Application of Norton's Theorem on Queueing Networks with Finite Capacities

I. F. Akyildiz and J. Liebeherr

School of Information and Computer Science
Georgia Institute of Technology
Atlanta, Georgia 30332
U. S. A.

ABSTRACT

The application of Norton's theorem from electrical circuit theory on queueing networks with infinite capacities is well-known and very useful for cases where a node of the network should be analyzed under different workload. In this work a method is developed which allows the application of Norton's theorem on queueing networks with finite capacities. A node is arbitrarily selected and the subnetwork containing all remaining nodes are replaced by a composite node with infinite capacity. The entire network is reduced to a two-node network having the node of interest and the composite node. Although blocking causes interdependencies between nodes in the network the selected node is totally isolated from the rest of the network by constructing phases in the server which reflect the blocking events. An algorithm is given to compute the parameters of the phases. Several examples are discussed to demonstrate the efficiency and generality of the technique. Comparisons with simulation results show that the proposed technique provides accurate results for throughput values.

Key Words: Performance Evaluation, Queueing Networks, Parametric Analysis, Blocking, Throughput

1. Introduction

Queueing networks have experienced a dramatic increase in their importance regarding performance evaluation of computer systems and communication networks. When considering systems in which the nodes have infinite capacities, numerous methods have been introduced in the past two decades. However, since in actual systems nodes have a finite capacity, queueing networks with blocking should be used for performance analysis. Queueing networks with blocking have thus become an important research topic within performance evaluation during recent years. Several computational methods have been developed to analyze queueing networks with blocking. These are networks where the nodes have finite capacities, hence blocking can occur if the node is full to its capacity. A job which wants to come to the full node must reside in the server of the source node until a place is available in the destination node. The interest in networks with blocking comes primarily from the realization that these models are useful in the study of the behavior of subsystems of computers and communication networks, in addition to detailed descriptions of several computer-related applications such as flexible manufacturing systems. In this work we consider the so-called transfer blocking mechanism in queueing networks. In this case, the blocking event occurs when a job completing service at node i cannot proceed to node j because node j is full. The job resides in node i's server, which stops processing until node j releases a job. This type of blocking has been used to model systems such as production systems and disk I/O subsystems.

Several investigators in recent years have published results on queueing networks with transfer blocking. Since we are investigating closed queueing networks with transfer blocking we discuss here the previous work only for this type of networks. Akyildiz [3] studied two-node closed queueing networks with transfer blocking and multiple server nodes. He showed that the equilibrium state probability distributions of such blocking systems are identical to those of a two-node closed queueing network without blocking. Akyildiz [5] also showed that the throughput of a blocking network with K total number of jobs is approximately equal to the throughput of a non-blocking network with an appropriate total number of jobs $\hat{K}$. The well-known mean value analysis algorithm [22] is extended by Akyildiz [6] to single server queueing networks with blocking. The approximation is based on the modification of mean residence times due to the blocking events that occur in the network. Two algorithms for the computation of throughput values and the mean queue lengths in Markovian blocking queueing networks with multiple servers is given in [7] which is extended in [4] to networks with general service time distributions and FCFS scheduling disciplines.

Suri/Diehl [25] developed a method for approximate analysis of closed tandem queueing networks with transfer blocking. They approximate groups of two nodes by a variable capacity node, defined as a superposition of fixed capacity nodes. They start with the last two nodes and successively reduce the network until two nodes in tandem remain. The method is easy to implement and shows good accuracy but involves much computation. At each step all conditional probabilities have to be found, since they are used to construct the equivalent variable capacity node. The major disadvantage of their technique is that one of the nodes must have an infinite capacity. Additionally, their method only gives the throughput of the entire network it does not give statistics for individual nodes. Another drawback is that the capacity of each downstream node must be smaller than the total number of jobs in the network.

Dallery/Frein [12] introduce an iterative technique to obtain performance measures for the same network configuration as investigated by Suri/Diehl. Their throughput values are generally less accurate than those provided by the method of Suri/Diehl. However, they obtain values for the mean number of jobs which cannot be computed by the method of Suri/Diehl. Perros, Nilsson and Liu [21] give an algorithm for an arbitrarily connected network where some nodes have finite capacity. They partition the set of nodes in a so-called blocking subnetwork and a non-blocking subnetwork. The non-blocking subnetwork containing infinite capacity nodes is replaced by a composite node using parametric analysis for infinite capacity networks. The reduced network is then analyzed numerically. However, if all nodes of the network have finite capacity this method reduces itself to a numerical analysis method which, as generally known, is applicable only on very small networks. Onvural/Perros [16] present

This work was supported in part by School of Information and Computer Science, ICS, of Georgia Institute of Technology and by the Air Force of the Scientific Research (AFOSR) under Grant AFOSR-88-0028.

an approximation for cyclic networks with blocking which calculates throughput values as a function of the number of jobs. They initially calculate throughput values for certain populations and then generate a function which fits the determined points. Equivalencies between closed networks with different blocking mechanisms are studied by Onvural/Perros [17] where they show if the number of jobs in a network with transfer blocking is one more the capacity of the node with the smallest capacity there is an exact product form solution.

Our work is mostly motivated by the studies of [8, 9, 18, 22]. Although these studies are focused on open queueing networks with blocking, the concept of constructing phases in order to represent blocking events helped us to execute the parametric analysis of closed queueing networks with blocking. In recent years several other investigators have published results on queueing networks with blocking. A bibliography concerning queueing network models with blocking is given by Perros [20]. A recent workshop gives also a good overview about the area of queueing networks with blocking [19].

2. Model Assumptions

We consider closed queueing networks with N nodes and K total jobs. The service time at node i is exponentially distributed with mean value $1/\mu_i$ (for $i = 1,...,N$). The scheduling discipline at each node is assumed to be FCFS. Each node has a fixed finite capacity B_i where $B_i = (queue\ capacity + 1)$, (for $i = 1,2,...,N$). Cases in which the nodes can have infinite capacity are also allowed, ($B_i = \infty$), (for some $i = 1, 2, .., N$). Any node whose capacity exceeds the total number of jobs in the network can be considered to have infinite capacity. A job which is serviced by the i-th node proceeds to the j-th node with probability p_{ij}, (for $i, j = 1, 2, \cdots, N$), if the j-th node is not full. That is, if the number of jobs in the j-th node, k_j, is less to B_j for $j = 1,2,...,N$. Otherwise, the job is blocked in the i-th node until a job in the j-th node has completed its servicing and a place becomes available. Furthermore it is valid that

$$K < \sum_{i=1}^{N} B_i \qquad (1)$$

which implies that the total number of jobs K in the network may not exceed the total capacity of the entire network.

One ot the most important problems to realize regarding blocking queueing networks is that finite node capacities and blocking can introduce the problem of system deadlock. Deadlock may occur if a job which has finished its service at node i's server wants to join node j, whose capacity is full. That job is blocked in node i. Another job which has finished its service at j-th node now wants to proceed to the i-th node, whose capacity is also full. It blocks node j. Both jobs are waiting for each other. As a result a deadlock situation arises. The following assumption states that a closed queueing network containing finite node capacities is deadlock free if and only if for each cycle C in the network the following condition holds [1]:

$$K < \sum_{j \in C} B_j \qquad (2)$$

Simply stated, the total number jobs in the network must be smaller than the sum of node capacities in each cycle. Since tandem queueing networks have only one cycle, this condition, equation (2), corresponds to equation (1). Equation (1) is a sufficient condition for tandem networks to be deadlock free.

3. Norton's Theorem Application on Queueing Networks

The parametric analysis is based on an application of Norton's Theorem from electrical circuit theory to queueing networks. Chandy, Herzog and Woo [11] showed that Norton's Theorem provides an exact analysis for product form queueing networks [10].

For this type of queueing network models an equivalent network is constructed where a node σ is arbitrarily selected. All other $(N - 1)$ nodes, which we refer as the subnetwork Γ, are replaced by a single node, called the composite (flow-equivalent) node. The total throughput of the original network can be computed with the load-dependent throughput $\lambda(k)$ (for all jobs $k = 1,2,..,K$) of the subnetwork and the service rate μ_σ of the selected node. As mentioned before, this method provides exact results for product form queueing networks. Queueing networks with blocking do not possess exact product form solution due to interdependencies between nodes caused by finiteness of capacities. Consequently the application of Norton's theorem on queueing networks with blocking will provide approximate results.

The application of Norton's theorem on queueing networks with blocking is executed in five steps.

3.1. Construction of the Subnetwork

3.2. Computation of Throughput of the Subnetwork

3.3. Construction of the Phases for the Selected Station

3.4. Computation of Service Times and Branching Probabilities for Phases

3.5. Analysis of the Two-Station Network

These steps will be explained in detail in the following sections.

3.1. Construction of the Subnetwork

We obtain the subnetwork Γ by shortening the selected node σ, i.e., setting its service time equal to zero as in the case of infinite capacity queueing networks. However, in case of finite capacity queueing networks we have to take the blocking events into account which occur on the paths between the selected node σ and the nodes of the subnetwork Γ. Therefore we assume the capacity of nodes in the subnetwork Γ to be infinite which are direct successors of node σ. However, if a node is a successor of σ and receives arrivals from other nodes within the subnetwork Γ its capacity remains unchanged. By this way the subnetwork Γ is converted to subnetwork Γ' which is explained formally as follows:

Let $\Psi = \{ n_1, n_2, ..., n_N \}$ be the set of all nodes in the originallly given network and Ψ_i be the set of all nodes which contains all successors of node i, i.e.,

$$(\forall\ i, j)\ \left[(1 \le i, j \le N\ \&\ i \ne j\ \&\ p_{ij} > 0) \rightarrow (n_j \in \Psi_i) \right] \qquad (3)$$

Subnetwork Γ is transformed into Γ' as follows:

$$(\forall\ j, m)\ \left[(1 \le j, m \le N\ \&\ j, m \ne \sigma\ \&\ n_j \in \Psi_\sigma\ \&\ n_j \notin \Psi_m) \rightarrow (B_j := \infty) \right] \qquad (4)$$

We give an example to illustrate this theory. The queueing model is given in Figure 1 where we assume that there are $N = 7$ nodes, $K = 12$ jobs and the capacity of each node is selected as $B_1 = 3$, $B_2 = 4$, $B_3 = 4$, $B_4 = 3$, $B_5 = \infty$, $B_6 = 3$, $B_7 = 7$.

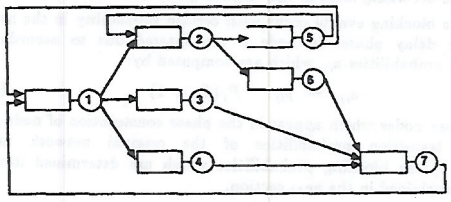

Figure 1.

We select node 1 to be analyzed under various workload. For that reason we shorten node 1, i.e., we set its service time equal to zero and obtain the subnetwork Γ given in Figure 2.

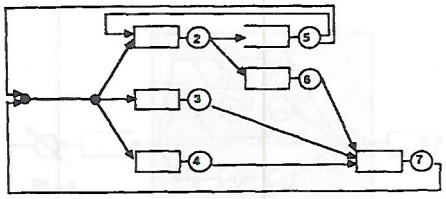

Figure 2.

In subnetwork Γ it can easily be seen that nodes 2,3 and 4 are direct successors of the selected node $\sigma = 1$. Since node 2 receives also arrivals from node 5 it keeps its finite capacity while nodes 3 and 4 receive arrivals only from σ. It follows that their capacity are assumed to be infinite in Γ' as given in Figure 3. Everything else in Γ' remains the same as in Γ.

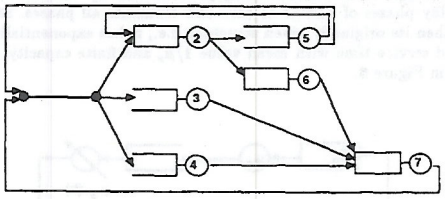

Figure 3.

For substituting Γ' by a composite (flow-equivalent) node we need to compute throughput values $\lambda(k)$ of Γ'.

3.2. Throughput Analysis of the Subnetwork

In order to obtain the throughput values of the subnetwork Γ' which is a blocking network itself we apply the technique by Akyildiz [5, 7]. The basic concept is that the state space of the blocking queueing network with K total number of jobs is transformed into the state space of a non-blocking queueing network with $\hat{K}$ total number of jobs. The number of states in both networks should be approximately the same, if not identical. This would indicate that Markov processes describing the evolution of both networks over time have approximately the same structure. That, in turn, would guarantee that the throughputs of both networks are approximately equal.

The following steps are executed in order to compute the throughput values in queueing networks with blocking [5, 7].

i) Determine the number of states in the blocking queueing network.

ii) Determine the total number of jobs $\hat{K}$ in the non-blocking queueing network. Note that $\hat{K}$ may be a non-integer number.

iii) Analyze the non-blocking queueing network NB with $\hat{K}$ jobs using the α-MVA [13] which is applicable to queueing networks with non-integer number of jobs, and obtain the throughput values which are approximately equal to the throughput value of the blocking network Γ' with K jobs.

$$\lambda_{\Gamma'}(K) \approx \lambda_{NB}(\hat{K}) \qquad (5)$$

3.3. Phase Construction of the Selected Station

So far we have analyzed the subnetwork Γ' and replaced all nodes in Γ' by a so-called composite (flow-equivalent) node c. The load-dependent service rates $\mu_c(k)$ of this composite node are set equal to the throughput values $\lambda_{\Gamma'}$ of the subnetwork Γ' which are computed as described in section 3.2. We assume the capacity of the composite node to be infinite. Thus, the composite node does not cause blocking at the selected node σ. The behavior of node σ is therefore not dependent on nodes of the subnetwork Γ. Here we show how the interdependencies between selected node σ and subnetwork Γ have to be taken into account.

By assuming the capacity of σ's successor nodes in Γ' to be infinite we neglected the blocking events at node σ which might have occurred due to some full nodes in Γ'. However, the blocking events must be considered for node σ, i.e., a job cannot leave node σ due to some full nodes in the subnetwork. Therefore, we modify the service mechanism of node σ such that all delays a job might undergo due to blocking events in the originally given network can be represented.

For each possible blocking delay at nodes of Γ' we add a service phase to node σ. The connection between the added phases and the original server of node σ are the same as the transitions between the nodes in the originally given network. However, blocking delays may not only be caused by node σ's immediate successors but also by nodes which occur in each cycle of the network where the selected node σ is represented. Let $C_\sigma(l)$ be the l-th cycle that starts and ends at node σ, i.e.,

$$C_\sigma(l) = (\sigma, j_{i_1}, j_{i_2}, \ldots, \sigma)$$

where j_{i_q} is the q-th node in the cycle l. Now let us consider one of these cycles, $(\sigma, j_{i_1}, j_{i_2}, \ldots, \sigma)$. For instance, assume that there are k_i jobs at node σ and the number of jobs in the network be such that nodes j_{i_1} through j_{i_s} can be full at the same time. In this case, a job upon service completion at node σ may find node j_{i_1} full, blocking node i's server. Now the question is when this blocked job will depart from node σ. If upon service completion at node j_{i_1} a job chooses to go to node j_{i_2} which is not full, then the job at node j_{i_1} will depart and at the same time another job at node i will join node j_{i_1} unblocking server of the node σ. However, if a job at node j_{i_2} gets blocked because its destination is full then the blocked job at node i cannot depart. Hence, in the worst case a job at node σ will wait for service completions at nodes $j_{i_1}, \ldots, j_{i_s}$ before leaving node σ. In other words, when we construct the delay phases for node σ we have to take into consideration those nodes j_{i_q} which occur in each cycle $C_\sigma(l)$ of node σ and which might cause blocking of node σ.

In constructing the phases the following rules must be obeyed:

R1) If two or more cycles are identical up to a certain element then the elements prior to that element are represented only once in the phase construction.

R2) If $\Omega_\sigma(l) = (\sigma, j_{i_1}, j_{i_2}, ..., j_{i_\nu})$ is a path in cycle $C_\sigma(l)$ starting from node σ with $\sum_{j_{i_\nu} \in \Omega_\sigma(l)\, \sigma \; j_{i_\nu} \neq \sigma} B_{j_{i_\nu}} \geq K$, then the nodes, $(j_{i_q}, j_{i_{q+1}}, ..., i)$ of $C_\sigma(l)$ are not considered in the phase construction for this cycle. In other words, if the sum of node capacities in $\Omega_\sigma(l)$ exceeds the total number of jobs then the last node of $\Omega_\sigma(l)$ and all its successors in $C_i(l)$ are not taken into account in the phase construction for node σ for that cycle.

Note that due to the deadlock freedom property, equation (2), it is not possible that one node l_{j_q} $(l_{j_q} \neq \sigma)$ occurs more than once in a cycle $C_\sigma(l)$.

Let us continue with the example given in Figure 1 and show how to construct the phases for node 1. As we can see, node 1 is a member of 4 cycles in this model:

$$C_1(1) = (1,2,5,1) \qquad C_1(2) = (1,2,6,7,1)$$
$$C_1(3) = (1,3,7,1) \qquad C_1(4) = (1,4,7,1)$$

Applying rule R1) node 2 is not included twice in the phase construction, i.e., in one case as a member of $C_1(1)$ and in the other case as a member of $C_1(2)$. The rule R1) does not hold for node 7 in $C_1(3)$ and $C_1(4)$ because these two cycles are not identical until the position of node 7, i.e., $(1,3,7) \neq (1,4,7)$. According to rule R2) we consider only node 2 out of the elements of $C_1(1)$ for the phase construction of node 1, since the property $\{B_5 = \infty\}$ holds. Considering $C_1(2)$ we see that $\{B_2 + B_6 + B_7 > K\}$ is satisfied. Hence, node 7 is not included in the phase construction for $C_1(2)$. In Figure 4 we show the complete phase construction for node 1.

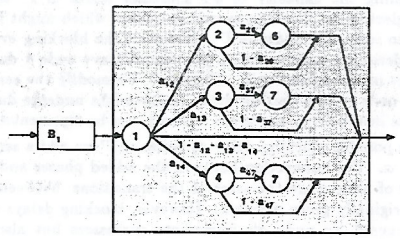

Figure 4.

3.4. Computation of Service Times and Branching Probabilities for Phases

After the construction of phases we need to determine the parameters such as branching probabilities and service times of the phases. In the transfer blocking case, a blocked job cannot leave the node until a place is available in the destination node. Therefore, the blocking time of a job is equal to the mean remaining service time of the particular destination node [6]. For an $(M\ /G\ /\ 1\ /\ FCFS)$ queue the mean remaing service time $\overline{r}$ is given by [14]:

$$\overline{r} = \frac{\overline{x^2}}{2\,\overline{x}} \tag{6}$$

where $\overline{x}$ and $\overline{x^2}$ denote the first and the second moment of the service time distribution.

For nodes with exponentially distributed service times we have $\overline{x} = \frac{1}{\mu}$ and $\overline{x^2} = \frac{2}{\mu^2}$ which gives us:

$$\overline{r} = \frac{1}{\mu} \tag{7}$$

Based on this argument the pseudo service time of each phase which represents the blocking events, is equal to the mean service time of the according node in the original given network.

Since blocking events occur with certain probability in the network, the delay phases in node i are entered due to according branching probabilities a_{ij} which are computed by:

$$a_{ij} = p_{ij} \cdot P_j(B_j + 1) \tag{8}$$

where j are nodes which appear in the phase construction of node i, p_{ij} are transition probabilities of the original network and $P_j(B_j + 1)$ are blocking probabilities which are determined iteratively as explained in the next section.

3.5. Analysis of the Two-Station Network

So far we reduced the entire network to two-node network which contains finite capacity node σ with complete constructed phases and the composite (flow-equivalent) infinite capacity node c representing all other nodes. The two-node network for the example given in Figure 1 has the following structure:

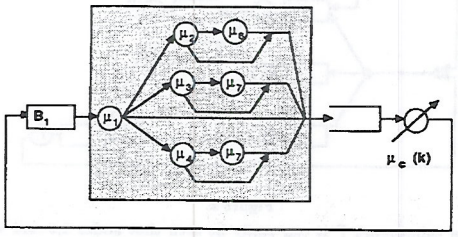

Figure 5.

As pointed out in equation (8) the blocking probabilities $P_j(B_j + 1)$ need to be determined for the analysis of the two-node network. These parameters $P_j(B_j + 1)$ and the desired throughput values λ of this two-node network given in Figure 5 which are also the throughput values of the originally given network, are computed by an iterative way.

Initially we set all branching probabilities a_{ij} between service and delay phases of node σ to zero and eliminate all phases. Station σ has then its originally given structure, i.e., it has exponentially distributed service time with mean value $1/\mu_\sigma$ and finite capacity B_σ as shown in Figure 6.

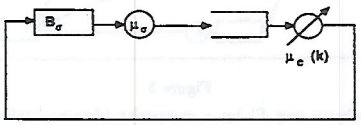

Figure 6.

This network is analyzed by the method given in [5] and the total throughput λ is computed. Using λ the throughput of each node j is determined by

$$\lambda_j = \lambda \cdot e_j \quad \text{for} \quad j = 1,.....,N \tag{9}$$

where e_j is the mean number of visits that a job makes to node j and is given by

$$e_j = \sum_{i=1}^{N} e_i \cdot p_{ij} \qquad \text{for} \quad j = 1,.....,N$$

Now these computed throughput values λ_j are used as the arrival rates for $[\ M\ /M\ /1\ /\ B_j + 1\ ]$ nodes which are considered in the phase construction of node σ. Here we assume that each node in the originally given network behaves approximately as a single server node with exponentially distributed service times and Poisson arrivals. Recall that a job in the transfer blocking protocol is already served, a destination is chosen and the job is blocked in the server. Logically this blocked job occupies the $(B_j + 1)$st place in the queue of the jth destination node. The probability that the $(B_j + 1)$th space in node j is occupied, provides the probability that one or more predecessors of node j are blocked in the originally given network.

Hence, using the well-known formula for steady state probabilities of $[M\ /M\ /1\ /\ B_j]$ nodes the values for the blocking probabilities $P_j(B_j + 1)$ are computed [14]:

$$P_j(B_j+1) = \begin{cases} \hat{\rho}_j^{B_j+1} \cdot \dfrac{1-\hat{\rho}_j}{1-\hat{\rho}_j^{B_j+2}} & \text{if } B_j < K \\ 0 & \text{if } B_j \geq K \end{cases} \quad (10)$$

with

$$\hat{\rho}_j = \frac{\hat{\lambda_j}}{\mu_j} \qquad \text{for } j = 1,...,N \quad (11)$$

where $\hat{\lambda_j}$ is the effective input rate to node j which can be expressed in terms of the arrival rate λ_j at node j:

$$\hat{\lambda_j} = \lambda_j \cdot \Big[\ 1 - P_j(B_j+1)\Big] \qquad \text{for } j = 1,...,N \quad (12)$$

Equations (10) and (12) are used as a fixpoint iteration for computation of $P_j(B_j + 1)$ values. Note that the convergency of this fixpoint iteration was shown by Altiok [8] for open queueing networks with blocking. The values for $P_j(B_j + 1)$ are then used to determine the branching probabilities a_{ij} from equation (8).

Let us continue here to discuss the example given in Figure 1. Since nodes 2,3,4,6 and 7 appear in the phase construction for node 1 we consider these as individual $[M/M/1/B_j + 1]$ nodes:

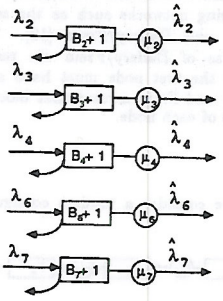

Figure 7.

Using the fixpoint iteration equations (10) and (12) we compute $P_j(B_j + 1)$ values. which are then used for determining the branching probabilities a_{ij} of delay phases in equation (8).

$$a_{12} = p_{12} \cdot P_2(B_2+1)$$
$$a_{26} = p_{26} \cdot P_6(B_6+1)$$
$$a_{13} = p_{13} \cdot P_3(B_3+1)$$
$$a_{14} = p_{14} \cdot P_4(B_4+1)$$
$$a_{37} = p_{37} \cdot P_7(B_7+1)$$
$$a_{47} = p_{47} \cdot P_7(B_7+1)$$

The two-node network has now a complicated structure since the branching probabilities a_{ij} in σ are not zero as shown in Figure 5. In order to analyze this type of networks efficiently we reduce the serving and delay phases of node σ to a Coxian server with two phases. Marie [15] showed that this transformation provides good results for nodes with general service time distribution if the squared coefficient of variation is greater than 0.5. Once all the parameters of the phase type distribution illustrated in Figure 5 are known we can construct a Coxian-2 representation by determining its first moment $1/\hat{\mu}_\sigma$ and the coefficient of variation $\hat{c}_\sigma$ and then fitting a Coxian-2 distribution according to the following equations:

$$\tilde{\mu}_{\sigma 1} = 2 \cdot \hat{\mu}_\sigma \quad (13)$$

$$\tilde{\mu}_{\sigma 2} = \hat{c}_\sigma^{\,2} \cdot \hat{\mu}_\sigma \quad (14)$$

$$\tilde{a}_{\sigma 1} = \frac{1}{2 \cdot \hat{c}_\sigma^{\,2}} \quad (15)$$

This transformation leads to the following queueing network model:

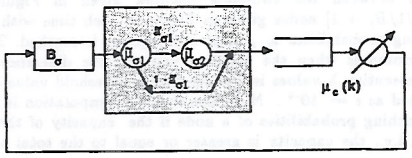

Figure 8.

This network can be analyzed by the numerical method [24] or by the load-dependent method of Marie [2]. Note that these techniques must be slightly modified because one node has finite capacity and blocking can occur in the network. Additionally, we have to consider that a job in node σ's server can be in two phases. In Figure 9 we show the state space diagram for the network given in Figure 8. The state $(l\,;p\ ,\ n\,)$ describes the situation where l jobs are in node σ, the job being served in node σ is in the p-th phase and n jobs are in the composite node. Blocking states are labeled with a 'ω'

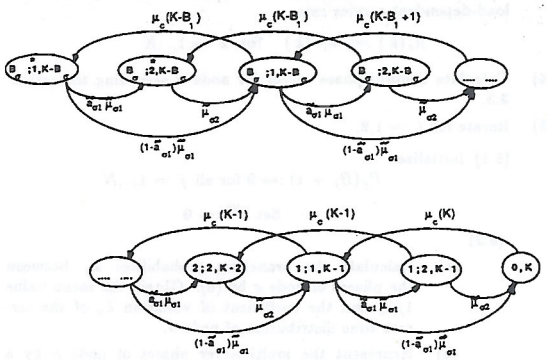

Figure 9.

A job after being served in the first phase of node σ can either proceed to the second phase (with probability $a_{\sigma 1}$), equation (15), or leave the node to join the composite node c. Recall that the composite node c has a load-dependent service rate. For the example given in Figure 1 where $B_1 = 3$ and the total number of jobs is $K = 12$ we obtain the following state space diagram:

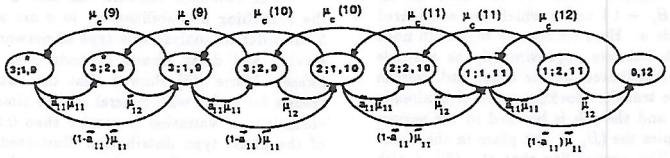

Figure 10.

From this diagram global balance equations can easily be derived and steady state probabilities can be obtained using the numerical technique or the load-dependent method of Marie. From the steady state probabilities the throughput values are computed. As mentioned before these throughput values are then used as the individual arrival rates λ_j for $[\,M/M/1/B_j+1\,]$ nodes and the values of $P_j(B_j+1)$ are computed iteratively from (10) and (12). The values for $P_j(B_j+1)$ are then used for determining the branching probabilities a_{ij}, equation (8) in the multi-phase server of Figure 4. Here the scheme of the iteration can be recognized. We repeat the analysis between the two-node network given in Figure 8 and $[M/M/1/B_j+1]$ nodes given in Figure 7, each time with modified branching probabilities a_{ij}, until convergency is reached. The iteration terminates when the absolute value of the difference between two consecutive λ values is smaller than a threshold value ϵ where ϵ is selected as $\epsilon = 10^{-4}$. Note also that no computation is necessary for branching probabilities of a node if the capacity of that node is infinite, i.e., the capacity is greater or equal to the total number of jobs.

4. Algorithm Summary

(1) Select an arbitrary node σ, set its service time equal to zero $(1/\mu_\sigma = 0)$ and obtain subnetwork Γ.

(2) Transform the subnetwork Γ into Γ' according to expression (4).

(3) Analyze Γ' with the throughput algorithm [5, 7] for finite capacity queueing networks and obtain $\lambda_{\Gamma'}(k)$ for $k = 1,..,K$. Construct the composite node c with infinite capacity and load-dependent service rate:

$$\mu_c(k) := \lambda_{\Gamma'}(k) \quad \text{for } k = 1,..,K$$

(4) Calculate a multi-phase server for node σ according to section 3.3.

(5) Iterate for $t = 1,2,...$

(5.1) Initialize

$$P_j(B_j+1) := 0 \text{ for all } j = 1,..,N$$

$$\text{Set } \lambda^{(0)} := 0$$

(5.2)

(a) Calculate the branching probabilites a_{ij} between the phases of node σ by (8). Obtain the mean value $1/\hat{\mu}_\sigma$ and the coefficient of variation $\hat{c}_\sigma$ of the service time distribution of node σ.

(b) Represent the multi-server phases of node σ by a Cox-distribution with two phases from equations (11, 12) and (13).

(5.3) Solve the two-node network containing Cox-two server and the composite node c and obtain throughput $\lambda^{(t)}$.

(5.4) For each node j which has a server in the phase of node σ solve the fixpoint iteration from equations (8) and (10).

(5.5) Terminate if $|\lambda^{(t)} - \lambda^{(t-1)}| < \epsilon$. Otherwise assign

$$\lambda^{(t-1)} := \lambda^{(t)}$$

and go to step (5.2).

5. Examples

After the termination of the iteration we obtain the total throughput of the two-node network which is also the total throughput of the originally given network. The advantage of the parametric analysis can be recognized when we study the entire network by changing only the parameters of the selected node σ. In that case we need to compute throughput values of subnetwork Γ' only once as described in section 3.2 which remain unchanged throughout the analysis. All we need to do is to construct the phases of node σ in the two-node network as described in section 3.4. The required computational effort is small since the equivalent network consists of only two nodes.

In this section we demonstrate the application of the parametric analysis for blocking networks by analyzing two models. The first model is an end-to-end communication network modeled by a tandem configuration with five nodes. The second model is a general communication network where all nodes are connected with each other. For each example, we illustrate the steps of the method and give numerical results. For checking the accuracy of our technique we compare our results with simulation values obtained by using RESQ package [23]. Relative errors $\delta\%$ are calculated by:

$$\delta\% = \frac{|\ \textit{simulated value} - \textit{analytical value}\ |}{|\ \textit{simulated value}\ |} \cdot 100$$

We also give the number of necessary iteration steps in order to demonstrate the speed of convergency of the phase construction. We also compare our results with results obtained by other algorithms for finite capacity queueing networks such as the methods of Akyildiz [5], Suri/Diehl [25] and Dallery/Frein [12]. The techniques of Suri/Diehl as well as of Dallery/Frein are restricted to tandem configurations where the first node must have an infinite capacity. Additionally, Dallery and Frein assume that blocking occurs only at immediate successors of each node.

Example 1.

In this case we consider a tandem configuration as given in Figure 11a.

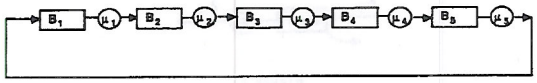

Figure 11a.

Network parameters are:

i	1	2	3	4	5
B_i	∞	3	3	3	3
$1/\mu_i$	1	0.5	1	0.5	1

Here we assume the capacity of the first node to be infinite. Therefore we will be able to compare our technique with the methods of Suri/Diehl [25] and Dallery/Frein [12] since their techniques are applicable only on this type of networks.

The subnetwork Γ' is shown in Figure 11b where we shorten node 1.

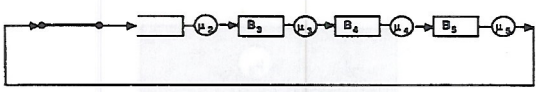

Figure 11b.

Analyzing the subnetwork Γ' we obtain the following throughput values:

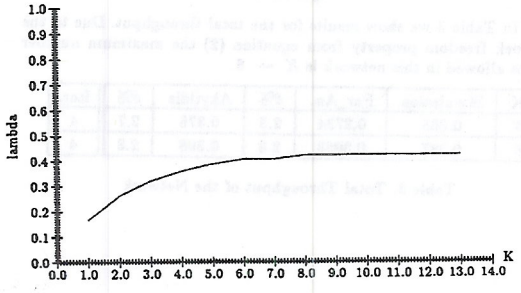

Figure 11c.

The phase construction for selected node 1 is given as follows:

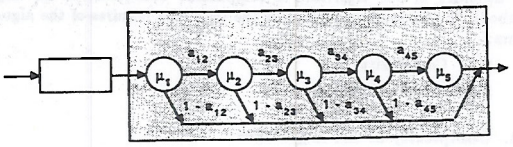

Figure 11d.

Final results are listed in the following Table:

K	Simulation	Par_An.	$\delta\%$	AKYL	$\delta\%$	SUDI	$\delta\%$	DAFR	$\delta\%$	Iter.
5	0.367	0.3658	0.3	0.367	0.0	0.364	0.8	0.356	3.0	4
6	0.389	0.3883	0.1	0.390	0.3	0.381	2.1	0.382	1.8	4
7	0.404	0.3975	1.6	0.407	0.7	0.392	3.0	0.400	1.0	4
8	0.413	0.4091	0.9	0.420	1.7	0.398	3.6	0.416	0.7	4
9	0.417	0.4144	0.6	0.420	0.7	0.400	4.2	0.426	2.1	3
10	0.419	0.4168	0.5	0.420	0.2	0.400	4.3	0.430	2.6	3
11	0.419	0.4178	0.2	0.420	0.2	0.401	4.3	0.430	2.6	3
12	0.419	0.4182	0.2	0.420	0.2	0.401	4.3	0.430	2.6	3
13	0.419	0.4182	0.1	0.430	2.6	0.401	4.3	0.430	2.6	3

Table 1. Total Throughput of the Network

Let us modify the parameters for the tandem network given in Figure 11a.

i	1	2	3	4	5
B_i	2	4	3	4	2
$1/\mu_i$	1	2	1.5	1.8	1.6

The structure of subnetwork Γ' and the multi-phase server of the selected node 1 are the same as given in Figure 11b and Figure 11d, respectively. Since the parameters of nodes in subnetwork Γ' are modified, the subnetwork Γ' must be analyzed again. We obtain the following throughput values for subnetwork Γ' :

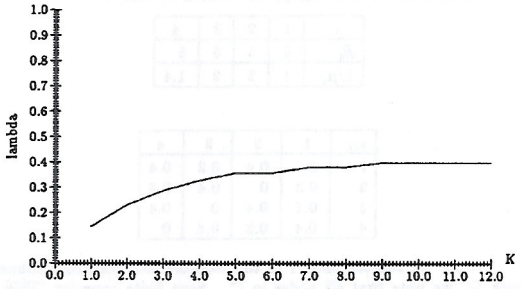

Figure 11e.

In Table 2 we compare our results with simulation and the method of Akyildiz. Since the capacity of node 1 is finite the methods of Suri/Diehl and Dallery/Frein cannot be applied.

K	Simulation	Par_An.	$\delta\%$	Akyildiz	$\delta\%$	Iter.
8	0.388	0.3631	6.4	0.3871	2.3	5
10	0.393	0.3775	3.9	0.3871	1.5	4
12	0.376	0.3787	2.3	0.3871	2.95	5
14	0.352	0.3787	7.5	0.3871	4.29	5

Table 2. Total Throughput of the Network

Example 2.

The queueing model of the second communication network has the following structure:

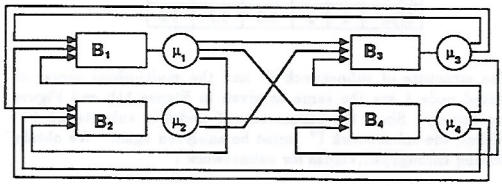

Figure 12a.

Parameters for this example are chosen as follows:

i	1	2	3	4
B_i	5	5	5	5
$1/\mu_i$	1	2	2	1.4

p_{ij}	1	2	3	4
1	0	0.4	0.2	0.4
2	0.3	0	0.4	0.3
3	0.2	0.4	0	0.4
4	0.4	0.2	0.4	0

When we shorten node 1 in this network and construct subnetwork Γ', we note that all nodes in Γ' have finite capacity. This is the case where each node receives arrivals from other nodes within Γ.

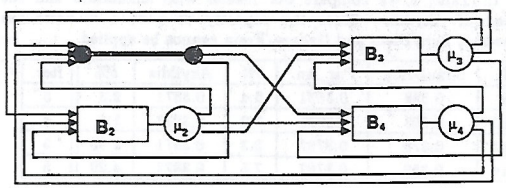

Figure 12b.

Analysis of Γ' provides the following throughput values:

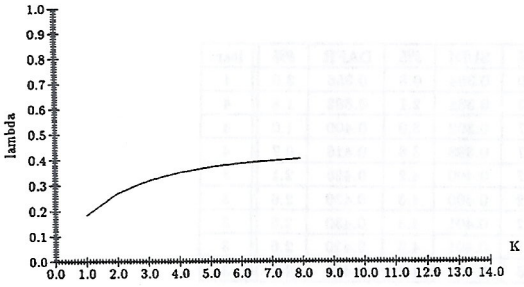

Figure 12c.

In this network blocking may occur only at immediate successor of each node. Otherwise deadlock freedom is not guaranteed [1]. Therefore we obtain a multi-phase server for node 1 as follows:

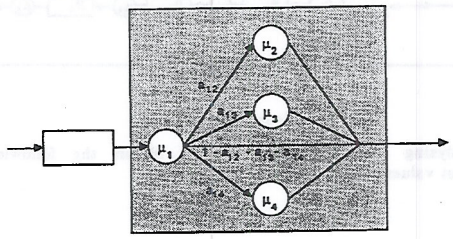

Figure 12d.

In Table 3 we show results for the total throughput. Due to the deadlock freedom property from equation (2) the maximum number of jobs allowed in this network is $K = 8$.

K	Simulation	Par_An.	δ%	Akyildiz	δ%	Iter.
6	0.365	0.3734	2.3	0.375	2.7	4
8	0.387	0.3963	2.8	0.398	2.8	4

Table 3. Total Throughput of the Network

6. Evaluation

In the previous section we discussed several examples and compared the results obtained by our approach with the results determined by other existing approximate techniques. It can be seen that the accuracy of our algorithm is comparable with those of existing methods. In this section we discuss the following features of the algorithm:

6.1. Complexity

6.2. Iterative behavior

6.1. Complexity of the Algorithm

Similar to parametric analysis for infinite capacity networks the advantages of the presented method become obvious if large scale queueing networks are analyzed. If a network with a large number of nodes is to be studied under various workload of a particular node, then the analysis of the entire network is costly. With conventional methods each time any parameter is varied, the total network must be reanalyzed. Applying parametric analysis we have to analyze just a two-node network once the throughput values of subnetwork Γ' are known.

The reduction of the complexity of the analyzed network is demonstrated with a comparison of the size of the state space of the original blocking network and the two-node network, respectively. The state space size of the the two-node network is computed by:

$$Number\ of\ states = 2 * \min(K, B_1+1) + 1 \qquad (16)$$

Note that the factor 2 is necessary because the finite capacity node σ has two server phases. The size of the state space for a transfer blocking network with N nodes can be computed approximately from the last component $Z'(K)$ of the following computation:

$$Z' = Z_1 * Z_2 \cdots * Z_N \qquad (17)$$

where

* is the convolution operator.

Z_i for $i = 1, 2, \ldots, N$ is a $(K+1)$-dimensional vector which is given by:

$$Z_i = \begin{bmatrix} a_i(0) \\ a_i(1) \\ \cdots \\ \cdots \\ a_i(K) \end{bmatrix}$$

with the binary function

$$a_i(k) = \begin{cases} 1 & \text{for} \quad k = 0, 1, 2, \ldots, M+1 \\ 0 & \text{for} \quad k = M_i+2, \ldots, K \end{cases}$$

If we apply both formulas to the network given in Figure 1 with the parameters provided in section 3.1 we see that the two-node network has $Z = 9$ states and the original network of Figure 1 has $Z = 10888$ states.

6.2. Iterative behavior

The examples given in section 5 clearly demonstrated the fact that the iteration terminates very quickly. Indeed, the highest number of iteration needed was 5 iterations. However, for other examples, the number of iterations can be larger based on our experiments. Solving a central server model described as follows:

$N =$ 4 nodes and $K =$ 7 jobs

i	1	2	3	4
B_i	∞	2	3	5
μ_i	8	2	3	2

Transition probabilities are:

$p_{12} = 0.25$; $p_{13} = 0.25$; $p_{14} = 0.5$; $p_{j1} = 1$ for $j = 2,3,4$.

the algorithm converges after 13 iteration steps. The values obtained during the iteration are shown in the following Figure.

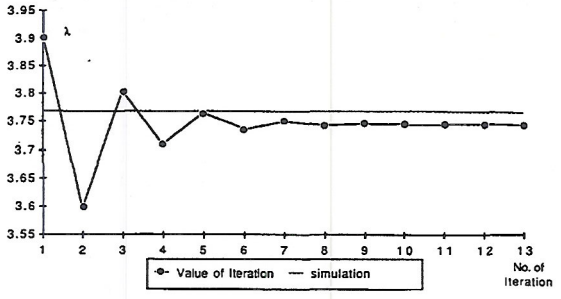

Figure 13.

The pattern of the throughput values obtained during each iteration is the same for all calculated values. In the first step, the branching probabilities in the multiphase server of the selected node are set to zero which gives the network shown in Figure 3. No blocking delays are considered and therefore, the throughput value in the first step is higher than in the following iteration. Since a high throughput value provokes high probabilities $P_j(B_j+1)$ the second iteration has the lowest throughput value of all iterations.

We note that in our experiments we could not find any case where the algorithm did not converge.

7. Conclusions

In this work we introduced a method which shows the application of Norton's theorem on queueing networks with so-called transfer blocking. We assumed that the networks investigated must be deadlock-free. We proposed an algorithm for queueing networks with finite capacities which enables us to profit from the advantages of applying Norton's theorem, i.e., selecting a node arbitrarily and analyzing the network by modifying the parameters of the selected node without repeating the analysis of the rest of the network. The subnetwork is replaced by a composite node with infinite capacity. We represented blocking delays in the selected node by phases. In order to compute the parameters of the constructed phases we introduced an iterative technique. The iteration was executed between different views of the network, i.e. the two-node network with delay phases in the selected node and each node in the originally given network as independent $M/M/1$ nodes with a finite capacity. We discussed numerical examples and compared our results with simulation as well as with other existing techniques for blocking networks.

References

1. I. F. Akyildiz and S. Kundu, "Deadlock Free Buffer Allocation in Closed Queueing Networks", *to appear in Queueing Systems Journal.*

2. I. F. Akyildiz and A. Sieber, "Approximate Analysis of Load-Dependent General Queueing Networks", *IEEE Trans. on Software Eng., Vol. 14, No. 11, Nov. 1988, pp. 1537-1545.*

3. I. F. Akyildiz, "Exact Product Form Solution for Queueing Networks with Blocking", *IEEE Trans. on Comp.*, Vol. 1, Jan. 1987, pp. 121-127.

4. I. F. Akyildiz, "General Closed Queueing Networks with Blocking", *Performance 87, Proc. North Holland , pp. 283-303.*

5. I. F. Akyildiz, "On the Exact and Approximate Throughput Analysis of Closed Queueing Networks with Blocking", *IEEE Trans. on Software Eng., Vol. SE-14, No. 1, Jan. 1988, pp. 62-71.*

6. I. F. Akyildiz, "Mean Value Analysis for Blocking Queueing Networks" *IEEE Trans. on Software Eng., Vol. SE-14, No. 4, April 1988, pp. 418-429.*

7. I. F. Akyildiz, "Product Form Approximations for Queueing Networks with Multiple Servers and Blocking", *IEEE Trans. on Comp., Vol. 15, No. 1, Jan. 1989, pp. 99-114.*

8. T. Altiok, "Approximate Analysis of Exponential Tandem Queues with Blocking", *European Journal of Operations Research*, Vol. 11, 1982, pp. 390-397.

9. T. Altiok and H. G. Perros, "Approximation Analysis of Arbitrary Configurations of Open Queueing Networks with Blocking", *AIIE Trans.*, March 1986.

10. F. Baskett, K. M. Chandy, R. R. Muntz and G. Palacios, "Open, Closed and Mixed Network of Queues with Different Classes of Customers", *Journal of the ACM*, Vol. 22, Nr. 2, Apr. 1975, pp.248-260.

11. K. M. Chandy, U. Herzog and L. Woo, "Parametric Analysis of Queueing Network Models, *IBM Journal Res. Dev.*, Vol. 19, Nr. 1, Jan. 1975, pp.43-49.

12. Y. Dallery and Y. Frein, "A Decomposition Method for the Approximate Analysis of Closed Queueing Networks with Blocking", *Pre-conf. Proc. First Int. Workshop on Queueing Networks with Blocking*, NCSU, Raleigh, May 1988, pp. 201-223.

13. L. W. Dowdy and K. D. Gordon, "Algorithms for Nonintegral Degrees of Multiprogramming in Closed Queueing Networks", *Performance Evaluation*, Vol. 4, No. 1, Febr. 1984, pp. 19-31.

14. L. Kleinrock, "Queueing Systems, Volume I: Theory", *John Wiley & Sons*, New York, 1975.

15. R. Marie, "Calculating Equilibrium Probabilities for λ(n)/C/1-N Queues", *Proc. Performance and ACM Sigmetrics 80 Conference, Vol. 9, No. 2, May 1980* , pp. 117-125.

16. R. O. Onvural and H. G. Perros, "Throughput Analysis of Closed Queueing Networks with Blocking", *to appear in IEEE Trans. on Software Eng.*.

17. R. O. Onvural and H. G. Perros, "Some Equivalencies for Queueing Networks with Blocking", *to appear in Performance Evaluation Journal.*

18. H. G. Perros and T. Altiok, "Approximate Analysis of Open Networks of Queues with Blocking: Tandem Configurations", *IEEE Trans. on Software Eng.*, Vol. SE-12, No. 3, March 1986, pp. 450-462.

19. H. G. Perros and T. Altiok (Co-Chairmen), "First International Workshop on Queueing Networks with Blocking", *Pre-conference Proceedings* NCSU, Raleigh, May 1988.

20. H. G. Perros, "Queueing Networks with Blocking: A Bibliography", *ACM Sigmetrics Performance Evaluation Review*, August 1984.

21. H. G. Perros, A. A. Nilsson and Y. C. Liu, "Approximate Analysis of Product Form Type Queueing Networks with Blocking and Deadlock", *Performance Evaluation, Vol. 8, Febr. 1988, pp. 19-99.*

22. M. Reiser and S. S. Lavenberg, "Mean Value Analysis of Closed Multichain Queueing Networks", *JACM*, Vol. 27, Nr. 2, April 1980, pp. 313-322.

23. C. H. Sauer, E. A. MacNair, J. F. Kurose, "The Research Queueing Package Version 2", *IBM Yorktown Heights, N.Y., 1982.*

24. W. J. Stewart, "A Comparison of Numerical Techniques in Markov Modelling", *CACM*, Vol. 21, Nr. 2, Febr. 1978, pp. 144-152.

25. R. Suri and G. W. Diehl, "A Variable Buffer-Size Model and its Use in Analyzing Closed Queueing Networks with Blocking", *Management Science*, Vol. 32, No. 2, Febr. 1986, pp. 206-225.