

Overallocation in a virtual circuit computer network

A. Kurinckx, G. Pujolle

► **To cite this version:**

A. Kurinckx, G. Pujolle. Overallocation in a virtual circuit computer network. [Research Report] RR-0031, INRIA. 1980. <inria-00076530>

HAL Id: inria-00076530

<https://hal.inria.fr/inria-00076530>

Submitted on 24 May 2006

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

IRIA

Rapports de Recherche

N° 31

**OVERALLOCATION
IN A VIRTUAL CIRCUIT
COMPUTER NETWORK**

**Alain KURINCKX
Guy PUJOLLE**

Septembre 1980

Institut National
de Recherche
en Informatique
et en Automatique

Domaine de Voluceau
Rocquencourt
B.P. 105 78150 Le Chesnay
France
Tél. 954 90 20

OVERALLOCATION IN A VIRTUAL CIRCUIT COMPUTER NETWORK

Alain Kurinckx
INRIA
78150 Le Chesnay

and

Guy Pujolle
ENST
75634 Paris Cedex 13
and INRIA

- level 2 - Link access procedure (LAP). This specifies a datalink control procedure for converting the error-prone physical circuit into a relatively error-free link. It is based on the high level data link control (HDLC) defined by the International Standards Organization (ISO).
- level 3 - packet level. This is the highest level. It specifies how information is structured into packets. Level 3 provides the facilities for establishing virtual circuits which are bidirectional associations between a pair of DTEs over which packets are exchanged.

1. Introduction

CCITT Recommendations X.25 [1] has been selected by some telecommunication organizations as one of their packet switching network user interfaces, e.g., Datapac (Canada), DDX (Japan), Telenet (USA), Transpac (France), KDD (Japan). However due to implementer's individual interpretations and additions, the X.25 recommendations are somewhat moving. The main differences have been listed in [2].

First we discuss the main characteristics of X.25 Recommendations and then we detail a possible interpretation. In this context, we develop a mathematical model to study a method that allows a number of users (sharing the resource of a node) greater than the theoretical number.

2. The X.25 Recommendations [3].

X.25 specifies the protocols or rules for exchanging information between similar levels in the Data Terminal Equipment (DTE) and Data Circuit - terminating Equipment (DCE) as shown in Figure 1.

These levels have been identified in the X.25 Recommendations. Each level accepts information from a higher level and adds a header and possibly a trailer before passing the information across the interface presented by the next lower level.

- level 1 - Physical interface. This specifies the electrical and physical characteristics of the interface to the leased or switched line into the network and how to establish and control these connections.

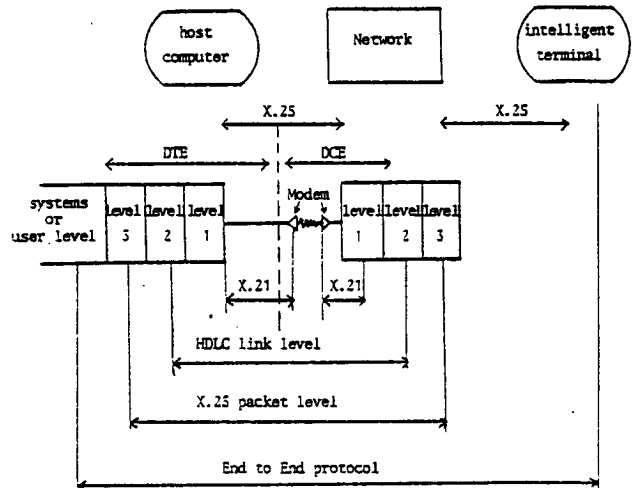


Figure 1 : The X.25 protocol

In this paper we are interested mainly in the level 3 and particularly in the use of virtual circuits.

The packet level of X.25 performs an asynchronous time division multiplexing function to transform the single logical channel provided by level 2 into a number of logical channels. The logical channels are used to provide bidirectional associations between two end DTEs as shown in Figure 2.

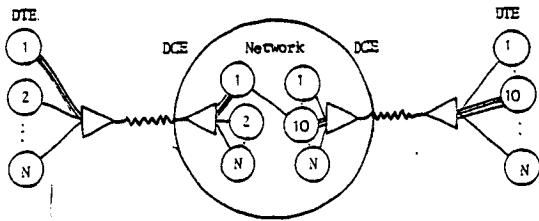


Figure 2 : A virtual circuit

These associations are known as virtual circuits. The difference between a virtual circuit (VC) and a physical one is the following : a bandwidth is allocated only when packets are being transferred. A virtual circuit has to be set up by an exchange of packets before informations transfer can take place.

X.25 Recommendations does not specify how the control information carried by the packets should be interpreted. This could depend on how level 3 is implemented : either to perform end-to-end control or to perform local control across the DTE/DCE interface.

X.25 Recommendations level 3 provide a window flow mechanism for flow control, i.e. preventing congestion by limiting the number of packets accepted by the network. An independent window-flow control is used either for each logical channel across the interface or for each virtual circuit with the end-to-end possibility. The window size w is the number of unacknowledged packets that a "end" can have outstanding at any time. It is a constant number chosen by the network at subscription time or at call set up time.

The X.25 Recommendation merely defines the technique of accessing the network but allows some freedom in the choice of the internal switching strategy within the network.

The service may be based on different switching technologies, for example datagram switching (see Datapac network) or virtual line switching (see Transpac network).

3. Overallocation in virtual line switching network

In this paper we are interested in the end-to-end control through virtual circuits. In this case, adaptative routing is used to select a path at call set-up time; thereafter fixed routing is used, i.e., data packets of a particular end-to-end virtual circuit follow a fixed path. The set-up packet should reserve some resources when going across a node. In particular, the call set-up packet specifies the window size w and stores up an equal number of buffers in each node it goes through. So, the maximum number of VCs a node can accept is determined by the sum of their window sizes. It is clear that this scheme is the main drawback of the virtual line switching technique : an important part of the resources are idle. On the other hand, the main advantage is that there is no overflow, since a packet trans-

mitted by a host finds a buffer at each node. Overallocation consists in allocating more virtual circuits in a node so that the sum of the window sizes exceeds the number of buffers. The aim is obviously to increase the utilization of buffers with the constraints that the overflow probability remains very small.

With this overallocation scheme, the virtual line switching technique takes benefits from the virtual circuit method (no overflow) and from datagram method (throughput and resource utilization). In the following section, we develop a mathematical model to predict the maximum overallocation with a fixed probability of overflow. We consider two cases, first when virtual circuits are saturated, then when virtual circuits are used only a fraction of the time.

4. The mathematical modelling of the saturated case.

For the sake of simplicity we assume that the mean size of a packet corresponds to the size of one buffer. Let M be the number of buffers available at each output line of a node. We assume all the switching nodes are identical. The problem under study is to find the maximum possible overallocation of buffers that a switching node can support with an overflow probability less than a fixed value ϵ . First let \bar{K}_i be the mean number of nodes a packet has to go through to reach its remote destination terminal equipment (DTE), given that it goes through node i . If the routing is able to balance the load such that each node supports approximately the same traffic, we may model virtual circuits using node i as a \bar{K}_i tandem queueing network (Figure 3). In the sequel we remove subscript i such that $\bar{K}_i = \bar{K}$. Notice that we study only node i , and the other nodes of the tandem queueing network represents the interferences due to packets traveling along all the other paths. The arrival process to node i is very close to a Poisson process in the physical system since this process is the superposition of a number of arbitrary processes. In our model the arrival process is also close to a Poisson process. So, our model conserves the mean of the arrival process to the node i and the arrival distribution is very well approximated. Notice also that acknowledgements are piggy-backed in return packets and do not increase the traffic.

Let an elementary virtual circuit (EVC) be a circuit whose window size is equal to 1. We assume there are only EVCs in the network : from the throughput point of view a virtual circuit with window w is equivalent to w EVCs.

Thus, in the model shown in Figure 3, without overallocation, only M EVCs may be set up. Our problem is to determine the largest number m of EVCs that the network can support so that the overflow probability is less than ϵ .

Let us assume that the queue lengths are not limited and let us compute the probability to have more than M packets in station i . It is well known that this approximate method yields accurate results if all the overflow probabilities are small. We are going to examine the worst case : we assume there is always one packet ready to be transmitted by the host. Due to this saturation condition the original model can be transformed into the closed queueing network shown in Figure 4. This closed network contains m packets (or m EVCs). This number

will be called the overallocation.

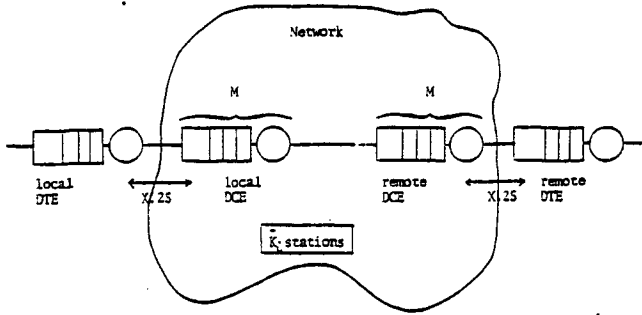


Figure 3 ; A first modelling of virtual circuits

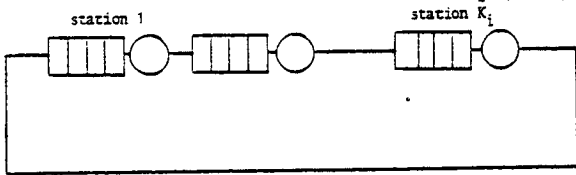


Figure 4 : A closed queueing network

As we assume identical stations, with exponentially distributed service times of parameter μ , if n_j is the number of customers at station j , the steady state joint queue size distribution is [4].

$$P(n_1, \dots, n_{\bar{K}}) = \frac{1}{G(m)} \cdot \prod_{i=1}^{\bar{K}} \left(\frac{1}{X_i}\right)^{n_i}$$

where $(X_1, \dots, X_{\bar{K}})$ is a solution of the equations

$$X_j = X_{j-1} \quad 1 < j \leq \bar{K}$$

$$X_1 = X_{\bar{K}}$$

Obviously one may choose $X_i=1$ ($1 \leq i \leq \bar{K}$) and

$$P(n_1, \dots, n_{\bar{K}}) = \frac{1}{G(m)} \cdot$$

If $S(m, \bar{K})$ is the set of possible states of the system we have :

$$G(m) = \sum_{n \in S(m, \bar{K})} P(n_1, \dots, n_{\bar{K}}) \\ = \text{Card}[S(m, \bar{K})] = \binom{\bar{K} + m - 1}{m}$$

where $n = (n_1, \dots, n_{\bar{K}})$ and $\text{Card}[X]$ denotes the cardinality of set X .

The probability of having more than M packets in station i is [4] :

$$P(n_i > M) = \sum_{\substack{n \in S(m, \bar{K}) \\ \text{and } n_i \geq M}} P(n_1, n_2, \dots, n_{\bar{K}}) = \frac{G(m-M-1)}{G(m)}, \quad m \geq M$$

$$= \frac{m(m-1) \dots (m-M)}{(\bar{K}+m-1) \dots (\bar{K}+m-1-M)}$$

The maximum value of m for which $P(n_i > M) \leq \epsilon$ is provided by the solution of the equation :

$$\frac{m \dots (m-M)}{(\bar{K}+m-1) \dots (\bar{K}+m-1-M)} = \epsilon$$

This equation has \bar{K} solutions. Notice that the largest of these solutions has to be taken. Notice also, that the solution grows nearly linearly with m . This could be explained by the following argument which gives a good approximation of the overallocation :

$$P(n_i > M) = \frac{(m-M) \dots (m-M+\bar{K}-2)}{(m+1)(m+2) \dots (m+\bar{K}-1)} = \epsilon$$

$$\Leftrightarrow (m-M) \dots (m-M+\bar{K}-2) = \epsilon(m+1)(m+2) \dots (m+\bar{K}-1)$$

The right side could be approximated by :

$$(m-M-1 + \frac{\bar{K}}{2})^{\bar{K}-1} \approx \epsilon(m + \frac{\bar{K}}{2})^{\bar{K}-1}$$

For the largest solution we obtain :

$$m-M-1 + \frac{\bar{K}}{2} \approx \epsilon^{\frac{1}{\bar{K}-1}} (m + \frac{\bar{K}}{2})$$

$$m = [M+1 - \frac{\bar{K}}{2} (\epsilon^{1/(\bar{K}-1)} - 1)] / [1 - \epsilon^{1/(\bar{K}-1)}]$$

We show in Figure 5 the overallocation as a function of \bar{K} and M , obtained by a simple numerical program from the general equation.

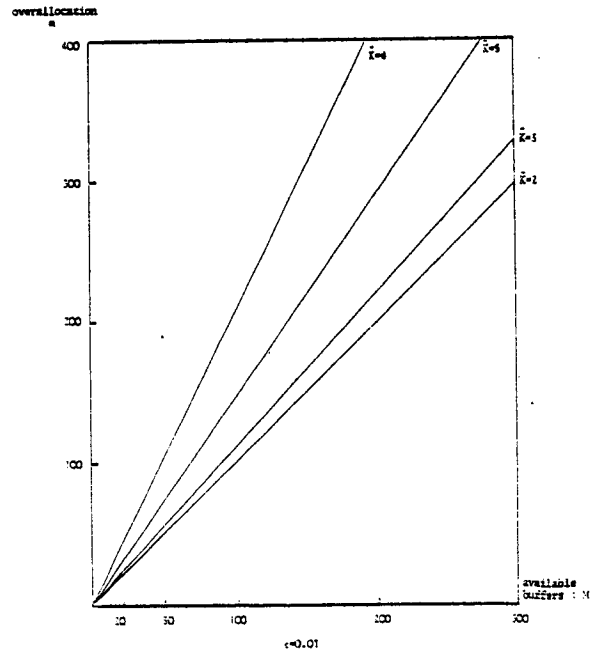


Figure 5 : Overallocation versus the number of available buffers.

To characterize the system more completely, we have to compute two other performance criteria : the throughput and the response time. The throughput α is obtained as follows :

From the first section the probability of overflow when there are i EVCs in the network of figure 4 is

$$\epsilon(i) = \frac{i \dots (i-M)}{(K+i-1) \dots (K+i-1-M)} \quad i > M$$

The probability of overflow can be written as :

$$p = \sum_{i=M+1}^N p(i)\epsilon(i)$$

$$= \sum_{i=M+1}^{m'} \frac{m'(m'-1) \dots (m'-i+1) \theta^i (1-\theta)^{m'-i}}{(i-M-1)! (K+i-1) \dots (K+i-1-M)}$$

As p is increasing with m' , it is easy by an algorithmic method to obtain the value of m' corresponding to $p=\epsilon$. Some results and conclusions are given in the next section.

6. Conclusion and practical consideration

We show in Figure 7 the maximum overallocation versus the available number of buffers for a probability of overflow of .001. The curves obtained for different θ and \bar{K} are almost linear except with very small number of buffers. Note again that the overallocation is very dependent on the mean length of the virtual circuits. But even with small \bar{K} , overallocation is allowed as soon as the virtual circuits are poorly used.

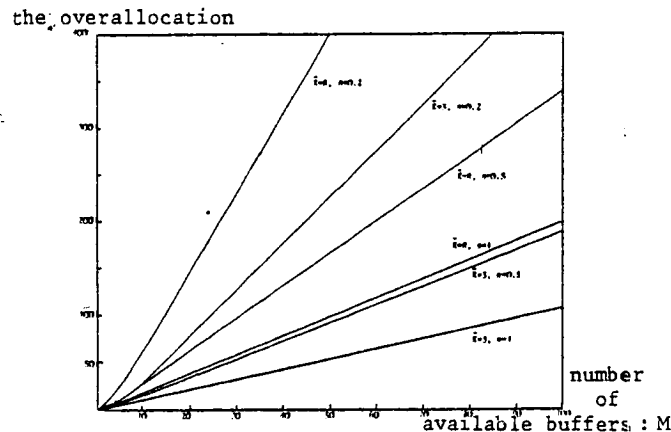


Figure 7 : The overallocation versus the available number of buffers.

Figure 8 shows the overallocation versus the occupancy of virtual circuits for \bar{K} and m in their practical range. Note that the values of the overallocation obtained with $\theta=1$ are these computed in the saturated case (section 4).

Let us take an example to exhibit the effect of overallocation on throughput when θ is small. With 8 stations, 20 buffers per output line, occupancy of VC 0.3, one can allocate 100 EVC at each node. Without overallocation, the equivalent number of saturated EVC is 6 and the observed throughput (from figure 6) will be close to 0.45. With the mechanism we describe in the previous section, the system supports 33 saturated EVC on the average and reaches a throughput of 0.82.

Now it remains to investigate the implementation of this mechanism. The first problem is the computation of \bar{K} for each node of a given network. This value depends on the topology and on the end-to-end traffics. With fixed routing the problem is rather simple but if an adaptive routing policy is used,

the only possibility is to have in each station an estimator of \bar{K} . The estimate must be updated each time a virtual circuit is set up or removed in the station.

the overallocation

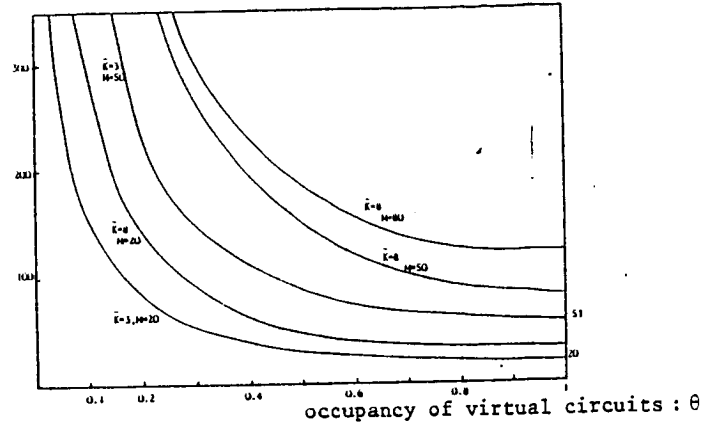


Figure 8 : The overallocation versus the occupancy of virtual circuits.

Figure 8 can be used to implement the dynamic control if the system supplies an estimator of θ (as in Transpac). Let us assume that, at time t , a station receives a call set up. Using the two estimates and the curves, one can decide if it is possible to accept the new VC or not. Note that, if the estimate of θ increases or the estimate of \bar{K} decreases, the system must release some virtual circuits to avoid high probability of buffer overflow. Obviously switched virtual circuits are to be released first.

If θ is small, a slight increment of θ requires a large reduction in overallocation. So to exploit the whole overallocation, it is necessary to make sure of the quality of the estimator.

In this paper we have shown that a dynamic overallocation mechanism can cure the drawback of resource underutilization which is often objected to in the VC computer network even if the activity of each VC is small.

Packet switching networks are now a reality. Two levels of facilities have been proposed : virtual circuit and datagram. The properties of the two facilities have been largely reviewed. The question is : what to choose datagram or virtual circuits ? Due to the decreasing price of memory, virtual circuit service is going to become a very suitable technique. However we have seen that it is necessary to implement an overallocation mechanism since virtual circuits are badly used.

The datagram flow control mechanism which is known to be a complex problem is replaced by the no less complex overallocation mechanism

Acknowledgements

We are grateful to G.S. Graham and to two anonymous referees for commenting on the early draft of this paper and for some improvements in section 4.

References

[1] CCITT "Interface between DTE and DCE for terminals operating in the packet mode on Public Data Networks", CCITT Recommendation X.25, 1977.

- [2] HESS M.L., BRETHERS M., SAITO A. "A Comparison of four X.25 Public Network Interfaces", Proc. of ICC 79, Boston 1979.
- [3] SLOMAN M.S. "X.25 explained", Computer Communications 1, 6, 1978.
- [4] BUZEN J.P. "Computational algorithms for closed queueing networks with exponential servers", CACM, 16, 9, 1973.

