Proceedings of the 1989 Winter Simulation Conference
E.A. MacNair, K.J. Musselman, P. Heidelberger (eds.)

A SIMULATION MODEL OF THE NFS SERVICE ON AN ETHERNET LOCAL AREA NETWORK

Floriane Blusseau
Centre National d'Etudes des Télécommunications
Service LAA/ITP/GMI
Route de Trégastel - BP 40
22301 Lannion Cédex - FRANCE

ABSTRACT

Today more and more firms use LAN (Local Area Networks), and the number of connected workstations increases quickly sometimes without any predefined policy. Moreover, numerous services are implemented, such as remote login or mailing, in order to help users in the best utilization of existing communication systems. This implies an increase of the medium load, so the local area network performance definitely deteriorates.

The problem now is to find out whether excess load brought about by the addition of connected workstations on the LAN, and by using a high level service such as NFS, saturates the network. This is the aim of our work.

The methodology we used relies on performance evaluation techniques, modelling queueing networks. In order to solve the model we obtained, we used discrete event simulation to give us the results we needed.

Keywords : Modelling - Simulation - Queueing networks - NFS - Local area networks.

1. OVERVIEW

We use NFS (Network File System, designed by Sun in 1984), between Sun workstations and an Amdahl 5840 supporting UNIX System V. NFS was recently implemented onto Amdahl which can act, due to its capabilities, as a file server. Our goal is to find an optimal files saving policy, by using modelling by queueing networks. The final model is resolved by simulation.

To achieve a model of a service such as NFS, we need to study and make a model of the underlying protocols. NFS is based on :

- XDR (eXternal Data Representation).
- RPC (Remote Procedure Call).
- UDP (User Datagram Protocol).
- IP (Internet Protocol).
- Ethernet.

The study of the protocols is carried out with performance in mind, that is, features which are of great interest are those that produce a load on the network or a delay in the request processing.

2. THE PRELIMINARY CHOICES IN THE REALIZATION OF A SIMULATION MODEL

2.1 Choice Criteria

In order to complete an NFS model, it's necessary to choose parameters that are most representative. It's a difficult part of the work, because the validity of the results relies on the quality of the choice. Having determined a list of "good" parameters, we must find all the corresponding numerical values.

The two criteria on which our choices are based are those of **load** and of **delay**. In fact, the results that interest us concern the medium load and the user request processing delay. Therefore it is necessary to analyse the different processes carried out on user requests, from the time that the request is sent, to the time that the answer is received, by retaining only the parameters that create a load or a delay. These parameters are as follows :

- The list of user commands for communicating with the NFS service.
- The time allowed for the user to think between each command sent.
- The number of user commands sent per unit of time.
- The percentage of various NFS requests produced by a user command.
- The percentage of RPC requests sent per NFS request.
- The length of NFS requests, at transmission and reception.
- The maximum length allowed of UDP datagrams and of Ethernet frames.
- The customers processing delay at transmission and reception in the different protocols used.
- The features of the communication medium (length, throughput, speed of transmission).
- The influence of a customer local workstation load on the time lapse between the reception of a reply, and the transmission of the following request; the percentage of local and remote requests at a customer workstation.
- The duration of the disk access on Amdahl.
- The priority level applicable to remote and local requests on Amdahl.
- The overall traffic in the system.

Once the list of parameters has been drawn up, the corresponding values of each parameters then have to be determined.

2.2 The Numerical Values Of The Parameters Retained

2.2.1 Introduction

The values of the parameters in the above list were gathered in part by consulting various documents and articles concerning NFS (Stein, M., 1986), and the Sun workstations (Sun 1986). We also made statistical studies on the Sun workstations and on the Amdahl computer. These studies include the realization of programmes for read and write functions in C language on Amdahl, in order to determine the duration of the disk access, and the use of the `nfsstat` command on the Sun workstations to obtain the percentages of the different NFS requests. Finally, many results are obtained by using a network analyser which takes the traffic out of the communication medium. Thus the value of the user thinking time, the length of the read and write functions performed by NFS, and the overall traffic within the system were established.

2.2.2 The List Of Numerical Values

Within the modelling scope of the NFS service, we are only interested in the user commands requiring the service, that is user commands working on files or directories. The main commands are as follows (Sun 1986) :

`cat file_name` : displays the contents of the file on the screen.
`cd dir_name` : change of directory.
`cp f_n_1 f_n_2` : copy of files.
`diff f_n_1 f_n_2` : supplies the differences between the two files.
`ls` : lists the contents of the current directory.
`mkdir dir_name` : creates a directory.
`mv f_n_1 f_n_2` : renaming of files.
`pwd` : prints the working directory.
`rm file_name` : removes the file file_name.
`rmdir dir_name` : removes the directory dir_name.

The distribution function of the statistical variable representing user thinking time, was obtained with the network analyser. The results show that 50% of the observed values are lower than 5 seconds, 85% of the observed values are lower than 16 seconds, and no values higher than 234 seconds were observed. Thus the average value is 15 seconds.

The percentages of the NFS requests produced by the user requests were calculated by using the statistical studies on the Sun workstations. The results obtained are as follows :

lookup 29% (obtaining the files attributes).

read 25% (reading of data).

getattr 22% (obtaining the files attributes).

write 9% (writing of data).

readlink 8% (acquisition of information on a link).

readdir 3% (reading of directory).

create 2% (creation of a file).

setattr 2% (setting of the file attributes).

The remaining NFS requests play only a very small part, and are subsequently not taken into account.

We observed that an NFS request was sent via the XDR and RPC protocols before being processed by the Transport protocol. The XDR protocol reorganizes the data so that all the different machines that communicate with each other will understand it. This process does not affect the network load as everything is dealt with locally and the request is not split up. Moreover, the processing time of the protocol is taken into account in the overall transmission time of a user request.

In order to process an NFS request the RPC protocol uses routines which operate locally, in the same way as a procedure call corresponding to the service request (Stein, M., 1986). Only this procedure call passes in transit via the network; it is therefore the only one to generate a load. So our investigation reveals that only one RPC request, which realizes the remote procedure call, corresponds to an NFS request.

At transmission and reception, each of the previous NFS requests uses a given number of parameters. We obtained the list of these parameters and their lengths by studying the sources of the protocols. So we were able to calculate the lengths of the NFS requests, as shown in the following table :

NFS request	Length at emission	Length at reception
read	44 bytes	Size of data read or of UDP buffer
write	Size of data written or of UDP buffer	72 bytes
lookup	287 bytes	104 bytes
readlink	32 bytes	1028 bytes
getattr	32 bytes	72 bytes
create	319 bytes	104 bytes
readdir	40 bytes	267 bytes
setattr	64 bytes	72 bytes

In the above table, there remains an unknown value corresponding to the sizes of data read and written. By using the network analyser we can observe the traffic in the system and work out the amount of data transmitted at the time of read and write operations. The following values were obtained :

`Read` : we observed that 71% of the reading generated 6 Ethernet frames, 7% generate 12 frames, 7% generate 31 frames, 7% generate 36 frames, and 8% generate 65 frames.

`Write` : we observed that 50% of the writing was due to a 6 Ethernet frame message, 8% resulted from a 9 frame message, 8% from a 14 frame message, 8% from a 17 frame message, 8% from a 38 frame message, 9% from a 48 frame message, and 9% from a 58 frame message.

Each NFS request is sent to the RPC protocol to be processed. Then it is passed on to the UDP protocol which transfers it. The NFS version currently implemented on Amdahl is restricted in that the maximum length of the UDP buffer is fixed not at the usual 8192 bytes, but at 2048 bytes. The requests whose length is greater than this value must subsequently be shortened to become several requests.

NFS is based on Ethernet at the level of Physical and Link layers of the OSI reference model. This protocol processes frames whose size does not exceed 1518 bytes of data. All datagrams whose size is greater than this value will therefore be segmented by the IP protocol.

The processing delay of a customer request at transmission and reception, in the different protocols used are as follows :

`UDP protocol` : the processing carried out by the UDP protocol is so minimal that the associated delay is negligible.

`IP protocol` : the processing delay of the IP protocol, corresponds to the segmentation of datagrams. The value of this delay is a 9 milliseconds, obtained by using the network analyser.

`Ethernet protocol` : the Ethernet protocol is divided up into several layers, that are the LLC layer (Logical Link Control), and the MAC layer (Medium Access Control). The processing delay produced by the LLC layer is 0.375 milliseconds, and the processing delay by the MAC layer is 0.175 milliseconds.

The local area network at CNET Lannion is made up of a 500 m long bus, with a throughput of 10 Mbits/s, and a transmission speed of 200000 km/s. This gives us an end to end propagation delay of 2.5 microseconds, and a minimum frame length of 50 bits (which corresponds to twice the end to end propagation delay).

There are several possibilities when considering the consequence of the local load of the customer workstation on the delay between the reception of a reply and the transmission of the following request. The first solution consists of studying and building a detailed model of the components which are involved in this delay. The second possibility is to consider a universal average value which takes the user thinking time, the average load of a Sun workstation, and the effective request production delay into account all at the same time. We choose the second solution because the values which are measured by the network analyser to calculate the user thinking time take the other components into account.

The read/write programmes of a data unit were used to determine the duration of disk access on Amdahl. The results are as follows :

`Read access` : in 99% of the case, the values observed were on average 35.21 milliseconds, and in the other 1% they were on average 110 milliseconds.

`Write access` : in 95% of the case, the values observed were on average 33.3 milliseconds, in 4.5% of the case they were on average 99 milliseconds, and in 0.5% of the case, they were on average 200 milliseconds.

Amdahl makes no distinction between a remote request and a local one, at the level of processing priorities. Therefore, in the case of a great load on Amdahl, it is not possible to favour the processing of a remote request, to avoid the retransmission of the RPC request due to the end of the timeout.

The network analyser, by simple observation, gives the characteristics of the overall traffic in the system. The values observed of the number of frames circulating in the network in one second are, in 99% of the case, on average 24, and in the other 1%, on average 570.

In 71% of the case the length of the frames circulating in the network is 64 bytes, in 5% 128 bytes, in 3% 512 bytes and in 21% 1024 bytes.

2.3 Summary Of The Choices Of Parameters In An Example

2.3.1 Introduction

In order to set the ideas out clearly, we will sum up the different information gathered, in the example of a file copy.

Let us assume that a user has mounted two directories, one under the local directory `mnt1` and the other one under the local directory `mnt2`, and that the user wants to copy file `mnt1/file_name`, under the directory `mnt2`, with the same name. He will use the command `cp mnt1/file_name mnt2/file_name`. The size of the copied file is taken to be 3500 bytes.

2.3.2 Study Of The Command Processing Sequence

We will study stage by stage the processes carried out on the user command from transmission to the reception of its reply. It should be noticed that the copy was done from one remote file to another remote file.

First of all we must determined which NFS requests are produced by the user command `cp`, by applying the `nfsstat` command to the Sun workstation. The NFS requests transmitted are : getattr - lookup - read - create - write. So the `cp` command sends five NFS requests and therefore five RPC requests.

The RPC protocol functions in a synchronous mode, therefore the requests are processed one after the other, the processing of a request starting only when the preceding request has been replied.

The RPC request is then processed by the IP protocol (the UDP protocol is not taken into account as it adds on negligible load and delay). The processing by IP protocol requires a delay of 9 ms. At this level the request can be segmented into several datagrams if the received message contains more than one data frame (1518 bytes). The messages corresponding to the NFS requests "lookup", "getattr", "read" and "create" will not be segmented. However the message corresponding to the write request is split into three datagrams as it concerns a file of 3500 bytes.

Each of the datagrams is then processed by the LLC layer with a delay of 0.375 ms, and then by the MAC layer with a delay of 0.175 ms. At the level of the MAC layer, it is

necessary to wait until the communication medium is free before transmitting. In the event of collision, the waiting time is determined before retransmission and then the frame is retransmitted.

The medium of communication transfers each frame to all the stations which are connected to it. This is expressed by a delay equal to the length of the frame in bits, divided by the medium throughput (here, 10 Mbits/s), that is 1.2 ms for a frame of 1518 bytes.

At reception, the frames are only retained by Amdahl. Their progress across the MAC and then the LLC layers, at reception, adds on a delay of 0.175 ms and 0.375 ms respectively. The datagram is reassembled at the level of the IP module. The processing of messages corresponding to NFS requests other than "read" or "write" does not cause a processing delay on Amdahl. On the other hand the "read" and "write" requests require a disk access, that is a delay corresponding to the values mentioned in the preceding part of the chapter. Once the read or write operations has been carried out, a reply is produced and transmitted across the IP and Ethernet protocols with the same delays as before. Only the reply to a "read" request causes a segmentation into three datagrams at the IP level. When the reply to the RPC request has been received, the next RPC request can be processed.

When all the replies to the RPC requests have been received, the NFS service is terminated and the user can continue with his work.

The summary diagram in appendix A represents the processing of the cp user command when there are no retransmissions at the RPC level and no collisions on the communication medium. Moreover, it is drawn within the scope of a complete implementation of NFS service, in other words, the length of UDP buffer is equal to 8192 bytes. Therefore no RPC requests are split.

The transmission of five NFS requests, therefore five RPC requests, corresponds well to the cp user command. These five requests cause the transmission of seven Ethernet frames on the medium. Each reply to an RPC request produces an Ethernet frame, except for the reply to the "read" request which produces three. In all there are fourteen Ethernet frames circulating on the medium. The processing delays are as follows :

- 2*5*9 ms for the IP protocol (9 ms), on the customer Sun workstation and Amdahl (*2), and for each RPC request (5).
- 2*2*7*(0.375ms+0.175ms) for the LLC layer (0.375 ms) and MAC layer (0.175ms), at transmission and reception (*2), on the customer Sun workstation and Amdahl (*2), and for each Ethernet frame sent (*7).
- 35.21 ms for the read access disk, and 33.3 ms for the write access disk.
- all the various frame transmission delays on the communication medium.

The sum of all these delays gives a total value of 0.17 seconds.

3. MODELLING

This stage includes the drawing up of the preliminary model diagrams in the form of queueing networks, taking into account the different choices of parameters previously taken. It also includes the building and the resolution of the model by using the package QNAP (Queueing Network Analysis Package) (Véran, M., and Potier, D., 1984), and lastly, analysis of the results.

3.1 QNAP Language And Simulation

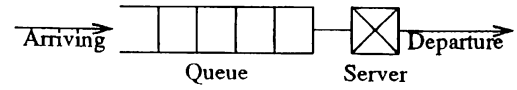
The QNAP language was chosen to model the problem, after looking at different existing modelling tools, which enabled us to choose the most appropriate method for our model. We need quantitative information, for example network load, queueing time. So we can eliminate a certain number of softwares which work on the basis of non timed Petri nets, and supply therefore qualitative information rather than quantitative. We need a tool which is based rather on queueing networks.

Another reason why we can eliminate many of the existing softwares is simply that the software must be available within CNET. Besides, softwares are often specific to a company (for example RESQ at IBM).

Lastly, some softwares are very specialised, that is, they only know how to process a certain type of problems (for example SURF which is specified for fault tolerant systems).

So, after investigating all the possibilities, we choose the package QNAP (Queueing Network Analysis Package) to build our model. This tool produces the description and resolution of queueing networks by simulation and exact or approximate analytical methods. For our study we will only use the simulation resolution due to the level of detail we need. QNAP is available within CNET.

QNAP was developed jointly by INRIA and Bull company. The latest version (QNAP2, 1982) is fitted up with a user interface which defines the network and supplies a print-out of results in the form of summarized tables. QNAP2 also includes a command language which defines the different elements in the model, and an algorithmic language. The writing of a programme is quite simple once the queueing network, on which the model is based, has been clearly defined. The following diagram shows a waiting queue :



When building the model, it is necessary to know the description of the customers arriving in the queue, the description of the server process for customers, and the queue discipline, that is the procedure by which the customers are chosen in the queue to be processed.

The QNAP language describes the configuration of the queueing network which is represented by a group of *stations* (a station includes one or more *servers*, serving one single *queue*), which the *customer* pass according to the *routing rules*. The customers can be allocated to different *classes* which characterize the various processes in the stations. The

QNAP stations represent the physical or logical processing characteristics of the system which is to be modeled (central unit, protocol). The customers represent processes carried out by the stations.

Next, the QNAP language describes the processing performed by each station on the customers. In the case of a simulation resolution, this procedure can be described as a simple delay defined by its probability distribution, or by a more complex algorithm, including synchonization operations or customers generation.

Lastly the QNAP language allows the user to carry out a resolution check, either by initialising or updating parameters, and by activating the resolution methods (in our case, simulation), and finally by the printing of the results.

3.2 Realization Of The NFS Service Model

3.2.1 Introduction

There are two necessary stages in the building process of the NFS service model. To start with we built a detailed model, without including many of the Sun workstations connected to the network. Here, one QNAP station represents a protocol for one machine, so there are as many QNAP stations as protocols, this number being multiplied by the number of machines (Sun workstations and Amdahl). This explains why the first model was needed as it took up a lot of memory space.

The results gained during simulation of the first model are then used to build a more general model where a QNAP station represents more than one protocol. This aggregated model can therefore be solved for a large number of Sun workstations.

3.2.2 The Aggregated Model

The QNAP programme corresponding to the aggregated model of the NFS service, is written on the basis of the results previously obtained. We then resolved the aggregated model by simulation for different values of the number of Sun Workstations and different user thinking time values.

So we measured the relationship between these parameters, and five results, that are : the medium load, the average user request processing delay, the collision rate on the network, the resending rate of RPC requests, and the drop out rate of RPC requests.

The results are summarized in the graphs in the appendix B. The first graph shows the medium load. We have studied this rate in a number of Sun workstations ranging from one to twenty five, and for a user thinking time with an average value of 15 seconds. The model is very useful in that it allows us to see beyond the only point supplied by the network analyser.

The second graph shows the average user request processing delay, according to the connected workstations, and to the user thinking time. We observed that for a user thinking time of 15 seconds there is a big increase in the average user request processing delay, from 20 connected workstations. The increase is greater for a lower user thinking time, with the processing delay lasting almost 50 seconds. The point at which the delays increase relates to the fact that the RPC timeout ends (0.7 second). Therefore the request has to be retransmitted, which in turn extends the processing delay accordingly.

On the third graph also, it can be seen that from 20 connected Sun workstations, the collision rate, for a user thinking time of 15 seconds, increases greatly.

The observations of the second graph are confirmed on the fourth graph, where the resending rate of RPC requests goes up to 200%.

The last graph shows the drop out rate of RPC requests, which remains low for a user thinking time of 15 seconds. For a user thinking time of 10 milliseconds, the drop out rate reaches 20% for 15 connected Sun workstations, and 50% for 25 connected workstations (that is drop out for one in two requests).

3.3 Use of the NFS service model for file savings

The Amdahl 5840 has a disk capacity that allows it to be a file server on the Ethernet LAN at CNET Lannion. Moreover, NFS has been implemented onto Amdahl.

Thus we can believe that the connected Sun workstations will save their files onto Amdahl using NFS. We want to know the LAN performance evolution while doing file saving on the Amdahl host. Furthermore, it's important to obtain the average file saving delay when there is no other load on the LAN.

We modified the NFS model considering that the connected workstations made one, and only one, file saving during the simulation. That is, the only NFS request is a `write`, whose length equals the file saving data length. When the file saving is done, the Sun work is finished.

First of all, we solved the NFS model for only one Sun workstation doing savings. We obtained a data rate of 70 Mbytes per hour. Different tests on the LAN validate this value. Averaging the available disk capacities of Sun workstations, the length of data to be saved is approximately 500 Mbytes. So, with a rate of 70 Mbytes/hour, the file saving delay is a bit more than seven hours. In this case, only one saving can be made per night.

Since successive file savings are not feasible in one night, is it possible to make simultaneous savings ?. To know that, we solved the NFS model for more than one connected Sun workstation. When two Sun workstations save their files simultaneously, the data rate is 41 Mbytes per hour, giving a file saving delay of about twelve hours. Thus simultaneous savings are no more feasible in one night than successives ones.

Finally we tried to know if a file saving can be made when there is other load on the LAN. We solved the NFS model for one Sun workstation doing a file saving, and ten Sun workstations sending requests every fifteen seconds. The result is the drop out of the file saving due to the retransmission of one RPC request more than three times, without receiving any reply. Thus file saving during the day is not feasible.

The NFS model is used to evaluate different file saving policies between Sun workstations and Amdahl. The results show that only one Sun workstation could make savings per night. Giving the increase of connected workstations, we can wonder if it was not more feasible to use other protocols such as FTP (File Transfer Protocol), indeed give up the idea of using Amdahl as a file server.

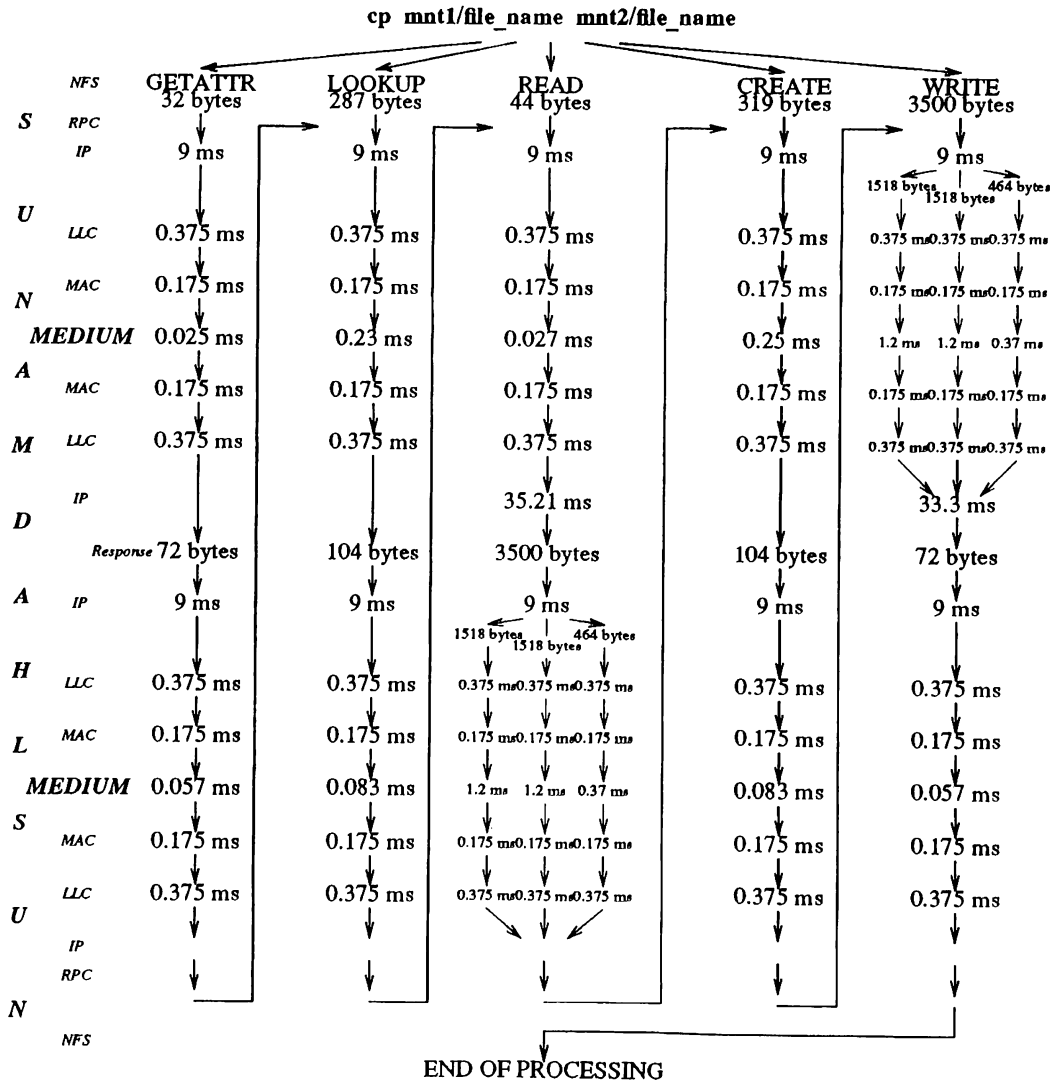
4. CONCLUSION

The methodology we used consists of breaking a service into its different underlying protocols, carrying out a study of each of these protocols with a view to performance, and then building a model based on the fundamental points of the study. Thus a collection of reusable models is built up. If, for example, one wanted to carry out a performance evaluation of a service based on the TCP (Transmission Control Protocol), IP and Ethernet protocols, one only has to model the elements concerned with load and delay in the TCP protocol (opening of connection, ...). The model could be stacked with the IP and Ethernet models which are already built. With this methodology, it is also possible to go back later to the modelling of a protocol, in order to improve it, without having to rebuild a complete model.

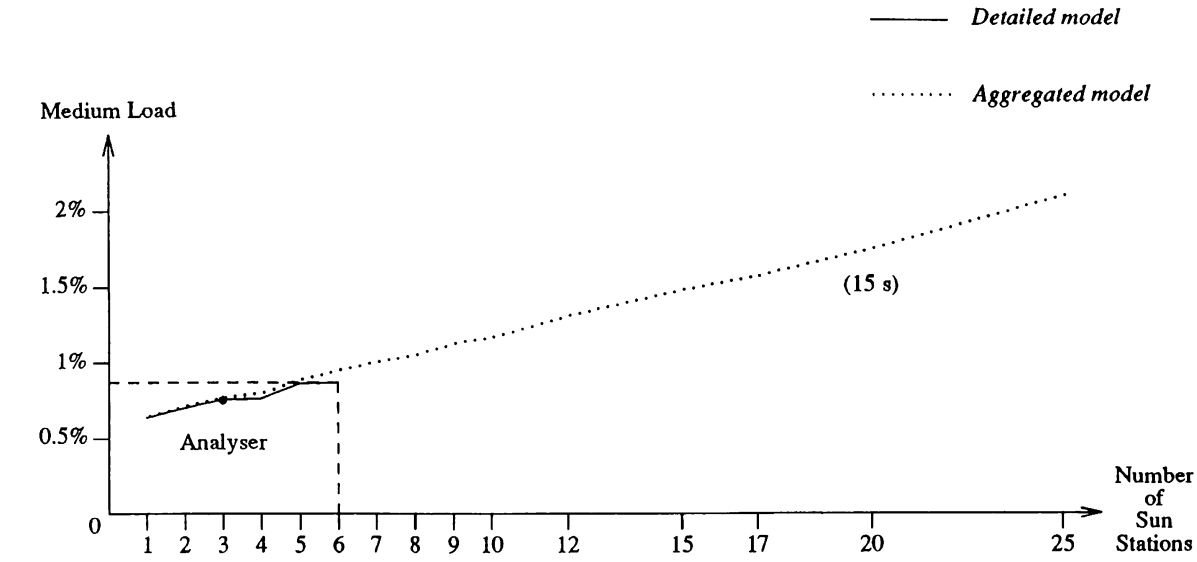
One of the main results of our work is the acquisition of a model of a service such as NFS which is considered to be the standard, and was recently implemented on the Amdahl computer.

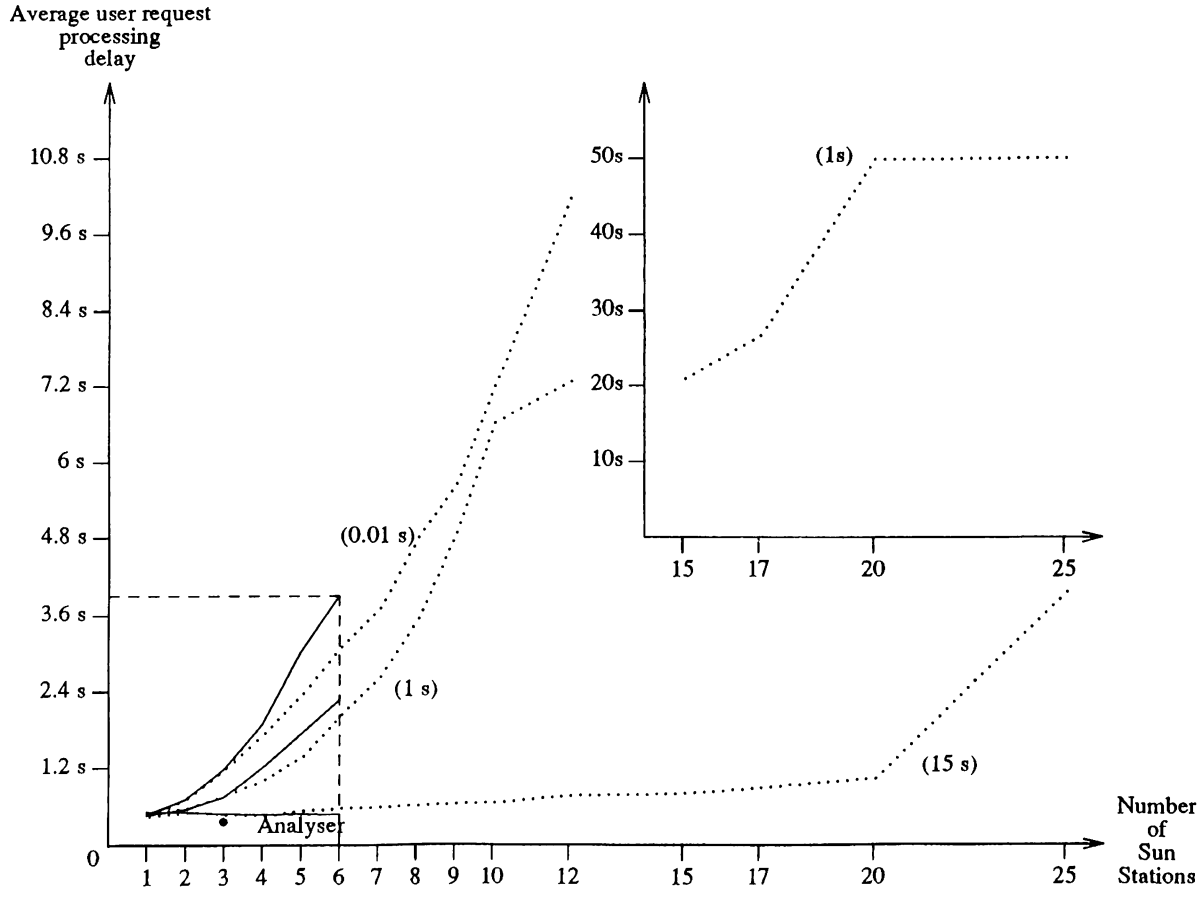
Now we must continue with the simulation to refine the results already obtained, and then we envisage to build the model of a bridge, that reduce the network load.

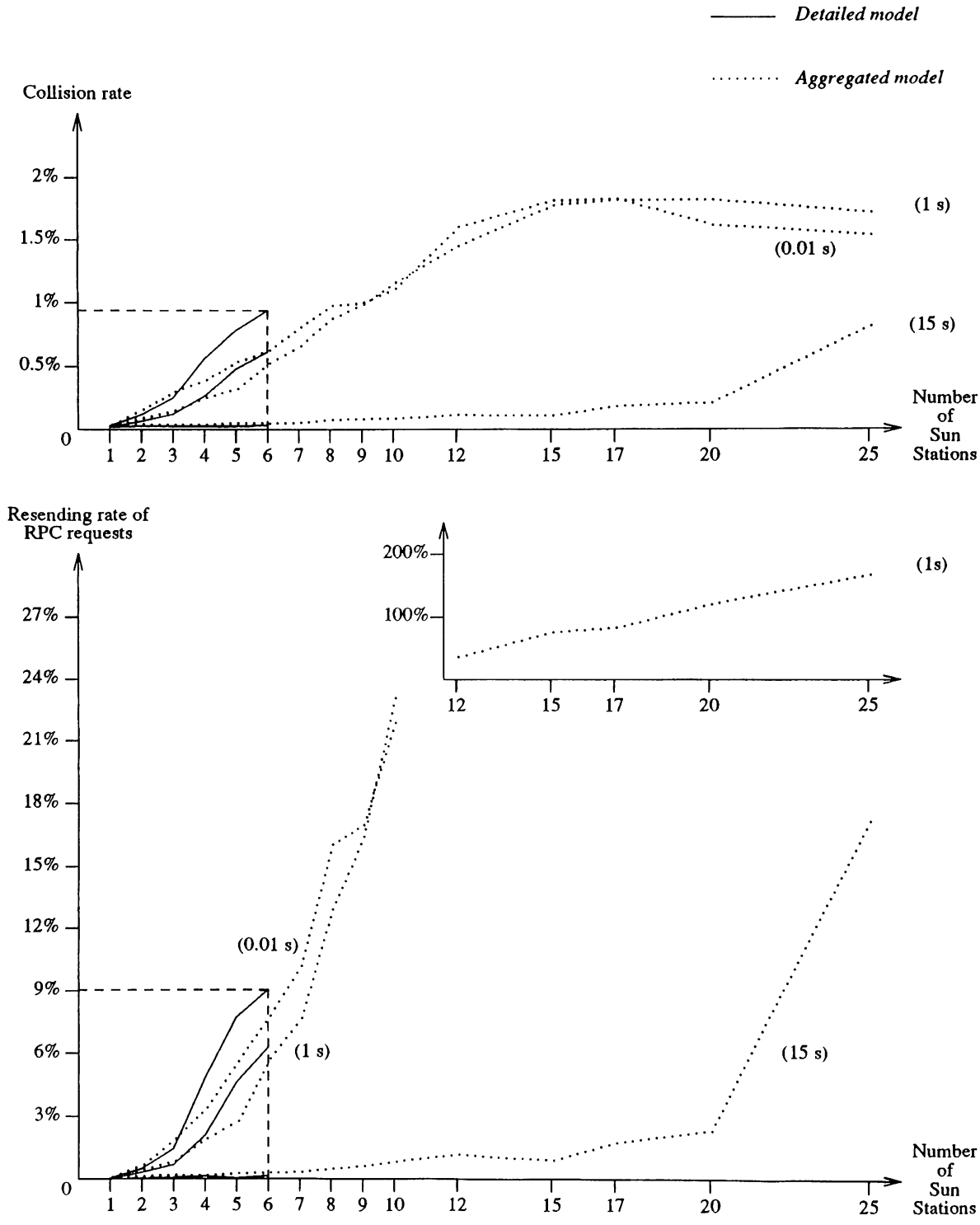
APPENDIX A : AN EXEMPLE OF A COMMAND PROCESSING

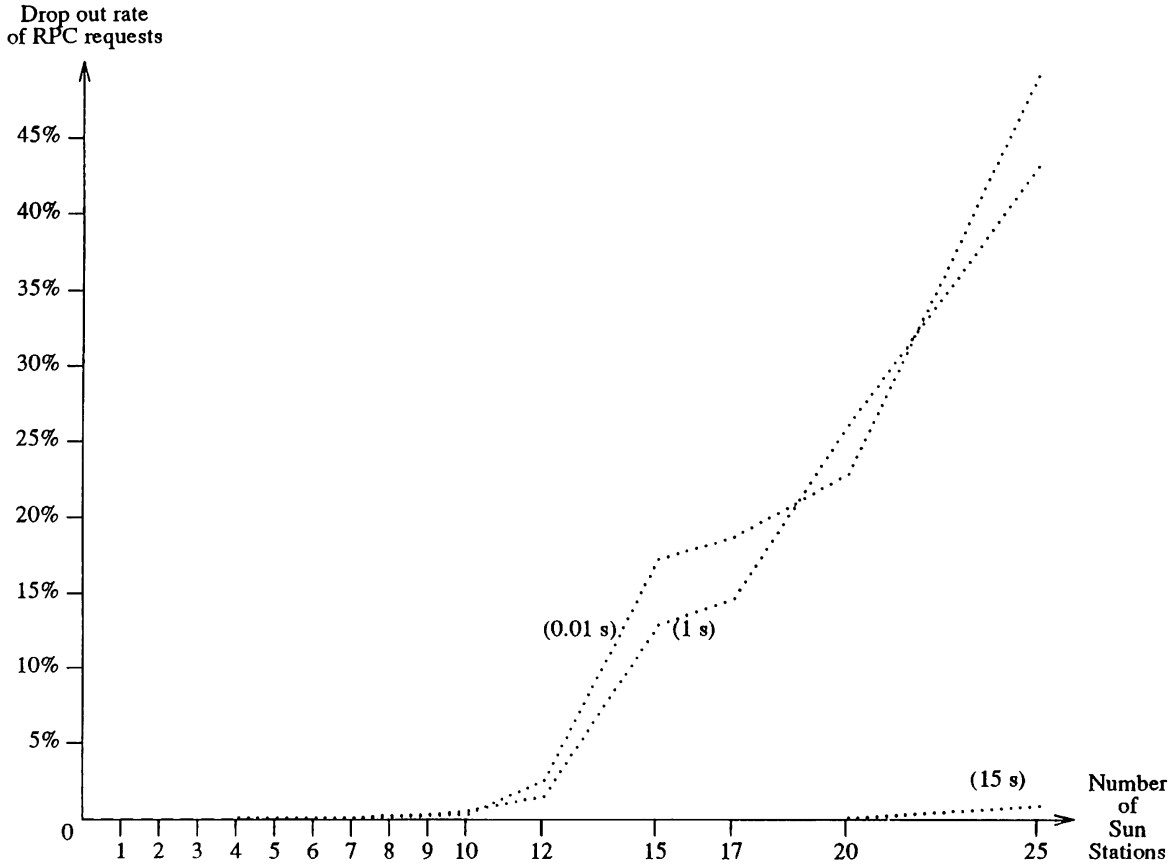


APPENDIX B : THE RESULTING GRAPHS






Detailed model
Aggregated model
Collision rate
2%
1.5%
1%
0.5%
0
(1 s)
(0.01 s)
(15 s)
Number of Sun Stations
1 2 3 4 5 6 7 8 9 10 12 15 17 20 25
Resending rate of RPC requests
200%
100%
(1s)
12 15 17 20 25
27%
24%
21%
18%
15%
12%
9%
6%
3%
0
(0.01 s)
(1 s)
(15 s)
Number of Sun Stations
1 2 3 4 5 6 7 8 9 10 12 15 17 20 25



REFERENCES

Bach, M.J. (1986). The design of the UNIX operating system. *Prentice-Hall International Editions.*

IEEE Computer society (Juillet 1983). IEEE Standard 802.3.

Rutgers (1987). Introduction to the Internet Protocols. *The State University of New Jersey.*

Sandberg, R. (1986). The Sun Network File System : Design, implementation and experience. *Sun Microsystems.*

SRI International (1982). Internet Protocol Transition Workbook. *Network information center.*

Stein, M. (1986). The Network File System. *Sun Microsystems.*

Sun Microsystems (1986). Sun System Overview.

Véran, M., and Potier, D. (1984). QNAP2 : A portable environment for queueing systems modelling. In *Colloque international sur la modélisation et les outils d'analyse de performance.* Paris.

AUTHOR'S BIOGRAPHY

FLORIANE BLUSSEAU works on a thesis about the modelling and performance evaluation of a distributed filesystem on a local area network. Her current research involves the Amdahl-FRANCE company and the Telecommunications Research Center of Lannion. She is a student member of the university of Rennes.

Floriane Blusseau
Centre National d'Etudes des Télécommunications
Service LAA/ITP/GMI
Route de Trégastel
BP 40
22301 Lannion Cédex - FRANCE
(33) 96-053881