Resource sharing and blocking; applications to connection-oriented protocols

G. Pujolle
Laboratoire MASI
Université P. et M. Curie
45 avenue des Etats-Unis
78000 Versailles
France

Abstract

In this paper, we present a general approach to obtain the exact stability condition of systems with resource sharing. Two kinds of allocation disciplines are studied:

1.- the allocation may be performed in parallel: as soon as a resource is free, if a customer is waiting, the allocation begins.

2.- the allocation is performed in series: the allocation queue is constituted with only one server, and the customers are treated in series.

We apply our results to different communication systems. In particular we develop a general model for connection-oriented transport model in computer networks.

1- Introduction

In computer science, resource sharing is a very common situation and it is necessary to model systems where customers are limited in numbers. This capacity constraint generally involves a severe simplification of the model in order to obtain some performance results.

In this paper, we present a general approach to obtain the exact stability condition of systems with resource sharing. Two kinds of allocation disciplines are studied:

1.- the allocation may be performed in parallel: as soon as a resource is free, if a customer is waiting, the allocation begins.

2.- the allocation may be performed in series: the allocation queue is constituted with only one server, and the customers are treated only in series.

We apply our results to different communication systems. In particular we develop a general model for connection-oriented transport model in computer networks. In such networks the flows of PDUs (Protocol Data Unit) are controlled through window mechanisms. A PDU may only be transmitted if the maximum allowed number of outstanding PDU's for which a positive acknowledgement has still to be notified to the transmitter (the window size) is not reached. When this limit is reached and still more

PDU's have to be forwarded, the sender stops transmitting and waits for acknowledgement. In this model, the resources to be shared are line capacities and buffers of the packet switching nodes. This is taken into account through the window flow control sizes.

There is a large number of papers concerning resource sharing and blocking. Refer for example to [1] for a survey of this literature. We are interested here in the maximum throughput of a general network. The evaluation method which is the subject of this paper has quite general assumptions. Actually, it applies accurately when a solution is known for the equivalent closed network.

2- The basic models

2.1- The parallel allocation resource sharing queue

The first model under study is shown in Fig. 1. The first queue provides the allocation of the common tokens (or credits or resources). This queue is composed of c parallel independent servers. Each server corresponds to a resource available in the network. These resources are shared between at most c customers. A customer enters the system if at least one token is available. If no server is free, the customer has to wait for an idle server. When a customer enters an allocation server, he spends here a time corresponding to the allocation of the resources necessary to go accross the network. Let μ^{-1} be the mean allocation service time. When a customer leaves the allocation server, he enters the network; but, the server of the allocation queue remains blocked until the customer leaves the network and releases the token (or the acknowledgment of the departure comes back to the allocation queue).

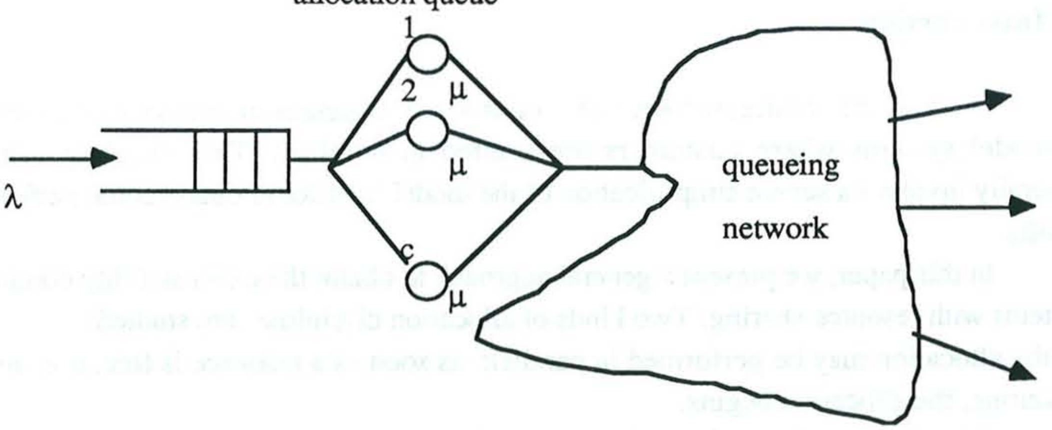


Figure 1: the model under study

The queueing network may be quite general with the assumption that we can obtain the steady state probabilities of the different queues. Namely, the network may be a Jackson network [9], a BCMP network [10], a Kelly network [11] or an "insensitive" network [12]. It may also be a more general queueing network where, when closed, the conditions of the

MVA technique or other iterative methods hold. Moreover, numerical techniques and approximations may also be applied. The allocation queue may have a general renewal service time process.

2.2- The series allocation resource sharing queue

In this case, the allocation queue has only one server and the allocation of a resource is provided in series. The total number of resources is still c. The resources are represented by tokens and a customer must catch a token to enter the network. The token becomes free again when the customer leaves the network. The token comes back to the idle token queue immediately, or after a time spent to go across back the network as illustrated in Fig. 2.

This figure uses as the first station a semaphore queue representation. A customer can enter the network only if his service time in the allocation queue is completed and a token is available (simultaneously).

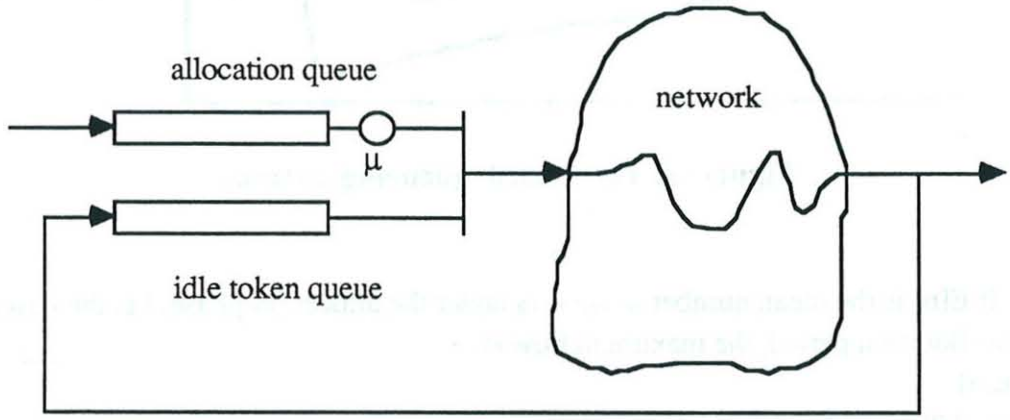


Figure 2: the series allocation resource sharing queue

Two different schemes are possible:

1- when the idle token queue is empty and a customer enters the service μ, he is served. At the end of the service time either a token is available and the customer enters the network or no token is available and the customer is blocked in the server. As soon as a token arrives in the idle token queue, the customer enters the network.

2- when the idle token queue is empty and a customer enters the service μ, he is blocked and he waits for a token before beginning his service time.

3- The general solution

We are interested in the maximum throughput of the system. This is equivalent to compute the stability condition of the system. We use all along this paper the result given by Lavenberg [13] : the maximum throughput Λ is obtained saturating the first queue and looking at the outputs of the system under some assumptions to be satisfied by the network (mainly the service process must be conservative). If λ is the arrival rate (for any renewal arrival process) in the allocation queue, the system will be stable if $\lambda < \Lambda$.

If we assume a saturated first queue, the total number of customers in the system is c. When a customer leaves the network, another one enters it immediately. Therefore, the maximum throughput is obtained studying the closed queueing network shown in Fig. 3, where c customers are being served or are consuming allocation time in the relevant queue.

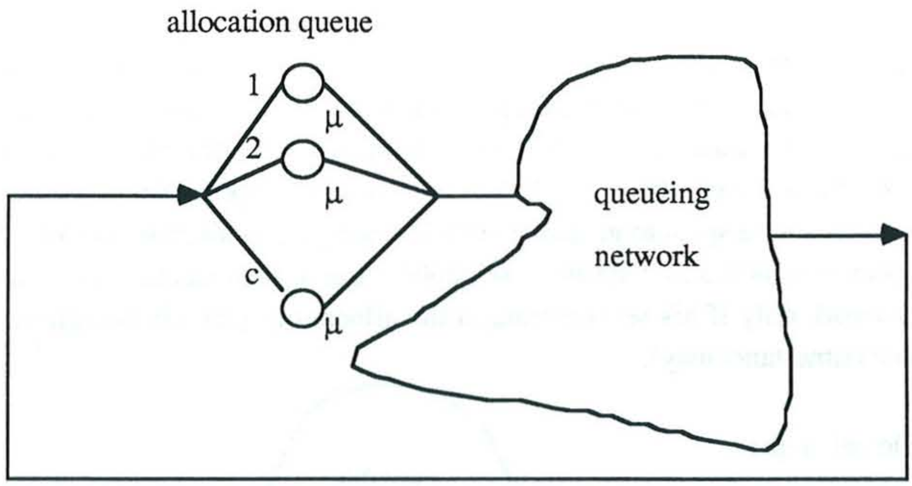


Figure 3: the closed queueing system

If E[n] is the mean number of servers under the allocation phase, i.e. the customer's resource allocation period, the maximum flow is:

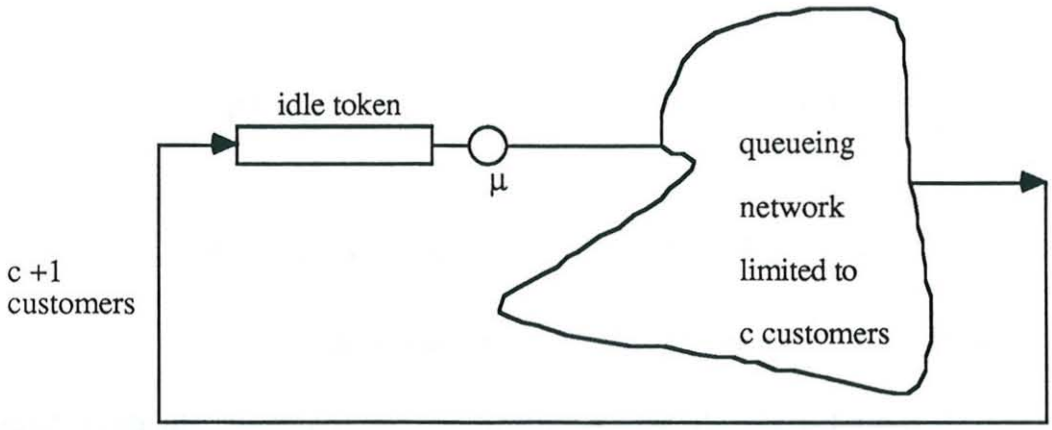
$\Lambda = \mu\, E[n]$

and the stability condition is:

$\lambda < \Lambda$.

The allocation queue in the closed queueing system may be modelled as an infinite server station. It turns out that the service distribution of this station may be general if classical BCMP, Kelly or "insensitive" networks are provided as the queueing network.

In both series allocation cases the maximum throughput is obtained assuming at least one customer is waiting in the allocation queue, i.e. assuming a saturation condition. It turns out that for case 1, we have to consider the closed model shown in Fig. 4 with c+1 customers. The queueing network is limited to c customers that implies blocking phenomenon on the idle token queue with the service μ. This case is generally more complicated to solve.



case 1: c+1 customers; case 2: c customers

Figure 4: the equivalent closed model

For case 2, we have the same closed model but with only c customers in the network. No blocking occurs.

If Λ_i i =1,2 are the throughputs of the previous closed models for cases 1 and 2 respectively, the stability conditions are: $\lambda_1 < \Lambda_1$ and $\lambda_2 < \Lambda_2$

4- Example

Let us consider a simple example. We consider that the network is a single FIFO queue with exponentially distributed service times with parameter γ. This ensures the BCMP theorem is satisfied.

We have to compute E[n] the mean number of customers in the allocation server considering the associated closed model with c customers. It follows

$\Lambda = \mu\, E[n] = \mu \sum\; k_1 G(1/k_1!)(1/\mu)^{k_1}(1/\gamma)^{k_2}$

where G is the normalizing constant of the closed BCMP network.

Another possibility to obtain Λ might be to compute the mean response time E[t] of the closed network; i.e. the time between two consecutive passages of the same customer at the same place. We shall obtain: $\Lambda = c / E[t]$. A last possibility might be to compute the probability that the second queue is not empty $1\text{-}P_2(0)$; in this case: $\Lambda = \gamma\,(1 - P_2(0))$.

Now, assume a series allopcation scheme and a network composed of a single queue with exponentially distributed service times with parameter γ.

Case 1

It is exactly equivalent to assume that the second queue has a finite capacity of c+1 and no blocking can occur. We obtain easily:

Λ\S(-1,1) = \F(1,μ) + \F(1,γ) θ_{c+1}

with

θ_{c+1} = \F(1, \B(\F(γ,μ))\S\UP8(c+1)+ \B(\F(γ,μ))\S\UP8(c) + ... +1)

because

$$\Lambda^{-1,1} = \{ \; \mu^{-1} \quad \text{if prob}(n>0), \; \mu^{-1} + \gamma^{-1} \text{ if prob}(n=0)$$

where n is the number of customers in the idle token queue.

This result has already been published by several authors: Hunt[14], Neuts [15], Konheim and Reiser[16], Pujolle and Potier [17], but using different ways.

Case 2

As the total number of customers is c, there is no blocking and the stability condition may be obtained directly studying the equivalent closed model. We can use the computation developed in the preceding section. We obtain:

Λ\S(-1,2) = \F(1,μ) + \F(1,γ) θ_c

5 - Multiclass resource sharing

We can assume that several classes of customers are allowed. We assume that only the parallel allocation resource sharing process and the series allocation with scheme 2 are permitted. These assumptions avoid a blocking situation. If the resources are modeled by a BCMP network, the stability condition is obtained closing the different chains and solving the network. If customers can change of class, they belong to the same chain and the resources are common. But, for two different chains corresponding to a partition of two classes of customers, the resources are different. Let i=1,...,M the subchains and c_i be the number of resources for the chain i.

Example

Let 3 classes of customers be denoted by 1,2 and 3 respectively and with arrival rates λ_1, λ_2, λ_3 respectively. The model is shown in Fig. 5.

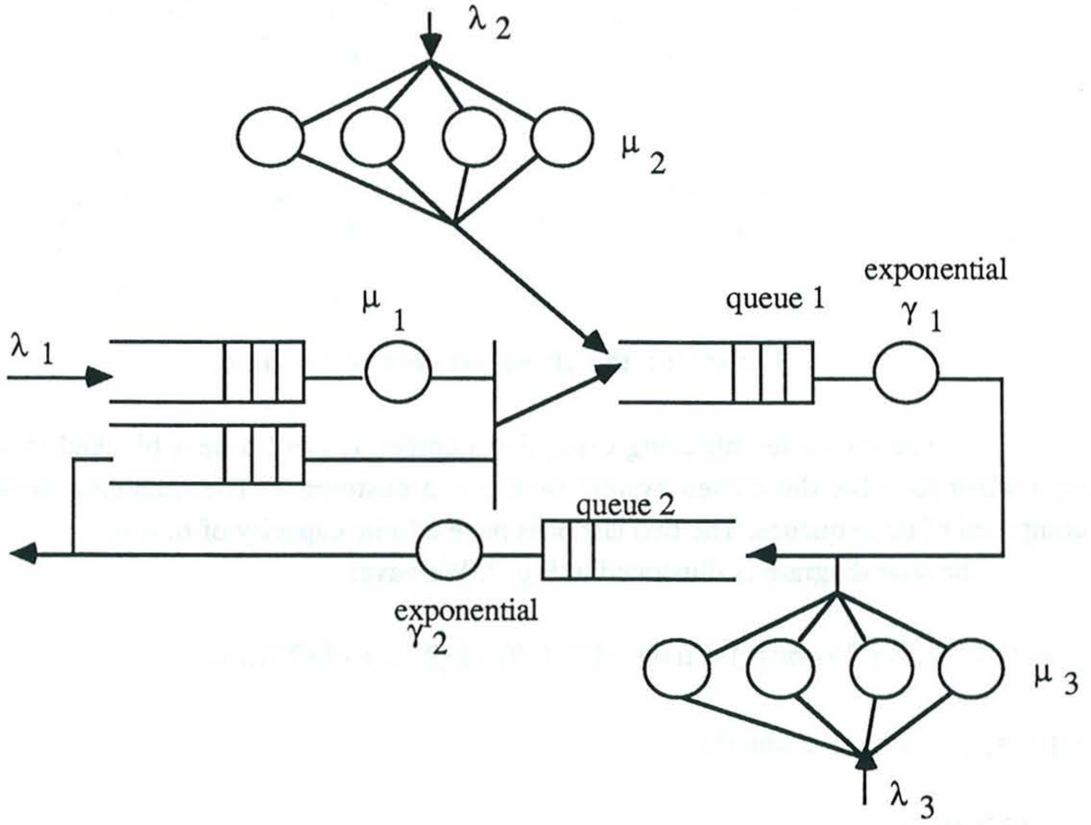


Figure 5 : a model with three classes of customers

A customer of class 1 leaves the system after the service γ_2 and regenerates a token. A customer of class 2 has to go through queue 1 and queue 2, then leaves. Customer of class 3 is routed to queue 2 then leaves the network. The ergodicity condition of such a system is obtained solving the closed system. In fact, the stability condition is given by an n dimensional surface: if a queue is not saturated the maximum throughput of the other queues are increasing. We are interested in the case where all the queues are saturated simultaneously.

If the limited number of customers of class 1, 2 and 3 are 2,4 and 4 respectively, we have to solve the closed model with 3 subchains. If we assume that $\gamma_1, \gamma_2 = 1$ (the service rate is identical for the 3 classes of customers) and $\mu_1 = \mu_2 = \mu_3 = \infty$, namely the allocation time is negligible, we obtain the following ergodicity condition:

$\lambda_1 < 0.35$

$\lambda_2 < 0.35$

$\lambda_3 < 0.18$

We can develop more sophisticated models of resource sharing and blocking systems. It is possible to assume that the central queueing system may have constraints. For example, one or several queues of the central system may have finite capacities. Let us study the tandem queueing system with limited capacities.

First we want to obtain the ergodicity condition or what it is equivalent, the maximum throughput, of the queueing system shown in Fig. 6.

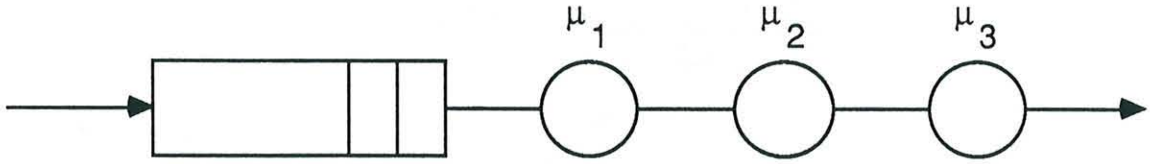


Figure 6: the three servers in tandem

We use the series blocking discipline number 1. Let ' note a blocked state. It is equivalent to solve the closed system with c = 3 customers. The queueing network is composed of three queues. The two last ones have a finite capacity of m = 1.

The state diagram is illustrated in Fig. 7. We have:

$\Lambda = \mu_3$ P(server 3 is busy) = μ_3[1 - P(3,0,0) - P(2,1,0) - P(2',1,0)]

If $\mu_1 = \mu_2 = \mu_3 = \mu$, we obtain:

$\Lambda = [22/39]\ \mu$.

This value has been obtained by Hunt [14].

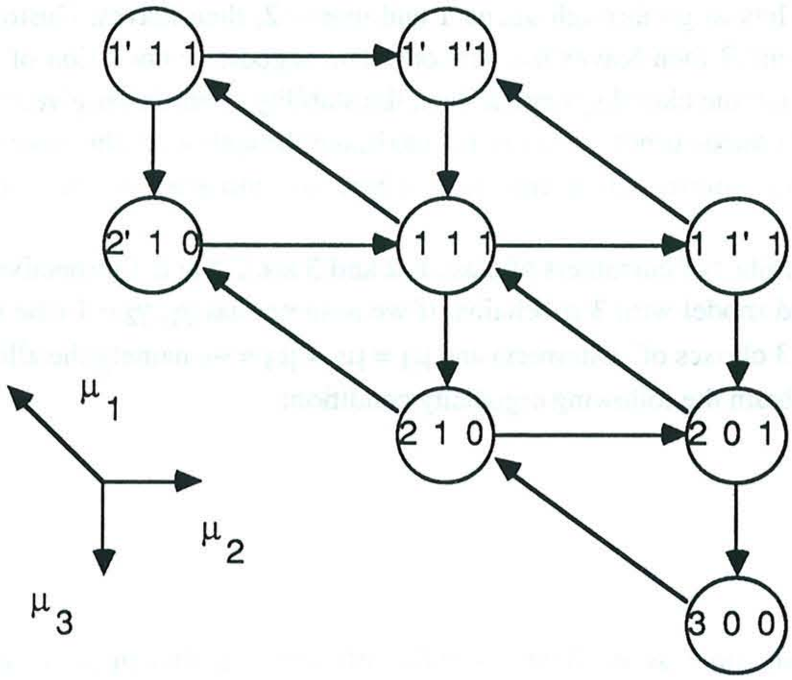


Figure 7: the state diagram of the three servers in tandem

We can solve more complex systems. However, the rate matrix increases very quickly. Let us assume identical service rates (μ =1) and identical buffer capacity m in each station. The maximum throughput Λ for a tandem queueing system composed of N limited capacity queues in series (the total number of queues is N + 1) is obtained:

	N=2	N=3	N=4	N=5
m=1	0.564	0.514	0.485	0.466
m=2	0.670	0.631	0.607	
m=3	0.734	0,700		
m=4	0.776	0.747		
m=5	0.807			

This values have been obtained by Hillier and Boling [15] using another numerical technique.

With the value m=1, for 100 stations in series, we obtain a throughput of 0.254. Then, it seems that when the number of queues in tandem goes to the infinity the throughput tends to 0.25.

We have assumed that all the service times were exponentially distributed. Now, by the same technique we can assume that the service time distribution is more complex. The solution is obtained either exactly using a markov solver or approximately using an approximate solution as Marie's method or MVA method for large networks. In the sequel, we apply the MVA approximation available in the QNAP package. If all the service times have an Erlang 2 distribution, we obtain:

	N=1	N=2	N=4	N=10
m=1	0.69	0.61	0.55	0.52
m=2	0.75	0.67	0.60	0.57
m=3	0.83	0.78	0.73	0.70
m=5	0.90	0.87	0.84	0.81

If all the service times have an Hyperexponential distribution with a squared coefficient of variation equal to 2 (with two servers), we obtain:

	N=1	N=2	N=4	N=10
m=1	0.43	0.32	0.25	0.24
m=2	0.59	0.49	0.44	0.40
m=3	0.66	0.58	0.55	0.49
m=5	0.75	0.68	0.64	0.61

For an hyperexponential distribution with a squared coefficient of variation equal to 10 (with two servers), we obtain:

	N=1	N=2	N=4	N=10
m=1	0.32	0.20	0.15	0.12
m=2	0.51	0.41	0.35	0.31
m=3	0.54	0.44	0.38	0.33
m=5	0.58	0.48	0.41	0.38

If now, instead to have a limited capacity per node, we have a global limited capacity for the whole stations except the first one. For three stations in series, the model to solve is shown in Fig. 8.

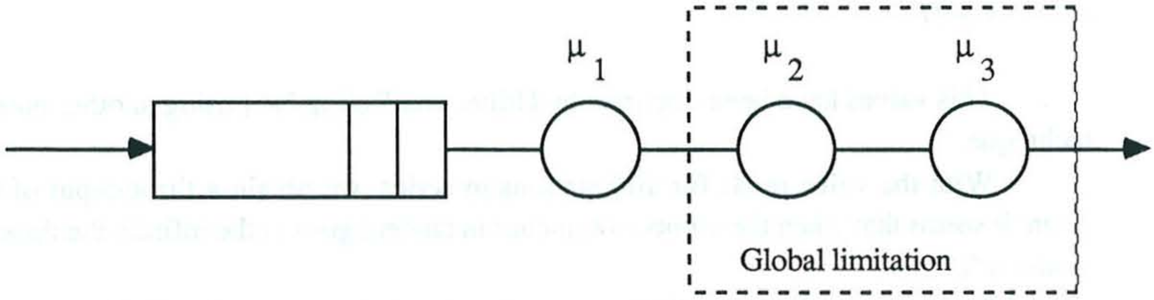


Figure 8: the global limited capacity model

The state space is the same as in the three queues in series plus the state (1',2,0). This implies that the maximum throughput is slightly greater than the maximum throughput of the three servers in tandem. We obtain the ergodicity condition as:

$\Lambda = \mu_3[1 - P(3,0,0) - P(2,1,0) - P(1,2,0) - P(1',2,0)]$

If all the service rates are equal to μ, the maximum throughput is:

$\Lambda = [164/255]\ \mu$.

6 - Applications to computer networks

6.1- Study of a connection-oriented cell switching network

The ATM (Asynchronous Transfer Mode) proposal is a connection-oriented network providing a cell switching network. We are interested in evaluating the performance of such a technique when an end-to-end window flow control is provided.

A call request cell is issued by the sender to form a connection between the sender and the receiver. It permits to negotiate a Quality Of Service (QOS) between the two ends of the connection. We assume that the negotiation leads to a window flow control size of W. Namely, up to W outstanding cells may circulate in the network on the virtual circuit under study. Depending upon how the implementation of the protocol is provided, different kinds of managements of intermediate nodes may be taken into account.

As a first approach, we assume that the call request cell is defining a route inside the network and that all the cells follow this virtual circuit. One possible model is drawn in Fig. 9.

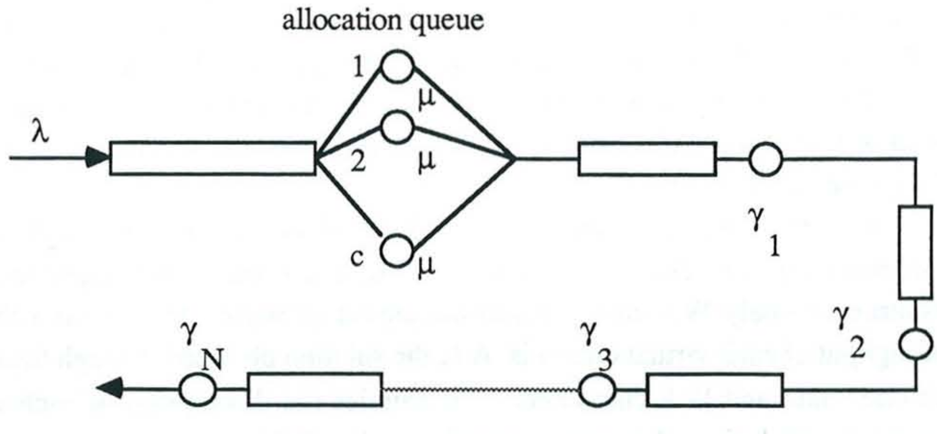


Figure 9: a virtual circuit

The quantity N is the number of nodes to go through and μ^{-1} is the time to execute the code of the ATM level protocol. Depending on the machine which executes this software, the allocation queue is in parallel or in series. We assume here, the machine is a multiprocessor working on several different packets in the same time. We assume that the packet lengths are exponentially distributed. This involves that the service times are exponentially distributed with parameters $\gamma_1,\gamma_2,\ldots,\gamma_N$ depending on the capacity of each line.

The maximum throughput of the virtual circuit depends upon the window size W and the number of nodes in tandem. If we assume $\gamma_i = 1$, $i = 1,2,3,\ldots,N$, we obtain the results depicted in Fig. 10 with the assumption that μ^{-1} is negligible in comparison with γ.

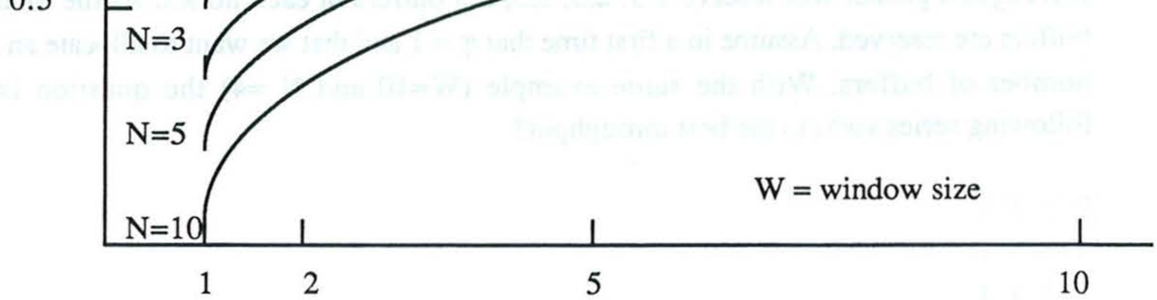


Figure 10: maximum throughput vs W, for different values of m.

As an example, the maximum throughput for a virtual circuit controlled by a window width of 2 for 5 nodes in series is 0.36 if the mean transmission time is 1.

Now, the buffers of the cell switching nodes are shared with other virtual circuits. Moreover, senders are used to forward packets of virtual circuits going to the same following node. The problem is to know the total throughput of the system when the buffers are shared by a large number of virtual circuits. Assume that the model is a common route for a total of k virtual circuits and that each virtual circuit is controlled by a window flow control with the same size: W.

When solving the closed equivalent model, all the infinite queues may be grouped to built only one infinite queue. The total throughput is obtained considering just one class with W.k resources, namely W.k customers in the closed network. It turns out that the maximum throughput of each virtual circuit is Λ/k the solution obtained through the closed network with one chain and W.k customers. This implies the throughput of each virtual circuit is decreasing with k since Λ is approximately γ when W.k is large.

6.2- A connection-oriented packet level protocol with capacity constraints

We assumed in the previous models that the number of buffers in each node is infinite. However, this capacity is finite and is allocated through the call request packet. When the call request packet is going across the network, it reserves resources at each node. If a node has no longer resources to allocate, the call request packet is refused and the virtual circuit cannot be established. In most of the packet switching networks where an end-to-end window flow control is defined (size is W), a total of φW intermediate buffers are reserved such that statistically the probability of overflow be negligible. The quantity φ is the overallocation factor that we are going to describe.

If θ is the utilization of a virtual circuit, on the average $W\theta$ packets are along the route. The value of θ ($0 \leq \theta \leq 1$) may involve an overallocation of buffers such that only φW buffers are reserved in the network where φ is a function of θ. The quantity φ is generally defined as a value greater than the maximum value of θ measured in the network in a whole day.

If W is the window flow control size, the call request packet will reserve only φW buffers in each node. The problem is to find what is the best allocation of buffers along a virtual circuit. For example, if W = 10 for a virtual circuit with 4 nodes and $\varphi = 0.25$, the call request packet will reserve 2.5, 2.5, 2.5, 2.5 buffers in each nodes. At the total $NW\varphi$ buffers are reserved. Assume in a first time that $\varphi = 1$ and that we want to allocate an integer number of buffers. With the same example (W=10 and N =4) the question is: what following series yield to the best throughput?

2 2 3 3
3 3 2 2
1 3 3 3
3 2 2 3
....

In the sequel, we assume a series allocation queue with scheme 2. Indeed, we have chosen this discipline because it works as in computer communication systems. We shall assume that all the service time are exponentially distributed with the rate γ_i, $i = 0,\ldots,N$.

The model with capacity constraints is now:

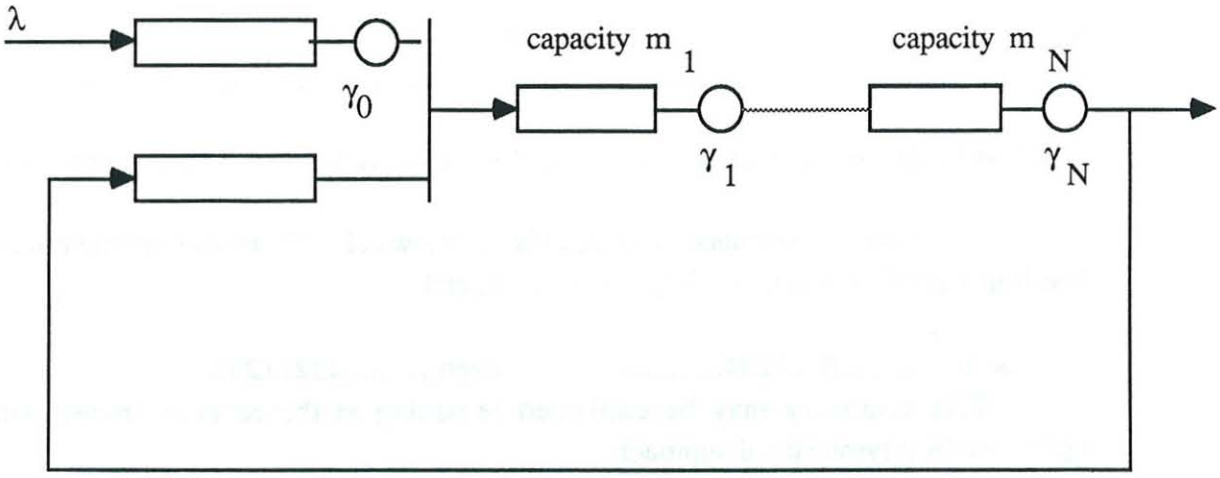


Figure 11: the model with capacity constraints

where m_i is the maximum buffer size at node i, $i = 1,\ldots, N$ to m and $\sum m_i = W$. We, assume that the first node is composed of the allocation queue.

Using the technique described in section 2, we have to solve the associated closed queueing system. In this model, a blocking situation occurs when a customer tries to enter a full queue. A packet which cannot enter a node is rejected. A rejected NPDU returns to the server of the preceding station and get another round of service independent from the one it received before. This is known as rejection blocking model and it is described in Perros [18] and Caseau and Pujolle [19].

We assume that mean service times are identical $\gamma_i = 1$, $i = 0, 1,\ldots\ldots,m$. The mean allocation time is equal to the mean service time of the other stations. The closed associated network is shown in Fig. 12. The total number of customers is W.

This system can be solved using a numerical approach when the number of queues in series and the number of customers are sufficiently small. Using the Markov method developed by Stewart [19], models with 10 queues in series and 20 customers can be solved. Moreover, MVA analysis allows to obtain the solution for huge network.

The result we obtain (but without a formal proof) on all the tested patterns shows it is necessary to advantage the stations situated in the middle of the system. More a station is near from the middle of the system, more it should be advantaged. This result has been obtained by Hillier and Boling for small tandem queueing systems [20].

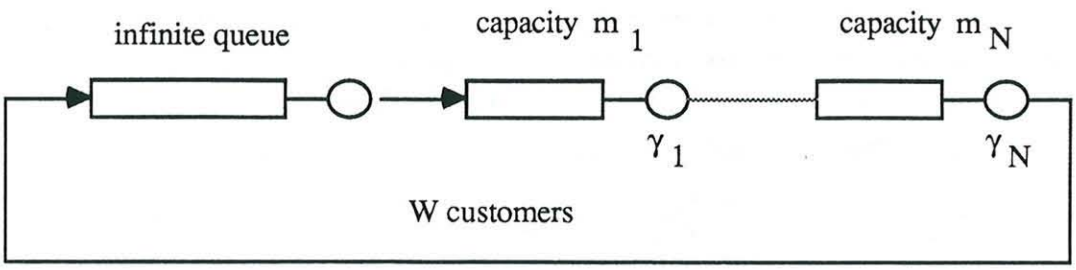


Figure 12: the closed queueing system for studying capacity constraints

The general importance of a queue is as follows (1 is the most important queue, 2 is less important than 1, 3 is less important than 2, etc)

odd..............4321234.............   even..........43211234

This symmetry may be explained regarding at the reverse process which is optimized by a symmetrical approach.

For example, for 4 nodes and a window flow control size of 10 the best performance is obtained by the series: 2 3 3 2.

If the window flow control size is only 9, for 4 stations, it is the second or the third station which must be enforced. We obtain the following series: 2 3 2 2 or 2 2 3 2.

For 8 as a size, the balanced state 2 2 2 2 is optimum and for a total capacity of 11, the two following states are equivalent 3 3 3 2 or 2 3 3 3.

In the real network, if the overallocation factor is 0.1 with a window of 10, the resources reserved at the respective nodes will be: 0.2 0.3 0.3 0.2. However, this does not seem to be the best allocation if the study is performed in the following way: let $W^* = \lceil W/\varphi \rceil$ an integer. For example, if $W = 10$ and $\varphi = 0.1$, $W^* = 100$. We obtain as the best solution:

23 27 27 23

References

[1] H. G. Perros, "Queueing networks with blocking: a bibliography" ACM Sigmetrics, Performance Evaluation Review, August 1984.

[2] P. A. Jacobson and E.D. Lazowska, "Analysing queueing networks with simultaneous possession", Proc. Performance 83, C. ACM 25, pp 142-151, Feb 1982.

[3] D.J. Freund and J.N. Bexfield, "A new aggregation approximation procedure for solving closed queueing networks with simultaneous resource possession", Proc. ACM Sig. Conf. on measurement and modeling of computer systems, Minneapolis, pp 214-224, 1983.

[4] C.H. Sauer, "Approximate solution of queueing networks with simultaneous resource possession", IBM J. Res. Develop. 25,6, Nov 1981.

[5] A. Thomasian, "Queueing network models to estimate serialization delays in computer systems", Proc. Performance 83, University of Maryland, College Park, North Holland, pp 45-59, 1983.

[6] S.C. Agrawal and J.P. Buzen, "The aggregate server method for analyzing serialization delays in computer systems" ACM TOCS, 1, 2, pp 116-143, May 1983.

[7] G. Pujolle, S. Fdida, "General resource sharing systems" Proc. International Seminar on Teletraffic Analysis and Computer Performance Evaluation, Amsterdam, North Holland, June 1986.

[8] S. Fdida, D. Mailles, G. Pujolle, TSI

[9] J.R. Jackson, "Jobshop like queueing system", Manag. Sc. 10, pp 131-142, 1963.

[10] BCMP, "Open, closed and mixed networks of queues with different classes of customers", J. ACM 22, pp 248-260, 1975.

[11] F.P. Kelly, Reversibility and stochastic networks, Wiley, 1979.

[12] FKAS, Queues and point processes, Akademie-Verlag, Wiley, 1981.

[13] S.S. Lavenberg, "Stability and maximum departure rate of certain open queueing networks having finite capacity constraints", RAIRO Bleue 12, pp 353-370, 1978.

[14] G.C. Hunt, "Sequential arrays of waiting lines" Oper. Res. 4, pp 674-683, 1956.

15] M. Neuts, "Two queues in series with a finite intermediate waiting room. J. Appl. Prob. 5, pp 123-135, 1968.

[16] M. Reiser and A. Konheim, "Finite capacity queueing systems with applications in computer modeling", SIAM Journal on Computing 7, 210-229, 1978.

[17] G. Pujolle and D. Potier, "Réseaux de files d'attente à capacité limitée", RAIRO Bleue 13, 2, pp175-197, 1979.

[18] P. Caseau and G. Pujolle, "Throughput capacity of a sequence of queues with blocking due to finite waiting room" IEEE Trans. on Soft. Eng. 5, 6, pp 631-642, 1979.

[19] H. G. Perros, "Approximation analysis of open networks of queues with blocking: tandem configuration", IEEE Trans. on Soft. Eng. 12, pp 450-462, 1986.

[20] F.S. Hillier and R.W. Boling, "Finite queues in series with exponential or Erlang service times - a numerical approach", Oper. Res. 4, pp 286-303, 1967.

DISCUSSION

Rapporteur: Jonathan Spencer

Lecture one
Professor Randell stated that the presentation gave the impression of ever increasing complexity. This raised the question of where modelling fitted into the design process. Professor Pujolle felt that it was essential to have modelling tools and that simulation can make the complexities of network systems far easier to handle: an accurate model should lead to an accurate system. Professor Randell asked what modelling had been used in the debate on whether to use 32-bit or 64-bit cells? Professor Pujolle answered that none had been used. Professor Shepherd commented that modelling is normally used after design decisions have been made but went on to note that DQDB originated in Australia and a lot of modelling was done to determine fairness. Another participant pointed out that performance and fairness are competing aims. Professor Shepherd also noted that it was a model which had shown that Ethernet could not possibly work.

Lecture Two
Mr Hughes asked what use are models, given that there are problems predicting traffic in systems such as an ATM? Professor Pujolle agreed that modelling arrivals was not easy, but felt that modelling was useful nonetheless. Professor Randell asked whether the development of specialised environments and tools for complex systems was deskilling? Professor Pujolle explained that the idea was to put all our models together and let the user adjust the parameters e.g. node numbers. Although specialised models have been developed the results obtained so far are unsatisfactory. Mr. Hughes suggested that two kinds of interface are required: one for the specialists (the modellers) and another which is domain-oriented (for the customer). Professor Randell asked whether these interfaces might help the users who need protecting from themselves e.g. in the interpretation of results? Mr. Hughes thought this was the only approach. Professor Gelenbe raised the problem of paradigms in which the modellers talk in terms of queues and performance, while the users talk in terms of CPUs and MIPS. The terminology must be restricted to the correct sphere. Modelling is sophisticated and must be kept away from the users although this is not deskilling. The solutions/results are distinct from the statistics and interpretations drawn from them.