ISDN Services Performance Analysis on a Multicomputer Node

Enrico Del Re and Elio Pasca

Dipartimento di Ingegneria Elettronica, Università degli Studi di Firenze
Firenze – Italy

ABSTRACT

Continuity is an important characteristic for ISDN services. A multicomputer node structure, i.e. a service unity comprised of several cooperating computers, is proposed in order to guarantee service availability even in the presence of failures.

A suitable decentralized strategy to access node's capabilities is presented, and the performance of multicomputer node with/without computers' crashes is analyzed. The most important result of the analysis is that a node, operating at small values of the utilization factor, is slightly affected by crash occurrences.

1. INTRODUCTION

ISDN is the future in communication networks and will be successful --besides the integration in one medium of every kind of information-- mainly because of the great and useful variety of services obtainable.

Today we can figure out only a small part of the services that in the near future will be available through ISDN. We know that people will do many things without moving, as today by telephone. Services coming from ISDN will be numerous and will cover a great variety of applications:

- Electronic mail,
- Information retrieval,
- Reservations (train, airplane, theatre, hospital) ...etc.

Once the use of ISDN services becomes popular, it will be difficult to live without them. Every service must guarantee continuity because users suffer from service failures, even if services failed may seem optional. People might accept a loss in performance (longer service time), but they will never accept lacking service.

A service node is studied which is comprised of many cooperating computers, i.e. a multicomputer node. This node works well even if some crashes strike the node computers; thus, continuous service will be guaranteed. Continuous node service can be represented through requested service availability. Service availability is defined as the probability that, during a fixed time interval, the system has at least one server working properly, namely that meets its specifications [4].

2. NODE STRUCTURE

A multicomputer node is a node made up of several computers which play the same role and can offer ISDN services through multiple access ports. These services can be accessed by users from different ports. Multicomputer node ISDN services can grow by means of modules as the user load increases without suffering from computer crashes, insofar it is possible to guarantee continuous service.

Fig. 1 shows a multicomputer node made up

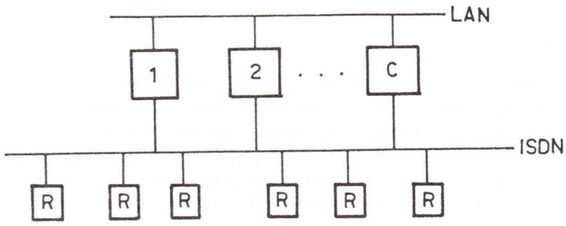

Fig. 1 - Node structure.

of C computers accessed through several ports on ISDN and connected together by LAN [5]. This connection is useful for speedy intercomputer communication and can be carried out in different ways, e.g. by ISDN too.

A user may ask for a certain service by sending a request to any computer on the node and the server requested on that computer will send back the reply.

In our scheme, there is no risk of node partition. The presence of two networks guarantees service communication between the node computers. Moreover, a multicomputer node guarantees reliability, availability and continuity of its services. These features do not refer to customer-computers (R in Fig. 1), which are not protected against crashes or failures. In the same way, the node activity and properties will not be affected by customer-computer crashes and failures.

3. NODE ACCESS STRATEGY

In general, when services working in a network environment are not replicated, customers address service requests to hosts where servers are allocated; otherwise, they address their requests to well-known master-servers. A master-server is a special server which deals with dispatching the requests to appropriate servers. In both cases, host crashes will make some or all (when master-server host crashes) services not available.

In a multicomputer node, servers are replicated on different computers; in this way, it is possible to satisfy more requests, while increasing the system availability and reliability.

If all servers (master-servers too) are replicated, the strategy to access one of the available servers, balancing the node load, has to be defined.

3.1 Three conditions

The strategy to access such replicated services must obey to the following conditions:

i) if the process representing node arrivals is a Poisson process, also single computer arrivals have to be a Poisson process;

Papers presented at ISDN Europe 86 the First Pan European Conference on ISDN, held 5-7 November, 1986 at the EWTCC Congress Centre, Basel/Switzerland. Sponsored by IGI Europe a Division of Information Gatekeepers, Inc., 214 Harvard Avenue, Boston, MA 02134 USA. Telephone: (617) 232-3111, Telex: 499-6088 fiber.

ii) every computer of the node will work in proportion to its service rate;
iii) the access algorithm will not decrease the availability in a multicomputer node.

The first condition, which simplifies the analysis, means that, if in a time interval node arrivals are uniformly distributed through the interval, also single computer arrivals are uniformly distributed, and arrival processes, regarding node and single computer, are both memoryless.

The second condition balances the load in a multicomputer node and would be better replaced by other types of access optimization [2]; however, this is not the aim of our study.

The third condition excludes algorithms which use a centralized implementation; if customers employ a centralized access strategy, a crash could affect availability and service continuity.

3.2 Strategy

Customer computers send service requests to the node which distributes them to node computers according to their service rates. If node arrivals are uniformly distributed (or the interarrival times are exponentially distributed) in a time interval t, and independent from one interval to the other, then the process is a Poisson process. Furthermore, we suppose that requests are randomly distributed among computers in proportion to service rates. If μ_i is the computer average service rate --with i = 1,2,...,C computers in the node-- the probability that any computer receives a service request is calculated as:

$$p_i = \frac{\mu_i}{\mu_N}$$

where

$$\mu_N = \sum_{i=1}^{C} \mu_i$$

is the node average service rate (Index i refers to single computer quantities, and N to node quantities).

We use $p_i = \mu_i/\mu_N$ to distribute (randomly) the requests reaching the node. The node arrival process, which is Poisson, is randomly splitted in C arrival processes, regarding single computers, which are still Poisson [3]. In this case, the access strategy will conform to conditions i) and ii), but it is not sure that it will comply with condition iii). In fact, if the device that distributes the requests is centralized, it will be sensitive to crashes.

To avoid this, customers distribute their requests among node computers.

STRATEGY: Customer computers choose their server in two steps:
1) They generate, randomly, a number between 0 and 1;
2) They select the addressed server by locating the interval corresponding to the generated number (Fig. 2).

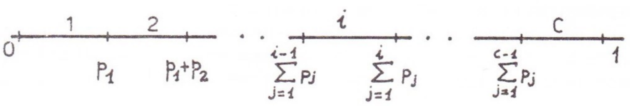

Fig. 2 - Splitting interval.

3.3 Adjustment of strategy to crashes

The strategy defined must work properly in crash conditions. When the customer system sends a request to a crashed host, it must try again after a timeout and it may happen that requests have to wait for replies longer than necessary.

A simpler way to adjust the strategy is that customers:
1) consider crashed the host that does not reply within a chosen timeout period;
2) try again the strategy described in the previous paragraph, applying it to commputers survived to crash;
3) repeat the adapted strategy (steps 1 and 2) until they receive a server reply.

This adjustment is not sufficient because we may have more than one server processing the same request. Moreover, the response time will increase dramatically with crashes, not only due to the increase in service time, but mainly to the necessity for the customer to repeat the service request after every timeout.

A better solution is to change the access strategy in the following way:
1) The customer system sends a service request to all computers through a logical address (MULTI-CAST).
2) Inside the request, the customer puts a permutation randomly chosen, of all computers in the node.
3) The request will be served by the working computer, first in the permuted list.
4) Other computers will discard the request.

Steps 3) and 4) are possible because every service computer knows the dynamic configuration of the node. A Node Manager in every computer cooperates to survey, almost instantaneously, crash conditions and recovery procedures [6].

In the present study, we will refer to the last mentioned adjustment of node access strategy.

4. NODE PERFORMANCE ANALYSIS WITHOUT CRASHES

Our aim is now to study the access characteristics to multicomputer node service from an infinite population of customers, when the node is crash free.

We assume:
1) C the number of node computers with one server for every computer;
2) computer average service time, $1/\mu_i$, exponentially distributed;
3) the set of service requests, addressed to the node, be a Poisson process with average rate λ_N;
4) to use our access strategy to split this set of requests in C sets of requests, addressed to the C computers of the node. Therefore, every computer receives a set of requests, which is a Poisson process, with average rate

$$\lambda_i = \frac{\mu_i}{\mu_N} \lambda_N \quad (1)$$

The previous assumptions ensure that every computer is a M/M/1 queue system with its own μ_i and λ_i [1], and a multicomputer node can be seen as the set of C M/M/1 queue systems (Fig. 3).

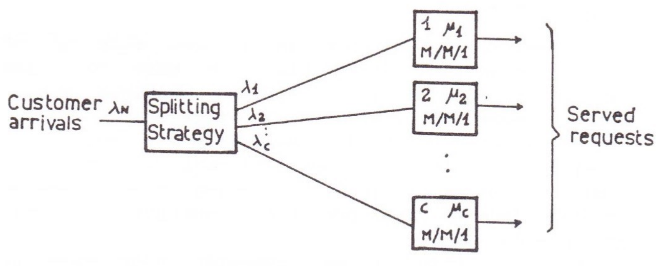

Fig. 3 - Access to node services.

4.1 Utilization factor

(2) $$\varrho_N \triangleq \frac{\lambda_N}{\mu_N} = \frac{\lambda_i}{\mu_i} = \varrho_i = \varrho$$

The node utilization factor is equal to the single computer utilization factor; this result comes directly from the splitting strategy (1).

Every computer has the same traffic intensity (utilization factor), even if they have a different average service rate.

The whole system is stable if

(3) $$\varrho_N < 1$$

that is

$$\sum_{i=1}^{C} \mu_i > \lambda_N \quad .$$

The node will work in stability conditions if the sum of single computer average service rates is greater than the arrival rate.

From relations (2) and (3), it follows the stability of single computer M/M/1 queue systems.

4.2 Average number of customers

In the node, the following relation can be applied [1]:

$$\overline{N}_N = \sum_{i=1}^{C} \overline{N}_i \quad \text{with} \quad \overline{N}_i = \frac{\varrho_i}{1-\varrho_i} = N$$

$\overline{N}_i$ is the average number of customers in a single computer, which depends on ϱ_i (utilization factor).

From relation (2), it comes that every computer has the same average number of customers, and therefore

$$\overline{N}_N = C \cdot N \quad .$$

The node average number of customers is directly proportional to the node computer number.

4.3 Average system time

By applying Little's relation [1], we find the average time spent in the system (queue plus service) for a service request that reaches the node:

$$\overline{T}_N = \frac{\overline{N}_N}{\lambda_N} = C \cdot \frac{N}{\lambda_N} = \frac{N}{\lambda_{N/C}} \quad .$$

If all C computers have the same average service rate $\mu_i = \mu$, they will serve the same average arrival rate $\lambda_i = \lambda$, and it will be

$$\frac{\lambda_N}{C} = \lambda \quad \text{and} \quad \overline{T}_N = \frac{N}{\lambda} = \frac{\overline{N}_i}{\lambda_i} = \overline{T}_i \quad .$$

The node average time will be equal to the average time spent in any single computer.

4.4 Steady-state probability of finding k customers in the node

If k_j is the number of customers for every computer,

$$k = \sum_{j=1}^{C} k_j$$

total number of customers for the node and, as k_j are random independent variables, we can calculate the steady-state probability of finding k customers, with probabilities convolution to find k_j customers in the single computers.

In the z-transform domain, the probability densities p_{i,k_j} of the single computers

$$p_{i,k_j} = (1-\varrho)\,\varrho^{k_j} \qquad [1]$$

(instead of ϱ_i, we use ϱ because all computers have the same utilization factor) give rise to the generating function

$$P_{i,k_j}(z) = \sum_{k=o}^{\infty} p_{i,k_j} \cdot z^{k_j} = \frac{1-\varrho}{1-\varrho \cdot z}$$

For the node we find the probability z-transform:

$$P_{N,k}(z) = \prod_{i=1}^{C} P_{i,k_j}(z) = (1-\varrho)^C \cdot \left(\frac{1}{1-\varrho \cdot z}\right)^C$$

and, applying the inverse-transform operation, we find the steady-state probability of the node:

$$p_{N,k} = (1-\varrho)^C \cdot \varrho^k \cdot \binom{C+k-1}{k}$$

This probability increases for lower values of ϱ, as the node computer number grows.

Figs. 4 and 5 report the behaviour of $p_{N,k}$ versus ϱ for C = 1 (single computer node) and C= = 4.

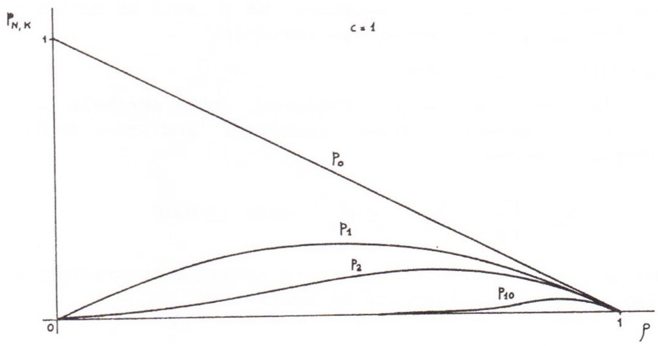

Fig. 4 - $p_{N,k}$ versus ϱ for C = 1.

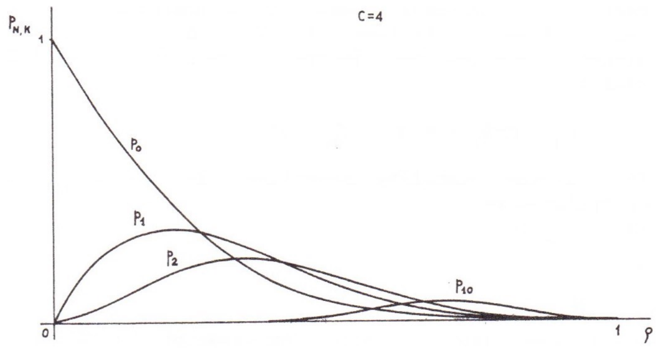

Fig. 5 - $p_{N,k}$ versus ϱ for C = 4.

5. NODE PERFORMANCE ANALYSIS WITH CRASHES

Our aim is now to study the degradation of node access performance when some computers crash. In this case, the access strategy (3.3) still allows the node to handle customer service requests. The traffic intensity λ_N which reaches the node, is not changed, while the node service rate is decreased.

In case of computer crashes, the average service rate lost by the node, is:

$$\mu_A = \sum_{i \in \mathscr{A}} \mu_i$$

with $\mathscr{A}$ the set of crashed computers. The average service rate now available, is

$$\mu'_N = \mu_N - \mu_{\mathscr{A}}$$

(the prime refers to quantities of node with crashed computers).

We define the node overload, the rate which is an assessment of the node damage due to failures.

$$\sigma = \mu_N / \mu'_N$$

We can define σ_{max} as the maximum crash conditions accepted for the node. For example, if we want to guarantee service availability, at least one node computer will survive to crash. In this case, for a node, where computers have the same service rate ($\mu_i = \mu$), σ_{max} = C.

In one node the overload σ does not vary with continuity, because every crash makes μ'_N to be decreased without continuity. However, we suppose that σ may assume any value greater than one. If we pass from one multicomputer node to all possible node configurations and then to all possible crash conditions, we see that σ may assume infinite values. For these reasons, functions of σ will be assumed as functions of a continuous variable.

5.1 Utilization factor

In the presene of failures, node arrivals are splitted between fewer computer, and so single arrivals increase

$$\lambda'_i = \frac{\mu_i}{\mu'_N} \cdot \lambda_N = \sigma \cdot \lambda_N \quad \text{with } i \in \mathscr{C}-\mathscr{A}$$

($\mathscr{C}-\mathscr{A}$ is the set of computers survived to crashes). The previous relation can be written as

$$\lambda'_i / \mu_i = \lambda_N / \mu'_N \quad .$$

This expression emphasizes an intuitive and reasonable consideration:

In a multicomputer node, the loss of service rate due to crashed computers, causes a proportional arrival rate increase to surviving computers.

The utilization factors of the node and of the single computers are increased, but they are still equal:

$$\varrho'_i = \sigma \cdot \varrho_i = \sigma \cdot \varrho_N = \varrho'_N = \varrho'$$

The system stability condition in the presence of failures is

(4) $\varrho' < 1$

or

$$\varrho < \frac{1}{\sigma}$$

In a multicomputer node, the value of the utilization factor must be chosen so that it guarantees system stability conditions in the presence of the maximum number of crashes accepted in the node. The maximum utilization factor is shown in Fig. 6 versus σ.

5.2 Average number of customers

In the presence of failures, the average number of customers in the single computers is:

$$\bar{N}'_i = \frac{\varrho'_i}{1-\varrho'_i} = \frac{\sigma\varrho_i}{1-\sigma\varrho_i} = \frac{1-\varrho}{\frac{1}{\sigma}-\varrho}\,\bar{N}_i$$

N_i grows with crash increase.

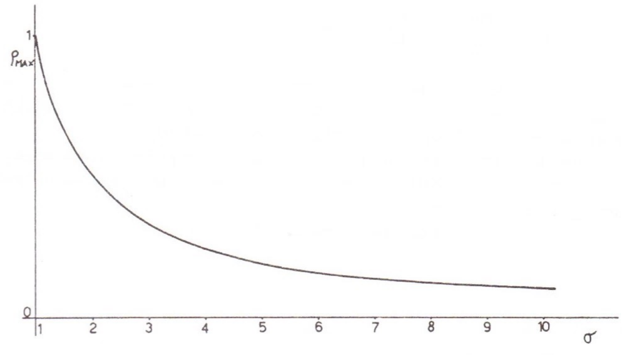

Fig. 6 - ϱ_{max} versus σ.

In the node, the average number of customers is

$$\bar{N}'_N = \sum_{i \in \mathscr{C}-\mathscr{A}} \bar{N}'_i = (C - A)\,\frac{\sigma\varrho}{1-\sigma\varrho} \quad .$$

We cannot establish a priori if the average number of customers is increased. This depends on the fact that every computer of the node has a different value of μ_i. When computers crash, the average number of customers grows in the single computers, while in the node it can decrease if crashed computers had single computer average numbers of customers lower than node's. Therefore, in a multicomputer node, performance considerations request the use of computers with the same service rate (μ_i = $= \mu$). In this case

(5) $\sigma = \dfrac{C}{C-A} \qquad \bar{N}'_N = \dfrac{C \cdot \varrho}{1-\sigma\varrho} = \dfrac{1-\varrho}{1-\sigma\varrho}\,N$

and

$$\bar{N}'_N > \bar{N}_N \quad .$$

5.3 Average system time

For every working (surviving) computer, by using Little's relation [1], we find:

(6) $\bar{T}'_i = \dfrac{\bar{N}'_i}{\lambda'_i} = \dfrac{1/\mu_i}{1-\sigma\varrho_i} = \dfrac{1-\varrho}{1-\sigma\varrho}\,\bar{T}_i$

and $\bar{T}'_i > \bar{T}_i$

For the node, we calculate:

$$\bar{T}'_N = \frac{\bar{N}'_N}{\lambda_N} = (C-A)\,\frac{\sigma/\mu_N}{1-\sigma\varrho}$$

We can repeat the same considerations (as in the preceding paragraph) applying them to the node average system time, and find that it is better if all computers have the same service rate ($\mu_i = \mu$).

In these conditions, the node average system time is:

(7) $\bar{T}'_N = \dfrac{C/\mu_N}{1-\sigma\varrho} = \dfrac{1-\varrho}{1-\sigma\varrho}\,\bar{T}_N \quad .$

Relations (5), (6) and (7) look alike (Fig. 7).

5.4 Steady-state probability of finding k customers

When the node has A crashes, the steady-state probability of finding k customers in the single computers is:

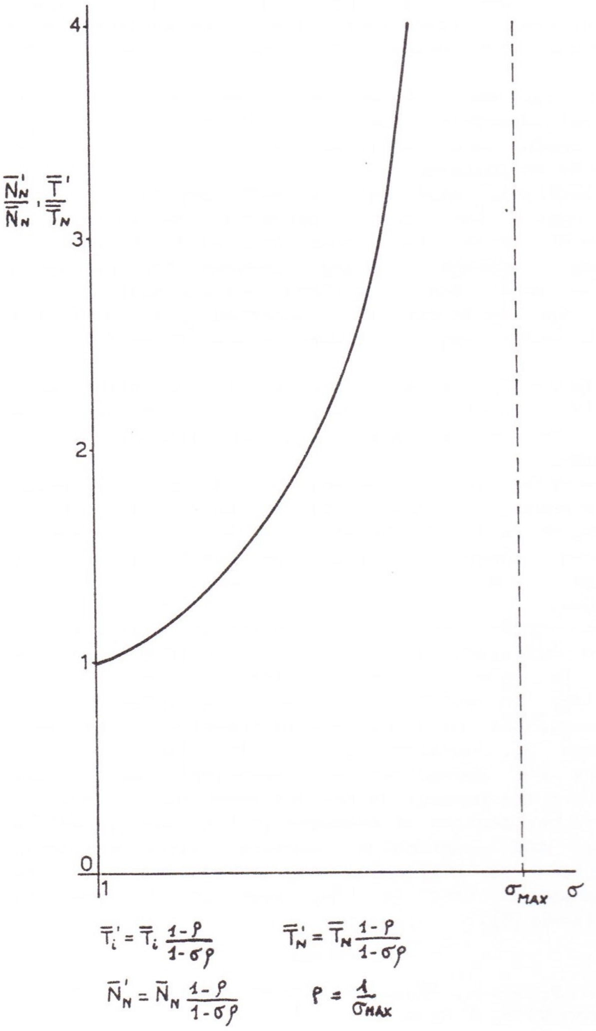

Fig. 7 - Relative behaviour of average system time (single computer and node) and node customer average number versus σ.

$$p'_{i,k} = (1-\varrho'_i)\,\varrho'^{\,k}_i = (1-\sigma\varrho)\sigma^k\cdot\varrho^k = \frac{1-\sigma\varrho}{1-\varrho}\,\sigma^k\cdot p_{i,k}$$

$p'_{i,k}$ is not necessarily greater than $p_{i,k}$: this relation depends on the respective values of ϱ, σ and k.

If we calculate the probability of finding at least k customers in the single computers, we have [1]:

$$p_i[\geqslant k] = \varrho^k$$

and

$$p'_i[\geqslant k] = \varrho'^{\,k}_i = \sigma^k\cdot\varrho^k = \sigma^k\, p_i[\geqslant k]$$

$$p'_i[\geqslant k] > p_i[\geqslant k] \quad .$$

In the presence of failures, the probability of finding at least k customers in any computer increases with failures.

For the node, we calculate:

$$p'_{N,k} = (1-\varrho'_N)^{C-A}\cdot\varrho'^{\,k}_N\cdot\binom{C-A-k-1}{k} =$$

$$= (1-\sigma\varrho)^{C-A}\cdot\sigma^k\cdot\varrho^k\binom{C-A-k-1}{k}$$

and, if all computers have the same average service rate, it becomes:

$$p'_{N,k} = \left(1-\frac{C}{C-A}\,\varrho\right)^{C-A}\cdot\left(\frac{C}{C-A}\right)^k\cdot\varrho^k\binom{C-A-k-1}{k} \quad .$$

We cannot esablish the relation between $p'_{N,k}$ and $p_{N,k}$ because it depends on the respective values of σ, ϱ and k.

Figs. 8 and 9 show the behaviour of $p'_{N,k}$ versus ϱ for C = 4 and different values of A and σ.

Figs. 10, 11 and 12 show $p'_{N,0}$, $p'_{N,1}$ and $p'_{N,2}$ respectively versus ϱ for C = 4 and different values of A and σ. It can be stressed that for small values of ϱ the probability behaviour is very slightly affected by the occurrence of crashes (i.e. they are almost independent of A and σ).

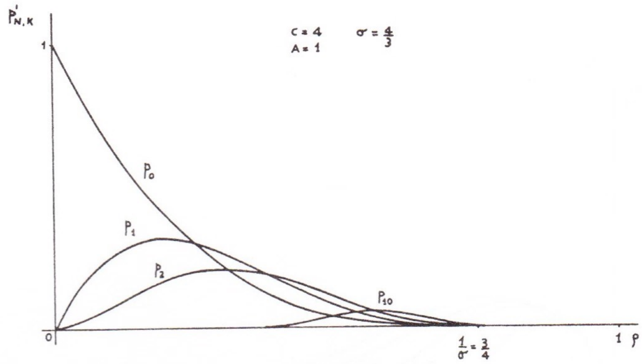

Fig. 8 - $p'_{N,k}$ versus ϱ for C = 4, A = 1 and $\sigma = \frac{4}{3}$.

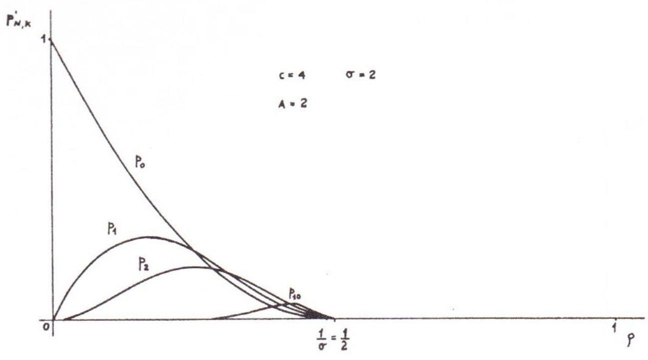

Fig. 9 - $p'_{N,k}$ versus ϱ for C = 4, A = 2 and $\sigma = 2$.

6. CONCLUSIONS

A multicomputer node has been defined as a structure that ensures high availability of services.

A decentralized access strategy to node services has also been defined which does not decrease availability in a multicomputer node.

The theoretical behaviour of the node has been studied as a set of M/M/1 queue systems, and its performance has been analyzed in the absence and in the presence of failures.

We found a performance degradation in node access due to crashes as a function of the node overload (σ).

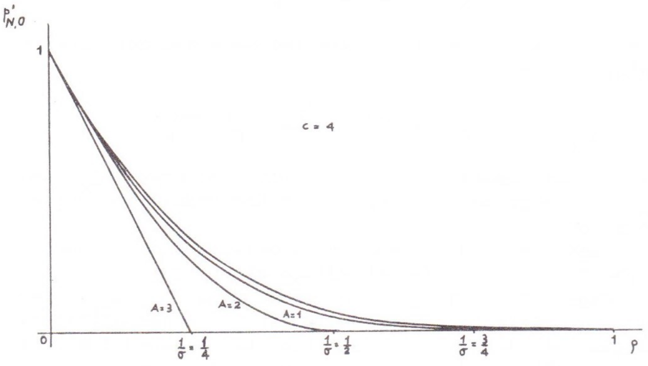

Fig. 10 – $p'_{N,0}$ versus ϱ for C = 4 and different values of A and σ.

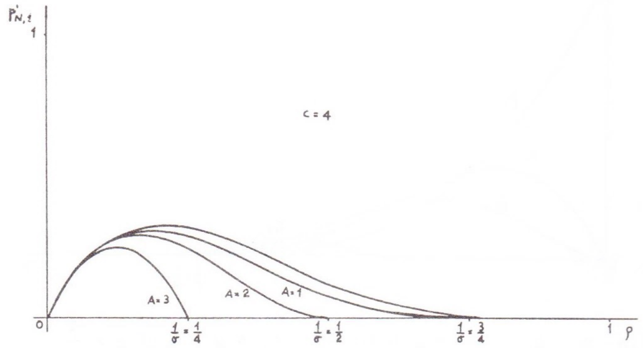

Fig. 11 – $p'_{N,1}$ versus ϱ for C = 4 and different values of A and σ.

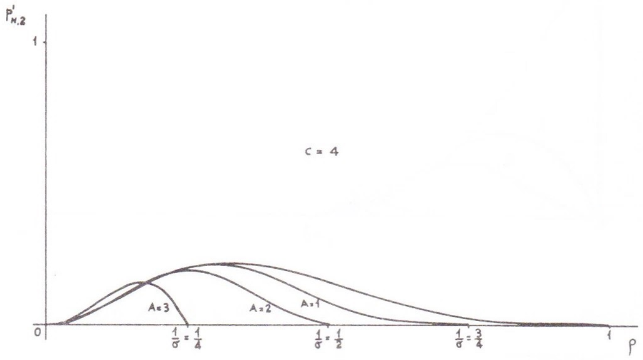

Fig. 12 – $p'_{N,2}$ versus σ for C = 4 and different values of A and σ.

The relationship was obtained between the node configuration, the maximum number of crashes accepted in the node and the average system time for every service (see Fig. 7). In a multicomputer node, performance considerations bring to use computers with the same service rate ($\mu_i = \mu$). In the same way, we found that for small values of the utilization factor ϱ the behaviour of the system is slightly affected by crash occurrences.

The results of the present study can be applied to every kind of services, when service time is known. The preceding formulae can be used to estimate the average system time (queue plus service), when an established number of computers crash in the node.

If customers request electronic mail service, the multicomputer node allows message replication for speedy and continuous service, even in the presence of failures.

Electronic mail with a multicomputer node does not include the use of permanent mail-boxes for customers wich have them allocated in customer systems. Message routing between the customers of the same node (the conditions are quite similar when the customers are connected with different multicomputer nodes) is done in one of the following ways:

1) Immediate message routing to recipient with direct connection sender/recipient through the node. The end of service is immediateley notified to sender.

2) Deferred message routing: recording of two copies of the message in provisional mail-boxes and postponed routing to recipient. Sender will receive two acknowledgments: immediately when the system accepts the message, and later when recipient receives the message.

In these two cases, electronic mail service has a different procedure and a different service time. In case 1), service time can be defined as the time to establish the connection between sender and recipient plus the time to transfer the message. In case 2), service time is the time necessary to try the connection with recipient plus the time to record the message in two different mail-boxes.

The replication of messages in the node guarantees speedy and continuous service, since recipient, as soon as available, may immediately retrieve the messages sent to him, even in the presence of failures [6].

REFERENCES

[1] L. Kleinrock, "Queueing Sytems, Vol. 1: Theory" – John Wiley & Sons, 1975.

[2] D. Ferrari, G. Serazzi and A. Zeigner, "Measurement and Tuning of Computer Systems" – Prentice -Hall, 1983.

[3] J.F. Hayes, "Modeling and Analysis of Computer Communication Networks" – Plenum Press, 1984.

[4] IEEE – SH09894, "Software Engineering Standards" – IEEE Std, 1984.

[5] W. Stalling, "Local Networks" – Computing Surveys 16(1), March 1984, pp. 3-41.

[6] E. Pasca, "Multicomputer Node: Continuous Service and Atomicity of Actions" (In Italian). – Ph. D. Thesis – Dipartimento di Ingegneria Elettronica, Università di Firenze, 1986 (In preparation).