Deadlock properties of queueing networks with finite capacities and multiple routing chains*

Jörg Liebeherr[a] and Ian F. Akyildiz[b]

[a]*Department of Computer Science, University of Virginia, Charlottesville, VA 22903, USA*

[b]*School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332, USA*

Received 12 May 1994; revised 19 April 1995

Blocking in queueing network models with finite capacities can lead to deadlock situations. In this paper, deadlock properties are investigated in queueing networks with multiple routing chains. The necessary and sufficient conditions for deadlock-free queueing networks with blocking are provided. An optimization algorithm is presented for finding deadlock-free capacity assignments with the least total capacity. The optimization algorithm maps the queueing network into a directed graph and obtains the deadlock freedom conditions from a specified subset of cycles in the directed graph. In certain network topologies, the number of deadlock freedom conditions can be large, thus, making any optimization computationally expensive. For a special class of topologies, so-called *tandem networks*, it is shown that a minimal capacity assignment can be directly obtained without running an optimization algorithm. Here, the solution to the minimal capacity assignment takes advantage of the regular topology of tandem networks.

Keywords: Queueing networks, blocking networks, deadlocks, deadlock prevention.

1. Introduction

Queueing network models are frequently applied for performance evaluation of computer systems and communication networks. Numerous methods are available for analyzing queueing networks under the assumption that all stations have infinite capacities. However, in actual systems the resources have finite capacities, and queueing networks with finite capacities should be used for performance analysis.

In a queueing network with finite capacities, each station has only a finite waiting room for buffering jobs. Blocking arises due to the limitations imposed by the capacity of these stations. In particular, blocking occurs when the flow of jobs through one station is interrupted due to another station that has reached its full capacity. The set of rules that dictate when a station becomes blocked or unblocked is commonly referred as the blocking mechanism. In this work, we consider the so-called *blocking-after-service* or BAS mechanism, also referred to

* This work was supported by the National Science Foundation under Grant No. CCR-90-11981.

as Type 1 or manufacturing blocking mechanism [1]. In BAS, a job which has completed service at a station i and attempts to proceed to station j must find an empty buffer space in station j. If station j is full, the job is blocked and forced to wait in station i's server until it can enter destination station j. A server which contains a blocked job cannot serve other jobs waiting in the queue.

Finite station capacities and blocking can lead to a deadlock situation in the queueing network. As an example, suppose a job has finished service at a station, say station 1, and wants to proceed to some other station, say station 2. If the waiting room of station 2 is full, the job is blocked in the server of station 1. Suppose another job has finished service at station 2 and has selected station 1 as its next station. If station 1's waiting room is also full, this job is blocked at station 1. In this situation, the jobs in the servers of both stations 1 and 2 are permanently blocked. As a result, a deadlock situation arises.

There are two approaches to solve deadlock problems in finite capacity queueing networks. First, one can extend the blocking mechanism by providing additional algorithms that dynamically resolve a deadlock situation. For example, some deadlocks can be resolved by allowing blocked jobs to select an alternate destination station. Note however, that adding a deadlock resolution mechanism significantly increases the complexity of the queueing network model, and, as a result, may render an analytical solution of the model intractable. Second, one can select the waiting room at each station sufficiently large such that deadlocks cannot occur. This solution requires knowledge of so-called *deadlock freedom conditions*, that is, conditions on the size of the waiting room of the stations which prevent deadlock situations. An advantage of the second approach is that it does not involve changes to the blocking mechanism.

In this study, we take a preventive approach to deadlocks and derive *deadlock freedom conditions* for finite capacity queueing networks with BAS blocking and multiple routing chains. For queueing networks with a single routing chain, Kundu and Akyildiz [3] proved that a network is deadlock-free if the number of jobs in the network is less than the capacity of the directed cycle with minimal waiting room. However, these conditions for deadlock freedom cannot be straightforwardly extended to networks with multiple routing chains since a deadlock may result from dependencies between jobs from different routing chains.

We demonstrate these dependencies in the network model shown in fig. 1. The figure depicts a queueing network with two stations and two routing chains. We denote by B_{ir} the waiting room, referred to as *buffer*, of chain r at station i and by K_r the total number of jobs in chain r. Let the parameters be given by

$$K_1 = 5; \qquad K_2 = 6;$$

$$\text{Capacity of } B_{11} = 3; \quad \text{Capacity of } B_{21} = 3;$$

$$\text{Capacity of } B_{12} = 4; \quad \text{Capacity of } B_{22} = 3.$$

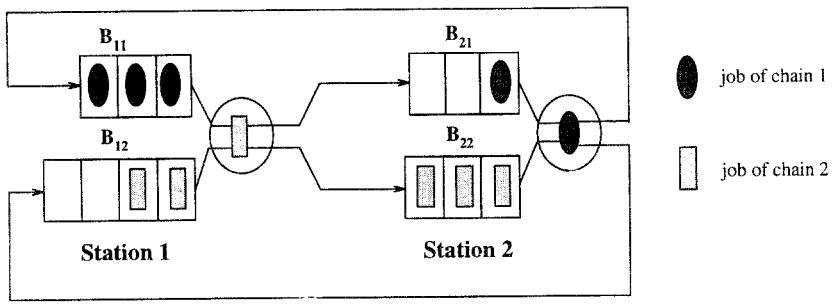

Fig. 1. Blocking network with 2 chains.

In fig. 1, the job from chain 2 residing in the server of station 1 cannot proceed to the full buffer B_{22}. Thus, the job waits in the server until space in B_{22} becomes available. On the other hand, the job in the server of station 2 cannot enter buffer B_{11}, since B_{11} is full. Thus, a deadlock situation occurs, even though the conditions for deadlock freedom given by Kundu and Akyildiz [3] are satisfied for each routing chain in isolation.

We show that deadlock situations in multiple-chain queueing networks always occur in so-called *buffer cycles*. A buffer cycle is a cyclic sequence of buffers in the networks where the buffers may belong to different routing chains. The set of feasible buffer cycles is obtained from the transition probability matrices of the routing chains. We show that a queueing network with multiple routing chains is deadlock-free if and only if each buffer cycle is deadlock-free.

Once the deadlock freedom conditions are available they can be applied to find an assignment of capacities to the buffers of each station such that deadlocks cannot occur. Of particular interest are capacity assignments which yield a deadlock-free network with the least total capacity. We refer to these assignments as *minimal capacity assignments*. We present an optimization algorithm which yields a minimal capacity assignment for multiple-chain queueing networks with arbitrary topology. The set of buffer cycles is obtained by mapping the queueing network into a directed graph such that each cycle in the graph corresponds to a buffer cycle in the queueing network. The minimal capacity assignment is obtained with standard linear optimization techniques.

A potential drawback of the optimization algorithm is its computational complexity which makes the algorithm impractical for networks with a large number of buffer cycles. As a worst case, we consider so-called *tandem networks*, that is, networks where all stations are connected in a sequence. In tandem networks with N stations and R routing chains, the number of buffer cycles is given by R^N. By taking advantage of the regular topology in a tandem network we can provide a minimal capacity assignment without running an optimization algorithm.

The remaining sections of this study are structured as follows. In section 2 we describe the class of queueing models which is considered in this study. The

conditions for deadlock freedom are stated and proved in section 3. In section 4 we define a capacity assignment to be minimal if it achieves deadlock freedom with the least total capacity. Then we present an optimization algorithm which generates a minimal capacity assignment for arbitrary network topologies. In section 5 we show that a minimal capacity assignment can be directly given if the network has a so-called *tandem topology*. In section 6 we conclude our results.

2. Model description

We consider a closed queueing network $\Gamma = (\mathcal{N}, \mathcal{R}, \mathcal{P})$ with the following properties:

- The network contains a finite set $\mathcal{N}$ of stations and a finite set $\mathcal{R}$ of disjoint routing chains. Each job in the network belongs to exactly one routing chain. Let $\mathcal{R}_i \subseteq \mathcal{R}$ denote the set of routing chains whose jobs visit station i. Let $\mathcal{N}_r \subseteq \mathcal{N}$ denote the set of stations visited by jobs from routing chain r. $\mathcal{P}$ is a set of matrices $\mathcal{P} = (P_1, P_2, \ldots, P_{|\mathcal{R}|})$ where P_r is the $|\mathcal{N}| \times |\mathcal{N}|$ transition probability matrix for routing chain r. The elements of P_r are denoted by $p_{ij,r}$ with the following interpretation. A job of routing chain r which has received service by station i proceeds to station j with probability $p_{ij,r}$. Throughout the paper, we assume that $p_{ij,r} > 0$ implies that $r \in \mathcal{R}_i$ and $r \in \mathcal{R}_j$.
- Each station i has a single server. The service times for jobs at station i are i.i.d. and given by a random variable Y_i. We assume that the service times are such that for all times $t > 0$ we have $Prob(Y_i > t) > 0$. The scheduling discipline of a station is arbitrary, but non-preemptive.
- Each station keeps separate buffers for jobs from different routing chains. B_{ir} denotes the buffer at station i (excluding the server) for jobs from routing chain r. Each buffer may accommodate only a finite number of jobs. Let Φ be an assignment of capacities to the buffers of Γ, i.e.,

$$\Phi : \{B_{ir} \mid i \in \mathcal{N}, r \in \mathcal{R}_i\} \rightarrow \{0, 1, 2, \ldots\}. \tag{1}$$

  A buffer can have infinite capacity, i.e., $\Phi(B_{ir}) = \infty$, or no capacity at all, i.e., $\Phi(B_{ir}) = 0$. The total capacity of station i is computed by $\sum_{r \in \mathcal{R}_i} \Phi(B_{ir}) + 1$.
- The jobs from chain r in Γ is fixed and given by K_r. The total number of jobs in the network is denoted by $K = \sum_{r \in \mathcal{R}} K_r$.
- The number chain-r jobs in buffer B_{jr} cannot exceed its capacity $\Phi(B_{jr})$. Assume a job of chain r has completed service at some station i and wants to proceed to a station j. If B_{jr} is *saturated*, i.e., $\Phi(B_{jr})$ jobs are waiting in B_{ir}, the job is *blocked* at buffer B_{jr} and must reside in the server of station i until a place in B_{jr} becomes available. A server which contains a blocked

job cannot serve other jobs. The described blocking mechanism is referred to as blocking-after-service or BAS.

3. Conditions for deadlock freedom

In this section we present the deadlock freedom conditions for a queueing network as described in section 2. The conditions consider a *worst case* scenario in the sense that any network will eventually be deadlocked even if the probability to enter a deadlock state is very small. The theorem for deadlock freedom (DLF theorem) is stated in terms of conditions that must hold for each *buffer cycle* in the network. A buffer cycle is defined as a cyclic sequence of buffers such that each buffer in the sequence belongs to a station which can have a blocked job at the next buffer of the sequence. A formal definition of a buffer cycle is given as follows.

DEFINITION 1

A *buffer cycle* is a sequence of buffers $C = (B_{i_1 r_1}, B_{i_2 r_2}, \ldots, B_{i_{M-1} r_{M-1}}, B_{i_M r_M})$ such that

$$\begin{aligned} &p_{i_m i_{m+1}, r_{m+1}} > 0 \quad \text{and} \quad r_m \in \mathcal{R}_{i_m} \quad \text{for all } 1 \leq m < M, \\ &p_{i_M i_1, r_1} > 0 \qquad \text{and} \quad r_M \in \mathcal{R}_{i_M}. \end{aligned} \tag{2}$$

Let $\mathcal{C}$ denote the set of all buffer cycles in Γ.

The next definition provides the set of stations in buffer cycle C that may contain a job from routing chain r which is blocked at a buffer in the same cycle.

DEFINITION 2

The set of stations with buffers from routing chain r in buffer cycle C is defined by

$$S_r^{(C)} = \{i \mid B_{ir} \in C\}. \tag{3}$$

For example, $C_1 = (B_{11}, B_{21})$ and $C_2 = (B_{11}, B_{22})$ are two buffer cycles in the network shown in fig. 1. From definition 2 we obtain the sets of stations with potentially blocked jobs to be $S_1^{(C_1)} = \{1,2\}, S_2^{(C_1)} = \emptyset$ for cycle C_1, and $S_1^{(C_2)} = \{1\}, S_2^{(C_2)} = \{2\}$ for cycle C_2.

Next we state the conditions for deadlock freedom in a queueing network with multiple routing chains. Theorem 1 states that for each buffer cycle C there must exist at least one routing chain r which, at the same time, cannot saturate

all its buffers in cycle C and have $S_r^{(C)}$ jobs blocked at some other buffers in the cycle.

THEOREM 1 (DLF THEOREM)

A multiple chain queueing network Γ is *deadlock-free* if and only if for all $C \in \mathcal{C}$ there exists a routing chain $r \in \mathcal{R}$ such that

$$\sum_{i \in S_r^{(C)}} \Phi(B_{ir}) + |S_r^{(C)}| > K_r. \tag{4}$$

For a given buffer cycle C and a given routing chain r, the term on the left of (4) is said to be the *DLF term* of routing chain r in buffer cycle C. We say that routing chain r satisfies the *DLF condition* in buffer cycle C, if inequality (4) holds for chain r in buffer cycle C. A capacity assignment Φ is said to satisfy the *DLF conditions for buffer cycle* C, if there is at least one routing chain which satisfies the DLF condition in cycle C. A capacity assignment Φ is said to be *deadlock-free* if Φ satisfies the DLF conditions for all buffer cycles $C \in \mathcal{C}$.

Proof

Necessity: Assume there exists a cycle $C = (B_{i_1 r_1}, B_{i_2 r_2}, \ldots, B_{i_{M-1} r_{M-1}}, B_{i_M r_M})$ such that for all $r \in \mathcal{R}$ we obtain

$$\sum_{i_m \in S_r^{(C)}} \Phi(B_{i_m r}) + |S_r^{(C)}| \leq K_r. \tag{5}$$

With our assumption on the service time distribution we can construct a feasible state where all buffers in C are saturated, i.e., $B_{i_m r_m}$ holds $\Phi(B_{i_m r_m})$ jobs of chain r_m for all $1 \leq m \leq M$, the server of each station i_m contains a job from chain r_{m+1} if $m < M$, and the server of station i_M contains a job from chain r_1. Note that the entire cycle will have $|S_r^{(C)}|$ servers which contain a job from chain r. There exists a positive probability that each job in the server of station i_m $(m < M)$ has picked station i_{m+1} as destination station, and the job in the server of station i_M has selected station i_1 as its next station. In this state, no server can release a job and eventually, each station i_m is blocked. Thus, a deadlock persists.

Sufficiency: Assume that assignment Φ satisfies the DLF condition for all buffer cycles, but the queueing network is in a deadlock state. Then there must exist a permanently blocked job in the server of a station, say i_1. Assume that the blocked job is from routing chain r_2 $(r_2 \in \mathcal{R}_{i_1})$. Assume that the blocked job in the server of station i_1 is blocked at a buffer, say $B_{i_2 r_2}$ of station i_2, that is, $B_{i_2 r_2}$ contains $\Phi(B_{i_2 r_2})$ jobs. Station i_2 itself must be blocked, otherwise a space in $B_{i_2 r_2}$ will eventually become available and station i_1 would not be permanently blocked. The

job in the server of station i_2 is blocked at a saturated buffer, say $B_{i_3 r_3}$ of station i_3. We can continue to apply this argument. Since there is only a finite number of stations in the network, we will eventually encounter a job from routing chain r_{M+1} in the server of some station, say station i_M, which is blocked at a saturated buffer $B_{i_k i_{M+1}}$ of a previously considered station i_k. Then, buffers $B_{i_k r_{M+1}}, B_{i_{k+1} r_{k+1}}, \ldots, B_{i_M r_M}$ define a buffer cycle $C = (B_{i_k r_{M+1}}, B_{i_{k+1} r_{k+1}}, \ldots, B_{i_{M-1} r_{M-1}}, B_{i_M r_M})$. Note that a job from chain r_m is blocked in the server of station i_{m-1} if $B_{i_m r_m}$ is a buffer in the cycle. Therefore, $|S_r^{(C)}|$ jobs from chain r are blocked in servers of stations which have a buffer in the cycle. Since all buffers in the cycle are saturated, there are $\sum_{i_m \in S_r^{(C)}} \Phi(B_{i_m r})$ jobs from routing chain r in the buffers of the cycle. The sum for each chain r must be less or equal the total number of jobs K_r, otherwise the shown construction would not have been feasible. This implies that for all routing chains r with buffers in cycle C we have

$$\sum_{i_m \in S_r^{(C)}} \Phi(B_{i_m r}) + |S_r^{(C)}| \leq K_r. \tag{6}$$

However, this contradicts our assumption that Φ satisfies the DLF condition for all cycles. □

If the DLF conditions (4) are satisfied for a buffer cycle C, they are clearly satisfied for any cycle which contains the buffers of C as a subset. Therefore, to guarantee deadlock freedom of a network, it is not required to test the DLF conditions for all buffer cycles. Let us denote by $C \subseteq C'$ if the set of buffers in a buffer cycle C is a subset of the set of buffers in a cycle C'. We refer to a buffer cycle C as a *minimal cycle* if no cycle in $\mathcal{C}$ is contained in C, i.e., for all cycles $C' \in \mathcal{C}$ we have $C' \not\subseteq C$. We denote by $\tilde{\mathcal{C}}$ the set of all minimal cycles. The following lemma states that a network is deadlock-free if the DLF condition is satisfied for all minimal cycles.

LEMMA 1

Given a multiple chain queueing network Γ with $\mathcal{C}$, the set of all buffer cycles and $\tilde{\mathcal{C}}$, the set of all minimal cycles. Then, Γ is deadlock-free if the DLF condition (4) is satisfied for all $C \in \tilde{\mathcal{C}}$.

Proof

If $C \subseteq C'$, then $S_r^{(C)} \subseteq S_r^{(C')}$ for all $r \in \mathcal{R}$. Therefore, if a routing chain r satisfies the DLF condition in cycle C, that is, $\sum_{i_m \in S_r^{(C)}} \Phi(B_{i_m r}) + |S_r^{(C)}| > K_r$, then chain r also satisfies the DLF condition in all cycles C' with $C \subseteq C'$. Now the claim follows immediately since for any cycle C, either $C \in \tilde{\mathcal{C}}$, or $C \supseteq C'$ and $C' \in \tilde{\mathcal{C}}$. □

4. Deadlock-free capacity assignment algorithm

With the deadlock freedom conditions of theorem 1, we can decide whether a given capacity assignment may result in deadlock situations. In this section, we will show that theorem 1 can be applied in a constructive manner. In particular, we will use theorem 1 to develop an algorithm which finds a deadlock-free capacity assignment with the least total capacity. Such a capacity assignment is referred to as *minimal.*

DEFINITION 3

A deadlock-free capacity assignment Φ^* for a queueing network Γ is *minimal* if for all deadlock-free assignments Φ for Γ it holds that

$$\sum_{i \in \mathcal{N}} \sum_{r \in \mathcal{R}_i} \Phi^*(B_{ir}) \leq \sum_{i \in \mathcal{N}} \sum_{r \in \mathcal{R}_i} \Phi(B_{ir}). \tag{7}$$

Hence, any capacity assignment which allocates less total capacity to the buffers of the queueing network than a minimal capacity assignment will have a deadlock. In the remaining part of this section we will present an algorithm which finds a minimal capacity assignment. The algorithm is executed in two steps:

(1) Find the set $\tilde{C}$ of minimal cycles and establish the DLF conditions for each cycle in $\tilde{C}$.

(2) Formulate an optimization problem of minimizing the total number of buffer capacities, subject to the constraints that the DLF conditions be satisfied for all cycles $C \in \tilde{C}$.

Next we discuss the steps of the algorithm in detail. At the end of the section, we present an example of the optimization algorithm.

We approach the problem of finding the buffer cycles of a queueing network as the problem of finding cycles in a directed graph. The following lemma allows to map the queueing network into a directed graph.

LEMMA 2

Given a multiple chain queueing network Γ. Obtain from Γ a directed graph $G_\Gamma = (V_\Gamma, E_\Gamma)$ with set of vertices V_Γ and set of arcs E_Γ by

(i) $V_\Gamma = \{B_{ir} \mid i \in \mathcal{N}, r \in \mathcal{R}_i\}$,

(ii) $E_\Gamma = \{(B_{ir}, B_{js}) \mid p_{ij,s} > 0\}$.

Then, there exists a one-to-one mapping between the buffer cycles in Γ and the cycles in G_Γ.

Proof

The proof follows immediately from the definition of a buffer cycle in definition 1 and the construction of the directed graph. □

The cycles in the directed graph can be obtained with a cycle-finding algorithm for directed graphs, e.g., Johnson [2]. Note that with lemma 1 we do not need to find *all* cycles in G_Γ. Rather, motivated by lemma 1, we are interested only in the set $\tilde{\mathcal{C}}$ of minimal cycles. An algorithm which finds the set of minimal cycles more efficiently than standard cycle-finding algorithms was presented by Liebeherr [4].

Once the set of buffer cycles is available, we can generate the maximal subset of buffer cycles $\tilde{\mathcal{C}}$ that must be examined for deadlock freedom of the queueing network. If the DLF condition is satisfied for all buffer cycles in $\tilde{\mathcal{C}}$, then the network is deadlock-free. Here, we present two approaches for finding a minimal capacity assignment for queueing networks. The first approach is based on integer programming techniques and guarantees a minimal capacity assignment. The second approach is a heuristic method which always provides a deadlock-free capacity assignment, but may yield a suboptimal solution. The advantage of the heuristic method is that it is computationally less demanding than solving the integer program.

A minimal capacity assignment Φ^* satisfies the DLF condition in all buffer cycles with the least total number of buffer capacities. Thus, for a given network Γ, a minimal capacity assignment is obtained by solving the following optimization problem:

$$\text{Find } \Phi^* \text{ which minimizes } \sum_{i \in \mathcal{N}} \sum_{r \in \mathcal{R}_i} \Phi^*(B_{ir}) \text{ subject to the constraints:}$$

$$(\forall C \in \tilde{\mathcal{C}})(\exists r \in \mathcal{R}) : \sum_{i \in S_r^{(C)}} \Phi^*(B_{ir}) + |S_r^{(C)}| > K_r.$$

Since all constraints of the optimization problem are linear the problem can be formulated as an integer program as shown by Liebeherr [4]. A solution of the integer program provides a minimal capacity assignment Φ^* for network Γ.

In large networks the number of constraints and variables may prohibit an exact solution by integer programming. In these cases, one must resort to heuristic methods which find a deadlock-free capacity assignment with a low number of total buffers, however, without guaranteeing minimality.

In algorithm 1 we present such a heuristic method. Algorithm 1 is started with $\tilde{\mathcal{C}}$, the set of all minimal cycles in Γ. In step 1 we define $\mathcal{C}_{ir} \subseteq \tilde{\mathcal{C}}$ as those minimal cycles which contain buffer B_{ir}. In step 2 we define $\Phi^{min}(B_{ir})$ as the smallest capacity assignment such that B_{ir} by itself satisfies the DLF term for routing chain r in all

Algorithm 1

Input: $\tilde{\mathcal{C}}$, the set of minimal cycles in Γ.
Output: Φ^*, a deadlock-free capacity assignment for Γ.

1. $\mathcal{C}_{ir} := \{C \in \tilde{\mathcal{C}} \mid B_{ir} \in C\}$
2. $\Phi^{min}(B_{ir}) := \min_{C \in \mathcal{C}_{ir}}\{K_r - |S_r^{(C)}| + 1\}$
3. Select (i', r') such that $\dfrac{\Phi^{min}(B_{i'r'})}{|\mathcal{C}_{i'r'}|} = \min\left\{\dfrac{\Phi^{min}(B_{ir})}{|\mathcal{C}_{ir}|} \mid \mathcal{C}_{ir} \subseteq \tilde{\mathcal{C}}, \mathcal{C}_{ir} \neq \emptyset\right\}$
4. $\Phi^*(B_{i'r'}) := \Phi^{min}(B_{i'r'})$
5. $\tilde{\mathcal{C}}_{ir} := \tilde{\mathcal{C}} \setminus \mathcal{C}_{i'r'}$
6. **if** $\tilde{\mathcal{C}} \neq \emptyset$ **then goto** Step 2.

buffer cycles in $\tilde{\mathcal{C}}_{ir}$, i.e.,

$$\Phi^{min}(B_{ir}) + |S_r^{(C)}| > +K_r \quad \text{for all } C \in \tilde{\mathcal{C}}_{ir}.$$

$\Phi^{min}(B_{ir})$ can be interpreted as the incurred *cost* for satisfying the DLF conditions in all cycles in $\tilde{\mathcal{C}}_{ir}$ through a capacity assignment to B_{ir}. In this sense, the ratio $\Phi^{min}(B_{ir})/|\mathcal{C}_{ir}|$ expresses the *relative cost per buffer cycle* incurred by assigning a capacity of $\Phi^{min}(B_{ir})$ to buffer B_{ir}. In steps 3 and 4 we select the buffer $B_{i'r'}$ which minimizes the *relative cost* $\Phi^{min}(B_{ir})/|\mathcal{C}_{ir}|$. Note that by assigning to buffer $B_{i'r'}$ a capacity of $\Phi^{min}(B_{i'r'})$, all DLF conditions of cycles in $\mathcal{C}_{i'r'}$ are satisfied. In step 5, these cycles are eliminated from the sets $\mathcal{C}_{ir}$. If there are minimal cycles left whose DLF conditions are not satisfied we begin a new iteration (step 6).

When the algorithm terminates all DLF conditions are satisfied and we have obtained a deadlock-free capacity assignment. However, there is no guarantee that the assignment is minimal. In an example given below we will see that algorithm 1 yields results which are close to or identical with a minimal assignment. Note that algorithm 1 is guaranteed to converge quickly since the number of iterations of the algorithms is bounded by the number of buffers in Γ.

Example

Next we apply the minimal capacity assignment algorithm to an example. Figure 2 depicts a queueing network with three routing chains. The network contains 429 elementary buffer cycles, that is, buffer cycles where no buffer appears twice. However, the set of minimal buffer cycles contains only 19 elements. Thus, with lemma 1, only 19 buffer cycles must be considered to determine a minimal capacity assignment. These cycles are given as follows:

$$C_1 = (B_{11}, B_{21}),$$

$$C_2 = (B_{92}, B_{10\,2}),$$

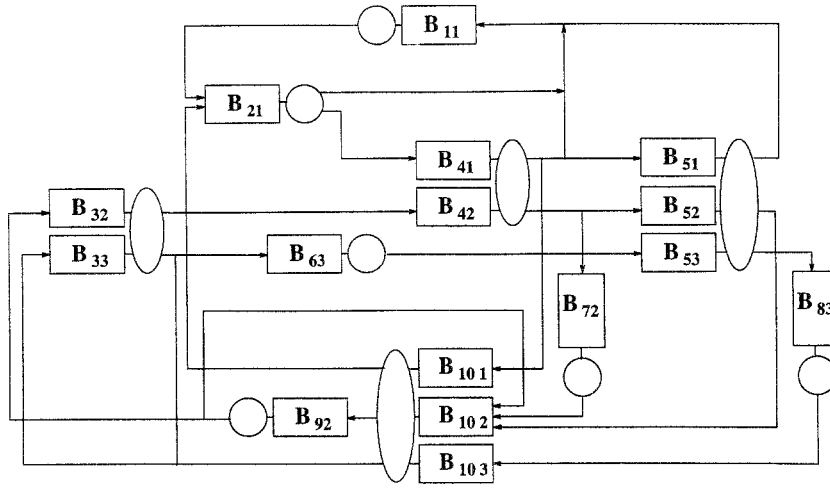

Fig. 2. Multiple chain queueing network.

$$C_3 = (B_{53}, B_{83}, B_{10\,2}, B_{63}),$$

$$C_4 = (B_{21}, B_{41}, B_{10\,1}),$$

$$C_5 = (B_{21}, B_{41}, B_{51}, B_{10\,2}),$$

$$C_6 = (B_{21}, B_{41}, B_{52}, B_{83}, B_{10\,3}),$$

$$C_7 = (B_{21}, B_{41}, B_{52}, B_{10\,2}),$$

$$C_8 = (B_{21}, B_{41}, B_{72}, B_{10\,2}),$$

$$C_9 = (B_{42}, B_{51}, B_{83}, B_{10\,3}, B_{33}),$$

$$C_{10} = (B_{42}, B_{51}, B_{83}, B_{10\,3}, B_{92}, B_{32}),$$

$$C_{11} = (B_{42}, B_{51}, B_{10\,2}, B_{33}),$$

$$C_{12} = (B_{42}, B_{52}, B_{83}, B_{10\,3}, B_{33}),$$

$$C_{13} = (B_{42}, B_{52}, B_{83}, B_{10\,3}, B_{92}, B_{32}),$$

$$C_{14} = (B_{42}, B_{52}, B_{10\,2}, B_{33}),$$

$$C_{15} = (B_{42}, B_{10\,1}, B_{33}),$$

Table 1
Results for example.

	$K_1 = 10$ $K_2 = 10$ $K_3 = 10$		$K_1 = 46$ $K_2 = 4$ $K_3 = 125$		$K_1 = 3$ $K_2 = 3$ $K_3 = 3$	
	minimal	heuristic	minimal	heuristic	minimal	heuristic
$\sum_{i \in \mathcal{N}} \sum_{r \in \mathcal{R}_i} \Phi^*(B_{ir})$	36	37	175	175	8	8
$\Phi^*(B_{21})$	9	9	45	45	2	2
$\Phi^*(B_{42})$	10	10	4	4	3	3
$\Phi^*(B_{10\,2})$	9	10	4	4	3	3
$\Phi^*(B_{63})$	7	9	122	122	0	0

$$C_{16} = (B_{42}, B_{10\,1}, B_{92}, B_{32}),$$

$$C_{17} = (B_{42}, B_{72}, B_{10\,2}, B_{33}),$$

$$C_{19} = (B_{53}, B_{10\,2}, B_{63}).$$

In table 1 we show the solution of the optimization for different values of $K = (K_1, K_2, K_3)$. The optimizations were solved with the programming package LINDO [6]. We also include the results obtained with the heuristic method given in algorithm 1. Table 1 only depicts non-zero capacity assignments. Note that for all values of K, the heuristic method of algorithm 1 provides results which are very close to or identical with an optimal solution.

5. Deadlock-free capacity assignments in tandem networks

The computational complexity of the capacity assignment algorithm presented in section 4 increases with the number of mimimal cycles in Γ, as given by the number of elements in $\tilde{C}$. For some networks the size of $\tilde{C}$ can make the optimization algorithm impractical. As a worst case, consider the network in fig. 3. Here $\tilde{C}$ is identical with the set of all buffer cycles in the network. Note that in the queueing network in fig. 3, the set $\tilde{C}$ contains R^N minimal cycles. In the following we show that the effort to obtain a minimal capacity assignment can be greatly reduced if we take advantage of networks with a regular topology such as the network shown in fig. 3.

We refer to networks which have a topology as shown in fig. 3 as *tandem networks*. Tandem networks can be formally defined as follows.

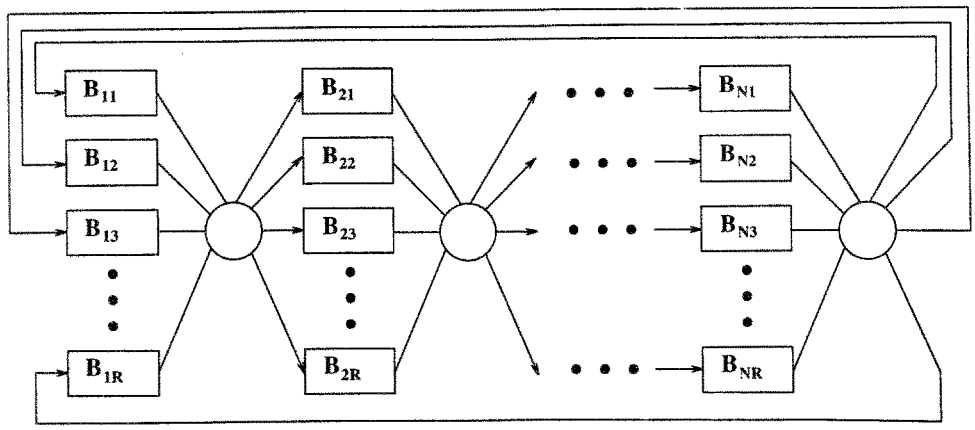

Fig. 3. Tandem network.

DEFINITION 4

A *tandem network* $\Gamma = (\mathcal{N}, \mathcal{R})$ is a multiple chain queueing network $\Gamma = (\mathcal{N}, \mathcal{R}, \mathcal{P})$ with

$$\mathcal{R}_i = \mathcal{R} \quad \text{for all } i \in \mathcal{N} \tag{8}$$

and for each station $i \in \mathcal{N}$ there exists a station $i' \in \mathcal{N}$ $(i \neq i')$ such that for all $r \in \mathcal{R}$:

$$p_{ij,r} = \begin{cases} 1 & \text{if } j = i', \\ 0 & \text{if } j \neq i'. \end{cases} \tag{9}$$

In a tandem network, each station has a buffer for all routing chains, i.e., $\mathcal{R}_i = \mathcal{R}$ for all $i \in \mathcal{N}$, and jobs from each routing chain can visit all stations, i.e., $\mathcal{N}_r = \mathcal{N}$ for all $r \in \mathcal{R}$. Each buffer cycle in a tandem network contains one buffer from each station, and the length of each buffer cycle is equal to $|\mathcal{N}|$.

Our main result in this section is that minimal capacity assignments in tandem networks can be provided without running any optimization algorithm. The result is presented in the following theorem where we state that a minimal assignment for tandem networks is obtained by allocating non-zero capacities to the buffers of only one station.

THEOREM 2

For a tandem network $\Gamma = (\mathcal{N}, \mathcal{R})$, the following capacity assignments

Φ_i^* $(i \in \mathcal{N})$ are minimal:

$$\Phi_i^*(B_{jr}) = \begin{cases} K_r & \text{if } j = i \text{ and } K - K_r \geq |\mathcal{N}|, \\ K - |\mathcal{N}| + 1 & \text{if } j = i \text{ and } K \geq |\mathcal{N}| > K - K_r, \\ 0 & \text{if } j = i \text{ and } |\mathcal{N}| > K, \\ 0 & \text{if } j \neq i, \end{cases} \tag{10}$$

$$\text{for } j \in \mathcal{N} \text{ and } r \in \mathcal{R}.$$

Note that theorem 2 defines a set of $|\mathcal{N}|$ different capacity assignments; That is, there is one capacity assignment Φ_i^* for each station $i \in \mathcal{N}$. The remainder of this section contains the proof of theorem 2. The proof consists of two parts. The first part is proven through lemma 7 which states that the capacity assignments given above do not allow deadlock situations. For the second part of the proof we have to show that any capacity assignment which assigns less total buffer capacities to the stations of the tandem network will result in a deadlock. This claim is made in lemma 8. The proof of lemma 8 requires considerable effort since we must effectively construct a deadlock for a large class of capacity assignments. Before we state and prove lemmas 7 and 8, we present a set of technical lemmas which will be used for the proof of lemma 8.

Remark

Note that the network shown in fig. 1 also is a tandem network. Thus, with theorem 2 we can give two minimal capacity assignments, Φ_1^* and Φ_2^*. Assuming the same network parameters as in section 1 we have $|\mathcal{N}| = 2$ and $K = K_1 + K_2 = 11$. Therefore, Φ_1^* and Φ_2^* assign capacities as shown here:

$$\Phi_1^*(B_{11}) = 5, \quad \Phi_2^*(B_{11}) = 0,$$

$$\Phi_1^*(B_{12}) = 6, \quad \Phi_2^*(B_{12}) = 0,$$

$$\Phi_1^*(B_{21}) = 0, \quad \Phi_2^*(B_{21}) = 5,$$

$$\Phi_1^*(B_{22}) = 0, \quad \Phi_2^*(B_{22}) = 6.$$

It can be easily verified that Φ_1^* and Φ_2^* are the only minimal capacity assignment for the tandem network in fig. 1. However, for some tandem networks, there may exist minimal capacity assignments which are different from the assignments given in theorem 2. For example, in a tandem network with $|\mathcal{N}| = 3$, $\mathcal{R} = \{1, 2\}$, and

$K_1 = K_2 = 3$, a capacity assignment Φ' with $\Phi'_{ir} = 1$ (for $i = 1,2,3$ and $r = 1,2$) is minimal.

In tandem networks, a violation of a DLF condition by a routing chain r can be easily detected if the number of stations in the tandem network is equal to the number of jobs of the routing chain, that is, $K_r = |\mathcal{N}|$. This assertion is shown in lemmas 3 and 4. In lemmas 5 and 6, we show how to modify the structure of tandem networks such that we can take advantage of lemmas 3 and 4, yet, any deadlock in the modified network corresponds to a deadlock in the original network.

LEMMA 3

Given a tandem network $\Gamma = (\mathcal{N}, \mathcal{R})$. Let Φ be a capacity assignment for Γ such that $\sum_{i \in \mathcal{N}} \Phi(B_{ir}) \leq K_r + 1$ and $K_r = |\mathcal{N}|$ for some routing chain $r \in \mathcal{R}$. Let C be a buffer cycle in Γ such that $|S_r^{(C)}| > 0$. If routing chain r satisfies the DLF condition for cycle C then it does not satisfy the DLF condition in any buffer cycle C' with $S_r^{(C')} = \mathcal{N} \setminus S_r^{(C)}$. (For two sets A and B, we use $A \setminus B$ to denote the relative complement of B with respect to A, i.e., $A \setminus B := \{x \in A \mid x \notin B\}$).

Thus, if the number of jobs in a routing chain is equal to the number of stations, and the total capacity which is assigned to the buffers of this routing chain exceeds the number of stations by at most one, we can always find a buffer cycle such that the routing chain does not satisfy the DLF condition.

Proof

Assume a capacity assignment Φ and a routing chain $r \in \mathcal{R}$ with $\sum_{i \in \mathcal{N}} \Phi(B_{ir}) \leq K_r + 1$. Let C be a buffer cycle in Γ such that chain r satisfies the DLF condition, i.e.,

$$\sum_{i \in S_r^{(C)}} \Phi(B_{ir}) + |S_r^{(C)}| > K_r. \tag{11}$$

Now assume a buffer cycle C' with $S_r^{(C')} = \mathcal{N} \setminus S_r^{(C)}$. Then, we obtain for the DLF term of chain r in cycle C':

$$\sum_{i \in S_r^{(C')}} \Phi(B_{ir}) + |S_r^{(C')}| = \sum_{i \in S_r^{(C')}} \Phi(B_{ir}) + (|\mathcal{N}| - |S_r^{(C)}|) \tag{12}$$

$$\leq \left(K_r + 1 - \sum_{i \in S_r^{(C)}} \Phi(B_{ir}) \right) + (K_r - |S_r^{(C)}|) \tag{13}$$

$$< K_r + 1. \tag{14}$$

The equality in (12) follows from $S_r^{(C')} = \mathcal{N} \setminus S_r^{(C)}$. For (13) we use $\sum_{i \in \mathcal{N}} \Phi(B_{ir}) \leq K_r + 1$ and $K_r = |\mathcal{N}|$. Finally, we obtain (14) from (11). Thus, the DLF condition of chain r in cycle C' is not satisfied. □

The following lemma provides an even stronger result. Similar to lemma 3, we assume that the number of stations in the tandem network is equal to the number of jobs of a particular routing chain. Here, however, we assume that the total capacities assigned to the buffers of the routing chain is less than the number of stations. Then we can partition the stations of the tandem network into two sets such that the DLF condition of the routing chain is violated in both sets. The result is proven by Liebeherr and Akyildiz [5] where we provide an algorithm that achieves the desired partition.

LEMMA 4

Given a tandem network $\Gamma = (\mathcal{N}, \mathcal{R})$. Let Φ be a capacity assignment for Γ such that $\sum_{i \in \mathcal{N}} \Phi(B_{ir}) < K_r$ and $K_r = |\mathcal{N}|$ for some routing chain $r \in \mathcal{R}$. Then $\mathcal{N}$ can be partitioned into two disjoint sets $\mathcal{N}_1$ and $\mathcal{N}_2$ such that

(a) $$\sum_{i \in \mathcal{N}_1} \Phi(B_{ir}) + |\mathcal{N}_1| \leq K_r, \text{ and}$$

(b) $$\sum_{i \in \mathcal{N}_2} \Phi(B_{ir}) + |\mathcal{N}_2| \leq K_r.$$

Lemmas 3 and 4 enable us to make statements regarding the deadlock properties of routing chains where the number of jobs is equal to the number of stations in the tandem network. The next two lemmas, lemmas 5 and 6, show how to modify a tandem network such that we can take advantage of lemmas 3 and 4. We refer to Liebeherr and Akyildiz [5] for complete proofs of lemmas 5 and 6.

LEMMA 5

Given an arbitrary network $\Gamma = (\mathcal{N}, \mathcal{R}, \mathcal{P})$, and Φ, a capacity assignment for Γ. Consider a routing chain $r \in \mathcal{R}$ with K_r jobs. Reduce the number of jobs in chain r by one and define a new capacity assignment Φ' which differs from Φ in the capacity assignment to only one buffer B_{jr} with $\Phi(B_{jr}) > 0$ as follows:

$$\Phi'(B_{iq}) = \begin{cases} \Phi(B_{iq}) - 1 & \text{if } q = r \text{ and } i = j, \\ \Phi(B_{iq}) & \text{otherwise.} \end{cases}$$

Then Φ violates the DLF condition for chain r with K_r jobs in a buffer cycle, if Φ' violates the DLF condition for chain r with $K_r - 1$ jobs in the same buffer cycle.

Lemma 5 says that given a routing chain with a large capacity assigned to its buffers, i.e., $\sum_{i \in \mathcal{N}} \Phi(B_{ir}) > K_r$, we can reduce both the capacity assignments and the number of jobs such that any deadlock in the modified network also results in a deadlock in the unmodified network. In lemma 6 we show how to modify a network such that at least one routing chain has more jobs than the modified network has stations, yet, any deadlock in the modified network can be extended to a deadlock in the original network.

LEMMA 6

Given a tandem network $\Gamma = (\mathcal{N}, \mathcal{R})$ and a capacity assignment Φ for Γ such that the following conditions hold:

($\alpha 1$) $\sum_{i \in \mathcal{N}} \Phi(B_{ir}) \leq K_r$ for all $r \in \mathcal{R}$, and

($\beta 1$) $\sum_{r \in \mathcal{R}} K_r \geq |\mathcal{N}|$.

Then one can effectively construct from Γ a tandem network $\Gamma' = (\mathcal{N}', \mathcal{R}')$ with $\mathcal{N}' \subseteq \mathcal{N}$ and $\mathcal{R}' \subseteq \mathcal{R}$, such that

($\gamma 1$) $K_r \geq |\mathcal{N}'|$ for at least one $r \in \mathcal{R}'$.

($\delta 1$) If Γ' is not deadlock-free, then Γ is not deadlock-free.

Now we proceed with the proof of theorem 2. In lemma 7 we show that the capacity assignments given in theorem 2 yield a deadlock-free network. Recall that theorem 2 defines $|\mathcal{N}|$ capacity assignments Φ_i^*, one for each station $i \in \mathcal{N}$. In the following, we assume that the index i of capacity assignment Φ_i^* is arbitrary, but fixed.

LEMMA 7

For a tandem network $\Gamma = (\mathcal{N}, \mathcal{R})$ the capacity assignments Φ_i^* ($i \in \mathcal{N}$) from eq. (10) yield a deadlock free network.

Proof

To prove freedom of deadlocks, we distinguish three cases: (a) $|\mathcal{N}| > K$, (b) $K - K_r \geq |\mathcal{N}|$, and (c) $K \geq |\mathcal{N}| < K - K_r$. We take advantage of the fact that in a tandem network each buffer cycle C must contain a buffer from station i. We will assume without loss of generality that the buffer from station i belongs to routing chain r, that is, $i \in S_r^{(C)}$.

(a) $|\mathcal{N}| > K$

In this case, it is not possible that the servers of all stations in Γ are occupied by jobs. Hence, for each buffer cycle C there must exist a chain $r \in \mathcal{R}$

with:

$$|S_r^{(C)}| > K_r. \tag{15}$$

With (15), routing r satisfies the DLF condition for cycle C.

(b) $K - K_r \geq |\mathcal{N}|$

Here, theorem 2 assigns capacities such that $\Phi_i^*(B_{ir}) = K_r$. Since per assumption we have $|S_r^{(C)}| \geq 1$, the DLF term for routing chain r in cycle C evaluates to

$$\sum_{j \in S_r^{(C)}} \Phi_i^*(B_{jr}) + |S_r^{(C)}| = K_r + |S_r^{(C)}| > K_r. \tag{16}$$

Clearly, the DLF condition is satisfied.

(c) $K \geq |\mathcal{N}| > K - K_r$

We first consider all cycles C with

$$i \in S_r^{(C)} \quad \text{and} \quad |S_r^{(C)}| < |\mathcal{N}| - (K - K_r). \tag{17}$$

Since all cycles in a tandem network have a length of $|\mathcal{N}|$, i.e., $\sum_{r \in \mathcal{R}} |S_r^{(C)}| = |\mathcal{N}|$, we obtain from (17) that there must exist a routing chain q $(q \neq r)$ such that $|S_q^{(C)}| > K_q$. But this satisfies the DLF condition for chain q in cycle C.

Now we consider $K \geq |\mathcal{N}| > K - K_r$ and all cycles C with

$$i \in S_r^{(C)} \quad \text{and} \quad |S_r^{(C)}| \geq |\mathcal{N}| - (K - K_r). \tag{18}$$

With $i \in S_r^{(C)}$ and $\Phi_i^*(B_{ir}) = K - |\mathcal{N}| + 1$ we obtain

$$\sum_{j \in S_r^{(C)}} \Phi_i^*(B_{jr}) + |S_r^{(C)}| \geq (K - |\mathcal{N}| + 1) + (|\mathcal{N}| - (K - K_r)) \tag{19}$$

$$= K_r + 1 \tag{20}$$

$$> K_r. \tag{21}$$

Thus, the DLF condition for routing chain r in cycle C is satisfied. □

As our last step for the proof of theorem 2, we show in lemma 8 the minimality of the assignments Φ_i^*. Lemmas 7 and 8 together prove the correctness of theorem 2.

LEMMA 8

Consider a tandem network $\Gamma = (\mathcal{N}, \mathcal{R})$. Any capacity assignment Φ which allocates less total capacities to the buffers of some routing chain r than assignments Φ_i^* $(i \in \mathcal{N})$ from eq. (10) will result in a deadlock situation.

Proof

For $K < |\mathcal{N}|$, no capacity is assigned to any buffer, and Φ_i^* is trivially minimal. In the following, we investigate the minimality of Φ_i^* for $K \geq |\mathcal{N}|$.

To show minimality, we must show that any capacity assignment which assigns less total capacity to the stations of the tandem network than Φ_i^* will result in a deadlock situation. Let us select an arbitrary routing chain $r \in \mathcal{R}$ and consider a capacity assignment $\bar{\Phi}$ which satisfies:

$$\sum_{j \in \mathcal{N}} \bar{\Phi}(B_{jq}) = \begin{cases} \sum_{j \in \mathcal{N}} \Phi_i^*(B_{jq}) - 1 & \text{if } q = r, \\ \sum_{j \in \mathcal{N}} \Phi_i^*(B_{jq}) & \text{if } q \neq r. \end{cases} \tag{22}$$

Thus, $\bar{\Phi}$ assigns one buffer space less to the buffers of routing chain r than capacity assignment Φ_i^*. The total capacity assigned to buffers from chains $q \neq r$ by $\bar{\Phi}$ remains unchanged. Note that we do not have any assumptions on how $\bar{\Phi}$ distributes the capacities to the stations of the tandem network. In particular, we do not assume that $\bar{\Phi}$ assigns non-zero capacities to only one station. For $K \geq |\mathcal{N}|$, we will show that assignment $\bar{\Phi}$ results in a deadlock in at least one buffer cycle in Γ.

Recall that in theorem 2, the capacities assigned to buffers from routing chain r are different for $K - K_r \geq |\mathcal{N}|$ and $K - K_r < |\mathcal{N}|$. Here we show only the proof of the case $K - K_r \geq |\mathcal{N}|$. A proof for the case $K - K_r < |\mathcal{N}|$ is given by Liebeherr and Akyildiz [5].

For $K - K_r \geq |\mathcal{N}|$, consider the tandem network $\Gamma' = (\mathcal{N}, \mathcal{R}\backslash\{r\})$ that is obtained from Γ by eliminating routing chain r. Further consider capacity assignment $\bar{\Phi}$ as given in (22). Since Γ' satisfies conditions $(\alpha 1)$ and $(\beta 1)$ in lemma 6, one can construct a network $\Gamma'' = (\mathcal{N}'', \mathcal{R}'')$ with $\mathcal{N}'' \subseteq \mathcal{N}$ and $\mathcal{R}'' \subseteq \mathcal{R}\backslash\{r\}$, such that Γ'' contains a routing chain $q \in \mathcal{R}''$ with $K_q \geq |\mathcal{N}''|$.

Now consider the tandem network $\Gamma''' = (\mathcal{N}'', \{r, q\})$. We will construct a deadlock in Γ'''. With lemma 6, any deadlock in Γ''' can be extended to a deadlock in Γ. Dependent on the size of K_r relative to $|\mathcal{N}''|$, the construction of the deadlock in Γ''' will be different for $K_r \geq |\mathcal{N}''|$ and $K_r < |\mathcal{N}''|$.

$K_r \geq |\mathcal{N}''|$:

Note that $K - K_q \geq |\mathcal{N}|$ must hold in this case. ($K - K_q < |\mathcal{N}|$ implies $\sum_{s \in \mathcal{R}''} K_s + K_r - K_q < |\mathcal{N}''|$ which contradicts $K_r \geq |\mathcal{N}''|$.) Therefore, capacity

assignment $\bar{\Phi}$ assigns the following total capacities to buffers from routing chains r and q:

$$\sum_{j \in \mathcal{N}} \bar{\Phi}(B_{jr}) = K_r - 1, \tag{23}$$

$$\sum_{j \in \mathcal{N}} \bar{\Phi}(B_{jq}) = K_q. \tag{24}$$

We apply lemma 5 repeatedly to chains r and q until $K_r = |\mathcal{N}''|$ and $K_q = |\mathcal{N}''|$. Then, by applying lemma 4 we can effectively partition $\mathcal{N}''$ into two sets $\mathcal{N}_1''$ and $\mathcal{N}_2''$ such that chain r does not satisfy the DLF condition in either subset.

Now consider two cycles C_1 and C_2 with

$$\begin{aligned} S_r^{(C_1)} &= |\mathcal{N}_1''|, \quad S_q^{(C_1)} = |\mathcal{N}_2''|, \\ S_r^{(C_2)} &= |\mathcal{N}_2''|, \quad S_q^{(C_2)} = |\mathcal{N}_1''|. \end{aligned} \tag{25}$$

Per construction of C_1 and C_2, the DLF condition for chain r is not satisfied in either cycle. With lemma 3 the DLF condition for chain q cannot be satisfied in both cycles C_1 and C_2. Therefore, the DLF conditions for both routing chains are violated in C_1 or C_2. Thus, either cycle C_1 or C_2 contains a deadlock. Since the modifications to Γ have preserved any possibly existing deadlock (see lemmas 5 and 6), one of the two cycles C_1 or C_2 can be extended to a buffer cycle in Γ such that no routing chain $r \in \mathcal{R}$ satisfies the DLF condition of this cycle. As a result, we have constructed a deadlock in Γ.

$K_r < |\mathcal{N}''|$:

If $K_r < |\mathcal{N}''|$ holds, both $K - K_q \geq |\mathcal{N}|$ and $K - K_q < |\mathcal{N}|$ are feasible. Here we only consider $K - K_q < |\mathcal{N}|$. Then the following holds for the number of stations in Γ'' and Γ''' as constructed above:

$$|\mathcal{N}''| = |\mathcal{N}| - \sum_{s \neq r,q} K_s. \tag{26}$$

Since per our assumptions, $K - K_r \geq |\mathcal{N}|$ and $K - K_q < |\mathcal{N}|$, capacity assignment $\bar{\Phi}$ assigns the following total capacities to routing chains r and q:

$$\sum_{i \in \mathcal{N}} \bar{\Phi}(B_{ir}) = K_r - 1, \tag{27}$$

$$\sum_{i \in \mathcal{N}} \bar{\Phi}(B_{iq}) = K - |\mathcal{N}| + 1 \tag{28}$$

$$= K_r + K_q - |\mathcal{N}''| + 1. \tag{29}$$

The equality in (29) is obtained with (26). Next we apply lemma 5 repeatedly to chain q until $K_q = |\mathcal{N}''|$. Thus, lemma 5 is applied exactly $K_q - |\mathcal{N}''|$ times. Applying lemma 5 changes the capacity assignment to chain q, and we use $\bar{\Phi}'$ to refer to the modified capacity assignment. $\bar{\Phi}'$ is such that

$$\sum_{j \in \mathcal{N}} \bar{\Phi}'(B_{jr}) = \sum_{j \in \mathcal{N}} \bar{\Phi}(B_{jr}), \tag{30}$$

$$\sum_{j \in \mathcal{N}} \bar{\Phi}'(B_{jq}) = \sum_{j \in \mathcal{N}} \bar{\Phi}(B_{jq}) - (K_q - |\mathcal{N}''|) \tag{31}$$

$$= (K_r + K_q - |\mathcal{N}''| + 1) - (K_q - |\mathcal{N}''|) \tag{32}$$

$$= K_r + 1. \tag{33}$$

Let us define $\mathcal{NZ}_q$ as the set of stations in Γ''' with non-zero capacities assigned to buffers of chain q. With (33) we obtain the following upper bound for $|\mathcal{NZ}_q|$:

$$|\mathcal{NZ}_q| \leq K_q + 1. \tag{34}$$

Next we select a set $\overline{\mathcal{NZ}}_q$ with $\mathcal{NZ}_q \subseteq \overline{\mathcal{NZ}}_q$ such that

$$|\overline{\mathcal{NZ}}_q| = K_r + 1. \tag{35}$$

Since $\sum_{j \in \mathcal{N}} \bar{\Phi}(B_{jr}) = K_r - 1$, and since $\sum_{j \in \overline{\mathcal{NZ}}_q} \bar{\Phi}(B_{jr}) \leq \sum_{j \in \mathcal{N}} \bar{\Phi}(B_{jr})$ we have

$$\sum_{j \in \overline{\mathcal{NZ}}_q} \bar{\Phi}(B_{jr}) = K_r - 1. \tag{36}$$

Hence, we can apply lemma 4 and partition $\overline{\mathcal{NZ}}_q$ into two sets $\overline{\mathcal{NZ}}_{q1}$ and $\overline{\mathcal{NZ}}_{q2}$ such that

$$\sum_{j \in \overline{\mathcal{NZ}}_{q1}} \bar{\Phi}'(B_{jr}) + |\overline{\mathcal{NZ}}_{q1}| \leq K_r \quad \text{and} \quad \sum_{j \in \overline{\mathcal{NZ}}_{q2}} \bar{\Phi}'(B_{jr}) + |\overline{\mathcal{NZ}}_{q2}| \leq K_r. \tag{37}$$

For routing chain q, we have with (33) that $\sum_{j \in \overline{\mathcal{NZ}}_q} \bar{\Phi}'(B_{jq}) = K_r + 1$. Thus, we

obtain with lemma 3 that either

$$\sum_{j\in\overline{\mathcal{NZ}}_{q1}} \bar{\Phi}'(B_{jq}) + |\overline{\mathcal{NZ}}_{q1}| \leq K_r + 1 \quad \text{or}$$

$$\sum_{j\in\overline{\mathcal{NZ}}_{q2}} \bar{\Phi}'(B_{jq}) + |\overline{\mathcal{NZ}}_{q2}| \leq K_r + 1. \tag{38}$$

Without loss of generality we will assume that the first inequality in (38) holds. Then we can perform the following manipulations:

$$\sum_{j\in\mathcal{N}''\setminus\overline{\mathcal{NZ}}_{q2}} \bar{\Phi}'(B_j) + |\mathcal{N}''\setminus\overline{\mathcal{NZ}}_{q2}|$$

$$= \sum_{q} \sum_{j\in\mathcal{N}''\setminus\overline{\mathcal{NZ}}_{q2}} \bar{\Phi}'(B_{jq}) + |\overline{\mathcal{NZ}}_{q1}| + |\mathcal{N}''| - |\overline{\mathcal{NZ}}_{q}| \tag{39}$$

$$= \sum_{j\in\overline{\mathcal{NZ}}_{q1}} \bar{\Phi}'(B_{jq}) + |\overline{\mathcal{NZ}}_{q1}| + |\mathcal{N}''| - (K_r + 1) \tag{40}$$

$$\leq (K_r + 1) + |\mathcal{N}''| - (K_r + 1) \tag{41}$$

$$= |\mathcal{N}''|. \tag{42}$$

Equation (39) follows from $|\overline{\mathcal{NZ}}_q| = |\overline{\mathcal{NZ}}_{q1}| + |\overline{\mathcal{NZ}}_{q2}|$. Equation (40) follows with (35) and $\sum_{j\in\mathcal{N}''\setminus\overline{\mathcal{NZ}}_q} \bar{\Phi}'(B_{jq}) = 0$. The inequality in (40) is obtained with (38), and (42) is obtained by canceling terms. With the above derivations, we can define a buffer cycle C in Γ''' such that

$$S_r^{(C)} = \overline{\mathcal{NZ}}_{q2}, \quad S_q^{(C)} = \mathcal{N}''\setminus\mathcal{NZ}_{q2}. \tag{43}$$

It follows from (37), and (38)–(42), that the DLF conditions for both routing chains r and q are not satisfied. Therefore, a deadlock results in network Γ'''. With the same arguments as used before, the deadlock in Γ''' results in a deadlock in Γ. □

References

[1] I.F. Akyildiz and H.G. Perros (eds.), Queueing networks with finite capacity queues, Special Issue of Perf. Eval. 10(4) (1989).

[2] B. Johnson, Finding all the elementary circuits in a directed graph, SIAM J. Comp. 4 (1975) 77–84.

[3] S. Kundu and I.F. Akyildiz, Deadlock free buffer allocation in closed queueing networks, Queueing Systems 4 (1989) 47–56.
[4] J. Liebeherr, Performance evaluation of synchronized systems, Ph.D. Thesis, Georgia Institute of Technology, College of Computing (1991).
[5] J. Liebeherr and I.F. Akyildiz, Deadlock properties of queueing networks with finite capacities and multiple routing chains, Technical Report CS-94-65, University of Virginia, Department of Computer Science (1993).
[6] L. Schrage, *Linear, Integer, and Quadratic Programming with LINDO*, 3rd Ed. (The Scientific Press, 1986).