OVERLOAD CONTROL IN A DISTRIBUTED SYSTEM

SABOURIN Thierry  Alcatel CIT FRANCE
FICHE Georges  Alcatel CIT FRANCE
LIGEOUR Michel  Alcatel CIT FRANCE

This paper deals with overload control in a distributed switching system where priorities have to be allocated to certain types of calls (for instance the urgent calls with short dialling numbers).

The study of this problem is made by means of simulations and of a theoretical method. This method, though simple, provides both a means to predict the behaviour of the system and a help to interpret the simulation results.

The results show that the effectiveness of an overload control mechanism depends highly on the relative cost (in term of processor utilization rate) of the rejected call. An emphasis is put on the risk resulting from the allocation of a high priority to calls whose identification is expensive.

We finally suggest an organization for the overload control in a distributed system and also an evaluation of the effectiveness of this organization under the different types of overloads which can occur on such a system.

1 INTRODUCTION

The use of effective overload control methods is a determining factor concerning the guarantee that a system will continue to operate normally when submitted to severe overload conditions. This is however a sensitive problem to be solved in so far as it is necessary to make a discrimination between the different types of calls in order to reject firstly the calls having a lower priority. This discrimination entails a significant cost for the rejected call.

This overload control problem is even more sensitive when it applies to a distributed system because the information is not centralized and the overloads may have different natures (local, general). As a matter of fact, the following choices must be made:

1) the type of overload control mechanism to be implemented (call counting procedure, measurement on utilization rates, etc.),
2) the values assigned to the parameters of this mechanism (measurement periods, thresholds from which the different overload states are detected, etc.),
3) the nature of this mechanism (distributed or centralized),
4) the physical location of these mechanisms (connection units, main control processor, service processors),
5) the priority order to be possibly allocated to the different types of calls.

The present study concentrates on these points. Several overload control mechanisms and their respective effectiveness are analyzed:

- using a theoretical method,
- using simulations.

2 PROBLEM FORMULATION

2.1 THE DISTRIBUTED SYSTEM

The system studied can be assimilated to the simplified system below:

![Figure 1: Simplified system model](image-url)
In that way, the system is divided into three fundamental parts:

1) Connection Units: the CUs are processors which connect the subscribers,

2) Main Processors: the MPs are logical machines (processors) which supervise the call treatments,

3) Service Processors: the other logical machines (translators, charging units, etc.).

2.2 THE PRIORITY PROBLEM

A few national and also international standards require a priority treatment for certain types of calls: incoming calls, urgent calls (short dialling numbers) and the calls coming from essential lines.

Consequently it is necessary for the exchange to dedicate a certain time to the identification of the calls so as to decide their treatment in case of overloads. As a consequence, the cost of a rejected call cannot be neglected in relation to the cost of an accepted call.

This type of overload control is named intelligent and within this context we execute a filtering of the calls.

The behaviour of an overload controlling system submitted to various overloads can be characterized by the figure 2.

![Ideal and practical cases of overload controlling system](image)

The ideal case corresponds to the case where the cost of a rejected (filtered) call is null. If it were the case the system could have, whatever the intensity of the traffic offered to it, its processed load maintained to a given threshold value.

In the practical case, the cost of a filtered call is significant (e.g. 30% of an accepted call). When the offered traffic increases, the system executes a growing number of overload control treatments to the detriment of the normal call handling ones.

The important cost of the filtered calls (30 to 50% of the total cost of an accepted call), depending on the resources used: main processors or connection units) imposes a flat rejection (without filtering, that is leading to a very low cost) in the case of very heavy overloads. The problem consists then in allocating judiciously these two types of mechanism (intelligent and categorical rejection) in the system.

2.3 A DISTRIBUTED OVERLOAD CONTROL

The problem of the overload control is more complex in a distributed system than in a centralized system in the sense that at the same moment, the overload states of the processors can be different. Indeed, the system can be submitted to two types of overload:

- general overloads (traffic evenly shared among all connection units) which concern the command units,
- local overloads (traffic concentrated on a few connection units) which concern the connection units.

This simple distinction justifies the interest in distributing the processes executing the overload control in the command units and in the connection units.

The overload control mechanisms implemented within the system and which constitute a basis for the study of the overload control in a distributed system are shared as follows:

- **Command units**
  - **Main processor:** it is logic to place there an intelligent mechanism, this unit having an access to all the information concerning the subscribers. Moreover, as the load of the main processor reflects the general load of the system command units, it is reasonable to create discriminations between the calls on the basis of observations made on that load.
  - **Service processors:** practically, they are always less loaded than the main processors and the overload control processes operated by the latter must be sufficient to protect them. If, despite all that, this is not the case, this means that an event has occurred which is severe enough to impose immediate rejection. We then choose to introduce flat rejection mechanisms in the processors of that type. Moreover, these processors do not know for which call they are working. To introduce there an intelligent rejection should thus be too expensive.

- **Connection units**

It has been decided to introduce a dual mechanism both intelligent and categorical, that is:
Intelligent: a slight increase in load on some connection units, representing no risk either for them or for the main processors, doesn't justify immediate call rejection. For that reason, a filtering is executed in case of non-important overloads (the control is given to the main processor which determines whether the call being processed can be rejected or not).

Categorical: if the load on the connection units becomes important, a flat rejection is undertaken. We will see that, at that level, this type of rejection is interesting because its cost is null for the command units and negligible for the connection units.

Remark: flat rejection is also introduced in the main processors. We do not consider this implementation because it only occurs under circumstances which do not belong to the domain of our study.

2.4 THE OBJECTIVES OF THE STUDY

On the basis of the organization presented above, the objectives of this paper are:

- to determine the most effective overload control mechanisms within the various units concerned,
- to choose the most profitable type of priority,
- to ensure the consistency of all those mechanisms, in accordance with the type of overload to which the system is submitted.

Remark: we will only consider the case where the MPs and the CUs are able to execute overload controls, the service processors doing it only in extreme cases which are not discussed here.

3 THEORETICAL APPROACH OF THE PROBLEM

The object of this part is to suggest a simple theoretical method based on a set of simplifying hypotheses and capable of forecasting the behaviour of the system when it is submitted to overload conditions. The term "behaviour" will be in turn characterized by the two following terms:

- total overload,
- handled overload.

We will only consider a stationary state of the system. Our approach will be deterministic: we will assimilate in our calculations the instantaneous value of the traffic processed to its mean value.

3.1 THE PARAMETERS

The relative costs The study focuses on the cost of calls in relation to the system. In order to simplify the different formulations we will consider that in average, an accepted call is assigned the value 1, the other costs being used with a relative value in relation to that one.

Let us assume that the traffic offered to the system is characterized by the following distribution of calls.

n types of calls : \( A_1, A_2, \ldots, A_n \)

Ratio for each type of call: \( P_1, P_2, \ldots, P_n \)

with \( \Sigma P_i = 1 \)

Cost of these calls (ms) concerning a unit \( P \):

Accepted : \( C_{a1}, C_{a2}, \ldots, C_{an} \)

Rejected : \( C_{r1}, C_{r2}, \ldots, C_{rn} \)

Mean cost for an accepted call:

\( \text{Cost}_{am} = \Sigma P_i \cdot C_{ai} \)

- Relative cost of an accepted call \( A_i \):

\( C_{ai} = C_{ai} / \text{Cost}_{am} \)

- Relative cost of a rejected call \( A_i \):

\( C_{ri} = C_{ri} / \text{Cost}_{am} \)

4 types of loads are considered:

- the offered load \( (L_0) \), is the amount of traffic offered to the system, if we note:

  • \( \text{Ncoms} \): number of calls offered per ms to the system,
  • \( \text{Cost}_{am} \): mean cost of an accepted call (ms) and concerning a processor \( P \),
  • \( \text{NbP} \): number of \( P \) copies,

  We obtain thus the load \( L_0 \) offered, as seen from \( P \):

\( L_0 = (\text{Ncoms}/\text{NbP}) \cdot \text{Cacms}, \)

- the total load \( (L_t) \), is the utilization rate for units \( P \),
- the load handled \( (L_h) \), is the load resulting from the accepted calls,
- the load rejected \( (L_r) \), is the load resulting from the rejected calls, \( (L_t = L_h + L_r) \).

3.2 BASIC RELATIONS

For a load \( L_0 \) offered to the system, the distribution of the calls accepted and rejected is noted as follows:

For a given type of call \( A_i \): \( T_{ai} \) are accepted \( T_{ri} \) are rejected

\( T_{ai} \) and \( T_{ri} \) represent rates and \( T_{ai} + T_{ri} = 1 \)

Relation between the offered load and the total load:

\( L_t = L_0 \cdot \Sigma P_i \cdot (T_{ai} \cdot C_{ai} + T_{ri} \cdot C_{ri}) \)
Relation between the offered load and the handled load
\[ L_h = \sum_{i=1}^{n} P_i . T_a_i . C_a_i \]

Relation between the offered load and the rejected load:
\[ L_r = \sum_{i=1}^{n} P_i . T_r_i . C_r_i \]

3.3 THE HYPOTHESES

The overload control mechanisms activation generally depends on estimates carried out on the traffic processed by the exchange. These estimates are evaluated by performing measurements:

- overall measurements: number of calls being processed, overall load on the internal switching matrix, efficiency rate of the exchange,
- individual measurements: number of circuits busy, number of messages in the waiting queues, utilization rate for the processors.

The analysis of the measurement results, performed to decide whether an overload occurs or not, consists in comparing the estimate with one or several thresholds (\( S_1, \ldots S_n \)). If there are several thresholds, each one corresponds to a particular level of overload control.

The basic hypothesis is then the following: as long as the call rejection permits it, the system is able to keep its total load at \( S_1 \), otherwise at \( S_2 \), etc.

Formally, the hypothesis on which our study will be based is:

- when the system does not reject any call (by means of overload control): \( L_t = L_o \),
- when the system executes an overload control procedure (rejects some calls) and when the preservation of \( S_1 \) is possible: \( L_t = S_1 \),
- etc.

Consequently, although the overload control method is a complete automatic control mechanism, we suppose that the amplitude of the oscillations induced by this method are negligible.

4. INFLUENCE OF THE NATURE OF THE OVERLOAD CONTROL MECHANISM

Within this section, we perform by means of simulations a comparison between the two following mechanisms:

- mechanism based on counting procedures on calls.

For each of them, we compare the results from the simulations with the theoretical curves and then we perform the real evaluation of the mechanism.

4.1 MECHANISM BASED ON PROCESSOR UTILIZATION RATE MEASUREMENTS

The mechanism based on the measurement of the utilization rate of the processor is the following. The exact load is measured periodically. When it exceeds a first threshold \( S_1 \), there is a rejection of all the outgoing calls which are not urgent during the complete duration of the following period and when it exceeds the second threshold \( S_2 \) there is an additional rejection of all incoming calls which are not urgent.

The practical curves are obtained with the following parameters:

- period used for load measurements: 100 ms,
- Thresholds fixed to 0.8 and 0.9.

![Figure 3: MP overload control based on utilization rate measurements](image)

The analysis of these curves shows that the system performs the overload control too early and that the total load is not maintained to the threshold value.

This is due to the fact that:

- on the one hand all the overloads are detected too late, the utilization rate represents the consequence of a traffic overload offered to the system (many calls have already been accumulated by the system which must process them anyway), the reaction to the overload is too slow,
- on the other hand, the acceptance and rejection process is not symmetric (acceptance quicker than the service), which leads to an increase in the mean load value.

4.2 MECHANISM BASED ON CALL COUNTING PROCEDURES

The principle of this mechanism is as follows: each MP performs the counting of the number of
calls offered and of the number of calls accepted and derives from them an estimate of the total load. If this estimate exceeds a first overload threshold (S1) or a second overload threshold (S2), it positions its overload indicator to 1 or 2.

If an MP is in overload state 1 it only rejects the non-urgent OUTGOING calls. If it is in overload state 2, it rejects all the non-urgent calls.

The estimate is performed by taking into account both the mean cost of an accepted call (CA) and the mean cost of a rejected call (CR) independently of the other call characteristics. It is calculated by using a slipping window (integration on the last N measurements).

The formula applied is as follows:

\[
\text{Estimate} = \frac{S\text{-OFF} \times CR + S\text{-ACC} \times (CA - CR)}{N \times \text{PERIOD}}
\]

- \( S\text{-OFF} \) is the number of calls offered to the MP during \( N \times \text{PERIOD} \)
- \( S\text{-ACC} \) is the number of calls accepted by the MP during \( N \times \text{PERIOD} \)

The estimate is calculated at the end of each PERIOD.

The practical curves are obtained with the following parameters:
- load measurement period: 100 ms,
- integration window: 3 s,
- thresholds set to 0.8 and 0.9.

These curves lead us to perform the following observations:
- The system starts its overload control only when \( L_0 = 0.74 \) (theory: 0.8), which is acceptable when compared with the method based on measurements of the MP utilization rates.
- A decrease in the total load is clearly shown during slight overloads. This decrease is due to the fact that the cost of an outgoing call is higher than that of an incoming call and during small overloads the system mainly rejects outgoing calls.

- The total load is correctly maintained to the threshold value.

The load of the system is estimated from mean costs and it is integrated over a long period, which gives a better view of the real load of the system. The estimate is a forecast of the system load: this anticipation leads to a reaction which is quicker than that occurring in the case of the utilization rate measurements.

4.3 CONCLUSION

The call counting procedure is the most effective mechanism. Indeed:
- the processors are completely protected (the total load is very correctly maintained),
- the handled load has peaks only slightly clipped (the overload control does not start too early),

Moreover it has been clearly shown that the priority allocated to the incoming calls is strictly respected.

Consequently this mechanism should be retained for the processors where a mechanism designated as intelligent is desirable (MP and CU).

The mechanisms studied have as a particular feature an important relative cost for a rejected call. For ensuring a protection against very heavy overloads and in order to avoid a total collapse of the system in such circumstances, it is thus necessary to introduce a flat rejection mechanism (based on load measurements) as a complement to an intelligent mechanism (based on call counting procedures).

Finally this study enables to perform a comparison between theory and practice. Apart from critical points like threshold changes (bends on the theoretical curves), the theoretical results are close to the practical results for the call counting method. Consequently the continuation of this study will be based only on theoretical studies for obtaining the curves related to the system overload control (applications to other costs or to other thresholds).

5 BEHAVIOUR OF THE SYSTEM IN ACCORDANCE WITH PRIORITIES

5.1 TYPES OF PRIORITY CONSIDERED

Two types of priority are considered within this document:

1) The first type of priority consists in attributing an absolute priority to the urgent calls. The order followed for the call rejection is: non-urgent outgoing calls, non-urgent incoming calls, urgent calls.
2) The second type of priority consists in allocating to the non-urgent incoming calls a priority higher than that of the urgent outgoing calls. The order followed for the call rejection is then modified in the following way: non-urgent outgoing calls, urgent outgoing calls, non-urgent incoming calls, urgent incoming calls.

5.2 ANALYSIS OF THE BEHAVIOUR IN ACCORDANCE WITH THE PRIORITIES

The theoretical curves corresponding to the first and second types of priority are drawn on the following figure.

![Figure 5: Theoretical curves for the two types of priority](image)

We observe with the second type of priority (in contrast with the first one) that the system remains capable of handling a lot of calls even in the case of very heavy overloads. The unique default shown (the "low value" of the processed load corresponding to small overloads) is very reduced in the practical cases. The simulations show that there is a very significant smoothing of the theoretical curves.

We note that this simple inversion of the priorities allows us:

- to handle a greater number of calls in the case of slight or medium overloads,
- to defer considerably the collapse of the handled load.

6 BEHAVIOUR OF THE SYSTEM IN ACCORDANCE WITH THE OVERLOAD TYPE

We study now the overall behaviour of the system after the set of overload control mechanisms previously analyzed have been installed. The existence of two types of overload justified the introduction of a distributed overload control, consequently we are now going to verify the correct consistency of this control in the two following cases: general overload and local overload.

6.1 CASE OF GENERAL OVERLOADS

A general overload is evenly shared among the whole set of connection units.

The theoretical study shows clearly the importance of the ratio (psi) of the nominal loads on the MPs and CUs.

In the system studied, this ratio (psi) has a value which is near to 1.5. We show thus that for a system correctly dimensioned, general overloads are totally processed by the main processors (MP). This is consistent with the organization of the overload control within the system. The figure 6 represents the responses from the MPs and CUs in this last case.

![Figure 6: Overload control by MP and CU in case of a general overload](image)

Other values can be considered for psi. On the basis of the hypotheses concerning the relative costs taken for our system, we distinguish the following cases:

- if $1.27 < \text{psi}$: whatever load is offered to the system, the CUs will not perform an overload control since the rejection of calls executed by the MPs is sufficient to keep the CU load under the first overload control threshold,
- if $1.19 < \text{psi} < 1.27$: the first CU threshold will sometimes be exceeded but without reaching the second one, and since all the calls rejected by the CUs (outgoing calls) would also be rejected by the MPs, the MP is thus in charge of the call rejections,
- if $\text{psi} < 1.19$: the MPs and CUs perform the control simultaneously. The drawing of the theoretical curves is more delicate. Load areas exist where the MP is the only unit which performs overload control and there are other areas where the CU only executes the call rejections.

6.2 CASE OF LOCAL OVERLOADS

The local overloads correspond to overloads concentrated on one or several connection units. We assume that the system is correctly dimensioned and that the overload is directed to only one CU.

We have analyzed the behaviour of the CUs when a local overload traffic is added to the normal overall traffic, but this is not detailed in this paper due to space limitations.
This study enables us to formulate the following conclusions concerning the effectiveness of the overload control confronted with a local overload either outgoing or incoming:

- a heavy local overload is needed for starting the central overload control (MP),
- the mechanisms operating on the CU protect efficiently against the very heavy overloads (higher than 1000%),
- we show that it is important to take into account the second type of priority in order to:
  - make a clear distinction between the outgoing and the incoming calls,
  - ensure that during the local outgoing overloads the outgoing calls are rejected first and relieve the system.

Inversely, during local incoming overloads, the system is relieved by the rejection of outgoing calls.

CONCLUSIONS

We have presented a simple theoretical method which is a means for predicting the behaviour of the system under overload conditions. This method, even if it only conforms to the reality when a set of hypotheses are respected, is a basis which guides the study performed by simulation:

- it reduces the number of simulations by limiting the areas where the curve points are calculated and by giving the general shape of these curves,
- it discloses some characteristics of the overload control methods studied, such as the degree at which the priorities are respected.

The overload control mechanism based on call counting procedures is highly better than the mechanism based on the direct measurement of the processor utilization rates in the sense that it reduces the oscillatory phenomena, leading to the clipping of the load handled and to a poor respect of the priorities.

It is preferable to allocate a priority to the calls whose identification results in the lowest cost. The reversal of priorities suggested in this document leads in most cases of local overload to an increase in the load handled by the system but it gives a lower importance to the urgent calls.

The various overload control mechanisms introduced into the system are consistent. According to the type of overload, one of these mechanisms will react to ensure the security of the whole system.

Finally, this paper shows clearly that the effectiveness of an overload control mechanism highly depends on the relative cost of the calls rejected, cost which is difficult to reduce if we want to establish a discrimination between the calls. From this viewpoint, it is important to choose correctly the location of the mechanisms used to identify the calls (CU or MP) and to determine adequately the priorities. This last point should deserve to be examined in the international standards organizations.

REFERENCE DOCUMENTS

1) ROMOEUF : Modèle de régulation de charge dans un autocommutateur téléphonique à commande centralisée, annales des télécommunications, 1977.
2) FICHE, RUVEN : The E10 System functional evolution and quality of service 21 A2 ISS 84 Florence.
4) WEGMANN, HANSELKA, OEHLERICH : Adaptation of the overload regulation method STATOR to multiprocessor controls and simulation results, ITC 12, Torino, June 1-8, 1988.
6) Doshi, Heffes : Analysis of overload control schemes for a class of distributed switching machines, ITC 10, Montréal, 1983.
7) WALLSTROM : A feedback queue with overload control, ITC 10, Montréal, 1983.