Revised Jan. '88

Semaphore queues: Modelling multi-layered window flow control mechanisms

by

S. Fdida[1], H.G. Perros[2]*, A. Wilk[3]

[1]Laboratoire MASI, Université P. et M. CURIE
4, Place Jussieu, 75252 PARIS cedex 05, FRANCE

[2] Computer Science Department and Center for Communication and Signal Processing
North Carolina State University,Raleigh, NC 27695-8206, USA

[3] Polish Academy of Sciences, Dept. of Complex Control Systems
ul. Baltycka 5, 44-100 GLIWICE, POLAND

CCSP-TR-88/9
February 1988

* Supported by a grant from AIRMICS through the Center for Communication and Signal Processing

ABSTRACT

We present an open queueing network for analysing multi-layered window flow control mechanisms consisting of different subnetworks. The number of customers in each subnetwork is controlled by a semaphore queue. The queueing network is analysed approximately using decomposition and aggregation. The approximation was validated against exact numerical and simulation data, and it was found to have an acceptable relative error. The queueing model is easy to implement and it can be included in a software package. Using this queueing network, we present a case study involving the modelling and analysis of the ISO X25 flow control mechanism.

1 - INTRODUCTION

In a typical local area network or wide area network, a data entity may have to traverse several layers of window flow control mechanisms before it reaches its destination. In this paper, we present a model for analysing the delays introduced by such multi-layered flow mechanisms.

Pennotti and Schwartz [15] analysed a virtual route as a closed network of queues arranged in series, under the assumption of a loss system. That is, they assumed that packets that find a full window upon arrival at the system are discarded. Each queue represented a node on the virtual path. The effect of external arrivals at each node was taken into account by reducing each service rate by the corresponding external arrival rate. Schwartz [23] extended this model to allow the case where the acknowledgements of transmitted packets are withheld until some proportion of the window has been received. Then, a single acknowledgement is sent back. Reiser [19] modelled a computer communication system consisting of many virtual routes with end-to-end window flow control, as a closed multichain queueing network. Each chain represented a different virtual route, under the assumption of a loss system. He proposed a computationally efficient approximation procedure based on mean value analysis for evaluating large closed multichain queueing networks.

Reiser [20] correctly observed that in real situations packets that arrive to find a full window are not lost, but are queued in an input queue. Consequently, he analysed a virtual route with a sliding window as an open tandem queueing network. Associated with this network there was a pool of W tokens, where W is the window size. A customer can not traverse the network unless it has a token. A customer arriving at the network to find the token queue depleted, is queued in the input queue. Upon departure of a customer from the network, the token is returned back to the pool after a delay (acknowledgement delay). This queueing network was analysed approximately by reducing the network to a flow equivalent server, thus simplifying the analysis of the input queue. Varghese, Chou, and Nilsson [26] analysed the same open queueing model without an acknowledgement delay using the same approximation method. However, unlike the above paper, the service process of the flow equivalent server was characterized by a Coxian distribution. A good review of analytical methods for evaluating data communication systems can be found in Reiser [21]. More recently, Gihr and Kuehn [5] analyzed multi-layered protocol systems using hierarchical decomposition and aggregation techniques. The window flow mechanism was modelled as an open queueing system controlled by a pool of tokens, as described in Reiser [20].

Virtual routes with node-to-node window flow control have been modelled as open tandem configurations of finite capacity queues (see Caseau and Pujolle [3]). Each node on the virtual route is represented by a finite queue, whose capacity is equal to the node's window size. A server is said to be blocked if the next downstream queue is full. For further details and references on queueing networks with finite capacity see Altiok and Perros [1].

The problem of analysing window flow control mechanisms, can be formulated as a single or multiple class closed queueing network with a population constraint. These models were originally developed for multiprogramming systems. If the population constraint is lifted, the resulting queueing network is assumed to be of the product-form type. The population constraint is imposed as follows. In the single class case, it is assumed that a subnetwork is subject to a population constraint. That is, only up to a predefined number of customers are allowed in the subnetwork. The remaining customers are queued in an input queue. In the multi-class case, for a particular subnetwork, each class has its own population constraint. These models have been analysed approximately by several authors. For a review of these approximations see Thomasian and Bay [25]. Open queueing networks with population constraint have also been considered. For two-node queueing systems see Perros [17] and the references within. Lam [10] extended the class of multichain queueing networks of the product-form type to include mechanisms of state-dependent lost and triggered arrivals. These

mechanisms permit for a more general type of population constraint. Finally. Goto, Takahashi, and Hasegawa [6] analysed an open tandem configuration with finite buffers and with an overall population constraint. This network permits the modelling of end-to-end and node-to-node window flow control mechanisms.

The problem of simultaneous resource possession arises in multiprogramming systems. In such systems, a job during its execution may require service from more than one server at the same time. This problem has been analysed using open or closed queueing networks (see Perros [16], Jacobson and Lazowska [7], and Freund and Bexfield [4]). A closely related topic, is the problem of serialization delays as arises in critical software sections and database locks. This model has been analysed by Agrawal and Buzen [2], Thomasian [24], and Jacobson and Lazowska [8].

In this paper, we present an open queueing network for analysing multi-layered window flow control mechanisms. These flow control mechanisms may be nested in any arbitrary way. This permits us to model node-to-node and end-to-end window flow controls. Also, each protocol layer at a switching node can be modelled by a separate queue, thus allowing us to represent delays introduced at each layer in a node. Each window flow control mechanism is modelled in a similar fashion as in Reiser [20] and in Gihr and Kuehn [5], through the means of a semaphore queue. The queueing models in this paper are analysed using standard hierarchical decomposition and aggregation.

This paper differs from other papers on window flow control in the following way. It deals with open queueing networks unlike papers [15], [20], and [23]. Papers [21], [26], and [5], deal with open queueing networks, but only one window flow control was considered. In papers [3], [1], and [6], each node is represented by a finite capacity queue, which implies a zero acknowledgement delay. This assumption was not made in this paper. Also, in these papers, the problem of multiple nested window flow control was not addressed. The problem of population constraint has mostly been analysed within the context of closed queueing networks. The open queueing models that have been proposed are either two-node models (see [16]), or require the assumption of a loss system (see [10]). Finally, the models reported for the analysis of simultaneous resource possession and serialization delays are either based on closed queueing networks, or they have been formulated specifically for multiprogramming systems.

In section 2, we introduce the concept of semaphore queue. An approximate solution to a queueing network involving one semaphore queue is given in section 2.1. In section 3, we give an approximation algorithm for analyzing a queueing network with multiple semaphore queues. This algorithm is validated in section 4. A case study involving the modelling and analysis of the ISO X25 flow control mechanism is given section 5. Finally, the conclusions are given in section 6.

2 - A QUEUEING NETWORK WITH A SINGLE SEMAPHORE

The management of a shared resource can be carried out efficiently using a semaphore. A semaphore station (S) consists of an input queue f(S) and a token queue e(S). A customer arriving at the semaphore queue requests a token. The customer departs immediately, if there is a token available in queue e(S). Otherwise, the customer is blocked and it is forced to wait in the input queue f(S) until a token becomes available. Therefore, if there are tokens in e(S), then there are no customers in the input queue. On the other hand, if there are customers in the input queue, then e(S) is empty.

A customer having received a token, leaves the input queue and enters network 1, as shown in figure 1. When it finally departs from network 1, the token is returned back to the token queue via network 2. The total number of tokens available is fixed to C. That is, network 1 can be used at most by C customers. Also, at any time, the customers in network 1 plus the returning tokens in network 2 is less or equal to C. Networks 1 and 2 are assumed to be of the BCMP type. Customers arrive at the semaphore queue in a Poisson fashion at the rate λ. We note that this model is similar to the one studied in [5].

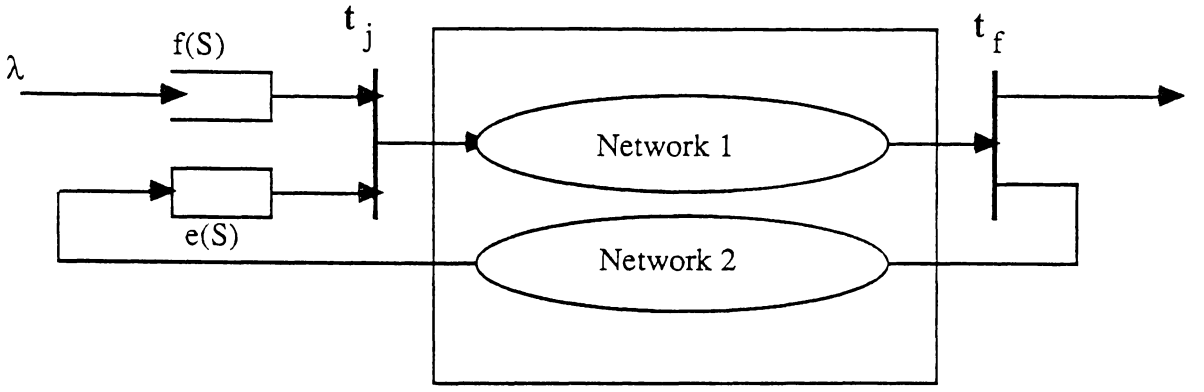
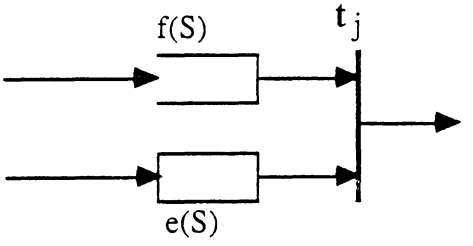
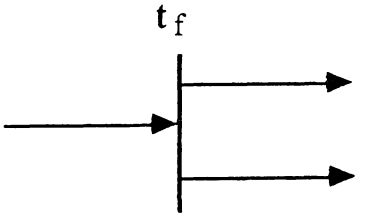


Figure 1: A queueing network with a single semaphore

In figure 1, we introduce two symbols commonly used in Petri Nets in order to depict the fork and join operation. In particular the join symbol



depicts the following operation. At the instance that queues f(S) and e(S) contain a customer each, the two customers instantaneously depart from their respective queues and merge into a single customer. The fork symbol



depicts the following operation. A customer arriving at this point, (i.e. departing from network 1) is split into two siblings. We use these two symbols for descriptive convenience.

2.1 THE APPROXIMATION ALGORITHM

Let us consider the queueing network described above and shown in figure 1. An exact analysis of this model is rather difficult. In view of this, we analyze it using decomposition and aggregation. In particular, we first analyze the system shown in figure 2 assuming that the arrival process at queue e(S) is described by a state-dependent arrival rate $\gamma(k)$.

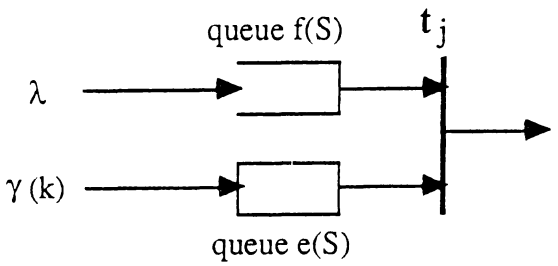


Figure 2: The semaphore queue

This queueing system depicts the semaphore operation described above. The arrival process at queue f(S) is assumed to be poisson distributed, and there are C tokens. We also assume that the inter-arrival times at queue e(S) are exponentially distributed with a rate γ(k), where k is the number of outstanding tokens, i.e. C-k is the number of tokens in queue e(S). The state of the system in equilibrium can be described by the tuple (i,j), where i is the number of customers in queue f(S) and j is the number of tokens in queue e(S). The rate diagram associated with this system is shown in figure 3.

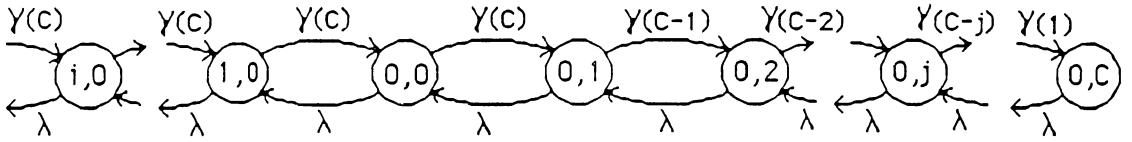


Figure 3: The rate diagram of the semaphore queue

We note that this system is identical to an M/M/1 queue with an arrival rate λ and a state dependent service rate $\gamma(n_q)$ if $n_q \leq C$, and $\gamma(C)$ if $n_q > C$, where n_q is the number of customers in this M/M/1 queue. The random variables i and j are related to n_q as follows: $i = \max(0, n_q - C)$, $j = \max(0, C - n_q)$.

The solution of this system is obtained by a direct application of classical results. Thus, we have

$$p(i,0) = \rho^i \, p(0,0) ,$$

$$p(0,j) = \frac{\Pi(j)}{\lambda^j} \, p(0,0) ; \qquad (2.1)$$

where $\rho = \lambda / \gamma(C)$ and $\Pi(j) = \begin{cases} \prod_{k=0}^{j-1} \gamma(C-k) & , j > 0; \\ 1 & , j = 0. \end{cases}$

The probability p(0,0) is chosen so that the equilibrum state probabilities sum to 1:

$$p(0,0)^{-1} = \frac{1}{1-\rho} + \sum_{j=1}^{C} \frac{\Pi(j)}{\lambda^j} . \qquad (2.2)$$

From (2.1) and (2.2), we obtain the following marginal probabilities for each queue (index 1 is for queue f(S) and 2 for queue e(S)) :

$$p_1(0) = \frac{1 - \rho\,(1 + p(0,0))}{1 - \rho} \qquad (2.3)$$

$$p_1(i) = \rho^i\, p(0,0)\,, \qquad i > 0\,;$$

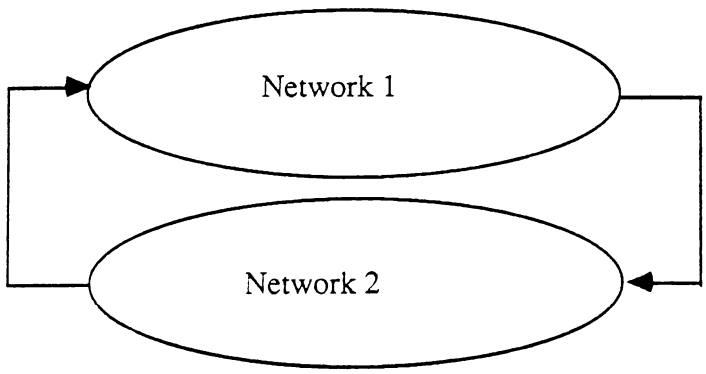
$$p_2(0) = \frac{1}{1 - \rho}\, p(0,0)\,; \qquad (2.4)$$

$$p_2(j) = \frac{\Pi(j)}{\lambda j}\, p(0,0)\,, \qquad 0 < j \leq C\,.$$

Also, from (2.1) and (2.2) we can obtain $p_S(c)$, the probability that there are c customers in queue f(S) and in networks 1 and 2. We have

$$p_S(c) = \begin{cases} p(0,C-c)\,, & 0 \leq c \leq C \\ p(c-C,0)\,, & c > C \end{cases} \qquad (2.5)$$

Now, expressions (2.1) to (2.4) were obtained assuming that $\gamma(k)$ is known. This can be approximately obtained by studying the closed queueing network (call it Q) obtained by linking queueing networks 1 and 2 as shown below.



The analysis of this queueing network can be carried out easily seeing that we have assumed that network 1 and 2 are of the BCMP type. Therefore, we can calculate the throughput of Q with k customers, where k=1,2,...,C. This is then set equal to the arrival rate $\gamma(k)$ of tokens at the token queue e(S).

Let us consider for a moment the last queue in network 2, from which departing customers immediately join queue e(S). Let k' be the number of customers in this queue, and μ its service rate. Then Mailles [12] has shown that

$$\lambda p(i,0) = \mu(i+1,0)\; p(i+1,0)$$

$$\lambda p(0,j) = \mu(0,j-1)\, p(0,j-1)$$

where $\mu(i,j) = \mu[1- p(k' = 0 \mid i,j)]$. The quantity $\gamma(k)$ can be seen as an approximation to $\mu(i,j)$.

We note that in the above formulation, the tokens are sent back via a separate network, network 2. This formulation can be easily changed so that to allow the tokens to travel back over the network used by the customers, network 1. To do this, it suffices to declare two classes of jobs, namely class 1 and class 2 representing customers and tokens respectively. These two classes of jobs will then circulate within network 1 competing for the same resources. This network can be still modelled as a BCMP type of queueing network as long the necessary BCMP assumptions are not violated.

Stability condition

The stability condition can be simply expressed as $\lambda<\gamma(C)$, where $\gamma(C)$ is the maximum throughput of the network Q (see Lavenberg [11]) .

3 - A QUEUEING NETWORK WITH MULTIPLE SEMAPHORES.

In general, we can regard a semaphore queue as the means of controlling the number of customers in a queueing network. Queueing networks controlled by semaphore queues can be combined by imbedding one network within another to make up larger more complex systems. In this section, we give a simple approximation algorithm for computing the solution of such multiple semaphore queueing networks. The algorithm can be used for any nested configuration involving BCMP queueing networks and semaphore queues. For presentation purposes, we consider the queueing network shown in figure 4. In this figure, $SN_{i,n}$, i=1,2,3,4 are four arbitrary BCMP queueing networks, and $S_{i,n-1}$, i=1,2, are two semaphore controlled queueing networks, as shown in figure 5. The index n refers to a level of semaphore control. That is, the semaphore controlled queueing network shown in figure 4, is associated with level n, and the one represented by $S_{i,n-1}$, i=1,2, is associated with level (n-1). Let C_n be the total number of tokens associated with the nth level semaphore controlled queueing network. Presumably, $S_{i,n-1}$, i=1,2, themselves may comprise of lower levels of semaphore queues. Likewise, level n may be imbedded in a higher level semaphore controlled queueing networks.

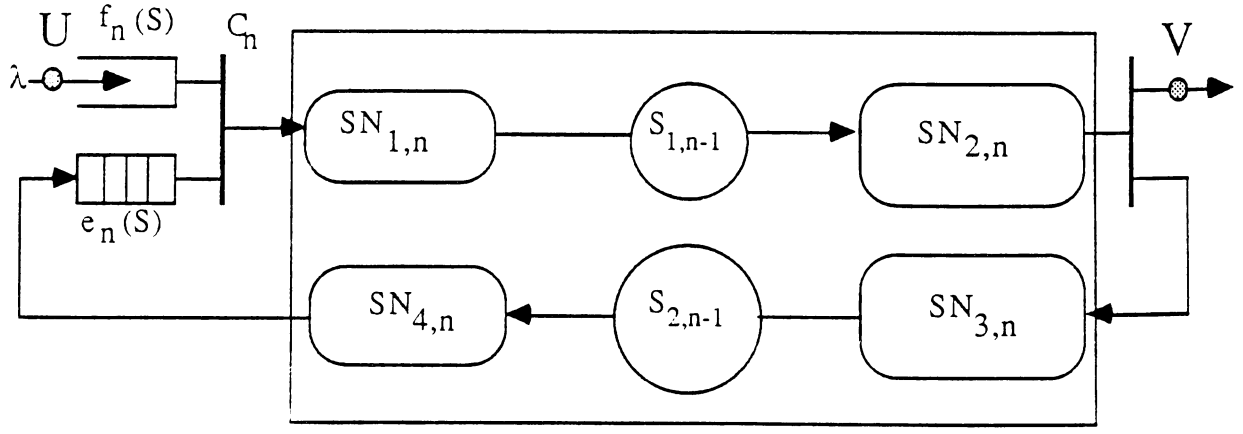


Figure 4: A level n semaphore controlled queueing network.

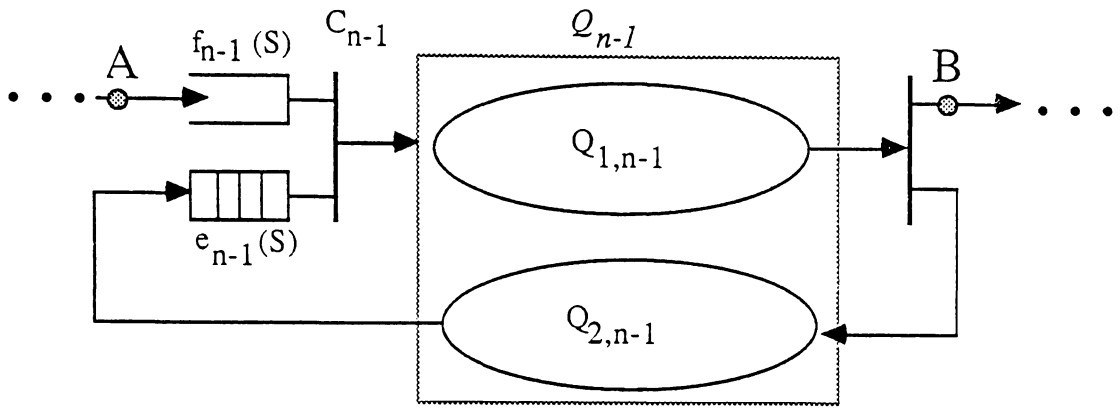


Figure 5: A level (n-1) semaphore subnetwork

Let us first consider the semaphore controlled queueing network $S_{1,n-1}$ as shown in figure 5. As in section 2, we can link networks $Q_{1,n-1}$ and $Q_{2,n-1}$ to form a closed BCMP queueing network. This closed queueing network, call it Q_{n-1}, can be analyzed using the MVA algorithm in order to obtain $R'_{n-1}(k)$, the mean time to traverse $Q_{1,n-1}$ as a function of the number of customers k in Q_{n-1}, where k=1,2,..., C_{n-1}. Similarly, we can obtain $R''_{n-1}(k)$, the mean time to traverse networks $Q_{1,n-1}$ and $Q_{2,n-1}$ as a function of k, the number of customers in Q_{n-1}. Using arguments as in section 2, we have that the rate $\gamma_{n-1}(k)$ at which tokens return back to the token queue is approximately equal to $k/R''_{n-1}(k)$, where k is the number of outstanding tokens. Hence, the mean response time $R_{n-1}(c)$ of a customer between points A and B, conditioned upon that he finds c customers (including himself) in queue $f_{n-1}(S)$ and in Q_{n-1} upon arrival, is approximately given by

$$R_{n-1}(c) = \begin{cases} R'_{n-1}(c), & c \le C_{n-1} \\ R'_{n-1}(C_{n-1}) + (c-C_{n-1})/\gamma_{n-1}(C_{n-1}), & c \ge C_{n-1}, \end{cases} \qquad (3.1)$$

The above expression can be easily derived. For, if $c \le C_{n-1}$, then all the customers are in Q_{n-1}. Thus, our customer is delayed by $R'_{n-1}(c)$. If $c > C_{n-1}$, then only C_{n-1} customers are in Q_{n-1}, and the remaining $(c-C_{n-1})$ are waiting in queue $f_{n-1}(S)$. These customers depart from this queue at the rate at which tokens return back to the token queue $e_{n-1}(S)$, i.e. at the rate $\gamma_{n-1}(C_{n-1}) = C_{n-1}/R''_{n-1}(C_{n-1})$. A customer in queue $f_{n-1}(S)$, therefore, can be seen as receiving a mean service time equal to $R''_{n-1}(C_{n-1})/C_{n-1}$, before it enters $Q_{1,n-1}$, where it is delayed on the average by $R'_{n-1}(C_{n-1})$. Thus, we can obtain the above expression for $R_{n-1}(c)$ when $c \ge C_{n-1}$.

Now, in figure 4, we can approximately substitute $S_{1,n-1}$ by a flow equivalent infinite server queue with a state dependent mean service time equal to $R_{n-1}(c)$, where $c=0,1,...,C_n$. Following similar arguments, we can also approximately substitute $S_{2,n-1}$ by a flow equivalent infinite server queue. Now, let $Q_{1,n}$ and $Q_{2,n}$ be queueing networks consisting of $SN_{1,n}$, $S_{1,n-1}$, $SN_{2,n}$ and $SN_{3,n}$, $S_{2,n-1}$, $SN_{4,n}$ respectively. Then, $Q_{1,n}$, $Q_{2,n}$ and the closed queueing network

consisting of $Q_{1,n}$ and $Q_{2,n}$ (call it Q_n) are all BCMP queueing networks.

If the nth level semaphore controlled queueing network is itself imbedded in a higher level semaphore queue ((n+1)st level), then we can use the arguments given above in order to construct a flow equivalent composite queue. This composite queue will then be used in the (n+1)st level semaphore network in order to substitute the original nth level semaphore network.

Now, let us assume that the nth level semaphore queue is the highest level. In this case, this queueing system can be analyzed using the arguments given in section 2. In particular, we can obtain p(i,j), where i is the number of customers in queue $f_n(S)$ and j is the number of tokens in queue $e_n(S)$. Based on these probabilities we can obtain the mean response time, i.e. the mean time to go from U to V as shown in figure 4, as follows.

Let p(i) and q(j) be the marginal probability distribution that there are i and j customers in queue $f_n(S)$ and in queue $e_n(S)$ respectively. Then, the mean number of customers in queue $f_n(S)$ is

$$L_{f_n(S)} = \sum_{i=1}^{\infty} i\, p(i) \qquad (3.2)$$

Now, let us consider queueing network Q_n. Then, the mean number of customers in queueing network $Q_{1,n}$ can be obtained as follows. Let $p_1(m|h)$ be the probability that there are m customers in $Q_{1,n}$ given that there are h customers in the closed queueing network Q_n, and let $p_1(m)$ be the probability that there are m customers in $Q_{1,n}$. For $m>h$, we have $p_1(m|h)=0$. For $0< m \leq h$, we obtain

$$p_1(m) = \sum_{h=m}^{C_n} p_1(m|h)\, q(C_n-h) \qquad m=1,\ldots, C_n$$

Hence, the mean number of customers in $Q_{1,n}$ is

$$L_{Q_{1,n}} = \sum_{m=1}^{C_n} m\, p_1(m)$$

$$= \sum_{m=1}^{C_n} m \left(\sum_{h=m}^{C_n} p_1(m|h)\, q(C_n-h) \right)$$

$$= \sum_{h=1}^{C_n} q(C_n-h) \sum_{m=1}^{C_n} m\, p_1(m|h)$$

The quantity $\sum m\, p_1(m|h)$, summed over $m=1,..,C_n$, is the mean number of customers in $Q_{1,n}$ given there are h customers in Q_n. Now, the mean response time of $Q_{1,n}$ as a function of the number of customers h in Q_n is $R'_n(h)$. Thus,

$$\sum_{m=1}^{C_n} m\, p_1(m|h) = \gamma_n(h)\, R'_n(h)$$

where $\gamma_n(h)$ is the rate at which tokens return to the token queue, queue $e_n(S)$. We have that $\gamma_n(h)$ is approximately equal to $h/R''_n(h)$, where $R''_n(h)$ is the mean time to traverse $Q_{1,n}$ and $Q_{2,n}$ as a function of h. Thus,

$$\sum_{m=1}^{C_n} m\, p_1(m|h) = h\, \frac{R'_n(h)}{R''_n(h)}$$

and hence

$$L_{Q_{1,n}} = \sum_{h=1}^{C_n} h\, \frac{R'_n(h)}{R''_n(h)}\, q(C_n-h) \qquad (3.3)$$

We have expressed $LQ_{1,n}$ in terms of the quantities $R'_n(\bullet)$, $R''_n(\bullet)$ so that to be consistent with the way we analyze each semaphore controlled queueing network. The mean response time between U and V is

$$R = \frac{1}{\lambda}\, [L_{f_n(S)} + L_{Q_{1,n}}] \qquad (3.4)$$

where $L_{fn(S)}$ and $LQ_{1,n}$ are given by (3.2) and (3.3) respectively.

4 - VALIDATION

The approximation procedure described above was validated against exact numerical and simulation data. In particular, we analysed the model in figure 1 with a single semaphore queue, and the model in figure 4 with two and three levels of semaphore control. In general, the accuracy of the algorithm depends mainly on the utilization of the semaphore queue, expressed as the percent of time queue e(S) is busy, i.e. 1-p(0,C).

Let us now consider the model given in figure 1. Network 1 is assumed to consist of two single server queues in tandem. Network 2 is omitted. (This is because the approximation requires the solution of a closed queueing network, and therefore, it is not necessary to consider network 2 explicitly.) Let C_1, μ_{11}, μ_{12}, and λ be the window size, the service rate at queue 1 and 2 in network 1, and the arrival rate at the input queue. In figures 6, 7, and 8, we give the approximate and exact queue-length distribution $p_S(c)$ of the total number of customers in the input queue and in network 1, for three different values of the input load λ. The exact results were obtained numerically using Neuts' matrix-geometric procedure (see Neuts[14]) as reported in Mailles [12]. Figure 9 gives the approximate and simulated mean response time as a function of the semaphore queue utilization. Confidence intervals are also given for each simulation point. The simulation model was constructed using QNAP2 [18]. The results were obtained by varying the parameters as follows: C_1=3,5,7, λ=1, μ_{11}=0.2,0.5,0.8, and μ_{12}=0.2,0.5,0.8. We note that in the case of

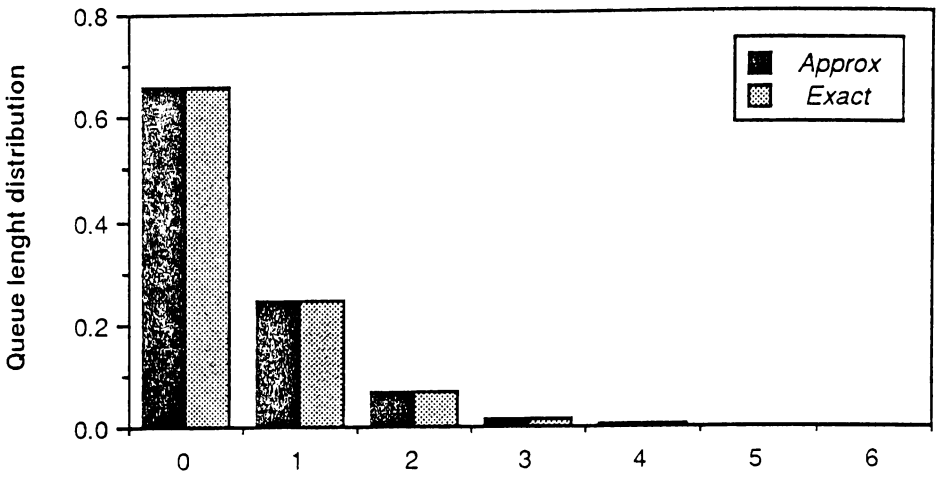


Figure 6: Queue-length distribution of the number of customers. $C_1=3$, $\lambda=.375$, $\mu_{11}=\mu_{12}=2$

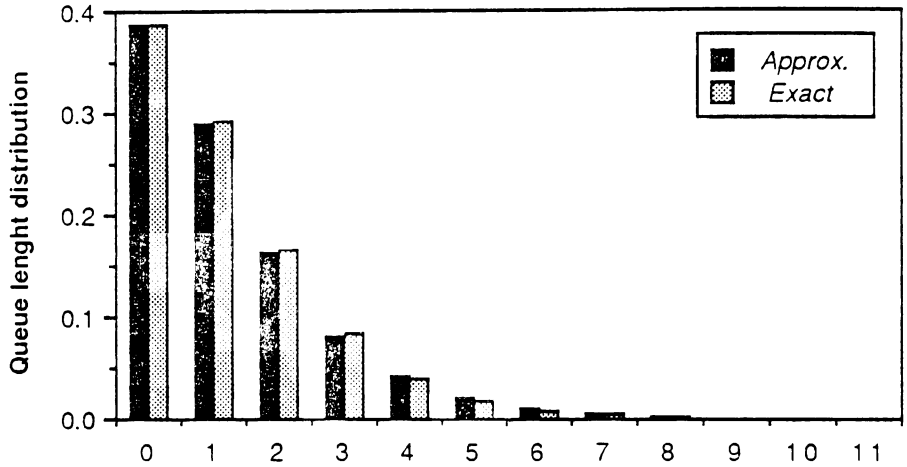


Figure 7: Queue-length distribution of the number of customers. $C_1=3$, $\lambda=.750$, $\mu_{11}=\mu_{12}=2$

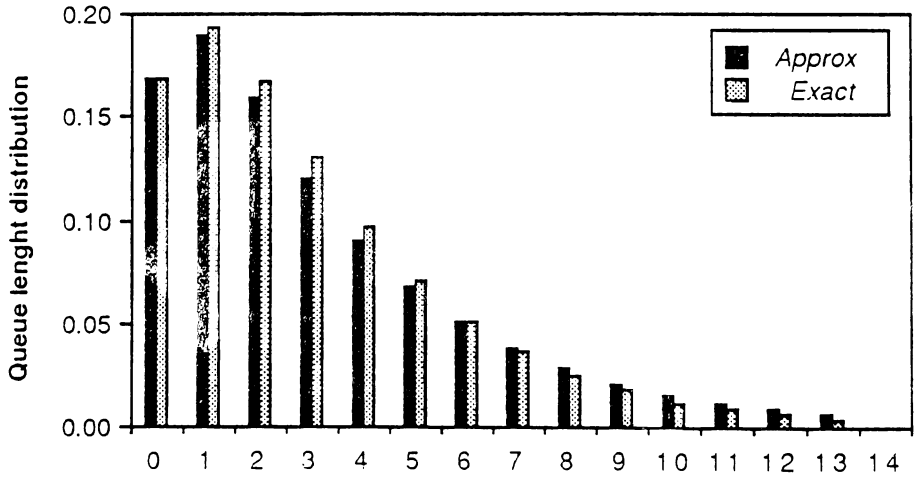


Figure 8: Queue-length distribution of the number of customers. $C_1=3$, $\lambda=1.125$, $\mu_{11}=\mu_{12}=2$

high semaphore utilization, the approximate response time is slightly overestimated. The relative error, expressed as 100(approximate-simulated)/approximate, is given in figure 10. We note that for utilizations of up to .85, the relative error is less than 5%.

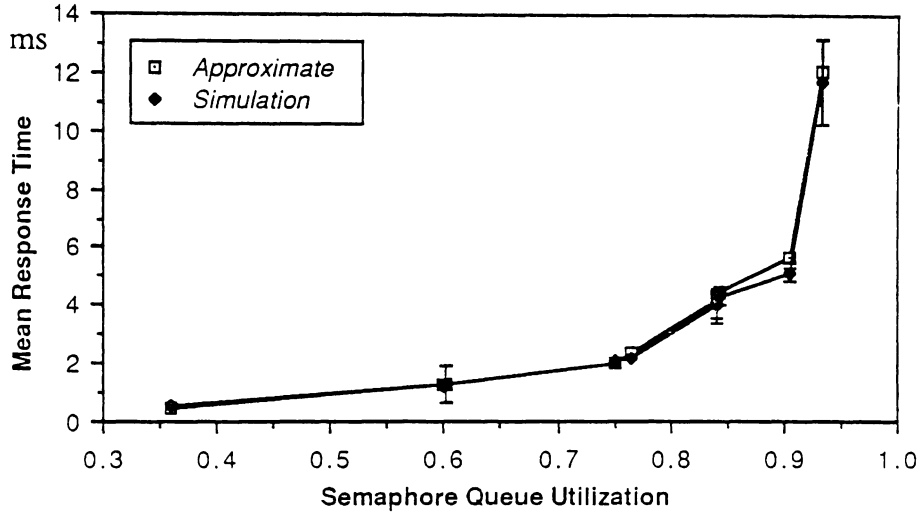


Figure 9: Mean response time vs semaphore queue utilization

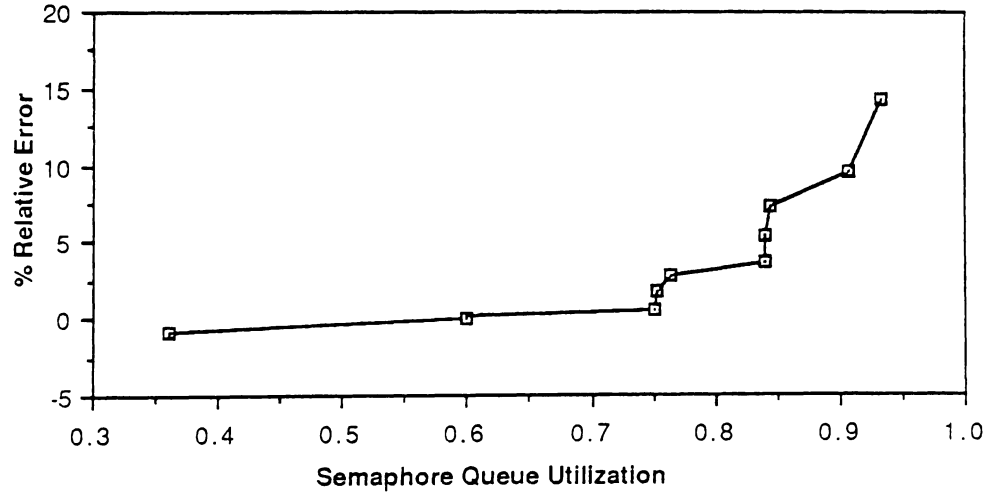


Figure 10: % relative error vs semaphore queue utilization for the results given in figure 9

Now, let us consider the model given in figure 4 with 2 levels, i.e. n=2. The semaphore controlled queueing network $S_{1,1}$ is assumed to be the network analysed above. Networks $SN_{1,2}$ and $SN_{2,2}$ are represented by a single server queue. As above, networks $SN_{4,2}$, $S_{2,1}$ and $SN_{3,2}$ are omitted. Let C_2, μ_{12}, μ_{22}, and λ be the window size, the service rate at the queue representing $SN_{1,2}$ and at the queue representing $SN_{2,2}$, and the total arrival rate at the input queue. Figure 11 gives the approximate and simulated mean response time as a function of the utilization of the level 2 semaphore queue. Confidence intervals (95%) are also given for the simulation results. The values of the level 1 queueing network were chosen so that to correspond to a semaphore queue

utilization ranging from .20 to .76. In particular, the parameters were varied as follows: C_2=5,7, λ=0.2, 0.6, 1, μ_{12}= 0.2, 0.5, 0.8, μ_{22}= 0.2, 0.5, 0.8; and C_1=3, μ_{11}=μ_{12}= 0.5. The relative error observed is given in figure 12. We note again that for semaphore utilizations of up to .85, the relative error is below 5%.

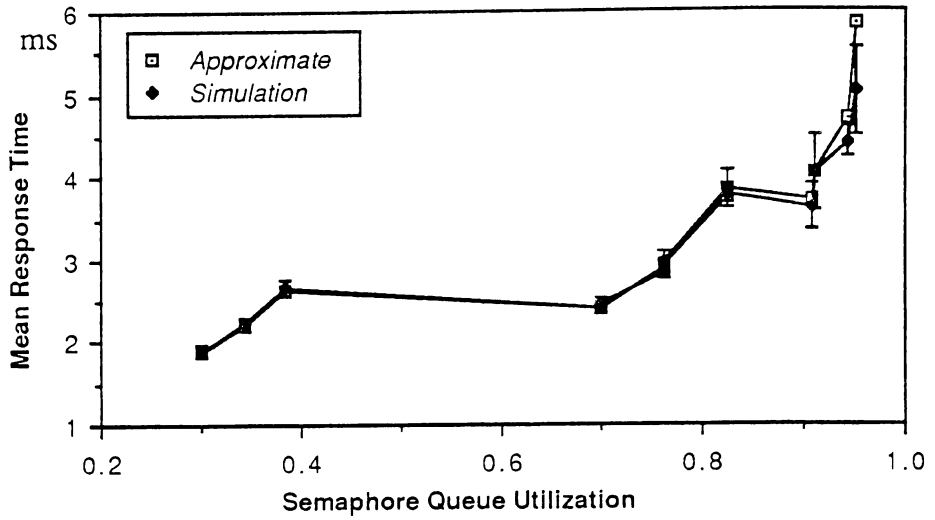


Figure 11: Mean response time vs semaphore queue utilization

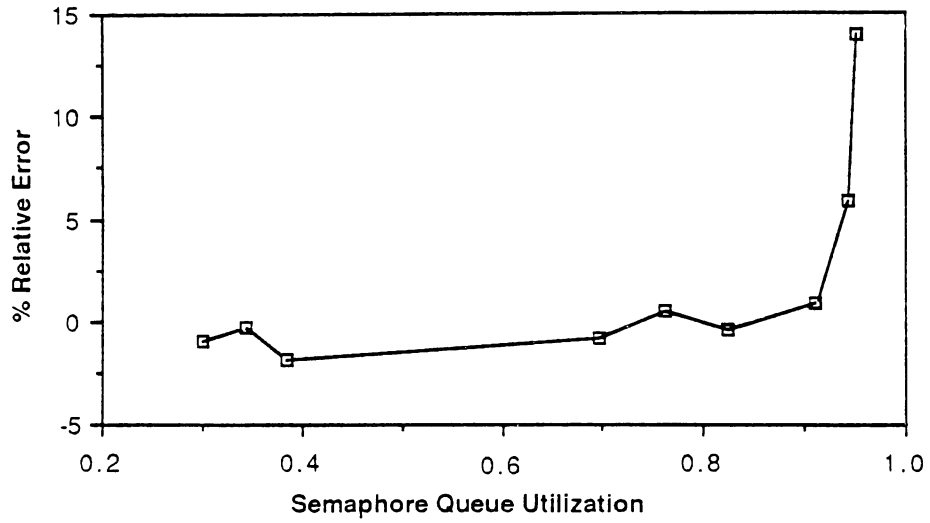


Figure 12: % relative error vs semaphore queue utilization for the results given in figure 11

Finally, we consider the model given in figure 4 with 3 levels, i.e. n=3. The semaphore controlled queueing network $S_{1,2}$ is assumed to be the two-level network analysed above. Networks $SN_{1,3}$ and $SN_{2,3}$ are represented by a single server queue, and networks $SN_{4,3}$, $S_{2,2}$ and $SN_{3,3}$ are omitted. Let C_3, μ_{13}, μ_{23}, and λ be the window size, the service rate at the queue representing $SN_{1,3}$ and at the queue representing $SN_{2,3}$, and the total arrival rate at the input queue. Figure 13 gives the approximate and simulated mean response time as a function of the utilization of the level 3 semaphore queue. For completeness, we also give the mean response time for the two

lower levels, i.e, levels 1and 2. The utilization of the semaphore queue of level 2 and 1 ranged from .10 to .85. The results were obtained by varying the parameters as follows: C_3=7, λ=0.1, 0.6, 1, μ_{13}= 0.5, μ_{23}= 0.2, 0.5, 0.8; C_2=5, μ_{12}=μ_{22}= 0.5; and C_1=3, μ_{11}=μ_{12}= 0.5. The relative error observed for the mean response time for levels 1,2, and 3 is given in figure 14. Again, we observe that the relative error is less than 5% for utilizations up to .85. We note that the relative error for the level 2 model is slightly higher than the one observed in figure 12. This is because of the way the mean response time is calculated. That is, having analysed level 3, we then work backwards using the standard disaggregation approach to compute lower level values.

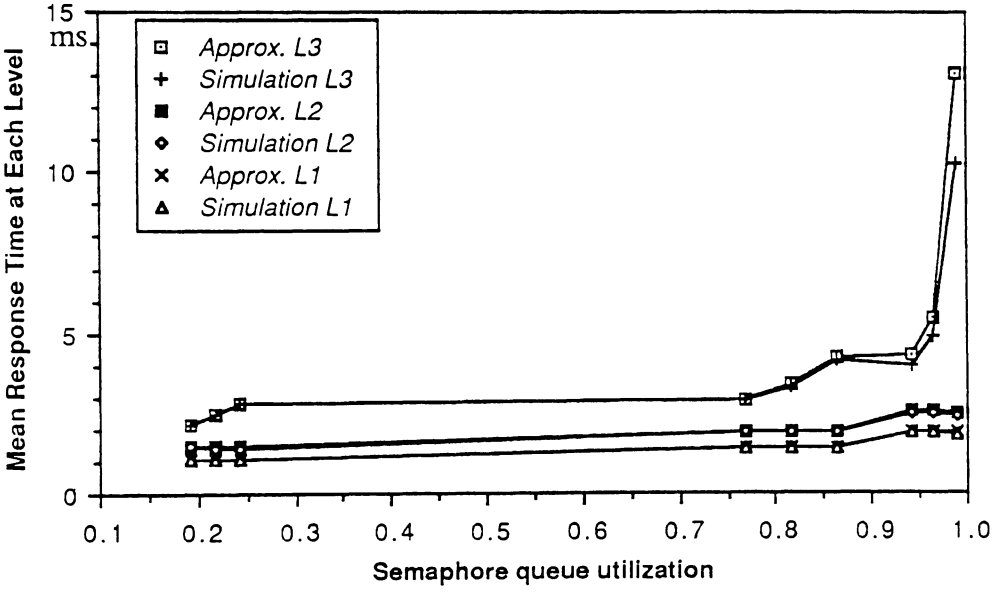


Figure 13: Mean response time vs semaphore queue utilization

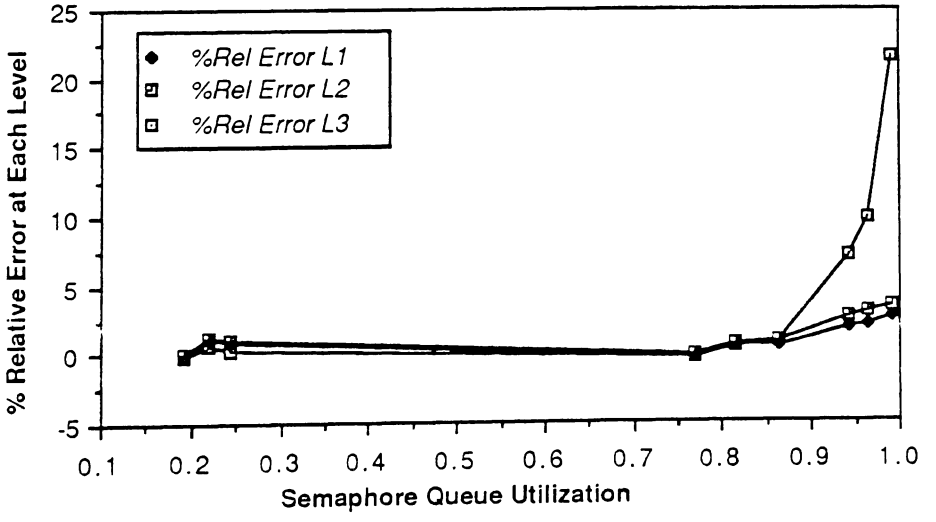


Figure 14: % relative error vs semaphore queue utilization for the results given in figure 13

In general, we observe that the approximate values for the mean response time have a relative error

less than 5% for semaphore queue utilizations of up to .70 . For very high utilizations, the relative error exceeds 5%. However, it is not likely that such cases will be encountered in real life. Similar conclusions can be drawn for other nested configurations of semaphore queues.

5 - CASE STUDY: THE ISO X25 FLOW CONTROL MECHANISM

The past few years have seen important developments in the field of computer communication systems. The ISO reference model defined the protocol layers of a data network architecture. The philosophy of the ISO model lies on the service given by a layer and on the protocols designed for the achievement of each service. Each level delivers a service quality to the upper level and makes use of the service quality provided by the lower level. Thus, in a network, the purpose of each layer is to offer services to higher layers, shielding those layers from the details of how the offered services are actually implemented. Processes at each level run asynchronously. As a result of this, queues are formed at the layer interfaces. The complexity of these systems makes their analysis performance evaluation quite difficult. The queueing model analysed above appears to be easy to use and well suited for studying communication protocols. In this section, we employ this queueing network to model the flow control of an X25 ISO protocol.

The communication system under study consists of the first three layers of the ISO model, i.e. the physical layer, the data link layer, and the network layer (see figure 15). The physical layer (layer

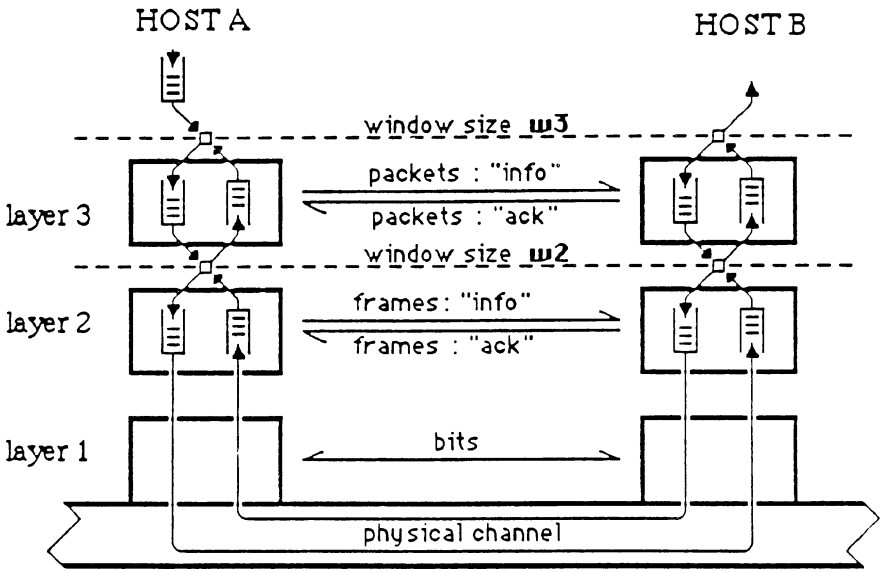


Figure 15: The X25 flow control mechanism

1) is concerned with transmitting bits over a communication medium.The task of the data link layer (layer 2) is to manage the data link control procedure responsible for the error correction over the physical channel. Layer 2 transforms the bit transmission facility into a line that appears to the network layer as being free of transmission errors.Three main frame types are used during a transmission phase: a) information frame (I), for transmitting information, b)receive ready (RR) frame, for the positive acknowledgment of information frame, and c) reject (REJ) frame, for the retransmission of an erroneously transmitted information frame. The network layer (layer 3) controls the operations of the network (i.e. routing, interface, etc.). A protocol unit exchanged on layer 3 is called a packet. There are information and supervision packets. A flow control mechanism is also implemented to avoid congestion (it can be an end-to-end flow control).

This communication system can be modelled using semaphore queues, as shown in figure 16. We assume a unidirectional communication from host A to host B. The external arrival of packets at host A is assumed Poisson distributed with parameter λ. These packets represent the user/application packets. Let W_2 and W_3 be the window size at layers 2 and 3 respectively. An arriving packet joins queue $f_3(S)$ if there are no tokens available in queue $e_3(S)$. When a token becomes available, the packet at the top of the queue is allowed to enter the layer 3 queue where it receives a service at the rate $\mu^A{}_3$. Upon completion of this service, the packet joins queue $f_2(S)$. When a token becomes available, the packet enters layer 2 queue where it receives a service at the rate $\mu^A{}_2$ .This service includes the transmission time of a frame (which is a function of the frame length and the line capacity). Upon completion of this service, the frame joins an infinite server queue reflecting the propagation delay which is usually several times lower than the transmission time.(The probability of overtaking is assumed to be negligible.) Following this layer 1 service, the packet is assumed to be at host B. In particular, it joins host B layer 2 queue where it is served at the rate $\mu^B{}_1$. Upon service completion, the frame may be rejected as being erroneous with probability p_{ei}. In this case, a REJ frame is sent back and the frame is retransmitted. (We assume that an erroneous frame is simply retransmitted by the layer 1 server on a selective repeat basis.) With probability $(1-p_{ei})$ the frame is found error-free and it is allowed to join the layer 3 queue where it is served at the rate $\mu^B{}_3$. Finally, the frame upon completion of its service at the layer 3 queue is delivered to host B. At the same time, a token (representing an RR frame being transmitted back) is placed in queue f'_2 (S). The token is returned back, after transmission and propagation delay, to queue $e_3(S)$ through a path which is similar to the forward path followed by the frame.

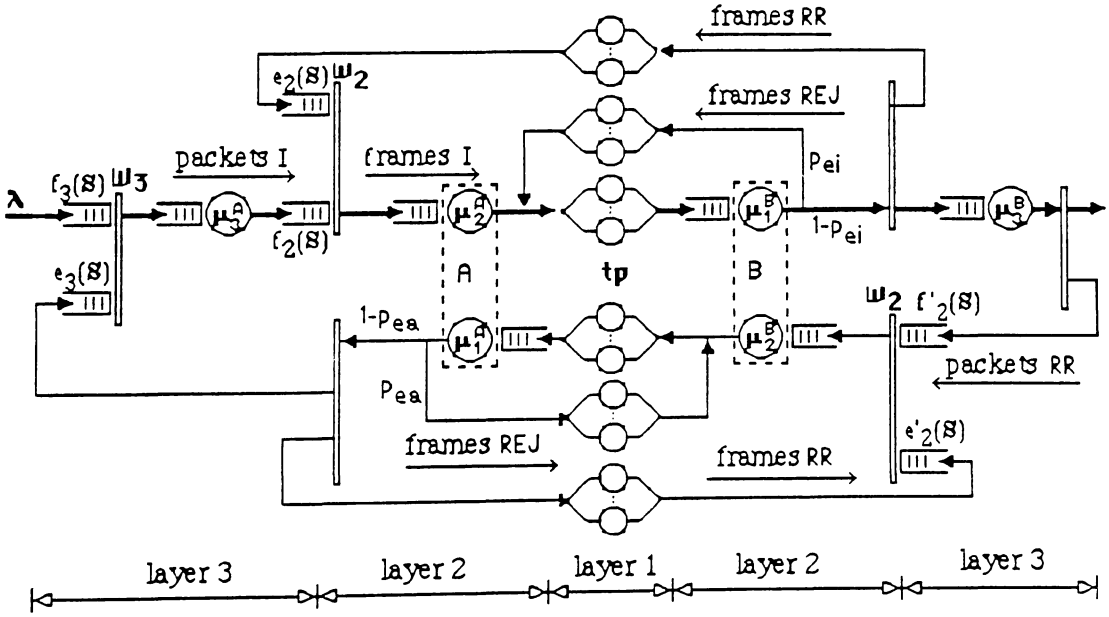


Figure 16: The queueing model of X25 flow control mechanism

The service times of the customers in each of the queues in figure 16 are assumed to be exponentially distributed. (This assumption is not necessary for the infinite server queues).We also assume that each layer is managed by a different processor. Layer 2 (link level) consists of two queues, namely a transmit and a receive queue. These two queues are served by the link layer processor on a priority basis (preemptive policy for the receive queue). In order to apply the approximation algorithm described in section 3, we need first to decouple these two queues. This

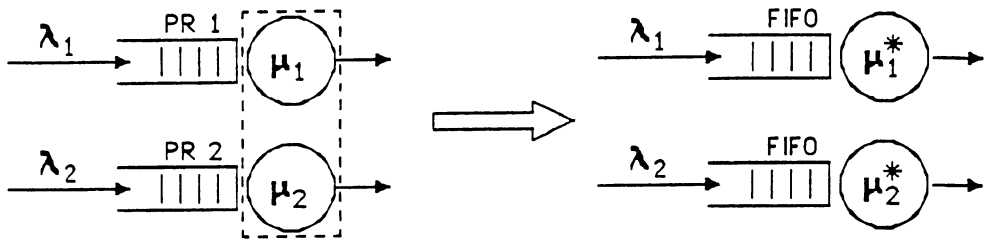


Figure 17: Decomposition of the priority queueing model

can be done approximately following Schmitt [22] as shown in Gihr and Kuehn [5]. In particular, these two queues can be decomposed into the individual queues shown in figure 17 with $\mu_1^* = \mu_1$ and $\mu_2^* = \mu_2 (1 - \alpha)$, where α is the probability that a customer entering queue 2 finds queue 1 busy. α is approximated by the following expression:

$$\alpha = \frac{W_3 - 1}{W_3} \frac{\lambda_1}{\mu_1},$$

where λ_1 is given by the expression:

$$\lambda_1 = \begin{cases} \lambda / (1 - p_{ea}), & \text{for host A;} \\ \lambda / (1 - p_{ei}), & \text{for host B.} \end{cases}$$

We now proceed to apply the approximation procedure described in section 3. In particular, we first analyze the subnetwork controlled by the semaphore queue f_2 (S). This is shown in figure 18, where $\mu^*_2 = \mu^A_2 (1-\alpha)$ and $\mu^*_1 = \mu^B_1$, and tp is the mean service time in the infinite server queues. The mean response time between points A and B in figure 9, $R_2(c)$, $c=1,2,\ldots,W_3$, can be obtained using expresion (3.1). Thus, this semaphore subnetwork can be substituted by a flow-equivalent infinite server queue with a state dependent mean service time equal to $R_2(c)$. Fol-

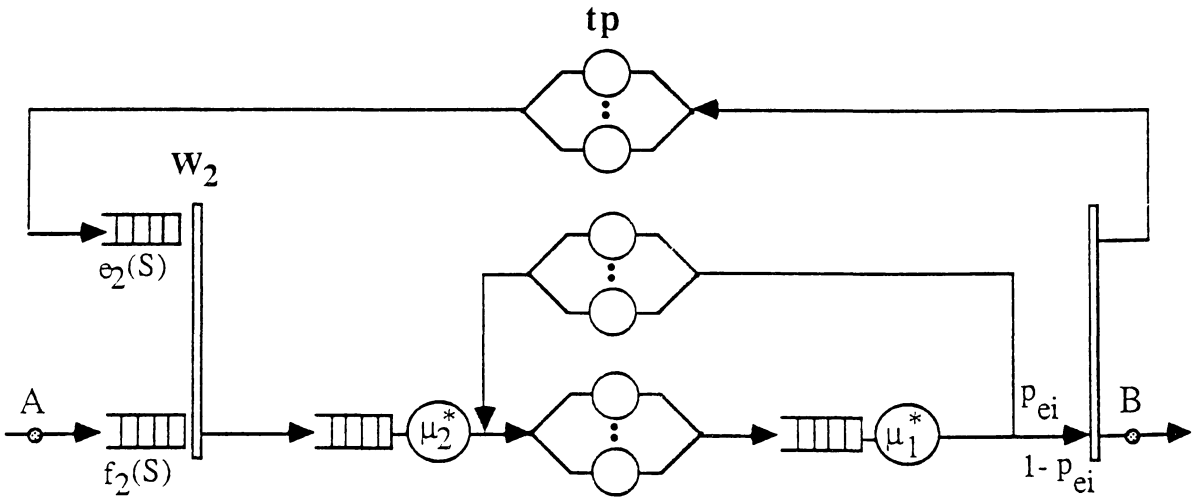


Figure 18: The link level semaphore subnetwork

lowing similar arguments, the subnetwork controlled by the semaphore queue f'_2(S) can be

substituted by a similar flow-equivalent infinite server queue with a state dependent mean service time equal to $R_2(c)$, c = 1,2,..., W_3. The queueing network given in figure 16 can now be reduced to the network shown in figure 19, which can be analyzed using the procedure outlined in section 3. In particular, we analysed this queueing network in order to obtain a) R, the mean response time between points U and V in figure 19; b) T, the mean waiting time in the input queue $f_3(S)$; and c) X, the mean time to traverse queues 1, 2, and 3 (i.e. X=R-T). These quantities were computed as a function of λ, W_2, W_3, the line capacity, and the bit-error rate.

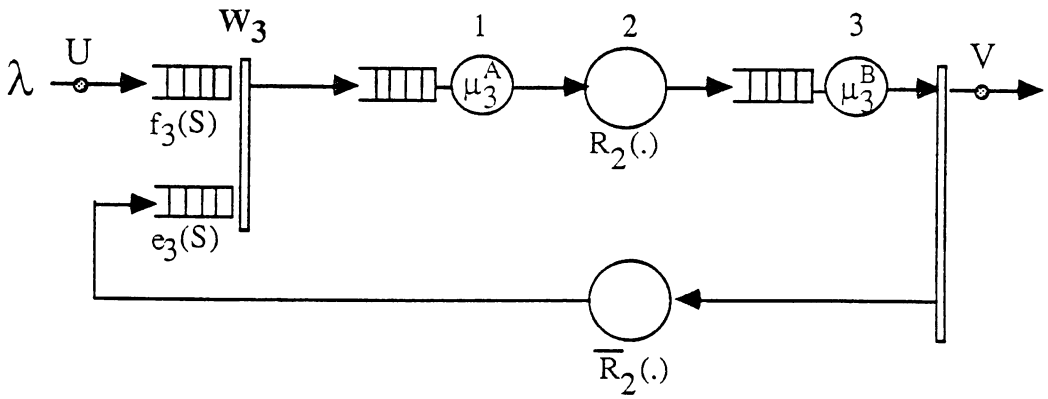


Figure 19: The level 3 semaphore network with aggregation

The following values were assumed for the input parameters: line capacity v=4.8, 19.2, 48 kb/s; information packet size L=1072 bits; RR acknowledgement frame size l=72 bits; RR and REJ frames l'=48 bits; bit error probability from which p_{ei} and p_{ea} are derived are set to 10^{-7}, 10^{-5} and 10^{-4}; service times $1/\mu^A_3$=1ms, $1/\mu^B_3$=1.5ms, $1/\mu^A_1 = 1/\mu^B_1 = (2 + l'/v)$ ms, $1/\mu^A_2$=(1 + L/v)ms, $1/\mu^B_2$=(1 + l/v)ms; line propagation time for a 4.8 kb/s line capacity tp = 11ms, for a 19.2 or 48 kb/s line capacity tp = 4ms.

The results obtained are presented in figures 20 to 24. Figures 20 to 22 show the influence of both layer 2 and layer 3 window sizes on R, T and X, and figures 23 and 24 give R as a function of the bit error rate and the line capacity respectively.

Figure 20 gives R as a function of the window sizes W_3 and W_2 for various values of the arrival rate λ (expressed in packets/s).We note that as λ increases, R increases as well. Also, for fixed value of W_3, R decreases as W_2 increases. Finally, for large values of λ, increasing W_3, while W_2 is kept constant, makes R increase slightly. This is due to the fact that more packets are competing for the same set of limited resources, and as a consequence X increases faster than T decreases. Thus, we have to limit the value of both windows in order to keep R small and to limit the number of resources used as buffers. For the given input parameters, it appears that a good choice of the two window sizes is: W_3=3 and W_2=2.

Figure 21 gives similar results as figure 20, but for X. We observe that for high values of λ, X increases as the two window sizes increase.

Figure 22 gives R, T, and X as a function of the two window sizes W_3, and W_2, where R = T + X. As the two window sizes increase, X increases and T decreases. This is because, more customers are allowed in the semaphore controlled network, which makes the delay inside the network to increase, and the waiting time in the input queue to decrease.

In figure 23, we plot R as a function of the (W_3,W_2) for three different bit error rates. As expected, increasing the bit error rate causes R to increase, seeing that X increases. We note that

when the bit error rate is 10^{-7} (respectively 10^{-4}) no substantial improvement on R is obtained for values of the two window sizes on the right-hand side of (6,3) (respectively (3,3)). Thus, as the bit error rate increases, we have to limit both window sizes.

Finally, figure 24 gives R as a function of λ for three different line capacities. This figure emphasizes the influence of the access line to the network, whose speed can be an order of magnitude lower than the network delay.

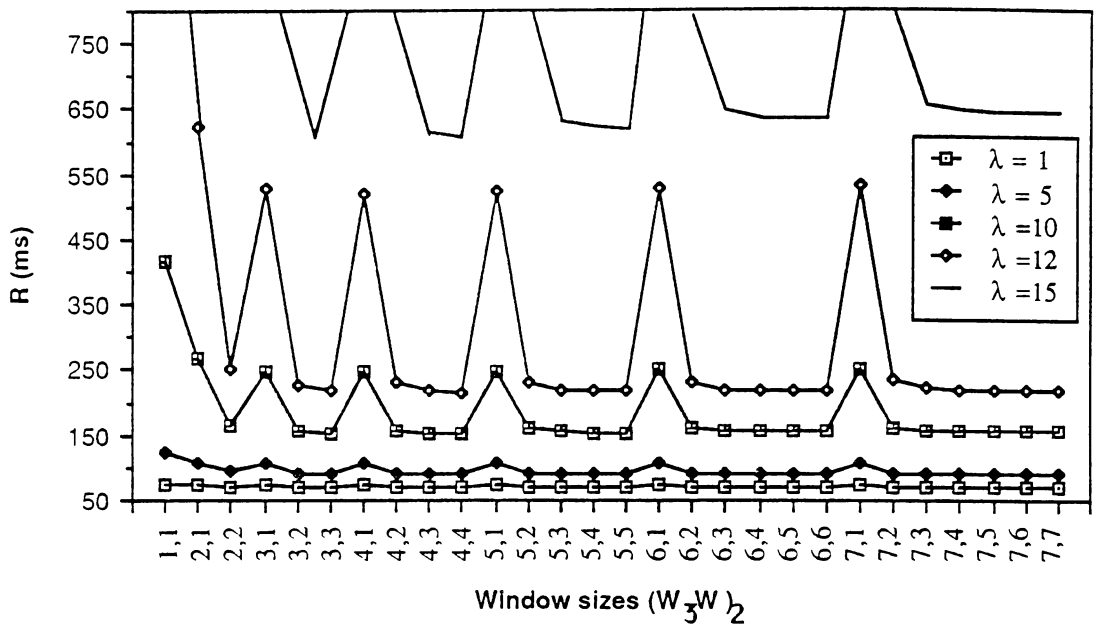


Figure 20: R vs (W_3,W_2) for different values of λ (packets/s); line capacity=19.2 kb/s, bit error rate = 10^{-7}

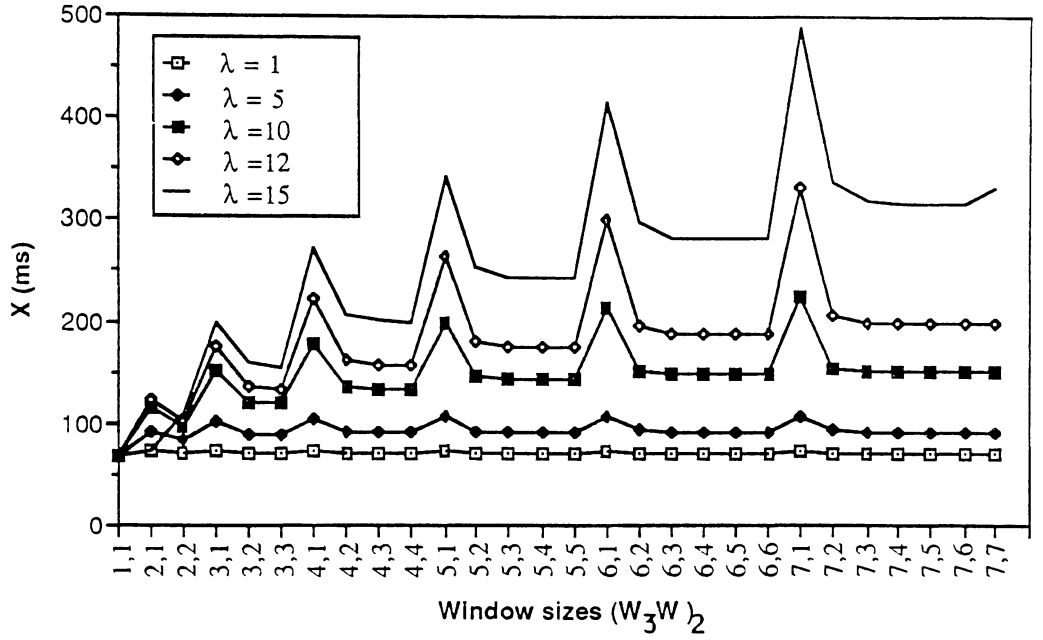


Figure 21: X vs (W_3,W_2) for different values of the arrival rate λ (packets/s); line capacity=19.2 kb/s, bit error rate = 10^{-7}

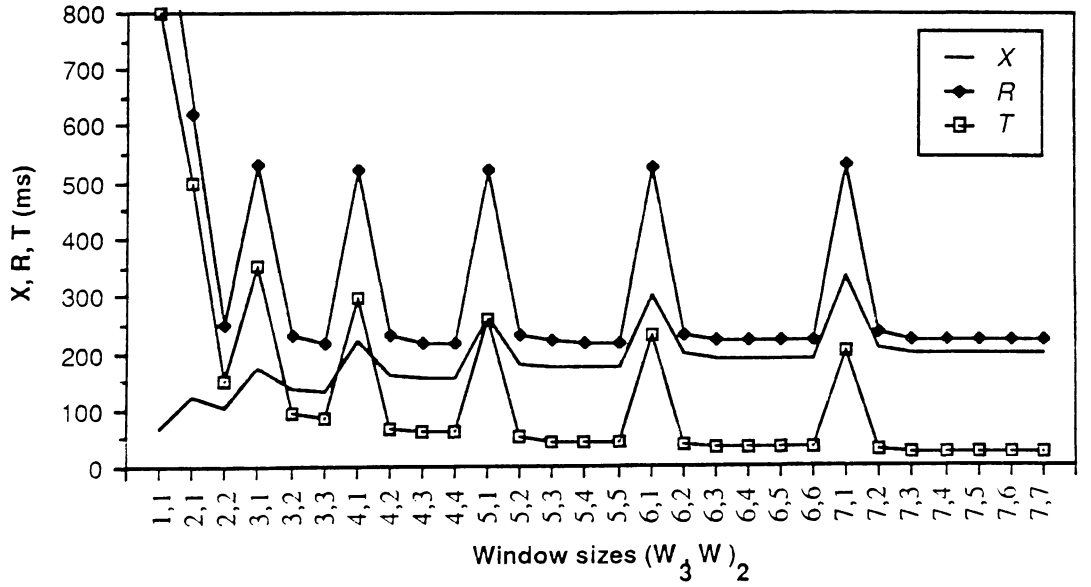


Figure 22: X, R, and T vs (W_3,W_2); arrival rate λ = 12 packets/s, line capacity =19.2 kb/s, bit error rate = 10^{-7}

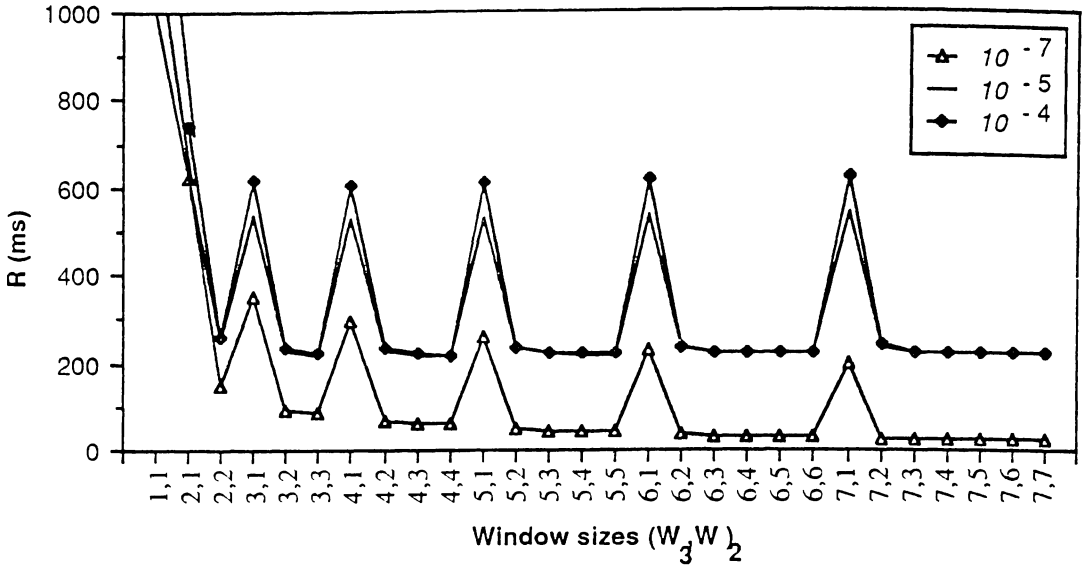


Figure 23: R vs (W_3,W_2)
for different values of the bit error rate: 10^{-7}, 10^{-5}, 10^{-4};
arrival rate λ = 12 packets/s, line capacity =19.2 kb/s

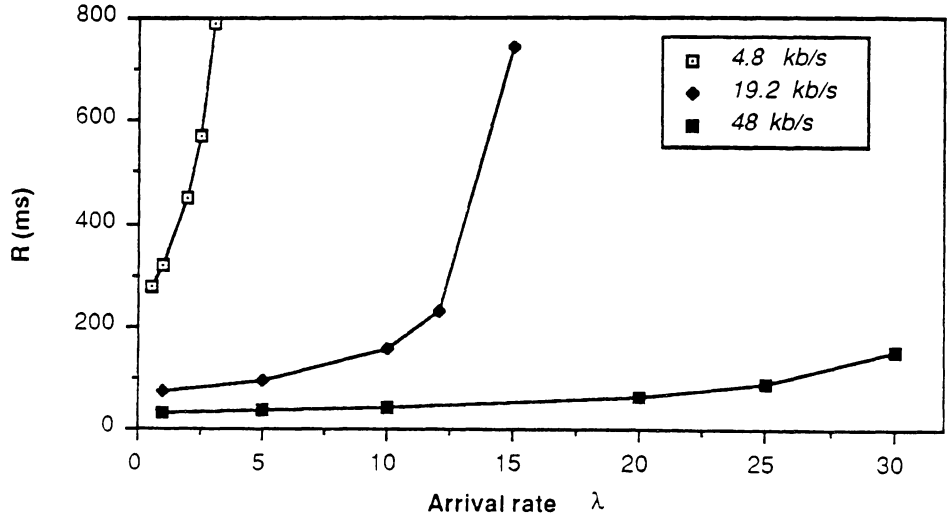


Figure 24: R vs arrival rate λ (packets/s)
for different values of the line capacity : 4.8, 19.2, 48 kb/s;
bit error rate = 10^{-7}, W3=3, W2=2

6. CONCLUSIONS

In this paper, we presented a queueing network model, where the population within each subnetwork is controlled by a semaphore queue. The queueing model was analysed approximately using hierachical decomposition and aggregation. The analysis was restricted to the case of nested subnetworks. Clearly, the same approximation is aplicable to the case where subnetworks (or nests of subnetworks) are arranged in tandem. Each subnetwork was assumed to be of the BCMP type. The queueing network analysed in this paper, was employed to model the ISO X25 flow control mechanism.The influence of the window mechanisms on the mean response time was shown in a number of figures. Finally, the reader is referred to [9] where the end-to-end delay in a catenet environment is analysed.

The challenge for the next few years will be to adapt queueing models to take into account the characteristics of new complex systems such as computer networks, parallel systems, and distributed architectures. Both Petri nets and queueing models can help to define new tools which can be easily used by the practitioner. We are currently involved in the development of other tools (like multiclass semaphores and flags, see [13]) which can be used to model synchronization mechanisms.

REFERENCES

[1] T. Altiok and H.G. Perros, "Approximate analysis of arbitrary configurations of open queueing networks with blocking," *Annals of Oper. Res.*, vol. 9, pp. 481-509, 1987.

[2] S.C. Agrawal and J.P. Buzen, "The aggregate server method for analyzing serialization delays in computer systems," *ACM TOCS*, vol. , pp.116-143, 1983.

[3] P. Caseau and G. Pujolle, "Throughput capacity of a sequence of queues with blocking due to finite waiting room," *IEEE Trans. Soft. Eng.*, vol. SE-5, pp. 631-642, 1979.

[4] D.J. Freund and J.N. Bexfield, "A new aggregation approximation procedure for solving closed queueing networks with simultaneous resource possession," in *Proc. ACM SIGMETRICS Conf.*, Minneapolis, Aug. 1983, pp. 214-224.

[5] O. Gihr and P.J. Kuehn, "Comparison of communication services with connected-oriented and connectionless data transmission", in *Proc. Int. Seminar on Computer Networking and Performance Evaluation*, Tokyo, Japan, Sept. 1985.

[6] K. Goto, Y. Takahashi, and J. Hasegawa, "An aproximate ananlysis of controlled tandem queues," in *Proc. Int. Seminar on Modelling and Performance Evaluation Methodology*, Paris, France, Jan 1983.

[7] P.A. Jacobson and E.D. Lazowska, "Analyzing queueing networks with simultaneous resource possession," *Comm. ACM*, vol. 25, pp.142-151, Feb. 1982.

[8] P.A. Jacobson and E.D. Lazowska "A reduction technique for evaluating queueing networks with serialization delays," *PERFORMANCE '83*, Agrawala and Tripathi (Eds.) , North Holland, 1983, pp. 45-59.

[9] U. Koerner, S. Fdida, H.G. Perros, G. Shapiro, "End to end delays in a catenet environment," in *Proc. Third International conference on Data Communication Systems and their Performance*, Rio de Janeiro, Brazil, June 1987, pp 453-464.

[10] S.S. Lam, "Queueing networks with population size constraint," *IBM J. Res. Develp.*, vol. , pp 370-378, 197.

[11] S.S. Lavenberg, "Stability and maximum departure rate of certain open queueing networks having capacity constraints," *RAIRO Informatique/Computer Science*, vol. 12, pp. 353-370, 1978.

[12] D. Mailles, "*Files d'attende descriptives pour la modelisation de la synchronization dans les system informatiques*", These d'Etat, Univ. Paris 6, Sept. 1987.

[13] D. Mailles, S. Fdida, "*Queueing Systems with Flag Mechanisms*", Proceedings of the International Workshop on Modelling Techniques and Performance Evaluation, Paris 87, North Holland, pp. 167-190.

[14] M. Neuts, "Matrix-geometric solutions in stochastic models - an algorithmic approach", The John Hopkins Univeristy Press, Baltimore 1981.

[15] M.C. Pennotti and M. Schwartz, "Congestion control in store and forward tandem links," *IEEE Trans. Comm.*, vol. COM-23, pp. 1434-1443, 1975.

[16] H.G. Perros, "A symmetrical exponential open queue network with blocking and feedback," *IEEE Trans. Soft. Eng.*, vol. SE-7, pp. 395-402, 1981.

[17] H.G. Perros, "A two-node queuing network with a maximum number of allowable jobs," *PERFORMANCE '83*, Agrawala and Tripathi (Editors), North-Holland, 1983, pp. 33-44.

[18] D. Potier, "New user's introduction to QNAP2", INRIA tech. rep. 40, Feb. 1984.

[19] M. Reiser, "A queueing network analysis of computer communication networks with window flow control," *IEEE Trans. Comm.*, vol. COM-27, pp. 1199-1209, 1979.

[20] M. Reiser, "Admission delays on virtual routes with window flow control," in *Proc. Performance of data Communication Systems and their applications*, Pujolle (Ed.), North-Holland, 1981, pp. 67-76.

[21] M. Reiser, "Performance evaluation of data communication systems," *Proc. IEEE*, vol.70, pp 171-196, 1982.

[22] W. Schmitt, "On a decomposition of Markovian priority queues and their application to the analysis of closed queueing networks," *PERFORMANCE '84*, Gelenbe (Ed.), North Holland, 1984, pp. 393-407.

[23] M. Schwartz, "Performance analysis of the SNA virtual route pacing control," I*EEE Trans. Comm.*, vol. COM-30, pp. 172-184, 1982.

[24] A. Thomasian, "Queueing network models to estimate serialization delays in computer systems," *PERFORMANCE '83*, Agrawala and Tripathi (Eds.) , North Holland, pp.45-59, 1983.

[25] A. Thomasian and P. Bay, " Analysis of queueing network models with population size constraints and delayed blocked customers," in Proc. *ACM SIGMETRICS Conf.*, Cambridge, 1984, pp. 202-216.

[26] G. Varghese, W. Chou, and A.A. Nilsson, "Queueing delays on virtual circuits using a sliding window flow control scheme," *Proc. ACM SIGMETRICS Conf.*, Minneapolis, 1983, pp. 275-281.