An analysis of processor load control in SPC systems

Jila SERAJ

Aktiebolaget LM Ericsson
126 25 Stockholm, Sweden

ABSTRACT

This paper treats a processor load control method and its properties in detail. Besides it discusses which factors affect the behaviour of this method, and by computer simulations shows that the load profile of calls arriving to the controlled system has a significant influence on the behaviour of this and other similar load control methods.

1. INTRODUCTION

In the evolving ISDN different traffic types are going to be carried by the same network and switching equipment.

Data communication utilizes a wide range of data speeds and, according to previous studies of data traffic characteristics, has different traffic profiles. At the same time users have very different demands on service quality for their applications. The ISDN switching equipment should thus handle a wide range of telematic services, giving satisfactory service to them all. This implies that SPC systems used for ISDN applications will have to meet harder requirements on system characteristics, such as overload protection.

An ideal load control method should have the following properties:

a) It should not disturb the normal function of the SPC system as long as the offered load is below the maximum SPC capacity
b) It should protect the system against overload, filled buffers and long delays
c) If a call is accepted it should receive a satisfactory service
d) It must be easy to implement
e) It must be able to handle the sudden jumps in the offered traffic, i.e. must not reject the calls if the offered traffic suddenly decreases and must not accept too many calls if the offered traffic suddenly increases

Two previous studies; ref. 1 and a not yet published study done by I. Anderssen and B. Nilsson at Ericsson; of two different processor load control methods; namely the LOAS method used in AXE telephone exchanges and the "ticket" method used in AXB data switches gave the following result. The ticket method satisfies all the properties listed above when it is applied on the AXB systems while the LOAS method is more suitable for AXE systems. These two studies have been the background of this work. With experiences from these two studies the author proposed that the goodness of a processor load control method is dependent on some factors among them the load profile of the calls handled by the system.

In the following we give a short description of the ticket method, describe a telephone call's and a circuit switched data call's typical load profile, describe a simplified model of the ticket method and present the results of a computer simulation of this model. In the last chapter conclusions from the simulation results are drawn.

2. THE TICKET METHOD

The principle of the ticket method is that every call arriving to the system must buy an entrance ticket. The idea is that the SPC system shall sell as many tickets as it can handle.

The central processor unit of a SPC system usually have several priority levels, aimed for processing jobs with different priorities, like traffic handling, operator commands, I/O, operation and maintenance routines and so on.

Speaking in AX systems terminology, traffic handling jobs are executed in level B, I/O and operator commands in level C, operation and maintenance in level D. Levels above level B are reserved for the operative systems. Figure 1 gives a functional picture of AX systems which interests us in this paper (for more detailed information about AX systems see ref. 3).
Arrival of a job to JBB interrupts execution of jobs in Level C and D.

To understand the ticket method we have to make some definitions.

- **Call queue**, a queue containing incoming calls waiting for a ticket.

- **Ticket queue**, a queue containing available tickets for arriving calls.

- **Ticket-coupon**, CPU marks used tickets by putting a coupon on them.

- **Ticket-coupon queue**, a queue for used tickets.

- **JBC-basket**, a waste basket for coupons torn from ticket-coupons.

- **JBC-basket queue**, a queue containing empty JBC-baskets.

- **JBC-basket-coupon**, CPU marks full JBC-baskets by putting a coupon on them.

- **JBC-basket-coupon queue**, a queue containing full JBC-basket.

- **JBD-basket**, a waste basket for coupons torn from JBC-basket-coupons.

- **JBD-basket queue**, a queue containing empty JBD-baskets.

An arriving call queues in the call queue if there are no tickets available in the ticket queue, otherwise it takes a ticket and continues to JBB. The call waits in the JBB for being served by the CPU. After the first visit in the CPU it leaves its ticket; this call is now free to continue through the system. The CPU puts a coupon on this used ticket and put it in the ticket-coupon queue.
If there is a JBC-basket available, the CPU tears off the coupon from the ticket-coupon, puts the coupon in the JBC-basket and sends back the ticket to the ticket queue.

When a JBC-basket becomes full, the CPU puts a coupon on it and puts it in the JBC-basket-coupon queue.

If there is any JBD-basket available, the CPU tears off the coupon from the JBC-basket-coupon, sends the basket to JBC and puts the coupon in the JBD-basket. When a JBD-basket becomes full, the CPU sends it to JBD.

A full JBC-basket in JBC becomes available when the CPU has executed all the jobs in front of that basket in the JBC. In that case CPU empties the JBC-basket and puts it back in the JBC-basket queue. It works exactly the same way for a full JBD-basket in JBD.

3. A TELEPHONE CALL'S LOAD PROFILE

A telephone call initiates when a subscriber lifts his handset. The SPC recognizes the call attempt and sends a dial tone to the subscriber, which causes a rather short job. After hearing the dial tone the subscriber dials the digits. Arrival of each digit to the SPC causes a short job to receive the digit and store it in some register. After receiving all (or may be 3 or 4 digits) the SPC analyses the received number to choose a path through the group switch and sends ring signal to the called subscriber, and ring tone to the calling subscriber.

Analysing the called subscriber's number, choosing a path and sending of tones and signals causes a rather long job. After sending the tones nothing happens until the called subscriber answers the phone. When the SPC system recognizes that the called subscriber has lifted his handset it connects the two subscribers and stops the sending of tones, which means a rather short job.

When one of the subscribers puts his handset back, the SPC system gets a lot to do, like releasing of the lines, registers, path and so on (ref. 4 and 6). Diagram 1 shows an example of the history of the load offered by a local call.

The times given on the time axis are mean values of time elapsed since the call attempt has been recognized in the SPC system.

4. A CIRCUIT SWITCHED DATA CALL'S LOAD PROFILE

A typical call usually begins with a "call request" signal sent by a data terminal. When the SPC recognizes "call request" it reserves registers for storing the incoming information and sends a "proceed to select" signal to the calling data terminal, which responds by sending the necessary information for establishing the call. Now the SPC system has to analyse all the information at once and to choose a path through the group switch and send a signal to the called data terminal. The called terminal responds the SPC by sending an acceptance signal to it. At this point the SPC system has to disconnect the subscribers, release the path, registers and so on. Diagram 2 shows an example of the history of the load offered by a call in 12 kb/s class (ref. 5).

The times given on the time axis are mean values of the time elapsed since the call attempt has been recognized in the SPC system.
5. PROPERTIES OF THE TICKET METHOD

Studies of the ticket method (both very detailed simulations and measurements) have shown that the ticket method is satisfying all of the properties mentioned in chapter 1 when it is applied on the AXB systems (switching nodes in the circuit switched data networks). The traffic studied in chapter 4 is an example of the traffic types handled by AXB systems. Later studies have shown that this load control method does not satisfy all of the above mentioned properties when it is applied to the AXE systems (telephone exchanges). This amazing fact is worth a deeper study to find out which factor/factors affect the performance of the ticket method.

5.1 Proposal

Studying differences between an AXE and an AXB exchange one finds that the central processor structure of AXE and AXB exchanges are quite similar, so that the main factor can not be the exchanges themselves. In ref. 1 the author makes the proposal that the determining factor in performance of the ticket method is the load profile of the calls. To explain the idea more clearly let us look at diagram 1.

Assume a telephone call generates x time units of processor load. When a telephone call arrives to an exchange it generates a processor load less than 20% of x. This leads to dial tone sending to the subscriber. The subscriber reacts to the dial tone and dials the first digit (or pushes the first button), which can take about 2 seconds, a time which is rather long for a SPC exchange. Receiving of each digit causes a load less than 2% of x.

After receiving all the digits which can take about 6 or 7 seconds the exchange analyses the dialed number, chooses the path, sends ring tone and so on. This causes a load about 35% of x.

When the called subscriber answers, the SPC has a rather short job for through-connection. While subscribers are connected SPC only supervises the connection and awaits a disconnection request. A disconnection request causes a great deal of job for clearing of the path, releasing the registers and so on. This last job is also about 35% of x. Thus the load of the CPU of the SPC system caused by a telephone call is spread over a very long time.

Now, if we look at the diagram 2 we will find that after less than 0.6 seconds passed from call request recognition, more than 70% of the total load caused by this call have already burdened the CPU of the SPC system. And, after less than 3.5 seconds this call has dissappeared from the whole system.

In other words if the load of calls are concentrated at the arrival time, the ticket method works satisfactorily. But, if the load is spread in time this load profile will affect the ticket method's performance in a negative sense.

To check whether this idea is correct the following simulation model has been considered.

5.2 Simulation model

A central processor with two priority levels B and C has been considered. Level B has strict priority (preemptive resume) over level C.

Calls arrive to this system according to a Poisson process. Service times at each visit to the CPU are independent and exponentially distributed. The ticket method with only two levels is implemented in the model (see figure 3). The capacity of the call queue is 60 calls.

![Simulation model diagram](image-url)
There are always jobs moving around on the C level and also other jobs arrive to the C level at a rather low rate.

To check the hypothesis that the load profile is the determining factor for the performance of the ticket method two options of this model are considered.

Option 1. After having received service in CPU calls will return to JBB with probability $\alpha$ and leave the system with probability $1-\alpha$.

\[ \text{Call queue} \xrightarrow{\ldots} \text{JBB} \xrightarrow{\ldots} \text{CP} \xrightarrow{\alpha} \xrightarrow{1-\alpha} \]

Figure 4. Option 1

Option 2. Calls return to JBB after having received service in CPU with probability $\beta$ or join an outer delay process with probability $\gamma$, or leave the system with probability $1-\alpha$. $\alpha = \beta + \gamma$. The outer delays are independent and exponentially distributed. Calls return to JBB after having been served in the outer delay process.

\[ \text{Call queue} \xrightarrow{\ldots} \text{JBB} \xrightarrow{\ldots} \text{CP} \xrightarrow{\beta} \xrightarrow{\gamma} \xrightarrow{\gamma} \text{JBC} \xrightarrow{\ldots} \]

\[ \alpha = \beta + \gamma \]

Figure 5. Option 2

The only difference between these two options is the load profile. In option 2 a call's load is spread over time (outer delays), while in option 1 a call's load arrives with the call itself.

5.3 Results

In the simulation the following values have been used. Mean number of feedbacks is fixed to 50 in option 1. In option 2 the mean number of feedbacks via outer delays is fixed to 10 and mean number of direct feedbacks to 40. Mean outer delay time is 2 seconds. Mean service time in the CPU is 0.4 ms.

3 jobs move around in JBC, service times being independent and exponentially distributed with mean 0.1 ms. Operator commands arrive according to a Poisson process with a rate of 10 commands/second. Each operator command job visits the CPU in the mean 3 times before it leaves the system. Service times at each visit are independent and exponentially distributed with mean 0.1 ms.

The offered calling rate increases from 10 calls/second by 10 calls/second every 10 seconds up to 100 calls/second, and then decreases by 10 calls/second every 10 second down to 10 calls/second.

The choice of this model is motivated by properties a and e in section 2.

The goal of these simulations has been to understand the behaviour of the ticket method rather than to find mean values for performance parameters of the system with good confidence intervals (which is the case in many simulation studies).

Thus statistics has been taken at the end of every 10 seconds interval. In the following diagrams statistics are shown over these intervals to demonstrate effect of changes in call intensity.

Each option has been run several times. In each run of an option certain random sequences are generated. Identically the same sequences are then used for a run of the other option. Therefore the measured values are directly comparable.

The following diagrams are examples of two comparable runs. For the sake of space one can not present the resulting diagrams from all runs. But it is important to point out that the results from other runs are very similar to those presented here.
Diagram 3. Offered calling rate

Diagram 4. Maximum JBB burst period time (ms)

Diagram 5. Maximum JBB waiting time (ms)

Diagram 6. Mean JBB waiting time (ms)

Diagram 7. Maximum JBC waiting time (ms)

Diagram 8. Mean JBC waiting time (ms)

Option 1

Option 2
The above diagrams strongly support the idea that the load profile of the calls has a very strong influence on the behaviour of the system. It should be mentioned here that simulations have been run for other options than those presented here. For example number of tickets, number of baskets, capacity of each basket, mean service time, etc., have been varied within reasonable limits to see their effect on the performance of the studied system. Results show that these variations do not affect the behaviour of the system dramatically.

6. CONCLUSION

We have shown that a load control method's goodness can be highly dependent on the load profile of the offered traffic. The particular method we have studied here sells tickets for the calls it think it can handle. But when the real load of the call does not arrive until late after the acceptance of the call, then this method is not able to adjust its call acceptance to the offered load. In other words the ticket method controls the call acceptance and therefore the load instantaneously. This means that this method is not appropriate for traffic types which do not generate load upon the arrival.

REFERENCES