Semaphore queues:  
Modelling window flow control mechanisms

Serge Fdida  
Andrzej Wilk  
and  
Harry Perros

Center for Communications and Signal Processing  
Department of Electrical Computer Engineering  
North Carolina State University

June 1987

CCSP-TR-87/5
Semaphore queues:
Modelling window flow control mechanisms

Serge FDIDA*, Andrzej WILK+, Harry PERROS#

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

+ Polish Academy of Sciences, Dept. of Complex Control Systems
ul. Bałtycka 5, 44-100 GLIWICE, POLAND

# Computer Science Department, North Carolina State University,
Raleigh, NC 27695-8206, USA

ABSTRACT

We present an approximation technique for solving open queueing networks consisting of
several subnetworks. Each subnetwork models a set of shared resources managed by a semaphore.
The queueing networks analyzed have features that are commonly used in Petri Nets, such as, fork
and join operations and semaphore queues. This type of queueing network arises naturally when
modelling multiple window flow control mechanisms. The approximation procedure is easy to
implement and it can be included in a software package. Using this procedure, we present a case
study involving the modelling and analysis of the ISO X25 flow control mechanism.

1 - INTRODUCTION

In this paper, we describe and analyze the sharing of a resource depicted by a queueing network.
The resource sharing is managed by a semaphore operation. Semaphore operations have traditionally
been described using Petri nets. The Petri Net approach, however, relies on the numerical solution of
an underlying rate matrix. The size of this matrix grows rapidly as the problem under study becomes
more complex. In view of this, we model such systems as queueing networks, which we then
analyze approximately using a hierarchical decomposition algorithm.

The sharing of several resources is a common phenomenon in computer systems. This problem
has been studied by several authors. In particular, Jacobson and Lazowska [Jaco 82] developed an
iterative approximation method for analyzing a system involving the sharing of a set of resources.
This set of resources is assumed to be partitioned into two disjointed groups, namely, the primary
and the secondary resource group. A customer, after it accesses a primary resource, is allowed to
access secondary resources in sequence. The primary resource elements are kept by the customer, as
long as the customer utilizes elements of the secondary resource group.

Freund and Bexfield [Freu 83] developed an approximation method based on decomposition, to
analyze the sharing of resources modelled as a closed queueing network. Sauer [Sauer 81a] extended
an approximation method to the case where heterogeneous customers share a resource. The solution
is obtained through the transformation of a network Q1 (where a subset of resources are held simultaneously) into a network Q2 to fulfill a flow equivalence. The system Q2 possesses an equivalent queue for the set of all stations of Q1 where the sharing of resources are available. The method was applied to two classes of customer. The error-level is acceptable in some applications.

Approximation methods have also been developed to take into account the effect of serialization delays. Serialization arises often, when using critical sections. Smith and Browne [Smit 80] proposed the use of a supplementary service station, which is added to the model and which represents the delay incurred while waiting for the critical section. The service time of this queue is computed iteratively so that the mean number of customers in this queue equals 1. This method is only correct for heavy traffic. Thomasian [Thom 83] developed two different approximation methods, one based on an iterative scheme and the other on the concept of decomposition. The iterative technique is similar to the method proposed by Smith and Browne. Semaphore queues are used in the iterative method in order to model serialization delays. This approximation method has a satisfactory error-level except when the utilization is high. The decomposition method gives better results but requires more CPU time.

Jacobson and Lazowska [Jaco 83] proposed a technique whereby they reduce a queueing network with simultaneous resource possession to a queueing network consisting of a subsystem with a population constrained. The network is then replaced by an equivalent network with a limited capacity subnetwork. The idea is that each customer visiting the shared resource is marked. The constraint on the resource is expressed as a limit on the number of customers allowed to enter the class associated with the resource.

Agrawal and Buzen [Agra 83] reported on an iterative scheme in which each phase of a critical section is modeled by an equivalent server. The service time of each server corresponds to the time spent in each phase of the critical section. The service times of customers outside the critical section are increased in order to take into account the contention due to the critical section.

Some approximation methods have been developed to extend previous work to multiple chain population constrained queueing networks. The direct application of exact methods involving the reduction of each subnetwork to a single flow equivalent composite queue, is not as straightforward. This is because a multi-dimensional markov chain needs to be constructed and solved. The computational complexity (time and space requirements) grows rapidly with the number of chains in the network making this approach infeasible except for small models (see [Saue 81b] for more details). Thus, approximate methods have been proposed. An overview of these methods can be found in [Thom 84] and [Krze 85]. In this last paper, each subnetwork is reduced to a single approximately flow equivalent composite center. This is done assuming that the effect of all the other chains on a given chain is only represented by their average customer populations. The accuracy of this approximation is demonstrated by comparing it against previously published concurrency constraint methods.

In section 2, we describe the queueing network under study involving one semaphore queue. An approximation solution of this system is given section 2.1. In section 3, we give an approximation algorithm for analyzing a queueing network with multiple semaphore queues. A case study involving the modeling and analysis of the ISO X25 flow control mechanism is given section 4. Finally, the conclusions are given in section 5.

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 a queue f(S) and a counter e(S). A customer arriving at the semaphore queue requests a pass. If there is a pass available, the customer is allowed to leave the queue. Otherwise, the customer is blocked and it is forced to wait in the queue. The total number of
available passes is represented by the counter e(S). If e(S) is positive, then an arriving customer will receive a pass immediately. Otherwise, the customer will wait in the semaphore queue until a pass becomes available. In view of this, if e(S) is positive, then there are no customers in the semaphore queue. On the other hand, if there are customers in the semaphore queue, then e(S) is zero.

A customer having received a pass, leaves the semaphore queue and enters a network (network 1). When it finally departs from network 1, the pass is returned back to the counter via another network (network 2) as shown in figure 1.

![Diagram](image.png)

**Figure 1.- description of the model under study**

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

![Join Symbol](image.png)

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

![Fork Symbol](image.png)

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 more descriptive convenience.
The total number of passes available is fixed to C. That is network 1 can be used at the most by C customers. We also note that the passes are returned back to the counter via a separate network (network 2). The total number of customers in network 1 and returning passes in network 2 is at most 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 \( \lambda \). We note that this model is similar to the one studied by [Gihr 85].

### 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. 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 named \( \gamma(n) \).

The semaphore queue

![Figure 2](image)

This queueing system depicts the join operation described in section 2. As discussed earlier, the arrival rate to queue f(S) is assumed to be poisson distributed, and there are C passes. We also assume that the inter-arrival times at queue e(S) are exponentially distributed with a rate \( \gamma(C-k) \) where k is the number of customers in queue e(S). It's easy to notice that k customers inside queue e(S) leads to (C-k) customers in network Q. 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 \( j \) in queue e(S). The rate diagram associated with this system is shown in figure 3.

![Figure 3](image)

We note that this system is identical to an M/M/1 queue, with an arrival rate \( \lambda \), and a dependent service rate \( \gamma(n) \) if \( n \leq C \), and \( \gamma(C) \) if \( n > C \). State \( i \) and \( j \) of the semaphore queue are linked to \( n \) with the expressions: \( i = \max(0, n-C) \), \( j = \max(0, C-n) \), where \( n \) is the number of customers in this M/M/1 queue.

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

\[
p(i,0) = \rho^i p(0,0),
\]

\[
p(0,j) = \frac{\prod(j)}{\lambda^j} p(0,0);
\]

(2.1)
where $\rho = \lambda / \gamma(C)$ and

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

The probability $p(0,0)$ is chosen to make the equilibrium state probabilities sum to 1:

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

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 $e(S)$):

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

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

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

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

Now, the expression (2.1) to (2.4) were obtained assuming that $\gamma(C-k)$ is known. This can be approximately obtained as follows. We can study the queueing networks 1 and 2 as a closed queueing network 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 the closed system with $n$ customers, where $n=1,2,\ldots,C$. This is then set equal to the arrival rate $\gamma(n)$ of customers at the counter queue $e(S)$.

We note that in the above formulation, the acknowledgments are sent back via a separate
network, network 2. This formulation can be easily changed so that to allow the acknowledgements to travel back over the network used by the packets, network 1. To do this, it suffices to declare two classes of customer, namely class 1 and class 2 representing packets and acknowledgements respectively. These two classes of customer 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. In general, BCMP type of queueing networks may be inadequate when modelling communication systems. In such cases, one can use more general queueing networks which can depict general service times, fragmentation/reassembly of packets, and priorities more realistically. Such queueing networks can then be analyzed either approximately or numerically in order to obtain the quantity $\gamma(n)$, $n=1,2,...,C$. (See for instance Koerner, Fdida, Perros, and Shapiro [Koer 87]).

**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 [Lave 75]) in view of the fact that $C$ customers maximum can be in the network.

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 in various ways to make up larger more complex systems. In this section, we give a simple approximation algorithm for computing the solution of such multiple semaphore networks. This algorithm can be used for any configuration involving BCMP queueing networks and semaphore queues. For presentation purposes, however we restrict ourselves to queueing systems which consist of nested semaphore controlled queueing networks, as 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 $n$th 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.](image)
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 $R'_{n-1}(l)$, the mean time to traverse $Q_{1,n-1}$ as a function of the number of customers $l$ in $Q_{n-1}$, where $l=1,2,...,C_{n-1}$. 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 $R_{n-1}(c)$ between points A and B is approximately given by

$$R_{n-1}(c) = \begin{cases} 
R'_{n-1}(c), & c \leq C_{n-1} \\
R''_{n-1}(C_{n-1}) + (c-C_{n-1})\gamma_{n-1}(C_{n-1}), & c > C_{n-1},
\end{cases} \quad (3.1)$$

where $c$ is the number of customers (i.e. packets) in the semaphore controlled queueing network, i.e. in queue $f_{n-1}(S)$ and in $Q_{1,n-1}$. The above expression can be easily derived. For, if $c \leq C_{n-1}$, then all the customers are in the process of traversing $Q_{1,n-1}$. Thus, they are all delayed by $R'_{n-1}(c)$. If $c > C_{n-1}$, then only $C_{n-1}$ customers are in the process of traversing $Q_{1,n-1}$, 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 (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 $c \geq C_{n-1}$.

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 $n$th 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 $n$th level semaphore network.

Now, let us assume that the $n$th 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

$$L_{f_n(S)} = \sum_{i=1}^{\infty} i p(i)$$  \hspace{1cm} \text{(3.2)}

The mean number of customers in the 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 $Q_n$. Let $p_1(m)$ be the probability that there are $m$ customers in $Q_{1,n}$. We have $\forall \ m > h, \ p_1(m|h)=0$. For $[0 < m \leq h]$, we obtain

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

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

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

$$= \sum_{m=1}^{c_{a}} \left( \sum_{h=m}^{c_{a}} p_1(m|h) q(C_a-h) \right)$$

$$= \sum_{h=1}^{c_{a}} q(C_a-h) \sum_{m=1}^{c_{a}} m p_1(m|h)$$

The quantity $\sum m p_1(m|h)$, for $m=1$ to $C_n$ 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,
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

$$
\sum_{m=1}^{c_n} m p_1(m|\text{h}) = \gamma_n(h) R'_{n}(h)
$$

and hence

$$
L_{Q_{1,n}} = \sum_{m=1}^{c_n} h \frac{R'_{n}(h)}{R''_{n}(h)} q(C_{n} \cdot \text{h})
$$

(3.3)

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

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

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

We finally note that in the above analysis we have assumed that queueing networks $Q_{1,n-1}$, $Q_{2,n-1}$, and $SN_{i,n}$, $i = 1,2,3,4$, have a product-form solution. As it was mentioned in section 2.1, this assumption is not necessary as long as the necessary performance measures can be obtained by analyzing these queueing networks approximately and/or numerically.

4 - CASE STUDY: A MULTIPLE LEVEL FLOW CONTROL

In this section, we employ the queueing network models developed above in order to model the flow control of an X25 ISO protocol.

The past few years have seen the development of important efforts 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 such architectures makes the study and the performance evaluation of those systems quite difficult. We can use the semaphore queueing model analyzed above, to study such a system. In particular, it can be used to model the sharing of resources by the outstanding frames. This semaphore queueing model appears to be easy to use and well suited for studying communication protocols.

We study the communication system defined by the first three layers of the ISO model (see figure 6).

![Figure 6.- the X25 flow control mechanism](image)

- The physical layer (layer 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:
- Information frame (I), for transmitting information,
- Receive ready (RR) frame, for the positive acknowledgment of information frame, and
- 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 7. 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 $\lambda$. 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 $f_2(S)$, if $e_2(S)$ is empty. When a token becomes available, the packet enters the layer 2 queue where it receives a service at the rate $\mu_A^2$. Upon completion of this service, the packet joins an infinite server queue reflecting the transmission and propagation delay. Following this 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 packet
may be rejected as being erroneous with probability \( p_{ei} \). In this case, a REJ frame is sent back and the packet is retransmitted. With probability \( (1-p_{ei}) \) the packet is found error-free and it is allowed to join the layer 3 queue where it is served at the rate \( \mu_3 \). At the same time, an RR frame is transmitted back. This is represented by a token which after transmission and propagation delay will join queue \( e_2(S) \). Finally, the packet upon completion of its service at the layer 3 queue is delivered to host B. At the same time, a token (representing an RR packet) is placed in queue \( f'_2(S) \). The token is returned back to queue \( e_2(S) \) through a path which is similar to the forward path followed by the packet.

![Figure 7: The queueing model of X25 flow control mechanism]

We assume that each layer is managed by a different processor. Thus, in host A (or B) the layer 3 queue is served by a different processor than the layer 2 queues. 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). The service times of the customers in each of the queues in figure 7 are assumed to be exponentially distributed (this assumption is not necessary for the infinite server queues). In order to apply the approximation algorithm described in section 3, we need to decouple these two queues. This can be done.

![Figure 8: Decomposition of the priority queueing model]
approximately following [Schm 84] as shown in [Gihr 85]. In particular, these two queues are decomposed into the individual queues shown in figure 8. We have:

\[ \mu_1^* = \mu_1 \quad \text{and} \quad \mu_2^* = \mu_2 (1 - \alpha), \]

where \( \alpha \) is the probability that a customer entering queue 2 finds queue 1 busy. \( \alpha \) is approximated by the following expression:

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

\( \lambda_1 \) is given by the expression:

\[ \lambda_1 = \begin{cases} \frac{\lambda}{1 - P_{ea}}, & \text{for host A}; \\ \frac{\lambda}{1 - P_{ei}}, & \text{for host B}. \end{cases} \]

We now proceed to apply the approximation procedure described in section 3.

Level 2 aggregation

We first analyze the subnetwork controlled by the semaphore queue \( f_2(S) \). This is shown in figure 9, 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, \( R_2(c), c=1,2,...,W_2 \), can be obtained using expression (3.1). Thus, this semaphore subnetwork can be substituted by a flow-equivalent infinite server queue with a state dependant mean service time equal to \( R_2(c) \).

![Figure 9: the link level semaphore subnetwork](image_url)

Following 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
equal to $R_2(c)$, $c = 1, 2, \ldots, W_2$.

**Level 3 aggregation**

The queueing network given in figure 7 can now be reduced to the network shown in figure 10, which can be analyzed using the procedure outlined in section 3. For instance, the mean response time $R$ between points U and V in figure 10, can be obtained using expression (3.4). This is the mean time it takes for a packet to be transmitted from host A to host B. This quantity was computed as a function of $\lambda$, $W_2$, $W_3$, and line capacity.

![Figure 10: the level 3 semaphore network with aggregation](image)

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 packet size $l=72$ bits; RR and REJ frames $l'=48$ bits; bit error probability $p_{ei} = p_{e3} = 10^{-7}$; service times $1/\mu_A^3=1$ms, $1/\mu_B^1=1.5$ms $1/\mu_A^1 = 1/\mu_B^2 = 2 + 1/v$ ms, $1/\mu_A^2=1 + L/v$ ms, $1/\mu_B^2=1 + l/v$ ms; line propagation times for a 4.8 kb/s capacity medium $t_p = 11$ms, for a 19.2 or 48 kb/s capacity medium $t_p = 4$ms.

The results obtained are presented in figures 11 to 15. We note that $R$ decreases as $W_2$ and $W_3$ increase. $R$ reaches a minimum when $W_2, W_3 > 5$ (see figures 14 and 15). In fact, for this example the optimum window sizes are $W_2$ and $W_3 = 3$. 

13
Figure 11.- Response time $R$ vs external arrival rate.
line capacity=19.2 kb/s, $W_2=1$

Figure 12.- Mean response time $R$, mean waiting time $S$ in queue $f_3(S)$, mean time $T$ for a packet to traverse queues 1, 2, 3 in figure 10 vs arrival rate $\lambda$.
line capacity =19.2 kb/s, $W_2=1, W_3=3$
Figure 13.- response time $R$ as a function of $\lambda$ for different values of line capacity, $W_2=2, W_3=3$

Figure 14.- response time $R$ vs level 3 window size $W_3$, line capacity=19.2 kb/s, $W_2=1$
The approximation model developed for this case study was validated using a detailed simulation model of the queueing system shown in figure 7. The approximate results appear to be quite accurate. Table 1 gives a comparison between the approximate and the simulation model for the following values of the input parameters: line capacity $v = 19.2$ kb/s, layer 3 window size $W_3 = 3$, layer 2 window size $W_2 = 2$, bit error probability $p_{ei} = p_{ea} = 10^{-7}$. The condition for stability was approximately estimated to be $\lambda < 16.272$ (1/sec). In view of this, $\lambda$ was varied from 0.1 to 15 (sec).
Table 1.- comparaison of the results obtained through simulation runs (simul.) and the approximate method (appro) vs arrival rate \( \lambda \).

<table>
<thead>
<tr>
<th>( \lambda ) [1/s]</th>
<th>method used</th>
<th>mean response time R [ms]</th>
<th>mean waiting time in ( f_3(S) ) [ms]</th>
<th>mean number of customers in ( f_3(S) )</th>
<th>mean number of tokens in ( e_3(S) )</th>
<th>mean number of customers not in ( f_3(S) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>simul</td>
<td>61.39</td>
<td>0.0 ( \times 10^{-7} )</td>
<td>0.0 ( \times 10^{-9} )</td>
<td>2.993 ( \times 10^{-2} )</td>
<td>0.612 ( \times 10^{-2} )</td>
</tr>
<tr>
<td></td>
<td>appro</td>
<td>68.18</td>
<td>1.7 ( \times 10^{-7} )</td>
<td>1.7 ( \times 10^{-9} )</td>
<td>2.992 ( \times 10^{-2} )</td>
<td>0.682 ( \times 10^{-2} )</td>
</tr>
<tr>
<td>1</td>
<td>simul</td>
<td>72.91</td>
<td>0.377 ( \times 10^{-1} )</td>
<td>0.364 ( \times 10^{-4} )</td>
<td>2.916 ( \times 10^{-2} )</td>
<td>0.703 ( \times 10^{-1} )</td>
</tr>
<tr>
<td></td>
<td>appro</td>
<td>71.49</td>
<td>0.181 ( \times 10^{-1} )</td>
<td>0.181 ( \times 10^{-4} )</td>
<td>2.914 ( \times 10^{-2} )</td>
<td>0.703 ( \times 10^{-1} )</td>
</tr>
<tr>
<td>10</td>
<td>simul</td>
<td>163.6</td>
<td>39.36</td>
<td>0.387</td>
<td>1.625</td>
<td>1.220</td>
</tr>
<tr>
<td></td>
<td>appro</td>
<td>159.2</td>
<td>38.50</td>
<td>0.385</td>
<td>1.642</td>
<td>1.207</td>
</tr>
<tr>
<td>12</td>
<td>simul</td>
<td>237.1</td>
<td>97.45</td>
<td>1.166</td>
<td>1.142</td>
<td>1.669</td>
</tr>
<tr>
<td></td>
<td>appro</td>
<td>229.9</td>
<td>94.94</td>
<td>1.139</td>
<td>1.197</td>
<td>1.619</td>
</tr>
<tr>
<td>15</td>
<td>simul</td>
<td>913.0</td>
<td>750.8</td>
<td>11.17</td>
<td>0.344</td>
<td>2.413</td>
</tr>
<tr>
<td></td>
<td>appro</td>
<td>755.4</td>
<td>597.3</td>
<td>8.959</td>
<td>0.393</td>
<td>2.373</td>
</tr>
</tbody>
</table>

6 - CONCLUSION

We presented an approximation technique for solving queueing networks containing several subnetworks where each subnetwork is modelling a set of shared resources. The management of each subnetwork is performed using a semaphore. A new representation (using both Petri nets and queueing systems concepts) is introduced to describe fork, join and semaphore tools in a queueing network. The approximation method is very easy to implement.

The challenge for the next few years is to adapt queueing models to take into account the characteristics of new complex systems such as computer networks, parallel systems or distributed architectures. Both Petri nets and queueing models can help to define new tools which can be easily used by the practitioners. We are currently involved in the development of other tools (like multiclass semaphores and flags [Mai 87]) which can be used to model synchronization mechanisms.

REFERENCES

[Agra 83]

[Freu 83]

[Gihr 85]
[Jaco 82]  

[Jaco 83]  

[Koer 87]  

[Krze 85]  

[Lave 75]  

[Mail 86]  

[Saue 81a]  

[Saue 81b]  

[Schm 84]  

[Smit 80]  

[Thom 83]  

[Thom 84]  
A. THOMASIAN "Analysis of queueing network models with population Size constraints and