# Robust Box Bounds: Throughput Guarantees for Closed Multiclass Queueing Networks with Minimal Stochastic Assumptions

S. Majumdar C. M. Woodside J. E. Neilson
D. C. Petriu
Telecommunications Research Institute of Ontario
Department of Systems and Computer Engineering, and
School of Computer Science
Carleton University
Ottawa K1S 5B6, Canada
majumdar@sce.carleton.ca



## Abstract

*To use queueing theory to analyze real systems such as computer communications networks, we make assumptions that are strictly speaking untrue. The problem is especially severe for multiclass networks with FIFO service and greatly differing service times. This paper provides an exact analysis for bounds, for cases with greatly relaxed assumptions. Service times can have general (IFR) distributions, different by class even at FIFO nodes. Routing can be arbitrary, including dependencies along the route, provided the number of visits to each node is a random variable. Only the mean service time and mean visit rates at nodes need be specified. A new lower throughput bound is found which gives a minimum guaranteed throughput for each class; together with the familiar multiclass asymptotic upper bounds they give a convex feasible region in a multidimensional throughput space. A detailed analysis is given for systems with FIFO and infinite-server nodes, and the extension to processor-sharing nodes is described.*

*Because the feasible region may be a complicated shape which is difficult to visualize, the results can be re-interpreted as a set of bounds on the separate throughputs. This is equivalent to a circumscribed rectangular region called the "Robust Box Bounds". Computation of these bounds is carried out by a novel technique based on interval arithmetic as implemented in BNR Prolog language. These bounds can be effectively used for the performance analysis of computer networks as well as other types of computing systems.*

## 1 Introduction

Many communications networks can be modelled as multiclass closed queueing networks, as described for example by Reiser [16] or by Stuck and Arthurs [2]. The queueing model can be solved for average performance figures which are useful at several stages in the system lifetime, for design and sizing, for performance tuning, and for network evolution in response to changing workloads. Analytic queueing models are used because they are much quicker and cheaper to use than simulations or benchmarks. However there are several ways in which queueing models are unsatisfactory in practice. For analytic solutions, separable queueing networks:

**(a)** require stochastic assumptions such as knowledge of service time distributions at each network node;

**(b)** require all classes to have exponential service distributions with the same mean service time at a node that uses a FIFO service discipline.

As noted by Reiser [16] the assumptions (a) and (b) are often untrue for computer communication networks. For example, traffic on a number of computer networks has been found to be dominated by packets of two or three different sizes [5, 11] which may lead to a violation of (b). Reiser proposed approximation techniques [16] for the solution of non-separable models of computer networks. Using approximation techniques help, but the error of the result is then unknown. Further, sometimes one does not know the distribution or even the variance of service times, so even the approximations may be unusable.

The approach taken here is not to approximate the throughput, but to bound it above and below by values which are independent of stochastic assumptions such as (a) and (b) above. The bounds are based on minimal stochastic assumptions (see section 2) about system behaviour and for this reason they are termed "robust". The lower bounds then give guaranteed levels of throughput, and the distance between the bounds indicates the variations which might occur under varying distributions or dependencies. The bounds will be found to depend on the mean service times and visit rates.

The novelty of the work is in the lower bounds, which are derived from fundamental and insensitive properties of queues and are not related to separable solutions. Consideration is mostly restricted to FIFO queues because these are difficult to deal with by using other analytical techniques, but processor sharing queues are discussed also. It must be emphasized that the lower bounds presented here are not a "back-of-the-envelope" calculation, to be done before other more detailed calculations. Instead, for many systems with unknown distributions these are the only exact results which can be found. The guaranteed throughput levels could be the basis of a conservative approach to design.

Upper bounds are also discussed for completeness, by generalizing the analysis of the well-known single class asymptotic bounds to multiple classes. The upper bounds presented in this paper are potentially tighter than the generalized version of the asymptotic upper bounds presented in [6].

Queueing network models are used to model computer networks as well as various other types of computing systems. This paper focuses on the computation of performance bounds for closed general multiclass queueing networks that can be used for the performance analysis of computer communication networks as well as other systems. Two sets of bounds are described in this paper. The exact bounds are defined by a complicated set of inequalities and are somewhat difficult to use. The "box" bounds are a set of bounds on the throughput of each class, which are easier to use although they are usually somewhat looser.

Existing research on performance bounds has spanned both the domains of single and multiclass queueing networks. A substantial literature attests to the importance of the topic; only a representative set of work is presented here. Previous research on the development of performance bounds began with single class queueing networks. Muntz and Wong carried out the study that led to the development of the well known asymptotic bounds that are applicable to both separable as well as non-separable queueing networks [15]. The asymptotic upper bound is based on the optimistic assumption that a customer experiences minimum contention at any device whereas the lower bound is based on a pessimistic assumption that a customer is forced to queue behind all the other customers at every device. Balanced Job Bounds (*BJB*) that are tighter than the asymptotic bounds but are applicable to separable queueing networks only are presented by Zahorjan *et al.* in [19]. The Balanced Job Bounds are derived by considering a balanced queueing network with equal service demand for each device. Eager and Sevcik have proposed a hierarchy of bounds with different accuracy levels and computational complexity for separable single class queueing networks [4]. Bound hierarchies that offer a tradeoff between computational cost and accuracy are also considered by Kriz [8], by Suri [17] and Stephens and Dowdy [9]. The algorithms presented in [4], [8], and [17] are based on the mean value analysis recursion equations whereas the bounds presented in [9] are based on the convolution algorithm recursion.

The work on bounding analysis was continued to cover the domain of multiclass queueing networks. For example, extension of BJBs to multiclass networks are proposed by Zahorjan *et al.* [19]. Hierarchies of bounds for multiclass queueing networks are derived by Eager and Sevcik in [3]. Hsieh and Lam [18] have described an iterative technique that is similar to the algorithm in [3]. A Composite Bound Method which uses the BJB lower bounds for obtaining an upper bound on throughput is outlined by Kerola in [6].

Of the papers cited, all but [15] are restricted to separable networks. This is a serious restriction, as real systems typically are not separable. For applying these bound analyses for separable models a number of assumptions need to be made about the nature of the system. As stated above, this paper attempts to fill this gap, by presenting a complete set of bounds for a very general class of queueing networks. The asymptotic upper bound on throughput for single class queueing networks is recast in a form that is suitable for multiclass queueing networks. The main contribution of the paper is the derivation of an upper bound for the delay at a FIFO queue, leading to lower bounds for throughput. The bounding technique developed for queueing networks with FIFO nodes can be extended to include infinite servers and processor sharing nodes in the network. The bounds on the throughput of a given customer class is expressed in terms of the throughputs of other classes of customers. Since the throughput of every customer class is in general unknown it is difficult to extract the bounds in closed analytic form. A technique based on interval arithmetic is introduced in the paper for the computation of these bounds. This research focuses on the development of the bounding technique; issues such as the relationship of the tightness of the bounds with system parameters can lead to important insights into system behaviour but are beyond the scope of this paper.

The paper is organized as follows. The upper and lower bounds on throughput are presented in Section 2. The interval arithmetic concept and its use in the computation of the bounds are described in Section 3. Application of the bounding technique to queueing network models of example computer communication networks is presented in the latter part of this section. A graphical presentation of the feasible throughput space that is useful for systems with a small number (typically two or three) of customer classes is described in Section 4. A brief summary of the main contributions of the paper is presented in Section 5.

## 2 Bounds on Throughput

The upper and lower bounds on the throughputs for different customer (job) classes in a closed multiclass queueing network with a FIFO queueing discipline used at each device (queueing centre) are presented in this section. Before presenting the bounds the necessary terminology is introduced. As usual the operation of the system is defined in terms of the execution of response cycles by the customers, in which each device is visited a certain number of times (with average $V_{kc}$ defined below) and for which the customer takes an average of $R_c$ seconds. Let,

$C$ = number of customer classes

$K$ = number of devices (queueing centres)

$N_c$ = Number of class $c \in (1 \ldots C)$ customers

$V_{kc}$ = number of visits made by a class $c$ customer at device $k \in (1 \ldots K)$ (visits/cycle)

$S_{kc}$ = service demand per visit for a class $c$ customer at device $k$ (seconds/visit)

$Z_c$ = think time (pure delay) for a class $c$ customer (seconds/cycle)

$R_c$ = cycle time for a single customer in class $c$ (seconds)

$f_c = 1/R_c$ = cycle rate for a single customer in class $c$

$X_c$ = throughput for class $c$ (cycles/second); note that $X_c = N_c f_c$.

*Minimal Stochastic Assumptions*:
The first six items of this list serve as the input parameters for the model whereas $X_c$ ($c = 1 \ldots C$) is the model output. Only the first moments for the input parameters, $V_{kc}$, $S_{kc}$, $Z_c$, are required for the computation of the bounds. The bounds do not depend on the nature of the distributions underlying the device service times and the number of visits made by a customer to a specific device. Only the assumptions that the device service times are independent with any general IFR (Increasing Failure Rate) distribution and the number of visits made by customers to any device are independent random variables are required for the existence of the bounds.

There are three components of the cycle time, $R_c$, for a class $c$ customer: the time taken for its own service at the different devices, the queueing delay experienced by the customer, and its think time. Expressions for upper and lower bounds on $R_c$ are derived in the following subsections. The upper and lower bounds on $X_c$ are computed from the lower and upper bounds on $R_c$ respectively.

### 2.1 Upper Bounds

A simple extension of the single class asymptotic upper bounds on throughput to a multiclass queueing network is presented in this section. The upper bound on the throughput of a class $c$ customer are of two types. These upper bounds are derived in a manner similar to the derivation of the asymptotic bounds. The first type of upper bound ignores contention at the devices and is termed the *no contention* upper bound. By ignoring the queueing delay at every device $k$ we obtain:

$$R_c \geq Z_c + \sum_{k=1}^{K} V_{kc} S_{kc}$$

and,

$$f_c = \frac{1}{R_c} \leq \frac{1}{Z_c + \sum_{k=1}^{K} V_{kc} S_{kc}} \tag{1}$$

Thus,

$$X_c = N_c f_c \leq \frac{N_c}{Z_c + \sum_{k=1}^{K} V_{kc} S_{kc}} \tag{2}$$

The second type of upper bound called the *utilization-based* upper bound is based on the fact that the utilization of a device cannot exceed one. Thus for every device $k \in (1 \ldots K)$ we have:

$$\sum_{m=1}^{C} X_m V_{km} S_{km} \leq 1.0 \quad (3)$$

A simple extension of the analysis of the single class asymptotic bounds to multiclass queueing networks is presented in [6]:

$$X_c \leq \min[\frac{N_c}{Z_c + \sum_{k=1}^{K} V_{kc} S_{kc}} , \frac{1}{V_{bc} S_{bc}}]$$

where $b$ is the bottleneck device for class $c$ customers.

Note that the upper bounds in (3) are potentially tighter than this whenever $N_c/(Z_c + \sum_{k=1}^{K} V_{kc} S_{kc}) > 1/(V_{bc} S_{bc})$ for any class $c$.

## 2.2 Lower Bounds

The derivation of the lower bound on throughput presented in Theorem 1 is based on the following lemma which captures the contention between two customers arriving at a FIFO queue.

**Lemma 1**: Consider a closed system in which two customers, $i$ and $j$ in class $c$ and class $m$ respectively, arrive at a FIFO queue $k$. The probability that customer $i$ upon arrival finds customer $j$ enqueued or in service at $k$ is upper bounded by:

$$P_{cm} \leq \min(1, \frac{f_m V_{km}}{f_c V_{kc}})$$

*Proof*:
*Case 1*: $\frac{f_m V_{km}}{f_c V_{kc}} > 1$
Clearly, $P_{cm}$ is a probability and cannot exceed 1.
*Case 2*: $\frac{f_m V_{km}}{f_c V_{kc}} \leq 1$
The proof for this case is obtained by contradiction. For suppose that $P_{cm} > \frac{f_m V_{km}}{f_c V_{kc}}$. The rate at which customer $i$ (in class $c$) visits $k$ is given by $f_c V_{kc}$. Out of these $f_c V_{kc}$ visits made per second by customer $i$ let $q_{cm}$ denote the mean number of visits per second that found customer $j$ enqueued or in service at $k$. Since with a FIFO queueing discipline $i$ can be overtaken by $j$ only once per visit and the rate at which $j$ (in class $m$) arrives at $k$ is $f_m V_{km}$,

$$q_{cm} \leq f_m V_{km} \quad (4)$$

Now by definition,

$$P_{cm} = \frac{q_{cm}}{f_c V_{kc}} \quad (5)$$

If $P_{cm} > \frac{f_m V_{km}}{f_c V_{kc}}$ it follows that:

$$q_{cm} > f_m V_{km}$$

which gives rise to a contradiction.

From the analyses of *Case 1* and *Case 2* it follows that:

$$P_{cm} \leq \min(1, \frac{f_m V_{km}}{f_c V_{kc}}) \quad (6)$$

QED

**Theorem 1**: Consider a closed multiclass queueing network composed of $K$ devices and $C$ customer classes. If the customers arriving at a device are served in FIFO order, then, $X_c$, the class $c$ throughput for every $c \in (1 \ldots C)$ is lower bounded by:

$$X_c \geq \frac{N_c}{Z_c + \sum_{k=1}^{K} V_{kc} \sum_{m=1}^{C} N_m S_{km} min(1, \frac{f_m V_{km}}{f_c V_{kc}})}$$

*Proof*: Consider a tagged customer $i$ that belongs to class $c$. A visit to device $k$ by the tagged customer can be delayed by another customer $j$ ($j \neq i$) in class $m$, which is enqueued or is in service at $k$ when $i$ arrives. Because the service time distributions for the devices are IFR, the mean residual service time of a customer in service is bounded above by the mean service time [7]. Thus the delay introduced by $j$ per visit of $i$ at $k$ is bounded above by $S_{km}$.

The cycle time $R_c$ of the tagged customer is given by:

$$R_c = Z_c + \sum_{k=1}^{K} V_{kc} d_{kc} \quad (7)$$

where $d_{kc}$ is the mean delay (that includes queueing as well as service) experienced per visit at device $k$ by customer $i$. Note that from the independence assumptions (presented in section 2) for service times and number of visits to a device made by a customer, the mean delay introduced by a device per cycle of customer $i$ is given by the product of $V_{kc}$ and $d_{kc}$. Now,

$$\begin{aligned} d_{kc} \leq \; & S_{kc} + (N_c - 1) S_{kc} + \\ & \sum_{(m=1; m \neq c)}^{C} N_m S_{km} min(1, \frac{f_m V_{km}}{f_c V_{kc}}) \end{aligned} \quad (8)$$

The first term on the right-hand-side is the mean device demand per visit for the tagged customer itself. The second term is the sum of the upper bounds on the queueing delay introduced at device $k$ by the other $(N_c - 1)$ class $c$ customers and follows from the observation that $P_{cc} \leq 1$ (by Lemma 1). The third term on the right-hand-side is the sum of the upper bounds on the queueing delay introduced at device $k$ by the customers in the other classes $m$ ($m \neq c$); the upper bound on queueing delay at $k$ introduced by each class $m$ customer follows directly from Lemma 1. From equation (7) and inequality (8) we get:

$$R_c \leq Z_c + \sum_{k=1}^{K} V_{kc}[N_c S_{kc} +$$

$$\sum_{(m=1;m\neq c)}^{C} N_m S_{km}\ min(1, \frac{f_m V_{km}}{f_c V_{kc}})]$$

A little manipulation of the previous inequality yields:

$$R_c \leq Z_c + \sum_{k=1}^{K} V_{kc} \sum_{m=1}^{C} N_m S_{km} min(1, \frac{f_m V_{km}}{f_c V_{kc}}) \qquad (9)$$

Note that for computing an upper bound on $R_c$ we include the entire mean duration of every service interval that can possibly conflict with the service of customer $i$. Because the device service times and the number of visits made by customers at a device are independent random variables and the inclusion of a service interval at device $k$ of a competing customer $j$ $(j \neq i)$ in the upper bound for $R_c$ does not depend on its duration, the mean of these intervals included in the upper bound for $R_c$ is the mean $S_{km}$. Thus:

$$X_c = \frac{N_c}{R_i} \geq \frac{N_c}{Z_c + \sum_{k=1}^{K} V_{kc} \sum_{m=1}^{C} N_m S_{km} min(1, \frac{f_m V_{km}}{f_c V_{kc}})} \qquad (10)$$

QED

Note that for a system with only a single class of customers inequality (10) reduces to the well known asymptotic lower bound for a single class queueing network [15].

*Generalization to Processor Sharing*

The analysis presented in this section for FIFO devices can be extended to networks which include devices that use *processor sharing* (PS) for scheduling and *infinite servers* (delay centres). For processor sharing a device is shared equally by all the customers that are currently visiting the device. Clearly, the inequalities 2 and 3 corresponding to the upper bounds on throughput are independent of the scheduling discipline and hold for devices with processor sharing as well. The inequality corresponding to the lower bound on throughput changes however when the queueing network includes PS nodes in addition to FIFO nodes. For brevity the modified inequality corresponding to the lower bound is stated without proof; the proof is similar to those of Lemma 1 and Theorem 1. The devices are labelled in accordance with their scheduling discipline in the inequality. Without loss of generality the devices using FIFO are labelled $1 \ldots F$ whereas the devices using PS are labelled $(F+1) \ldots K$.

$$X_c \geq N_c / [Z_c + \sum_{k=1}^{F} V_{kc} \sum_{m=1}^{C} N_m S_{km} min(1, \frac{f_m V_{km}}{f_c V_{kc}})$$

$$+ \sum_{k=F+1}^{K} V_{kc} S_{kc} \sum_{m=1}^{C} N_m min(1, \frac{f_m V_{km} S_{km}}{f_c V_{kc} S_{kc}})] \qquad (11)$$

Since there is no queueing delay at an infinite server node the inequalities corresponding to the upper bound on throughput (introduced in section 2.1) hold for a network containing infinite server nodes. The contribution of an infinite server node $k$ to the upper bound in the cycle time (that yields the lower bound on throughput) of a class $c$ customer is simply $V_{kc} S_{kc}$.

The discussion on processor sharing and infinite servers was included to show the applicability of the bounding analysis presented in this paper to a broad range of systems. The remaining part of the paper focuses primarily on an interval arithmetic based technique used for the computation of the bounds. The computation technique does not change, however, with the incorporation of Processor Sharing nodes in the network. Thus for brevity the discussions in the remaining sections of the paper are presented in the context of queueing networks that use FIFO queueing centres.

## 3 Computation of the Bounds

The upper and lower bounds on $X_c$ given by the inequalities (3) and (10) are expressed in terms of the throughputs of the other customer classes in the network. The fact that the throughputs of the different customer classes are unknown and the existence of the *min* terms on the right-hand-side of inequality (10) make it difficult to extract the bounds in analytic form. A technique based on *interval arithmetic* is used to compute the numerical values for the upper and lower bounds for each customer class in the queueing network. A brief description of the interval arithmetic concept and how it is used to compute the bounds are presented.

### 3.1 Interval Arithmetic

The concept of interval arithmetic is centred around a data type called *interval*. An interval is an object that represents a real number lying between an upper and lower bound. The bounds may vary during the lifetime of a computation. The interval arithmetic capability, available as a special feature of the BNR Prolog [1] language developed at Bell Northern Research, has been used in a variety of different problem areas such as Critical Path (PERT) analysis [14], X-ray diffraction crystallography [13], and lends itself naturally to the computation of the bounds presented in the paper. The interval arithmetic provided by BNR Prolog allows the expression of arithmetic relationships among intervals and offers a mechanism for solving sets of linear and nonlinear equations and inequalities. Typically a set of equalities and inequalities may be specified as a set of Prolog subgoals. The BNR Prolog interpreter evaluates these subgoals and computes the intervals that satisfy the arithmetic relationship specified by each of these subgoals. As a computation proceeds forward intervals can change their values only by *narrowing*. An interval is narrowed if its lower bound is raised or its upper bound is lowered or both. Such narrowing of ranges for intervals may occur due to the evaluation of an arithmetic expression and the intervals are said to have been constrained by the subgoal corresponding to the arithmetic expression. BNR Prolog uses backtracking to undo the narrowing of an interval which is analogous to the unbinding of ordinary variables.

For the queueing network bounds, the interval arithmetic can be used to calculate upper and lower bounds on each class throughput separately, giving a rectangular box circumscribed around the feasible throughput region. The inequalities derived in Section 2 are converted to Prolog subgoals. Inequality (2) corresponding to the no contention upper bound gives rise to $C$ subgoals, one for each class $c \in (1 \ldots C)$. Inequality (3) corresponding to the utilization-based upper bound gives rise to $K$ subgoals, one for each device $k \in (1 \ldots K)$. Similarly we get $C$ additional subgoals, one for each customer class, for the lower bound on throughput as captured by inequality (10). When the program is executed each subgoal is evaluated and a set of upper and lower bounds $(X_c^+,\ X_c^-)$ satisfying the inequalities is obtained for each customer class $c$:

$$X_c^+ \geq X_c \geq X_c^- \tag{12}$$

These set of upper and lower bounds form the corner points of a rectangular polytope in a C-dimensional throughput space $(X_1, X_2, \ldots X_c)$ and is termed *Robust Box Bounds (RBB)*. The feasible set of throughputs for the model must lie inside the rectangular polytope. A Prolog program for computing the RBBs for an example queueing network model, *QNM1*, with two devices and two customer classes is given in Figure 2(a). The parameters for the model are:
*QNM1*

$$N_1 = 2,\ V_{11} = 1,\ S_{11} = 0.9,\ V_{21} = 1,\ S_{21} = 0.1,\ Z_1 = 0$$

$$N_2 = 3,\ V_{12} = 1,\ S_{12} = 0.9,\ V_{22} = 10,\ S_{22} = 0.1,\ Z_2 = 0$$

The cycle rates of customers $(f_c)$ and the throughputs for the different customer classes $(X_c)$ are represented by intervals whereas all the other variables in the program are integers or floating point numbers.

The Robust Box Bounds for *QNM1* computed by the program are given by:

$$X_1^- = 0.3999, X_1^+ = 0.74075, X_2^- = 0.37037, X_2^+ = 0.71112.$$

When plotted in the $X_1 - X_2$ plane the bounds form the corner points of a rectangle (see Figure 2(b)). The bounds are robust in the sense that they rely on minimal assumptions about the distributions underlying the service demands of the customers and are called Box Bounds because of the shape of the region they enclose in the C-dimensional throughput space (see Figure 2(b) for example).

## 3.2 Numerical Examples

The Robust Box Bounds can be used for the performance analysis of computer communication networks as well as other types of computing systems. Two small examples will be described in this section to compare the RBBs to other Bounds. Then four larger networks drawn from data communications examples will be used to explore the practicality of the RBB approach for larger systems.

### Comparison of RBB with Other Bounding Techniques for Separable Queueing Networks

In this discussion the RBBs computed by using the interval arithmetic technique will be compared to the Composite Bounds of Kerola [6], which are applicable only to separable multiclass queueing networks. Methods for separable networks such as [6] use only the total device loading parameters ($L_{kc} = V_{kc}S_{kc}$), while RBBs depend on $V_{kc}$ and $S_{kc}$ separately. The model *QNM1* was made equivalent to the model used in [6] by choosing $V_{kc}$ and $S_{kc}$ to give the values $L_{kc}$ quoted by Kerola. Note that provided the service means $(S_{kc})$ for different classes are the same at each device, the inequalities (2), (3) and (10) show that the RBBs also depend only on $L_{kc}$, and not on $V_{kc}$ and $S_{kc}$ separately. This is also true for non separable networks (e.g., with general service times) as well.

Figure 3 shows the bounds on $X_1$ as a function of the number of class 1 customers, $N_1$, with $N_2$ fixed at 3. For the Composite Bound method, $CUB^+$, the upper bound on the throughput of a given customer class, is computed from the lower bound on throughput given by the Balanced Job Bounds $(BJB^-)$. The RBBs are found to be loose in comparison to the Composite Bounds especially when the system is lightly loaded. This is expected since both the $CUB^+$ and $BJB^-$ bounds are applicable to separable queueing networks only whereas the RBBs shown as $X_1^+$ and $X_1^-$ are more general and are applicable to separable as well as non-separable queueing networks. As in the case of Composite Bounds the RBBs for $X_1$ become tighter with an increase in the number of customers $N_1$.

The Robust Box Bounds for a different set of parameter values for the same queues and classes that result in a non-separable network, *QNM2*, are presented as a function of $N_1$ in Figure 4. The parameters for

*QNM2* are as follow:

$$N_1 = 2,\ V_{11} = 1,\ S_{11} = 0.9,\ V_{21} = 1,\ S_{21} = 0.1,\ Z_1 = 0$$

$$N_2 = 3,\ V_{12} = 10,\ S_{12} = 0.09,\ V_{22} = 10,\ S_{22} = 0.1,\ Z_2 = 0.$$

Note that *QNM2* is not separable because $S_{11} \neq S_{12}$. The only difference between *QNM1* and *QNM2* is the way in which the total demand for the first device ($k = 1$) exhibited by a class 2 customer is split among the different visits (i.e. in the $V_{12}$ and $S_{12}$ parameters). The bounds presented in Figure 4 seem to exhibit the same trend as the bounds presented in Figure 3. The bounds for *QNM2* are observed to be tighter, however, in comparison to the bounds for *QNM1*. To the best of our knowledge, there is no other available method that can compute both the upper and lower bounds on throughput for this system.

**Scalability of Bound Computations for Larger Systems**

Given that the RBBs are easily computed and give understandable results, are they practical for larger systems? This discussion considers four larger models, representing star communications networks with varying numbers of nodes, and investigates the execution time of the Prolog program.

The details of the queueing model are not essential factors in determining the execution time of the computation; but brief descriptions of the models are provided to explain the scale of the examples when they are viewed as communications networks. Each model is characterized by $K$ queueing centers, each of which represents a node in the communication network. The hub node that is connected to all the nodes and performs the necessary message switching is represented by the queueing centre with $k = 1$, whereas the other queueing centres correspond to the other $K - 1$ nodes in the system. $C$ of these nodes (labelled $2 \ldots C + 1$) are special in the sense that they represent sources and sinks of messages. Each such node generates messages of a particular class, modelled by a customer class (class $c$ for node $c+1$), which is characterized by a particular message length distribution (and thus also, a particular set of service time distributions). As described in [16], window flow control restricts the number of unacknowledged messages in a particular class to $N_c$, the number of customers in class $c$ and gives a closed queueing network. After a message is generated it is serviced at the local queueing centre $c + 1$, so $V_{c+1} = 1$; the message is then forwarded to the queueing centre number one that represents the hub node. Transmission delay is neglected. (Depending on the type of communication between nodes, modelling of transmission delay could be easily achieved through the incorporation of an infinite server or a queueing center in the network.) Each of the remaining $(K - 2)$ nodes (excluding nodes 1 and $c$) is selected with equal probability to be the destination node for the message: $V_{kc} = 1/(K - 2)$. After a message reaches its destination it is processed by the destination node and an acknowledgement message is sent back to the originating node. Since the hub node is visited twice, once during the forwarding of the message and once during the switching of the acknowledgement back to the originator of the message, $V_{1c} = 2$ for $c = 1 \ldots C$. No processing is required for the acknowledgement message at the originating node. However, $Z_c$ units of time are spent on an average by the node for absorbing the acknowledgement and generating a new message. This is modelled as the think time for a customer in class $c$. The different $S_{kc}$ values represent the processing times for messages at the different nodes.

Numerical results for the four different queueing networks, are presented in Table 1. Although the queueing networks are non-separable, the assumptions outlined in Section 2 for the validity of the bounds are assumed to be satisfied. $K$ is held at 5 and $C$ is held at 3 for the queueing networks *QNM3*, *QNM4*, and *QNM5*. The particular values of service demands ($S_{kc}$) for each class are indicated in the last column of the table. The only difference between *QNM3* and *QNM4* is in the $Z_c$ parameters. As expected with an increase in $Z_c$ the ratio between the upper and lower bounds tends to decrease and tighter bounds are obtained for *QNM3* in comparison to those for *QNM4*. As in the case of *QNM4*, $Z_c$ is held at 0 ($c = 1 \ldots C$) in *QNM5*. *QNM5* is characterized by a bottleneck at one of its queueing centres ($k = 1$) and tight bounds are obtained (see Table 1).

The interval arithmetic based technique has been used for the computation of larger networks as well. The results for queueing network *QNM6* for which $K = 31$ and $C = 3$ are presented in Table 1. The approximate time to run the BNR Prolog program on an Apple Macintosh IIfx system for the computation of the bounds for each queueing network model is also included in the table. *QNM6* is one of the largest networks that we could experiment with before we ran into memory problems. The size of the program used for the computation of the bounds increases both with the number of devices as well as the number of classes in the model (see section 3.1). Other large networks, in addition to *QNM6* that we experimented with include one with $K = 8$ and $C = 7$. It is encouraging to note that in all of these cases the run time of the program was less than or equal to 12 seconds. The memory problem is due to the lack of virtual memory support from the system running BNR Prolog. This problem is expected to be alleviated in the near future when we can continue the investigation with larger networks.

Intrinsically these computations appear to scale well, although our experience is limited. Other, related BNR Prolog problems have scaled nearly linearly, up to hundreds of inequalities [12]. Further experience is being sought to understand this scalability issue.

*Networks with "Classes of Nodes"*
Larger networks, with hundreds or even thousands of nodes, can be analyzed more efficiently if symmetry exists in the computer network, or can be assumed. If sets of nodes exist with the same traffic parameters, each set or class of nodes needs to be represented only once in the bound calculation. For example, only one inequality of the form (3) is required for each node (device) class for representing the utilisation-based upper bound. The complexities of the inequalities corresponding to the no contention upper bound and the lower bound on throughput are also greatly reduced. The set of nodes $\{C+2, \ldots K\}$ in the four larger examples above could be represented in this way by a single representative node. In planning studies for large networks this kind of symmetry is frequently assumed in order to make the network more understandable, so this is a practical possibility in many cases.

# 4 Feasible Throughput Region

The RBBs are generally somewhat looser than the inequalities 2, 3, and 10 because they ignore dependencies between the classes. For example, in inequality 3 the upper bound on $X_1$ depends explicitly on the value of $X_2$. In a two-class system the bounds found from 3 would produce oblique straight lines if plotted in the $X_1 - X_2$ throughput space, while the RBB upper bounds provide a simple maximum value for each axis, giving rise to two straight lines parallel to the axes. The same thing is true for the RBB lower bounds. Figure 5 illustrates how the inequalities 2, 3, 10 define a convex region bounded by straight lines, while the RBBs give the (unique) circumscribed rectangle parallel to the axes. This generalizes to $C$ classes. We will call the inner region bounded by inequalities 2, 3, 10 the *feasible throughput region* to differentiate it from the larger region delimited by the RBBs.

For the sake of explaining the feasible throughput region concept a simple queueing network with a single queueing centre is used. The concept is also applicable to larger queueing networks which for example may be used to model computer communication networks. Figure 5 is drawn for a two-class queueing network *QNM7*. The network is characterized by a single device and a single customer in each class (ie., a two-class "repairman" model) with parameters:

$$N_1 = 1, V_{11} = 5, S_{11} = 4, Z_1 = 1,$$
$$N_2 = 1, V_{12} = 1, S_{12} = 4, Z_2 = 10$$

The no contention upper bound on throughput gives rise to the following inequalities:

$$X_1 \leq \frac{1}{Z_1 + V_{11}\ S_{11}} \tag{13}$$

$$X_2 \leq \frac{1}{Z_2 + V_{12}\ S_{12}} \tag{14}$$

By using the utilization-based upper bound on throughput (inequality 3) and noting that $X_c = f_c$ yield:

$$(X_1 V_{11} S_{11} + X_2 V_{12} S_{12}) \leq 1.0 \tag{15}$$

The lower bound on the throughputs are obtained by using inequality 10:

$$X_1 \geq \frac{1}{Z_1 + V_{11} S_{11} + V_{11} S_{12}\ min(1, \frac{X_2 V_{12}}{X_1 V_{11}})} \tag{16}$$

$$X_2 \geq \frac{1}{Z_2 + V_{12} S_{12} + V_{12} S_{11}\ min(1, \frac{X_1 V_{11}}{X_2 V_{12}})} \tag{17}$$

The inequalities corresponding to the bounds are displayed graphically in Figure 5. The positive quadrant bounded by the $X_1$-axis and the $X_2$-axis is divided: one for region $X_2 V_{12} > X_1 V_{11}$, and the other for $X_2 V_{12} \leq X_1 V_{11}$. The other lines are numbered by the equations (13), (14), and (15), with two segments each, marked A and B, for (16) and (17). The dark area in Figure 5 which is enclosed by the straight lines is the the feasible throughput region for the system. The area enclosed by the feasible throughput region is smaller than the area of the rectangle corresponding to the RBBs (see Figure 5). It may be noted that because of its convexity the center of the rectangular polytope corresponding to the RBBs is guaranteed to lie within the feasible throughput region.

## 4.1 Computation of the Feasible Throughput Region

The analytic technique for obtaining the exact feasible region, described above, is intractable for more than two classes and is quite complex for larger numbers of nodes. In larger systems, a graphical visualization of two classes (for fixed values of the other throughputs) can be computed very simply by considering a single ray projected from the origin of the throughput space. Suppose we wish to see the feasible region in $(X_1, X_2)$. Along such a ray with slope $\gamma$ we have $X_2 = \gamma X_1$ and the other throughputs are fixed. The intersection of the ray with the exact feasible space is defined by the values $[X_1^{-(\gamma)}, X_1^{+(\gamma)}]$ for that ray, and once they are known the intersection can

Table 1: RBBs for Larger Queueing Network Models with $C = 3$, $N_c = 5$ for all $c$.

| Model | Class ($c$) | Lower Bound ($X_c^-$) | Upper Bound ($X_c^+$) | Approximate Run Time | Parameters |
|---|---|---|---|---|---|
| *QNM3* | 1<br>2<br>3 | 0.076213<br>0.053896<br>0.041730 | 0.22702<br>0.11351<br>0.07567 | less than 1 second | $K = 5$,<br>for $\forall c, k > 1$ :<br>$S_{kc} = c$<br>for $\forall c$:<br>$S_{1c} = c/80$<br>$Z_c = 20c$ |
| *QNM4* | 1<br>2<br>3 | 0.082304<br>0.082304<br>0.082304 | 0.86283<br>0.44514<br>0.30590 | less than 1 second | same as *QNM3* except:<br>$Z_c = 0$ $(c = 1 \dots 3)$ |
| *QNM5* | 1<br>2<br>3 | 0.014393<br>0.014286<br>0.014181 | 0.021532<br>0.021425<br>0.021320 | 1 second | same as *QNM3* except:<br>$Z_c = 0$ $(c = 1 \dots 3)$<br>$S_{1c} = 10$ $(c = 1 \dots 3)$ |
| *QNM6* | 1<br>2<br>3 | 0.13034<br>0.11436<br>0.10187 | 0.98158<br>0.49249<br>0.32921 | 12 seconds | $K = 31$, $C = 3$<br>for $\forall c$:<br>$N_c = 5$<br>$S_{kc} = c4$, $k = 2 \dots 31$<br>$S_{1c} = c/80$<br>$Z_c = 0$ |

be plotted as a line segment as shown (by a dark line) in Figure 6. Note that this figure is used to explain the methodology and it does not correspond to any specific queueing network.

The values $[X_1^{-(\gamma)}, X_1^{+(\gamma)}]$ are easily found by the RBB calculation, by adding the constraints $X_2 = \gamma X_1$, and the fixed values for $X_3, X_4, \dots$. Figure 6 displays only three line segments. By increasing $\gamma$ in small increments and by plotting the line segments all together the feasible region can be delineated.

## 5 Summary and Conclusions

Most of the existing research on the exact analysis and bounds for queueing network models are applicable only to separable queueing networks. The assumptions that underly separable queueing network models are usually violated by real computer communication networks as well as various other classes of real computing systems, particularly the assumptions on FIFO queues. A set of throughput bounds called the Robust Box Bounds that are applicable to general closed multiclass queueing networks and can be used for the performance analysis of such systems is presented in this paper. The upper bound on class throughputs is an extension of the well known asymptotic upper bound for single class networks. A new lower bound on the throughput of a customer class that is applicable to general closed multiclass networks with IFR service is introduced in this paper.

The analytic expressions for both the upper and lower bounds for a given customer class, $X_c$, are expressed in terms of inequalities that involve the throughputs of other customer classes in the network. Representation of the bounds for many classes, for proper visualization by an analyst is a problem. For two classes of customers, the bounds can be plotted. For two or more classes, a technique based on interval arithmetic can be used to extract the numerical values for upper and lower bounds on the throughput of each customer class, which are called the "Robust Box Bounds". The important features of this new bounding technique are briefly summarized:

- The bounds are robust in the sense that they are independent of the distributions that underly the service demands of customers at the different devices and are insensitive to routing dependencies.

- Approximate analysis of systems that do not satisfy the assumptions for separable queueing networks is possible but the error of the result is unknown. Moreover, sometimes the distributions or even the variance of service times are unknown which may make the approximation techniques unusable. Robust Box Bounds that require only first moments are useful in these situations: system performance is guaranteed to lie within the upper and lower bounds that are computed by this bounding technique.

- Using inequalities corresponding to both the upper and lower bounds on throughput results in an upper bound that is potentially tighter (see section 2.1) than the generalized version of the asymptotic upper bounds presented in [6]. The RBB also provides a performance guarantee that the throughput of each class is at least as great as the lower bound. Designs to provide performance guarantees usually lead to conservative loading of systems, which suggests that they might be useful in an overall conservative design method for contention systems.

This paper has concentrated on the development of the bounding technique that is a viable way of performance estimation especially in the absence of information about the distribution of service times or routing dependencies. Investigation of the tightness of the bounds in the context of various architectures for computer networks forms an important direction for future research. Extension of the bounding technique to include systems that employ priority based scheduling is currently under investigation. Research is also planned for applying the bounding technique to models of software [10]. Interval arithmetic can also be performed by programming languages other than BNR Prolog. Investigating the effectiveness of different languages for performing bound computations is important.

## Acknowledgements

This research was supported by the Natural Sciences and Engineering Research Council of Canada and by the Telecommunication Research Institute of Ontario through its project on Telecom Software Methods. We would like to thank Bell-Northern Research Ltd. for donating BNR Prolog to the university and Alex Hubbard for his help in testing the bounding technique for large queueing network models. Thanks are also due to Elena Keen for her help in typing the manuscript and in preparing the figures.

## References

[1] *BNR Prolog User Guide* Bell-Northern Research Ltd., Ottawa, Canada, 1988.

[2] B.W. Stuck and E. Arthurs, *A Computer and Communications Network Performance Analysis Primer.* Englewood Cliffs, NJ, USA:Prentice-Hall, Inc., 1985.

[3] D.L. Eager and K.C. Sevcik, "Bound Hierarchies for Multiple-Class Queueing Networks," *Journal of the ACM*, Vol. 33, No. 2, 1986.

[4] D.L. Eager and K.C. Sevcik, "Performance Bound Hierarchies for Queueing Networks," *ACM Trans. on Computer Systems*, Vol. 1, No. 2, pp. 99–115, May 1983.

[5] D.R. Cheriton and C.L. Williamson, "Network Measurement of the VMTP Request-Response Protocol in the V Distributed System," In *Proceedings of the ACM SIGMETRICS Conference on Measurement and Modeling of Computer Systems*, pp. 216–225, May 1987.

[6] T. Kerola, "The Composite Bound method for Computing Throughput Bounds in Multiple Class Environments," *Performance Evaluation*, Vol. 6, No. 1, pp. 1–9, January 1986.

[7] L. Kleinrock, *Queueing Systems.* Volume 2, John Wiley and Sons, 1976.

[8] J. Kriz, "Throughput Bounds for Closed Queueing Networks," *Performance Evaluation*, Vol. 4, No. 1, pp. 1–10, January 1984.

[9] L.E. Stephens and L.W. Dowdy, "Convolutional Bound Hierarchies," In *Proc. of the 1984 ACM SIGMETRICS Conf., Cambridge, U.S.A.*, pp. 120–133, 1984.

[10] S. Majumdar, C.M. Woodside, J.E. Neilson, and D.C. Petriu, "Performance Bounds for Concurrent Software with Rendezvous," *Performance Evaluation*, Vol. 13, No. 4.

[11] D. Michandani and P. Biswas, *Characterizing and Modeling Ethernet Performance of Distributed DECwindows Applications*,, Nashua, NH, U.S.A., September 1989.

[12] W. Older.

[13] W. Older *The Application of Relational Arithmetic to X-Ray Diffraction Crystallography* Ottawa, Canada, February 1988.

[14] W. Older, A. Vellino, and B. Farrahi *Relational Arithmetic in Prolog Using Interval Constraints* Bell-Northern Research, Ottawa, Canada.

[15] R.R. Muntz and J.W. Wong, "Aymptotic Properties of Closed Queueing Network Models," In *Proc. Eight Conf. on Information Sciences and Systems*, pp. 348–352, 1974.

[16] M. Reiser, "A Queueing Network Analysis of Computer Communication Networks with Window Flow Control," *IEEE Trans. on Communications*, Vol. COM-27, No. 8, pp. 1199–1209, August 1979.

[17] R. Suri, "Generalized Quick Bounds for Performance of Queueing Networks," *Computer Performance*, Vol. 5, No. 2, pp. 116–120, 1984.

[18] V-T. Hsieh and S.S. Lam, "Two Classes of Performance Bounds for Closed Queueing Networks," *Performance Evaluation*, Vol. 7, No. 3, pp. 3–30, March 1987.

[19] J. Zahorjan, K.C. Sevcik, D.L. Eager, and B. Galler, "Balanced Job Bound Analysis of Queueing Networks," *Communications of the ACM*, Vol. 25, No. 2, pp. 134–141, January 1982.

Figure 1: Examples of Interval Arithmetic

Figure 2: Robust Box Bounds for *QNM1* (a) The Prolog Program (b) The Bounds

Figure 3: Comparison of the Robust Box Bounds with Composite Bounds for *QNM1*

Figure 4: The Robust Box Bounds for *QNM2*

Figure 5: The Feasible Throughput Region

Figure 6: Computation of the Feasible Throughput Region