MODELING AND PERFORMANCE EVALUATION OF A PACKET SWITCHING MODULE IN AN X.25 NETWORK

GONZALEZ SOTO, O* / MARTINEZ MIGUEZ, L* / NIEMEGEERS, I.G**

* Centro de Investigación de Standard Eléctrica, S.A. Madrid, Spain.
** Bell Telephone Mfg. Co. Antwerp, Belgium (now at Twente Univ., Holland).

ABSTRACT

This paper presents the analysis of a packet switching module (PSM) as the most important element of a packet switching network. The PSM modeling is the lowest study level of the four hierarchical levels developed for the complete analysis of a packet switching network.

Analytical and simulation models were developed and applied, with the twofold objective of investigating the importance of the simplifying assumptions and the behavior of the PSM under various environmental conditions. From the results obtained, and their sensitivity to the traffic parameters involved, the importance of the processing in the modeling of higher levels in the packet switching network is derived.

INTRODUCTION

A public packet switching network that supports CCITT X.25 interfaces (1) consists of a number of switching and user interface nodes which are interconnected by communication links. In this paper we start with a network of this type, the DN-1 (2), and describe the structure of its nodes and the organization of one of its basic components, the packet switching module (PSM).

Because of the distributed nature of such a network, the approach to its analysis was to build a hierarchy of four performance modeling levels: global network, user communication path, node and module. The criteria used to define the four levels are based on the semi-independence of the levels, the use at each level of results obtained in the lower ones, and an autonomous performance analysis at each level. Basic thoughts for that criteria are treated in (3).

Several attempts at hierarchical modeling have been made in the past in other fields and with slightly different criteria, such as the hierarchical set of models for multiprogrammed virtual memory time-shared computer systems using paging demand (3), the two macro-micro levels for the modeling in complex computer systems (5), and the hierarchy by complexity and degree of representation for simulation of processing systems (6).

This paper deals with the key element of the lowest level in the hierarchy, the packet switching module; a companion paper (4) deals with the third level of the hierarchy, the user communication path. The performance analysis of the PSM, which implements protocol and switching functions above the link protocol level, is performed by means of analytical and simulation approaches. The processing activities, queueing structure and service discipline of a PSM are described. The general traffic assumptions considered necessary for the capacity analysis are presented, as well as the additional simplifying assumptions that were applied to the analytical and simulation models. The annex to the paper describes the feedback model developed to represent the packet flow through the PSM.

Results obtained are the processor occupancy, delay per activity and sequence, memory used by the work queues, and grade-of-service related delays such as distributions of call requests and data packet delays. The various analytical and simulation approaches are compared in order to investigate the importance of simplification in priority and feedbacks, and to evaluate the future need for model development.

As a central objective, the behavior of the packet switching module under various environmental conditions and traffic classes is investigated. The knowledge of this behavior will make it possible to dimension the module and provide the performance inputs for the upper levels of the hierarchy of the packet switching network analysis.

2. NETWORK STRUCTURE

The purpose of the network under consideration is to provide a switched and permanent virtual circuit service to its users, in accordance with the CCITT recommendation. The packet level of the X.25 protocol defines the multiplexing of a number of bidirectional logical channels on a frame link between a data terminal equipment or a user and the network. A logical channel corresponds to a switched virtual circuit (also called virtual call), or to a permanent virtual circuit.

Since the network uses the packet switching technique for multiplexing network bandwidth among its users, the network does not handle virtual circuits internally. It operates on packets which have no apparent relationship with each other, and which travel independently towards their destination. Packets in the highest hierarchical level of this network have an additional network header identifying a destination, and are called "netgrams".
The virtual circuit service is implemented at the boundaries of the network, which operates as a general transport network. There are two types of nodes: the packet switching exchanges which form a meshed subnetwork, and the packet data satellites, which give access to the data terminal equipments and are linked to the packet switching exchanges subnetwork via a single packet switching exchange (PSE) (Figure 1).

A node consists of a variable number of functionally dedicated processors, each with its own memory and interconnected via a high-bandwidth node bus. The functions implemented at each processing module depend on the facilities associated with each node and protocol implementation.

3. HIERARCHICAL APPROACH

It is not appropriate to analyze the network structure described above as a unit, because it would be impossible to represent the system with sufficient realism to provide the many performance outputs. Because of this complexity, and the distributed nature of packet switching, the approach followed was the creation of a nesting hierarchy of four performance modeling levels. The criteria are based on maintaining semi-independence between levels, using at each level the results of the lower ones and providing autonomous performance results at each level. By applying these criteria, a network performance hierarchical modeling (NEPER-HIM) was built consisting of four models (Figure 2).

The first and most basic level for the module analysis (MODA) consists of the modeling of the various functional modules involved, e.g. packet switching module (PSM), line access module (LAM), interconnection module (IM), node data base (NDB), etc. The second level considers the evaluation of the node capacity (NODA) at the different network stages e.g. node access, switching node. The third level deals with the user communication path (UCOP), consisting of the permanent or switched virtual circuits established through the network. The fourth level is a network global model (NETG) for the analysis of general network mechanisms such as routing strategy and congestion control. This partition allows detailed analysis at each level from the point of view of performance, grade of service evaluation and optimization, and permits the necessary realism by the use of simplified results provided by the lower levels.

In the following analysis we consider only the MODA level and the corresponding application to the most important element in packet switching, the packet switching module.

4. STRUCTURE OF A PSM

4.1 Configuration

A PSM consists of a CPU, a memory and an I/O interface connected by a module bus 16 data bits wide. There is a single 16 bit wide I/O path; all communication with the outside world is by messages that travel across the node bus to which the I/O interface is connected.

Two types of messages are defined: short messages containing 16 bits of data and long ones containing multiples of 16 bits. Short messages are exchanged between modules by direct sending without previous request of permission; long messages are subject to flow control. A relatively simple module-to-module protocol is defined which uses requests for transfer and acknowledgements of information. I/O operations are handled by microcode and are transparent to the software.

4.2 Task Scheduling

The software of the PSM consists of a number of processes and a monitor. The monitor schedules the processes, performs I/O operations and manages processor resources such as memory and timers. The processes may have up to four work queues where the specific tasks are executed. A task can be described as a program that processes one message at a time; a work item is either a message received from another module in the node (e.g. a netgram) or a message sent by another task. When it starts processing a message, the task has exclusive use of the processor until it finishes and returns control to the monitor. Each process can
have several work item queues, and all work queues are priority ordered by a scheduler which uses head-of-line priority scheme. Messages in the same work queue are processed in first-in-first-out order. Every time a task finishes processing a message and returns control, the monitor services the input queues which contain messages arriving from other modules. The PSMs in the packet switching exchange deal mainly with netgrams; three processes are defined: virtual call request, netgram routing and billing.

5. PSM Modeling

5.1 General

The PSM can be considered as a single server priority queueing system with different classes of flows and feedbacks between queues. The single server is the CPU, which is shared by the monitor, the processes with their corresponding working queues, and the asynchronous input of messages by a microcoded input handler. The units to be served by the PSM are the netgrams already explained in the previous sections.

![Figure 3 - General PSM Model](image)

Figure 3 represents the general model of the PSM with the associated queues, activities and feedbacks. There is a set of assumptions common to the analytical and simulation approaches:

a) Since the asynchronous input is interleaved with task and monitor processing at the instruction level, its effect is to slow down this processing. Therefore, the input of messages can be seen as the highest preemptive priority activity.

b) Each task work queue has a priority level; the task and monitor are not allowed to preempt one other. The monitor processes all items in the long-message input queue and short-message input queue before scheduling a task with the highest priority messages in a work queue. Thus, the message input queues can be seen as the highest non-preemptive priority queues.

c) The arrival of units to be served, i.e. netgrams, is assumed to be Poissonian.

d) Based on the projected use of the network, the decision was made to distinguish two classes of calls: interactive and batch. Interactive calls were assumed to contain an average of 2.5 data packets of 600 bits, and positive acknowledgement (RR packets) for every data packet. Batch calls were assumed to consist of 114 full-size data packets (1000 data bits), and acknowledgement for every two packets. (Window size for the basic service is as defined in Recommendation X.25.)

e) A parameter was defined to characterize the mix of traffic classes; the interactive-to-batch net user traffic ratio. \( \text{I/B} = \frac{\text{interaction}}{\text{batch}} \)

f) In addition to the call rate and traffic mix, which characterize the work load, the processing times also determine the PSM behavior. These processing times were estimated at various stages in the design evolution, either from instruction count or environment simulation procedures.

5.2 Analytical Approach

The inherent difficulty of treating the complexity of the mixed priority with feedbacks associated with each traffic class, together with the need to obtain exhaustive performance sensitivities to each traffic environment, greatly reduce the application of analytical models to the solution of the problem. The analytical approach employed in the study makes it possible to obtain only average values suitable for use as general guidelines to qualitative behavior, and to investigate the impact of the various simplifications introduced.

In order to make the models analytically tractable, the following additional assumption was made: Messages entering a queue at a particular priority level require a processing time, which is a random variable having a distribution function which is independent of message type. Two models considered in accordance with this assumption are explained in the following paragraphs.

5.2.1 Independent Priority Levels Model

The independent priority levels (IPL) model is based on the assumption that there is no feedback between levels. This model considers mixed preemptive priority for the highest microcode priority, and head-of-line priority for working queues. Fresh arrivals of work are considered for all the queues, which implies no feedback between them.

The results for a pure head-of-line priority (first obtained by Cobham (8) for a k-class priority process) are also applicable in this case, with the addition of a subsequent waiting time for each level corresponding to the first level preemption.

5.2.2 Feedback Model

This second, more complex, analytical model considers the same degree of detail in the priority organization as the IPL model, but includes the pro
The annex contains a summary of the methodology and level, and hence the derived performance parameters waiting and residence times at each priority process, and individual item to be served through tasks, from the first external message to the module until the last task execution before leaving the module. We call "sequence model" to this degree of representation.

(b) The microcode tasks, short and long message input queues and work queues are fully represented with their priority within the process and between processes as in Figure 3, and there is a general monitor responsible for finding the first task to be executed.

(c) The external module responses, e.g., due to access to the node data base, are represented by means of a response time distribution obtained from a separate study.

The results obtained by means of the sequence model are the average values already mentioned, and others like the netgram crossing delay (from entry of the first task at the PSM to departure of the last task) in average and distribution, as well as any other stream of tasks that could be of interest for evaluation of grade of service parameters.

6. RESULTS

Extensive results were obtained from the applications to the PSM module during the optimization phase. The following brief set of results corresponds to an application of the models to an intermediate phase before final optimization.

6.1 Comparison of the Different Models

Selecting the results provided by the three models, i.e. average values, Figures 5 and 6 give the delay of tasks in order of their priority. The importance of the priority representation for a detailed analysis of the various packet switching functions can be obtained from both figures.

The lack of feedbacks representation produces errors at all the margins of load (of the order of 100% in the IPL model). Therefore the feedbacks have an important impact on the delays and cannot be neglected in a performance estimate. The effect of lack of priority details representation varies with the priority level and the load margin. We have differences of 12% at the low priority task and 60% at a high priority task between both feedback models. The IPL model produces cumulative errors by both lack of representation up to 300% in a high non-preemptive priority. In our example the lack of priority details produces reverse sense of deviation at high and low priority levels.

If we restrict the comparison range to low and medium loads, both models (the analytical considering feedback and the sequence simulation) agree with regard to delay results. Nevertheless, for analysis of heavy loads and for optimization purposes where the detailed impact of the monitor needs to be investigated, the more accurate simulation model should be used.

6.2 Effect of Traffic Class

The traffic mix of interactive and batch classes
produces a change in the relative load per task and in the feedback between queues. Figure 7 illustrates the effect of the interactive/batch mix in the average routing task delay for a packet, as a function of the PSM load. Pure batch traffic produces more relative load on the high priority tasks, and as a consequence produces higher increases of delay than interactive traffic.

Figure 8 shows the distribution of delay for two typical sequences: the sequence associated with a call request packet and the one for a data packet in the transfer phase. The sequence of the call request has greater delays than the one of data packets, due to the fact that it contains more tasks and some of them require more processing time. This result implies the convenience (at least for models at higher levels in the proposed hierarchy) of differentiation between these two packet types (Figure 2).

Figure 9 shows the net user throughput allowed for each value associated with the 95th percentile of the data packet delay distribution. The impact of the interactive/batch class relation is very important.

6.3 Processing versus Transmission Delays

Table 1 compares processing delays and transmission delays at nominal loads (60%) for the PSM under analysis and a typical link of 64 kb/s. Three typical packet types in processing and transmission are considered:

- Call request packet as representative of the packets associated to the establishment and disconnection phases;
- Data packet of 128 octects as representative of the data transfer phase;
- Isolated acknowledge as the shortest packet within a network.

The values show that for grade of service evaluation purposes the processing with regard to the transmission component cannot be neglected; furthermore, for a packet in the establishment or disconnection phase or an acknowledge, the processing is more important.

<table>
<thead>
<tr>
<th>PACKET TYPE</th>
<th>PROCESS DELAYS</th>
<th>WAITING FOR TRANSMISSION</th>
<th>TRANSMISSION</th>
</tr>
</thead>
<tbody>
<tr>
<td>CALL REQUEST</td>
<td>33.0 ms.</td>
<td>5.44 ms.</td>
<td>3.62 ms.</td>
</tr>
<tr>
<td>DATA PACKET</td>
<td>9.5</td>
<td>24.9</td>
<td>16.6</td>
</tr>
<tr>
<td>ACKNOWLEDGE</td>
<td>9.5</td>
<td>1.5</td>
<td>1.0</td>
</tr>
</tbody>
</table>

Table 1: Processing vs. Transmission Delays
7. CONCLUSIONS

The modeling of a packet switching network by four hierarchical levels has been proposed and the most basic element of the first hierarchical level, the packet switching module, has been analyzed using both analytical and simulation models. The comparison of results from the different models shows the importance of the assumptions associated to priority and feedback, which need to be considered in an application of the type under analysis.

The analysis of the different classes of traffic, and the important impact of these classes on load, delays and throughput, implies a need for their consideration during the dimensioning phase. The comparison of processing times with transmission times in a hop of a typical network shows that the former cannot be neglected in an accurate study, and it is dominant for an interactive class of traffic.

The type of results obtained and their sensitivities shows that the use of these estimates on models at the higher levels of the proposed hierarchical modeling (node, user communication path and global network) will increase the accuracy of results in studies of these levels.

ANNEX

DERIVATION OF WAITING TIMES IN THE FEEDBACK ANALYTICAL MODEL

The feedback analytical model gives the waiting times of the levels of processing, taking into account the feedback between them in a probabilistic way.

Basically, the derivation of the waiting times of level i is calculated in an iterative form as a function of the waiting times of the i-1 levels with higher priority than i. This calculation is extended up to the M priority levels in the system and up to the m feedback stages since a request enters the system.

The flow or delay time $F_i$ at each priority level i is the sum of the waiting time $W_i$ and the processor residency time $R_i$. Hence:

$$F_i = W_i + R_i \quad /1/$$

By processor residency time of level i, $R_i$, we represent the elapsed time between the starting of level i execution and the end of processing, including all the interrupts due to preemption levels

Since only the highest priority level corresponding to the microcoded input handler preempt the lower levels, $R_i$ can be expressed as (13):

$$R_i = \frac{h_i}{1 - A_i} \quad /2/$$

and

$$R_i = h_i$$

where $h_i$ is the average processing time at level i and $A_i$ is the partial processor occupancy due to level i processing.

To determine $W_i$, the results of generalized busy periods are used (8/13). It can be shown (11) that:

$$W_i = \frac{B_{i1}}{1 - A_{i1}}$$

where:

$B_i$: Is the expected backlog of work found in the system by a request at the instant it arrives at the level i queue, and which has to be processed before this request.

$A_i$: Is the partial effective processor utilization induced by external arrivals generating work to be performed at higher priority than i, and which extends the initial backlog seen by a request arriving at level i.

$A_{i'}$ can be easily computed by considering the flow of requests into level higher than i: (11)

$$A_{i'} = A_i + \sum_{j=1}^{M} P_{i,j} A_{j'}, \quad /4/$$

where $P_{i,j}$ is the feedback probability of a request leaving level i to enter level j.

The main difficulty consists of determining $B_i$, since only the external arrival process is a Poisson process. $B_i$ is not the expected value of a random sampling of the backlog. Instead one has to follow the history of a request from the instant it enters the system ($t_1$) until the moment it enters the queue at level i ($t_i$).

The procedure consists of determining the backlog of work encountered by the request at time $t_1$ and adding to this the amount of work associated with external requests arriving in the interval ($t_1, t_i$) and which extends the initial backlog seen by the request at time $t_1$.

Figures A.1 and A.2 represent the two different histories of flow encountered in the study.

$$F_i = W_i + R_i \quad /1/$$

Figure A.1. The request is a first IQJ entry.

Figure A.2. The request is a second IQJ entry.

$$W_i = \frac{B_{i1}}{1 - A_{i1}} \quad /3/$$

where:

$t_k$ : is the instant at which the request enters level k and waits.

$t_k'$ : is the instant at which the request begins to be served.
\( W_i(m,j) \): is average waiting time of a request at level \( i \) being an \( m \) IJQ entry coming from level \( j \).

Introducing the variable \( B_i(m,j) \) by parallelism with \( W_i(m,j) \), we obtain:

\[
B_i = \sum_{m} \sum_{j} B_i(m,j) \cdot \text{Prob}(\text{the request arrives to level } i \text{ from level } j \text{ and is an } m \text{ IJQ entry}) \quad /5/
\]

and substituting /4/ and /5/ in /3/ the average waiting time at level \( i \) can be obtained.

Due to preemption of level \( 1 \) the obtained expected value of its waiting time coincides with the average waiting time of a M/G/1 queue (12).

- **Expected value of the waiting time at level 2**

The backlog of work found by a request in the moment of arriving at level 2 is:

\[
B_i = A_i W_2 + F_1 (A_1 + A_2) + \sum_{k=2}^{M} \sum_{l=1}^{K} \frac{A_k}{2} R_k \quad /6/
\]

where:

- \( A_k \) is the partial processor occupancy due to level \( k \) processing.
- \( C_k \) is the coefficient of variation of the level \( k \) execution time.
- \( A_2 W_2 \) is the work associated with all the requests queued at level 2.
- \( \sum_{k=2}^{M} \sum_{l=1}^{K} \frac{A_k}{2} R_k \) is the residual time of any request in service in the moment of arrival into the system that has to be finished before the level 2 processing.
- \( F_1 A_2 \) is the work associated with the request queued at level 1 in the moment of arrival that has to be executed at level 2 before the request under study.
- \( F_1 A_1 \) is the work associated at level 1 of the requests arriving in the interval \((t_1,t_2)\) of Figures A.1 and A.2.

Applying /3/ the average waiting time at level 2 is:

\[
W_2 = \frac{1}{1-A_1-A_2} \left[ F_1 (A_1 + A_2) + \sum_{k=2}^{M} \frac{1+C_k^2}{2} A_k R_k \right] \quad /7/
\]

- **Expected value of waiting time at Level i**

Because of the characteristics of the system, a request can be queued in an input job queue (IJQ) in a first entry coming from the level 2 or in a second entry coming from any other IJQ. Applying this to /5/ we have:

\[
B_i = B_i(1,2) \cdot \text{Prob}(\text{the request arrives at level } i \text{ from level } 2) + B_i(2,j) \cdot \text{Prob}(\text{the request arrives at level } i \text{ from level } j) \quad /8/
\]

We now introduce the following definitions:

- \( \lambda_i \): total arrival rate at level \( i \).
- \( \lambda_{i,j} \): arrival rate at level \( i \) due to termination of the processing at level \( j \) that generates an entry at level \( i \).

The following relation can be obtained:

\[
\lambda_i = \lambda_{i,1} + \sum_{j=3}^{M} \lambda_{i,j} = \frac{\lambda_i}{2} (P_{2,i} + \sum_{j=3}^{M} P_{2,j} P_{j,1}) \quad /9/
\]

and in relation with /8/ and /3/ we obtain:

\[
W_i = \sum_{i=1}^{2} \left[ \frac{\lambda_i}{2} (P_{2,i} + \sum_{j=3}^{M} P_{2,j} P_{j,1}) \right] \quad /10/
\]

Thus the problem to calculate \( W_i \) is twofold in the calculation of \( W_i(1,2) \) and \( W_i(2,j) \).

For the calculation of the backlog of work found by any request that enters at level \( i \) (i.e. \( B_i(1,2) \) and \( B_i(2,j) \)) we will follow and generalize the steps followed for the calculation of the backlog of levels 1 and 2 that can be summarized thus: a) The initial work component or the work associated with all the requests existing in the system at the moment of the external arrival \( (t_1 \text{ in Figures A.1 and A.2}) \) that has to be executed before the one under study (tagged request). b) The feedback of the initial work component or the work associated to all the possibilities of feedback of the requests existing at time \( t_1 \) that will have to be executed before the tagged request. c) The work caused by the external arrivals component, or by the requests that enter the system between \( t_1 \) and \( t_2 \) (Figures A.1 and A.2).

**Expected value of \( W_i(1,2) \):**

The backlog of work found by a request that arrives at level \( i \) in a first IJQ entry, i.e. \( i \geq 3 \), is:

\[
B_i(1,2) = \sum_{k=3}^{M} \frac{1}{A_k} W_k + \sum_{k=3}^{M} \sum_{l=1}^{K} \frac{1}{A_k} P_{2,k} h_k + \sum_{k=3}^{M} \sum_{l=1}^{K} \frac{1}{A_k} P_{2,k} P_{k,l} h_k \quad /11/
\]

where the first term corresponds to the initial work component, the second, third and the fourth are the feedback of the initial work component and the fifth is the work of the external arrivals component.

Grouping all the components that are independent of \( W_i \) and \( W_i(1,2) \), calling that value \( B_i/2 \) and applying /3/, the expected value of \( W_i(1,2) \) will be:
The backlog of work found by a request that arrives at level $i$, being a feedback from a level $j, i,j \geq 3$, will depend on the relative priority between $i$ and $j$, and taking into account the components of the backlog already defined, $B_1(2,j)$ is:

\[
B_1(2,j) = \sum_{k=j+1}^{M} \frac{A_k(W_k + \sum_{k=j+1}^{M} \lambda_k P_k, h_k, 1) + \sum_{k=j+1}^{M} \lambda_k P_k, h_k, 1)}{1-A_i - \sum_{k=j+1}^{M} \lambda_1, 2, P_1, 1, h_1} + \sum_{j=3}^{i-1} \frac{\lambda_1, 2, P_1, 1, h_1}{1-A_i - \sum_{j=3}^{i-1} \lambda_1, 2, P_1, 1, h_1} + \sum_{j=3}^{i-1} \frac{\lambda_1, 2, P_1, 1, h_1}{1-A_i - \sum_{j=3}^{i-1} \lambda_1, 2, P_1, 1, h_1}
\]

and:

\[
H_i = \frac{\lambda_1, 2, P_1, 1, h_1}{1-A_i - \sum_{j=3}^{i-1} \lambda_1, 2, P_1, 1, h_1}
\]

**REFERENCES**


