Throughput analysis of a flow-controlled communication network with buffer space limitations

J.L. van den Berg, O.J. Boxma
Centre for Mathematics and Computer Science
P.O. Box 4079, 1009 AB Amsterdam, The Netherlands

This paper studies the traffic flow in a virtual circuit of a computer communication network with window flow control. Due to finite buffer capacity, overflow of data packets is possible. Lost packets have to be retransmitted. To maintain the original order of the packets, subsequently sent packets also have to be retransmitted, causing a deterioriation of throughput.
An approximation method, based on a queueing network model, is developed to analyse the throughput behaviour. This method leads to exact results for the single-hop network. For the multi-hop circuit, it yields very accurate approximations, as is illustrated by simulation.

KEY WORDS & PHRASES: Computer communication network; flow control; virtual circuit; overflow; (negative) acknowledgment; closed queueing network; throughput.

1. Introduction

A computer communication network consists of a number of facilities (processors, links between processors, etc.) shared by competing users (messages or packets). Senders generate messages, which are subsequently transmitted over the network to receivers. The acceptance of a message by a receiver usually consists of (i) storing the message in a buffer, and (ii) sending an acknowledgment (ACK) back to the sender. As soon as the sender receives this ACK, he knows that there is no longer any need to store a copy of that message; a buffer space can be emptied. However, when the sender has not received an ACK within a certain period of time after transmission (the time-out), or when he receives a negative acknowledgment (NACK), he will retransmit that message in order to prevent endlessly long waiting times.

The finite capacities of the network facilities cause conflicts between the messages. Queues of messages are formed in front of certain facilities, and finite buffers are filled to completion so that messages are lost and have to be retransmitted. This may lead to a drastic reduction of the performance of the system, what is reflected in ever longer waiting times and a decrease of the effective throughput of the network.

If too many messages lay a claim on the available resources in the network, flow control procedures are needed to prevent the system from becoming overloaded. Generally a distinction is made between *local* and *global* flow control. Local flow control is the collection of procedures that regulate the traffic *within* the network, for example between two nodes; global flow control is the collection of procedures that regulate the externally offered traffic (Gerla and Kleinrock [8]). The principle of global flow control is to shift congestion from the interior of the network to the points of traffic admittance (Reiser [12]).

In the following our attention will be devoted to a form of global flow control, viz. end-to-end flow control, that is being exercised on so-called virtual circuits of the network (a virtual circuit is a fixed route, along which messages are transmitted between a particular sender and a particular receiver, see Fig. 1).

NATO ASI Series, Vol. F38
Flow Control of Congested Networks
Edited by A. R. Odoni et al.

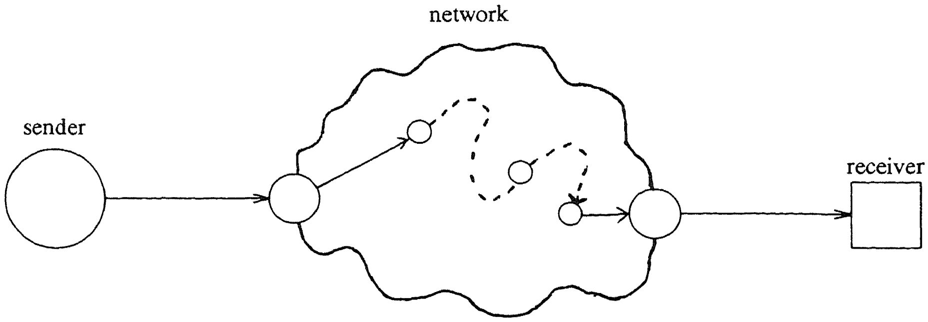

Fig. 1 A virtual circuit.

The most common end-to-end protocol is the *window flow control protocol* (see Cerf and Kahn [5] and Reiser [12,13]). The principle behind this protocol is that an upper limit is imposed on the number of not yet acknowledged messages that can simultaneously be present in the virtual circuit. This maximum is called the *window size*.

Several variants of the window flow control protocol are in existence. One of them, the *sliding window protocol*, plays an important role in this study, and will therefore now be discussed in more detail (see Reiser [12,13]).

The sliding window mechanism operates as follows (see Fig. 2).

1. For each message (or packet) that is being transmitted over the virtual circuit, a counter (which is initially set to N, the window size) is decreased by one.
2. As soon as the counter reaches the value zero, no more new messages are admitted to the virtual circuit (the sender stops transmission).
3. Each message is individually acknowledged by the receiver upon reception. As soon as the sender receives an ACK, the counter is increased by one. The sender is reactivated when the counter goes up from zero to one.

It is often assumed that ACK's are received after a negligibly short network delay (see Fig. 2).

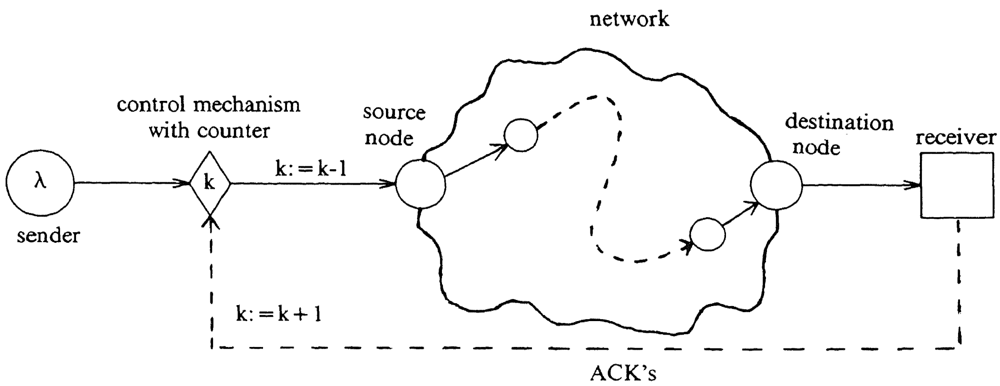

Fig. 2 Virtual circuit operating a sliding window.

A virtual circuit can easily be represented by a *queueing network*. In that case the sender is a source which, with a certain intensity, generates customers (messages); these customers travel along the successive links of the virtual circuit. Each link can be modelled as a service system with one server, with service times that equal the transmission times of messages over this link. Thus the service time of a message at a service system is determined by the length of that message.
For a further analysis it is important to remark that the sliding window mechanism keeps the sum of the total number of customers and of the ACK's present on the virtual circuit, and of the counter, constant and equal to N. Consequently, a virtual circuit with a sliding window mechanism can be modelled as a closed cyclic queueing network (see Fig. 3, and cf. Reiser [12]). It is thereby of course assumed that each node in the original system has a buffer capacity of at least N, so that retransmission is never necessary. The queue length at the service system with service intensity λ now represents the counter in the original system. Of course, the server with service intensity $1/\alpha_i$ represents the link between the i-th and (i+1)-th node of the virtual circuit (with average transmission time α_i), $i = 1,...,M$.

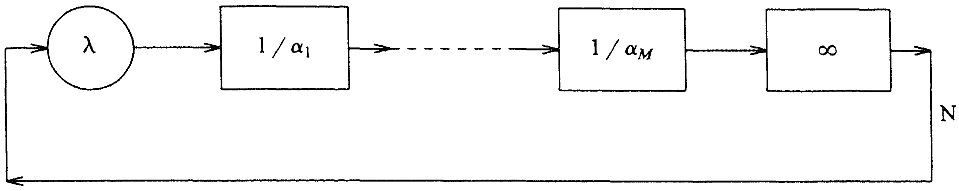

Fig. 3 Queueing model of the virtual circuit pictured in Fig. 2.
The queue with infinite service intensity represents the receiver.

A complete network, with sliding window mechanisms on the virtual circuits, can hence be modelled as a queueing network consisting of several closed chains. If we suppose that the transmission times of one and the same message at successive links are independent (Kleinrock's "Independence Assumption", see Kleinrock [10]) and negative exponentially distributed, then an exact analysis of the joint queue-length distribution in the model is possible (Baskett et al. [1]). However, the computational complexity rapidly grows with the number of closed chains so that, actually, exact solutions are only possible in rather simple cases (see Reiser and Kobayashi [14]). In [12], Reiser presents an efficient heuristic method to handle this problem.

As remarked above, the buffer capacities of the nodes - in particular in a local area network - cannot always be assumed to be infinite. Therefore it is possible that a message, upon arrival at a node, cannot be accepted because the buffer space is full. In that case this message, K say, has to be retransmitted by the sender. There are various procedures to let the sender know that K has been lost. First of all, a time-out mechanism may be used: a clock starts running at the time a message transmission starts, and when no ACK has been received for this message within a certain period of time, the time-out, the sender assumes that the message has been lost. Another procedure is the following. When the receiver receives a message with a higher sequence number than K without having received K, he sends a negative acknowledgment (NACK) - which contains K's sequence number - back to the sender. Usually the order of the messages has to be maintained, in order to prevent the need for sorting messages at the receiver. Consequently, not only K has to be retransmitted, but also those messages that were subsequently sent.

These blocking and retransmission phenomena generally make an exact mathematical analysis prohibitive. There is a strong need for mathematical models that reasonably accurately reflect the behaviour of real networks with overflow and flow control, and that are still amenable to an exact or approximate analysis. The goal of the present paper is to develop and analyse such a model. It is a queueing model of a virtual circuit with a sliding window mechanism, in which only the destination node has a finite buffer capacity. The model is an extension of the cyclic queueing model of Reiser [12], which was described above. An approximation method for the analysis of this queueing model is developed. Attention is restricted to the throughput of the circuit, and in particular to the maximum attainable throughput (in the case of an infinitely fast source). Throughput is a very important performance measure in these flow-controlled networks. Another useful performance measure, the end-to-end delay of messages over the network, will be the subject of a future study.

The rest of the paper is organized in the following way. Below we describe the queueing network model which represents a virtual circuit with limited buffer capacity at the destination node. Section 2 is devoted to an approximate analysis of the model. This analysis is shown to be exact in the case of a single-hop circuit (i.e., no intermediate nodes between source node and destination node). In Section 3 we compare the approximation results with results obtained by simulation. Several tables and graphs are presented, which illustrate the accuracy of our approximation and which expose the influence of retransmissions on the (maximum) throughput in a virtual circuit with overflow and flow control. In Section 4 we summarize our findings and we mention some points which are considered for future research.

Queueing model description

We have already seen that a virtual circuit which operates a sliding window protocol can be modeled as a closed queueing network. We now add the assumption that the last node of the circuit has finite buffer space of size L ($<N$). In Fig. 4 the corresponding queueing network has been drawn.

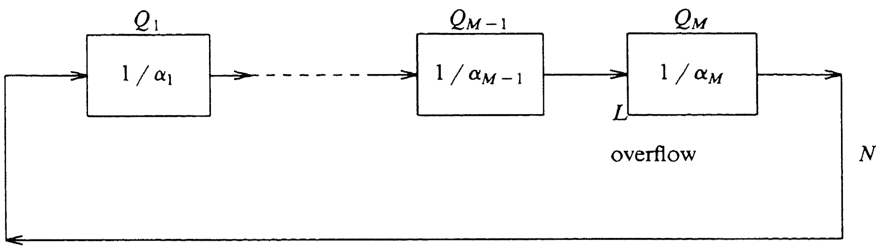

Fig. 4 Queueing model of a virtual circuit with buffer overflow.

The service systems $Q_1, \ldots, Q_M$ represent the M nodes of the virtual circuit with their outgoing links. Q_i is a single-server system, $i = 1,2,...,M$. Service times at Q_i are independent, negative exponentially distributed stochastic variables with mean α_i, $i = 1,2,...,M$; service times at different queues are also independent. The service times represent transmission times of messages between the nodes of the virtual circuit plus, possibly, an additional nodal delay. The receiver is not modelled because we assume that as soon as a message has obtained its service at Q_M, it has reached the receiver; the receiver immediately sends an acknowledgment back to the sender, and this ACK has infinitely short transmission time and delay. In reality ACK's are either piggybacked on other messages which are sent along the reverse route, or are stand-alone. Stand-alone ACK's are often assigned a higher priority. In that case the assumption of infinitely short transmission time and delay for ACK's (which are much shorter than ordinary dataflow messages) is not unrealistic; see also Remark 2.4.

We have not modelled the sender, because we assume that its message generation intensity is infinite: as soon as the sender has received an ACK, a new message is admitted to Q_1. In this way we can study the maximum attainable throughput of the virtual circuit (to study throughput per se we would have to include another queue, with service rate equal to the intensity of the sender). We have thus arrived at a closed cyclic queueing model. The number of customers in the system equals N, the window size.

$Q_1, \ldots, Q_{M-1}$ have "infinite" waiting rooms (waiting rooms of size at least $N-1$). Q_M has a finite waiting room of size $L-1$: at most L customers can be in service or waiting at Q_M. We assume that $N>L$. When a customer, say K, leaves Q_{M-1} and finds L customers in Q_M upon arrival, then overflow occurs. Earlier, several procedures were mentioned to let the sender know that a message has been lost (time-out, NACK). In the present study we shall model the NACK mechanism but not the time-out mechanism. Our model could be adapted to incorporate the time-out mechanism (cf. Remark 2.3), but the minor complications arising from this inclusion would obscure the essence of the approximation method.

Suppose that message K is the first one that is lost. The receiver only notices that K has been lost, when he receives a message with a higher sequence number than K. The receiver now sends a NACK for K back to the sender. As in the case of ACK's, it is assumed that NACK's experience no transmission time and delay. The implications for the queueing model are the following. Suppose that customer K is the first one to cause overflow, and suppose that customer $\hat{K}$ is the first one, sent after K, to be admitted to Q_M. As soon as the service of $\hat{K}$ in Q_M is completed, K and all customers who have left Q_1 after K instantaneously return to Q_1 (to fit the overflow phenomenon in the context of our closed queueing network, we might introduce a "wait" queue WQ; overflown customers immediately enter WQ, and stay there until a NACK has reached the sender). Consequently, immediately after the service completion of $\hat{K}$ in Q_M, all customers are in Q_1. Because the original order of the customers has to be maintained, K is the first one to be taken into service in Q_1, preëmpting another service if necessary. Apart from this, the service discipline in all queues is first-come first-served.

Remark 1.1
In the present study it is assumed that, upon the arrival of a NACK at the source node, old messages are instantaneously removed from the nodes of the virtual circuit. An interesting question is whether a protocol which effectively accomplishes this can be implemented efficiently. Such a protocol should work as follows. Suppose A and B are communicating over a virtual circuit. At a certain epoch B sends a NACK to A (because B received a message with a wrong sequence number). The source and destination addresses of this NACK control message (B and A, respectively) are used to recognize messages, sent from A to B, in the buffers of the intermediate nodes of the virtual circuit. At each node visited by the NACK, messages with destination address B and source address A can be removed because copies of these messages will be retransmitted upon arrival of the NACK at the source node.

2. Analysis of the queueing model

In this section we study the maximum attainable throughput, T, of the queueing model described above, with

$$T := E[\textit{number of customers leaving } Q_M \textit{ per unit of time who do not have to be retransmitted}], \quad (2.1)$$

or alternatively,

$$T := E[\textit{number of customers entering } Q_M \textit{ per unit of time who do not have to be retransmitted}]. \quad (2.2)$$

Let t_j denote the j-th epoch, after time 0, at which a customer completes service in Q_M, having been the first admitted customer in this queue after the occurrence of overflow; t_j corresponds to the epoch of the j-th transmission of a NACK. A basic observation is that the joint queue length process at

$Q_1, \ldots, Q_M$ is a *regenerative process* (cf. Cohen [7]), with regeneration epochs $t_1, t_2, \ldots$. So the queue length process, and in particular also the departure process from Q_{M-1}, repeats itself probabilistically speaking after each t_j epoch. Moreover, the lengths $r_j := t_j - t_{j-1}$ of the successive regeneration intervals are independent. Let r denote a stochastic variable with the same distribution as $r_1, r_2, \ldots$. Because of the regenerative character of the queue length and departure processes,

$$T = \frac{E[\textit{number of customers entering } Q_M \textit{ in } [0,r] \textit{ who do not have to be retransmitted}]}{E[r]}. \tag{2.3}$$

So we can restrict ourselves to one arbitrary regeneration interval $[0,r]$. It appears to be natural to divide this interval into three consecutive subintervals I_1, B, I_2. Here
I_1 is the period from 0 until the first arrival of a customer at Q_M (note that all N customers are at Q_1 at time 0);
B is the period from the end of I_1 until the first occurrence of overflow at Q_M;
I_2 is the period from the end of B until the end of the regeneration interval.
Denote the lengths of these periods by i_1, b, i_2, respectively. Then

$$r = i_1 + b + i_2.$$

During the I_1-period no contribution to the throughput is made. During the I_2-period at least one customer is admitted to Q_M, but no contribution to the throughput is made because all those admitted during this period must be retransmitted. Hence only during the B-period a contribution to the throughput can be made. Let α denote the mean interdeparture time from Q_{M-1} during a B-period. Then, cf. (2.3),

$$T = \frac{1}{\alpha}\frac{E[b]}{E[r]} = \frac{1}{\alpha}\frac{E[b]}{E[i_1] + E[b] + E[i_2]}. \tag{2.4}$$

Trivially,

$$E[i_1] = \sum_{j=1}^{M-1} \alpha_j. \tag{2.5}$$

Below we introduce one single approximation assumption which immediately yields a simple approximation for $E[b]$ and $E[i_2]$ and hence for T.

Approximation assumption
During B- and I_2-periods, the arrival process at Q_M is a Poisson process with intensity $1/\alpha$; $1/\alpha$ equals the throughput in the closed cyclic queueing system with N customers, obtained from the one under consideration by removing Q_M.

Before motivating this approximation assumption, we first demonstrate how it yields simple approximations for $E[b]$ and $E[i_2]$, and hence for T.

Approximation for $E[b]$
According to the assumption, Q_M behaves during a B-period like an M/M/1 queue, say Q, with arrival intensity $1/\alpha$ and mean service time α_M. At the beginning of a B-period, the queue length at Q (Q_M) increases from zero to one. Let $\mu_{j,k}$, $j,k \geqslant 0$, denote the first entrance time in state k starting from state j for the queue length process in Q. Then, *under the approximation assumption,*

$$E[b] = E[\mu_{1,L+1}]. \tag{2.6}$$

Introducing

$$a := \alpha_M / \alpha,$$

we can prove the following lemma:

LEMMA 2.1

$$E[\mu_{1,L+1}] = \alpha_M \frac{(L+1)(a-1) + (1/a)^{L+1} - 1}{(1-a)^2} - \alpha, \quad a \neq 1, \tag{2.7}$$

$$= \alpha(\frac{1}{2}L^2 + \frac{3}{2}L), \qquad a = 1.$$

PROOF

First observe that

$$E\{\mu_{i,j+1}\} = E\{\mu_{i,j}\} + E\{\mu_{j,j+1}\}, \quad i<j, \tag{2.8}$$

and

$$E\{\mu_{0,1}\} = \alpha. \tag{2.9}$$

Based on the assumption that there are j customers in Q, $E\{\mu_{j,j+1}\}$ can be calculated by just looking at the next event (the departure or the arrival of a customer). Denoting by τ_M a service time at Q_M, by τ a negative exponentially distributed random variable with $E\{\tau\}=\alpha$, and by (A) the indicator function of event A, it holds that

$$E\{\tau_{j,j+1}\} = E\{\tau(\tau_M \geqslant \tau)\} + E\{\tau_M(\tau_M < \tau)\} + E\{\mu_{j-1,j+1}(\tau_M < \tau)\}, \; j = 1,2,\ldots. \tag{2.10}$$

From (2.8), because of the mutual independence of $\mu_{j-1,j+1}$, τ_M and τ in the right hand side of (2.10), it follows that

$$E\{\mu_{j,j+1}\} = E\{\tau(\tau_M \geqslant \tau)\} + E\{\tau_M(\tau_M < \tau)\} + Pr\{\tau_M < \tau\}[E\{\mu_{j-1,j}\} + E\{\mu_{j,j+1}\}].$$

From this $E\{\mu_{j,j+1}\}$ can easily be solved. Using

$$E\{\tau(\tau_M \geqslant \tau)\} + E\{\tau_M(\tau_M < \tau)\} = \frac{\alpha_M}{1+a},$$

we find

$$E\{\mu_{j,j+1}\} = \frac{\alpha_M + E\{\mu_{j-1,j}\}}{a}. \tag{2.11}$$

It is now easy to derive (2.7) from (2.8), (2.9) and (2.11).

Approximation of $E[i_2]$

Under the approximation assumption,

$$E[i_2] = (L+1)\alpha_M + \alpha(\frac{1}{1+a})^{L-1}. \tag{2.12}$$

Indeed, i_2 consists of $L+1$ services in Q (Q_M) and, possibly, an idle period. The probability that Q (and hence Q_M) becomes empty in an I_2-period before a customer is admitted, equals $(1/(1+a))^{L-1}$ (remember that admission is not possible before the queue length has decreased to $L-1$, and that $Pr\{\tau > \tau_M\} = 1/(1+a)$).

In fact it is possible to obtain the *distributions* of b and i_2 under the approximation assumption, but they are not needed in the present study.

From (2.4)-(2.7) and (2.12) the following approximation for the maximum attainable throughput is obtained:

For $a \neq 1$,

$$T \approx \frac{a\frac{(L+1)(a-1) + (1/a)^{L+1} - 1}{(1-a)^2} - 1}{(L+1)\alpha_M + \alpha(\frac{1}{1+a})^{L-1} + \alpha_M \frac{(L+1)(a-1) + (1/a)^{L+1} - 1}{(1-a)^2} - \alpha + \sum_{j=1}^{M-1} \alpha_j}, \tag{2.13}$$

For $a = 1$,

$$T \approx \frac{\frac{1}{2}L^2 + \frac{3}{2}L}{(L+1)\alpha + \alpha(\frac{1}{2})^{L-1} + \alpha(\frac{1}{2}L^2 + \frac{3}{2}L) + \sum_{j=1}^{M-1} \alpha_j}.$$

We would like to emphasize that (2.13) has been derived after the introduction of one single approximation assumption.

Motivation of the approximation assumption

We draw upon results from [3] concerning a closed cyclic queueing model that is identical to the one of Fig. 4, with one exception: all waiting rooms are infinite. In [3] exact results from [4] are used to analyse the influence of the "slowest server" (the server with largest mean service time) on cycle time- and sojourn time distributions. Of particular interest to us is the result for the mean cycle time, $E[C]$, because the throughput T_0 equals $N / E[C]$. It is shown in [3] that, when α_i and α_j are the largest and one-but-largest mean service time:

$$E[C] = N\alpha_i[1 + O((\frac{\alpha_j}{\alpha_i})^N)], \quad N \to \infty, \tag{2.14}$$

so

$$T_0 = \frac{1}{\alpha_i}[1 - O((\frac{\alpha_j}{\alpha_i})^N)], \quad N \to \infty. \tag{2.15}$$

As an example, for $M = 2$,

$$T_0 = \frac{\alpha_2^N - \alpha_1^N}{\alpha_2^{N+1} - \alpha_1^{N+1}} = \frac{1}{\alpha_2}\left[1 - \frac{\frac{\alpha_2}{\alpha_1} - 1}{(\frac{\alpha_2}{\alpha_1})^{N+1} - 1}\right], \quad \alpha_2 \neq \alpha_1.$$

Approximation of $E[C]$ by $N\alpha_i$ and of T_0 by $1/\alpha_i$ yields remarkably accurate results, even when N is rather small and α_j is close to α_i. The worst case is when all mean service times are equal; in that case $T_0 = N/((N+M-1)\alpha_1)$. The explanation for the high accuracy is that in a closed cyclic system the queue with the slowest server is seldom empty - so the departure process of this queue is closely approximated by a Poisson process, and the throughput in the system is closely approximated by the service rate at this queue.

Now consider our closed cyclic model, with finite waiting room in Q_M, during a B-period. We distinguish two cases.

Case (i): α_M is not the largest mean service time.

The output rate from Q_{M-1} during a B-period is in this case hardly influenced by the presence of Q_M. Hence this output rate may be put equal to the throughput in the cyclic queueing model consisting of $Q_1, \ldots, Q_{M-1}$. The output process is closely approximated by a Poisson process, corresponding to the departure process from the queue with the slowest server.

The above reasoning still holds if there is not one unique slowest server in the set $\{Q_1, \ldots, Q_{M-1}\}$. The assumption of a Poisson process is less justified, but the departure rate from Q_{M-1} is still closely approximated by the throughput in the $(M-1)$-queue cyclic model.

Case (ii): α_M is the largest mean service time.

If L would have been infinite, the throughput would have been dominated by Q_M. But for finite L, the other queues still contain at least $N - L$ customers; the queue with the slowest server out of these $M-1$ queues will hardly ever be empty, and the output process from Q_{M-1} is still determined by that queue.

A clear illustration of this fact is provided by the model with $M=2$ queues. During a B-period Q_1 contains at least $N-L>0$ customers, and the departure process from Q_1 is a Poisson process with rate $1/\alpha_1$, regardless of Q_2. Indeed, our approximation procedure (which involves only one approximation assumption) yields exact results for the case of $M=2$ queues.

The above reasoning can also be applied to the I_2-period. The approximation will be slightly less accurate, because Q_M may contain close to L customers during the larger part of this period, and because there may be less than $N-L$ customers in $\{Q_1, \ldots, Q_{M-1}\}$ due to overflow.

In the next section numerical results will be presented, which illustrate the accuracy of our approximation and which reveal the influence of window flow control and retransmissions on the throughput in a virtual circuit. We close the present section with some remarks.

REMARK 2.1

Additional motivation for the approximation assumption is provided by referring to the Chandy-Herzog-Woo theorem (Norton's theorem in the context of electrical circuits; cf. Chandy et al. [6] and Lavenberg [11]). This theorem applies to the entire class of product-form networks, but for our purpose it is sufficient to restrict attention to the cyclic product-form network of Fig. 4 with $L=\infty$. The principle is as follows. Study one queue, say Q_M, by replacing the network by a two-queue closed cyclic system consisting of Q_M and one service center, C. C should be "flow-equivalent" to $\{Q_1, \ldots, Q_{M-1}\}$, i.e., the service rate at C, when it contains j customers, should equal the arrival rate at Q_M when $Q_1, \ldots, Q_{M-1}$ together contain j customers. According to the Chandy-Herzog-Woo theorem, this replacement procedure yields *exact* results for Q_M when C's service rate with j customers is chosen to be the throughput in the $(M-1)$-queue closed cyclic system $\{Q_1, \ldots, Q_{M-1}\}$ with j customers.

If this theorem would still hold in our model, then we could study Q_M during B- and I_2-periods by introducing an arrival rate $\mu(j)$, equal to the throughput in the $(M-1)$-queue model $\{Q_1, \ldots, Q_{M-1}\}$ with j customers. But this is just the idea we have used, apart from the fact that we have taken $\mu(j)$ equal to $\mu(N)=1/\alpha$ for all j. Note that $j \geqslant N-L$ in the B-period, while the throughput hardly varies with the number of customers. We could have calculated $\mu(j)$ for all j-values, but this would have complicated the approximation without a clear improvement of accuracy.

REMARK 2.2

In [2] the throughput in a virtual circuit is studied for the (idealized) case in which *overflow is observed as soon as it occurs.* Again the regenerative approach is applicable. In this case the I_2-period has length zero, and at the beginning of an I_1-period there are $N-L$ customers in Q_1 and L customers in Q_M. Using M/M/1-theory, it is easy to calculate the probability p_j of there being j customers in Q_M at the end of the I_1-period. The mean length of the B-period is subsequently estimated by calculating mean entrance times $E[\boldsymbol{\mu}_{j,L+1}]$, multiplying by p_j and summing over j.

In Section 3 we numerically compare the maximum attainable throughput for the case with NACK's and for this idealized situation in which overflow immediately leads to retransmission. For the latter case, too, the approximation procedure yields exact results when $M=2$.

REMARK 2.3

The approximation procedure can also be modified to incorporate a time-out mechanism. It is reasonable to assume that the time-out is chosen so large, that the probability of a message time-out excess due to very long sojourn times (instead of overflow) is negligibly small. Now a similar regeneration approach as before can be used, with similar choices of the I_1- and B-periods. The I_2-period should be replaced by a period between overflow and time-out excess. The length of this period can be estimated.

REMARK 2.4
In the present model the network delay of ACK's and NACK's is neglected. Such delays can be modelled by the introduction of an infinite server queue between Q_M and Q_1. The approximation method can easily be adapted to this new situation.

REMARK 2.5
A performance measure which has not been discussed until now is the overflow intensity (O) defined as

$$O = \frac{1}{E[\textit{time between two subsequent arrivals of a NACK at the source node}]}.$$

Obviously

$$O = \frac{1}{E[i_1]+E[b]+E[i_2]}.$$

Hence, the approximation method for the throughput can also be used to approximate the overflow intensity.

3. NUMERICAL RESULTS

In this section we present tables and graphs to show the accuracy of our throughput approximation (2.13), and to reveal the influence of window flow control and overflow on the throughput. From the large variety of possible parameter combinations, some representative examples have been chosen. For these examples the approximation results are compared to results obtained by simulation.
Table I and Figure 5 expose the dependence of the throughput T on the buffer size L; Table II and Figures 6a, 6b show the influence of the number of customers N (window size) on the throughput, and Table III and Figure 7 consider models with a large number of queues.
In all tables three possible situations are distinguished:
(i) $\max\{\alpha_i,\ i=1,...,M-1\}<\alpha_M$,
(ii) $\max\{\alpha_i,\ i=1,...,M-1\}=\alpha_M$,
(iii) $\max\{\alpha_i,\ i=1,...,M-1\}>\alpha_M$.

The tables show that the approximation yields very good results, even when the overflow intensity is large (as in case (i) above). The relative approximation errors indicated in the tables are defined as:

$$\frac{\textit{approximation result} - \textit{simulation result}}{\textit{simulation result}} \times 100\% .$$

We now discuss the results in some detail.

In Table I the approximation is tested for different values of the buffer size (L) at Q_M. $T_{sim.}$ and $T_{approx.}$ denote the simulation result and the approximation result for the throughput, respectively. In all cases $M=4$, $N=6$ and $\alpha_4=1$. Of course the throughput grows when L becomes larger.
The last four cases in Table I (with $\alpha_1=2>>0.2=\alpha_2=\alpha_3$) provide additional justification for the idea behind our approximation assumption: if there is a unique queue in $\{Q_1, \ldots, Q_{M-1}\}$ with largest mean service time, then the arrival process at Q_M during a B-period is very well approximated by a Poisson process with intensity $1/\max\{\alpha_i,\ i=1,...,M-1\}$ (cf. Section 2). Indeed, the approximation yields extremely accurate results for these cases.
A typical curve of the throughput as function of the buffer size is shown in Figure 5. The results are compared with the throughput in the corresponding model with infinite buffer space at all queues (see Figure 3). Of course the throughput remains constant as soon as L becomes larger than the number of customers (N) in the system. We also display the throughput for the idealized case in which overflow is observed as soon as it occurs (cf. Remark 2.2). As expected, the throughput for that case is somewhat higher.

In Table II it is shown that the approximation method accurately reflects the influence of the window size (N) on the maximum throughput. Again the results are presented for a model consisting of $M=4$ queues. It can be concluded that the dependence of the throughput on N increases when the overflow intensity becomes smaller. In the first cases of Table II (with $\alpha_4=2>1=\max\{\alpha_1,\alpha_2,\alpha_3\}$, hence with a high overflow intensity) the throughput hardly depends on N.
In Figure 6a the throughput, as a function of N, is depicted for a model consisting of four queues of which Q_4 has two buffer places ($M=4$, $L=2$). To show the influence of the finite buffer space, the throughput results for the corresponding model with infinite buffer space are indicated in the same figure. It is interesting to see how, due to the finite buffer space (which makes retransmissions necessary in case of overflow), the throughput decreases if N is enlarged from two to three, whereas the throughput increases for the model with infinite buffer space. From Figure 6b it follows that this is not a general phenomenon. The shape of the throughput curve depends on the relation between α_m and $\max\{\alpha_i,\ i=1,...,M-1\}$. Indeed, for the case of Figure 6b, $\alpha_M << \max\{\alpha_i,\ i=1,...,M-1\}$ (small overflow intensity) whereas for the case depicted in Figure 6a, $\alpha_M=\max\{\alpha_i,\ i=1,...,M-1\}$ (large overflow intensity).

Table III contains some results for larger values of M, the number of queues. In all cases $M=11$ and $\alpha_i=1$, $i=1,...,10$. The approximations for these larger networks are still very accurate.
Figure 7 shows the influence of M on the throughput. It is seen that the difference with the throughput in the model with infinite buffer space becomes smaller when the number of queues grows. Obviously this is due to the fact that the overflow intensity in the model with finite buffer space decreases when M becomes larger (the customers are spread over more queues).

L	α_1	α_2	α_3	$T_{sim.}$	$T_{approx.}$	% error
1	0.5	0.2	0.2	0.359	0.361	+0.6
2	0.5	0.2	0.2	0.565	0.571	+1.1
3	0.5	0.2	0.2	0.672	0.682	+1.5
4	0.5	0.2	0.2	0.742	0.748	+0.8
1	1	1	1	0.248	0.247	-0.4
2	1	1	1	0.415	0.420	+1.2
3	1	1	1	0.519	0.524	+1.0
4	1	1	1	0.587	0.589	+0.3
1	2	1	1	0.217	0.213	-1.8
2	2	1	1	0.348	0.349	+0.3
3	2	1	1	0.415	0.417	+0.5
4	2	1	1	0.449	0.451	+0.4
1	2	0.2	0.2	0.240	0.242	+0.8
2	2	0.2	0.2	0.372	0.374	+0.5
3	2	0.2	0.2	0.438	0.436	-0.5
4	2	0.2	0.2	0.467	0.467	+0.0

Table I. Comparison of approximation results with simulation results for various values of the buffer size (L). In all cases $M=4$, $N=6$, $\alpha_4=1$.

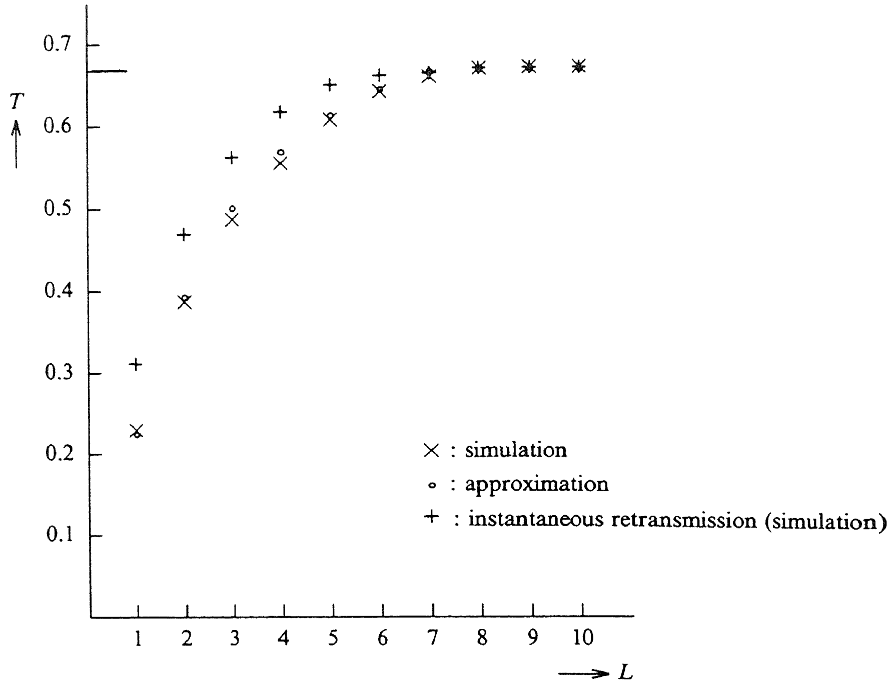

Fig. 5 The influence of L on the throughput. $M=5$, $N=8$, $\alpha_i=1$, $i=1,...,5$.

N	L	α_1	α_2	α_3	α_4	$T_{sim.}$	$T_{approx.}$	% error
4	3	1	0.5	0.5	2	0.336	0.335	-0.3
6	3	1	0.5	0.5	2	0.336	0.335	-0.3
8	3	1	0.5	0.5	2	0.338	0.336	-0.6
10	3	1	0.5	0.5	2	0.338	0.336	-0.6
4	5	2	1.5	1.5	2	0.323	0.323	0.0
6	5	2	1.5	1.5	2	0.339	0.339	-0.0
8	5	2	1.5	1.5	2	0.341	0.344	+0.9
10	5	2	1.5	1.5	2	0.342	0.347	+1.5
4	2	0.2	1	1	0.1	0.770	0.778	+1.0
6	2	0.2	1	1	0.1	0.825	0.836	+1.3
8	2	0.2	1	1	0.1	0.858	0.868	+1.2
10	2	0.2	1	1	0.1	0.875	0.888	+1.5

Table II. Comparison of approximation results with simulation results for various values of the window size (N). In all cases $M=4$.

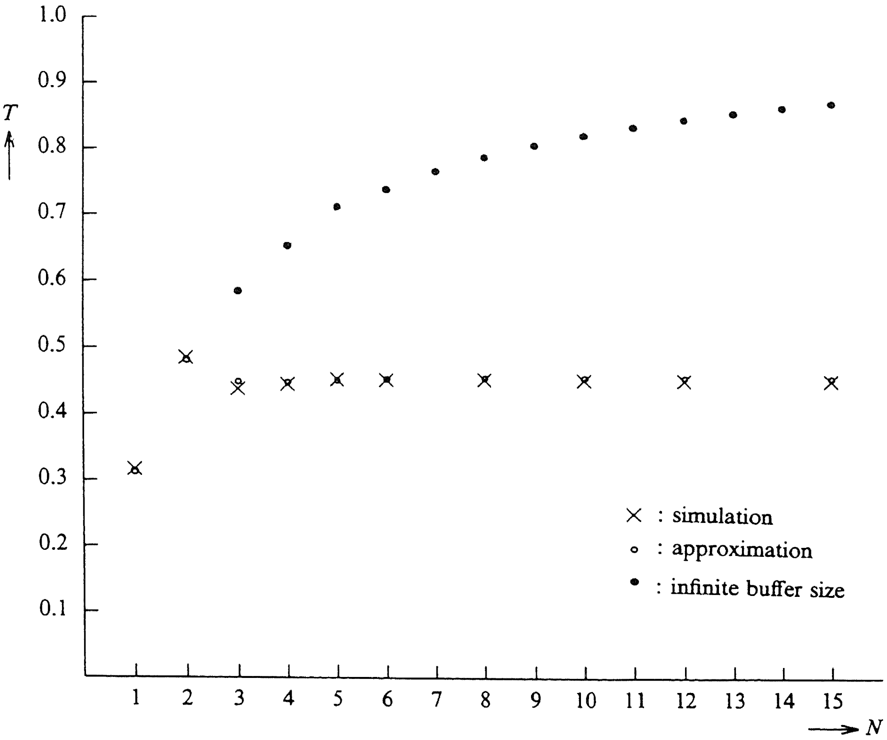

Fig. 6a The influence of N on the throughput. $M=4$, $L=2$, $\alpha_1=0.2$, $\alpha_i=1$, $i=2,3,4$.

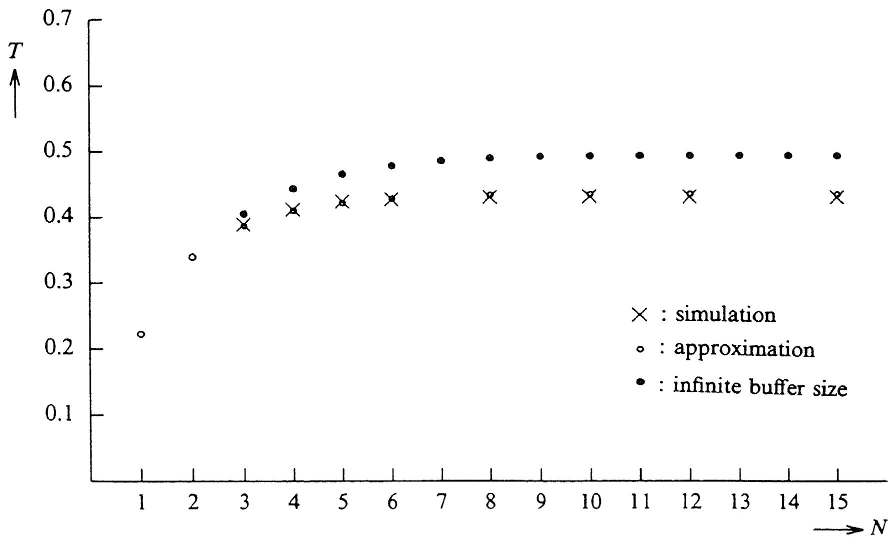

Fig. 6b The influence of N on the throughput. $M=4$, $L=2$, $\alpha_1=2$, $\alpha_2=\alpha_3=1$, $\alpha_4=0.5$.

N	L	α_{11}	$T_{sim.}$	$T_{approx.}$	% error
30	5	5	0.138	0.136	-1.4
30	5	1	0.562	0.565	+0.5
30	5	0.2	0.767	0.769	+0.3
30	10	5	0.164	0.162	-1.2
30	10	1	0.695	0.713	+2.6
30	10	0.2	0.767	0.769	+0.3

Table III. Comparison of approximation results with simulation results for the case that the system consists of $M = 11$ queues. In all cases $\alpha_i = 1$, $i = 1,..,10$.

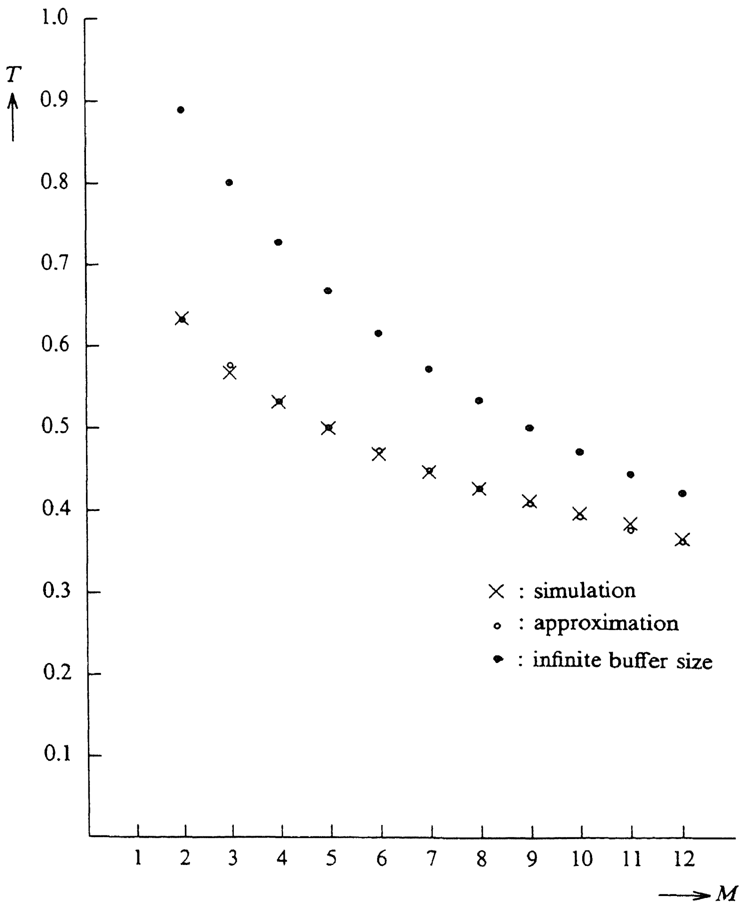

Fig. 7 The influence of the number of queues (M) on the throughput. $N = 8$, $L = 3$, $\alpha_i = 1$, $i = 1,...,M$.

4. Conclusions

A simple throughput approximation for virtual circuits in computer communication networks with window flow control and overflow has been derived and investigated. The results can be summarized as follows.

- The approximation is exact for the case of a single-hop circuit (no intermediate nodes between the source node and the destination node).
- In general, the relative error of the approximation is at most a few percent.
- The approximation formula (2.13) and its construction give much insight into the qualitative and quantitative behaviour of the throughput as a function of several system parameters (window size, buffer size of the destination node, number of nodes, transmission speeds).

3 For example, the throughput is seen to be almost independent of the window size when the overflow intensity is high.

Finally, we mention some points which are considered for future research.

(i) The end-to-end delay of messages over a virtual circuit of the network is an important performance measure. The approach of the present paper may be useful for obtaining end-to-end delay approximations. Preliminary results are contained in [2].

(ii) The ultimate goal of studies like the present one, is to obtain insight into the behaviour of a network consisting of several virtual circuits. When studying one circuit of such a network in isolation, the presence of other circuits can be represented by external traffic entering and leaving the nodes of the isolated circuit. In the case of infinite buffer sizes and negative exponentially distributed service times (cf. Fig.3), the assumption that the external arrival streams are Poisson leads to an easily analyzable queueing network of BCMP type (cf. Hayes [9], Section 11.3.2). It appears that one can eliminate the external streams in this case by adjusting the service rates. If node i has mean service time α_i and external arrival rate λ_i, then $1/\alpha_i^{adjusted} = 1/\alpha_i - \lambda_i$ (and no external traffic) leads to exact throughput results for the virtual circuit. This justifies the study of virtual circuits *without* interfering traffic, in the case without buffer size restrictions. It remains to be investigated whether a similar approach can be used when not all buffer sizes are infinite.

(iii) In relation to Remark 1.1 concerning the instantaneous removal of messages from the nodes of the virtual circuit, we mention the investigation of a model in which old packets are neither removed nor ignored. Generally speaking, the presence of old packets in the network is the source of throughput degradation in a situation with an increasing offered load. The method described in the present paper can easily be modified to incorporate the presence of old packets; introduction of one additional approximation assumption is needed. Preliminary numerical results are encouraging.

REFERENCES

1. F. BASKETT, K.M. CHANDY, R.R. MUNTZ AND F.G. PALACIOS (1975). *Open, closed and mixed networks of queues with different classes of customers,* J. ACM 22, 248-260.
2. J.L. VAN DEN BERG (1986). *Queueing analysis of a virtual circuit in a computer communication network with window flow control,* Report OS-N8602, Centre for Mathematics and Computer Science.
3. O.J. BOXMA (1985). *Response times in cyclic queues - the influence of the slowest server,* preprint 380, Mathematical Institute, University of Utrecht.
4. O.J. BOXMA, F.P. KELLY AND A.G. KONHEIM (1984). *The product form for sojourn time distributions in cyclic exponential queues,* J. ACM 31, 128-133.
5. V.G. CERF AND R. KAHN (1974). *A protocol for packet network intercommunication,* IEEE Trans. Commun. 22, 637-648.
6. K.M. CHANDY, U. HERZOG AND L.S. WOO (1975). *Parametric analysis of queueing networks,* IBM J. Res. Develop. 19, 43-49.
7. J.W. COHEN (1976). *On Regenerative Processes in Queueing Theory,* Springer-Verlag, Berlin.
8. M. GERLA AND L. KLEINROCK (1980). *Flow control: a comparative survey,* IEEE Trans. Commun. 28, 553-574.
9. J.F. HAYES (1984). *Modeling and Analysis of Computer Communication Networks,* Plenum Press, New York.
10. L. KLEINROCK (1964). *Communication Nets - Stochastic Message Flow and Delay,* McGraw-Hill, New York.
11. S.S. LAVENBERG (1983). *Computer Performance Modeling Handbook,* Academic Press, New York.
12. M. REISER (1979). *A queueing network analysis of computer communication networks with window flow control,* IEEE Trans. Commun. 27, 1199-1209.
13. M. REISER (1982). *Performance evaluation of data communication systems,* Proc. IEEE 70, 171-196.
14. M. REISER AND H. KOBAYASHI (1975). *Queueing networks with multiple closed chains: theory and computational algorithms,* IBM J. Res. Develop. 19, 282-294.