Insensitivity Revisited

Stan Zachary

Heriot-Watt University

9 June 2009

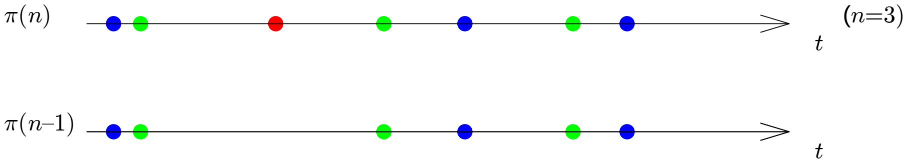
Outline

Single-resource loss system – original model

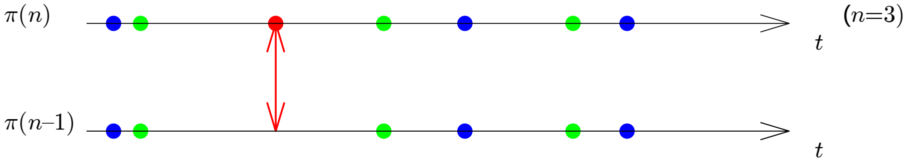
- Single-resource loss system with capacity C
- Individuals (customers, calls, or jobs) arrive as Poisson process rate α and are accepted subject to the capacity constraint
- Individual workloads are i.i.d. with mean 1
- When there are n individuals in the system ($n \leq C$), it works at rate n, dividing this effort equally (processor sharing)

Single-resource loss system – associated closed model

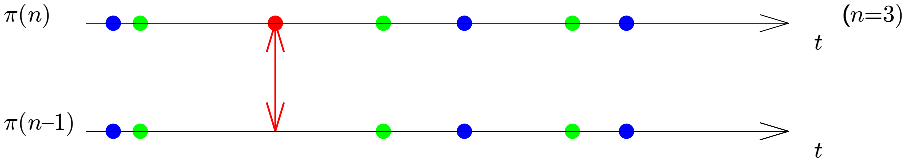
- As original model, but no arrivals and no departures
- When an individual's workload is complete, it immediately acquires a new one and remains in the system
- Hence n independent stationary renewal processes
- System has n individuals with probability $\pi(n)$, $n = 0, 1, \ldots, C$



Single-resource loss system – comparison of closed and original models

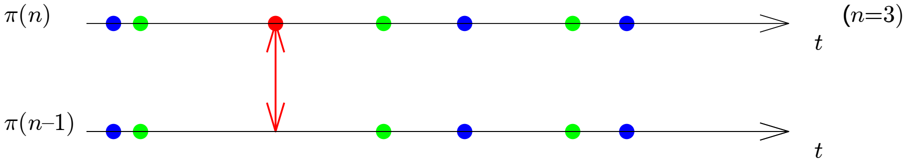


Single-resource loss system – comparison of closed and original models



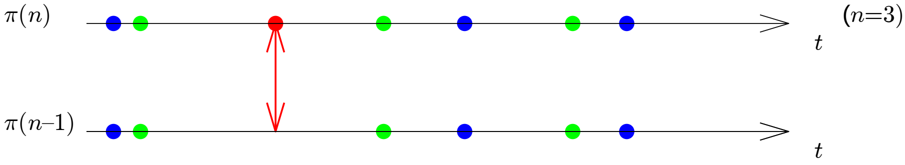
- Now modify the stationary closed model by allowing arrivals (rate α) and departures to obtain the original model

Single-resource loss system – comparison of closed and original models



- Now modify the stationary closed model by allowing arrivals (rate α) and departures to obtain the original model
- A departing individual leaves the system stationary and an arriving individual finds the system stationary

Single-resource loss system – comparison of closed and original models



- Now modify the stationary closed model by allowing arrivals (rate α) and departures to obtain the original model
- A departing individual leaves the system stationary and an arriving individual finds the system stationary
- Hence the closed system and the original system are indistinguishable—and so the latter is also stationary—provided

$$\pi(n)n = \pi(n-1)\alpha$$

Outline

Example: Erlang loss system (Pechinkin, 1987)

Single-class networks

Multi-class networks

Single-class networks: model

Whenever there are n individuals in the system:

- Individuals arrive as Poisson process rate $\alpha(n)$
- Individual workloads are i.i.d. with mean 1
- When there are n individuals in the system ($n \geq 0$), it works at total rate $\beta(n)$ (with $\beta(0) = 0$), dividing this effort equally (processor sharing)

Single-class networks: result

Theorem

Suppose that the distribution π on $\mathbb{Z}_+$ is the solution of the balance equations

$$\pi(n)\beta(n) = \pi(n-1)\alpha(n-1), \qquad n \geq 1, \tag{1}$$

and that

$$\sum_{n \geq 0} \pi(n)\alpha(n) < \infty.$$

Then the distribution π is stationary for the associated number of individuals in the system.

Single-class networks: result

Theorem

Suppose that the distribution π on $\mathbb{Z}_+$ is the solution of the balance equations

$$\pi(n)\beta(n) = \pi(n-1)\alpha(n-1), \qquad n \geq 1, \tag{1}$$

and that

$$\sum_{n \geq 0} \pi(n)\alpha(n) < \infty.$$

Then the distribution π is stationary for the associated number of individuals in the system.

Further, under the stationary distribution, and conditional on there being n individuals in the system, the distribution of their residual workloads is the same as that of n independent stationary renewal processes (with inter-event distribution that of the original workload).

Proof.

Introduce Markovian framework by defining the state of the system at any time to be the number of individuals together with their residual workloads.

Proof.

Introduce Markovian framework by defining the state of the system at any time to be the number of individuals together with their residual workloads.

Consider also the modified closed system in which neither arrivals nor departures are permitted, and in which workloads are continually renewed.

Proof.

Introduce Markovian framework by defining the state of the system at any time to be the number of individuals together with their residual workloads.

Consider also the modified closed system in which neither arrivals nor departures are permitted, and in which workloads are continually renewed.

Let $(P_t)_{t\geq 0}$ and $(\hat{P}_t)_{t\geq 0}$ be the semigroups of transition kernels associated respectively with the original and closed processes.

Proof.

Introduce Markovian framework by defining the state of the system at any time to be the number of individuals together with their residual workloads.

Consider also the modified closed system in which neither arrivals nor departures are permitted, and in which workloads are continually renewed.

Let $(P_t)_{t\geq 0}$ and $(\hat{P}_t)_{t\geq 0}$ be the semigroups of transition kernels associated respectively with the original and closed processes.

For any distribution π on $\mathbb{Z}_+$, let μ_π be the distribution of the closed system in which π is the distribution of the number n of individuals in the system, and in which, conditional on n, the corresponding n renewal processes are independent and stationary. Then, for any π,

$$\mu_\pi \hat{P}_t = \mu_\pi \qquad \text{for all } t > 0.$$

Proof.

Introduce Markovian framework by defining the state of the system at any time to be the number of individuals together with their residual workloads.

Consider also the modified closed system in which neither arrivals nor departures are permitted, and in which workloads are continually renewed.

Let $(P_t)_{t\geq 0}$ and $(\hat{P}_t)_{t\geq 0}$ be the semigroups of transition kernels associated respectively with the original and closed processes.

For any distribution π on $\mathbb{Z}_+$, let μ_π be the distribution of the closed system in which π is the distribution of the number n of individuals in the system, and in which, conditional on n, the corresponding n renewal processes are independent and stationary. Then, for any π,

$$\mu_\pi \hat{P}_t = \mu_\pi \qquad \text{for all } t > 0.$$

But, under the balance condition (1), we have

$$\mu_\pi \hat{P}_t = \mu_\pi P_t \qquad \text{for all } t > 0.$$

Single-class networks: generalisation

In the case where we do not have processor-sharing, the same insensitivity result continues to hold, provided the discipline is symmetric in the sense of Kelly ("Rev. & Stoch. Networks").

Single-class networks: generalisation

In the case where we do not have processor-sharing, the same insensitivity result continues to hold, provided the discipline is symmetric in the sense of Kelly ("Rev. & Stoch. Networks").

Essentially departures must be seamlessly substituted for by arrivals (in terms of probability fluxes) without disruption of the work-sharing discipline.

Single-class networks: generalisation

In the case where we do not have processor-sharing, the same insensitivity result continues to hold, provided the discipline is symmetric in the sense of Kelly ("Rev. & Stoch. Networks").

Essentially departures must be seamlessly substituted for by arrivals (in terms of probability fluxes) without disruption of the work-sharing discipline.

Then the stationary distribution π is again as given by the balance equations (1), and, under stationarity, the residual workload of each individual is again as given by the stationary residual workload in the corresponding renewal process.

Single-class networks: generalisation

In the case where we do not have processor-sharing, the same insensitivity result continues to hold, provided the discipline is symmetric in the sense of Kelly ("Rev. & Stoch. Networks").

Essentially departures must be seamlessly substituted for by arrivals (in terms of probability fluxes) without disruption of the work-sharing discipline.

Then the stationary distribution π is again as given by the balance equations (1), and, under stationarity, the residual workload of each individual is again as given by the stationary residual workload in the corresponding renewal process.

Example: "last-in-first-out preemptive resume"

Outline

Multi-class networks: no transitions between classes

Here, under the same symmetry condition (e.g. processor sharing) the same insensitivity result again holds.

Multi-class networks: no transitions between classes

Here, under the same symmetry condition (e.g. processor sharing) the same insensitivity result again holds.

The stationary joint distribution of the number of calls in each class again being given by the solution of the corresponding detailed balance equations.

Multi-class networks: no transitions between classes

Here, under the same symmetry condition (e.g. processor sharing) the same insensitivity result again holds.

The stationary joint distribution of the number of calls in each class again being given by the solution of the corresponding detailed balance equations.

Example: Multi-class, processor-sharing loss network, in which individuals are admitted subject to overall capacity constraints.

Multi-class networks: transitions between classes

Here again we require the same symmetry condition with regard to arrivals and departures in each class (e.g. processor sharing).

Multi-class networks: transitions between classes

Here again we require the same symmetry condition with regard to arrivals and departures in each class (e.g. processor sharing).

However, we now require the transition rates between classes to be such that, under stationarity, for each state $\mathbf{n}$ and for each class i, we have partial balance between the total flows out of and into class i.

Multi-class networks: transitions between classes

Here again we require the same symmetry condition with regard to arrivals and departures in each class (e.g. processor sharing).

However, we now require the transition rates between classes to be such that, under stationarity, for each state $\mathbf{n}$ and for each class i, we have partial balance between the total flows out of and into class i.

Then again the same insensitivity result holds, the stationary distribution being given by the solution of the partial balance equations.

Multi-class networks: transitions between classes

Here again we require the same symmetry condition with regard to arrivals and departures in each class (e.g. processor sharing).

However, we now require the transition rates between classes to be such that, under stationarity, for each state $\mathbf{n}$ and for each class i, we have partial balance between the total flows out of and into class i.

Then again the same insensitivity result holds, the stationary distribution being given by the solution of the partial balance equations.

Example: processor-sharing Whittle network.

Multi-class networks: transitions between classes

Here again we require the same symmetry condition with regard to arrivals and departures in each class (e.g. processor sharing).

However, we now require the transition rates between classes to be such that, under stationarity, for each state $\mathbf{n}$ and for each class i, we have partial balance between the total flows out of and into class i.

Then again the same insensitivity result holds, the stationary distribution being given by the solution of the partial balance equations.

Example: processor-sharing Whittle network.

Special case: processor-sharing Jackson network – in which the stationary distribution has product form.

References

- Bonald, T. and Proutière, A. (2002). Insensitivity in processor-sharing networks. *Performance Evaluation*, **49**, 193–209.
- Burman, D. Y., Lehoczky, J. P. and Lim, Y. (1984). Insensitivity of blocking probabilities in a circuit switching network. *J. Appl. Prob.* **21**, 850–859.
- Kelly, F. P. (1979). Reversibility and Stochastic Networks. Wiley, Chichester.
- Pechinkin, A. V. (1987). A new proof of Erlang's formula for a lossy multichannel queueing system. *Soviet J. Comput. System Sci.* **25**, 165–168. Translated from: *Izv. Akad. Nauk. SSSR. Techn. Kibernet*, (1986), No. 6, 172–175 (in Russian).
- Whittle, P. (1985). Partial balance and insensitivity. *J. Appl. Prob.*, **22**, 168–176.