On Deadlocking in Queueing Networks

Geraint Ian Palmer, Paul Harper, Vincent Knight

September 30, 2015

Abstract

In this paper a deadlock in open restricted queueing networks is investigated. A method to detect when deadlock occurs in simulations of these networks in presented, using a state digraph. Markov models of five deadlocking queueing networks are given, and their properties on expected time to deadlock explored. These networks are a one node single server, one node multi server, two node single server without loops, two node multi server without loops, and two node single server with loops. These properties are compared to results obtained using the simulation model. Finally a bound on the time to deadlock of the two node single server with loops network is derived.

1 Introduction

Restricted open queueing networks that can experience deadlock are under-discussed in the literature. This paper addresses this by investigating deadlock properties in these queueing networks. A method of detecting deadlock in discrete event simulations of queueing networks is presented, and markovian models of five open restricted queueing networks are analysed, and the properties of their time to deadlock investigated. Finally a bound on the time to deadlock is given for one of these queueing networks.

Central to the study of deadlock in restricted queueing networks is the concept of blocking. Given two queues in tandem such that customers leaving the first service station enter the second, and the second queue has limited queueing capacity. If the second queue is full, and a customer finishes service at the first queue, that customer cannot join the next queue due to lack of capacity. This customer remains with the server, blocking other customers from beginning service with that server, until space becomes available at the second queue. This is referred to as blocking.

Throughout this paper service centers will be referred to as nodes, and an open unrestricted queueing network will use the following notation for the ith node:

- Λ_i denotes the external arrival rate.
- μ_i denotes the service rate.
- c_i denotes the number of parallel servers.
- n_i denotes the queueing capacity.

- r_{ij} denotes the routing probability from node i to node j upon completion of service at node i.

Exponential service times and Poisson arrivals are assumed.

For the purposes of this paper, deadlock is defined as follows.

Definition 1. *When a simulation is in a situation where at least one service station, despite having arrivals, ceases to begin or finish any more services due to recursive upstream blocking, the system is said to be in deadlock.*

Figure 1 shows a three node queueing network in a deadlocked state. The customers occupying servers B_1 and B_2 are blocked from entering the top node, while the customer occupying server A_1 is blocked from entering the middle node. Due to mutual blocking, these customers are preventing any more natural movement in these two nodes. Note however that on part of the network need be in deadlock, as the bottom node is free to continue services as normal.

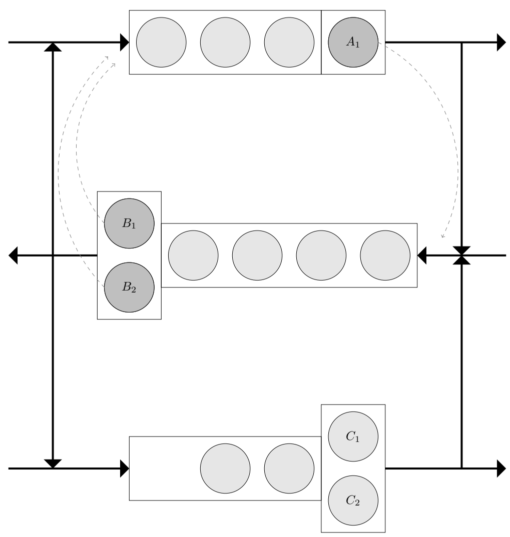


Figure 1: A three node queueing network in deadlock.

This paper is structured as follows: Section 2 discusses the exsisting literature on deadlock and deadlock strategies. Section 3 presents a method of detecting deadlock in discrete event simulations of queueing networks. Section 4 briefly describes the discrete event simulation model used to obtain the results in this paper. Section 5 presents Markov models of five deadlocking queueing networks, derives expected time to deadlock, and compares with resutls obtain through the simulation model. Finally Section 6 derives a bound for the expected time to deadlock for one of the queueing networks discussed.

2 Literature Review

Restricted queueing networks that exhibit blocking are well discussed in the literature [1, 2, 8, 10, 11, 13, 16]. Discussion on restricted queueing networks with feedback loops, that may exhibit deadlock, are sparse however.

General deadlock situations that are not specific to queueing networks are discussed in [5]. Conditions for this type of deadlock, also referred to as deadly embraces, to potentially occur are given:

- Mutual exclusion: Tasks have exclusive control over resources.
- Wait for: Tasks do not release resources while waiting for other resources.
- No preemption: Resources cannot be removed until they have been used to completion.
- Circular wait: A circular chain of tasks exists, where each task requests a resource from another task in the chain.

In open restricted queueing networks the mutual exclusion condition is satisfied as customers cannot share servers; the wait for condition is satisfied due to the blocking rules defined previously; the no preemption condition is satified in networks that have no or non-preemptive priority (this report will only look at networks with no priority); and the circular wait condition is satisfied if the queueing network contains a cycle where all nodes have limited queueing capacity, that is feedback loops.

In general there are three strategies for dealing with deadlock [7, 9]:

- Prevention, in which the system cannot possibly deadlock in the first place.
- Avoidance, in which decisions are made as time unfolds to avoid reaching deadlock.
- Detection and recovery.

2.1 Deadlock Prevention

Deadlock prevention has been discussed in queueing networks. For closed networks of K customers with only one class of customer, [12] proves the following condition to ensures no deadlock: for each minimum cycle C, $K < \sum_{j \in C} B_j$, the total number of customers cannot exceed the total queueing capacity of each minimum subcycle of the network. The paper also presents algorithms for finding the minimum queueing space required to ensure deadlock never occurs, for closed cactus networks, where no two cycles have more than one node in common. This result is extended to multiple classes of customer in [14], with more restrictions such as single servers and each class having the same service time distribution. Here a integer linear program is formulated to find the minimum queueing space assignment that prevents deadlock. The literature does not discuss deadlock properties in open restricted queueing networks.

2.2 Deadlock Avoidance

There are algorithms discussed in the literature for the dynamic avoidance of deadlock. In the Banker's Algorithm [6, 9], unsafe states, those that will lead to deadlock, are avoided by ensuring actions leading to these states are not carried out.

2.3 Deadlock Detection & Recovery

General deadlock detection in systems unspecific to queueing networks are discussed in [5]. A popular method of detecting general deadlock is the use of wait-for graphs, state-graphs and their variants [3, 4, 5, 7]. These wait-for graphs, keep track of all circular wait relations between tasks.

In [5] dynamic state-graphs are defined with resources as vertices and requests as edges. For scenarios where there is only one type of each resource, deadlock arises if and only if the state-graph contains a cylce. In [4] the vertices and edges of the state graph are given labels in relation to a reference node. Using these labels *simple bounded circuits* are defined whose existence within the state graph is sufficient to detect deadlock.

3 Detecting Deadlock

In order to detect when deadlock has occured in a queueing network simulation, the state digraph is used, defined below.

Definition 2. *The state digraph $D(t)$ of a queueing network defines that network's state at any time t. Vertices of the state digraph correspond to servers of the network. A directed edge denotes a blockage relationship in the following manner: if a customer at the kth server of node i is blocked from entering node j, then there are directed edges from the vertex corresponding to node i's kth server to every vertex corresponding to the servers of node j.*

To illustrate this concept Figure 2 and Figure 3 show a three node network in and out of deadlock, and the corresponding state digraph in each case.

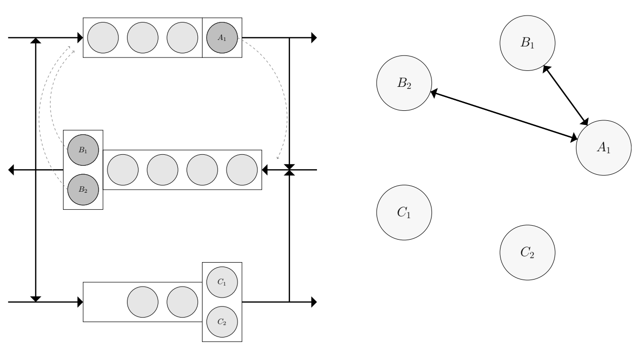


Figure 2: A three node queueing network in deadlock, with state digraph.

Consider one weakly connected component $G(t)$ of $D(t)$. Consider the node $X_a \in G(t)$. Some observations:

- Consider the node $X_a \in G(t)$. If X_a is unoccupied, then X_a has no incident edges.
- Consider the case when X_a is occupied by individual a, whose next destination is node j. Then X_a's direct successors are the servers occupied by individuals who are blocked or in service at node j.

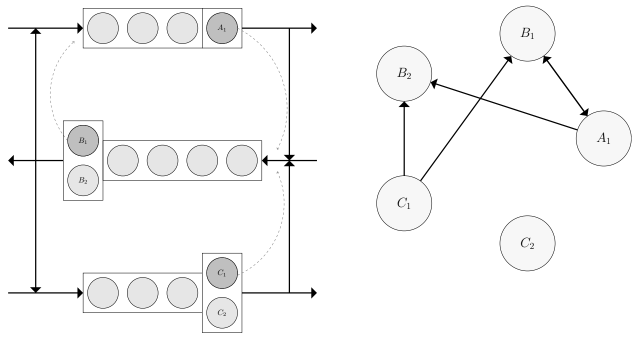


Figure 3: A three node queueing network not in deadlock, with state digraph.

- It can interpreted that all X_a's descendants are the servers whose occupants are directly or indirectly blocking a, and interpret all X_a's ancestors as those servers whose individuals who are being blocked directly or indirectly by a.
- Note that the only possibilities for $\deg^{\text{out}}(X_a)$ are 0 or c_j. If $\deg^{\text{out}}(X_a) = c_j$ then a is blocked by all its direct successors. The only other situation is that a is not blocked, and $X_a \in G(t)$ because a is in service at X_a and blocking other individuals, in which case $\deg^{\text{out}}(X_a) = 0$.
- It is clear that if all of X_a's descendants are occupied by blocked individuals, then the system is deadlocked at time t.
- By definition all of X_a's ancestors are occupied by blocked individuals.

The following results detect deadlock for open restricted queueing networks.

Theorem 1. *A deadlocked state arises at time t if and only if $D(t)$ contains a knot.*

Proof. Consider one weakly connected component $G(t)$ of $D(t)$ at time t. All vertices of $G(t)$ are either descendants of another vertex and so are occupied by an individual who is blocking someone; or are ancestors of another vertex, and so are occupied by someone who is blocked.

Assume that $G(t)$ contains a vertex X such that $\deg^{\text{out}}(X) = 0$, and there is a path from every other non-sink vertex to X. This implies that X's occupant is not blocked and is a descendant of another vertex. Therefore Q is not deadlocked as there does not exist a vertex whose descendants are all blocked.

Now assume that we have deadlock. For a vertex X who is deadlocked, all descendants of X are are occupied by individuals who are blocked, and so must have out-degrees greater than 0. And so there is no path from X to a vertex with out-degree of 0.

□

Proposition 1. *For queueing networks:*

- *with one node*

- *with two nodes, each with two or fewer parallel servers*
- *with a finite amount of nodes, each with a single server*

a deadlocked state arises if and only if there exists a weakly connected component without a sink node.

Proof. Consider a one node queueing network.

If there is deadlock, then all servers are occupied by blocked individuals, and so all servers have an out-edge.

Consider a two node queueing network, each node with 2 or fewer parallel servers.

If both nodes are involved in the deadlock, so there is a customer in node 1 blocked from entering node 2, and a customer from node 2 blocked from entering node 1, then all servers in node 1 and node 2 in $D(t)$ will have out edges as they are occupied by a blocked individual. The servers of node 1 and 2 consist of the entirety of $D(t)$, and so there is no sink nodes.

Now consider the case when only one node is involved in the deadlock. Without loss of generality, let's say that node 1 is in deadlock with itself, then the servers of node 1 have out-edges. For the servers of node 2 to be part of that weakly connected component, there either needs to be an edge from a server in node 1 to a server in node 2, or and edge from a server in node 2 to a server in node 1. An edge from a server in node 1 to a server in node 2 implies that a customer from node 1 is blocked from entering node 2, and so node 1 is not in deadlock with itself. An edge from a server in node 2 to a server in node 1 implies that a customer in node 2 is blocked from entering node 1. In this case one server in node 2 has an out-edge. Now either the other server of node two is still in server, and so isn't part of that weakly connected component, or the other server's customer is blocked and so has an out edge.

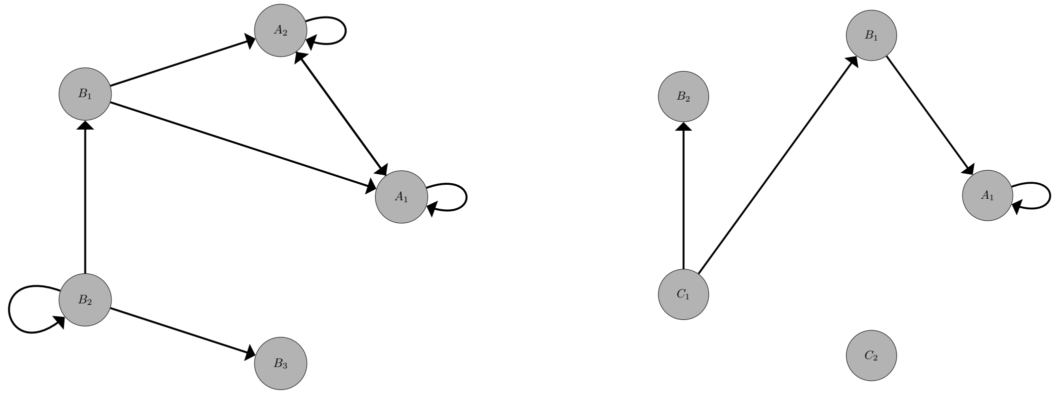
For the case of a queueing network with two nodes and more than 2 servers at node 2, consider the following counter-example:

Node A has two parallel server, node B has three parallel sevrers. Begin with all servers occupied by customers in service and full queues. The customer at server A_1 is blocked to node A. The customer at server B_1 is blocked to node A. The customer at server B_2 is blocked to node B. The customer at server A_2 is blockec to node A. The resulting state digraph in Figure 4a has a weakly connected component with a sink.

Consider a queueing network with N nodes, each with a single server.

If $1 \leq n \leq N$ nodes are involved in the deadlock, then each server in those n nodes has a blocked customer, and so has an out-edge. Of the other nodes, they can only be in the same weakly connected component if either they contain a indiviual blocked by those in deadlock, in which case they will have an out edge; or they contain a blocked individual blocked by those directly or indirectly blocked by those in deadlock, in which case they will have an out edge; or they are blocking someone who is blocked directly or indirectly by those in deadlock. However this last case cannot happen, as every node is single server each person can only be blocked by one other individual at a time.

For the case of a queueing network with more than two nodes with multiple servers, the following counter-example proves the claim:



(a) State Digraph of Counter-Example 1.

(b) State Digraph of Counter-Example 2.

Figure 4: Counter examples.

Node A has one parallel server, node B has two parallel servers, and node C has three parallel servers. Begin with all servers occupied by customers in service and full queues. The customer at server B_1 is blocked from entering node A. Then the customer at server C_1 is blocked from entering node B. Then the customer at server A_1 is blocked from entering node A. The resulting state digraph in Figure 4b has a weakly connected component with a sink.

□

4 Simulation Model

The simulation model used to run a discrete event simulation of the open restricted queueing network, detect deadlock and stop the simulation at this point. The time the simulation ran for until deadlock is recorded.

5 Markovian Models of Deadlocking Queueing Networks

In this section markov models are build for the following deadlocking queueing networks

- Open one node, single server restricted queueing network with feedback loop.
- Open two node, single server restricted queueing network with routes between nodes.
- Open two node, single server restricted queueing network with routes between nodes and feedback loops.
- Open one node, multi server restricted queueing network with feedback loop.
- Open two node, mutli server restricted queueing network with routes between nodes.

In general a continuous Markov chain model of a deadlocking queueing network is a set of states $s \in S$ and the transition rates between these states q_{s_1,s_2}. Each state s uniquely defines a configuration of customers around the queueing network. Deadlocked states are also present, either denoted by that specific configuration of customers, or by negative numbers, for example -1. Any deadlocked states cannot transition to any other state, and so is an absorbing state of the Markov chain. Therefore any queueing network that can experience deadlock is guaranteed to experience deadlock, as absorbing Markov chains are guaranteed to enter one of its absorbing states.

The expected time until deadlock is reached is equivalent to the expected time to absorbtion of the Markov chain, which can easily be found [15]. The canonical form of an absorbing Markov chain is

$$P = \begin{pmatrix} T & U \\ 0 & I \end{pmatrix}$$

where I is an identity matrix.

Now the expected number of time steps until absorption starting from state i is the ith element of the vector

$$(I - T)^{-1} e \tag{1}$$

where e is a vector of 1s.

Therefore by discretising the continuous Markov chain and ensuring the correct order of states, the number expected of time steps to absorption, or deadlock can be found.

When there are more than one deadlock state, there is more than one absorbing state in the Markov chain. The probabilities of which absorbing state a Markov chain will reach can be found. Now the $(i, j)^{\text{th}}$ element of the vector

$$(I - T)^{-1} U \tag{2}$$

corresponds to the probability of reaching absorbing state j from transient state i.

5.1 One Node Single Server

Consider the open one node single server restricted queueing network with feedback loop shown in Figure 5. This shows an $M/M/1$ queue with room for n customers to queue at any one time, customers arrive at a rate of Λ and served at a rate μ. Once a customer has finished service they rejoin the queue with probability r_{11}, and so exit the system with probability $1 - r_{11}$.

Let this system be denoted by Ω_1 with parameter set (Λ, μ, n, r_{11}), and the time to deadlock of this system be denoted by ω_1.

State space:

$$S = \{i \in \mathbb{N} \mid 0 \leq i \leq n + 1\} \cup \{-1\}$$

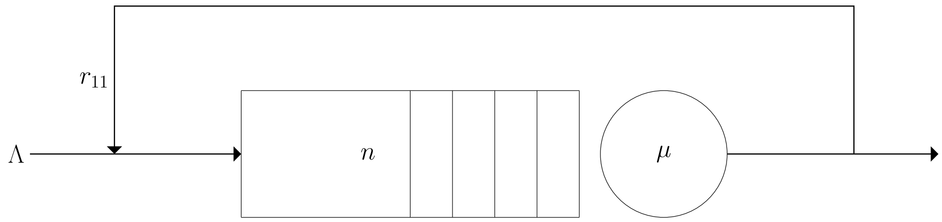


Figure 5: An open one node single server restricted queueing network.

where i denotes the number of individuals in service or waiting, and -1 denotes the deadlocked state.

If we define $\delta = i_2 - i_1$ for all $i_k \geq 0$ then the transitions are given by:

$$q_{i_1,i_2} = \begin{cases} \left.\begin{array}{ll} \Lambda & \text{if } i < n+1 \\ 0 & \text{otherwise} \end{array}\right\} & \text{if } \delta = 1 \\ (1 - r_{11})\mu & \text{if } \delta = -1 \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

$$q_{i,-1} = \begin{cases} r_{11}\mu & \text{if } i = n+1 \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

and

$$q_{-1,i} = 0 \tag{5}$$

The Markov chain is shown in Figure 6.

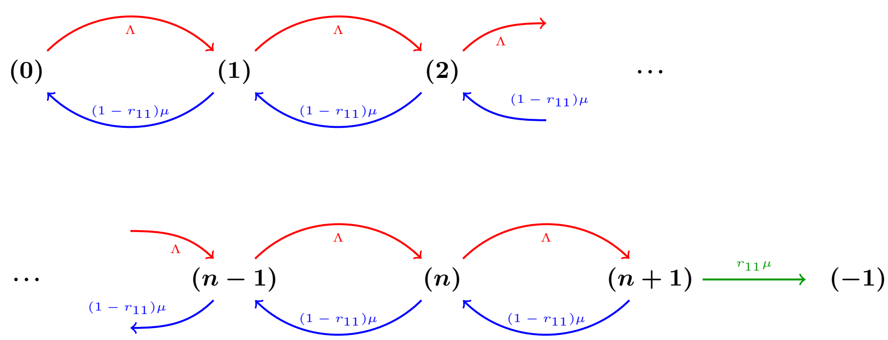
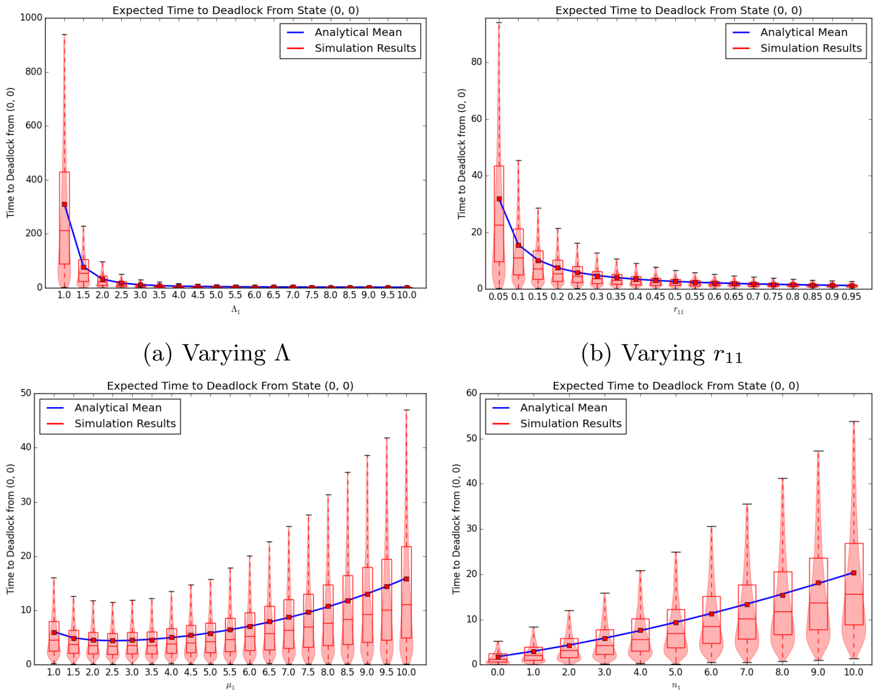


Figure 6: Diagrammatic representation of the Markov chain for Ω_1.

Figure 7 shows the effect of varying the parameters of the queueing network on times to deadlock. Base parameters of $\Lambda = 10$, $n = 3$, $\mu = 5$ and $r_{11} = 0.25$ were used.

It can be seen that increasing the arrival rate Λ and the transition probability r_{11} results in reaching deadlock faster. This is intuitive as increasing these parameters results in the first node's queue filling up quicker.

Increasing the queueing capacity n results in reaching deadlock slower. Again this is intuitive, as increasing the queueing capacity allows more customers in the system before becoming deadlock.



(a) Varying Λ

(b) Varying r_{11}

(c) Varying μ

(d) Varying n

Figure 7: Time to deadlock in Ω_1, analytical & simulation results (10,000 iterations).

5.2 Two Node Single Server without Self-Loops

Consider the open two node single server restricted queueing network shown in Figure 8. This shows two $M/M/1$ queues, with n_i queueing capacity at at each service station and service rates μ_i. Λ_i is the external arrival rates to each service station. All routing possibilities except self loops are possible, where the routing probability from node i to node j is denoted by r_{ij}.

Let this system be denoted by Ω_2 with parameter set (Λ_1, Λ_2, μ_1, μ_2, n_1, n_2, r_{12}, r_{21}).

- State space:

$$S = \{(i, j) \in \mathbb{N}^{(n_1+2\times n_2+2)} \mid 0 \leq i + j \leq n_1 + n_2 + 2\} \cup \{(-1)\}$$

  where i denotes the number of individuals:

  - In service or waiting at the first node.
  - Occupying a server but having finished service at the second node waiting to join the first.

  where j denotes the number of individuals:

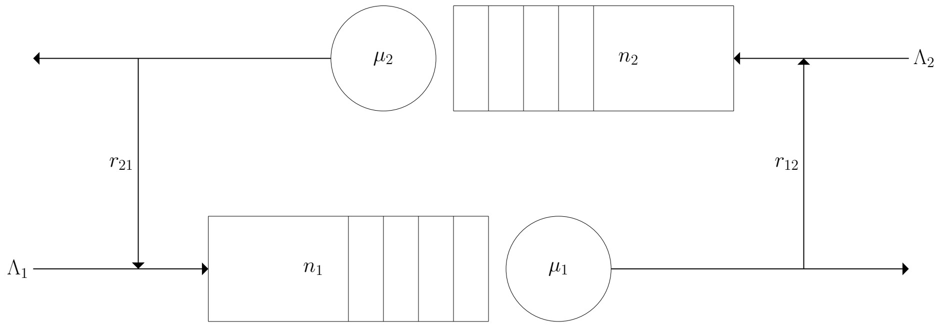


Figure 8: An open two node single server restricted queueing network.

- In service or waiting at the second node.
- Occupying a server but having finished service at the first node waiting to join the second.

and the state (-1) denotes the deadlocked state.

If we define $\delta = (i_2, j_2) - (i_1, j_1)$ for all $(i_k, j_k) \in S$, then the transitions are given by:

$$q_{(i_1,j_1),(i_2,j_2)} = \begin{cases} \left.\begin{array}{rr} \Lambda_1 & \text{if } i_1 < n_1 + 1 \\ 0 & \text{otherwise} \end{array}\right\} & \text{if } \delta = (1,0) \\ \left.\begin{array}{rr} \Lambda_2 & \text{if } j_1 < n_2 + 1 \\ 0 & \text{otherwise} \end{array}\right\} & \text{if } \delta = (0,1) \\ \left.\begin{array}{rr} (1 - r_{12})\mu_1 & \text{if } j_1 < n_2 + 2 \\ 0 & \text{otherwise} \end{array}\right\} & \text{if } \delta = (-1,0) \\ \left.\begin{array}{rr} (1 - r_{21})\mu_2 & \text{if } i_1 < n_1 + 2 \\ 0 & \text{otherwise} \end{array}\right\} & \text{if } \delta = (0,-1) \\ \left.\begin{array}{rr} r_{12}\mu_1 & \text{if } j_1 < n_2 + 2 \text{ and } (i_1, j_1) \neq (n_1 + 2, n_2) \\ 0 & \text{otherwise} \end{array}\right\} & \text{if } \delta = (-1,1) \\ \left.\begin{array}{rr} r_{21}\mu_2 & \text{if } i_1 < n_1 + 2 \text{ and } (i_1, j_1) \neq (n_1, n_2 + 2) \\ 0 & \text{otherwise} \end{array}\right\} & \text{if } \delta = (1,-1) \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

$$q_{(i_1,j_1),(-1)} = \begin{cases} r_{21}\mu_2 & \text{if } (i,j) = (n_1, n_2 + 2) \\ r_{12}\mu_1 & \text{if } (i,j) = (n_1 + 2, n_2) \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

and

$$q_{-1,s} = 0 \tag{8}$$

For $n_1 = 1$ and $n_2 = 2$, the resulting Markov chain is shown in Figure 9.

Figure 10 shows the effect of varying the parameters of the above Markov model. Base parameters of $\Lambda_1 = 4$,

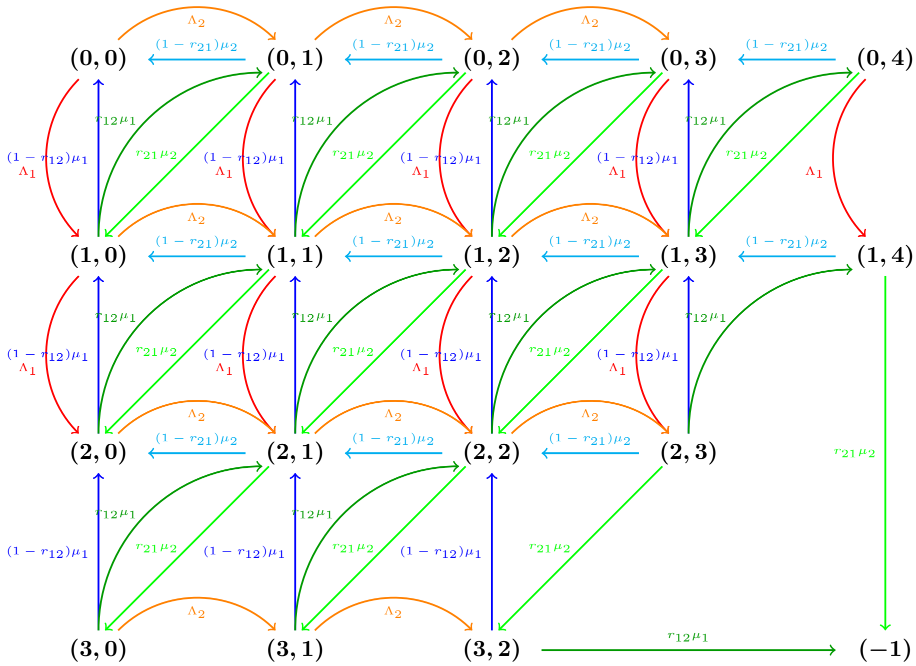


Figure 9: Diagrammatic representation of the Markov chain for Ω_2 with $n_1 = 1$ and $n_2 = 2$.

$\Lambda_2 = 5$, $n_1 = 3$, $n_2 = 2$, $\mu_1 = 10$, $\mu_2 = 8$, $r_{12} = 0.25$ and $r_{21} = 0.15$ were used. similar behaviour to Figure 7 can be seen.

5.3 Two Node Single Server with Self-Loops

Consider the open two node single server restricted queueing network shown in Figure 11. This shows two $M/M/1$ queues, with n_i queueing capacity at at each service station and service rates μ_i. Λ_i is the external arrival rates to each service station. All routing possibilities are possible, where the routing probability from node i to node j is denoted by r_{ij}.

Let this system be denoted by Ω with parameter set $(\Lambda_1, \Lambda_2, \mu_1, \mu_2, n_1, n_2, r_{11}, r_{12}, r_{21}, r_{22})$.

- State space:

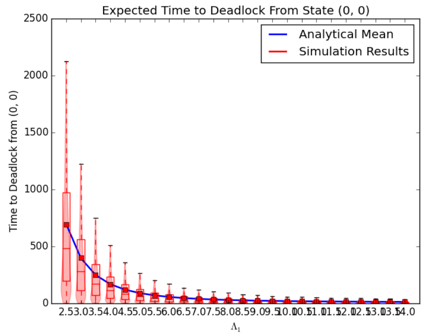
$$S = \{(i, j) \in \mathbb{N}^{(n_1+2\times n_2+2)} \mid 0 \leq i + j \leq n_1 + n_2 + 2\} \cup \{(-1), (-2), (-3)\}$$

  where i denotes the number of individuals:

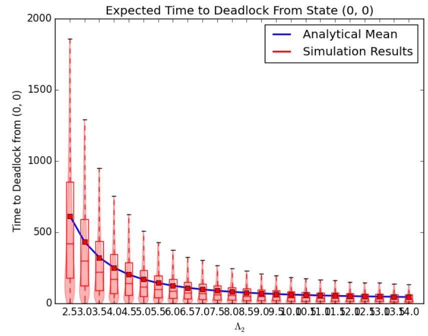
  - In service or waiting at the first node.
  - Occupying a server but having finished service at the second node waiting to join the first.

  where j denotes the number of individuals:

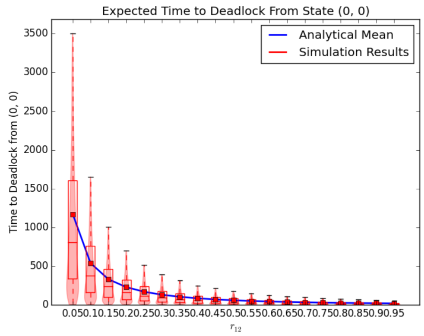
  - In service or waiting at the second node.



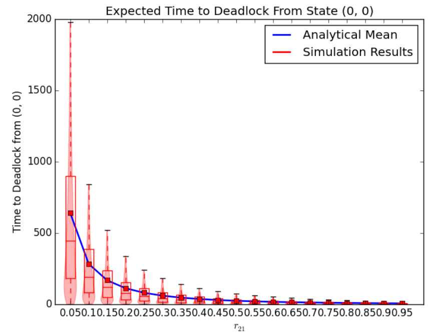
(a) Varying Λ_1



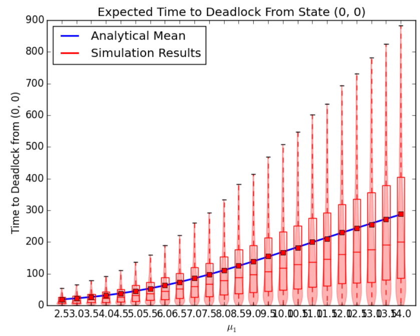
(b) Varying Λ_2



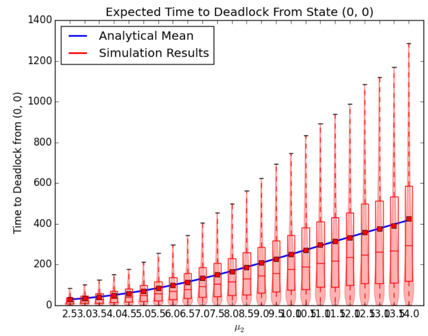
(c) Varying r_{12}



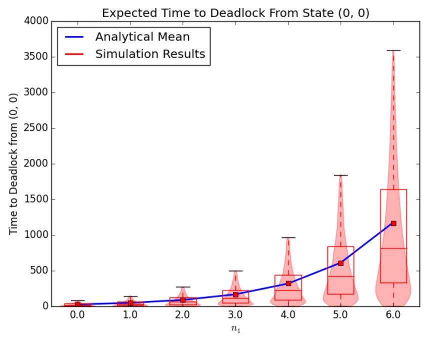
(d) Varying r_{21}



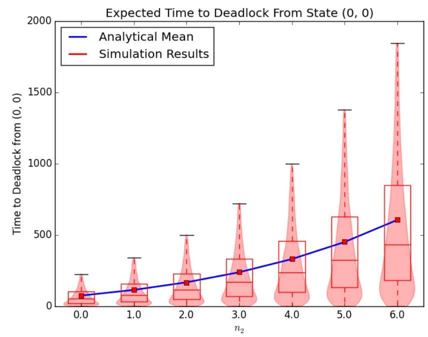
(e) Varying μ_1



(f) Varying μ_2



(g) Varying n_1



(h) Varying n_2

Figure 10: Time to deadlock in Ω_2, analytical & simulation results (10,000 iterations).

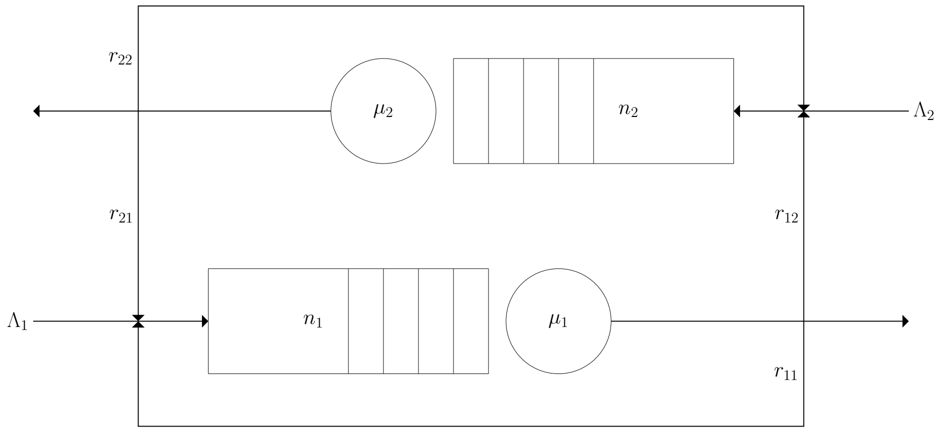


Figure 11: An open two node single server restricted queueing network.

- Occupying a server but having finished service at the first node waiting to join the second.

and the state (-3) denotes the deadlocked state cause by both nodes; (-1) denotes the deadlocked state caused by the first node only; and (-2) denotes the deadlocked state caused by the second node only. (Recall the different configurations of deadlock in Figure **??**)

If we define $\delta = (i_2, j_2) - (i_1, j_1)$ for all $(i_k, j_k) \in S$, then the transitions are given by:

$$q_{(i_1,j_1),(i_2,j_2)} = \begin{cases} \left.\begin{array}{rr} \Lambda_1 & \text{if } i_1 < n_1 + 1 \\ 0 & \text{otherwise} \end{array}\right\} & \text{if } \delta = (1,0) \\ \left.\begin{array}{rr} \Lambda_2 & \text{if } j_1 < n_2 + 1 \\ 0 & \text{otherwise} \end{array}\right\} & \text{if } \delta = (0,1) \\ \left.\begin{array}{rr} (1 - r_{11} - r_{12})\mu_1 & \text{if } j_1 < n_2 + 2 \\ 0 & \text{otherwise} \end{array}\right\} & \text{if } \delta = (-1,0) \\ \left.\begin{array}{rr} (1 - r_{21} - r_{22})\mu_2 & \text{if } i_1 < n_1 + 2 \\ 0 & \text{otherwise} \end{array}\right\} & \text{if } \delta = (0,-1) \\ \left.\begin{array}{rr} r_{12}\mu_1 & \text{if } j_1 < n_2 + 2 \text{ and } (i_1, j_1) \neq (n_1 + 2, n_2) \\ 0 & \text{otherwise} \end{array}\right\} & \text{if } \delta = (-1,1) \\ \left.\begin{array}{rr} r_{21}\mu_2 & \text{if } i_1 < n_1 + 2 \text{ and } (i_1, j_1) \neq (n_1, n_2 + 2) \\ 0 & \text{otherwise} \end{array}\right\} & \text{if } \delta = (1,-1) \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

$$q_{(i_1,j_1),(-1)} = \begin{cases} r_{11}\mu_1 & \text{if } i > n_1 \text{ and } j < n_2 + 2 \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

$$q_{(i_1,j_1),(-2)} = \begin{cases} r_{22}\mu_2 & \text{if } j > n_2 \text{ and } i < n_1 + 2 \\ 0 & \text{otherwise} \end{cases} \tag{11}$$

$$q_{(i_1,j_1),(-3)} = \begin{cases} r_{21}\mu_2 & \text{if } (i,j) = (n_1, n_2+2) \\ r_{12}\mu_1 & \text{if } (i,j) = (n_1+2, n_2) \\ 0 & \text{otherwise} \end{cases} \tag{12}$$

and

$$q_{-1,s} = 0 \tag{13}$$

$$q_{-2,s} = 0 \tag{14}$$

$$q_{-3,s} = 0 \tag{15}$$

Note that the there are only two differences between this formulation and the formulation given in Subsection **??**: the probabilities of leaving nodes 1 and 2 are now $(1 - r_{11} - r12)\mu_1$ and $(1 - r_{21} - r_{22})\mu_2$; and there are now two more ways to reach deadlock, Equation 10 and Equation 11.

For $n_1 = 1$ and $n_2 = 2$, the resulting Markov chain is shown in Figure 12.

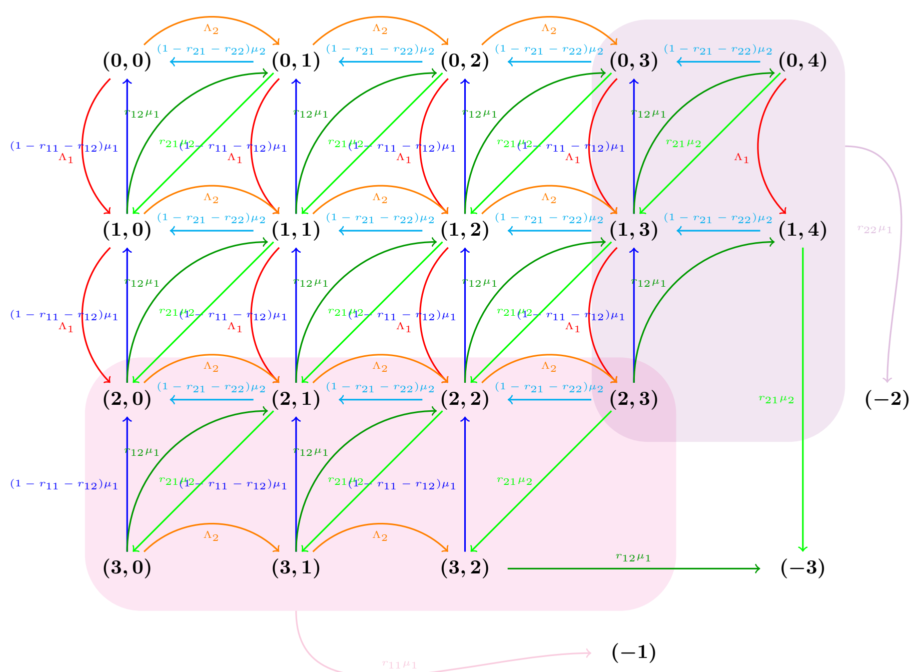
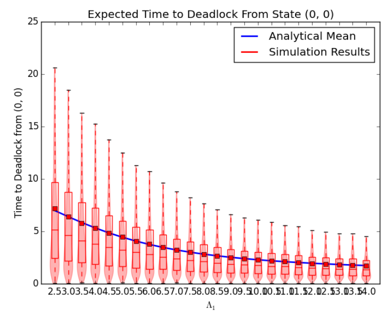
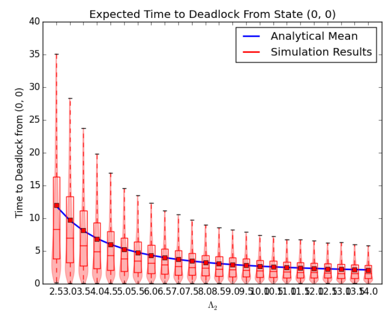


Figure 12: Diagrammatic representation of the Markov chain for Ω with $n_1 = 1$ and $n_2 = 2$.

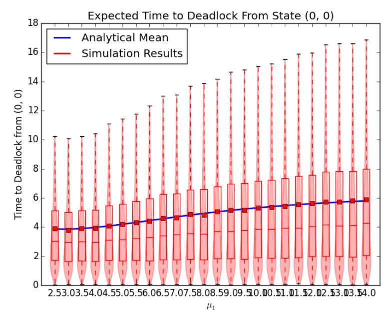
Figure 13 shows the effect of varying the parameters of the above Markov model. Base parameters of $\Lambda_1 = 4$, $\Lambda_2 = 5$, $n_1 = 3$, $n_2 = 2$, $\mu_1 = 10$, $\mu_2 = 8$, $r_{11} = 0.1$, $r_{12} = 0.25$, $r_{21} = 0.15$ and $r_{22} = 0.1$ were used.



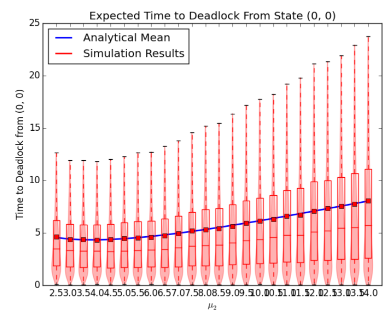
(a) Varying Λ_1



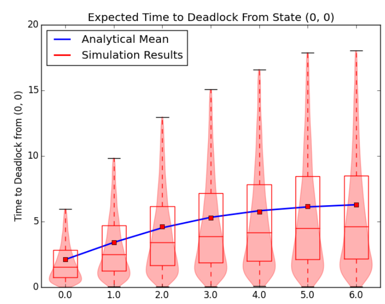
(b) Varying Λ_2



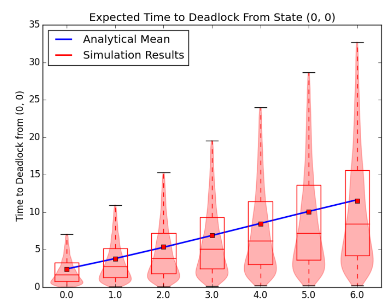
(c) Varying μ_1



(d) Varying μ_2



(e) Varying n_1



(f) Varying n_2

Figure 13: Time to deadlock in Ω, analytical & simulation results (10,000 iterations).

5.4 One Node Multi Server

Consider the open one node multi server restricted queueing network shown in Figure 14. This is an Ω_1 system with c parallel servers.

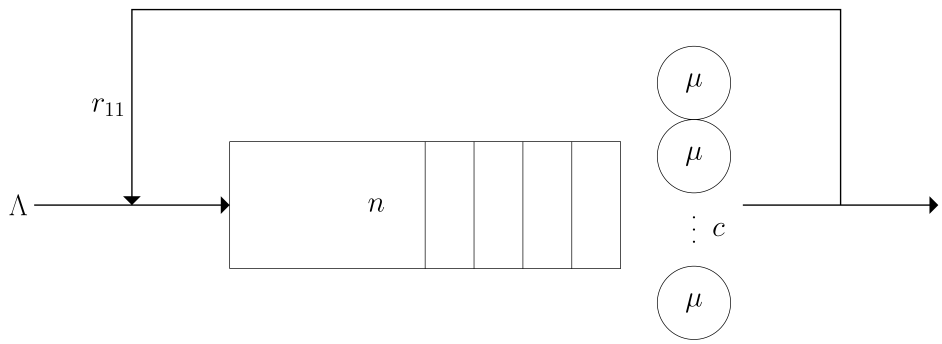


Figure 14: An open one node multi server restricted queueing network.

State space:

$$S = \{i \in \mathbb{N} \mid 0 \leq i \leq n + 2c\}$$

where i denotes the number of individuals at the node plus the number of individuals blocked at that node. For example, $i = n + c + 2$ denotes a full system, $n + c$ individuals in the node, and 2 of those individuals are also blocked. The state $i = n + 2c$ denotes the deadlocked state.

If we define $\delta = i_2 - i_1$ for all $i_k \in S$, then the transitions are given by:

$$q_{i_1,i_2} = \left\{ \begin{array}{rr} \Lambda & \text{if } \delta = 1 \\ (1 - r_{11})\mu \min(i, c) & \text{if } \delta = -1 \\ 0 & \text{otherwise} \end{array} \right\} \quad \text{if } i_1 < n + c \tag{16}$$

$$q_{i_1,i_2} = \left\{ \begin{array}{cr} (c - b)r_{11}\mu & \text{if } \delta = 1 \\ (1 - r_{11})(b - k)\mu & \text{if } \delta = -b - 1 \\ 0 & \text{otherwise} \end{array} \right\} \quad \text{if } i_1 = n + c + b \quad \forall \quad 0 \leq b \leq c \tag{17}$$

where b denotes the number of blocked customers. The Markov chain is shown in Figure 15.

Increasing the amount of servers has a similar effect to increasing the queueing capacity, there are now more transient states to go through before reaching the deadlocked state. Varying the amount of servers has a greater effect on the time to deadlock however, as any states in which customers are blocked ($i = n + c + 1$ to $i = n + 2c$) can jump back to state $i = n + c - 1$ simply with a service and an exit. Increasing the amount of servers also increases the rate at which i are reduced for most states, but not the rates at which i is increased.

Figure 16 shows the effect of varying the parameters of the above Markov model. Base parameters of $\Lambda = 6$, $n = 3$, $\mu = 2$, $r_{11} = 0.5$ and $c = 2$ were used.

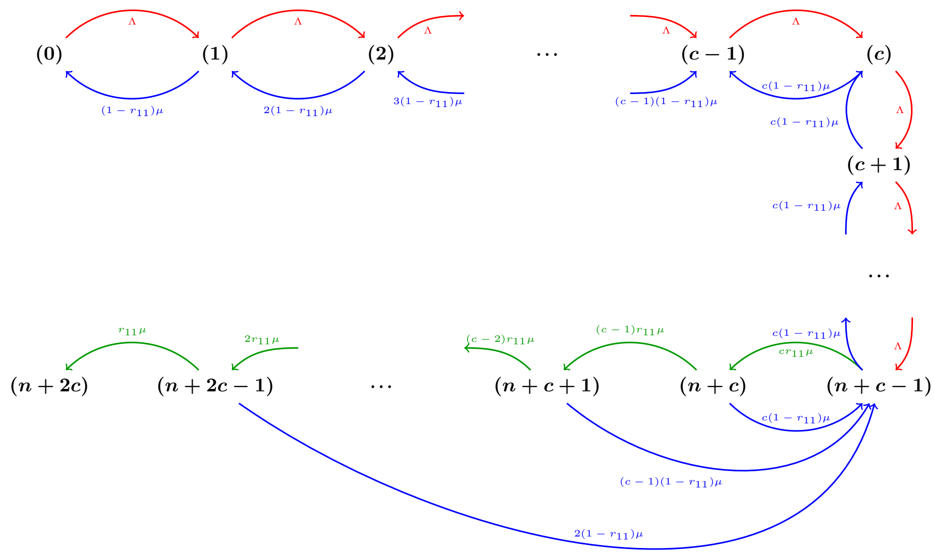


Figure 15: Diagrammatic representation of the Markov chain for a multi server Ω_1 system.

5.5 Two Node Multi Server without Self-Loops

Consider the open two node mutli server restricted queueing network shown in Figure 17. This is an Ω_2 system with c_i parallel servers at node i.

State space:

$$S = \{(i, j) \in \mathbb{N}^{(n_1+c_1+c_2)\times(n_2+c_2+c_1)} \mid i \leq n_1 + c_1 + j,\ j \leq n_2 + c_2 + i\}$$

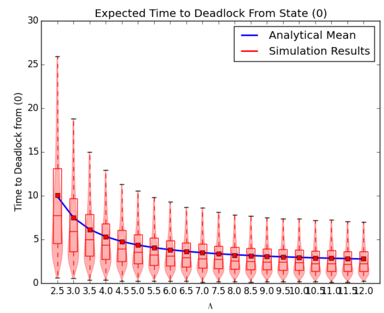
where i denotes the number of individuals at Node 1 plus the number of individuals blocked waiting to enter Node 1, and j denotes the number of individuals at Node 2 plus the number of individuals blocked waiting to enter Node 2. For example, $(i, j) = (n_1 + c_1 + 2, n_2 + c_2 + 1)$ denotes a full system, $n_1 + c_1$ individuals at Node 1, two of whom are blocked waiting to enter Node 2; $n_2 + c_2$ individuals at Node 2, one of whom is blocked waiting to enter Node 1. The state $(i, j) = (n_1 + c_1 + c_2, n_2 + c_2 + c_1)$ denotes the deadlocked state.

The Markov chain is shown in Figure 18.

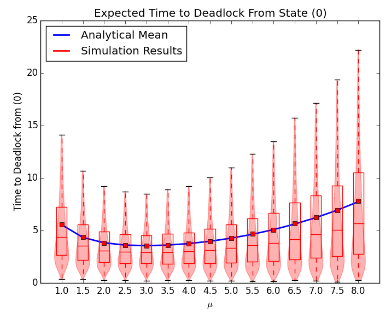
Define $\delta = (i_2, j_2) - (i_1, j_1)$, $b_1 = \max(0, i_1 - n_1 - c_1)$, $b_2 = \max(0, i_2 - n_2 - c_2)$, $s_1 = \min(i_1, c_1) - b_2$ and $s_2 = \min(i_2, c_2) - b_1$,then the transitions $q_{(i_1,j_1),(i_2,j_2)}$ are given by Table 1.

The values b_1 and b_2 correspond to the number of people blocked to Node 1 and Node 2 respectively. The values s_1 and s_2 correspond to the amount of people currently in service at Node 1 and Node 2 respectively.

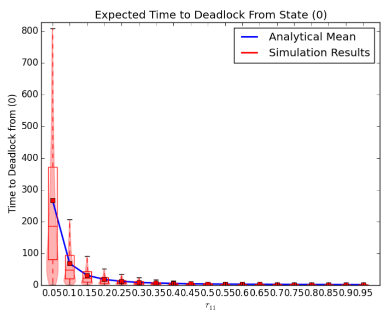
Figure 19 shows the effect of varying the parameters of the above Markov model. Base parameters of $\Lambda_1 = 9$, $\Lambda_2 = 7.5$, $n_1 = 2$, $n_2 = 1$, $\mu_1 = 5.5$, $\mu_2 = 6.5$, $r_{12} = 0.7$, $r_{21} = 0.6$, $c_1 = 2$ and $c_2 = 2$ were used.



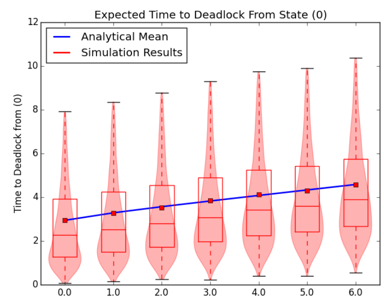
(a) Varying Λ



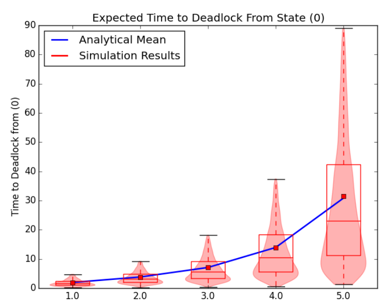
(b) Varying μ



(c) Varying r_{11}



(d) Varying n



(e) Varying c

Figure 16: Time to deadlock in multi-server Ω_1, analytical & simulation results (10,000 iterations).

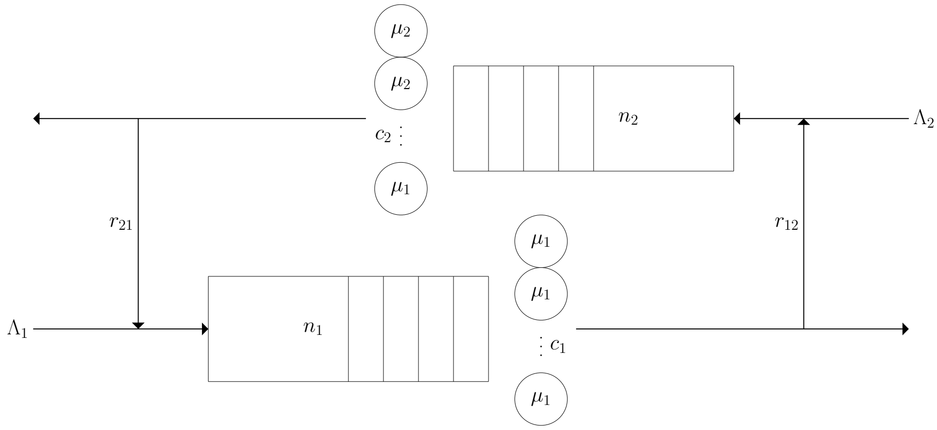


Figure 17: An open two node mutli server restricted queueing network.

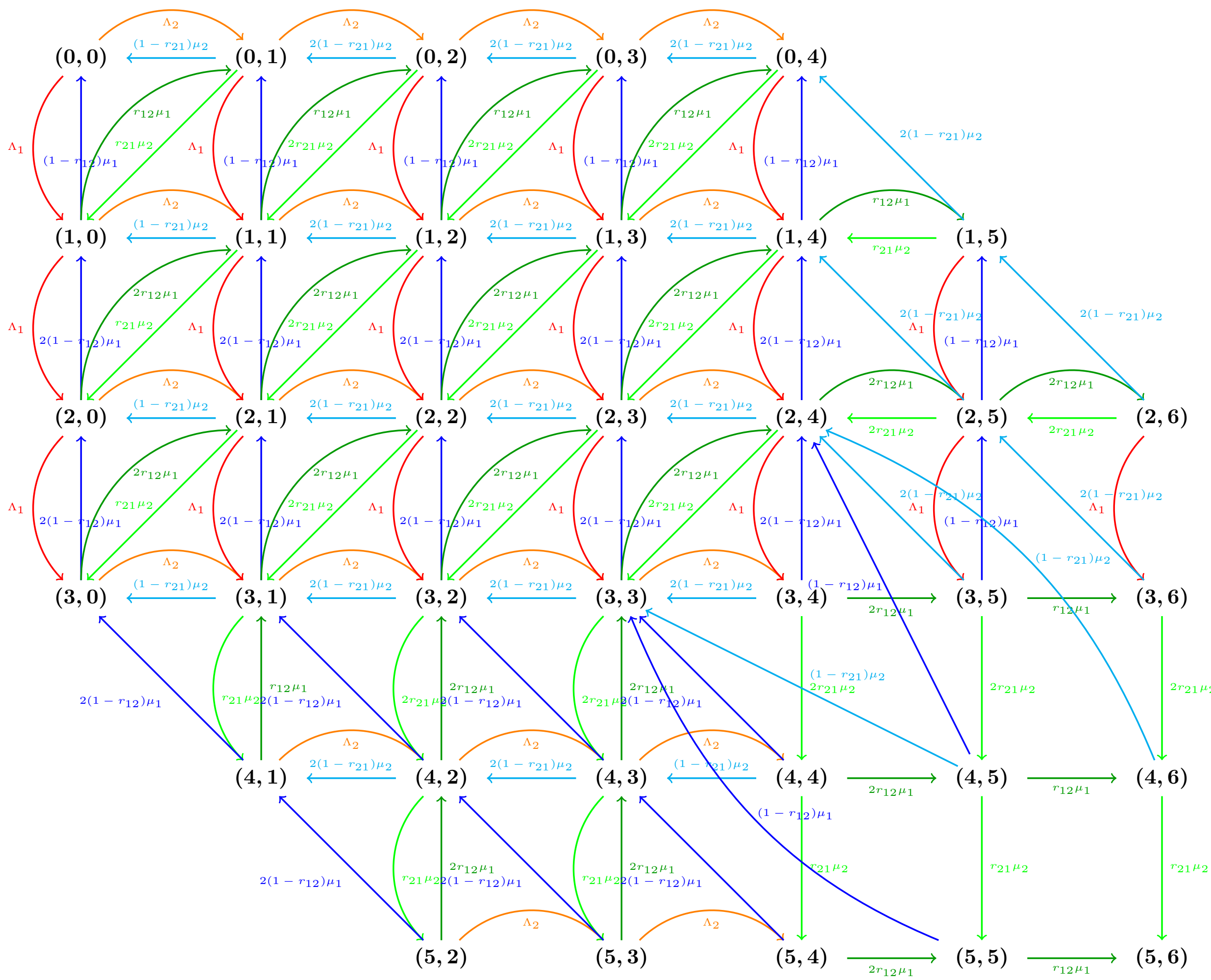
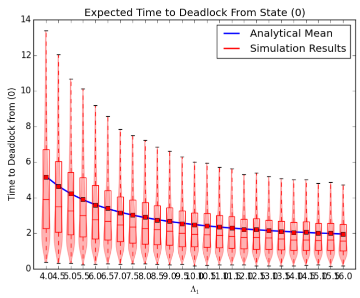
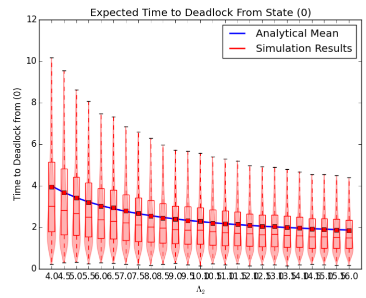


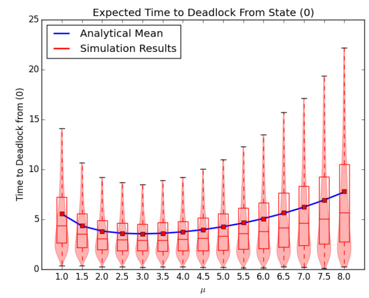
Figure 18: Diagrammatic representation of the Markov chain for a multi server Ω_2 system with $n_1 = 1$, $n_2 = c_1 = c_2 = 2$. The deadlocked state is $(5, 6)$.



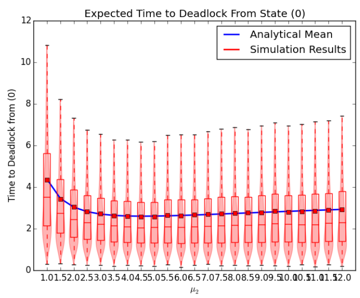
(a) Varying Λ_1



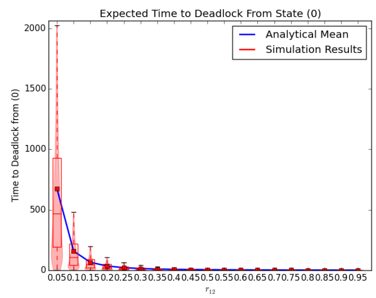
(b) Varying $\Lambda_@$



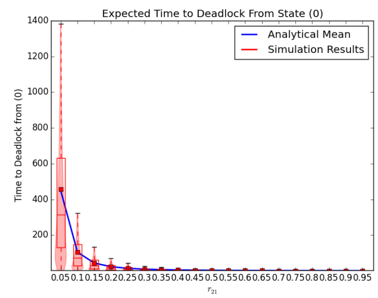
(c) Varying μ_1



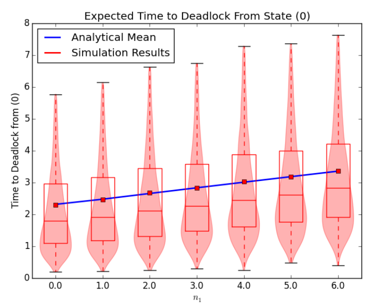
(d) Varying μ_2



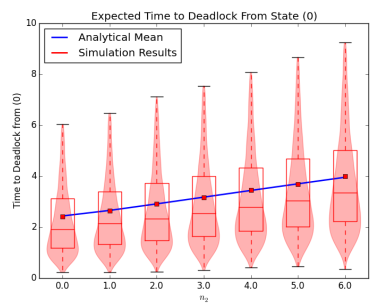
(e) Varying r_{12}



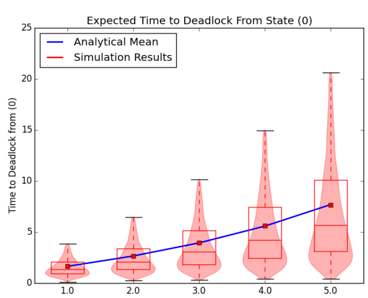
(f) Varying r_{21}



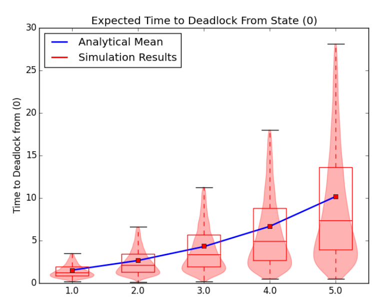
(g) Varying n_1



(h) Varying n_2



(i) Varying c_1



(j) Varying c_2

Figure 19: Time to deadlock in multi-server Ω_2, analytical & simulation results (10,000 iterations).

	$j_1 < n_2 + c_2$	$j_1 = n_2 + c_2$	$j_1 > n_2 + c_2$
$i_1 < n_1 + c_1$	Λ_1 if $\delta = (1, 0)$ Λ_2 if $\delta = (0, 1)$ $r_{12}s_1\mu_1$ if $\delta = (-1, 1)$ $r_{21}s_2\mu_2$ if $\delta = (1, -1)$ $(1 - r_{12})s_1\mu_1$ if $\delta = (-1, 0)$ $(1 - r_{21})s_2\mu_2$ if $\delta = (0, -1)$	Λ_1 if $\delta = (1, 0)$ $r_{12}s_1\mu_1$ if $\delta = (0, 1)$ $r_{21}s_2\mu_2$ if $\delta = (1, -1)$ $(1 - r_{12})s_1\mu_1$ if $\delta = (-1, 0)$ $(1 - r_{21})s_2\mu_2$ if $\delta = (0, -1)$	Λ_1 if $\delta = (1, 0)$ $r_{12}s_1\mu_1$ if $\delta = (0, 1)$ $r_{21}s_2\mu_2$ if $\delta = (0, -1)$ $(1 - r_{12})s_1\mu_1$ if $\delta = (-1, 0)$ $(1 - r_{21})s_2\mu_2$ if $\delta = (-1, -1)$
$i_1 = n_1 + c_1$	Λ_2 if $\delta = (0, 1)$ $r_{12}s_1\mu_1$ if $\delta = (-1, 1)$ $r_{21}s_2\mu_2$ if $\delta = (1, 0)$ $(1 - r_{12})s_1\mu_1$ if $\delta = (-1, 0)$ $(1 - r_{21})s_2\mu_2$ if $\delta = (0, -1)$	$r_{12}s_1\mu_1$ if $\delta = (0, 1)$ $r_{21}s_2\mu_2$ if $\delta = (1, 0)$ $(1 - r_{12})s_1\mu_1$ if $\delta = (-1, 0)$ $(1 - r_{21})s_2\mu_2$ if $\delta = (0, -1)$	$r_{12}s_1\mu_1$ if $\delta = (0, 1)$ $r_{21}s_2\mu_2$ if $\delta = (1, 0)$ $(1 - r_{12})s_1\mu_1$ if $\delta = (-1, 0)$ $(1 - r_{21})s_2\mu_2$ if $\delta = (-1, -1)$
$i_1 > n_1 + c_1$	Λ_2 if $\delta = (0, 1)$ $r_{12}s_1\mu_1$ if $\delta = (-1, 0)$ $r_{21}s_2\mu_2$ if $\delta = (1, 0)$ $(1 - r_{12})s_1\mu_1$ if $\delta = (-1, -1)$ $(1 - r_{21})s_2\mu_2$ if $\delta = (0, -1)$	$r_{12}s_1\mu_1$ if $\delta = (0, 1)$ $r_{21}s_2\mu_2$ if $\delta = (1, 0)$ $(1 - r_{12})s_1\mu_1$ if $\delta = (-1, -1)$ $(1 - r_{21})s_2\mu_2$ if $\delta = (0, -1)$	$r_{12}s_1\mu_1$ if $\delta = (0, 1)$ $r_{21}s_2\mu_2$ if $\delta = (1, 0)$ $(1 - r_{12})s_1\mu_1$ if $\delta = (-\min(b_1 + 1, b_2 + 1), -\min(b_1, b_2 + 1))$ $(1 - r_{21})s_2\mu_2$ if $\delta = (-\min(b_1 + 1, b_2), -\min(b_1 + 1, b_2 + 1))$

Table 1: Table of transitions for a multi server two node network.

6 A Bound on the Time to Deadlock

References

[1] B. Avi-Itzhak and M. Yadin. A sequence of two servers with no intermediate queue. *Management science*, 11(5):553–564, 1965.

[2] J. Baber. *Queues in series with blocking.* PhD thesis, Cardiff University, 2008.

[3] J. Cheng. Task-wait-for graphs and their application to handling tasking deadlocks. In *Proceedings of the conference on TRI-ADA'90*, pages 376–390. ACM, 1990.

[4] H. Cho, T. Kumaran, and R. Wysk. Graph-theoretic deadlock detection and resolution for flexible manufacturing systems. *IEEE transactions on robotics and automation*, 11(3):413–421, 1995.

[5] E. Coffman and M. Elphick. System deadlocks. *Computing surveys*, 3(2):67–78, 1971.

[6] Edsger W Dijkstra. The mathematics behind the bankers algorithm. In *Selected Writings on Computing: A Personal Perspective*, pages 308–312. Springer, 1982.

[7] A. Elmagarmid. A survey of distributed deadlock detection algorithms. *ACM Sigmod Record*, 15(3):37–45, 1986.

[8] G. Hunt. Sequential arrays of waiting lines. *Operations research*, 4(6):674–683, 1956.

[9] P. Kawadkar, S. Prasad, and A.D. Dwivedi. Deadlock avoidance based on bankers algorithm for waiting state processes. *International journal of innovative science and modern engineering*, 2(12), 2014.

[10] N. Koizumi, E. Kuno, and T.E. Smith. Modeling patient flows using a queueing network with blocking. *Health care management science*, 8(1):49–60, 2005.

[11] R. Korporaal, A. Ridder, P. Kloprogge, and R. Dekker. An analytic model for capacity planning of prisons in the netherlands. *The journal of the operational research society*, 51(11):1228–1237, 2000.

[12] S. Kundu and I. Akyildiz. Deadlock buffer allocation in closed queueing networks. *Queueing systems*, 4(1):47–56, 1989.

[13] G. Latouche and M. Neuts. Efficient algorithmic solutions to exponential tandem queues with blocking. *SIAM journal on algebraic discrete methods*, 1(1):93–106, 1980.

[14] J. Liebeherr and I. Akyildiz. Deadlock properties of queueing networks with finite capacities and multiple routing chains. *Queueing systems*, 20(3-4):409–431, 1995.

[15] W. Stewart. *Probability, markov chains, queues, and simulation.* Princeton university press, 2009.

[16] Y. Takahashi, H. Miyahara, and T. Hasegawa. An approximation method for open restricted queueing networks. *Operations research*, 28(3):594–602, 1980.