November 23rd 1994
draft

High Performance TCP/IP communications with HPs and IBMs connected around an Ancor Fibre Channel Fabric

Fabrice Chantemargue

I. Introduction

This note is aimed at providing communication times for transferring data from memory to memory, between *IBM RS/6000* and *HP/9000* workstations, via a 16 port Ancor Fibre Channel Fabric, using *TCP/IP* protocols, when running *High Performance TCP/IP* on machine's kernels. *High Performance TCP/IP* is nothing else than the common name for *RFC 1323*. Today, most workstations (*IBM*, *HP*, *DEC*, *SGI* at least) support this potentiality, but to our knowledge, none implements it by default. This is the user's responsibility to switch it on (see for instance appendix for switching *RFC 1323* on under *AIX* Operating System).
Figure *1* shows the experimental hardware configuration used (all the Fibre Channel products are currently running *Fibre Channel* version 3.9).

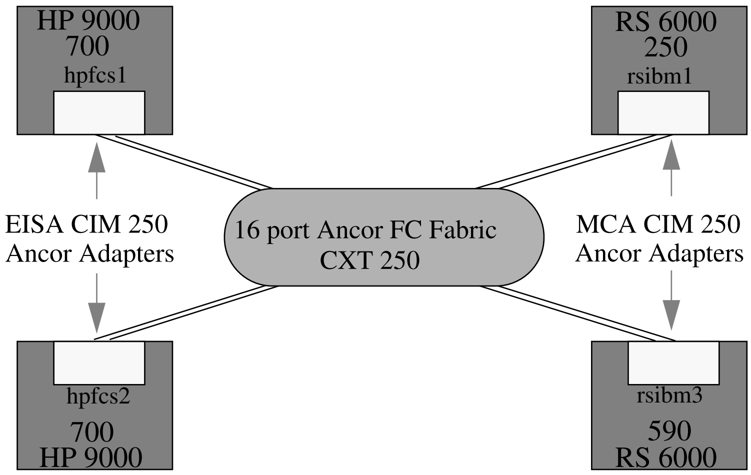

Figure 1: Experimental hardware configuration.

The two *HP/9000* machines are *700 series (735)* with a *PA7100* processor running at 99 MHz. The *IBM RS/6000* workstation pool is composed of a *590* model with a *Power2* processor running at 66 MHz and a *250* model with a *PowerPC 601* at 66 MHz. Two kinds of adapters, depending on the machine in which they are intended to be located, can be used to provide access to the Fibre Channel world (see [3], [4], [5], [6] for more detail):

- an *EISA CIM* (Extended Industry Standard Architecture Communications Interface Module) *250* Ancor adapter for the *HP* workstations. It is a 32 bit bus Master capable of data transfer rates up to 33 Mbytes/s
- a *MCA CIM* (Micro Channel Architecture Communications Interface Module) *250* Ancor adapter for the *IBM* workstation. It is a 32 bit bus Master capable of data transfer

rates up to 40 Mbytes/s.

Some communication times have been previously reported in papers [1] and [2]: they were performed strictly with the same hardware, but the *RFC 1323* was not, at that time, turned on. The comparison of the communication times presented in this note and in the previous notes ([1] and [2]) will allow to quantify the improvement brought by running *High Performance TCP/IP* on workstations, for *TCP/IP* communications.

II. Benchmark test conditions

The program used to test out the performance is '*netperf*'. This program is in the public domain and has been originally evolved by Rick Jones (*raj@cup.hp.com*, HP Lab in Cupertino, California, *USA*). This program can be obtained by anonymous *ftp* to *iworks.ecn.uiowa.edu*. In [1] and [2], communication benchmarks were achieved using 'ttcp' program. The reason why we used 'netperf' is because of a more flexible and more convenient user interface and because of a stronger potentiality. Otherwise, there is no difference in the performance issued from the one or the other.

We used five input parameters to this program, which can be chosen independently:

- the size of the message to be sent,
- the duration length of the test (to perform an average measurement),
- the socket buffer size,
- the port number (mainly useful for performing bi-directional communications),
- the *TCP_no_delay* option: when set (on the sender side only), this option will shun packet grouping before sending.

The size of the message to be transferred ranges from 1 byte up to 2 Mbytes. There is no limitation on the message size at the user level.
The duration length for the test was equal to 30 seconds, in order to provide relevant measurements.
Several values for the socket buffer size have been tested according to the experiments and to the machine specifications: 131072 bytes, 262144 bytes, 349525 bytes, 524288 bytes and 1048576 bytes.
When the *TCP_no_delay* option is not set, messages are grouped before being sent. It does not affect particularly the bandwidth estimation, which is generally measured for large messages. It is completely different in the case of evaluating the overhead of the system (interface plus fabric plus all the software): the measurement will indeed return an over optimistic expectation. The influence of grouping messages is of course more sensitive on small messages than on large messages. To allow a realistic overhead measurement, the *TCP_no_delay* option must be turned on. In that case, it will ensure the sender to send packets one by one.

Graphs show hereinafter how the communication time and therefore the bandwidth scale with the size of the message to be transferred and the socket buffer size. Figures *a* display bandwidth measurements, whilst figures *b* display corresponding communication times.

III. Communications between *IBMs*

A. *TCP_no_delay* option disabled

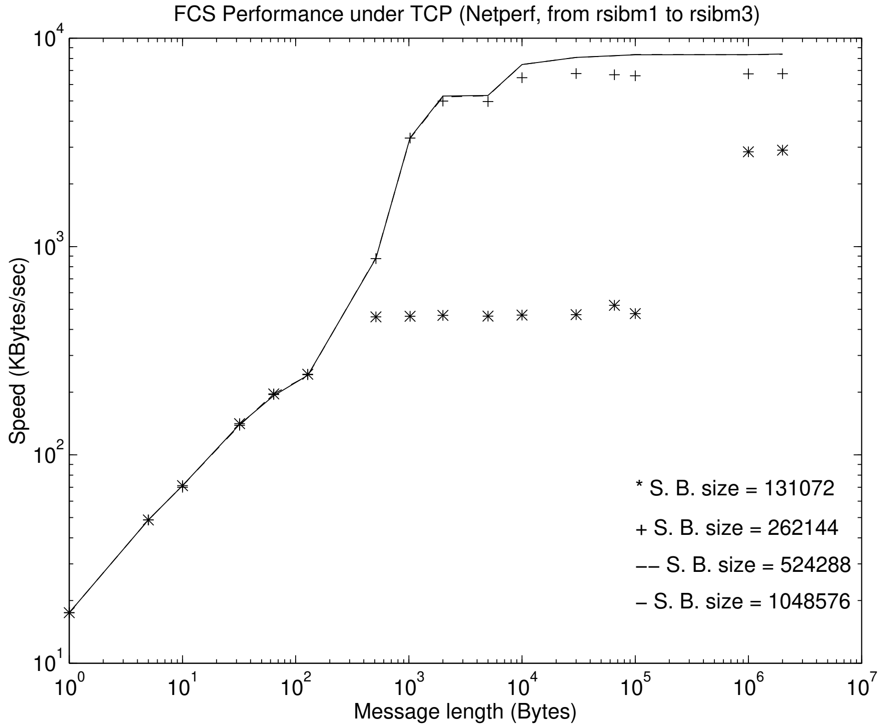

Figure 2a: Bandwidth performance from *rsibm1* to *rsibm3*.

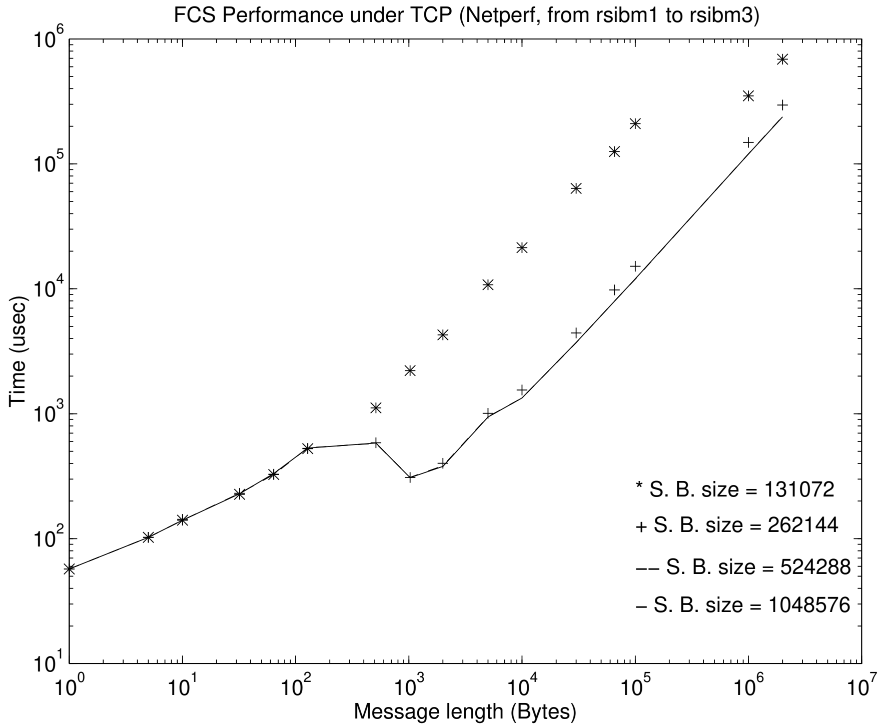

Figure 2b: Latency performance from *rsibm1* to *rsibm3*.

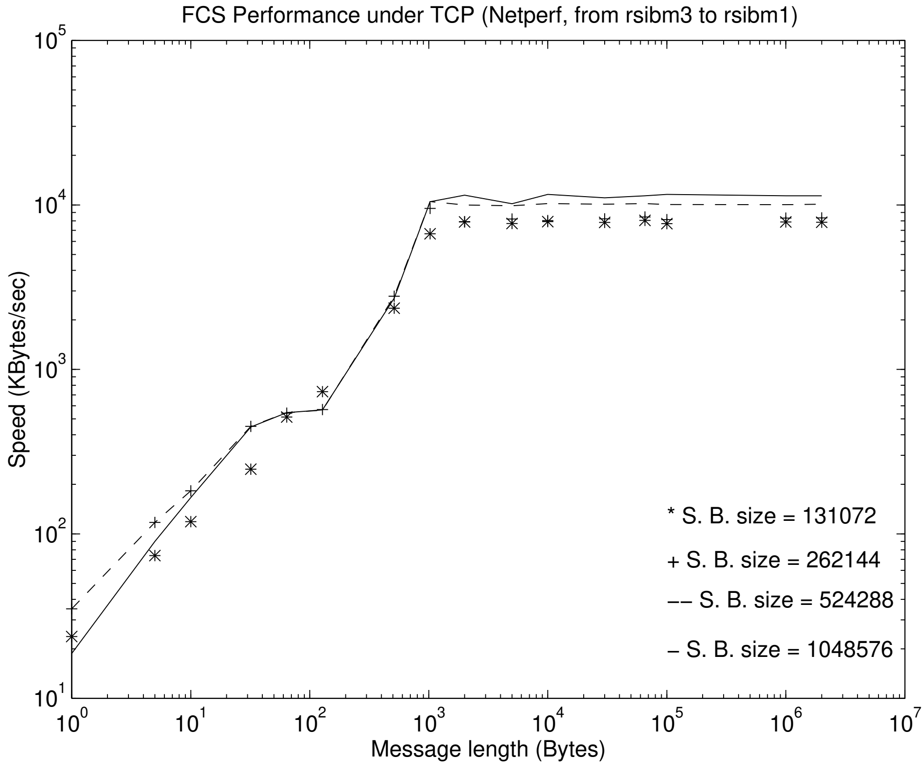

Figure 3a: Bandwidth performance from *rsibm3* to *rsibm1*.

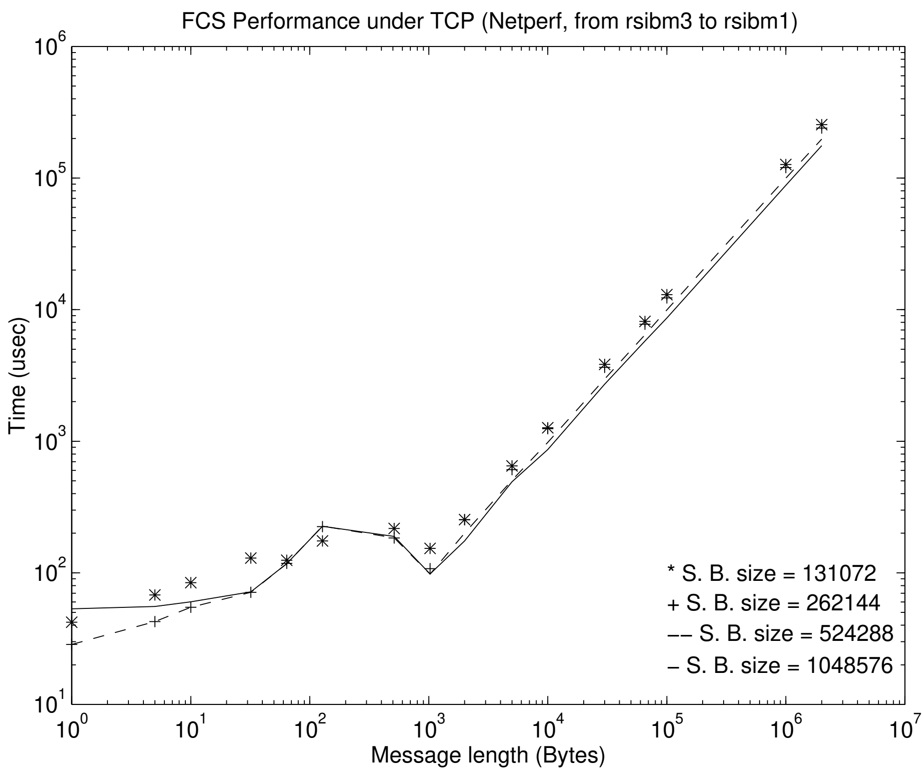

Figure 3b: Latency performance from *rsibm3* to *rsibm1*.

B. *TCP_no_delay* option set

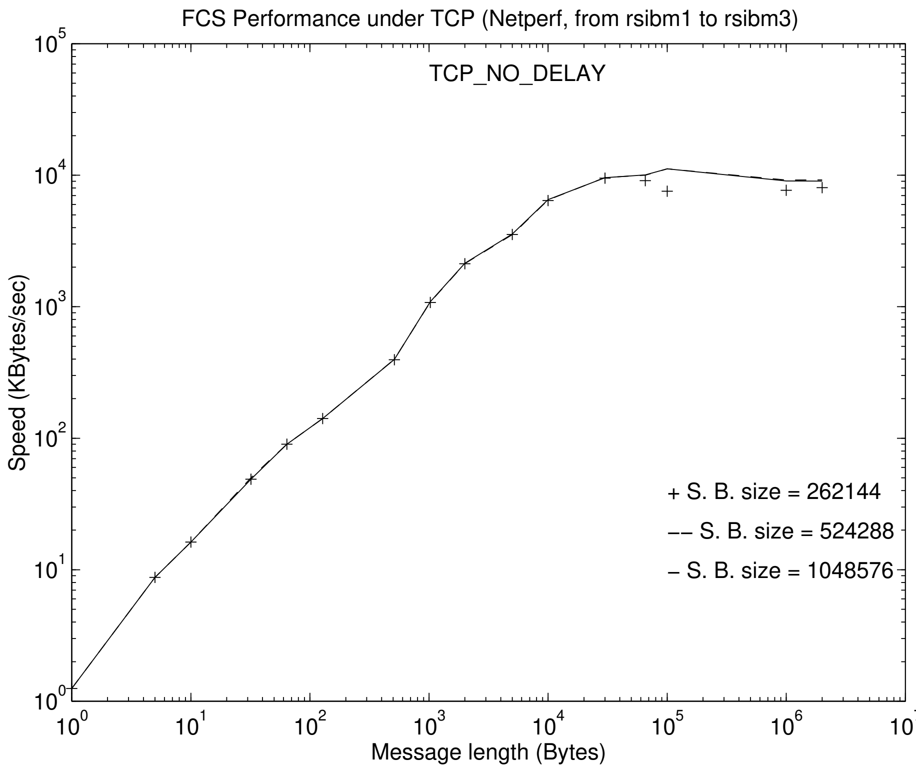

Figure 4a: Bandwidth performance from *rsibm1* to *rsibm3*.

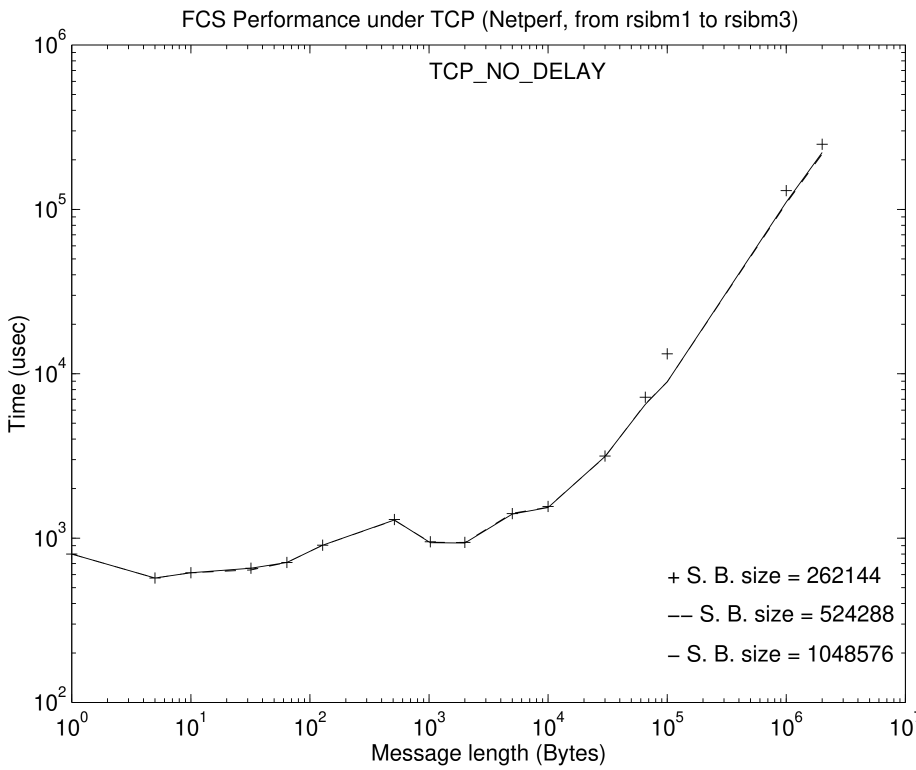

Figure 4b: Latency performance from *rsibm1* to *rsibm3*.

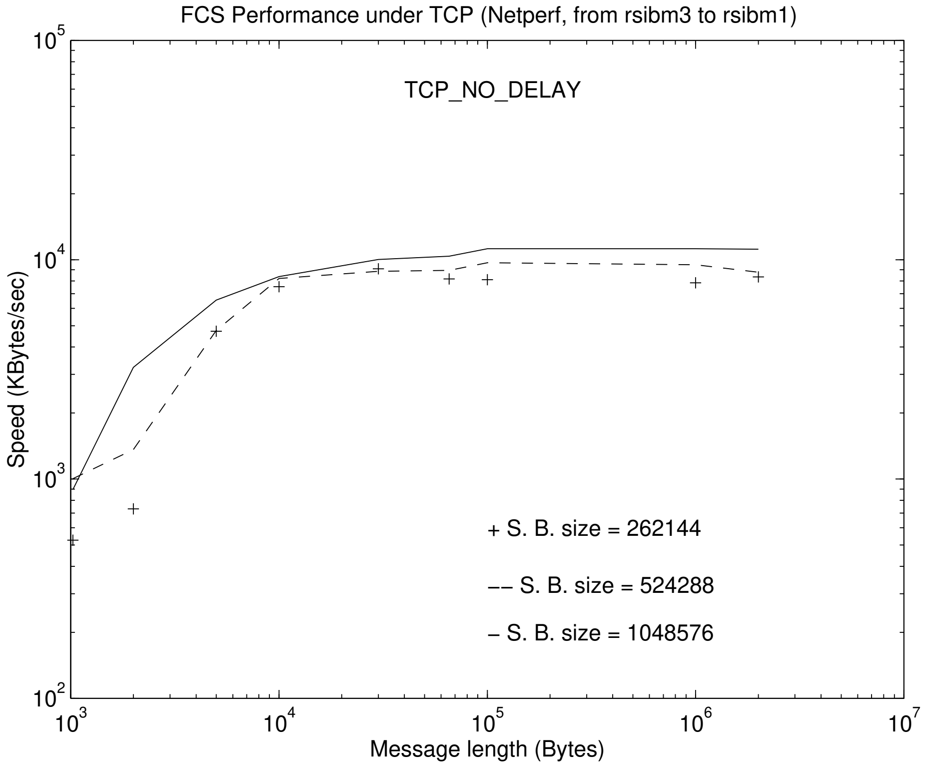

Figure 5a: Bandwidth performance from *rsibm3* to *rsibm1*.

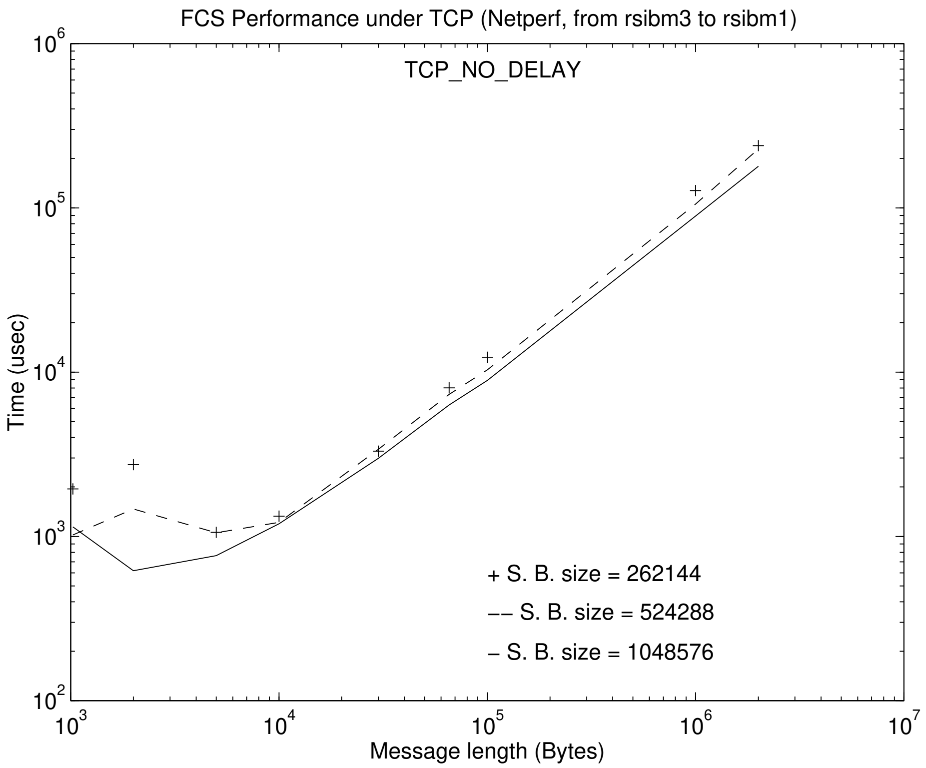

Figure 5b: Latency performance from *rsibm3* to *rsibm1*.

C. Simultaneous bi-directional communications (*TCP_no_delay* option disabled)

Bi-directional communications have been performed in order to measure the degradation of performance, if any. On figure 6a, the upper curve group (lower curve group on figure 6b) represents the expected bi-directional performance for three different socket buffer sizes. By definition, the expected bi-directional performance is the sum of the performance measured in each direction independently. The lower curve group on figure 6a (upper curve group on figure 6b) represents the measured bi-directional performance.

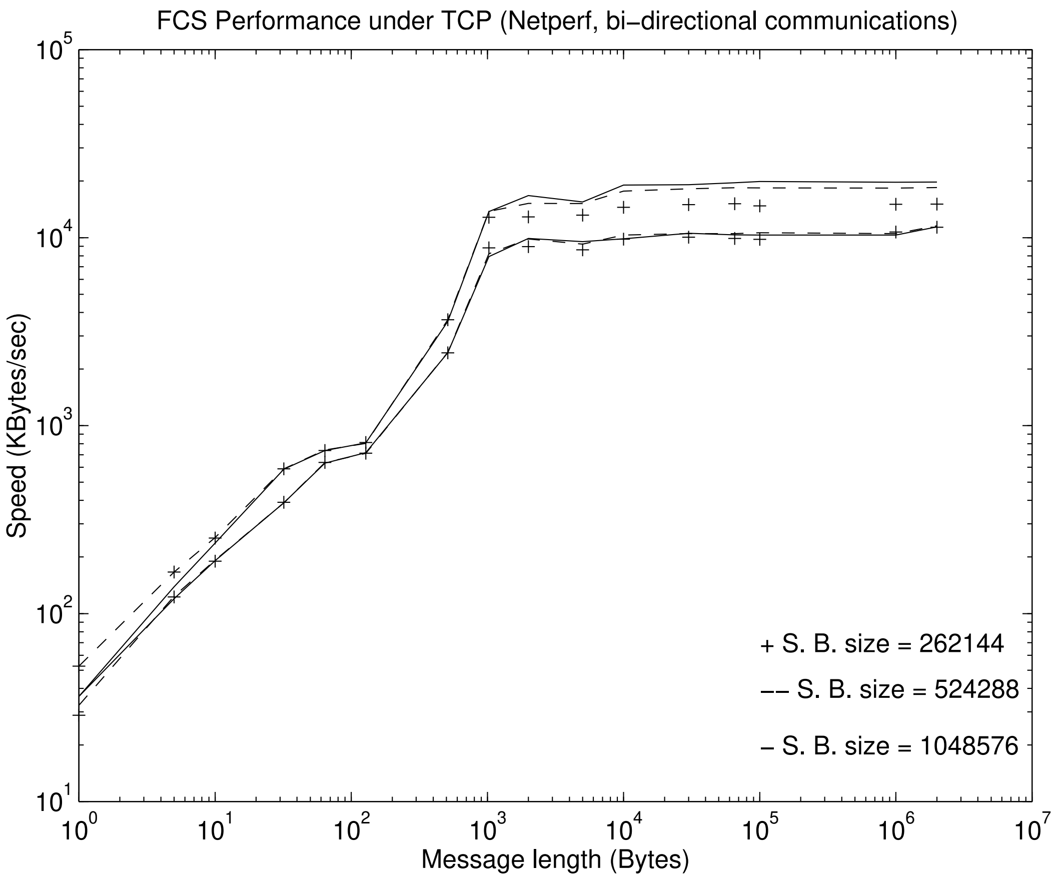

Figure 6a: Bandwidth performance when bi-directional communications are performed.

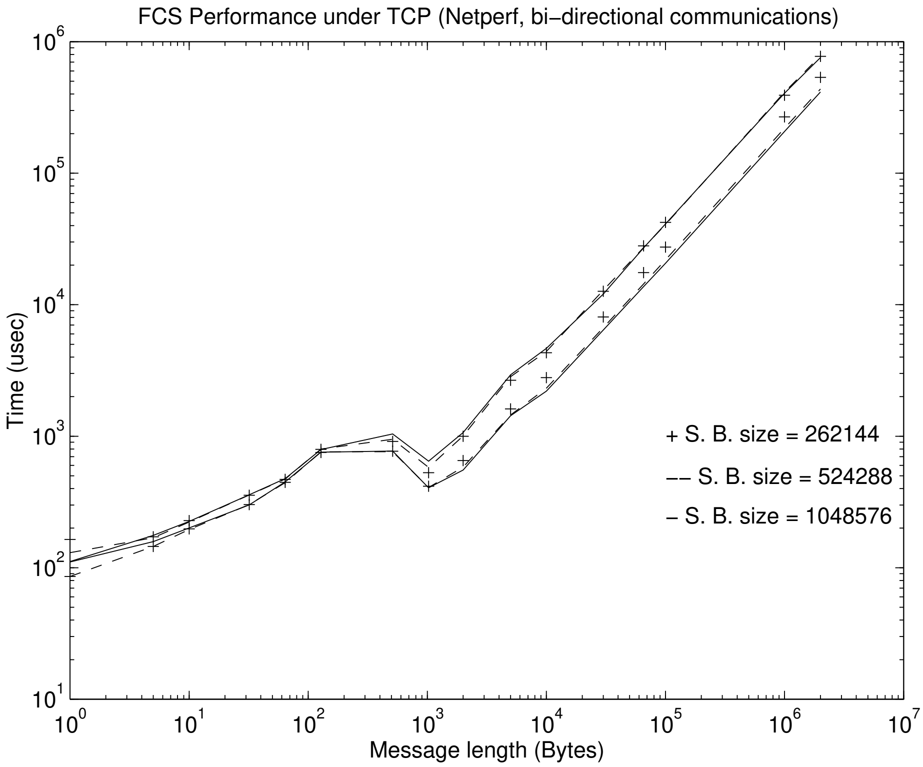

Figure 6b: Latency performance when bi-directional communications are performed.

On figure *6c*, the table is summarizing the degradation as a function of the socket buffer size and the packet size.

Table 1: Degradation measured for bi-directional communications

Socket Buffer Size \ Packet Size	< 1 Kbytes	> 1 Kbytes
1 Mbytes	21%	45%
512 Kbytes	28%	41%
256 Kbytes	28%	32%

A performance degradation of around 30% on average is reported, with regard to the expected performance. The degradation is a function of the message size and the socket buffer size. For the largest socket buffer size, the degradation is more important for large packets (greater than 1 Kbytes) than for small packets (smaller than 1 Kbytes). For the smallest socket buffer size the degradation is almost constant whatever the packet size is.

IV. Communications between *HPs*

A. *TCP_no_delay* option disabled

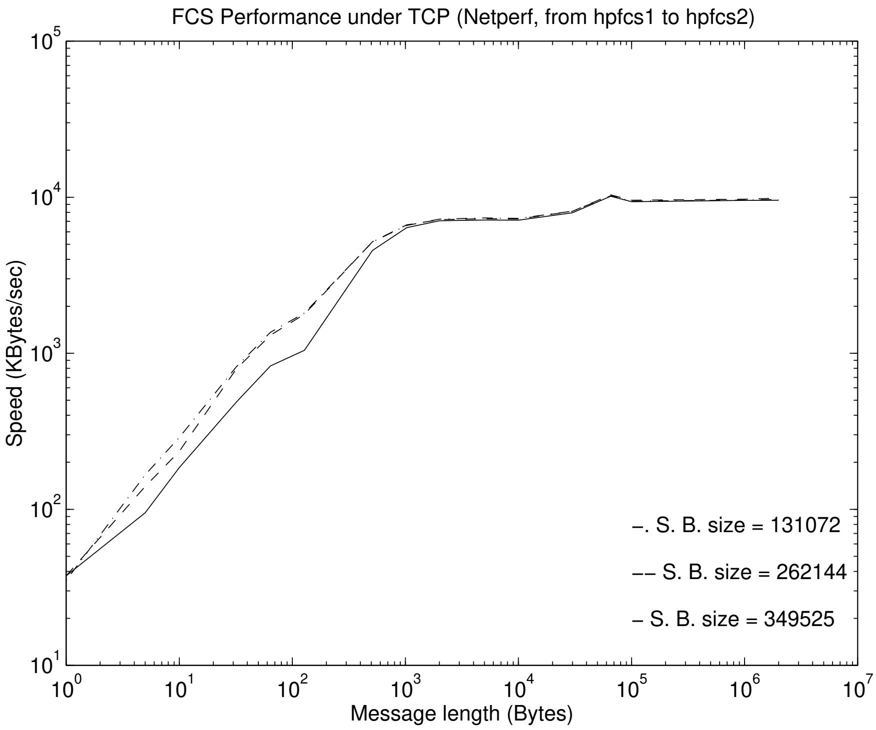

Figure 7a: Bandwidth performance from *hpfcs1* to *hpfcs2*.

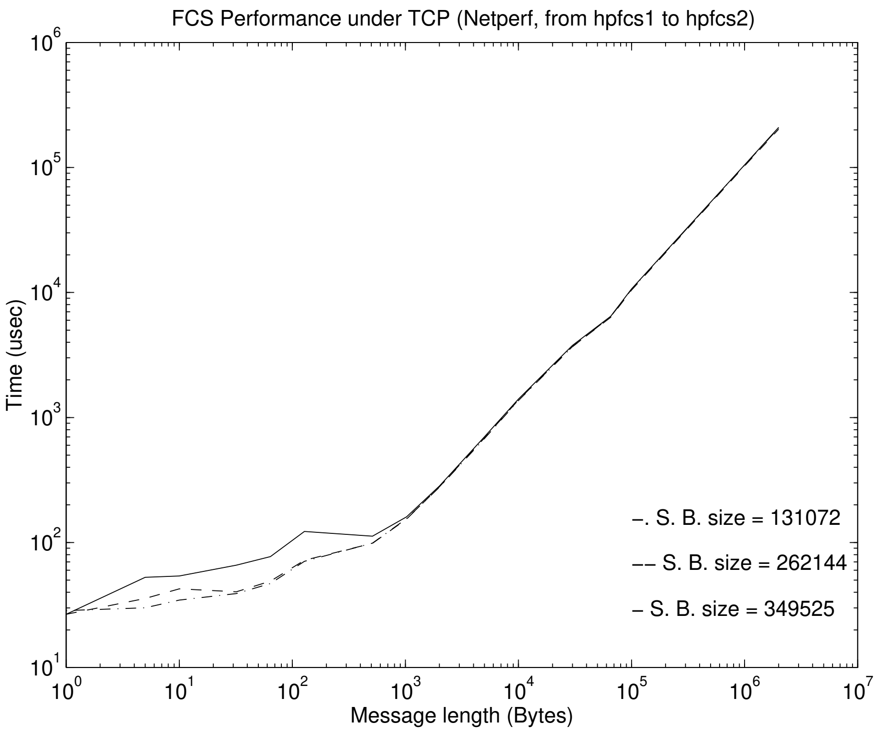

Figure 7b: Latency performance from *hpfcs1* to *hpfcs2*.

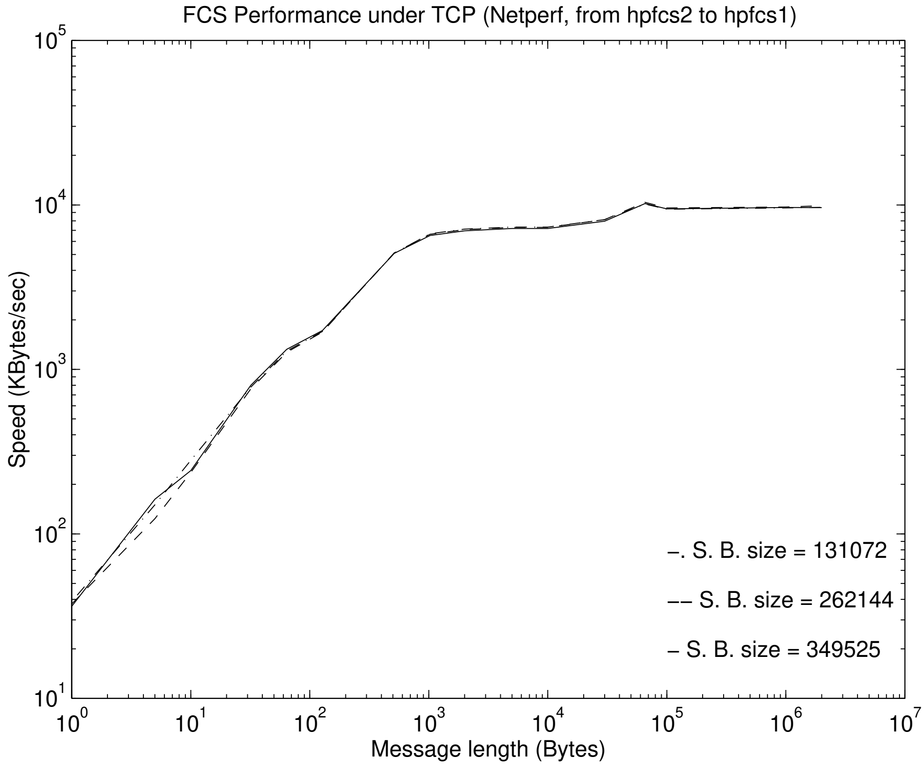

Figure 8a: Bandwidth performance from *hpfcs2* to *hpfcs1*.

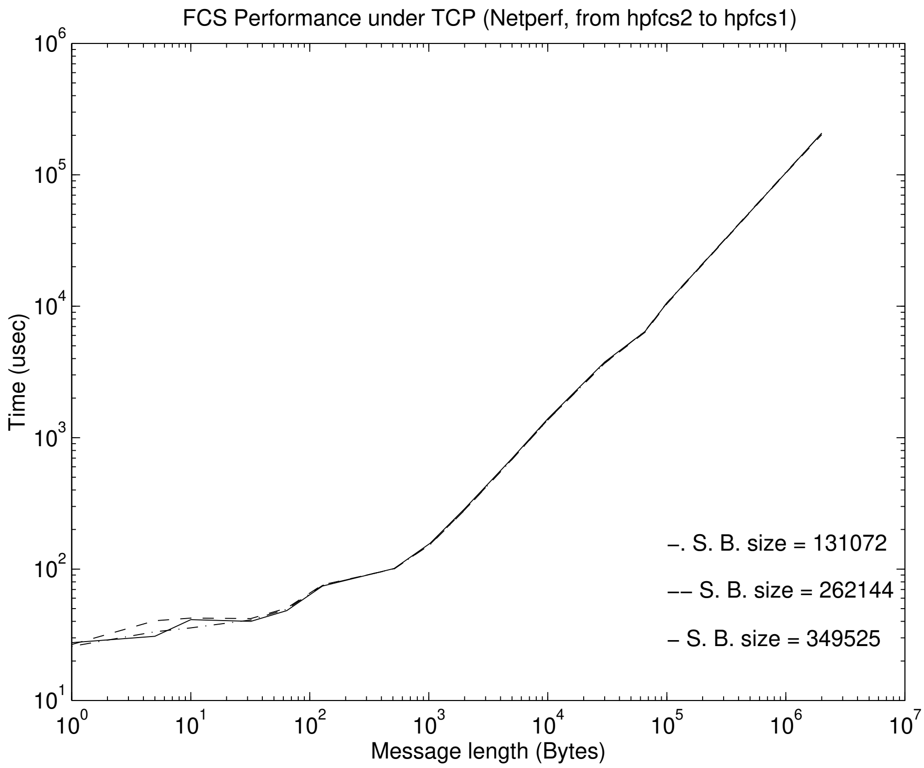

Figure 8b: Latency performance from *hpfcs2* to *hpfcs1*.

B. *TCP_no_delay* option set

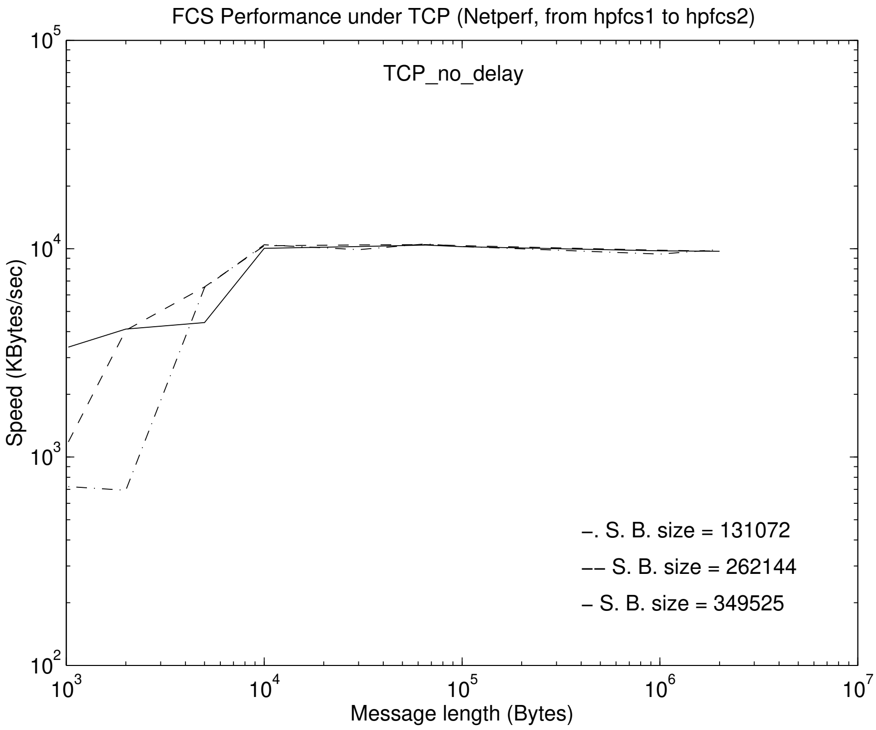

Figure 9a: Bandwidth performance from *hpfcs1* to *hpfcs2*.

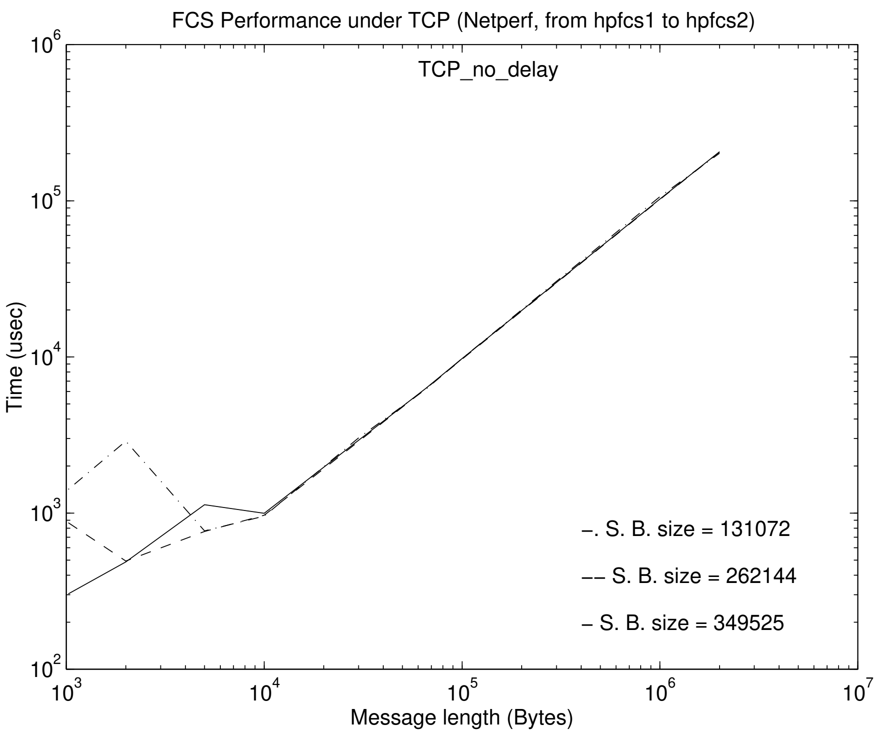

Figure 9b: Latency performance from *hpfcs1* to *hpfcs2*.

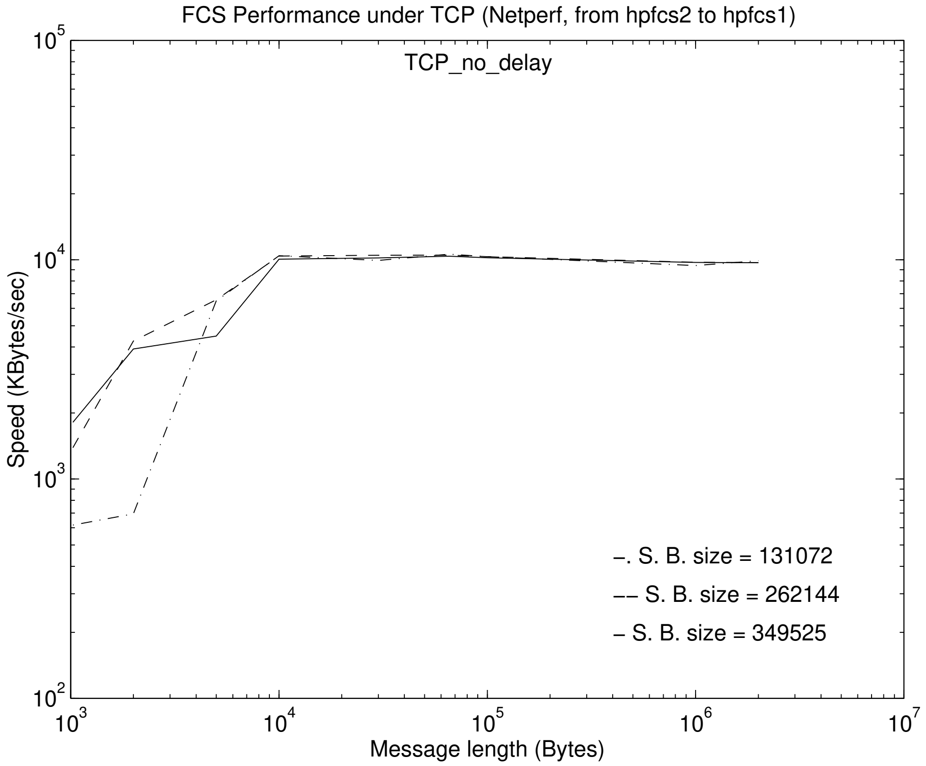

Figure 10a: Bandwidth performance from *hpfcs2* to *hpfcs1*.

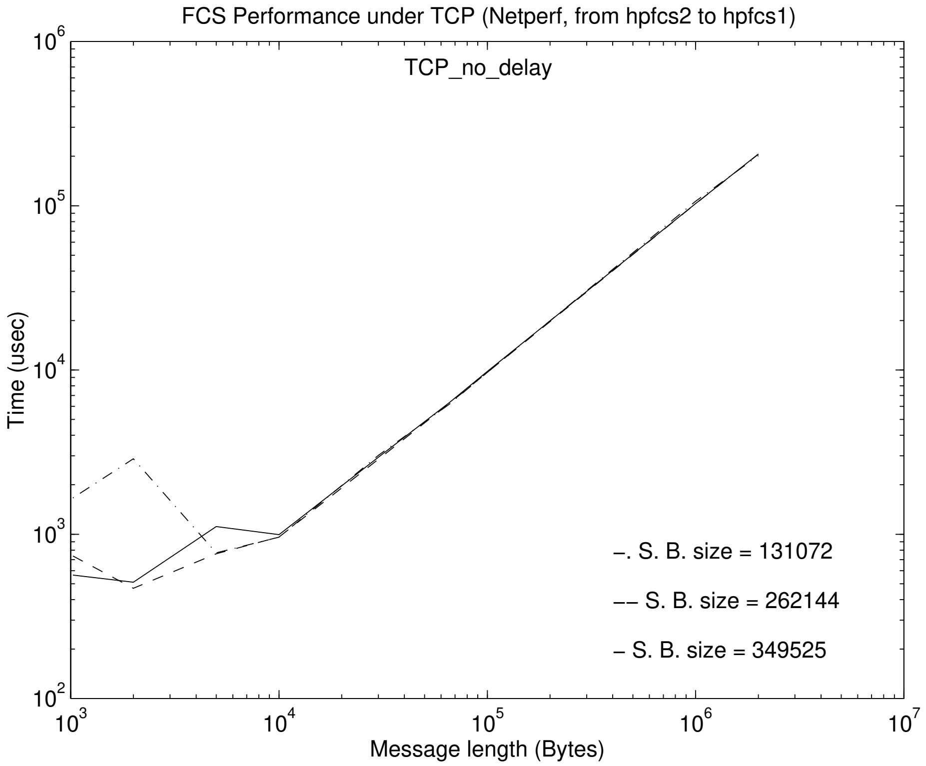

Figure 10b: Latency performance from *hpfcs2* to *hpfcs1*.

V. Communications between *IBMs* and *HPs*

A. *TCP_no_delay* option disabled

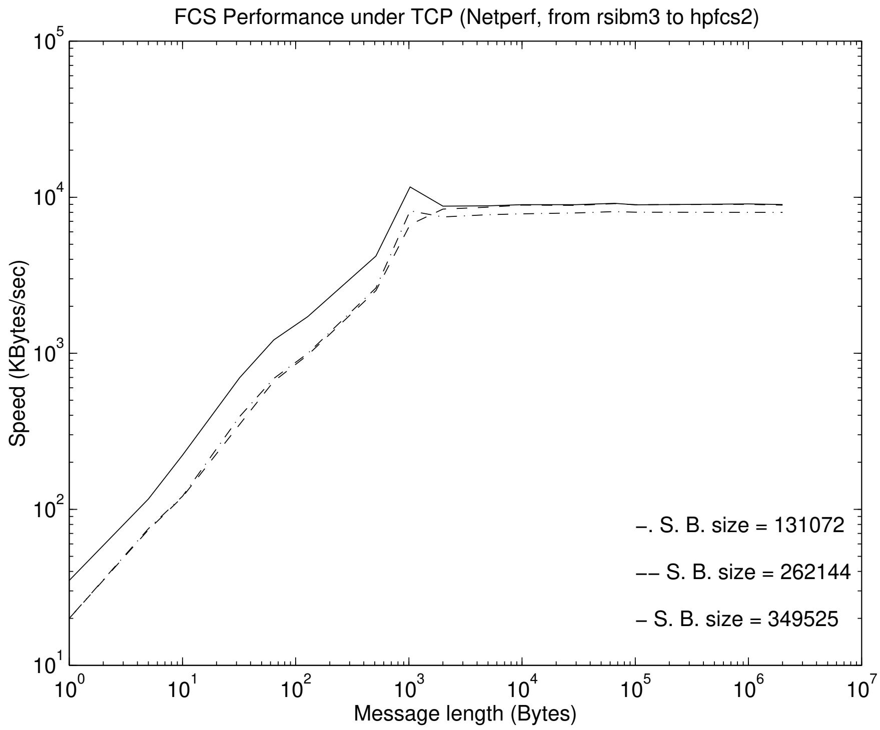

Figure 11a: Bandwidth performance from *rsibm3* to *hpfcs2*.

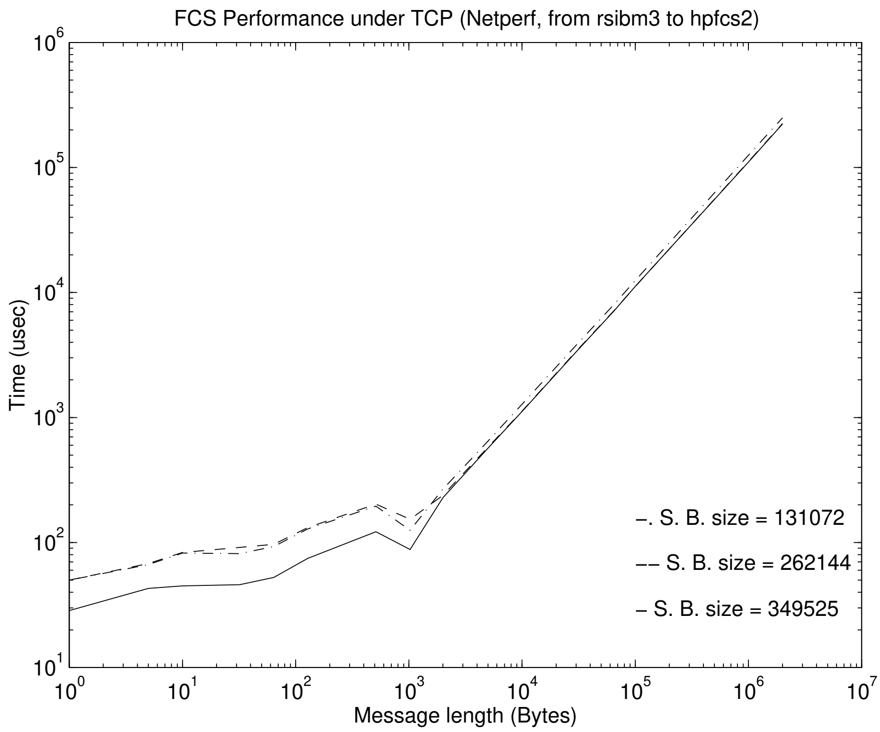

Figure 11b: Latency performance from *rsibm3* to *hpfcs2*.

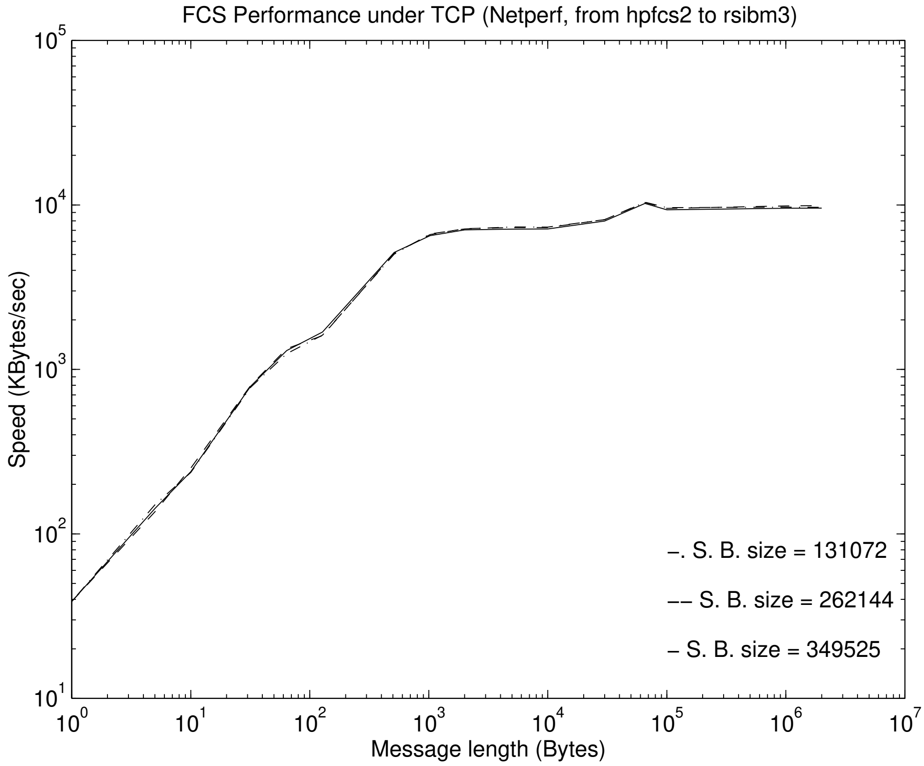

Figure 12a: Bandwidth performance from *hpfcs2* to *rsibm3*.

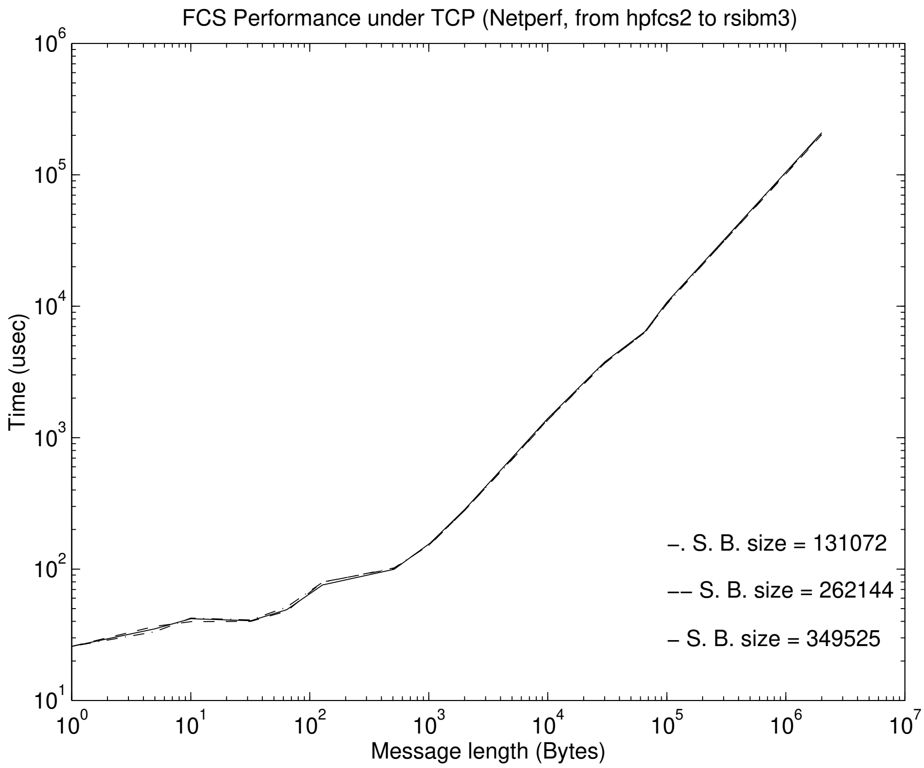

Figure 12b: Latency performance from *hpfcs2* to *rsibm3*.

B. *TCP_no_delay* option set

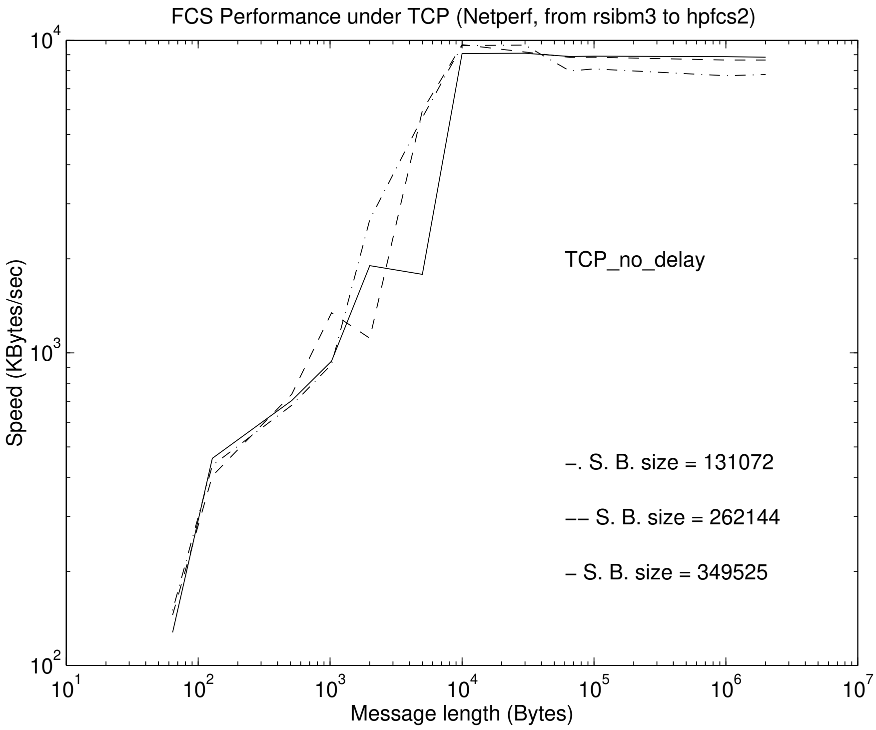

Figure 13a: Bandwidth performance from *rsibm3* to *hpfcs2*.

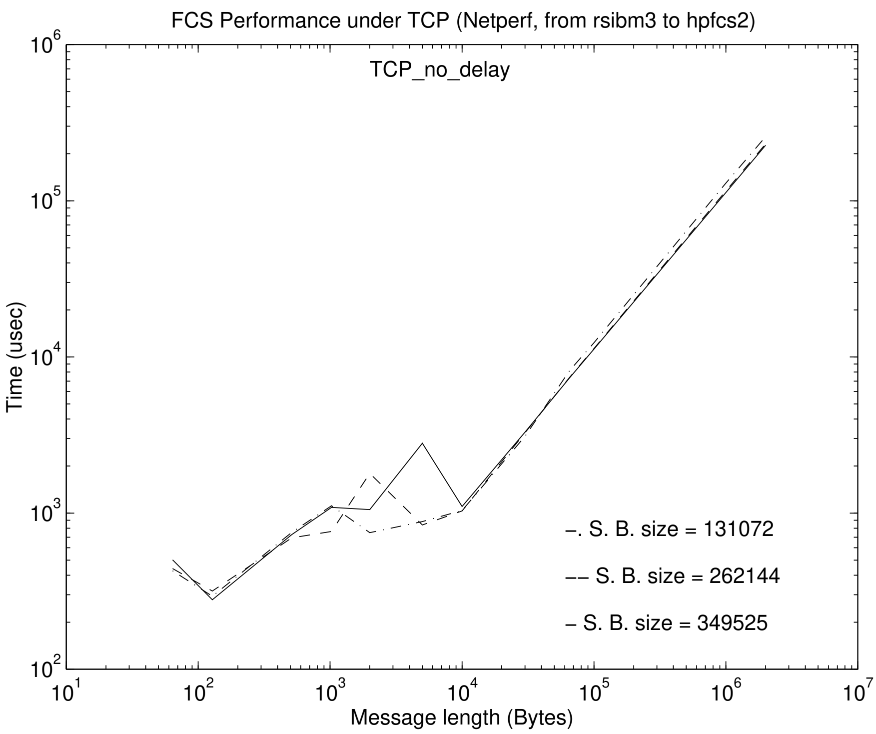

Figure 13b: Latency performance from *rsibm3* to *hpfcs2*.

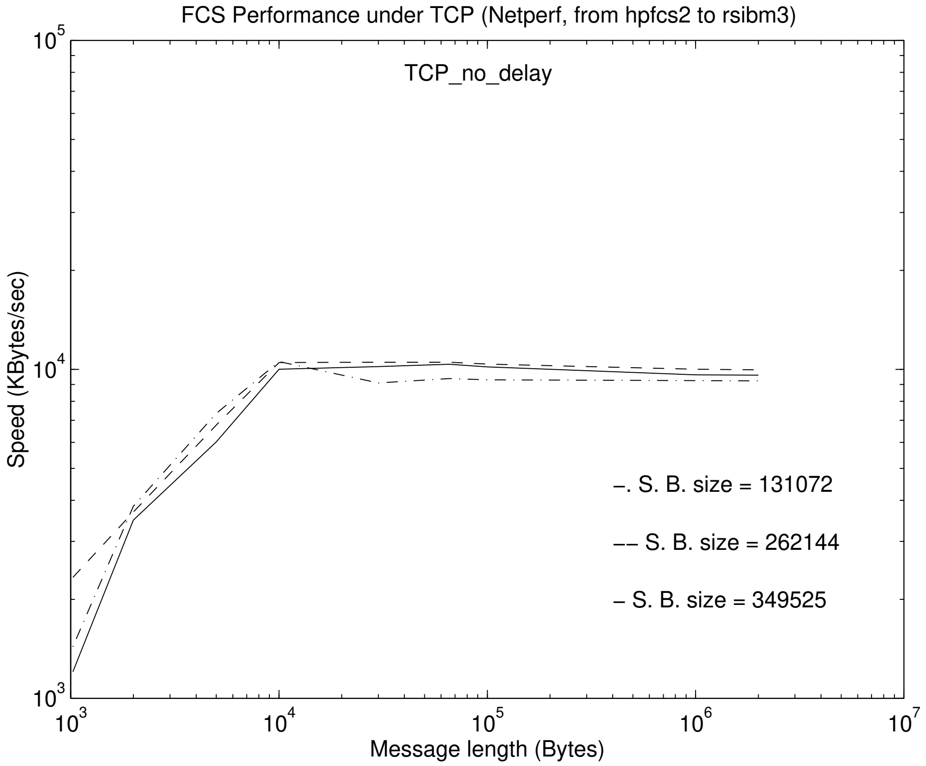

Figure 14a: Bandwidth performance from *hpfcs2* to *rsibm3*.

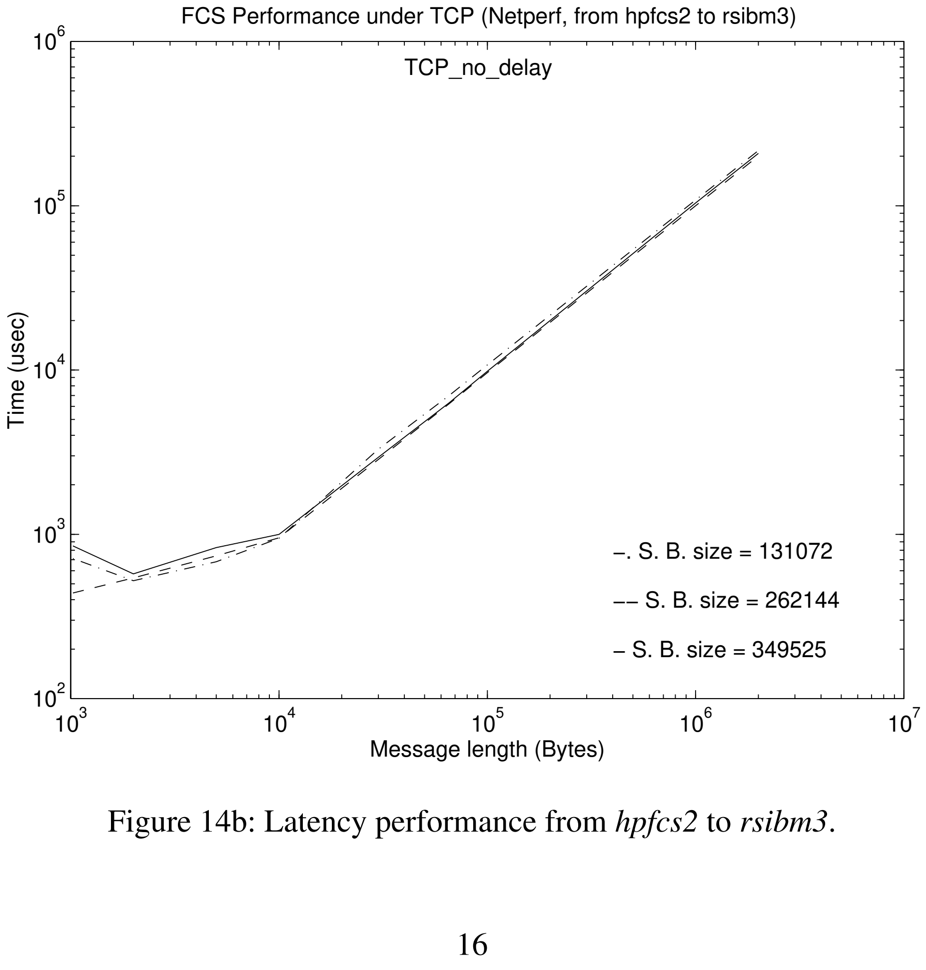

Figure 14b: Latency performance from *hpfcs2* to *rsibm3*.

VI. Conclusion

Switching *High Performance TCP/IP* on workstations is really worthwhile, because it provides substantial improvements on the performance for *TCP/IP* communications.
Setting *TCP_no_delay* option or not during the tests has practically no influence on the bandwidth, which is generally measured for large packets. On the contrary, this option must be set to ensure a realistic overhead measurement. Unfortunately, it was seldom possible to run '*Netperf*' with *TCP_no_delay* option set for transmitting small messages (less than 1 Kbytes), because it was running out of memory (the maximum available memory space reserved for the network on a node was exceeded).
The following is a summary of the performances we measured, according to the hardware configuration used:

- communications between *IBM* workstations: the overhead of the system is around 700 microseconds (measured with *TCP_no_delay* option set). When *TCP_no_delay* option was disabled, we measured an over optimistic value of 50 microseconds. The bandwidth can reach 11.25 Mbytes/s for transmitting 10^5 byte messages (for a 1 Mbyte socket buffer size).

- bi-directional communications between *IBM* workstations: a degradation of around 30% on average is reported, with regard to the expected performance.

- communications between *HP* workstations: the over optimistic overhead of the system is around 25 microseconds (measured without *TCP_no_delay* option set, because it ran out of memory), and the bandwidth can reach 10.5 Mbytes/s for transmitting 64 Kbyte messages (for a 349525 byte socket buffer size, which corresponds to the maximum size available with the current network configuration).

- communications between *RS/6000 590* and *HP/9000 700* workstations: the overhead of the system is around 25 microseconds (completely unrealistic because measured without *TCP_no_delay* option set), and the bandwidth can reach 10.3 Mbytes/s for transmitting 64 Kbyte messages (for a 349525 byte socket buffer size).

VII. References

[1] "Communications Benchmarks with the Ancor Fibre Channel Fabric", Fabrice Chantemargue, CERN/EAST note 94-22, July 28th 1994 (draft).
[2] "Communications Benchmarks with HPs and IBMs connected around the Ancor Fibre Channel Fabric", Fabrice Chantemargue, CERN/EAST note 94-24, August 29th 1994 (draft).
[3] Fibre Channel Association: "Fibre Channel: Connection to the Future", ISBN: 1-878707-19-1, 1994
[4] "CXT 250 16-port Switch", Installer's/User's manual, Publication N°. PUB 007 A Rev. D.
[5] "EISA CIM 250", Driver Interface Manual, Publication N° PUB 010 Rev. B.
[6] "MCA CIM 250", Installer's/User's Manual, Publication N° PUB 009 A Rev. Draft.

VIII. Appendix: How to configure the network parameters under *Aix O.S.* to run *High Performance TCP/IP*.

RS/6000 IBM workstations, running *Aix* version *3.2* Operating System, are implementing *High Performance TCP/IP*. The command "*no -a*" returns informations on the configuration network on a node. If the *RFC 1323* flag is set to *0*, that means *High Performance TCP/IP* is disabled. To enable it, just turn the "*rfc1323*" flag on to *1* by using the following command (as a super user): "*no -o rfc1323=1*".
To have a fully efficient network configuration, some other parameters need to be modified. The following list provides one configuration example that is currently working on our workstations:

"*no -o sb_max=6000000*"
"*no -o rfc1323=1*"
"*no -o thewall=6000*"
"*no -o lowclust=200*"
"*no -o mb_cl_hiwat=2000*"
"*no -o tcp_sendspace=655360*"
"*no -o tcp_recvspace=655360*"
"*no -o udp_sendspace=655360*"
"*no -o udp_recvspace=655360*"

There is no need to recompile the kernel nor to reboot the machine.