Detecting Deadlock in Queueing Network Simulations

Geraint Palmer

1 Explanation of Deadlock

Any open queueing network with feedback loops, at least one service station that has limited queueing capacity, and where individuals can be blocked from joining the queue at the next destination can experience deadlock. Deadlock occurs when a customer finishes service at node i and is blocked from transitioning to node j; however the individuals in node j are all blocked, directly or indirectly, by the blocked individual in node i. NEED A LOT MORE HERE. DIAGRAMS TO EXPLAIN TOO.

2 Literature Review

Most of the literature on blocking conveniently assumes the networks are deadlock-free. For closed networks of K customers with only one class of customer, [1] 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 [2], 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.
NEED A LOT MORE HERE

3 Definitions

$\|V(D)\|$	The order of the directed graph D is its number of vertices.
Weakly connected component	A weakly connected component of a digraph containing X is the set of all nodes that can be reached from X if we ignore the direction of the edges.
Direct successor	If a directed graph contains an edge from X_i to X_j , then we say that X_j is a direct successor of X_i.
Ancestors	If a directed graph contains a path from X_i to X_j , then we say that X_i is an ancestor of X_j.
Decendents	If a directed graph contains a path from X_i to X_j , then we say that X_j is a descendant of X_i.
$deg^{out}(X)$	The out-degree of X is the number of outgoing edges emanating from that vertex.

NEED CONSISTANT NOTATION, REFERENCES FOR DEFINITIONS.

4 Explaining Digraph

Here is a method of detecting when deadlock occurs in an open queueing network Q with N nodes. Let the number of servers in node i be denoted by c_i.
Define D as a directed graph associated with Q, where $|V(D)| \leq \sum_{i=1}^{N} c_i$. Each vertex of D is an individual who is either in service or blocked.
When an individual finishes service at node i, and this individuals next destination is node j, but there is not enough queueing capacity for j to accept that individual, then that individual remains at node i and becomes blocked. At this point c_j directed edges between this individual and each individual in service or blocked at node j are created in D.
Once an individual is released, that individual is removed as a vertex of D, and all edges in and out of that vertex removed. If another individual enters service at the same time, then that next individual node, and aquires the previous individual's in-edges.
CLEARER, VERTICES SHOULD BE SERVERS THAT CAN BE OCCUPIED. LOTS OF DIAGRAMS. SUPGRAPHS THING.

5 Observations

Consider one weakly connected component of D, G. Consider the node $X_a \in G$, where X_a's next destination is node j. Then X_a's direct successors are the individuals who are blocked or in service at node j. We can interpret all X_a's decendents as those individuals who are directly or indirectly blocking X_a, and we can interpret all X_a's ancestors as those individuals who are being blocked directly or indirectly by X_a.
We do not need to worry about the situation $deg^{out}(X_a) > c_j$ as it will never occur because we only create edges from X_a to individuals in service or blocked in node j. If $deg^{out}(X_a) = c_j$ then X_a is blocked by all its direct successors. The only other situation is that X_a is not blocked, and belongs to G because X_a is in service and blocking other individuals, in which case $deg^{out}(X_a) = 0$.

It is clear that if any of X_a's decendents are not blocked, then we do not have deadlock, and if all of X_a's descendents are blocked, then we have deadlock. We also know that by definition all of X_a's ancestors are blocked. It is therefore also clear that, for any individual X, if all of X's descendants are blocked, then the descendents of every node in G are blocked. Therefore to check that we haven't got deadlock, we only need to check that there is at least one node of G that is not blocked; hence checking for a node in G that has $deg^{out} = 0$.

6 Theorem

Theorem

A deadlock situation arises if and only if there exists a weakly connected component of D containing no vertices with $deg^{out} = 0$.

Observations

Consider one weakly connected component G of D. Consider the node $X_a \in G$, where X_a's next destination is node j. Then X_a's direct successors are the individuals who are blocked or in service at node j. We can interpret all X_a's decendents as those individuals who are directly or indirectly blocking X_a, and we can interpret all X_a's ancestors as those individuals who are being blocked directly or indirectly by X_a.
We do not need to worry about the situation $deg^{out}(X_a) > c_j$ as it will never occur because we only create edges from X_a to individuals in service or blocked in node j. If $deg^{out}(X_a) = c_j$ then X_a is blocked by all its direct successors. The only other situation is that X_a is not blocked, and belongs to G because X_a is in service and blocking other individuals, in which case $deg^{out}(X_a) = 0$.
It is clear that if any of X_a's decendents are not blocked, then we do not have deadlock, and if all of X_a's descendents are blocked, then we have deadlock. We also know that by definition all of X_a's ancestors are blocked.

Proof

Consider one weakly connected component G of D.
Assume that G contains a vertex X such that $deg^{out}(X) = 0$. This imples that X is not blocked, therefore Q is not deadlocked.
Now assume that we have deadlock, and all of the members of G are blocked. For a given vertex X, all decendents of X are blocked, and so must have out-degrees greater than 0. However, as our choice of X was arbitrary, this must be true for all members of G, and no members of G have out-degree of 0.

References

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

[2] 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.