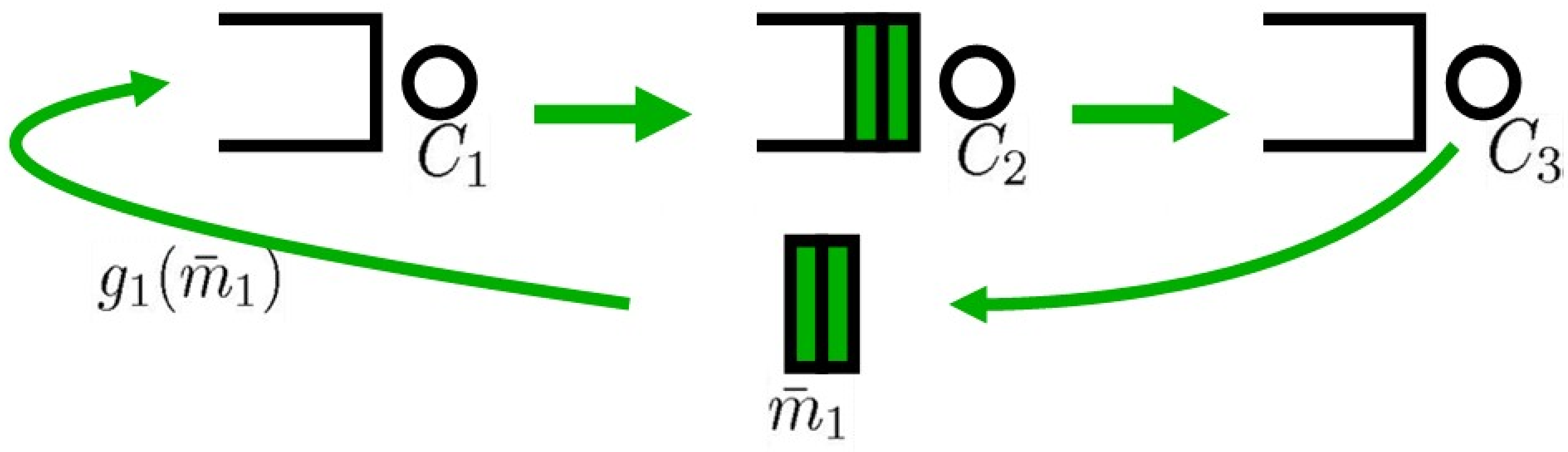
A System of Queues



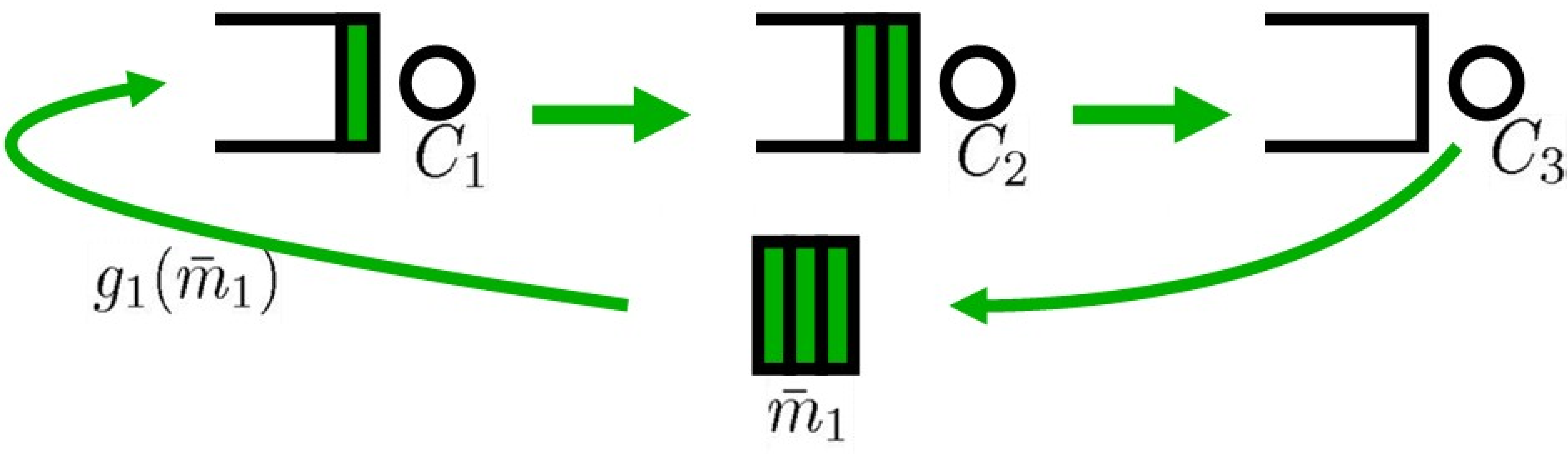
A System of Queues



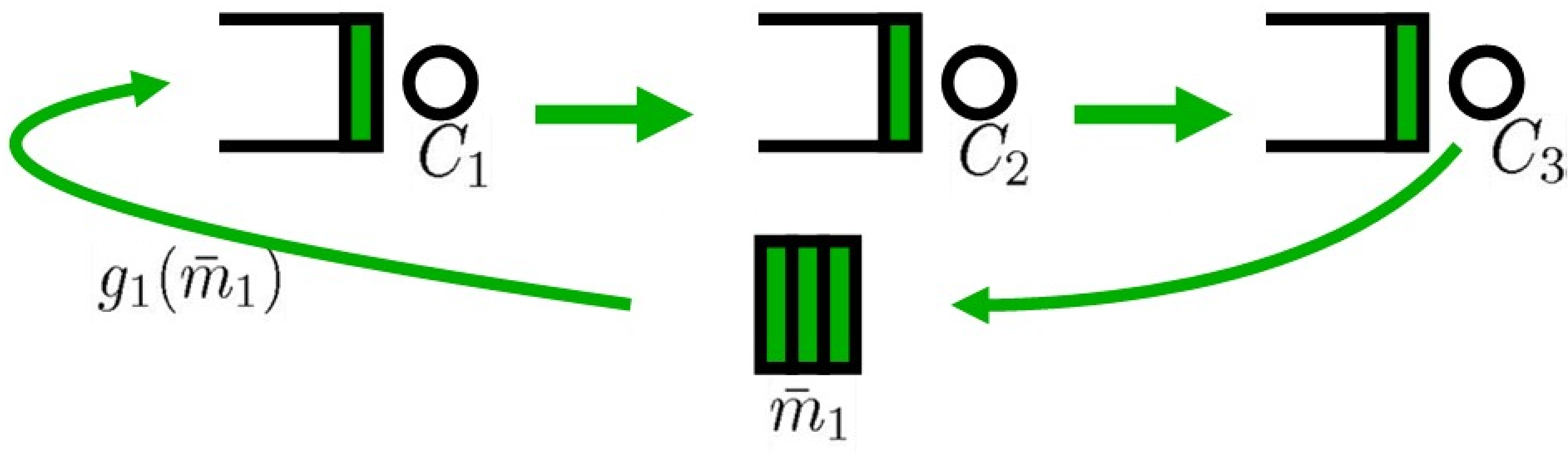
A System of Queues



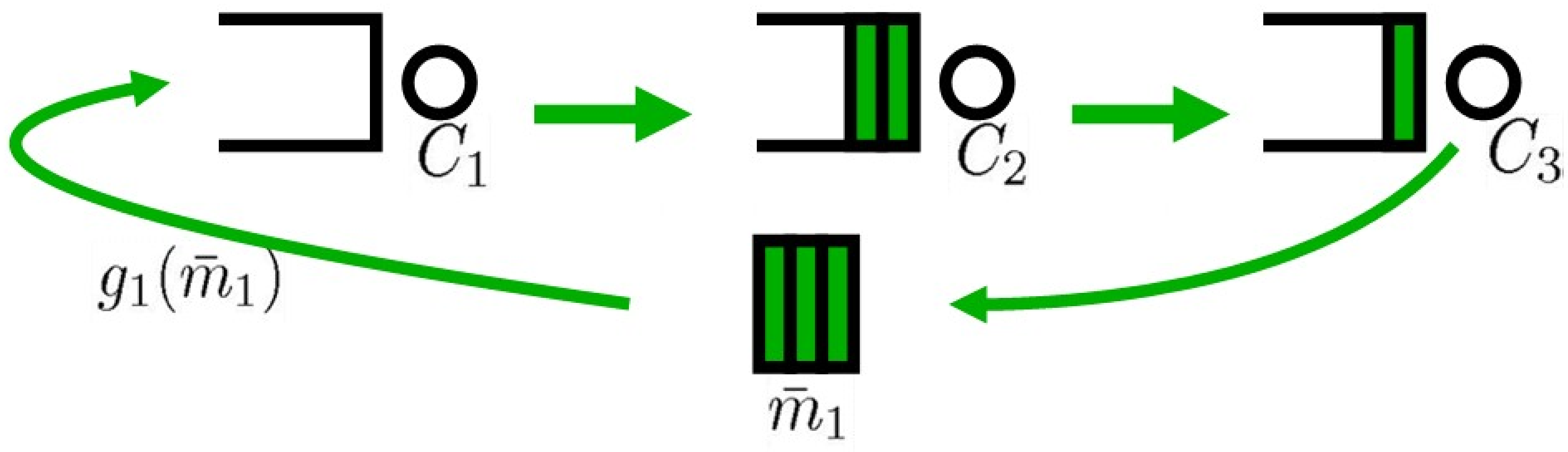
A System of Queues



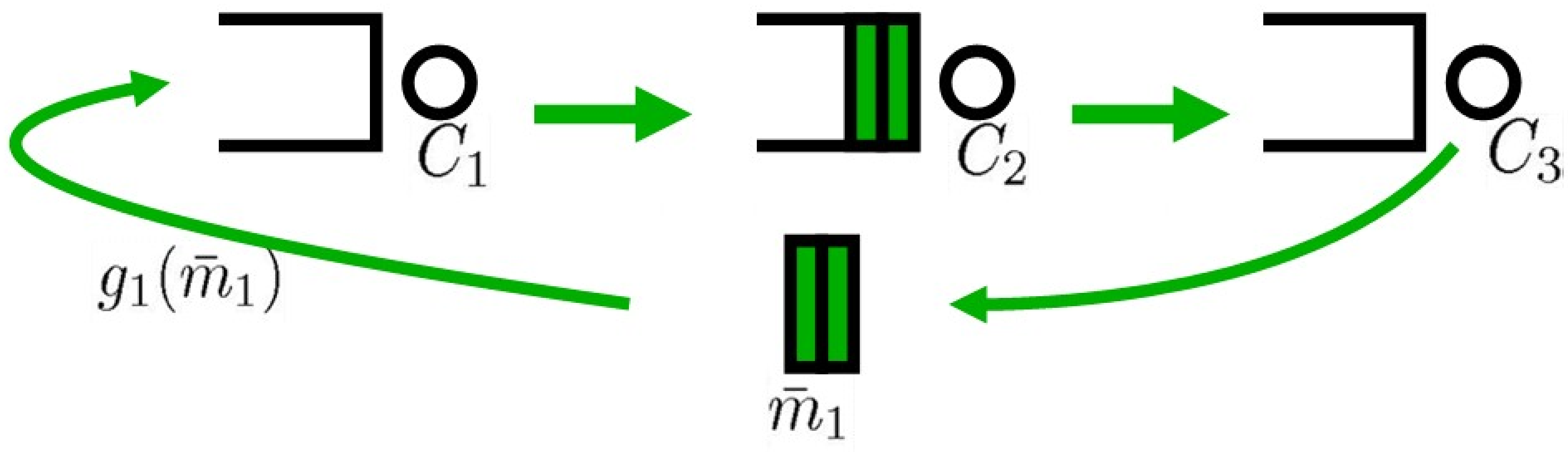
A System of Queues



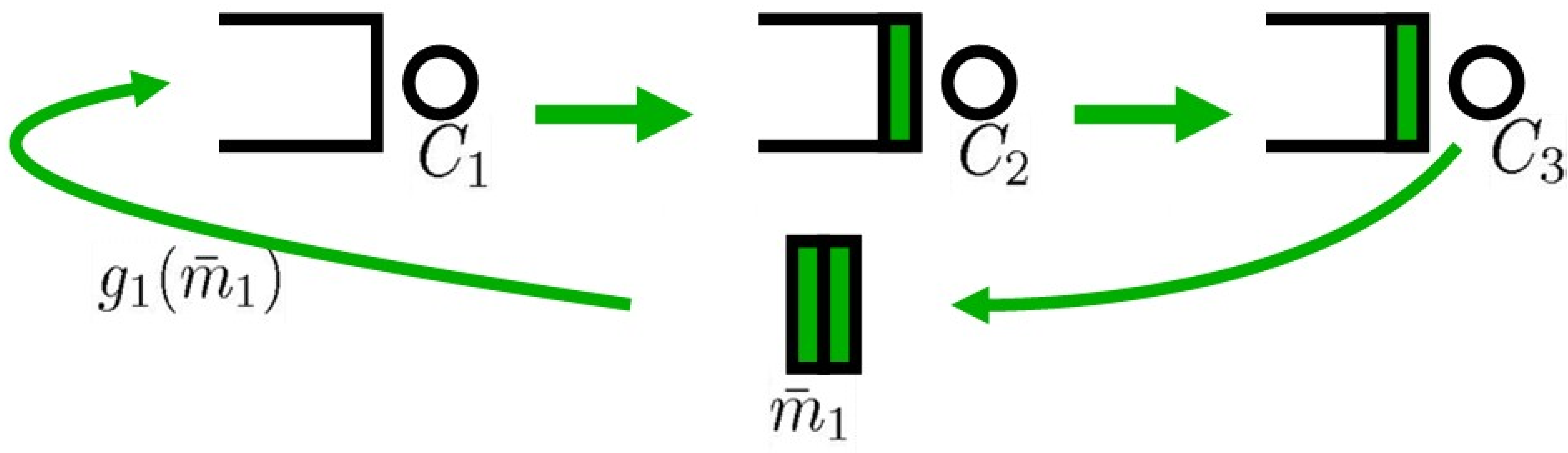
A System of Queues



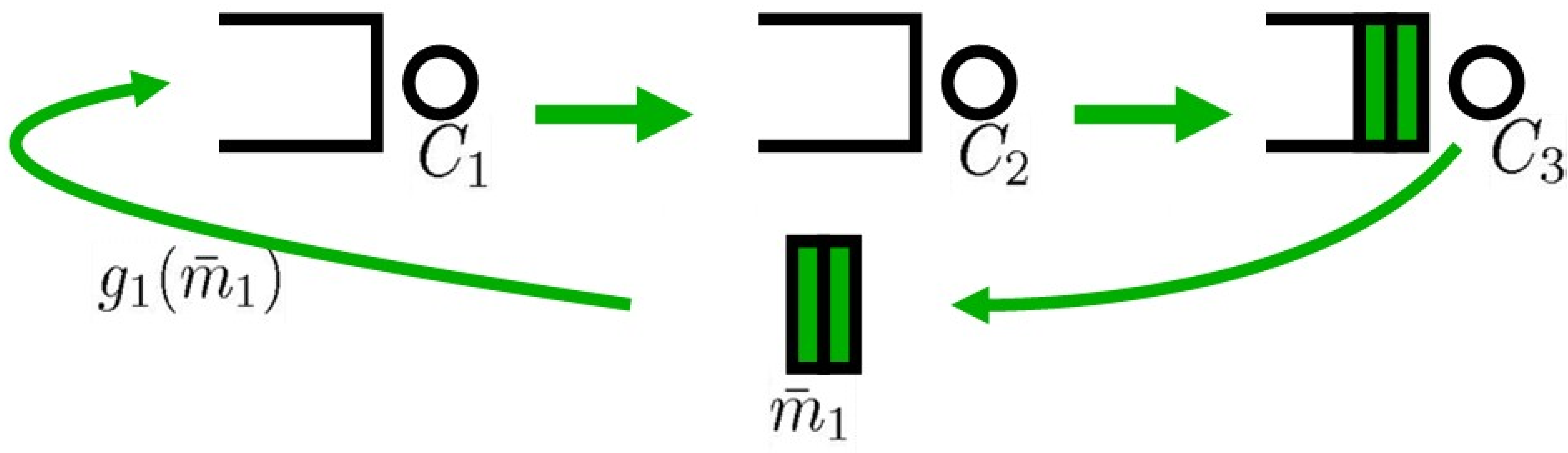
A System of Queues



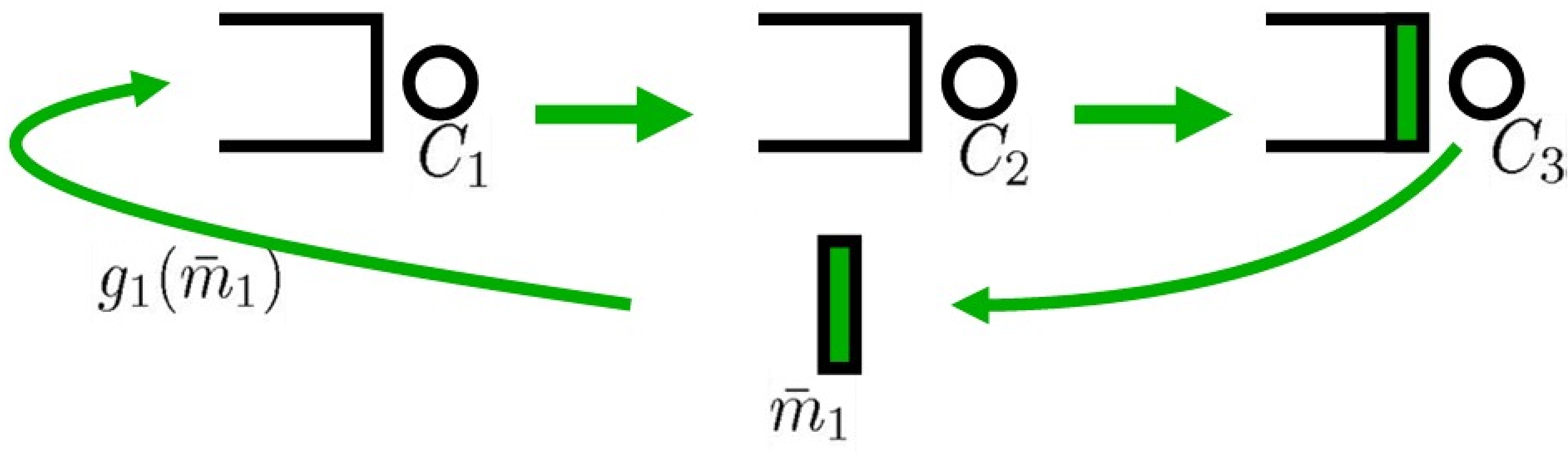
A System of Queues



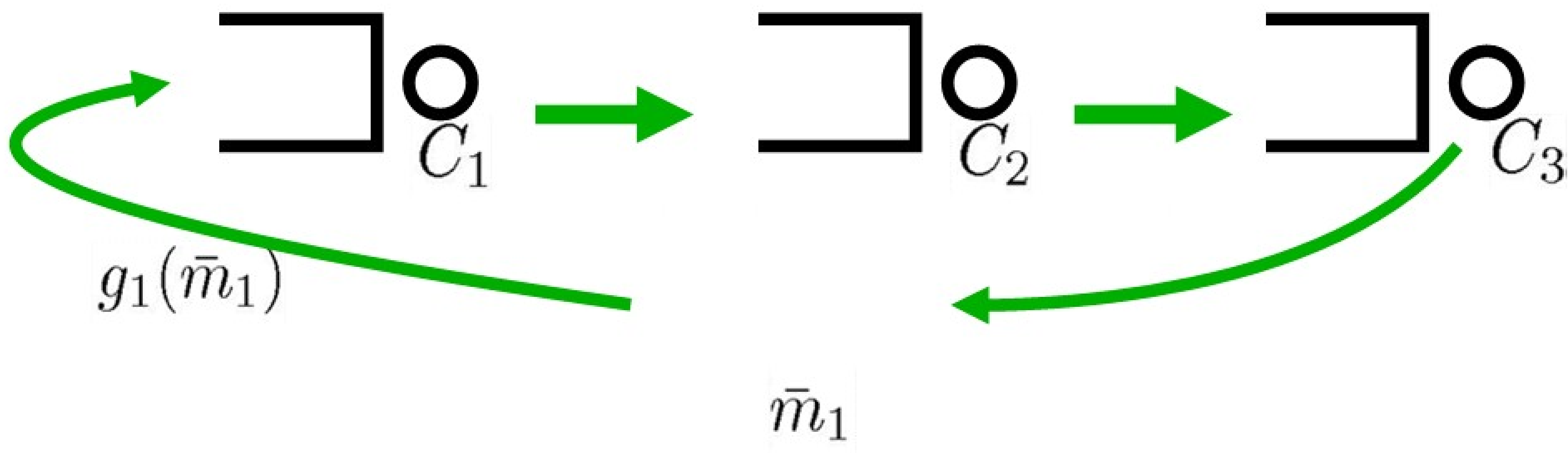
A System of Queues



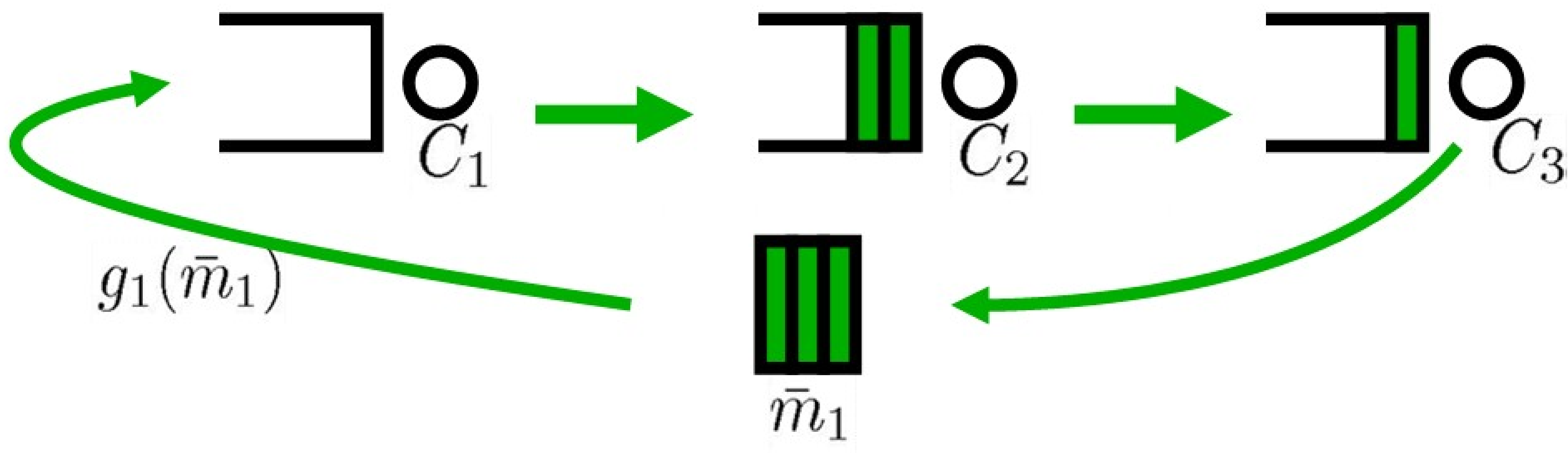
A System of Queues



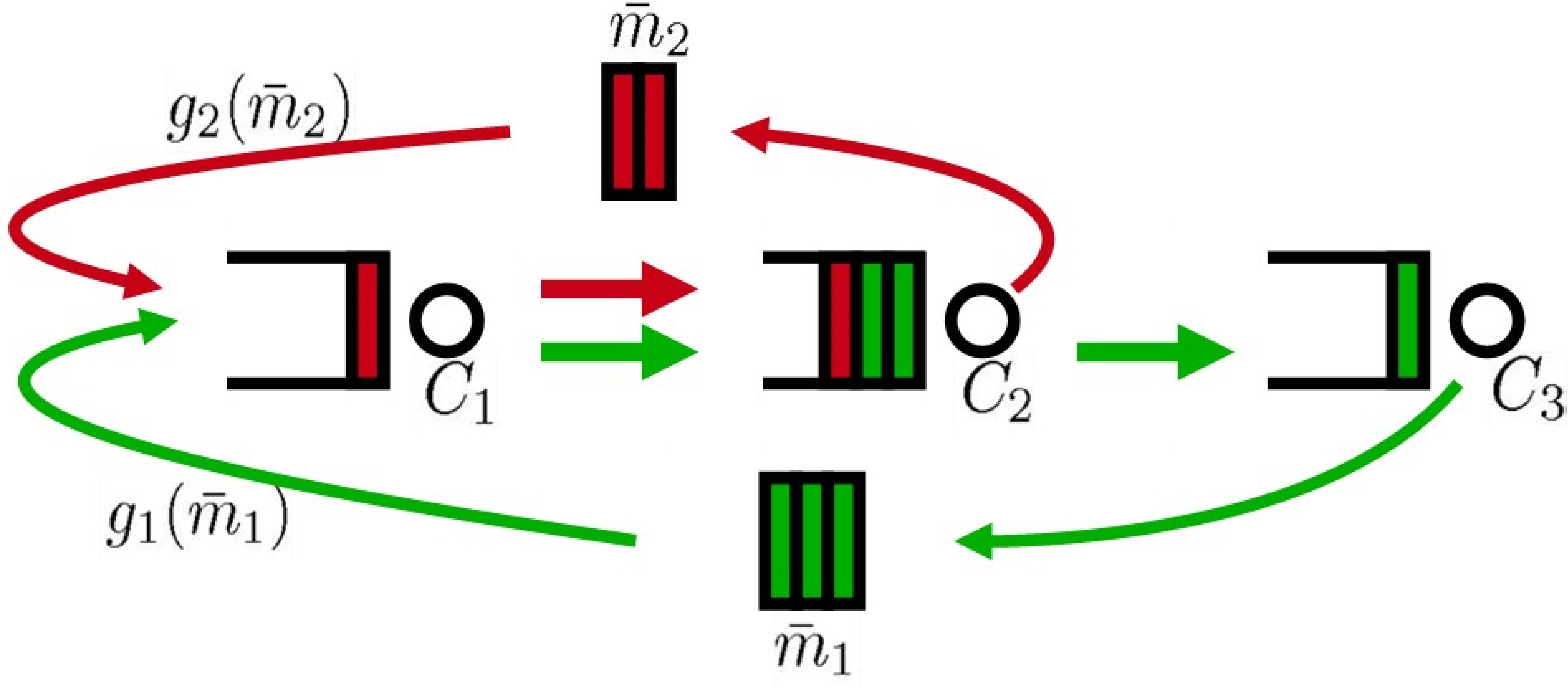
A System of Queues



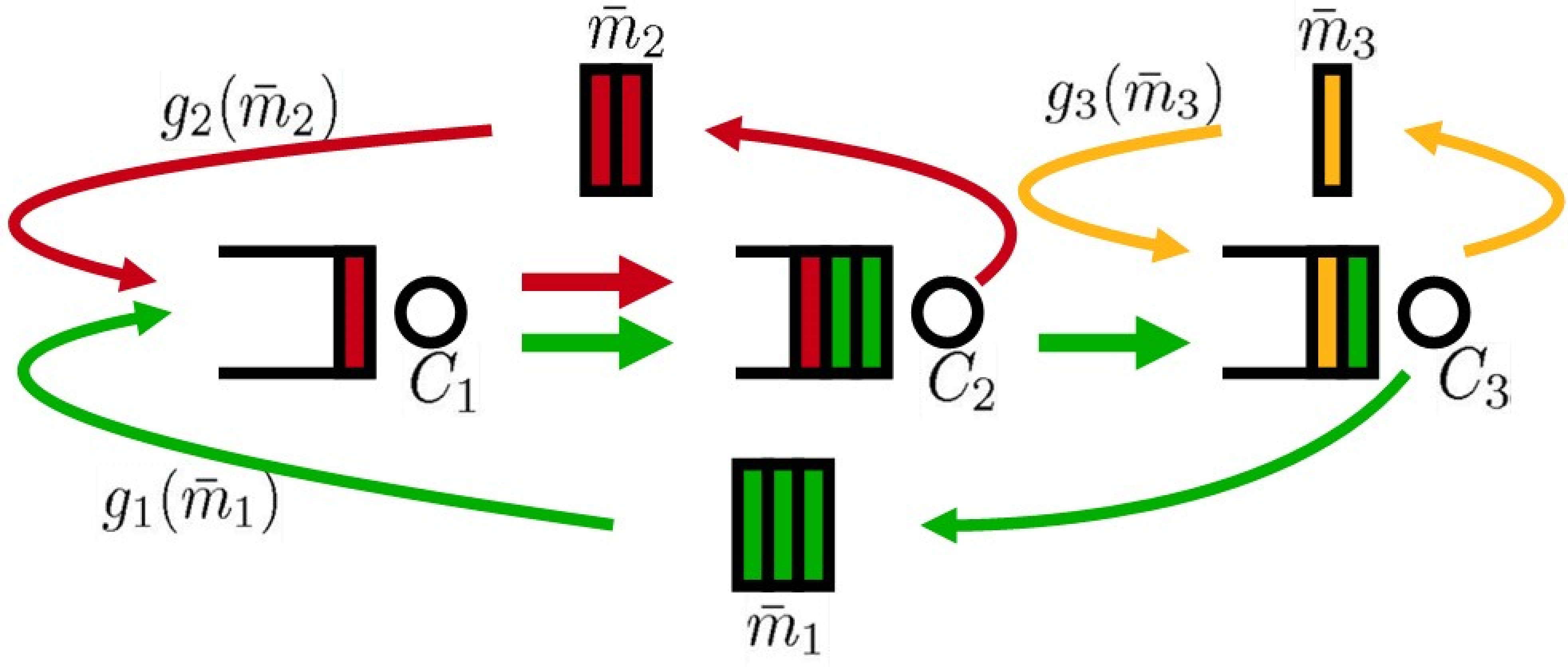
A System of Queues



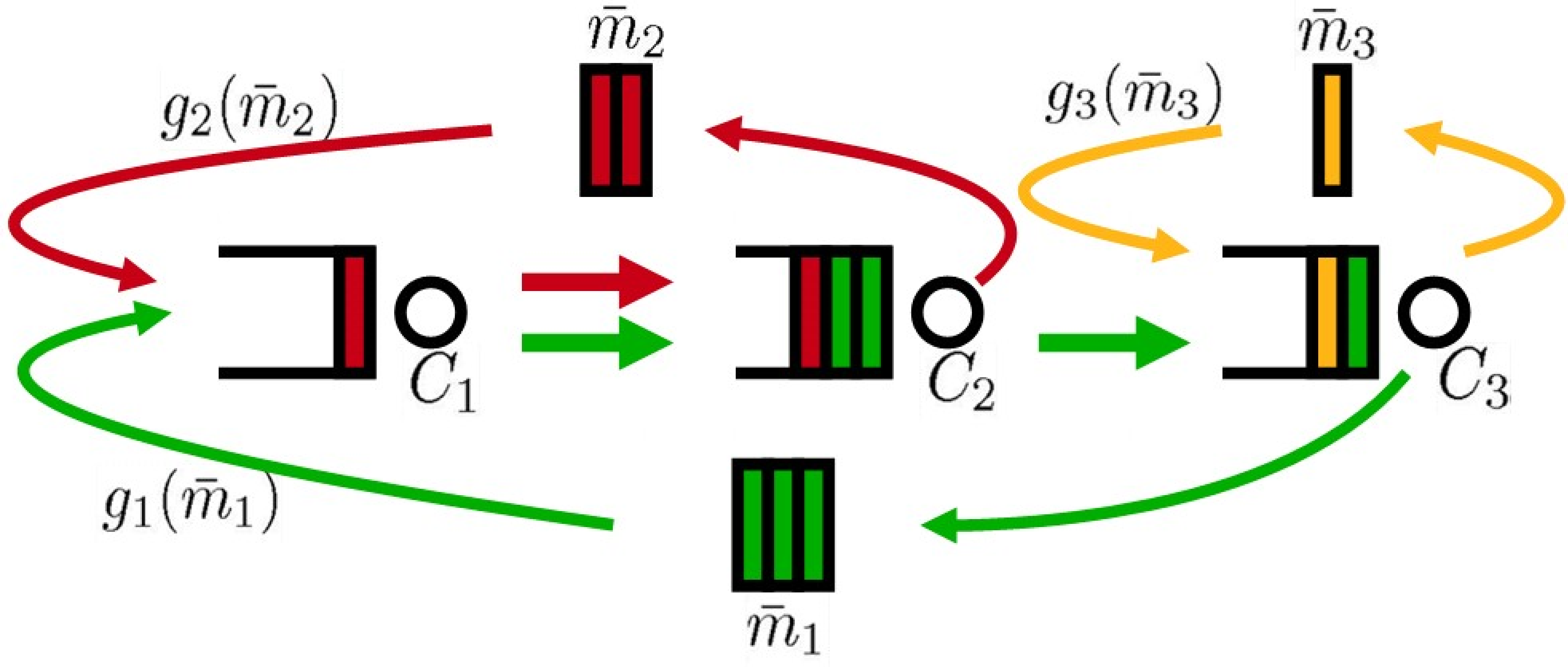
A System of Queues



A System of Queues

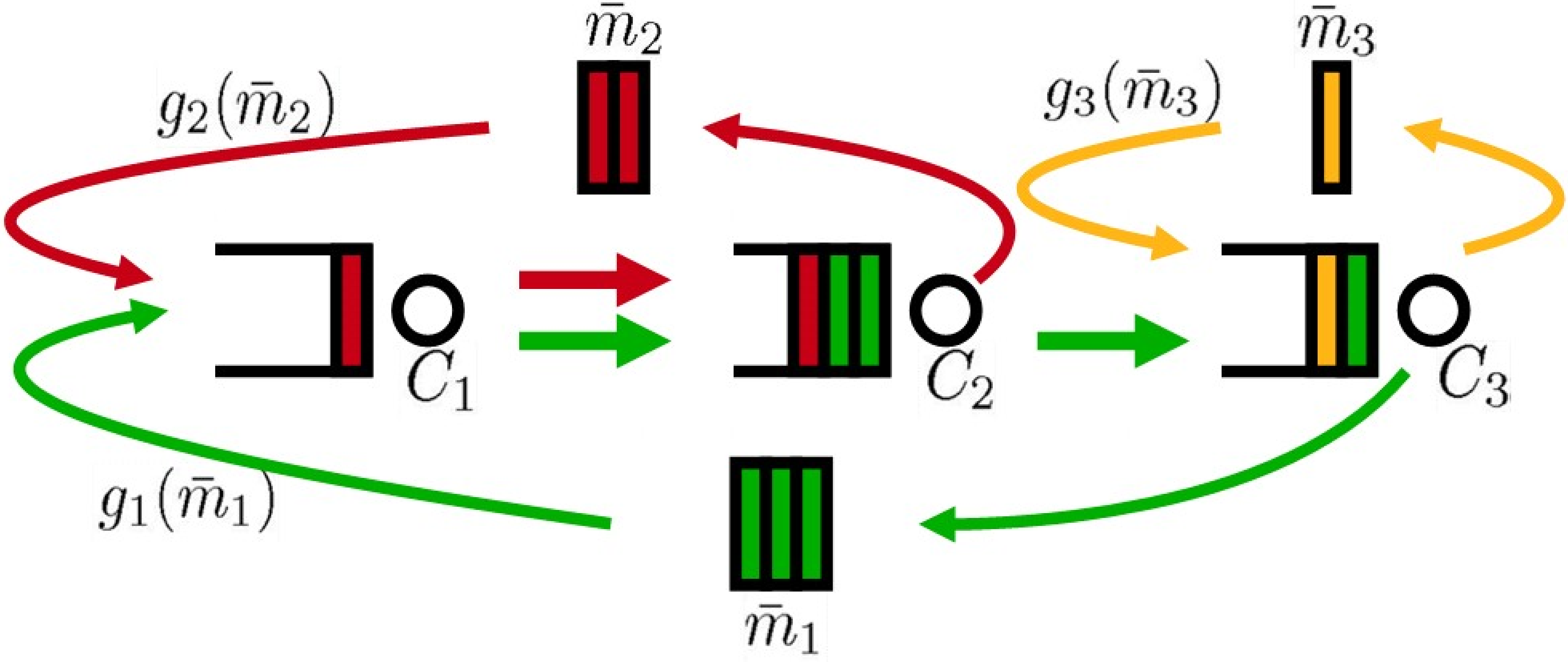


A System of Queues



Stationary distribution:

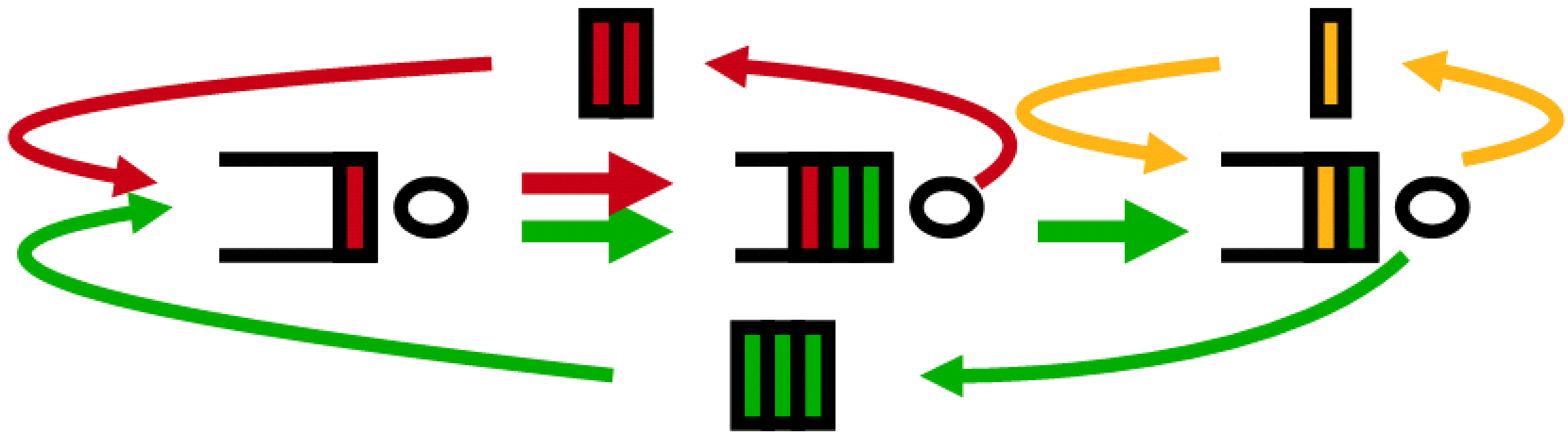
A System of Queues



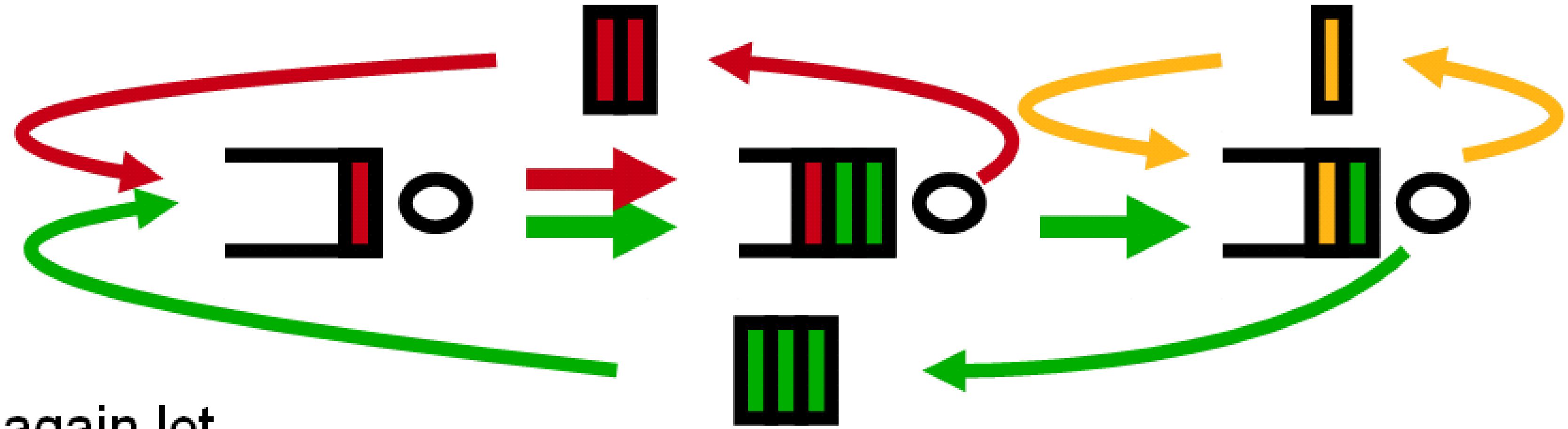
Stationary distribution:

$$\mathbb{P}(M = m) = \frac{1}{B_g} \prod_{j \in \mathcal{J}} \binom{m_j}{m_{ji} \; : \; i \ni j} \frac{1}{C_j^{m_j}} \times \prod_{i \in \mathcal{I}} \prod_{k=1}^{\bar{m}_i} g_i(k)$$

A System of Queues



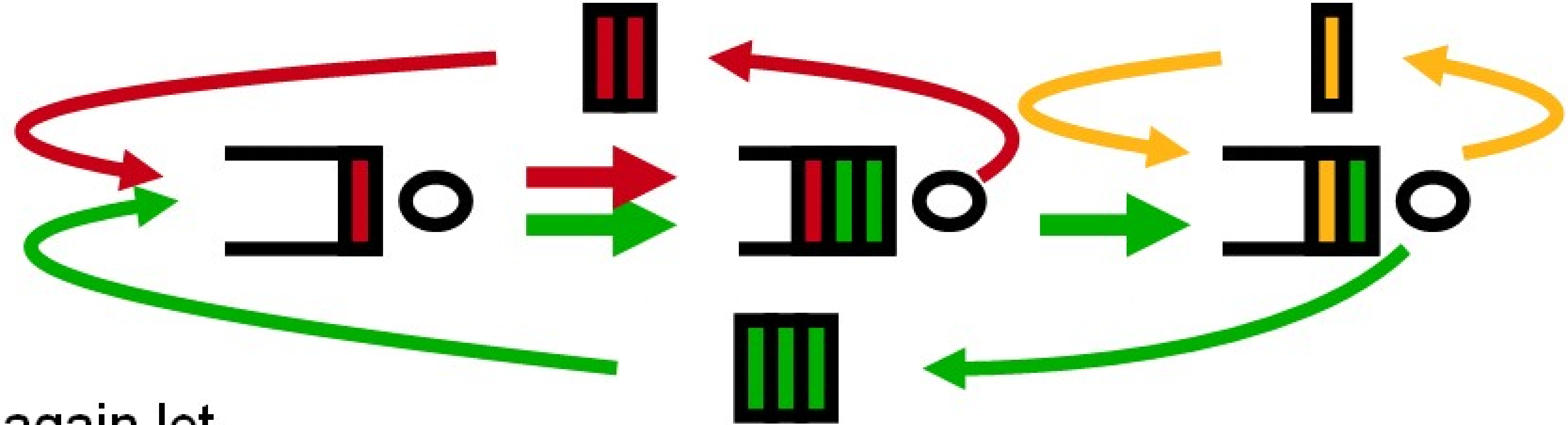
A System of Queues



Once again let

$$g_i^{(c)}(\bar{m}_i) = e^{cG_i\left(\frac{\bar{m}_i+1}{c}\right) - cG_i\left(\frac{\bar{m}_i}{c}\right)}$$

A System of Queues



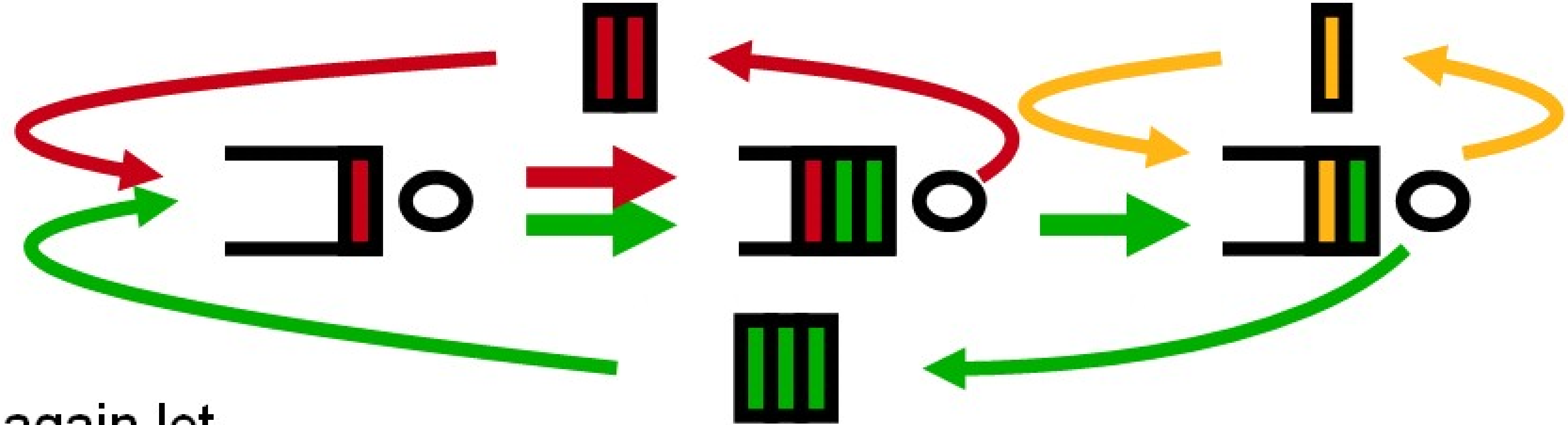
Once again let

$$g_i^{(c)}(\bar{m}_i) = e^{cG_i\left(\frac{\bar{m}_i+1}{c}\right) - cG_i\left(\frac{\bar{m}_i}{c}\right)}$$

Large Deviations

$$\lim_{c\to\infty} \frac{1}{c} \log \mathbb{P}^{(c)}(M^{(c)} = m) = -\alpha_G(m)$$

A System of Queues



Once again let

$$g_i^{(c)}(\bar{m}_i) = e^{cG_i(\frac{\bar{m}_i+1}{c}) - cG_i(\frac{\bar{m}_i}{c})}$$

Large Deviations

$$\lim_{c\to\infty} \frac{1}{c} \log \mathbb{P}^{(c)}(M^{(c)} = m) = -\alpha_G(m)$$

where

$$\alpha_G(m) = \sum_{j,i} m_{ji} \log \frac{m_{ji} C_j}{m_j \rho_i} - \sum_i G_i(\bar{m}_i)$$

with

$$\bar{m}_i = \sum_j m_{ji}$$

A System of Queues

Most likely state

A System of Queues

Most likely state

$$\min_{m,\bar{m}} \quad \sum_{j,i} m_{ji} \log \frac{m_{ji} C_j}{m_j} - \sum_i G_i(\bar{m}_i) \quad \text{subject to} \quad \bar{m}_i = \sum_{j \in i} m_{ji} \quad i \in \mathcal{I}.$$

A System of Queues

Most likely state

$$\min_{m,\bar{m}} \quad \sum_{j,i} m_{ji} \log \frac{m_{ji} C_j}{m_j} - \sum_i G_i(\bar{m}_i) \quad \text{subject to} \quad \bar{m}_i = \sum_{j \in i} m_{ji} \quad i \in \mathcal{I}.$$

Lets calculate its dual

A System of Queues

Most likely state

$$\min_{m,\bar{m}} \quad \sum_{j,i} m_{ji} \log \frac{m_{ji} C_j}{m_j} - \sum_i G_i(\bar{m}_i) \quad \text{subject to} \quad \bar{m}_i = \sum_{j \in i} m_{ji} \quad i \in \mathcal{I}.$$

Lets calculate its dual

$$\min_{m,\bar{m}} L(m, \bar{m}; \lambda) = \min_{m,\bar{m}} \sum_{j,i} m_{ji} \log \frac{m_{ji} C_j}{m_j} - \sum_i G_i(\bar{m}_i) + \sum_i \lambda_i \big(\bar{m}_i - \sum_{j \in i} m_{ji}\big)$$

A System of Queues

Most likely state

$$\min_{m,\bar{m}} \quad \sum_{j,i} m_{ji} \log \frac{m_{ji} C_j}{m_j} - \sum_i G_i(\bar{m}_i) \quad \text{subject to} \quad \bar{m}_i = \sum_{j \in i} m_{ji} \quad i \in \mathcal{I}.$$

Lets calculate its dual

$$\min_{m,\bar{m}} L(m, \bar{m}; \lambda) = \min_{m,\bar{m}} \sum_{j,i} m_{ji} \log \frac{m_{ji} C_j}{m_j} - \sum_i G_i(\bar{m}_i) + \sum_i \lambda_i \Big(\bar{m}_i - \sum_{j \in i} m_{ji}\Big)$$

$$= \min_m \sum_{j,i} m_{ji} \log \frac{m_{ji} C_j}{m_j e^{\lambda_i}} \quad - \quad \sum_i \max_{\bar{m}_i} \{G_i(\bar{m}_i) - \lambda_i \bar{m}_i\}$$

A System of Queues

Most likely state

$$\min_{m,\bar{m}} \quad \sum_{j,i} m_{ji} \log \frac{m_{ji} C_j}{m_j} - \sum_i G_i(\bar{m}_i) \quad \text{subject to} \quad \bar{m}_i = \sum_{j \in i} m_{ji} \quad i \in \mathcal{I}.$$

Lets calculate its dual

$$\min_{m,\bar{m}} L(m,\bar{m};\lambda) = \min_{m,\bar{m}} \sum_{j,i} m_{ji} \log \frac{m_{ji} C_j}{m_j} - \sum_i G_i(\bar{m}_i) + \sum_i \lambda_i \big(\bar{m}_i - \sum_{j \in i} m_{ji}\big)$$

$$= \min_m \sum_{j,i} m_{ji} \log \frac{m_{ji} C_j}{m_j e^{\lambda_i}} \quad - \quad \sum_i \max_{\bar{m}_i} \{G_i(\bar{m}_i) - \lambda_i \bar{m}_i\}$$

$$= \begin{cases} \sum_{i \in \mathcal{I}} U_i(e^{\lambda_i}) & \text{if} \quad \sum_{i : j \in i} e^{\lambda_i} \le C_j, \quad j \in \mathcal{J} \\ -\infty & \text{otherwise.} \end{cases}$$

A System of Queues

Most likely state

$$\min_{m,\bar{m}} \quad \sum_{j,i} m_{ji} \log \frac{m_{ji} C_j}{m_j} - \sum_i G_i(\bar{m}_i) \quad \text{subject to} \quad \bar{m}_i = \sum_{j \in i} m_{ji} \quad i \in \mathcal{I}.$$

Lets calculate its dual

$$\min_{m,\bar{m}} L(m, \bar{m}; \lambda) = \min_{m,\bar{m}} \sum_{j,i} m_{ji} \log \frac{m_{ji} C_j}{m_j} - \sum_i G_i(\bar{m}_i) + \sum_i \lambda_i \Big(\bar{m}_i - \sum_{j \in i} m_{ji}\Big)$$

$$= \min_{m} \sum_{j,i} m_{ji} \log \frac{m_{ji} C_j}{m_j e^{\lambda_i}} \quad - \quad \sum_i \max_{\bar{m}_i} \{G_i(\bar{m}_i) - \lambda_i \bar{m}_i\}$$

$$= \begin{cases} \sum_{i \in \mathcal{I}} U_i(e^{\lambda_i}) & \text{if} \quad \sum_{i: j \in i} e^{\lambda_i} \leq C_j, \quad j \in \mathcal{J} \\ -\infty & \text{otherwise.} \end{cases}$$

We have dual:

A System of Queues

Most likely state

$$\min_{m,\bar{m}} \quad \sum_{j,i} m_{ji} \log \frac{m_{ji} C_j}{m_j} - \sum_i G_i(\bar{m}_i) \quad \text{subject to} \quad \bar{m}_i = \sum_{j \in i} m_{ji} \quad i \in \mathcal{I}.$$

Lets calculate its dual

$$\min_{m,\bar{m}} L(m,\bar{m};\lambda) = \min_{m,\bar{m}} \sum_{j,i} m_{ji} \log \frac{m_{ji} C_j}{m_j} - \sum_i G_i(\bar{m}_i) + \sum_i \lambda_i \Big(\bar{m}_i - \sum_{j \in i} m_{ji}\Big)$$

$$= \min_m \sum_{j,i} m_{ji} \log \frac{m_{ji} C_j}{m_j e^{\lambda_i}} \quad - \quad \sum_i \max_{\bar{m}_i} \{ G_i(\bar{m}_i) - \lambda_i \bar{m}_i \}$$

$$= \begin{cases} \sum_{i \in \mathcal{I}} U_i(e^{\lambda_i}) & \text{if} \quad \sum_{i:j \in i} e^{\lambda_i} \le C_j, \quad j \in \mathcal{J} \\ -\infty & \text{otherwise.} \end{cases}$$

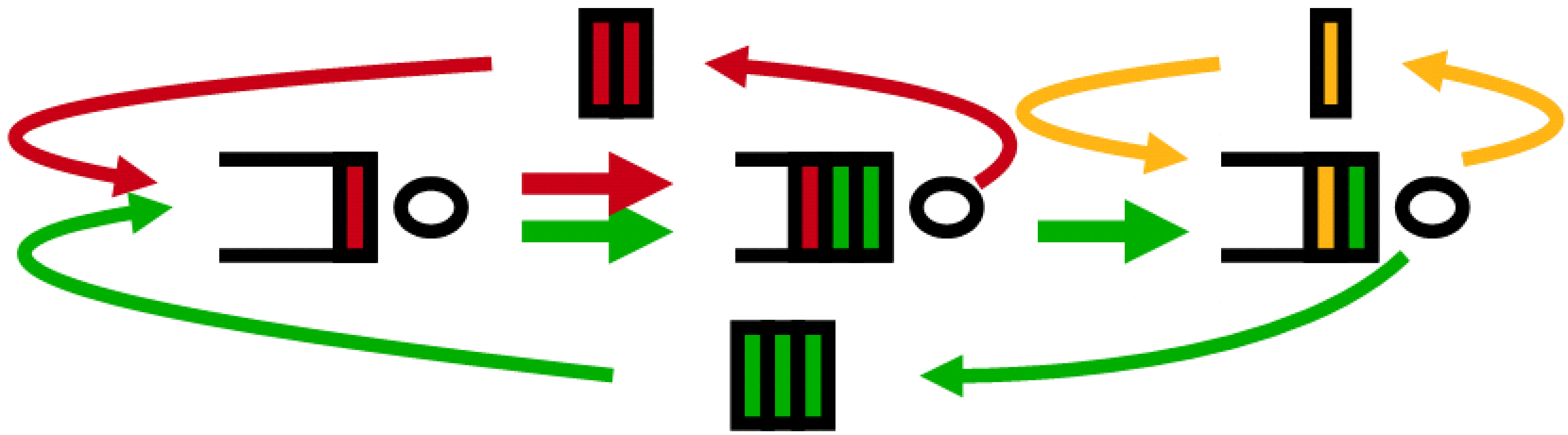
We have dual:

$$\text{maximize} \quad \sum_i U_i(\Lambda_i)$$

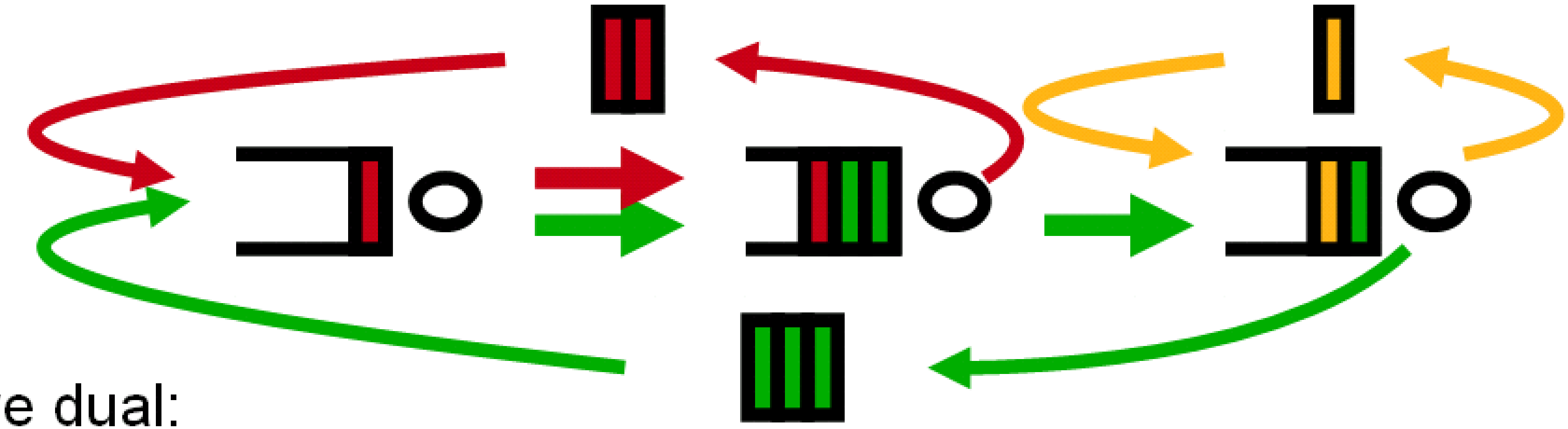
$$\text{subject to} \quad \sum_{i:j \in i} \Lambda_i \le C_j, \quad j \in \mathcal{J}$$

$$\text{over} \quad \Lambda_i \ge 0, \quad i \in \mathcal{I}.$$

A System of Queues

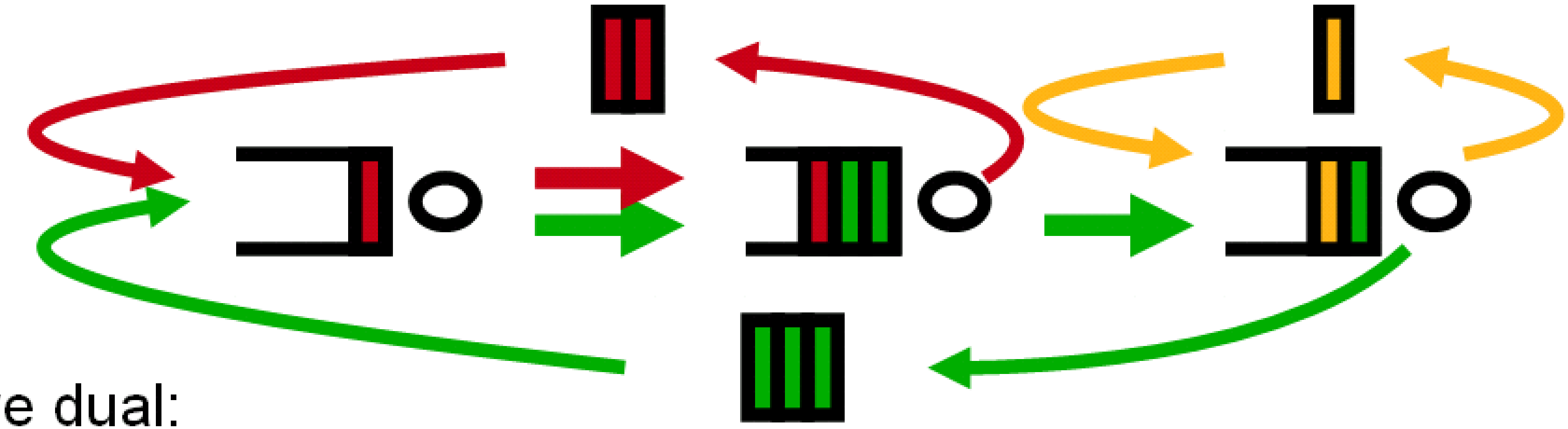


A System of Queues



We have dual:

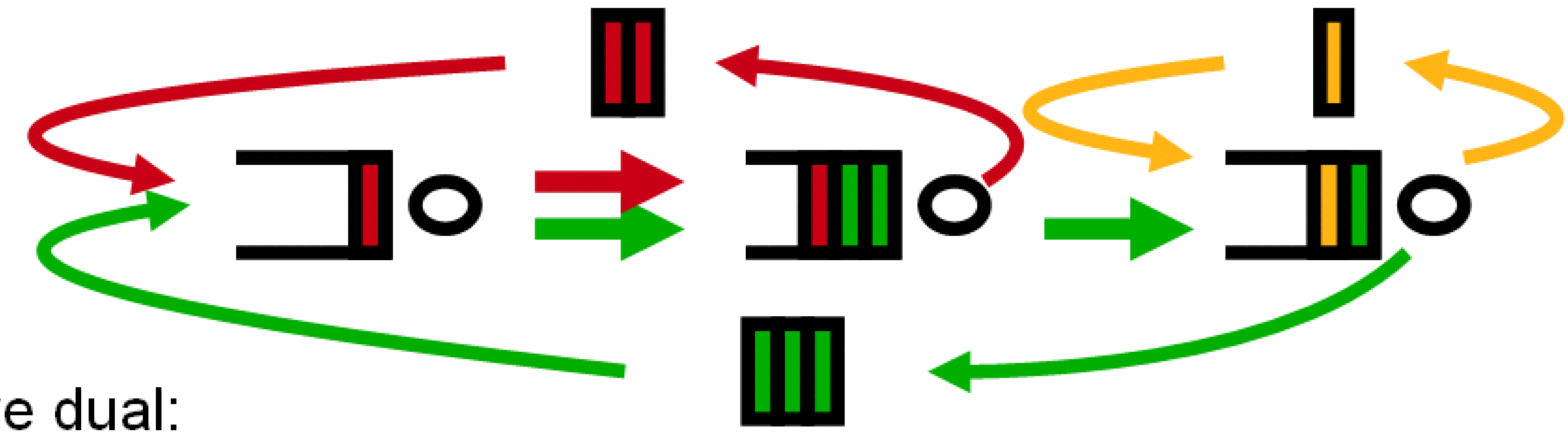
A System of Queues



We have dual:

The SYSYEM PROBLEM

A System of Queues



We have dual:

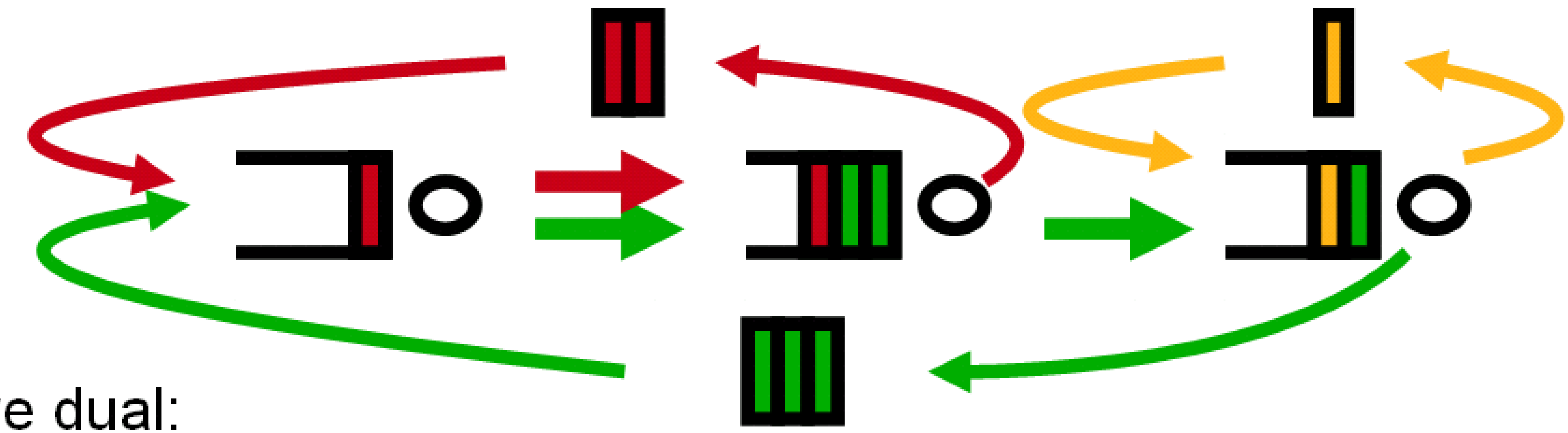
The SYSYEM PROBLEM

$$\text{maximize} \quad \sum_i U_i(\Lambda_i)$$

$$\text{subject to} \quad \sum_{i:j\in i} \Lambda_i \le C_j, \quad j \in \mathcal{J}$$

$$\text{over} \quad \Lambda_i \ge 0, \quad i \in \mathcal{I}.$$

A System of Queues



We have dual:

The SYSYEM PROBLEM

$$\text{maximize} \quad \sum_i U_i(\Lambda_i)$$

$$\text{subject to} \quad \sum_{i:j\in i} \Lambda_i \leq C_j, \quad j \in \mathcal{J}$$

$$\text{over} \quad \Lambda_i \geq 0, \quad i \in \mathcal{I}.$$

The most likely state for the queueing system solves the SYSTEM PROBLEM.

Conclusion

Conclusion

USER PROBLEMS

$$\max_{\bar{m}_i}\{G_i(\bar{m}_i) - \lambda_i \bar{m}_i\}$$

Conclusion

USER PROBLEMS

$$\max_{\bar{m}_i}\{G_i(\bar{m}_i) - \lambda_i \bar{m}_i\}$$

NETWORK PROBLEM

$$\max \sum_{i\in\mathcal{I}} \bar{m}_i \log \Lambda_i$$

$$\text{subject to} \sum_{i:j\in i} \Lambda_i \le C_j,$$

Conclusion

USER PROBLEMS

$$\max_{\bar{m}_i}\{G_i(\bar{m}_i) - \lambda_i \bar{m}_i\}$$

NETWORK PROBLEM

$$\max \sum_{i\in\mathcal{I}} \bar{m}_i \log \Lambda_i$$

$$\text{subject to} \sum_{i:j\in i} \Lambda_i \leq C_j,$$

AND

$$\bar{m}_i = \Lambda_i q_i$$

Conclusion

USER PROBLEMS

$$\max_{\bar{m}_i}\{G_i(\bar{m}_i) - \lambda_i \bar{m}_i\}$$

NETWORK PROBLEM

$$\max \sum_{i\in\mathcal{I}} \bar{m}_i \log \Lambda_i$$

$$\text{subject to} \sum_{i:j\in i} \Lambda_i \le C_j,$$

AND

$$\bar{m}_i = \Lambda_i q_i$$

Window Size

Conclusion

USER PROBLEMS

$$\max_{\bar{m}_i}\{G_i(\bar{m}_i) - \lambda_i \bar{m}_i\}$$

NETWORK PROBLEM

$$\max \sum_{i \in \mathcal{I}} \bar{m}_i \log \Lambda_i$$

$$\text{subject to} \sum_{i: j \in i} \Lambda_i \leq C_j,$$

AND

$$\bar{m}_i = \Lambda_i q_i$$

Window Size

Delay

Conclusion

USER PROBLEMS

$$\max_{\bar{m}_i}\{G_i(\bar{m}_i) - \lambda_i \bar{m}_i\}$$

NETWORK PROBLEM

$$\max \sum_{i \in \mathcal{I}} \bar{m}_i \log \Lambda_i$$

$$\text{subject to} \sum_{i: j \in i} \Lambda_i \le C_j,$$

AND

Little's Law

$$\bar{m}_i = \Lambda_i q_i$$

Window Size

Delay

Conclusion

USER PROBLEMS

$$\max_{\bar{m}_i}\{G_i(\bar{m}_i) - \lambda_i \bar{m}_i\}$$

NETWORK PROBLEM

$$\max \sum_{i\in\mathcal{I}} \bar{m}_i \log \Lambda_i$$
$$\text{subject to} \sum_{i:j\in i} \Lambda_i \le C_j,$$

Network choses prices with **Queue sizes**:

$$q_i = \sum_{j\in i} q_j$$

AND

Little's Law

$$\bar{m}_i = \Lambda_i q_i$$

Window Size

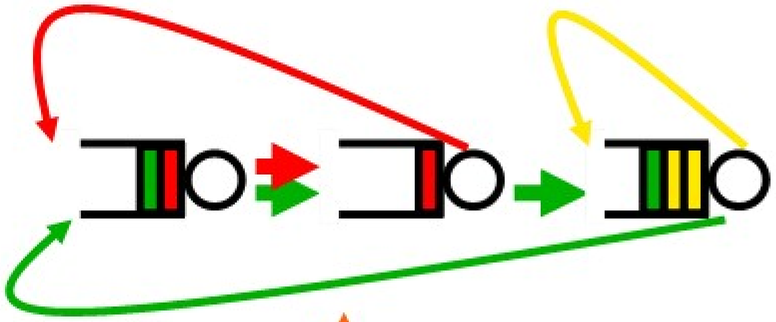
Delay

Conclusion

USER PROBLEMS

$$\max_{\bar{m}_i}\{G_i(\bar{m}_i) - \lambda_i \bar{m}_i\}$$

NETWORK PROBLEM



Network choses prices with **Queue sizes**:

$$q_i = \sum_{j \in i} q_j$$

AND

Little's Law

$$\bar{m}_i = \Lambda_i q_i$$

Window Size

Delay

Conclusion

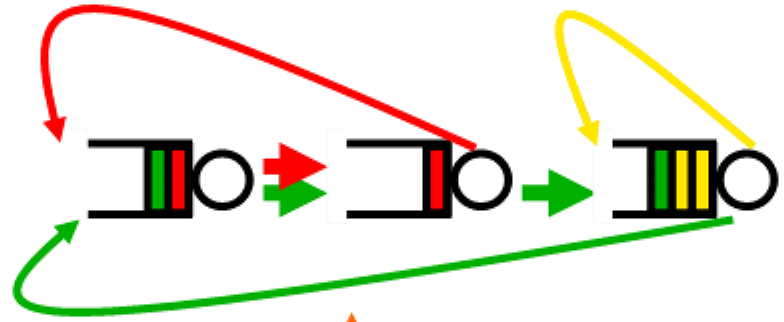
USER PROBLEMS

$$\max_{\bar{m}_i}\{G_i(\bar{m}_i) - \lambda_i \bar{m}_i\}$$

User *i* choses
Congestion window:

$$\bar{m}_i$$

NETWORK PROBLEM



Network choses prices
with **Queue sizes**:

$$q_i = \sum_{j \in i} q_j$$

AND

Little's Law

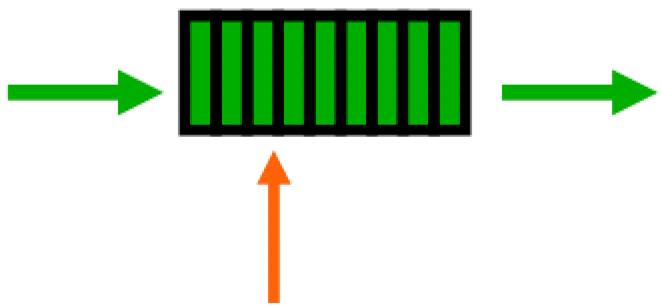
$$\bar{m}_i = \Lambda_i q_i$$

Window
Size

Delay

Conclusion

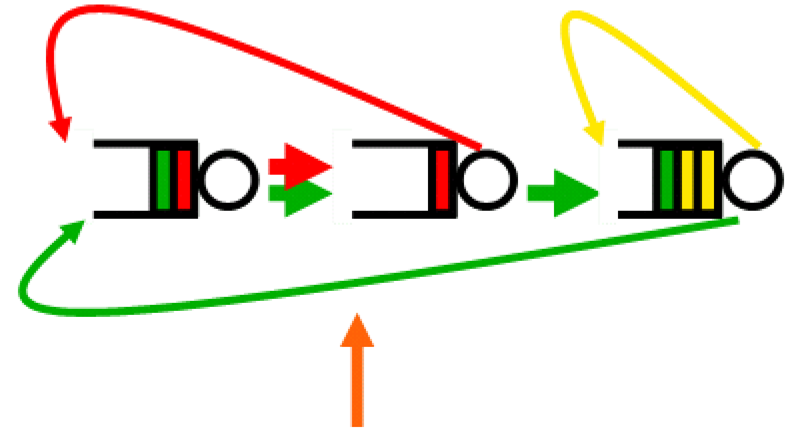
USER PROBLEMS



User *i* choses
Congestion window:

$$\bar{m}_i$$

NETWORK PROBLEM



Network choses prices
with **Queue sizes**:

$$q_i = \sum_{j \in i} q_j$$

AND

Little's Law

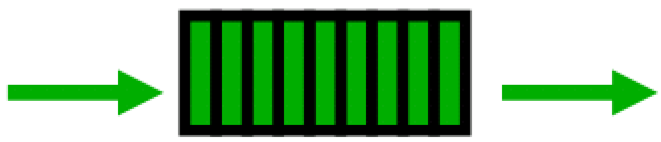
$$\bar{m}_i = \Lambda_i q_i$$

Window Size

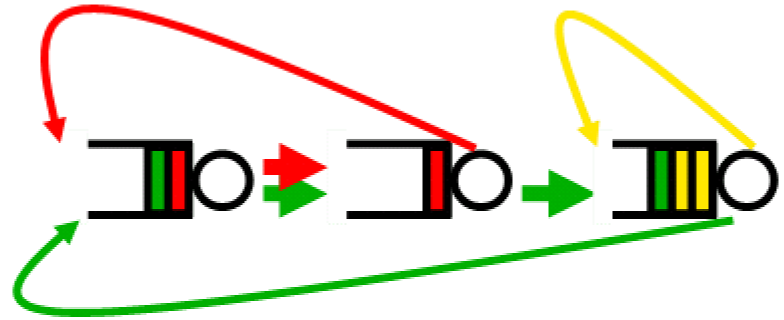
Delay

Conclusion

USER PROBLEMS



NETWORK PROBLEM



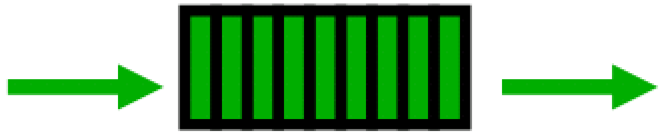
AND

Little's Law

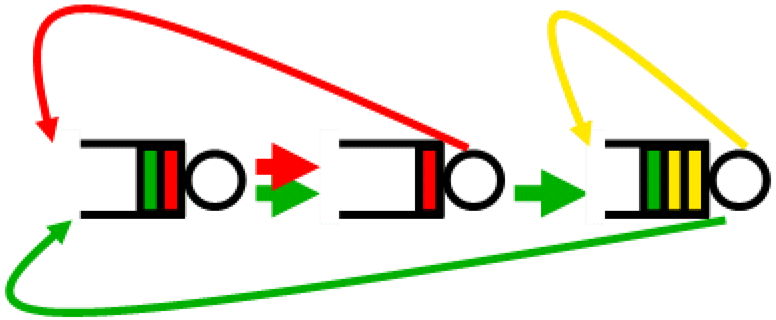
$$\bar{m}_i = \Lambda_i q_i$$

Conclusion

USER PROBLEMS



NETWORK PROBLEM



AND

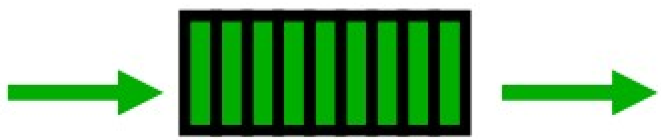
Little's Law

$$\bar{m}_i = \Lambda_i q_i$$

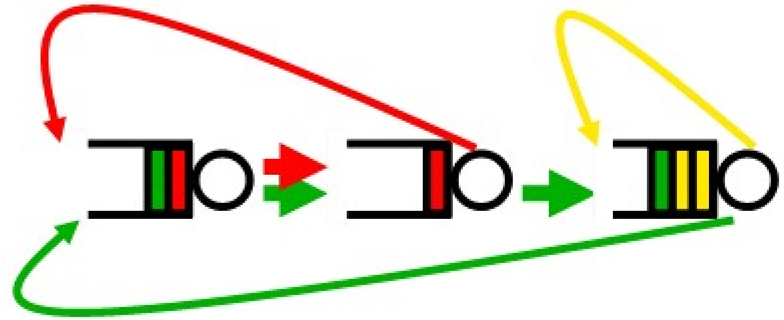
Solves

Conclusion

USER PROBLEMS



NETWORK PROBLEM



AND

Little's Law

$$\bar{m}_i = \Lambda_i q_i$$

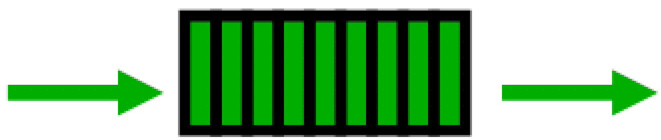
Solves

SYSTEM PROBLEM

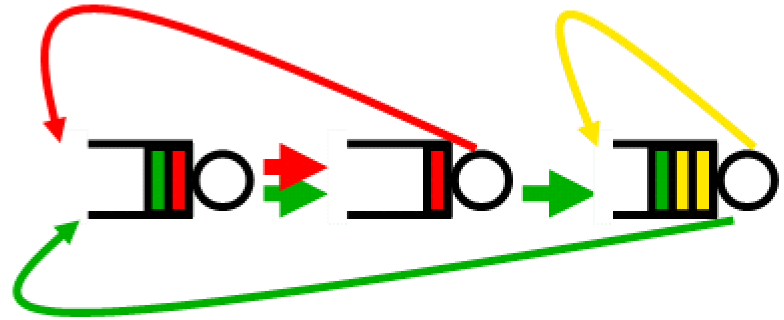
$$\begin{aligned} &\max && \sum_{i \in \mathcal{I}} U_i(\Lambda_i) \\ &\text{subject to} && \sum_{i : j \in i} \Lambda_i \leq C_j, \; j \in \mathcal{J} \\ &\text{over} && \Lambda_i \geq 0, \quad i \in \mathcal{I}. \end{aligned}$$

Conclusion

USER PROBLEMS



NETWORK PROBLEM



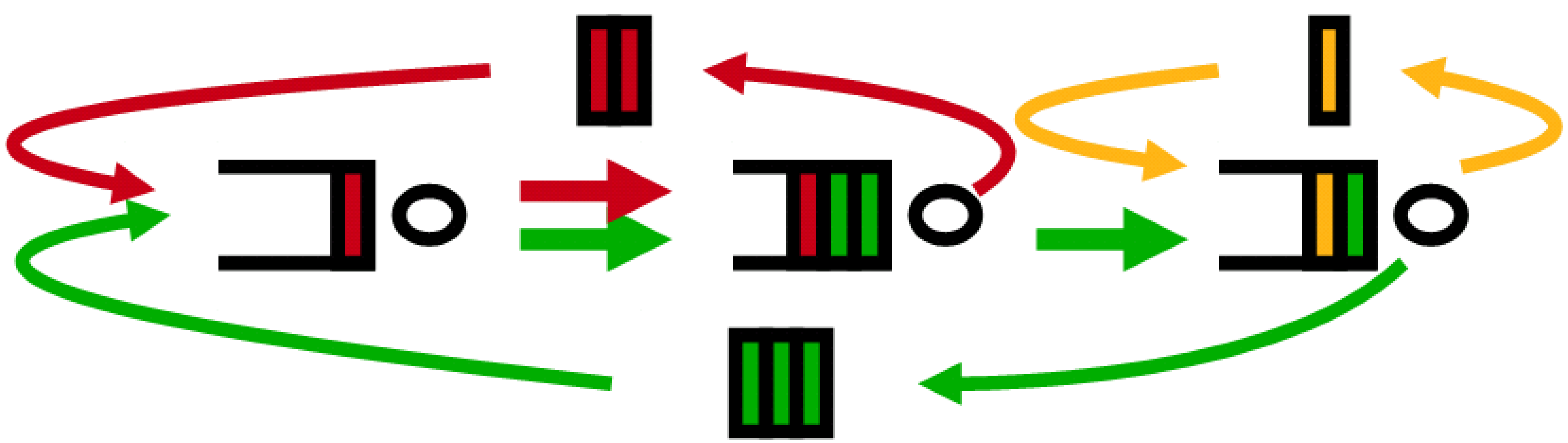
AND

Little's Law

$$\bar{m}_i = \Lambda_i q_i$$

Solves

SYSTEM PROBLEM



THANK YOU FOR LISTENING!