Ergodicity properties of rerouting strategies in queueing networks

R. Boel[1,2] & J. Van Horebeek [1,3]

Published in Queueing Systems, 1996

Abstract *In this paper we study rerouting policies for queueing systems with two Poisson arrival streams and two exponential servers. The arrival customers can be rerouted from their normal server to the other server depending on limited information on the two queue length. The policies are compared on the basis of necessary and sufficient conditions for ergodicity. For this purpose we make use of Lyapunov functions.*

1 Introduction

In the seminal work of Foster [4], Lyapunov functions were used as a tool for finding necessary conditions for ergodicity of simple Markov processes. In recent years there has been a lot of work on the much harder problem of finding necessary and sufficient conditions for ergodicity, using test functions similar to Lyapunov functions. The work of Malyshev [3],[6],[9],[10] on random walks in simplices with boundary conditions, the work of Hajek [5] on adaptive ALOHA, and the work of Meyn and Tweedie [11] should be cited in this respect. Application of this work to specific examples with several regions of uniform transition probabilities, with fairly general behavior on the boundaries between the regions, turns out to be often quite hard.

In this paper we treat a few examples of Markov processes occurring in the analysis of rerouting strategies for queueing systems with several arrival streams and several servers. Specifically we consider models with two independent Poisson arrival streams and two exponential service stations. We assume that arrival stream i, $i = 1, 2$, is associated to and normally served by its preferred service station i. However, provided one pays a certain penalty, it is possible to have tasks from arrival stream i served on station $3-i$. To obtain a simple Markovian model we assume that the penalty consists of creating t ($t \geq 1$) tasks instead of one task each time a rerouting away from the preferred service station occurs. We consider two rerouting strategies. Rerouting can occur if the preferred service station has a queue length exceeding a certain threshold and if in the other one the number of jobs is less than a given threshold. Another rerouting strategy reroutes a task as soon as the preferred queue length is longer, by a certain threshold, than the other queue length. In both cases we consider both the asymmetrical case where only tasks from arrival stream 1 can be rerouted to server

[1]Department of Mathematics, K.U.Leuven, Heverlee, Belgium

[2]Senior Research Associate N.F.W.O. (Belgian National Foundation for Scientific Research), Vakgroep Elektrische Energietechniek, Universiteit Gent, Technologiepark-Zwijnaarde, B-9052 Gent, Belgium. Part of the research leading to this paper was carried out within the Belgian Programme on Interuniversity Poles of Attraction, initiated by the Belgian State, Prime Minister's Office of Science, Technology and Culture. The scientific responsibility rests with the authors

[3]CIMAT, Apartado Postal 402, 36000 Guanajuato, Mexico

2 but no rerouting from stream 2 to server 1 occurs, and the symmetrical case where rerouting in both directions is considered.

The methods of Malyshev can not be applied directly because in our model the positive quadrant is divided in two parts, with different transition probabilities. Hence, besides the boundaries where one of the queues becomes empty, we also have to consider interior boundaries, where the transition probabilities change discontinuously. This interior discontinuity poses a number of new problems for the analysis. By extending the methods of Malyshev [10] and Hajek [5] we succeed in obtaining, for the different rerouting strategies, a partition of the parameter space in an ergodic region and a transient region. To simplify the graphical representation we take the service rate μ the same in both service stations. The (λ_1, λ_2)-plane is then partitioned in a region where ergodicity is ensured, and a region where transient behavior is certain. Only on the one-dimensional boundary between the two regions can no conclusion be drawn.

The above model is an abstraction of many interesting problems in communication and computer sciences. In multi-processor computing systems, or in distributed database systems, each task can in principle be assigned to any processor [1],[13]. However since the different processors may have different "bits" of information in RAM or in fast peripheral memory, the speed of executing tasks on different processors may be different. In communication networks, each route between the source and destination node can be considered as a service station. However there is usually a shortest route between any two nodes, and normally the network should use this shortest route since it uses the least amount of bandwidth. Only when this shortest route is heavily overloaded, should the network management decide to use alternate routes, which are longer and thus require more bandwidth [7], [2]. Similar problems are encountered in mobile or satellite communication, where most connections can be established via one of several alternative ground stations.

It is clear from the applications described in the above paragraph that our model is a severe idealization. The assumptions of Poisson arrivals and exponential service times are probably not very restrictive. We believe the main limitation of our analysis is that only the case of two arrival streams and of two servers has been treated. Extending the methods of this paper to more arrival streams and more servers without further analysis is very cumbersome since the number of different transition regions to be considered grows exponentially fast. In the case where the system has a ring-like structure - with tasks being rerouted only to the left or right neighbour - the inductive methods as used in Gregoriades et al. [16] may prove useful. Also the fact that some of our techniques resemble fluid approximation techniques gives some hope that extensions to larger systems are possible.

In section 2 of this paper we give the generalization of Foster's criterion in the form in which it is needed in this paper. This is based on theorems of Hajek [5], Malyshev [10] and Meyn and Tweedie [11]. In section 3, we treat in detail the case based on rerouting if the difference between the two queue lengths exceeds a given constant. In section 4 we discuss and give, without proof, the results for the case of rerouting based on one queue exceeding a threshold. Both in section 3 and 4 we give some graphs illustrating how the proposed rerouting strategies influence the ergodic and the transient region in the parameter space. In section 5 we discuss some difficulties in extending the results to more realistic models, and we draw some conclusions based on the analysis of this

paper.

Note that in this paper we assume that all Markov processes under consideration are discrete-time, stationary, with state space a subset of Z^n.

2 Lyapunov Functions

The technique of Lyapunov functions as we will need it, is based on the following property:

Theorem 2.1 *Consider a Markov process $\{X_i\}$ with known initial condition X_0 and known transition probabilities. Consider a positive valued measurable function $V(X_i)$ where we assume that the function V achieves its minimum value 0 in at least one point x_0 in its state space. Assume there exist constants $\alpha, d > 0$ such that for v large enough:*

$$P(|V(X_i) - V(X_{i-1})| = v | X_{i-1}) < d \exp(-\alpha v). \tag{1}$$

Denote by T_b the first passage time to b, for $b > 0$:

$$T = \begin{cases} \inf_t\{t \geq 0 : V(X_t) \leq b\} & \text{if } V(X_0) > b \\ 0 & \text{if } V(X_0) \leq b. \end{cases}$$

1. *If for some $\epsilon > 0$ and all i:*

$$E(V(X_{i+1}) - V(X_i) | V(X_i) > b) \leq -\epsilon, \tag{2}$$

then there exist constants c and δ_2 such that for any i:

$$P(V(X_i) > 0) < c \exp(-\delta_2 i) \text{ and } E(T) < \infty \tag{3}$$

and the Markov process X_i is geometrically ergodic.

2. *If for some $\epsilon > 0$ and all i:*

$$E(V(X_{i+1}) - V(X_i) | V(X_i) > b) \geq \epsilon, \tag{4}$$

then

$$P(T = \infty | V(X_0) > b) > 0 \tag{5}$$

and the Markov chain is transient.

Proof
The proof can be found in [11].

□

The non-negative functions V used in the above theorem are called Lyapunov functions or test functions. The expected increment $E(V(X_{i+1}) - V(X_i)|X_i = x)$ of the test function, are called the (V-)drift functions. If no explicit test function is mentioned, we use the identity test function, $V(x) = x$, in our calculation. In the sequel we restrict

ourselves to linear test functions only. Note that Malyshev and Mensikov [10] show that the "best possible" test function is the expected return time to a compact region around the origin, and this is linear in the state far away from the origin. However there is of course no guarantee that the test functions we use, coincide with these test functions, and so we are not a priori guaranteed that we obtain the best possible ergodicity regions. However we will show that outside the closure of our ergodicity region the system is transient, so that indeed we obtain the best possible results. In general the technique [8] of choosing a Lyapunov function so that the average drift points towards the origin outside a compact region around the origin, does not lead to necessary and sufficient ergodicity conditions. Our method almost achieves this - only on the boundary between the two regions in the parameter space are we unable to draw a conclusion.

Let us illustrate the above with an easy application (cfr. [9]):

Corollary 2.1 *Suppose that one has given a two dimensional random process X_i with exponentialy bounded jumps and with two different drift vectors $d^1 = E(X_i|X_{i-1} = x_1) - x_1$, $d^2 = E(X_i|X_{i-1} = x_2) - x_2$ in the regions resp. D^1 and D^2.*

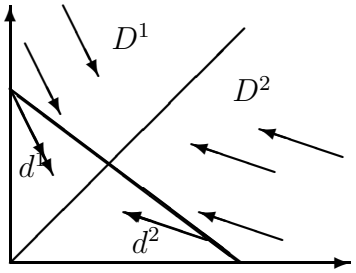


figure 1

If there exists a line with a negative slope so that all the drift vectors starting at this line (outside a compact region around the origin) have their endpoint under it (see figure 1), then we can construct a linear Lyapunov function such that all equipotentials are parallel to this. This function will satisfy (2) and the process will be ergodic. If all the vectors lie above the equipotential surface, (4) is satisfied and the system is transient.

It is easy to see that in case of only two different drift vector each with at least one negative component, it is always possible to find equipotentials such that all vectors lie above it or all below it. In case of one drift vector with two positive components, the process will always be non ergodic. This can be easily generalized to higher dimensions. In order to obtain only two drift vectors (e.g. to get rid of drift vectors at the boundaries which are different from the drift vectors inside a region) we use the following two techniques:

1. consider m-step vectors in some places (i.e. consider $E(V(X_{i+m})-V(X_i)|X_i = x)$;
2. aggregate several states into one state (by skipping time steps).

These techniques amount to calculating the drift averaged over several steps (where the number of steps can be deterministic or random depending on the case). This is

somewhat similar to the "fluid limit" approximation used by Dai [14], Dai and Meyn [15] and others. These fluid models can be applied to higher dimensional models, giving us hope of extending our analysis also to higher dimensional models.

3 Necessary and Sufficient Conditions for Ergodicity of Queueing Systems with Rerouting

3.1 Model specification

In this section we derive conditions for ergodicity and transience for the system with rerouting based on the difference between the queue lengths (symmetrical and asymetrical case). We consider the following situation:

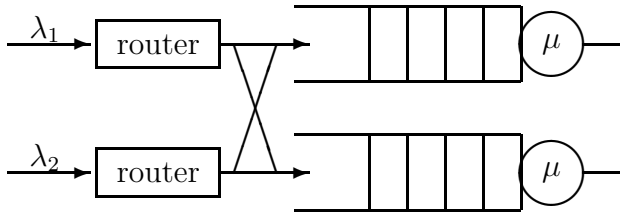


figure 2

Both arrival processes are Poisson and independent of each other. All the service times are independent, exponentially distributed random variables, with parameter μ. Each rerouted task with normal server i generates t tasks when it is assigned to server $3 - i$. Since we are only interested in ergodicity, not in the calculation of the equilibrium distribution, we can consider the embedded Markov chain observing the system at each arrival and departure epoch, with the additional assumption that $\lambda_1 + \lambda_2 + \mu + \mu = 1$. It turns out that the analysis is simplified by also considering dummy or virtual departures, occurring when the queues are empty. This means that we observe the system at all the jump epochs of the 4 Poisson processes of arrivals and of potential departures, i.e. besides the arrival Poisson processes we also define two Poisson processes with rate μ such that all the actual departures from server i are included in the Poisson process of potential departures i. Note that the equilibrium distribution of the embedded Markov chain will thus be the same as the equilibrium distribution of the original continuous time process. However for our analysis the extended embedding is useful in simplifying the aggregation step of subsection 3.3 below.

If we call $x_i(n)$ the length of queue i at the time of the nth event (arrival or departure), tasks are rerouted from queue 1 to queue 2 iff $(x_1(n) - x_2(n)) \geq k$ provided the nth event is an arrival. In the symmetrical case we have additionally that tasks will be rerouted from queue 2 to 1 if $(x_2(n) - x_1(n)) \geq k$.

3.2 Main results

For the asymetrical case we describe in the next property a complete characterisation of the ergodicity region for all values of k and t. For the symetrical case however, the

expressions become quite complicated since the solutions are expressed in terms of the zero's of a higher order polynomial. Therefore we derive only explicitly the formulas for the case $k = 1$, $t = 2$ and illustrate graphically the influence of other values of the two parameters on the ergodicity region.

Property 3.1

• *Rerouting based on differences in queue length (asymmetrical case) is ergodic iff*

$$\lambda_2 + t(\lambda_1 - \mu) < \mu \tag{6}$$

which gives a non-trivial result only when $\lambda_1 > \mu > \lambda_2$*;*

• *Rerouting based on differences in queue length (symmetrical case) for* $t = 2$ *and* $k = 1$ *is ergodic iff*

$$d_x(1, \lambda_1)d_y(0, \lambda_1, \lambda_2) - d_y(1, \lambda_1, \lambda_2)d_x(0, \lambda_1, \lambda_1) < 0 \tag{7}$$

with:

$$\begin{aligned}
d_x(1, \lambda_1, \lambda_2) &= \lambda_1 + \lambda_2(2 + d_x(-1, \lambda_1, \lambda_2)) + \mu(d_x(2, \lambda_1, \lambda_2) - 1) \\
d_x(0, \lambda_1, \lambda_2) &= \mu d_x(-1, \lambda_1, \lambda_2) + \lambda_1(1 + d_x(-1, \lambda_1, \lambda_2)) - \mu \\
d_y(1, \lambda_1, \lambda_2) &= -(\lambda_1 + \lambda_2 p(0| - 1, \lambda_1, \lambda_2) + \mu + \mu(1 - p(0| - 1, \lambda_2, \lambda_1))) \\
d_y(0, \lambda_1, \lambda_2) &= 1 - (\lambda_1 p(0| - 1, \lambda_1, \lambda_2) + \mu p(0| - 1, \lambda_1, \lambda_2))
\end{aligned}$$

where

$$\begin{aligned}
p(0| - 1, \lambda_1, \lambda_2) &= \frac{1 + e_1(\lambda_1, \lambda_2)\lambda_1 - \mu - e_1(\lambda_1, \lambda_2)\mu}{\mu - e_1(\lambda_1, \lambda_2)\mu} \\
d_x(-1, \lambda_1, \lambda_2) &= -\frac{e_2(\lambda_1, \lambda_2)}{1 - e_2(\lambda_1, \lambda_2)} \\
d_x(2, \lambda_1, \lambda_2) &= -\frac{e_2(\lambda_2)}{1 - e_2(\lambda_2)} + \frac{\lambda_2}{\mu} e_2(\lambda_2) + 1
\end{aligned}$$

and

$$\begin{aligned}
e_1(\lambda_1, \lambda_2) &= \frac{\mu - 1 - \sqrt{(1 - \mu)^2 + 4\mu\lambda_1}}{2\lambda_1} \\
e_2(\lambda_1, \lambda_2) &= \frac{\mu - 1 + \sqrt{(1 - \mu)^2 + 4\mu\lambda_1}}{2\lambda_1}.
\end{aligned}$$

Numerical evaluation of (7) shows that the line defined by the equation:

$$\lambda_1 + \lambda_2 = \mu + \mu/2 \tag{8}$$

is a good approximation of the upper bound of the ergodicity region. This is also illustrated at figure 3: the strategy is ergodic in the area bounded above by $*$'s; the

non ergodicity region can be split up in the area marked by $*$'s under the line $\lambda_1+\lambda_2 = \mu+\mu/2$ (i.e. the error induced by using approximation (8)) and the entire area above this line. For the sake of completeness we also pictured the area of ergodicity if no rerouting would be available; it is the convex polygon ODBE.

Further on, we also calculated and marked the region of ergodicity in case $k = 2$; the corresponding area of ergodicity is the union of OAC and the one marked by $*$'s. Numerical calculations confirm that as k gets larger, the ergodicity region converges to the convex polygon OABC.

We also investigated the influence of t for a fixed value of k $(= 2)$. The corresponding recursion relations are pretty hard to solve; therefore we resorted to a numerical recursive estimation method. The results are depicted in figure 4. For $\lambda_1, \lambda_2 < 0.36$ the region marked by $*$'s defines the difference between the ergodicity region when $t = 3/2$ and when $t = 2$. Since the approximations error where rather large for $\lambda_1 > 0.36$ or $\lambda_2 > 0.36$, , we did not show the outcomes. Nevertheless we have a firm belief that the region will be as indicated by the (gearceerd, hoe is dat in het Engels?) part since for those cases one of the arrival parameters is so small, that the algorithm will actually behave as a random rerouting schedule (see further).
As in the previous case, if t comes closer to 1, numerical calculations show that the ergodicity region converges to the convex polygon OFBG.

Finally, we compared our family of rerouting schemes with random rerouting: suppose e.g. in the asymetrical case, that queue 1 reroutes an arriving task with probability $\frac{\lambda_1-\mu+\epsilon}{\lambda_1}$. Since this thinned Poisson process is again Poisson with arrival parameter $\mu-\epsilon$, the first queue will be ergodic. Because of the independence assumptions, the arrival stream at the second queue will also be Poisson with parameter $\lambda_2+t(\lambda_1-\mu+\epsilon)$. Consequently the condition that this scheme is ergodic is equal to the previous one but the rerouting scheme based on differences will give rise to lower average waiting times and a better use of the system because it never reroutes a job (and does not cause overhead) when the queue where it arrived, is almost empty.

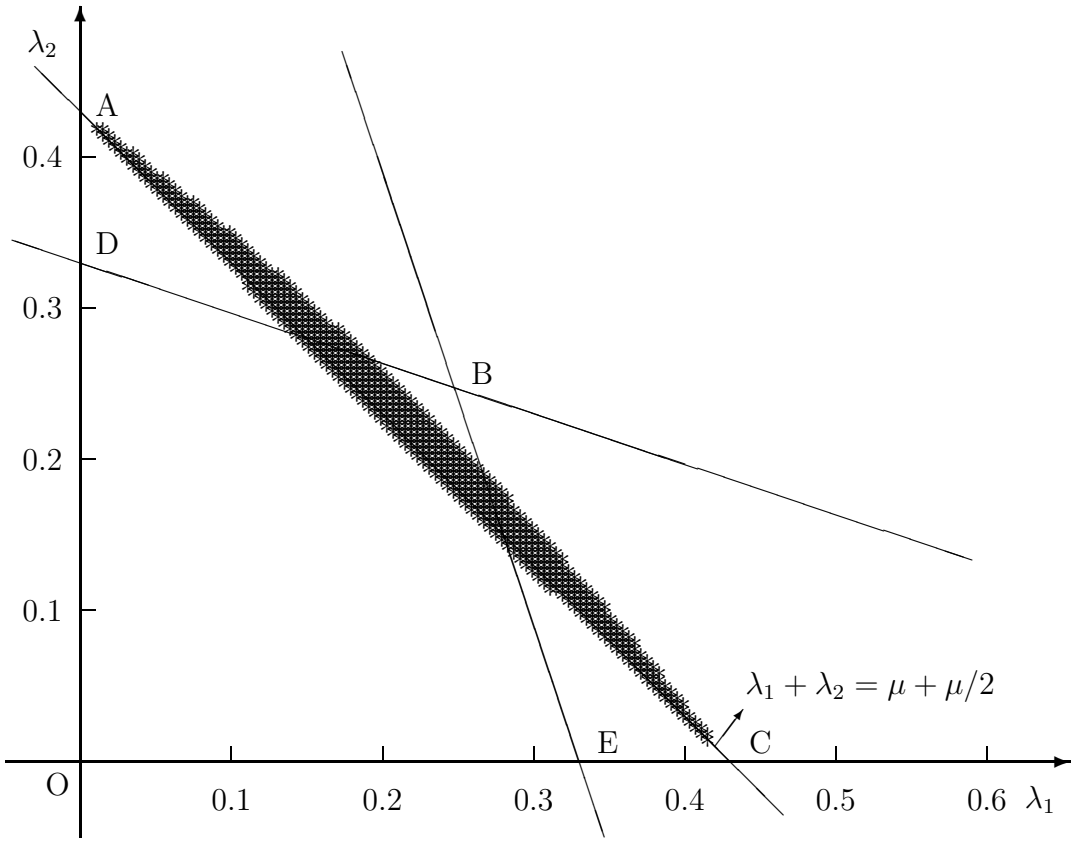

λ_2
A
0.4
D
0.3
B
0.2
0.1
$\lambda_1 + \lambda_2 = \mu + \mu/2$
E
C
O
0.1
0.2
0.3
0.4
0.5
0.6
λ_1

figure 3

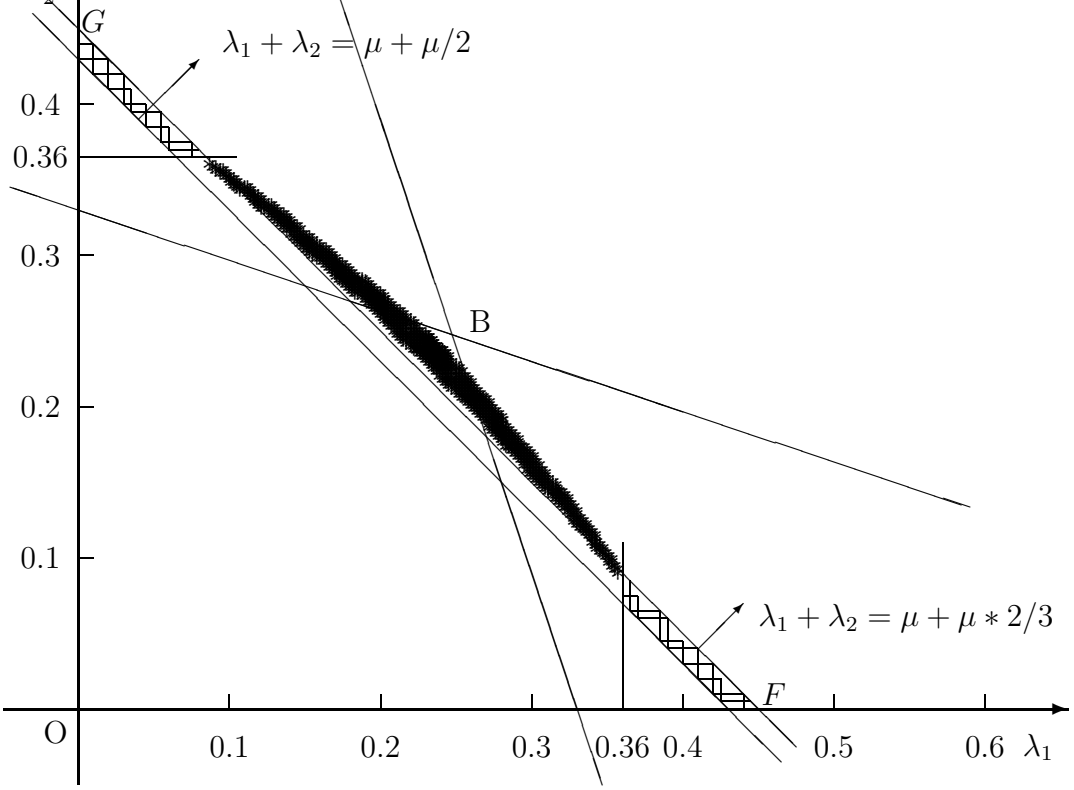
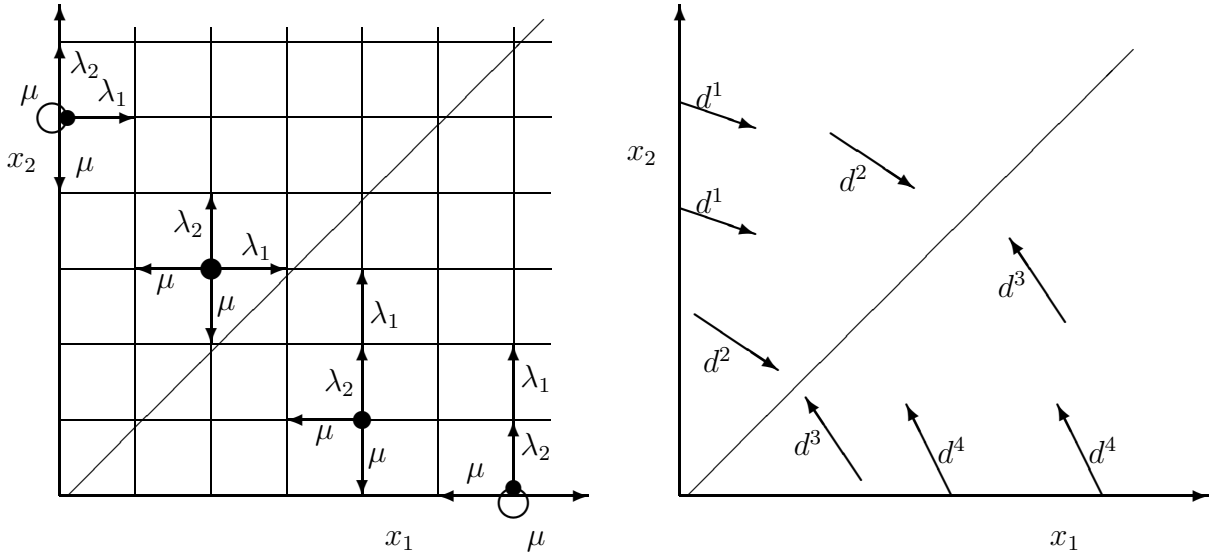


figure 4

3.3 Proof of Property 3.1, asymmetrical case

Proof:
Suppose first that $j = 2$.
The transition probabilities and the drift vectors look as in figure 5; d^1 resp. d^4 represents the drift vector on the boundary $x_1 = 0$ resp. $x_2 = 0$ while the drift vectors d^2 resp. d^3 represent the average drift in the region without, resp. with rerouting.



with
$d^1 = (-\mu, \lambda_1 + 2\lambda_2); d^2 = (\lambda_1 - \mu, \lambda_2 - \mu); d^3 = (-\mu, \lambda_2 + 2\lambda_1 - \mu); d^4 = (-\mu, \lambda_2 + 2\lambda_1);$

figure 5

In order to be able to apply Corollary 2.1, we show first that outside a compact region around the origin, the orientation of the m-steps driftvector at the border $d^{4(m)}$ and $d^{1(m)}$ becomes arbitrarily close to the orientation of the $d^{3(m)}$ resp. $d^{2(m)}$ for m sufficient large.

Due to the virual departures, it is possible to calculate both components of $\frac{d^{4(m)}}{m}$ separately:

1. The first component equals d_1^3;
2. The second component can be derived by looking at the projected process on the x_2-axis and by writing down explicitly the corresponding recursion relation of p_i^n, i.e., the probability of being in $(0, i)$ after n steps if one starts at $(0, 0)$. One gets:

$$\begin{aligned} p_1^n &= \mu p_1^{n-1} + \mu p_2^{n-1} + \lambda_2 p_0^{n-1} \\ p_i^n &= \mu p_i^{n-1} + \mu p_{i+1}^{n-1} + \lambda_2 p_{i-1}^{n-1} + \lambda_1 p_{i-2}^{n-1} \text{ if } i > 1. \end{aligned}$$

Consequently if we call $d_2^{(n)}$ the n-step drift at $(x, 0)$, we obtain:

$$d_2^{(n)} = d_2^{(n-1)} + d_2^3 + \mu p_0^{n-1}$$

and therefore the m-step drift in the x_2 direction equals:

$$d_2^{(m)} = (m-1)d_2^3 + d_2^4 + \mu \sum_{n=1}^{m-1} p_0^n.$$

Since $d_2^3 > 0$, we know that the projected process is transient. Therefore $\frac{1}{m}\sum_{n=1}^{m} p_0^n$ converges to zero. This implies that the orientation of $\frac{d^{4(m)}}{m}$ $(= d_2^{4(m)}/d_1^{4(m)})$, converges to the one of d^3 as $m \to \infty$. A similar proof holds for the relation between d^1 and d^2.

The above implies that in figure 5 outside a bounded region around the origin, the drift vectors d^1 and d^4 can be discarded and Corollary 2.1 can be applied.

Looking at this corollary, the necessary and sufficient conditions of ergodicity are that the angle between d^2 and the x_2 -axis should be less than the angle between the x_1-axis and d^3. Because of the rerouting schema, every vector has at least one negative component and therefore, the angle requirement is equivalent to (cfr. [9]):

$$(\lambda_1 - \mu)(2\lambda_1 + \lambda_2 - \mu) + (\lambda_2 - \mu)\mu < 0.$$

Making use of the fact that $\lambda_1 + \lambda_2 + \mu + \mu = 1$, one can show that the above is equivalent to:

$$\lambda_2 + 2(\lambda_1 - \mu) < \mu.$$

In case j has an arbitrary value different from 1, we can approximate j by a rational number a/b and use the above derivation where an arrival task generates now b tasks at his own queue and a tasks when it is rerouted to the other queue, and where a departure is equal to a jump of b steps.

□

3.4 Proof of Property 3.1, symmetrical case

This situation gives rise to 5 different drift vectors, one in the area $x_1 > x_2 > 0$, one in the symmetric area $x_2 > x_1 > 0$, one on the bisecting line (the only area where there is no rerouting in this case) and one on each boundery (that can be deleted as in the previous case).

With the remaining three driftvectors, we cannot apply Corollary 2.1, because it is not clear what we can conclude in case two drift vectors lie below and one drift vector lies above an equipotential line. In order to further reduce the number of drift vectors from 3 to 2 we aggregate now a random number of steps in each of the rerouting areas $x_1 > x_2 > 0$ and $x_2 > x_1 > 0$. To simplify the notation we change the coordinate basis of our state space to $(x, y) = (x_1, x_2 - x_1)$. The transition probabilities in this new coordinate system are shown in fig. 7.

The aggregation of the steps is defined as follows: suppose that the process is at time n in state $(i, 0)$ (with respect to the new basis). The possible transitions in the embedded discrete chain are to $(i-1, 1), (i, 1)$ and $(i+1, -1), (i, -1)$. In the last two cases the rerouting area is entered. The rerouting policy guarantees that the process will leave this area in a finite time, by crossing the axis at $(j, 0)$ or $(j, 1)$ with $j \leq i + 1$. We consider $(j, 0)$ or $(j, 1)$ as the next position at time $n+1$, on a new time axis where all the time instants when the state is in the region $x_1 > x_2$ are dropped.
We can apply the same procedure if the process is at time n in state $(i, 1)$ and enters the region $x_2 > x_1 + 1$ Finally we will obtain a process with a state space as shown in figure

6: there are now only two different drift vectors such that we can apply straightforward Corollary 2.1.

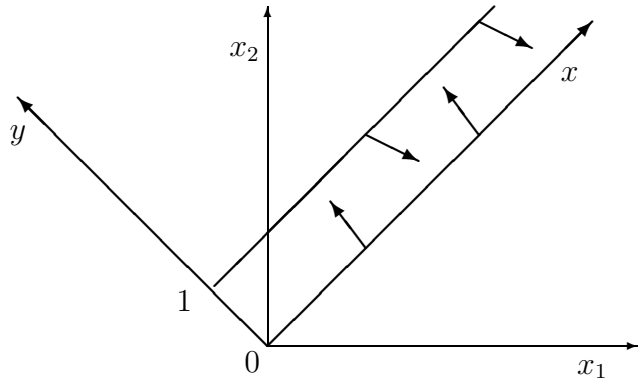


figure 6

We denote the two drift vectors for the new process as follows:

$$(d_x((i,0),\lambda_1,\lambda_2), d_y((i,0),\lambda_1,\lambda_2)): \text{ the drift vector at } (i,0)$$

$$(d_x((i,1),\lambda_1,\lambda_2), d_y((i,1),\lambda_1,\lambda_2)): \text{ the drift vector at } (i,1).$$

and for $k > 0$:

$p(0|(i,-k),\lambda_1,\lambda_2)$: the probability that the position (called *escape position*) where we leave the rerouting area $\{x_1 > x_2\}$ is on the line $y = 0$ and not on the line $y = 1$, if one starts at $(i,-k)$;

$p(1|(i,k),\lambda_1,\lambda_2)$: the probability that the position where we leave the rerouting area $\{x_2 > x_1 + 1\}$ is on the line $y = 1$ and not on the line $y = 0$, if one starts at $(i,-k)$;

M_k: $= i - j$, with j the x coordinate of the escape position if one starts at $(i,-k)$; It is the number of steps to the left in the original process before leaving the rerouting area;

$d_x((i,-k),\lambda_1,\lambda_2)$: $= -EM_k$.

If we look at figure 7, we obtain the following relations by conditioning each time on the next transition of the process and by making use of the symmetry relation $p(1|(i,k),\lambda_1,\lambda_2) = 1 - p(0|(i,-k+1),\lambda_2,\lambda_1)$:

$$\begin{aligned}
d_x((i,0),\lambda_1,\lambda_2) &= \mu d_x((i,-1),\lambda_1,\lambda_2) + \lambda_1(1 + d_x((i+1,-1),\lambda_1,\lambda_2)) - \mu \\
d_y((i,0),\lambda_1,\lambda_2) &= 1 - (\lambda_1 p(0|(i+1,-1),\lambda_1,\lambda_2) + \mu p(0|(i,-1),\lambda_1,\lambda_2)) \\
&\text{and} \\
d_x((i,1),\lambda_1,\lambda_2) &= \lambda_1 + \lambda_2(2 + d_x((i+2,-1),\lambda_1,\lambda_2))) + \mu(d_x((i-1,2),\lambda_1,\lambda_2) - 1), \\
d_y((i,1),\lambda_1,\lambda_2) &= -(\lambda_1 + \lambda_2 p(0|(i+2,-1),\lambda_1,\lambda_2) + \mu + \mu(1 - p(0|(i-1,-1),\lambda_2,\lambda_1))).
\end{aligned}$$

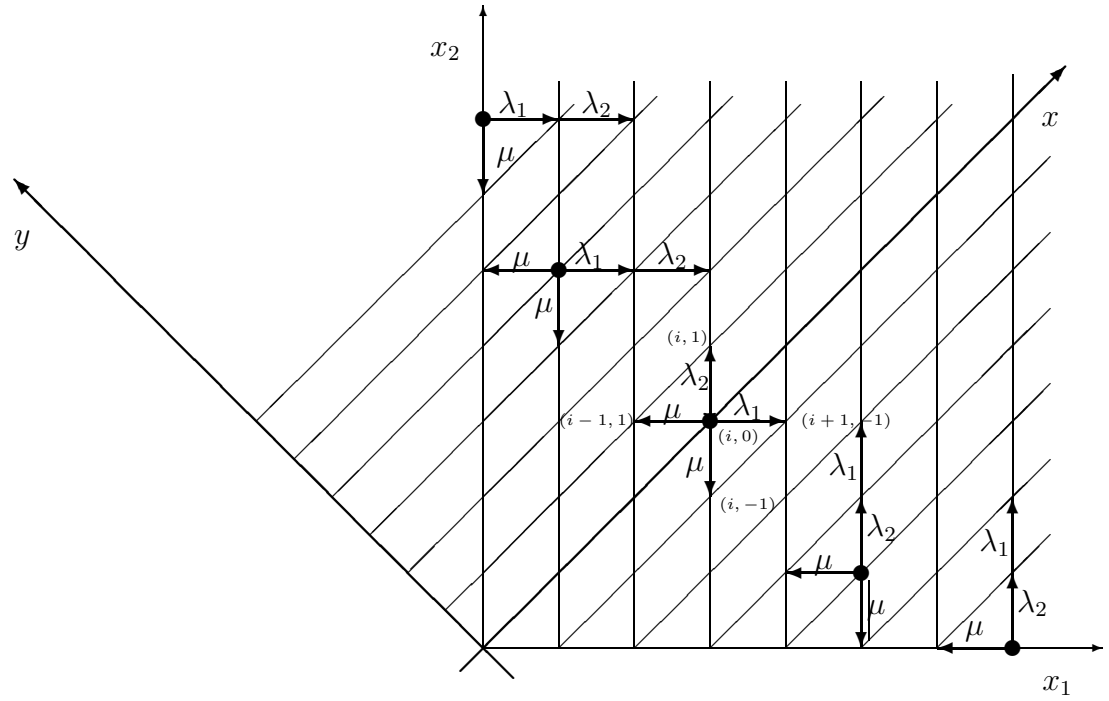


figure 7

3.4.1 Calculation of $(d_x((i,0),\lambda_1,\lambda_2), d_y((i,0),\lambda_1,\lambda_2)$

By writing down recursion relations for $p(0|(i+1,-1),\lambda_1,\lambda_2)$ and $d_x((i,-1),\lambda_1,\lambda_2)$ we can obtain explicit expressions for $\lim_{i\to\infty} p(0|(i+1,-1),\lambda_1,\lambda_2)$ and $\lim_{i\to\infty} d_x((i,-1),\lambda_1,\lambda_2)$.

• If we call $a_k = (i,-k)$ and take the limit $i \to \infty$ (once more, this is allowed because we are only interested in what happens outside a bounded region around the origin), we get the following recursion:

$$\begin{aligned}
p(0|a_1,\lambda_1,\lambda_2) &= \lambda_2 + \mu + \mu p(0|a_2,\lambda_1,\lambda_2) \\
p(0|a_2,\lambda_1,\lambda_2) &= \lambda_1 + (\lambda_2+\mu)p(0|a_1,\lambda_1,\lambda_2) + \mu p(0|a_3,\lambda_1,\lambda_2) \\
p(0|a_k,\lambda_1,\lambda_2) &= \lambda_1 p(0|a_{k-2},\lambda_1,\lambda_2) + (\lambda_2+\mu)p(0|a_{k-1},\lambda_1,\lambda_2) + \mu p(0|a_{k+1},\lambda_1,\lambda_2), k>2.
\end{aligned}$$

Using generating functions, one can deduce:

$$p(0|a_k,\lambda_1) = D - \frac{e}{\lambda_1} e_1(\lambda_1)^{-k-1} - \frac{f}{\lambda_1} e_2(\lambda_1)^{-k-1}, \tag{9}$$

with $e_1(\lambda_1)$ and $e_2(\lambda_1)$ the roots of the equation $\lambda_1 x^2 + (1-\mu)x - \mu$, and

$$\begin{aligned}
D &= \frac{\mu p(0|a_1,\lambda_1,\lambda_2) - 1 + \mu}{2\mu - 1 - \lambda_1} \\
f &= \frac{\mu D - \mu + \lambda_1 D e_2(\lambda_1,\lambda_2)}{e_2(\lambda_1,\lambda_2) - e_1(\lambda_1,\lambda_2)} \\
e &= \lambda_1 D - f,
\end{aligned}$$

where for the sake of simplicity, the dependency on λ_1 is deleted.

Writing $p(0|a_1,\lambda_1,\lambda_2)$ in terms of $p(0|a_k,\lambda_1,\lambda_2)$, Equation (9) becomes:

$$p(0|a_1,\lambda_1,\lambda_2) =$$

$$-(-(e_1(\lambda_1,\lambda_2)^{1+k}e_2(\lambda_1,\lambda_2)\lambda_1\mu) + e_1(\lambda_1,\lambda_2)e_2(\lambda_1,\lambda_2)^{1+k}\lambda_1\mu - e_1(\lambda_1,\lambda_2)^{2+k}e_2(\lambda_1,\lambda_2)^{1+k}\lambda_1\mu +$$
$$e_1(\lambda_1,\lambda_2)^{1+k}e_2(\lambda_1,\lambda_2)^{2+k}\lambda_1\mu - e_1(\lambda_1,\lambda_2)^{1+k}\mu^2 + e_2(\lambda_1,\lambda_2)^{1+k}\mu^2)^{-1} *$$
$$(e_1(\lambda_1,\lambda_2)^{1+k}e_2(\lambda_1,\lambda_2)\lambda_1 - e_1(\lambda_1,\lambda_2)e_2(\lambda_1,\lambda_2)^{1+k}\lambda_1 + e_1(\lambda_1,\lambda_2)^{2+k}e_2(\lambda_1,\lambda_2)^{1+k}\lambda_1 -$$
$$p(0|a_k,\lambda_1,\lambda_2)e_1(\lambda_1,\lambda_2)^{2+k}e_2(\lambda_1,\lambda_2)^{1+k}\lambda_1 - e_1(\lambda_1,\lambda_2)^{1+k}e_2(\lambda_1,\lambda_2)^{2+k}\lambda_1 + p(0|a_k,\lambda_1,\lambda_2)e_1(\lambda_1,\lambda_2)^{1+k}e_2(\lambda_1,\lambda_2)^{2+k}\lambda_1 -$$
$$p(0|a_k,\lambda_1,\lambda_2)e_1(\lambda_1,\lambda_2)^{2+k}e_2(\lambda_1,\lambda_2)^{1+k}\lambda_1^2 + p(0|a_k,\lambda_1,\lambda_2)e_1(\lambda_1,\lambda_2)^{1+k}e_2(\lambda_1,\lambda_2)^{2+k}\lambda_1^2 - e_1(\lambda_1,\lambda_2)^{1+k}\lambda_1\mu -$$
$$e_1(\lambda_1,\lambda_2)^{1+k}e_2(\lambda_1,\lambda_2)\lambda_1\mu + e_2(\lambda_1,\lambda_2)^{1+k}\lambda_1\mu +$$
$$e_1(\lambda_1,\lambda_2)e_2(\lambda_1,\lambda_2)^{1+k}\lambda_1\mu - e_1(\lambda_1,\lambda_2)^{2+k}e_2(\lambda_1,\lambda_2)^{1+k}\lambda_1\mu +$$
$$2p(0|a_k,\lambda_1,\lambda_2)e_1(\lambda_1,\lambda_2)^{2+k}e_2(\lambda_1,\lambda_2)^{1+k}\lambda_1\mu + e_1(\lambda_1,\lambda_2)^{1+k}e_2(\lambda_1,\lambda_2)^{2+k}\lambda_1\mu - 2p(0|a_k,\lambda_1,\lambda_2)e_1(\lambda_1,\lambda_2)^{1+k}e_2(\lambda_1,\lambda_2)^{2+k}\lambda_1\mu +$$
$$e_1(\lambda_1,\lambda_2)^{1+k}\mu^2 - e_2(\lambda_1,\lambda_2)^{1+k}\mu^2).$$

Divide the numerator and the denominator by $e_1(\lambda_1,\lambda_2)^k$ and take the limit $k\to\infty$. By using the fact that $\lim_{k\to\infty} p(0|a_k,\lambda_1,\lambda_2)$ is finite and $|e_1(\lambda_1,\lambda_2)| > 1 > |e_2(\lambda_1,\lambda_2)|$, one obtains:

$$\lim_{i\to\infty} p(0|(i,-1),\lambda_1,\lambda_2) = \frac{1+e_1(\lambda_1,\lambda_2)\lambda_1-\mu-e_1(\lambda_1,\lambda_2)\mu}{\mu-e_1(\lambda_1,\lambda_2)\mu}. \tag{10}$$

• In order to prove a similar result to (10) for $d_x((i,-1),\lambda_1,\lambda_2)$, we derive the following recursion relation (once more by conditioning on the next step in the process)

$$\begin{aligned}
d_x(a_1,\lambda_1,\lambda_2) &= \mu d_x(a_2,\lambda_1,\lambda_2)-\mu & (11)\\
d_x(a_2,\lambda_1,\lambda_2) &= \mu d_x(a_3,\lambda_1,\lambda_2)+\lambda_2 d_x(a_1,\lambda_1,\lambda_2)+\mu(d_x(a_1,\lambda_1,\lambda_2)-1) & (12)\\
d_x(a_k,\lambda_1,\lambda_2) &= \lambda_2 d_x(a_{k-1},\lambda_1,\lambda_2)+\lambda_1 d_x(a_{k-2},\lambda_1,\lambda_2)+\mu d_x(a_{k+1},\lambda_1,\lambda_2)+ & (13)\\
&\quad \mu(d_x(a_{k-1},\lambda_1,\lambda_2)-1).
\end{aligned}$$

Using generating functions, one gets:

$$d_x(a_k,\lambda_1,\lambda_2) + \frac{\mu a(k-1)-\mu bq-\mu bqe_1(\lambda_1,\lambda_2)^{-(k-1)}-\mu cr-\mu cse_2(\lambda_1,\lambda_2)^{-(k-1)}}{\lambda_1} =$$

$$\frac{-d_x(a_1,\lambda_1,\lambda_2)\mu}{\lambda_1}(a+be_1(\lambda_1,\lambda_2)^{-k}+ce_2(\lambda_1,\lambda_2)^{-k}), \tag{14}$$

with

$a=((e_1(\lambda_1,\lambda_2)-1)(e_2(\lambda_1,\lambda_2)-1))^{-1}$ $\quad b=((e_1(\lambda_1,\lambda_2)-e_2(\lambda_1,\lambda_2))(e_1(\lambda_1,\lambda_2)-1))^{-1}$
$c=((e_2(\lambda_1,\lambda_2)-e_1(\lambda_1,\lambda_2))(e_2(\lambda_1,\lambda_2)-1))^{-1}$ $\quad p=(1-e_1(\lambda_1,\lambda_2))^{-1}$
$q=(e_1(\lambda_1,\lambda_2)-1)^{-1}$ $\quad r=(1-e_2(\lambda_1,\lambda_2))^{-1}$
$s=(e_2(\lambda_1,\lambda_2)-1)^{-1}$

As in the first part , we multiply both parts in (14) with $e_2(\lambda_1,\lambda_2)^k$, take the limit for $k\to\infty$, and get:

$$\lim_{i\to\infty} d_x((i,-1),\lambda_1,\lambda_2) = -\frac{e_2(\lambda_1,\lambda_2)}{1-e_2(\lambda_1,\lambda_2)}, \tag{15}$$

under the condition that we can show that $\lim_{k\to\infty} d_x(a_k, \lambda_1, \lambda_2) e_2^k(\lambda_1, \lambda_2) = 0$. It suffices to show that $d_x(a_k, \lambda_1)$ $(= -EM_k)$ goes to infinity at most with a linear rate in k because $|e_2^k(\lambda_1, \lambda_2)| < 1$.
Since the variable M_k can also be defined as the number of steps to the left in the original process and will be smaller than the total number of steps, the variable N_k, defined as the number of steps of the process of figure 8 before reaching the origin when one starts in k, is an upper bound for M_k. Since $N_k \leq kN_1$ and since EN_1 is finite because of the ergodicity of the process, we get the desired result.

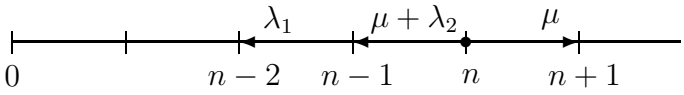


figure 8

3.4.2 Calculation of $(d_x((i,1), \lambda_1, \lambda_2), d_y((i,1), \lambda_1, \lambda_2)$

Given (10) and (15), we only have to calculate $d_x((i,2), \lambda_1, \lambda_2)$. The limit for $i \to \infty$ can be calculated by means of a recursion relation very similar to (11), (12) and (13) and by making use of a symmetry argument. We get:

$$\lim_{i\to\infty} d_x((i-1,2), \lambda_1, \lambda_2) = -\frac{e_2(\lambda_2)}{1-e_2(\lambda_2)} + \frac{\lambda_2}{\mu} e_2(\lambda_2) + 1.$$

3.4.3 All together

Since we have a random walk in the state space $Z^+ \times \{0,1\}$ represented in figure 6 in the original coordinates (x_1, x_2), with only two different drift vectors, we can proceed as in Corollary 2.1 if we can argue that the probabilities of large jumps (or equivalently, the tail of M_1) decrease exponentially fast to zero but this follows from [11].

Combining all together, the necessary and sufficient condition of ergodicity becomes, for $i \to \infty$:

$$d_x((i,1),\lambda_1,\lambda_2)d_y((i,0),\lambda_1,\lambda_2) - d_y((i,1),\lambda_1,\lambda_2)d_x((i,0),\lambda_1,\lambda_2) < 0. \qquad (16)$$

□

4 Rerouting based on Thresholds

Depending on the amount of information the queues know about each other and on the way they use it, different rerouting schedules can be studied. Fortunately, it turns out that the above methodology can be used to study many of them. We illustrate this with the so called rerouting policy based on a fixed threshold. A queue will reroute a task if it contains at least c_2 jobs and if there are in the other queue less then c_1 jobs waiting. We obtain:

Property 4.1
Rerouting based on a threshold (asymmetrical and symmetrical case) with $\lambda_1 > \mu > \lambda_2$ and $t = 2$ is ergodic iff

$$(\lambda_1 - \mu)d_y(c_1, \lambda_1, \lambda_2) - (\lambda_2 - \mu)d_x(c_1, \lambda_1, \lambda_2) < 0 \tag{17}$$

with

$$\begin{aligned}
d_x(c_1, \lambda_1, \lambda_2) &= \lambda_1 - \mu + \\
&\frac{\mu(-\mu\lambda_1 + \lambda_1(-\mu a(c_1 - 3) + \mu bqd_1^{3-c_1} + \mu csd_2^{3-c_1}))}{-(\lambda_1 + \lambda_2)\mu(bd_1^{-c_1} + cd_2^{-c_1}) + \lambda_1\mu(bd_1^{2-c_1} + cd_2^{2-c_1}) + \lambda_2\mu(bd_1^{1-c_1} + cd_2^{1-c_1})} + \\
&\frac{\mu\lambda_2(-\mu a(c_1 - 2) + \mu bqd_1^{2-c_1} + \mu csd_2^{2-c_1})}{-(\lambda_1 + \lambda_2)\mu(bd_1^{-c_1} + cd_2^{-c_1}) + \lambda_1\mu(bd_1^{2-c_1} + cd_2^{2-c_1}) + \lambda_2\mu(bd_1^{1-c_1} + cd_2^{1-c_1})} + \\
&\frac{\mu(\lambda_1 + \lambda_2)(\mu a(c_1 - 1) - \mu bqd_1^{1-c_1} - \mu csd_2^{1-c_1})}{-(\lambda_1 + \lambda_2)\mu(bd_1^{-c_1} + cd_2^{-c_1}) + \lambda_1\mu(bd_1^{2-c_1} + cd_2^{2-c_1}) + \lambda_2\mu(bd_1^{1-c_1} + cd_2^{1-c_1})}, \\
d_y(c_1, \lambda_1, \lambda_2) &= \lambda_2 + \mu - \\
&\frac{\mu(\lambda_1\mu + \lambda_2(\mu - 1))(-bd_1^{1-c_1} - cd_2^{1-c_1}) + (\lambda_2\mu - (\lambda_1 + \lambda_2)(\mu - 1))(-bd_1^{-c_1} - cd_2^{-c_1})}{(\lambda_1 + \lambda_2)\mu(-bd_1^{-c_1} - cd_2^{-c_1}) - \lambda_1\mu(-bd_1^{2-c_1} - cd_2^{2-c_1}) - \lambda_2\mu(-bd_1^{1-c_1} - cd_2^{1-c_1})} - \\
&\frac{-\mu(\lambda_1 + \lambda_2)\mu(-bd_1^{-1-c_1} - cd_2^{-1-c_1}) + \lambda_1(\mu - 1)(-bd_1^{2-c_1} - cd_2^{2-c_1})}{(\lambda_1 + \lambda_2)\mu(-bd_1^{-c_1} - cd_2^{-c_1}) - \lambda_1\mu(-bd_1^{2-c_1} - cd_2^{2-c_1}) - \lambda_2\mu(-bd_1^{1-c_1} - cd_2^{1-c_1})}
\end{aligned}$$

with

$a = ((d_1 - 1)(d_2 - 1))^{-1}$ $b = ((d_1 - d_2)(d_1 - 1))^{-1}$ $c = ((d_2 - d_1)(d_2 - 1))^{-1}$
$p = (1 - d_1)^{-1}$ $q = (d_1 - 1)^{-1}$ $r = (1 - d_2)^{-1}$
$s = (d_2 - 1)^{-1}$

and

$$\begin{aligned}
d_1 &= \frac{-(\lambda_1 + \lambda_2) - \sqrt{(\lambda_1 + \lambda_2)^2 + 4\mu\lambda_1}}{2\lambda_1} \\
d_2 &= \frac{-(\lambda_1 + \lambda_2) + \sqrt{(\lambda_1 + \lambda_2)^2 + 4\mu\lambda_1}}{2\lambda_1}.
\end{aligned}$$

We calculated the area of ergodicity as defined by (17) for the case $c_1 = 2$. The result is given in figure 9. Since we are looking at the case $\lambda_1 > \mu > \lambda_2$ we are only interested in the region under the line DB and above BE. The corresponding ergodicity region is compared with the one for the rerouting policy based on differences in queue length:

in area 1 (bounded above by $*$'s) both policies are ergodic. In the area marked by $*$'s (which has as upper bound the line $\lambda_2 + 2(\lambda_1 - \mu) = \mu$) only the policy based on differences (as in section 3, limited to the case $k = 1$) is ergodic. Above the line $\lambda_2 + 2(\lambda_1 - \mu) = \mu$ none of them is ergodic. Numerical calculations show that in the limit (for large c_1) the two conditions become equivalent to each other.

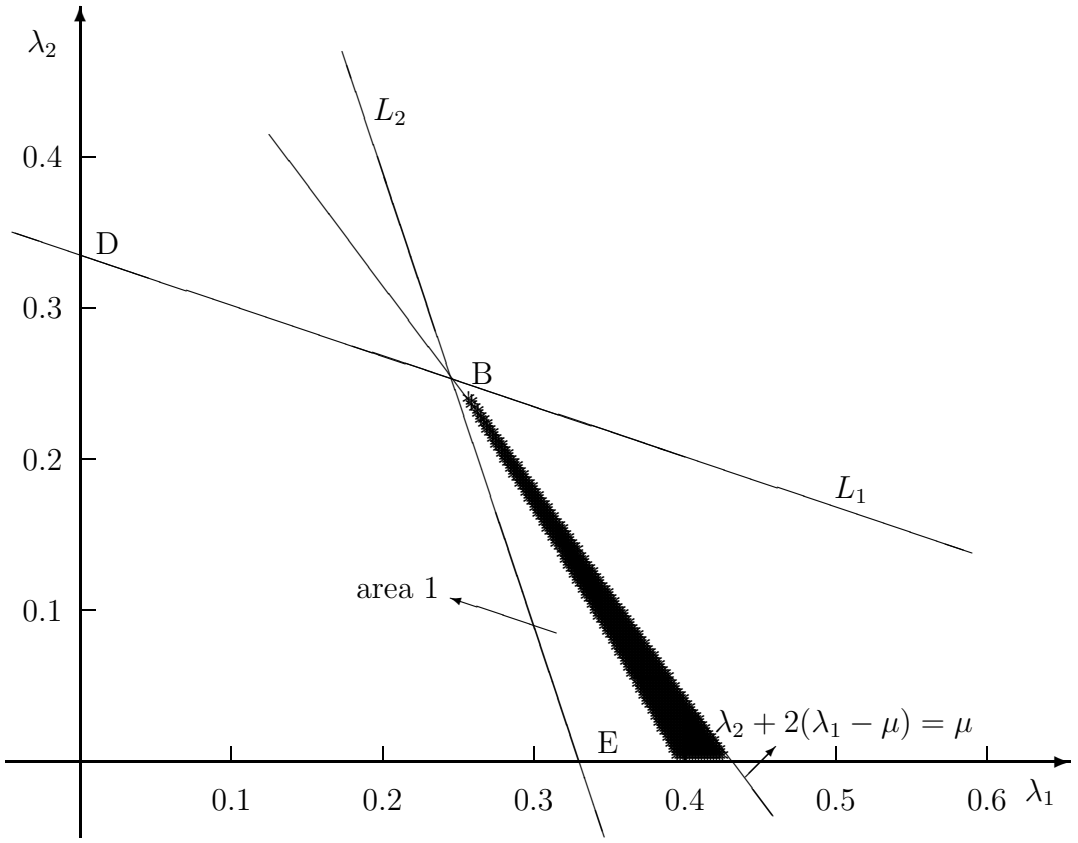


figure 9

5 Conclusions

In the above rerouting schemes it was assumed that each queue knew at every time instant the queue length of the other one. This is not always a reasonable assumption because one often resorts to estimations of the queue length.

Under the unrealistic assumption that rerouting errors (i.e., rerouting a job when it was actually not permitted by the rerouting strategy) are independent one can get an idea of the robustness of the strategies.

Consider a rerouting policy based on differences and assume that queue 1 makes with probability p the error that at the diagonal a task is wrongly rerouted (the easiest case to consider). This means that (take $i \to \infty$) :

$$
\begin{aligned}
&d_y((i,0),\lambda_1,\lambda_2) \\
&\quad = \mu(1-p(0|(i,-1),\lambda_1,\lambda_2)) + (1-p)\lambda_1(1-p(0|(i+1,-1),\lambda_1,\lambda_2) + \mu + \lambda_2 + \\
&\qquad p\lambda_1 p(0|(i,-1),\lambda_2) \\
&d_x((i,0),\lambda_1,\lambda_2) \\
&\quad = \mu d_x((i,-1),\lambda_1,\lambda_2) + (1-p)\lambda_1(1+d_x((i+1,-1),\lambda_1,\lambda_2)) - \mu + p\lambda_1 d_x((i,2),\lambda_1,\lambda_2).
\end{aligned}
$$

In order to analyze a model that takes fully into account that every queue has at time

t only an estimation a_t of the number of jobs in the other queue and that the exact queue length is included as some additional information in a rerouted job, many new problems have to be tackled. The dimensionality of the state space increases and the walk becomes more and more inhomogeneous since large jumps are possible when, e.g., a rerouted job arrives and the current estimator a_t is set equal to the obtained exact value. By using quadratic Lyapunov functions, one can obtain conditions on the parameters that guarantee ergodicity but it seems pretty hard to get any insight about how far they are away from the necessary and sufficient conditions.

Acknowledgments

We would like to thank T.S. Turova for some useful comments.

References

[1] R.K. Boel and J.H. van Schuppen, Distributed routing for load balancing. In *Proceedings of the IEEE, 1989*.

[2] W.H. Cameron, J. Regnier, P. Galloy and A.M. Savoie, Dynamic routing for intercity telephone networks. In: *Proceedings International Teletraffic Congress ITC-11* (Kyoto, 1983).

[3] G. Fayolle, I.A. Ignatyuk, V.A. Malyshev and M.V. Mensikov, Random walks in two-dimensional complexes, Queueing Systems 9 (1991) 269-300.

[4] F.G. Foster, On the stochastic matrices associated to certain queueing processes, Annals of Mathematical Statistics 24 (1955) 355-360.

[5] B. Hajek, Hitting time and occupation time bounds implied by drift analysis with applications, Advances of Applied Probability 14 (1982) 502-525.

[6] I. Ignatyuk and V.A. Malyshev, Classification of random walks in $\mathcal{Z}_+^4$, Selecta Mathematica 12 (1993) 129-194.

[7] R. Krupp, Stability of alternate routed networks. In: *IEEE International Communication Conference* (Baltimore, 1982).

[8] H. Kushner, *Introduction to Stochastic Control* (Holt, Rinehart and Winston, 1971).

[9] V.A. Malyshev, Classification of two-dimensional positive random walks and almost linear semi-martingales, Soviet Math. Dokl. 13 (1972) 136-139.

[10] V.A. Malyshev and M.V. Mensikov, Ergodicity, Continuitity and analyticity of countable markov chains, Transactions of the Moscow Mathematical Society 39 (1981) 1-48.

[11] S.P. Meyn and R.L. Tweedie, *Markov Chains and Stochastic Stability* (Springer Verlag, 1993).

[12] T.S. Turova, Analysis of a stochastic neural model with inhibitory connections. Working Paper, Departement Wiskunde, K.U.Leuven, 1993.

[13] Y.T. Wang and R.J.T. Morris, Load sharing in distributed systems, IEEE Transactions on Computing 34 (1985) 204-217.

[14] J.G. Dai, On the Positive Harris Recurrence for Multiclass Queueing Networks: a unified approach via fluid limt models, Annals Applied Probability 5 (1995) 49-77.

[15] J.G.Dai and S.P.Weyn, Stability and convergence of moments for Multiclass queueing networks via fluid limit models, to appear in Transactions on Automatic Control.

[16] L. Georgiadis, W. Szpankowski and L. Tassiulas, Stability analysis of quota allocation access protocols in ring networks with spatial reuse, (????) ????.