On Deadlocking in Queueing Networks

Geraint Ian Palmer, Paul Harper, Vincent Knight

October 22, 2015

Abstract

Open restricted queueing networks give rise to the phenomenon of deadlock, that is some customers may be unable to ever leave a server due to recursive upstream blocking. This paper explores deadlock in restricted queueing networks, presents a method of detecting deadlock in these systems, and builds Markov chain models of these deadlocking networks. The five networks for which Markov models are given are:

- 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.

These models are compared to results obtain using a simulation and the above deadlock detection method. Finally a bound on the time to deadlock of the two node single-server with loops network is derived by comparing properties with other queueing networks that are embedded within it. It is hoped that this paper addresses a gap in the literature on the deadlocking properties of open restricted queueing networks.

1 Introduction & Background

The study and modelling of queueing networks with blocking is an important tool in many aspects of operational research, both analyically and through simulation. These models have applications in many varied settings such as healthcare, supply chains, manufacturing and communications systems. However, these types of models have their limitations, due to their potential to become permanently blocked in deadlock, or a deadly embrace of resources. These deadlocks can be real in which case accurately modelling deadlock is needed, or they can occur in models where deadlock situations are easily adjusted in reality. In the latter case, such as by swapping two customers, a good understanding of deadlock is needed in order to model the adjusted reality.

Figure 1 shows an open two node restricted queueing network in deadlock. The customer at the top server is blocked from entering the bottom node as there is a full queue, and similarly the customer at he bottom server is blocked from entering the top node as there is a full queue. It is clear that by following the rules of a queueing network, no more natural movement can happen.

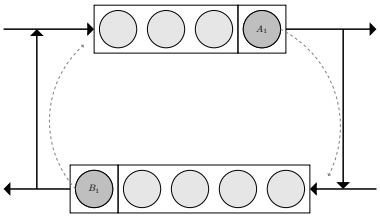


Figure 1: Example of as open two node restricted queueing network in deadlock.

Restricted open queueing networks that can experience deadlock are under-discussed in the literature. This paper aims to explore deadlock to gain this understanding in order to build more accurate models of deadlocking systems

This paper is structured as follows: The remainder of this section sets up the notation and definitions used for the rest of the paper. Section 2 discusses the exsisting literature on deadlock and deadlock strategies for queueing networks. Section 3 presents a method of detecting deadlock in 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 obtained through the simulation model. Finally Section 6 derives a bound for the expected time to deadlock for one of the queueing networks discussed.

1.1 Open Restricted Queueing Networks

Queueing networks are described as open if customers can enter and leave the system from the exterior. Restricted networks are those where at lease one service centre has limited queueing space or capacity before it.

This paper is concerned with open restricted queueing networks that experience Type I blocking. Throughout this paper service centers will be referred to as nodes, and for the ith node of an open restricted queueing network the following notation is used:

- 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 queueing network is in a state 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.

2 Literature Review

Restricted queueing networks that exhibit blocking are well discussed in the literature, both exact [2, 3, 11, 13,16,19] and approximate methods [14,18,19,21]. Discussion on restricted queueing networks with feedback loops, that may exhibit deadlock, are sparse however. In fact, the problem of deadlock in queueing networks has either been ignored, not studied, or assumed resolved in much of the literature [18, 19].

Central to the study of deadlock in queueing networks is the concept of blocking. In [18] three types of blocking are described: Type I blocking (blocked at service, BAS, transfer blocking) occurs when a customer is blocked after completing service, remains with the server until capacity at their destination node becomes available. Type II blocking (blocked before service, BBS, service blocking) occurs when a customer declares their destination before beginning service, and is only granted service if there is available capacity at their destination node. In Type III blocking (repetitive services, RS-FD and RS-RD, rejection blocking) instead of getting blocked a customer is required to repeat thier service if there is no capacity at theor destination. This type of blocking comes in two flavours, fixed destination where the customer's destination does not change at each repitition of service, and random destination, where the customer's destination is resamples from a probability distribution after each repitition.

There has been a body of reseach into detection and prevention of deadlock which doesn't consider the underlying stochastic structure of the system [7]. These general deadlocks occur in flexible manufacturing systems and distributed communication systems. Conditions for this type of deadlock, also referred to as deadly embraces [7], 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.

Allowing a system to reach deadlock can be problamatic in cases where automated systems cannot continue operations, or where simulations cannot accurately model reality. In general there are three strategies for dealing with the problem of deadlock [9, 12]:

- 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 under Type I blocking. For closed networks of K customers with only one class of customer, [15] 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 [17], 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.

Further conditions on deadlock prvention in closed queueing networks are reviewed in [18], including closed networks under different blocking mechanisms such Type II and Type III blocking.

2.2 Deadlock Avoidance

There are algorithms discussed in the literature for the dynamic avoidance of deadlock. In the Banker's Algorithm [8, 12], unsafe states, those that will lead to deadlock, are avoided by ensuring actions leading to these states are not carried out.

A common deadlock avoidance technique in flexible manufacturing systems is the use of dynamic resource allocation policies using petri net nodels of the system [10, 23]. Some resource allocations are disallowed if that allocaton will lead to deadlock.

Another deadlock avoidance algorithm in [4], for systems with one type of each resource, makes use of a directed resource allocation graph. Resource requests are not allocated resources if that allocation leads to a directed cycle in the resource allocation graph.

2.3 Deadlock Detection & Recovery

A popular method of detecting general deadlock is the use of wait-for graphs, state-graphs and their variants [5, 6, 7, 9]. These wait-for graphs, keep track of all circular wait relations between tasks.

In [7] 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 [6] 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. This type strategy of is used in this paper to detect deadlock in queueing systems.

Deadlock detection and recovery in closed queueing networks through swapping customers is assumed in [19], with zero transition time assumed between deadlocked states and the corresponding resolved state. Though time to resove deadlock may not be negligible in reality. In [1], deadlock detection and recovery is listed as one of the two possible soulutions for handling deadlock in queueing networks, although this isn't discussed further.

3 Deadlock Detection

In order to detect when deadlock has occured in a queueing network simulation, the state digraph is used. This is a form of wait-for graph, and is 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.*

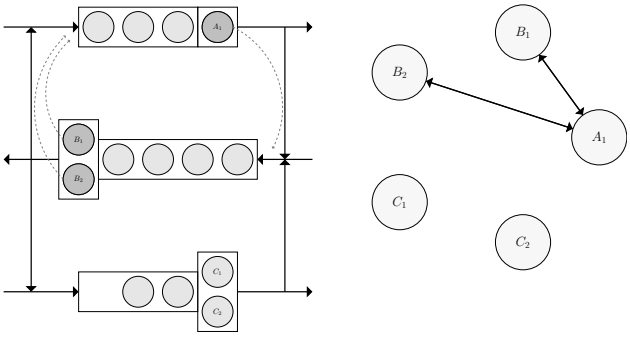
To illustrate this concept Figure 2 show examples of queueing network in and out of deadlock, and the corresponding state digraph in each case.

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

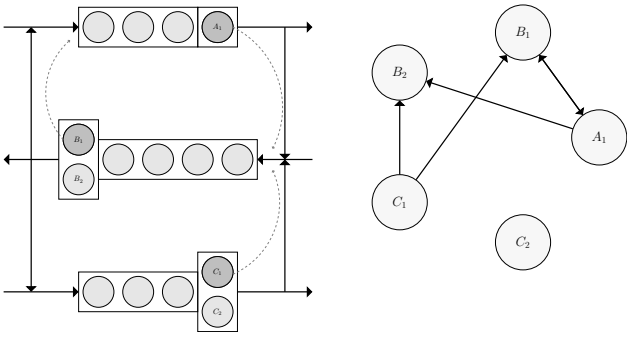
- If X is unoccupied, then X has no incident edges.
- Consider the case when X is occupied by individual a, whose next destination is node j. Then X's direct successors are the servers occupied by individuals who are blocked or in service at node j.
- It can interpreted that all X's descendants are the servers whose occupants are directly or indirectly blocking a, and interpret all X's ancestors as those servers whose individuals who are being blocked directly or indirectly by a.
- 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.
- Note that the only possibilities for $\deg^{\text{out}}(X)$ are 0 or c_j. If $\deg^{\text{out}}(X) = c_j$ then a is blocked by all its direct successors. The only other situation is that a is not blocked, and $X \in G(t)$ because a is in service at X and blocking other individuals, in which case $\deg^{\text{out}}(X) = 0$.
- It is clear that if all of X's descendants are occupied by blocked individuals, then the system is deadlocked at time t.
- By definition all of X's ancestors are occupied by blocked individuals.

The following results detect deadlock for open restricted queueing networks.

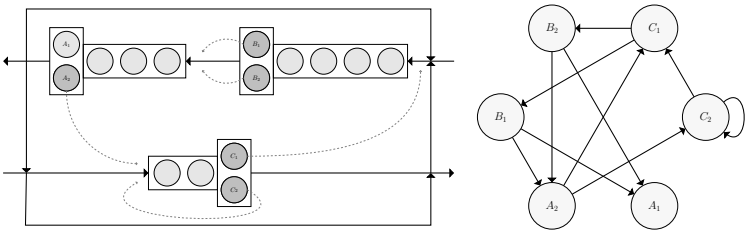
Theorem 1. *A deadlocked state arises at time t if and only if $D(t)$ contains a knot.*



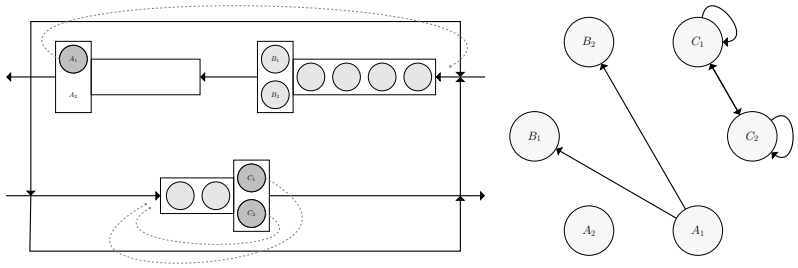
(a) A three node queueing network in deadlock, with state digraph.



(b) A three node queueing network not in deadlock, with state digraph.



(c) A three node queueing network in not deadlock, with state digraph.



(d) A three node queueing network in deadlock, with state digraph.

Figure 2: Examples of state digraphs with their corresponding queueing networks.

Proof. Consider one weakly connected component $G(t)$ of $D(t)$ at time t.

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.

□

The knot condition can be simplified for specific cases.

Proposition 1. *For queueing networks:*

1. *with one node*
2. *with two nodes, each with two or fewer parallel servers*
3. *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.

1. Consider a one node queueing network.

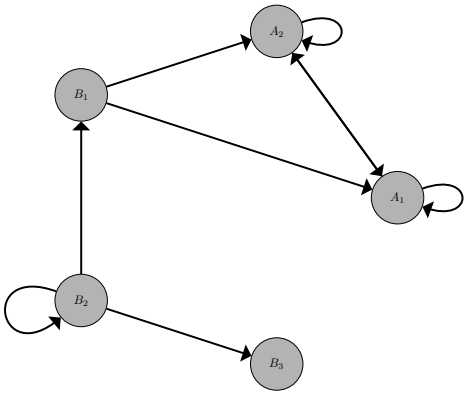
   If there is deadlock, then all servers are occupied by blocked individuals, and so all servers have an out-edge.

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

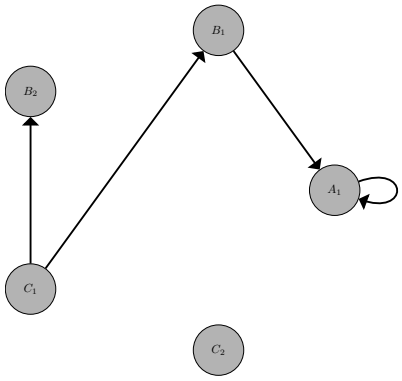
   If both nodes are involved in the deadlock, so there is a customer in node 1 blocked from entering node 2, and a customer from node 2 blocked from entering node 1, then all servers in node 1 and node 2 in $D(t)$ will have out edges as they are occupied by a blocked individual. The servers of node 1 and 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 an 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 empty or still in service, 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 two node queueing network with at least one node with more than 2 servers, consider the following counter-example:



(a) State Digraph of Counter-Example 1.



(b) State Digraph of Counter-Example 2.

Figure 3: Counter examples.

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 3a has a weakly connected component with a sink.

3. 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:

   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 3b has a weakly connected component with a sink.

□

4 Simulation Model

A simulation model is 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. The model is generic, and can handle any number of service center nodes, each defined by its external arrival rate, service rate, number of servers and queueing capacity, which may be set to ∞ is needed. A routing matrix with entries r_{ij} describes how every node is connected to every other node.

The simulation model is built in an object-orientated manor using Python. Customers, nodes, servers and the queueing network itself are objects that interact or contain each other. The code for the simulation can be found here: `https://github.com/geraintpalmer/SimulatingAQingNetwork`.

The state digraph $D(t)$ described in Section 3 is an attribute of the queueing network. Edges are drawn or removed whenever the following events take place: a customer begins service at a server, a customer becomes blocked, a customer leaves a server. The model is a discrete event simulation, and the deadlocked condition is checked immediately before the clock is moved on. Here a brute force algorithm is used to check whether each strongly connected component of $D(t)$ is a knot. Once a knot is found the simulation terminates and the time elapsed recorded.

5 Markovian Models of Deadlocking Queueing Networks

In this section markov models are build for the following deadlocking queueing networks, and their expected time to deadlock found:

- Open one node, single-server restricted queueing network with feedback loop. (Section 5.1)
- Open two node, single-server restricted queueing network with routes between nodes. (Section 5.2)
- Open two node, single-server restricted queueing network with routes between nodes and feedback loops. (Section 5.3)
- Open one node, multi-server restricted queueing network with feedback loop. (Section 5.4)
- Open two node, mutli server restricted queueing network with routes between nodes. (Section 5.5)

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. Deadlocked states cannot transition to any other state, and so are absorbing states 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 [20]. The canonical form of an absorbing Markov chain is

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

where I is the 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 expected number of time steps to absorption, or deadlock can be found.

When there is more than one deadlocked state, there is more than one absorbing state in the Markov chain. Here the expected time to absorbtion is the expected time to a deadlocked state, whichever one that may be.

5.1 One Node Single-Server

Consider the open one node single-server restricted queueing network with feedback loop shown in Figure 4. This shows an $M/M/1/n$ queue where 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.

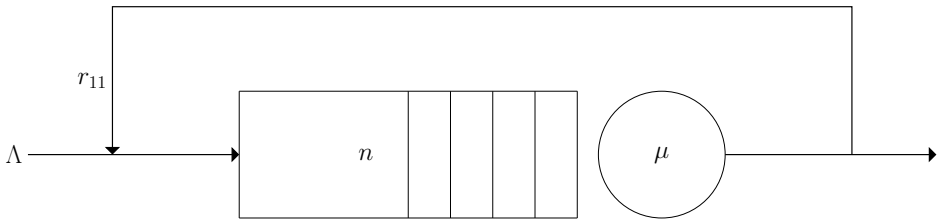


Figure 4: An open one node single-server restricted queueing network.

State space:

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

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

Define $\delta = i_2 - i_1$ for all $i_k \geq 0$. The transitions are given by:

$$q_{i_1,i_2} = \begin{cases} \left.\begin{matrix} \Lambda & \text{if } i_1 < n+1 \\ 0 & \text{otherwise} \end{matrix}\right\} & \text{if } \delta = 1 \\ (1-r_{11})\mu & \text{if } \delta = -1 \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

$$q_{i,(-1)} = \begin{cases} r_{11}\mu & \text{if } i = n+1 \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

$$q_{-1,i} = 0 \tag{4}$$

The Markov chain is shown in Figure 5.

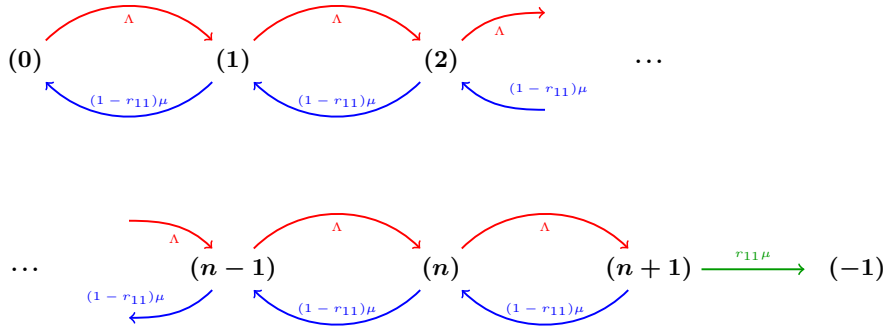


Figure 5: Diagrammatic representation of the Markov chain for Ω_1.

Figure 6 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 $r_{11} = 0.25$ are used.

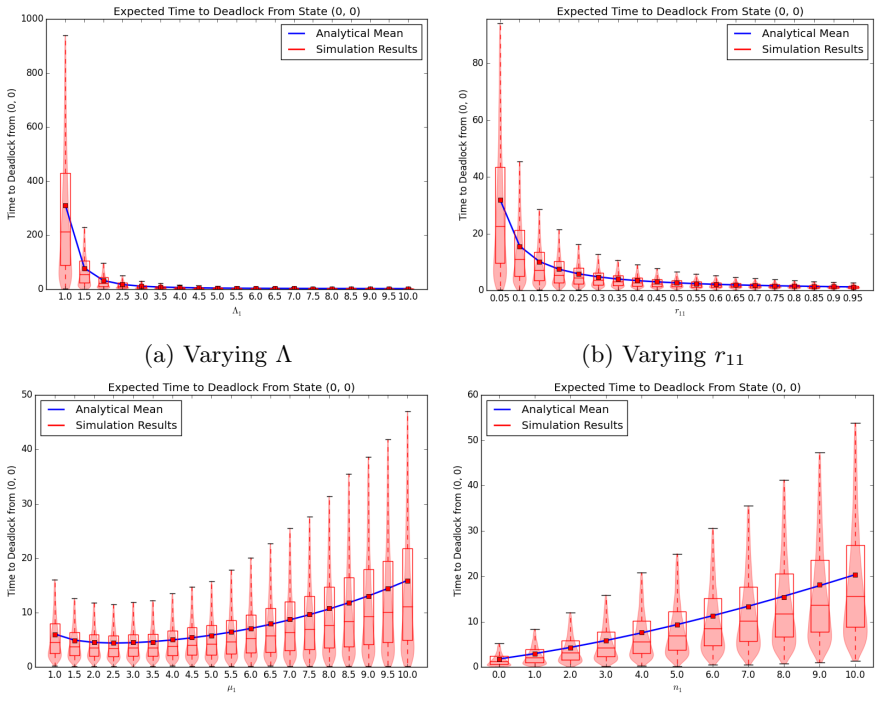
It can be seen that increasing the arrival rate Λ and the transition probability r_{11} results in reaching deadlock faster. This is intuitive as increasing these parameters results in the queue filling up quicker. Increasing the queueing capacity n results in reaching deadlock slower. Again this is intuitive, as increasing the queueing capacity allows more customers in the system before becoming deadlocked.

The behaviour as the service rate μ varies is not monotonic, as the service rate contributes towards both moving customers from the system and allowing customers to rejoin the queue, causing blockages and deadlock. This behaviour is described in the following remark.

Remark 1. *The function $\omega_1(\mu)$ that describes the time to deadlock of an Ω_1 system as the service rate μ varies, and all other parameters are fixed, has one critical point and is a local minimum for $\mu \in (0, \infty)$.*

This can be seen by interpretation. The behaviour of $\omega_1(\mu)$ can be interpreted as follows:

- At $\lim_{\mu \to 0} \omega_1(\mu)$ there is infinite service time, and so infinite time until deadlock.
- At $\lim_{\mu \to \infty} \omega_1(\mu)$ there is zero service time, the queue can never fill up, and so infinite time to deadlock.



(a) Varying Λ

(b) Varying r_{11}

(c) Varying μ

(d) Varying n

Figure 6: Time to deadlock in Ω_1, analytical & simulation results (10,000 iterations).

- At low service rates below a certain threshold $\hat{\mu}$, the arrival rate is relatively large compared to the service rate, and we can assume a saturated system. At this point services where a customer exits the system does not have much of an effect, as we can assume another arrival immediately. However services where a customer wishes to rejoin the queue results in a blockage as the system is saturated. Therefore, increasing the service rate here increases the chance of a blockage, and so the chance of deadlock.
- Above $\hat{\mu}$ the service rate is large enough that we cannot assume a saturated system, and so services where the customer exits the system does have an affect on the number of customers in the system. Thus increasing the service rate removes people from the system, and as such there is less chance of getting blocked and deadlocked.

Remark 2. $\arg\max_{\mu\in[a,b]} \omega_1(\mu)$ *is either* a *or* b.

From the closed interval method of finding absolute maximum [22], the absolute maximum of $\omega_1(\mu)$ *on the closed interal* $[a, b]$ *is either the critical points in* (a, b), a *or* b. *The only critical point in* (a, b) *is* $\hat{\mu}$, *and is a local minimum (from Remark 1), and so* $\hat{\mu} \leq a$ *and* $\hat{\mu} \leq b$. *Therefore* $\omega_1(\mu)$ *obtains its maximum at either* a *or* b.

5.2 Two Node Single-Server without Self-Loops

Consider the open two node single-server restricted queueing network shown in Figure 7. This shows two $M/M/1/n_i$ queues, with service rates μ_i and external arrival rates Λ_i. All routing possibilities r_{ij} are possible except self-loops r_{ii} for each node i.

Let this system be denoted by Ω_2 with parameter set $(\Lambda_1, \Lambda_2, \mu_1, \mu_2, n_1, n_2, r_{12}, r_{21})$.

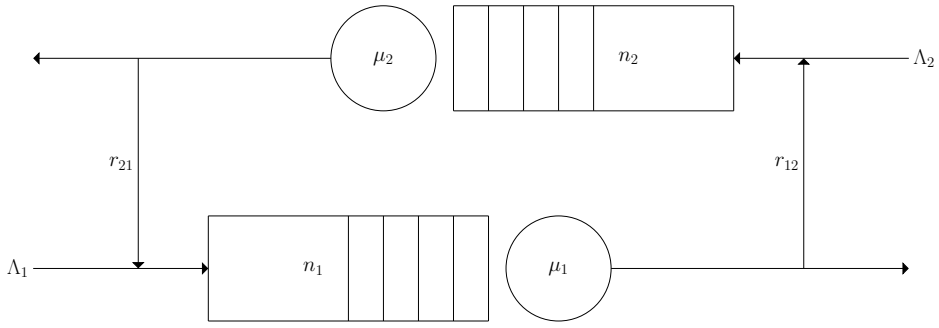


Figure 7: An open two node single-server restricted queueing network.

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:

- 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.

Define $\delta = (i_2, j_2) - (i_1, j_1)$ for all $(i_k, j_k) \in S$. The transitions are given by:

$$q_{(i_1,j_1),(i_2,j_2)} = \begin{cases} \left.\begin{array}{rl} \Lambda_1 & \text{if } i_1 < n_1 + 1 \\ 0 & \text{otherwise} \end{array}\right\} & \text{if } \delta = (1,0) \\ \left.\begin{array}{rl} \Lambda_2 & \text{if } j_1 < n_2 + 1 \\ 0 & \text{otherwise} \end{array}\right\} & \text{if } \delta = (0,1) \\ \left.\begin{array}{rl} (1-r_{12})\mu_1 & \text{if } j_1 < n_2 + 2 \\ 0 & \text{otherwise} \end{array}\right\} & \text{if } \delta = (-1,0) \\ \left.\begin{array}{rl} (1-r_{21})\mu_2 & \text{if } i_1 < n_1 + 2 \\ 0 & \text{otherwise} \end{array}\right\} & \text{if } \delta = (0,-1) \\ \left.\begin{array}{rl} r_{12}\mu_1 & \text{if } j_1 < n_2 + 2 \text{ and } (i_1,j_1) \neq (n_1+2,n_2) \\ 0 & \text{otherwise} \end{array}\right\} & \text{if } \delta = (-1,1) \\ \left.\begin{array}{rl} r_{21}\mu_2 & \text{if } i_1 < n_1 + 2 \text{ and } (i_1,j_1) \neq (n_1,n_2+2) \\ 0 & \text{otherwise} \end{array}\right\} & \text{if } \delta = (1,-1) \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

$$q_{(i_1,j_1),(-1)} = \begin{cases} r_{21}\mu_2 & \text{if } (i,j) = (n_1, n_2+2) \\ r_{12}\mu_1 & \text{if } (i,j) = (n_1+2, n_2) \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

$$q_{-1,s} = 0 \tag{7}$$

For $n_1 = 1$ and $n_2 = 2$, the resulting Markov chain is shown in Figure 8.

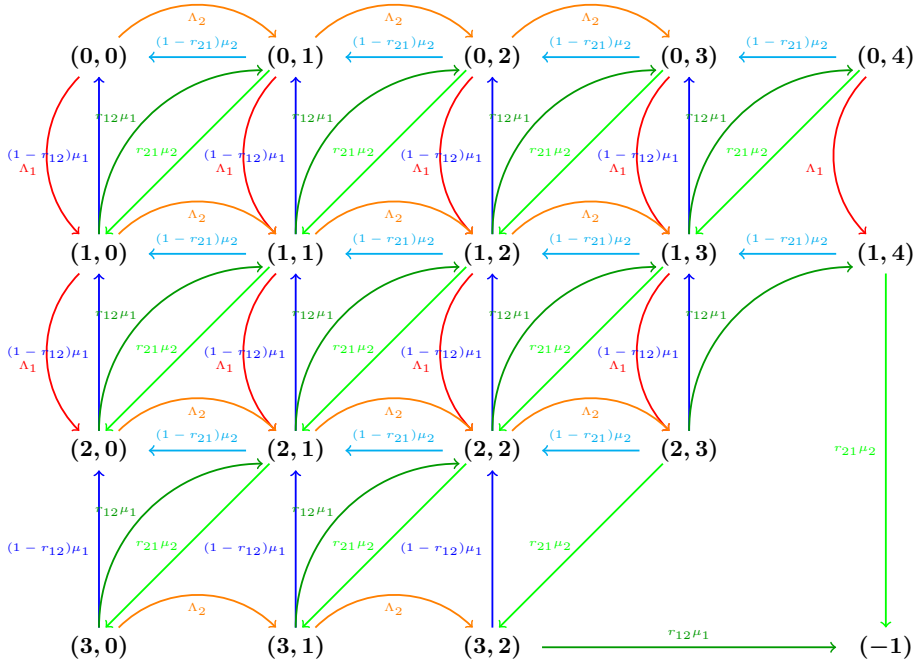


Figure 8: Diagrammatic representation of the Markov chain for Ω_2 with $n_1 = 1$ and $n_2 = 2$.

Figure 9 shows the effect of varying the parameters of the above Markov model. Base parameters of $\Lambda_1 = 4$, $\Lambda_2 = 5$, $n_1 = 3$, $n_2 = 2$, $\mu_1 = 10$, $\mu_2 = 8$, $r_{12} = 0.25$ and $r_{21} = 0.15$ are used. Similar behaviour to Figure 6 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 10. This shows two $M/M/1/n_i$ queues with service rates μ_i and external arrival rates Λ_i. 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 (Λ_1, Λ_2, μ_1, μ_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.

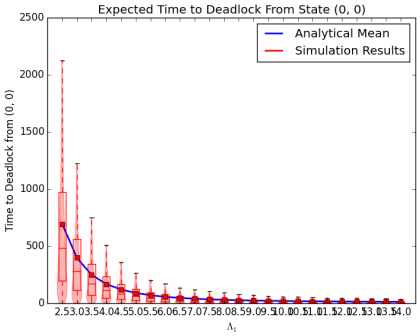
where j denotes the number of individuals:

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

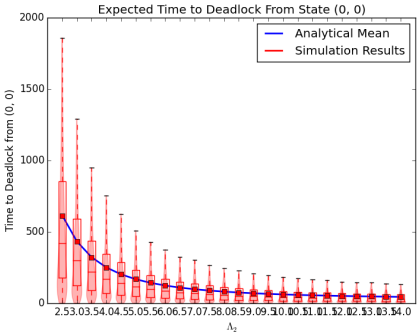
and the state (-3) denotes the deadlocked state cause by both nodes; (-1) denotes the deadlocked state caused by the first node only; and (-2) denotes the deadlocked state caused by the second node only.

Define $\delta = (i_2, j_2) - (i_1, j_1)$ for all $(i_k, j_k) \in S$. The transitions are given by:

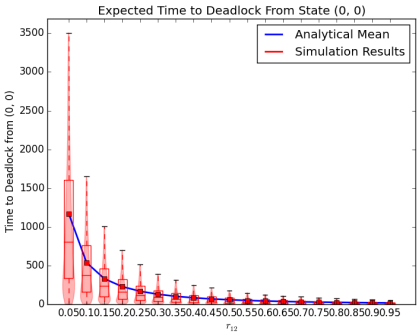
$$q_{(i_1,j_1),(i_2,j_2)} = \begin{cases} \left.\begin{array}{rl} \Lambda_1 & \text{if } i_1 < n_1+1 \\ 0 & \text{otherwise} \end{array}\right\} & \text{if } \delta = (1,0) \\ \left.\begin{array}{rl} \Lambda_2 & \text{if } j_1 < n_2+1 \\ 0 & \text{otherwise} \end{array}\right\} & \text{if } \delta = (0,1) \\ \left.\begin{array}{rl} (1-r_{11}-r_{12})\mu_1 & \text{if } j_1 < n_2+2 \\ 0 & \text{otherwise} \end{array}\right\} & \text{if } \delta = (-1,0) \\ \left.\begin{array}{rl} (1-r_{21}-r_{22})\mu_2 & \text{if } i_1 < n_1+2 \\ 0 & \text{otherwise} \end{array}\right\} & \text{if } \delta = (0,-1) \\ \left.\begin{array}{rl} r_{12}\mu_1 & \text{if } j_1 < n_2+2 \text{ and } (i_1,j_1) \neq (n_1+2,n_2) \\ 0 & \text{otherwise} \end{array}\right\} & \text{if } \delta = (-1,1) \\ \left.\begin{array}{rl} r_{21}\mu_2 & \text{if } i_1 < n_1+2 \text{ and } (i_1,j_1) \neq (n_1,n_2+2) \\ 0 & \text{otherwise} \end{array}\right\} & \text{if } \delta = (1,-1) \\ 0 & \text{otherwise} \end{cases} \tag{8}$$



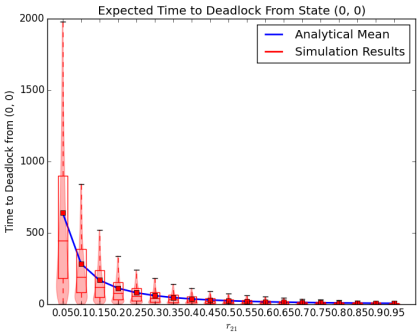
(a) Varying Λ_1



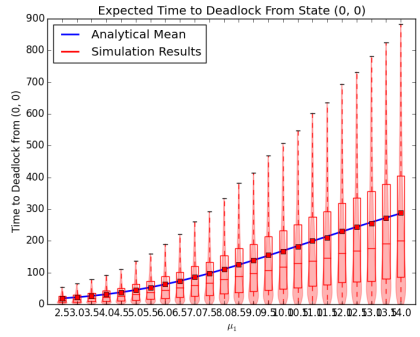
(b) Varying Λ_2



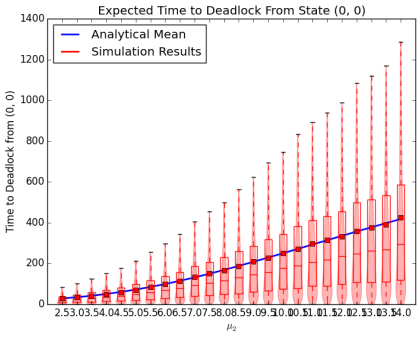
(c) Varying r_{12}



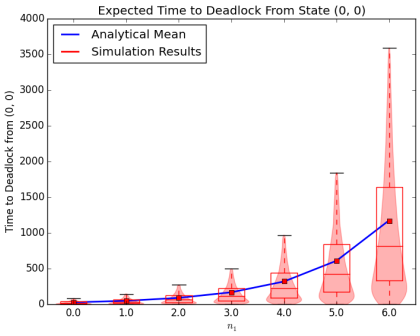
(d) Varying r_{21}



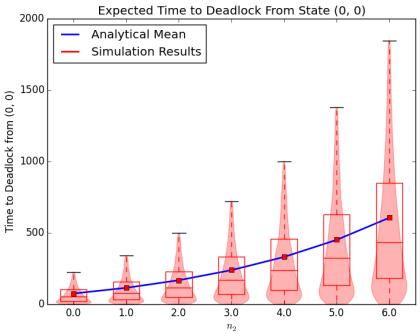
(e) Varying μ_1



(f) Varying μ_2



(g) Varying n_1



(h) Varying n_2

Figure 9: Time to deadlock in Ω_2, analytical & simulation results (10,000 iterations).

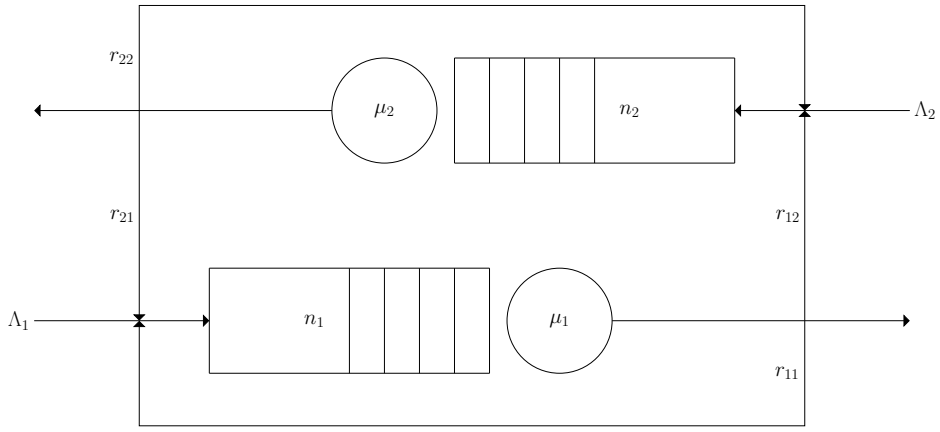


Figure 10: An open two node single-server restricted queueing network.

$$q_{(i_1,j_1),(-1)} = \begin{cases} r_{11}\mu_1 & \text{if } i > n_1 \text{ and } j < n_2 + 2 \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

$$q_{(i_1,j_1),(-2)} = \begin{cases} r_{22}\mu_2 & \text{if } j > n_2 \text{ and } i < n_1 + 2 \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

$$q_{(i_1,j_1),(-3)} = \begin{cases} r_{21}\mu_2 & \text{if } (i,j) = (n_1, n_2 + 2) \\ r_{12}\mu_1 & \text{if } (i,j) = (n_1 + 2, n_2) \\ 0 & \text{otherwise} \end{cases} \tag{11}$$

$$q_{-1,s} = 0 \tag{12}$$

$$q_{-2,s} = 0 \tag{13}$$

$$q_{-3,s} = 0 \tag{14}$$

Note that the there are only two differences between this formulation and the formulation given in Subsection 5.2: the rates at which customers leave nodes 1 and 2 are now $(1 - r_{11} - r_{12})\mu_1$ and $(1 - r_{21} - r_{22})\mu_2$; and there are now two more ways to reach deadlock, Equation 9 and Equation 10.

For $n_1 = 1$ and $n_2 = 2$, the resulting Markov chain is shown in Figure 11.

Figure 12 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 $r_{22} = 0.1$ are used.

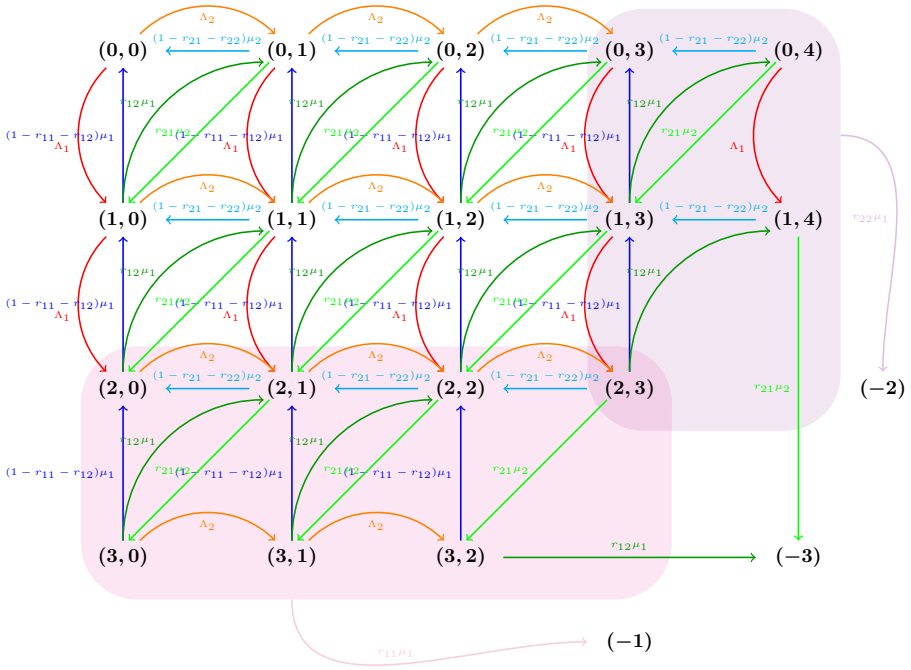


Figure 11: Diagrammatic representation of the Markov chain for Ω with $n_1 = 1$ and $n_2 = 2$.

5.4 One Node Multi-Server

Consider the open one node multi-server restricted queueing network shown in Figure 13. This is an Ω_1 system with c parallel servers.

State space:

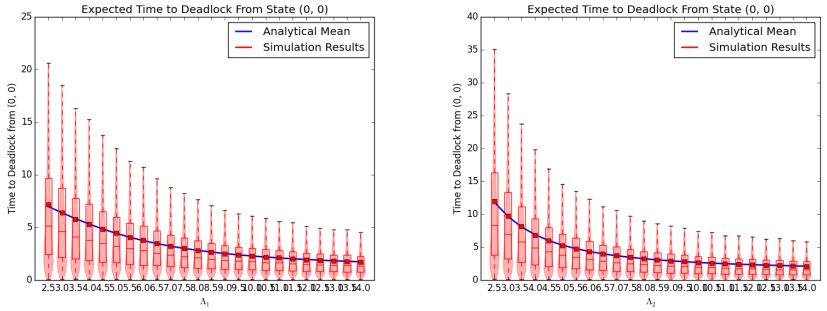
$$S = \{i \in \mathbb{N} \mid 0 \leq i \leq n + 2c\}$$

where i denotes the number of individuals in the system plus the number of individuals blocked who are blocked. 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.

Define $\delta = i_2 - i_1$ for all $i_k \in S$. The transitions are given by:

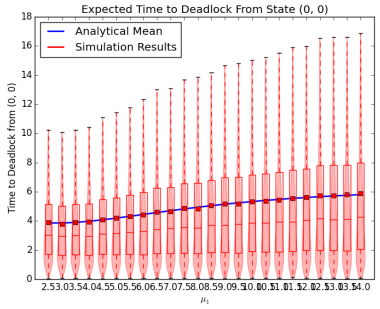
$$q_{i_1,i_2} = \left\{ \begin{array}{rl} \Lambda & \text{if } \delta = 1 \\ (1 - r_{11})\mu \min(i, c) & \text{if } \delta = -1 \\ 0 & \text{otherwise} \end{array} \right\} \quad \text{if } i_1 < n + c \tag{15}$$

$$q_{i_1,i_2} = \left\{ \begin{array}{rl} (c - b)r_{11}\mu & \text{if } \delta = 1 \\ (1 - r_{11})(b - k)\mu & \text{if } \delta = -b - 1 \\ 0 & \text{otherwise} \end{array} \right\} \quad \text{if } i_1 = n + c + b \quad \forall \quad 0 \leq b \leq c \tag{16}$$

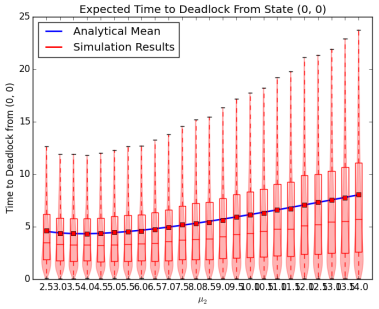


(a) Varying Λ_1

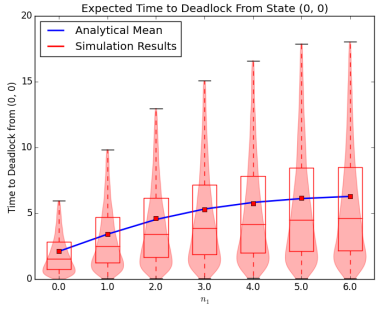
(b) Varying Λ_2



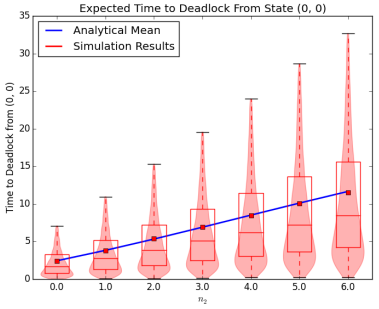
(c) Varying μ_1



(d) Varying μ_2



(e) Varying n_1



(f) Varying n_2

Figure 12: Time to deadlock in Ω, analytical & simulation results (10,000 iterations).

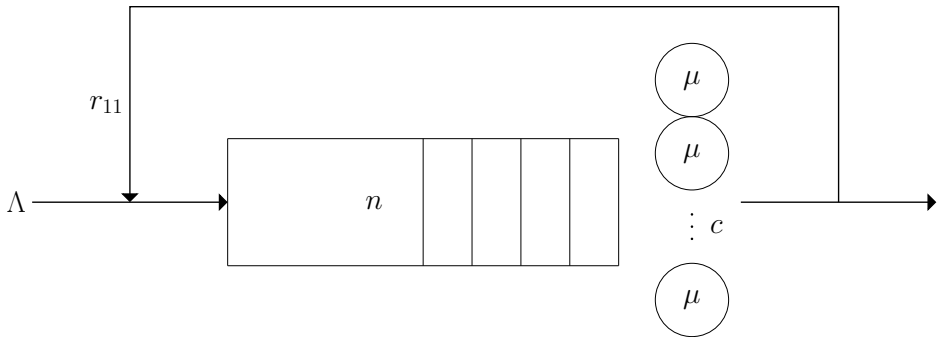


Figure 13: An open one node multi-server restricted queueing network.

where b denotes the number of blocked customers. The Markov chain is shown in Figure 14.

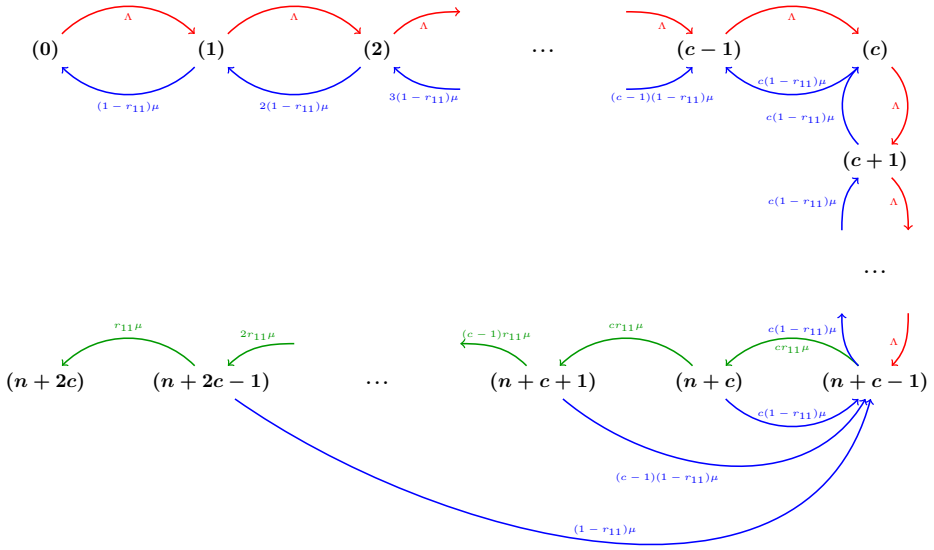


Figure 14: Diagrammatic representation of the Markov chain for a multi-server Ω_1 system.

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 ($i \in [n+c+1, i = n+2c]$) 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 15 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.

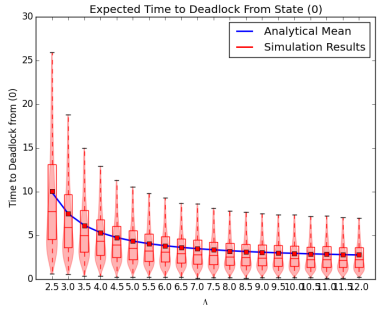
5.5 Two Node Multi-Server without Self-Loops

Consider the open two node mutli server restricted queueing network shown in Figure 16. This is an Ω_2 system with c_i parallel servers at node i.

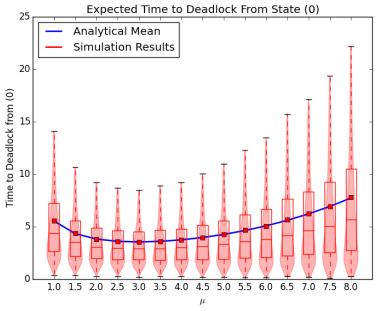
State space:

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

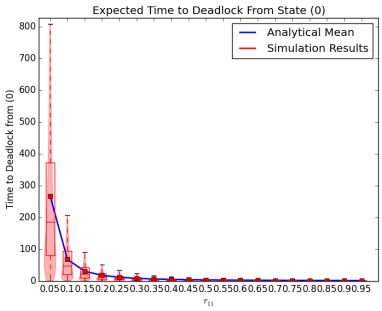
where i denotes the number of individuals at Node 1 plus the number of individuals blocked waiting to enter Node 1, and j denotes the number of individuals at Node 2 plus the number of individuals blocked waiting to enter Node 2. For example, $(i, j) = (n_1 + c_1 + 2, n_2 + c_2 + 1)$ denotes a full system, $n_1 + c_1$ 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.



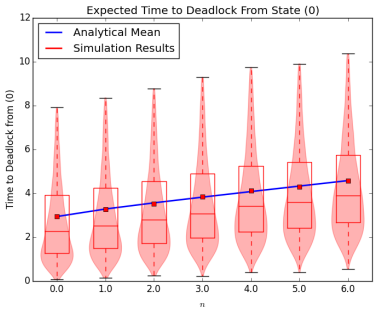
(a) Varying Λ



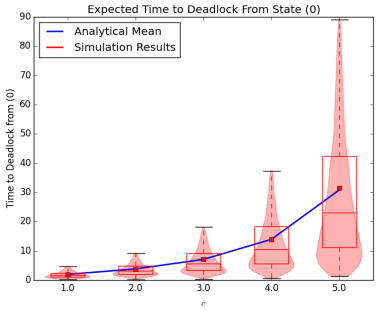
(b) Varying μ



(c) Varying r_{11}



(d) Varying n



(e) Varying c

Figure 15: Time to deadlock in multi-server Ω_1, analytical & simulation results (10,000 iterations).

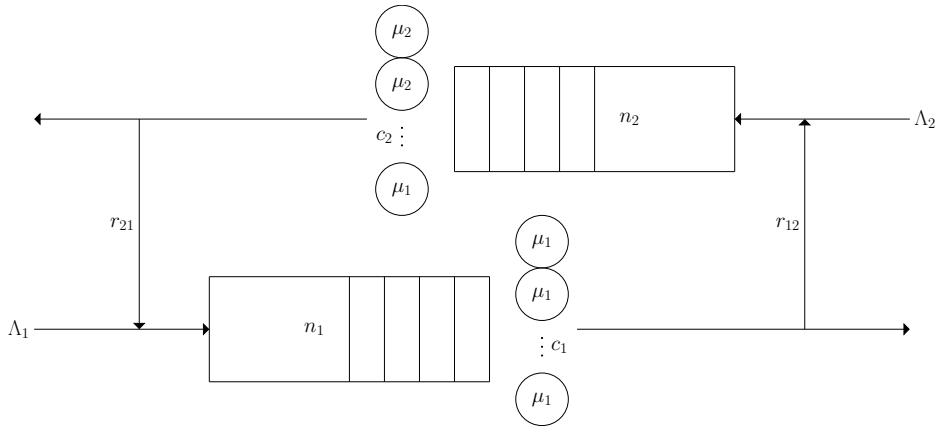


Figure 16: An open two node mutli server restricted queueing network.

	$j_1 < n_2 + c_2$	$j_1 = n_2 + c_2$	$j_1 > n_2 + c_2$
$i_1 < n_1 + c_1$	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 $q_{(i_1,j_1),(i_2,j_2)}$ for a multi-server two node network.

The Markov chain is shown in Figure 17.

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 $s_2 = \min(i_2, c_2) - b_1$ for all $(i_k, j_k) \in S$. 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 18 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.

6 A Bound on the Time to Deadlock

This section gives a bound on the mean time to deadlock of the Ω system. In order to do this, six deadlocking queueing networks are defined as follows, all single-server:

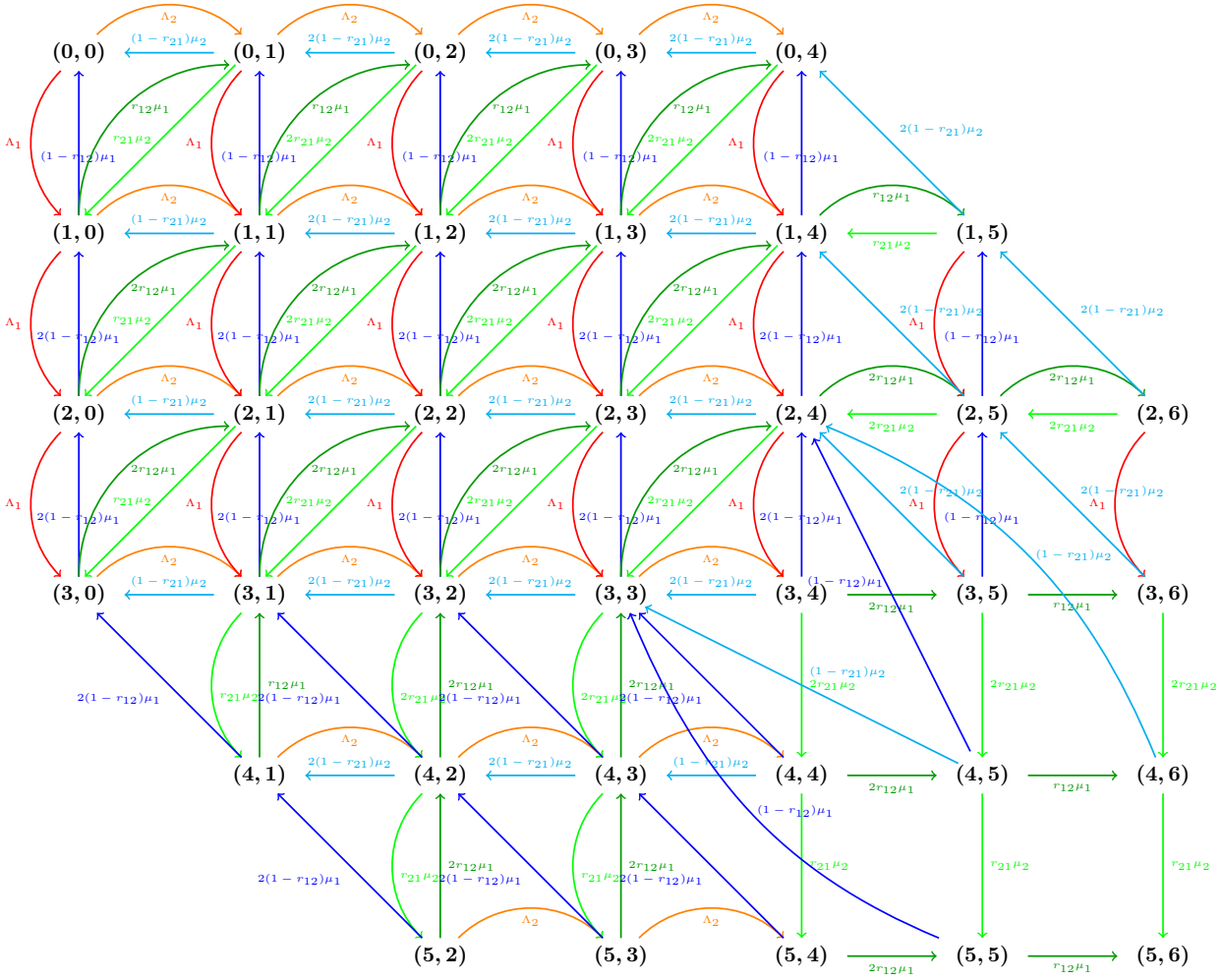
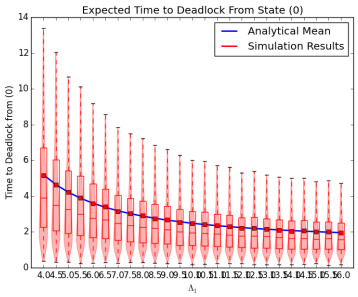
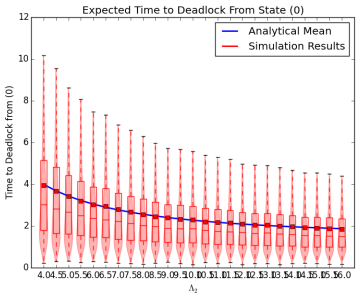


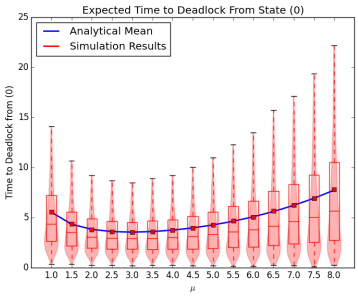
Figure 17: Diagrammatic representation of the Markov chain for a multi-server Ω_2 system with $n_1 = 1$, $n_2 = c_1 = c_2 = 2$. The deadlocked state is $(5, 6)$.



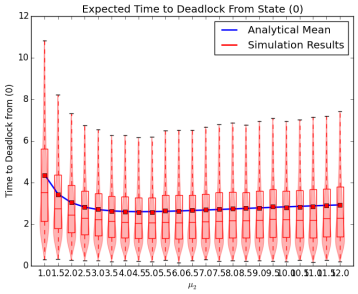
(a) Varying Λ_1



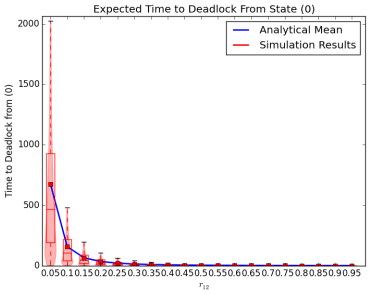
(b) Varying $\Lambda_@$



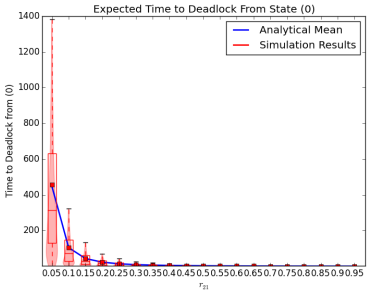
(c) Varying μ_1



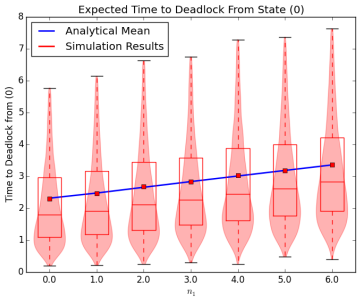
(d) Varying μ_2



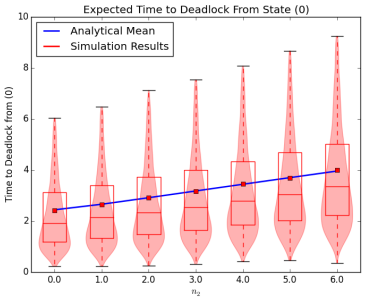
(e) Varying r_{12}



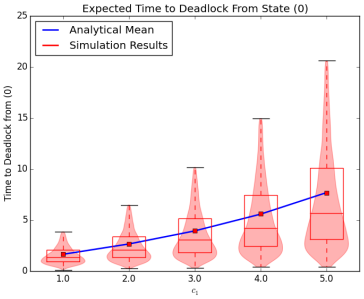
(f) Varying r_{21}



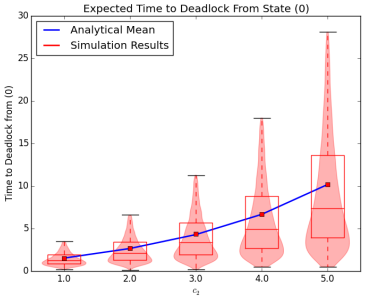
(g) Varying n_1



(h) Varying n_2



(i) Varying c_1



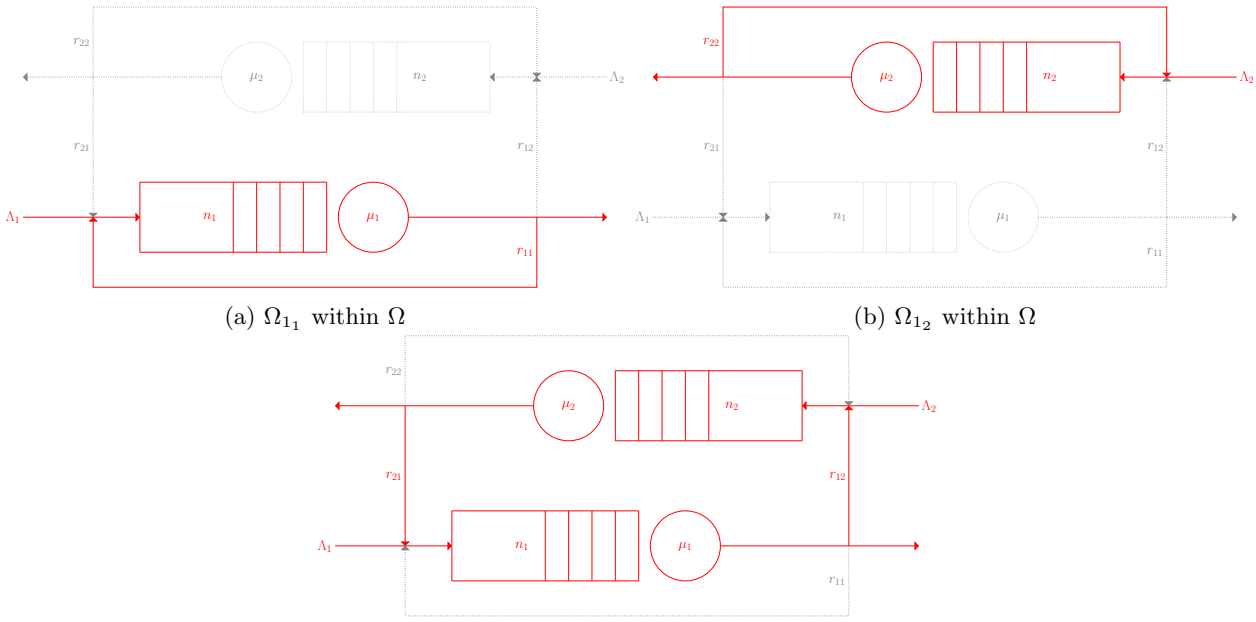
(j) Varying c_2

Figure 18: Time to deadlock in multi-server Ω_2, analytical & simulation results (10,000 iterations).

- Define $\Omega^{\star}_{1_1}$ as the 1 node queueing network described in Subsection 5.1 with the parameter set (Λ_1, μ_1, n_1, r_{11}). Let its mean time to deadlock be denoted by $\omega^{\star}_{1_1}$.
- Define $\Omega^{\star\star}_{1_1}$ as the 1 node queueing network described in Subsection 5.1 with the parameter set (Λ_1, m_1, n_1, r_{11}). Let its mean time to deadlock be denoted by $\omega^{\star\star}_{1_1}$.
- Define $\Omega^{\star}_{1_2}$ as the 1 node queueing network described in Subsection 5.1 with the parameter set (Λ_2, μ_2, n_2, r_{22}). Let its mean time to deadlock be denoted by $\omega^{\star}_{1_2}$.
- Define $\Omega^{\star\star}_{1_2}$ as the 1 node queueing network described in Subsection 5.1 with the parameter set (Λ_2, m_2, n_2, r_{22}). Let its mean time to deadlock be denoted by $\omega^{\star\star}_{1_2}$.
- Define Ω_2 as the 2 node queueing network described in Subsection 5.2 with the parameter set (Λ_1, Λ_2, μ_1, μ_2, n_1, n_2, r_{12}, r_{21}). Let its mean time to deadlock be denoted by ω_2.
- Define Ω as the 2 node queueing network described in Subsection 5.3 with the parameter set (Λ_1, Λ_2, μ_1, μ_2, n_1, n_2, r_{11}, r_{12}, r_{21}, r_{22}). Let its mean time to deadlock be denoted by ω.

where $m_1 = \frac{\mu_2}{3+2\frac{\mu_2}{\mu_1}+\frac{\mu_1}{\mu_2}}$, and $m_2 = \frac{\mu_1}{3+2\frac{\mu_1}{\mu_2}+\frac{\mu_2}{\mu_1}}$.

Figure 19 shows how Ω contains, and is made up by, Ω_{1_1}, Ω_{1_2} and Ω_2.



(a) Ω_{1_1} within Ω

(b) Ω_{1_2} within Ω

(c) Ω_2 within Ω

Figure 19: Decomposition of Ω into Ω_{1_1}, Ω_{1_2} and Ω_2

Defining $\omega_{1_j} = \max(\omega^{\star}_{1_j}, \omega^{\star\star}_{1_j})$ for $j \in [1, 2]$, we get the following bound:

Theorem 2. *For any parameter sets the following inequality holds:* $\omega \leq \min(\omega_{1_1}, \omega_{1_2}, \omega_2)$

Proof. First, define the following systems:

- Let $\widetilde{\Omega}_{1_1}$ denote the Ω_{1_1} system embedded within Ω. Let $\widetilde{\omega}_{1_1}$ denote the mean time to deadlock of $\widetilde{\Omega}_{1_1}$.
- Let $\widetilde{\Omega}_{1_2}$ denote the Ω_{1_2} system embedded within Ω. Let $\widetilde{\omega}_{1_2}$ denote the mean time to deadlock of $\widetilde{\Omega}_{1_2}$.
- Let $\widetilde{\Omega}_2$ denote the Ω_2 system embedded within Ω. Let $\widetilde{\omega}_2$ denote the mean time to deadlock of $\widetilde{\Omega}_2$.

It follows that $\widetilde{\omega}_{1_1}$ is the mean time to state (-1) in Ω, $\widetilde{\omega}_{1_2}$ is the mean time to state (-2) in Ω and $\widetilde{\omega}_2$ is the mean time to state (-3) in Ω. Therefore $\omega = \min(\widetilde{\omega}_{1_1}, \widetilde{\omega}_{1_2}, \widetilde{\omega}_2)$, as the mean time to deadlock in Ω is the expected time it takes to reach either (-1), (-2) or (-3).

This proof compares each embedded system with its respective non-embedded counterpart.

1. Consider $\widetilde{\Omega}_2$. The effective arrival rate to Node 1 in $\widetilde{\Omega}_2$ is greater than or equal to the effective arrival rate to Node 1 in Ω_2, due to the extra customers who are rejoining the queue after service. Similarly the effective arrival rate to Node 2 in $\widetilde{\Omega}_2$ is greater than or equal to the effective arrival rate to Node 2 in Ω_2. As an increase in the arrival rate causes the mean time to deadlock to decrease, we can conclude $\widetilde{\omega}_2 \leq \omega_2$
2. Consider $\widetilde{\Omega}_{1_1}$. Both the service rate and arrival rate differ in the embedded system to the non-embedded system.
   - Consider the expected effective service time, the time that $\widetilde{\Omega}_{1_1}$'s state does not change due to services or outside factors.
     - The lower bound on the effective service time is $\frac{1}{\mu_1}$, corresponding to when neither Node 1 nor Node 2 are full. Therefore the upper bound on the effective service rate is μ_1.
     - The upper bound on the effective service time corrseponds to the following cycle of events:
       * A customer, a, begins service at Node 1: $\frac{1}{\mu_1}$
       * Customer a finishes service at Node 1, but is blocked from transitioning to Node 2: $\frac{1}{\mu_2}$
       * A customer, b, finishes service at Node 2 and transitions to Node 1. Now the a moves to Node 2, and another the customer begins service at Node 1: $\frac{1}{\mu_1}$

       This cycle is repeated with probability P_{repeat}. As all rates are Markovian then whatever point in b's service a gets blocked, a's expected wait is $\frac{1}{\mu_2}$. Therefore the upper bound for the effective service time is:

$$\begin{aligned}
\frac{1}{m_1} &= \frac{2}{\mu_1} + \frac{1}{\mu_2} + P_{\text{repeat}}\left(\frac{2}{\mu_1} + \frac{1}{\mu_2} + P_{\text{repeat}}\left(\frac{2}{\mu_1} + \frac{1}{\mu_2} + P_{\text{repeat}}\Big(\cdots\right.\right.\right. \\
&= \left(\frac{2}{\mu_1} + \frac{1}{\mu_2}\right) \times \left(1 + P_{\text{repeat}} + P_{\text{repeat}}^2 + P_{\text{repeat}}^3 + \dots\right) \\
&= \left(\frac{2}{\mu_1} + \frac{1}{\mu_2}\right) \times \left(\frac{1}{1 - P_{\text{repeat}}}\right)
\end{aligned}$$

       If S_1 is the time a spends in service, and S_2 is the time b spends in service, then $S_1 \sim \text{Exp}(\mu_1)$ and $S_2 \sim \text{Exp}(\mu_2)$. Now $P_{\text{repeat}} = P(S_1 < S_2) = \frac{\mu_1}{\mu_1 + \mu_2}$.

Therefore the lower bound on the effective service rate is:

$$\begin{aligned} m_1 &= \frac{1}{\left(\frac{2}{\mu_1}+\frac{1}{\mu_2}\right)\left(\frac{1}{1-P_{\text{repeat}}}\right)} \\ &= \frac{1}{\left(\frac{2}{\mu_1}+\frac{1}{\mu_2}\right)\left(\frac{1}{1-\frac{\mu_1}{\mu_1+\mu_2}}\right)} \\ &= \frac{1}{\left(\frac{2}{\mu_1}+\frac{1}{\mu_2}\right)\left(\frac{\mu_1+\mu_2}{\mu_2}\right)} \\ &= \frac{\mu_2}{\left(\frac{2}{\mu_1}+\frac{1}{\mu_2}\right)(\mu_1+\mu_2)} \\ &= \frac{\mu_2}{3+2\frac{\mu_2}{\mu_1}+\frac{\mu_1}{\mu_2}} \end{aligned}$$

Now the actual effective service rate $\widetilde{\mu}_1$ for $\widetilde{\Omega}_{1_1}$ must lie in the interval (μ_1, m_1). Using Remarks 1 and 2, we can conclude that $\widetilde{\omega}_{1_1} \leq \max(\omega^{\star}_{1_1}, \omega^{\star\star}_{1_1})$ when all other parameters are fixed.

- Consider the effective arrival rate of $\widetilde{\Omega}_{1_1}$. The effective arrival rate in $\widetilde{\Omega}_{1_1}$ is greater than or equal to the effective arrival rate in an Ω_{1_1} system; this is due to the extra customers who have transitioned from the Node 2 to Node 1. An increase in the arrival rate causes the mean time to deadlock to decrease.

Combining the above, it is concluded that $\widetilde{\omega}_{1_1} \leq \omega_{1_1}$.

3. A similar argument yields $\widetilde{\omega}_{1_2} \leq \omega_{1_2}$.

Therefore

$$\begin{aligned} \min(\widetilde{\omega}_{1_1}, \widetilde{\omega}_{1_2}, \widetilde{\omega}_2) &\leq \min(\omega_{1_1}, \omega_{1_2}, \omega_2) \\ \omega &\leq \min(\omega_{1_1}, \omega_{1_2}, \omega_2) \end{aligned}$$

□

This bound has been tested for 7.2 mllion parameter sets.

7 Conclusions

This paper, motivated by a gap in the literature, has explored deadlock in open restricted queueing networks. It has been shown that analysing a queueing network's corresponding state digraph is sufficient to detect

when deadlock occurs in queueing network systems. In general the presence of a knot in the state digraph will highlight that deadlock has occured in the network, however for special cases only the presence of a weakly connected component with no sink is required. Encorporating this into a simulation model, time to deadlock could be recorded.

Markov models of five deadlocking queueing networks were built. Using linear algebraic techniques the expected time to deadlock from each state was found, and its behaviour as system parameters varied explored. These analytical results were compared with results obtained from the simulation model. Finally a bound on the expected time to deadlock for one of these models was given.

References

[1] I. Akyildiz. Product form approximations for queueing networks with multiple servers and blocking. *IEEE Transactions on Computers*, 38(1):99–114, 1989.

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

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

[4] F. Belik. An efficient deadlock avoidance technique. *IEEE Transactions on Computers*, 39(7):882–888, 1990.

[5] 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.

[6] 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.

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

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

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

[10] J. Ezpeleta, F. Tricas, F. Garca-Valls, and J.M. Colom. A banker's solution for deadlock avoidance in fms with flexible routing and multiresource states. *IEEE Transactions on Robotics and Automation*, 18(4):621–625, 2002.

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

[12] 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.

[13] 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.

[14] 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.

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

[16] 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.

[17] 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.

[18] R. Onvural. Survey of closed queueing networks with blocking. *ACM Computing Surveys*, 22(2):83–121, 1990.

[19] H.G. Perros, A.A. Nilsson, and Y.C. Liu. Approximate analysis of product-form type queueing networks with blocking and deadlock. *Performance Evaluation*, 8(1):19–39, 1988.

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

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

[22] S.T. Tan. *Calculus.* Brooks/Cole/Cengage Learning, 2009.

[23] N. Viswanadham, Y. Narahari, and T.L. Johnson. Deadlock prevention and deadlock avoidance in flexible manufacturing systems using petri net models. *IEEE Transactions on Robotics and Automation*, 6(6):713–723, 1990.