GG

CERN-CN-96-013

CN / 96 / 13

Su 9639

[redacted]

HIPPI in the CERN Computer Centre

Ben Segal, Jacek Kaim & Maria Dimou (CN-PDP/NS)
Gabor Gyori (OPAL)

Abstract

CORE, the Centrally Operated RISC Environment, which dominates the CERN central computing scene, relies on a scalable "Hybrid Backplane Network" that can support the appropriate data rates. The "Hybrid Backplane Network" consists of FDDI and UltraNet, which work in parallel providing high data rates and redundancy. The newest component of this Network - HIPPI, the High Performance Parallel Interface, was recently introduced in the CERN Computer Centre. HIPPI's role is to complement FDDI as a high bandwidth interconnect between powerful machines (especially SGI Power Challenges), provide a high speed tape staging link for powerful machines and allow CORE to phase out UltraNet.

This paper presents the results of tests, conducted at the CERN Computer Centre, trying to prove the feasibility of HIPPI as a high bandwidth interconnect, compare the performance of HIPPI to UltraNet and show the performance of HIPPI used for tape staging between SGI Power Challenges and high performance tape drives on the IBM SP/2.

A later phase of the tests involved two HIPPI to FDDI "GigaRouters". These permit high performance tape staging to large numbers of FDDI-based tape servers, as well as the support of a very high speed 10 km serial HIPPI link connecting the NA48 experiment to the Meiko CS2 machine in the Computer Centre.

The conclusion is that HIPPI is a highly viable solution for the demanding data-rate requirements posed by the CORE system.

26 August, 1996 CERN CN Report 96-13

1. Introduction

The tests were performed with small client-server applications. The server part is started on one machine and the client part on the other machine to be tested. The client establishes a TCP connection to the server using the socket interface. Parameters like buffer size, socket buffer size, user buffer size and amount of data to be transferred can be selected. The transfer gets initiated and at the end of the transfer statistics are presented.

Two different connection modes were used - a memory-to-memory pure TCP test and a memory-to-memory RFIO test. RFIO is the Remote File I/O protocol developed at CERN, which runs on top of TCP and allows secure bulk data transfers. RFIO is of special interest in this context, since it is currently used by the CORE system, and transfers using it constitute more than 90% of the data shipped on the "Hybrid Backbone Network". Good performance of RFIO transfers over HIPPI is therefore of great importance.

At the time the tests were performed the HIPPI setup in the CERN Computer Centre consisted of four SGI Challenges, the IBM SP/2 and the Meiko CS2. These machines were interconnected by parallel interfaces using a 16X16 non-blocking HIPPI switch from Avaika.

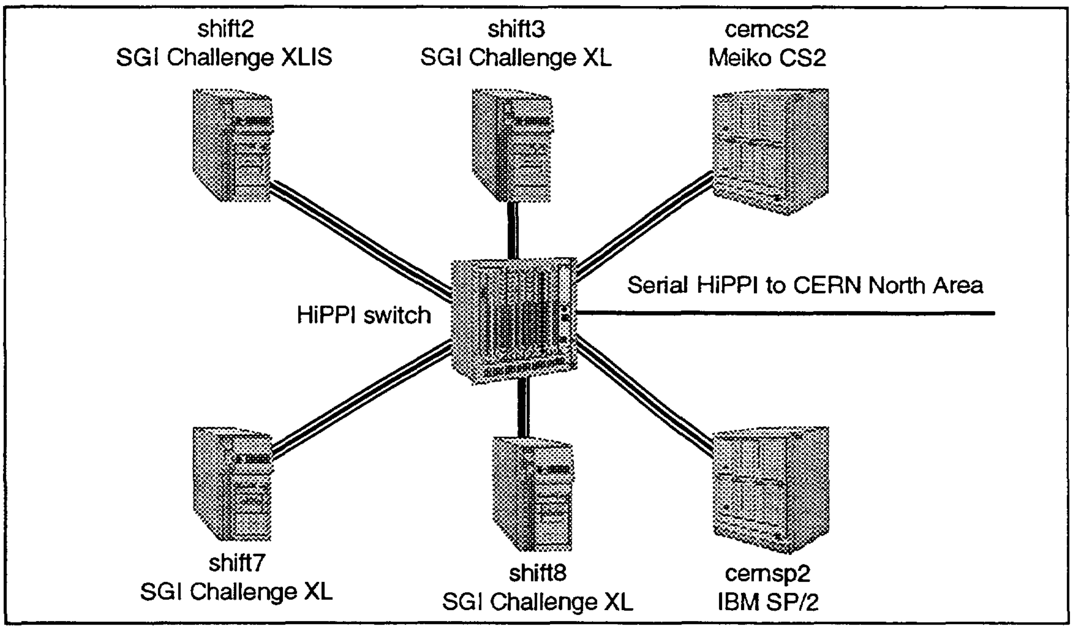

Fig.1 HIPPI setup in the CERN Computer Centre

Below is a table containing the technical specifications of the machines involved in the tests at the time of the tests.

Machine name	Manufacturer, Model	# CPUs	CPU manufacturer, model	CPU frequency
shift2	SGI, Challenge XLIS	8	MIPS, R4400	200MHz
shift3	SGI, Challenge XL	20	MIPS, R4400	200MHz
shift7	SGI, Challenge XL	12	MIPS, R4400	200MHz
shift8	SGI, Challenge XL	12	MIPS, R4400	200MHz
cernsp	IBM, SP/2	64	IBM, Power2	67MHz
cerncs2	Meiko, CS2	32 double	ROSS, HyperSPARC	100MHz

Fig.2 Test machine specifications

2. UltraNet pure TCP tests

A number of memory-to-memory pure TCP transfer tests were first conducted over UltraNet. The goal of these tests was to determine the maximum achievable performance of this network medium between two hosts running multiple concurrent TCP streams. Different measurements were obtained by varying the buffer size and the number of concurrent TCP streams. The tests were performed between two Silicon Graphics Challenge XL machines with 12 CPUs each. The kernel buffer size allocated to sockets (socket buffering) was set to 512kB.

In the tables and graphs below the results of these tests are presented. The columns represent the number of concurrent TCP streams and the rows represent the buffer sizes in kB. The values are the total bandwidth of the concurrent streams in kB/s.

	number of streams									
bufsize	1	2	3	4	5	6	7	8	9	10
1024	10240	11986	12280	12032	11928	11810	11741	12191	12171	11952
512	10240	11444	11165	11294	11072	11285	11740	11555	11442	11356
256	8533	11264	11161	11183	10781	11106	10811	10793	11653	11294
128	6826	11150	11245	10907	11060	11038	10946	11019	11145	11010
64	6023	8818	10802	11541	11533	11028	10659	10607	10609	10761
32	3657	5993	7633	9164	9260	9947	10168	9943	9704	10119
16	2560	4172	5354	6257	7246	8194	8457	8431	9075	9066
8	1706	2965	3551	4344	4718	4979	5223	5450	5815	5844

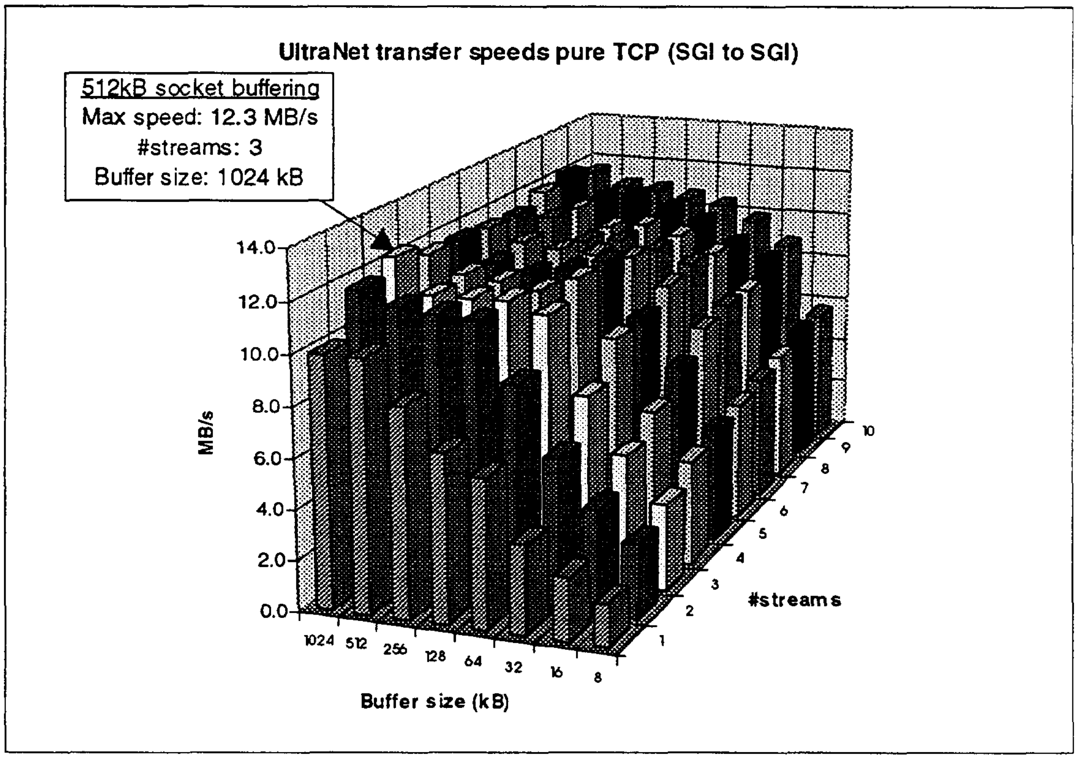

Fig. 3 Results of memory-to-memory pure TCP test over UltraNet

This graph shows clearly that the performance changes only slightly with the number of concurrent streams, and the buffer size has a substantial impact on the overall performance. At 128kB buffer size, which is the buffer size used by RFIO in production, and with more than one parallel stream, the performance reaches a level close to the maximum.

2. HIPPI pure TCP tests

The same memory-to-memory pure TCP transfer tests were now repeated using HIPPI as the network medium. The goal of these tests was again to determine the maximum achievable performance of the network medium running multiple concurrent TCP streams. Different measurements were obtained by varying the buffer size and the number of concurrent TCP streams. The tests were performed between the same two Silicon Graphics Challenge XL machines with 12 CPUs each.. The kernel buffer size allocated to sockets (socket buffering) was set again to 512kB.

In the tables and graphs below the results of these tests are presented. The columns represent the number of concurrent TCP streams and the rows represent the buffer sizes in kB. The values are the total bandwidth of the concurrent streams in kB/s.

	number of streams									
bufsize	1	2	3	4	5	6	7	8	9	10
1024	17066	26543	36988	42575	51505	56380	62054	66901	70249	73696
512	17066	28117	37888	46301	52128	58940	62357	68653	71805	77253
256	16516	28401	40178	48006	55976	62293	68065	72822	77540	81117
128	20480	34112	42861	57417	71094	62848	75064	79089	88025	91534
64	19692	38008	48489	59904	68568	76624	84053	84191	87038	87564
32	21333	36244	48565	59113	70201	77485	82933	87273	87442	87968
16	20480	33200	46255	61495	68070	78418	80996	86838	87616	86670

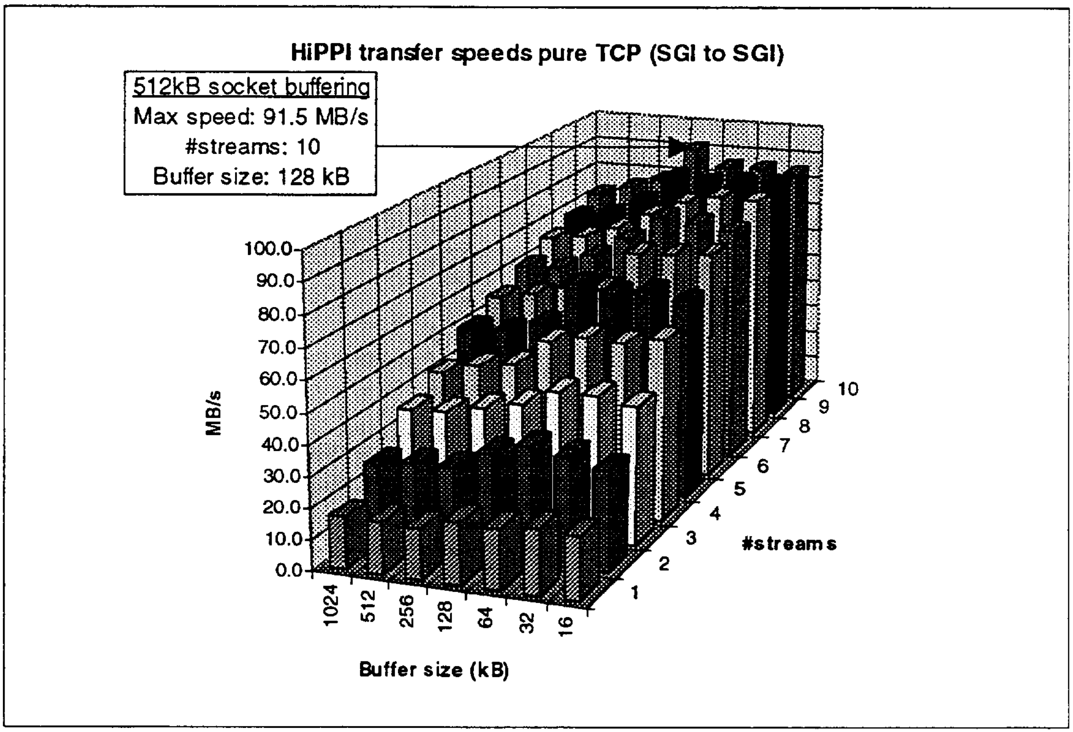

Fig. 4 Results of memory-to-memory pure TCP test over HIPPI

This graph shows a totally different behaviour of HIPPI compared to UltraNet. The transfer rates do not change significantly with changing buffer size. This can be explained by the TCP window size, which is limited to 64kB. This is not the case with UltraNet since it uses a proprietary ISO-like scheme with no such limitation. The HIPPI performance has become "latency limited". We can actually deduce the effective system latency to be approximately 3ms, which corresponds to the observed maximum transfer rate per stream of around 20MB/s. A workaround to this problem would be to use the Window Scaling Option of Extended TCP. Setting the TCP window size to a value of about 400kB, would allow to extend the maximum transfer rate per stream close to the media limit of 100MB/s.

For the same number of streams there is in fact a peak in performance at around 64-128kB buffer size. The overall performance scales with increasing number of streams. Note that the absolute peak of 91.5MB/s is only 8.5% less than the nominal maximum transfer speed achievable on HIPPI (800 Mb/s or 100 MB/s). This peak was achieved with ten concurrent streams and a buffer size of 128kB. With each stream consuming ~20% CPU power of a single MIPS 4400 processor, the total CPU consumption was around two full MIPS 4400 processors (of the twelve available) on each SGI Challenge.

3. HIPPI RFIO tests

Some measurements were done with a simplified version of RFIO software. The tests were performed between the same two Silicon Graphics Challenge XL machines through HIPPI. The simple RFIO program called "grfio" was linked with the Ultra libraries.

	number of streams									
bufsize	1	2	3	4	5	6	7	8	9	10
1024	10666	17767	23763	29237	34304	38652	39936	46203	48670	49797
512	2723	6964	9427	12297	14750	17159	19958	24109	25336	27705
256	7757	11868	16017	19050	21171	24684	26700	29300	30643	32381
128	6736	11807	16422	20824	25094	28632	31703	35428	37662	39650
64	320	640	960	1280	1592	1920	2240	2546	2792	3200
32	160	320	480	638	754	942	1117	1260	1437	1595
16	80	160	238	317	396	476	555	639	719	799

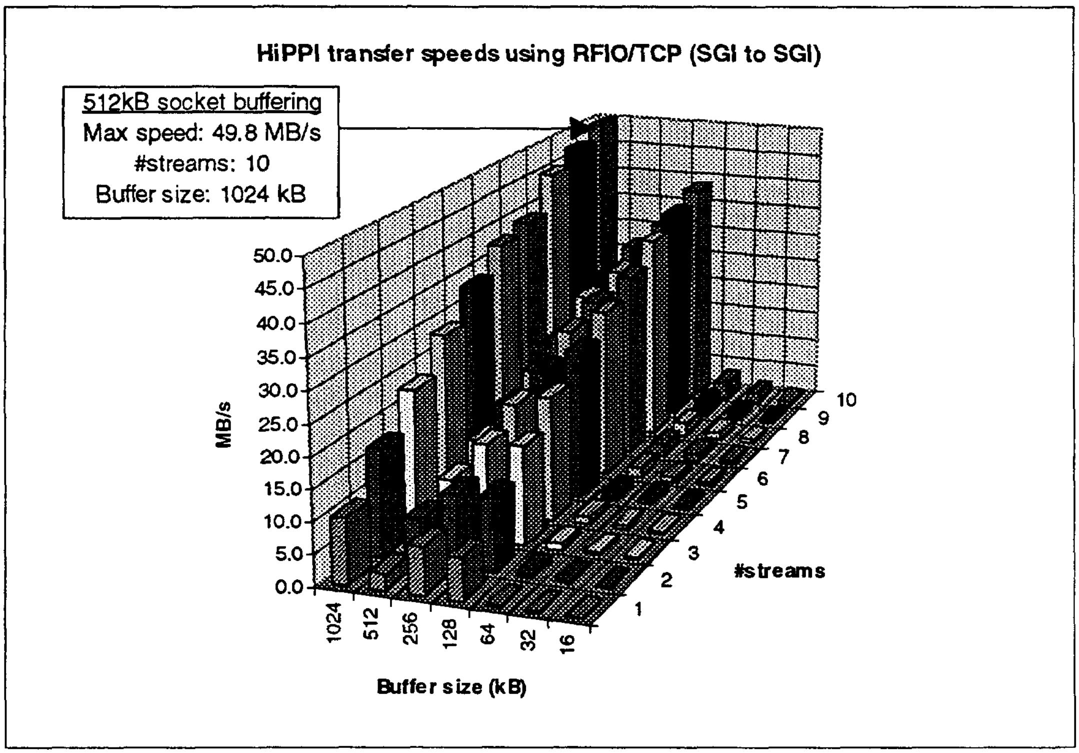

Fig. 5 Results of memory-to-memory RFIO test over HIPPI

What can be noted here is the extraordinary sensitivity of RFIO with respect to the chosen buffer size. Below the nominal 128kB, the performance is disastrous. At 128kB the performance reaches a peak, to descend again with increasing buffer size up to 512kB. Only at 1024kB the performance picks up again to reach a new peak. This behaviour is thought to be due to delayed transmission of RFIO ACK messages due to the TCP Nagle algorithm (discussed further below). Fortunately at the buffer size of 128kB - the standard RFIO buffer size, the performance is acceptable.

4. HIPPI RFIO tests to tape server (3590)

A number of tests were done to measure the performance of transfers between shift7 - a CPU server (SGI Challenge) and sp023 - a tape server (IBM SP/2 node). The transfer goes from the SGI Challenge to an IBM SP/2 gateway node - sp041. From there the transfer is routed to sp023 via the internal IBM SP/2 fast switch. The setup is described with the figure below:

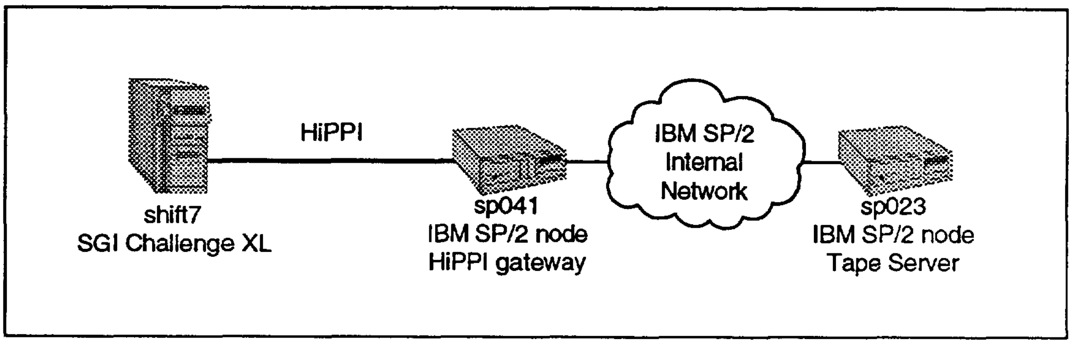

Fig. 6 Setup of memory-to-memory RFIO test over HIPPI to tape server.

The first test to be performed was a pure TCP writing test between shift7 and sp023. The receive queue-size on sp023 is set to 200kB. The amount of transferred data is 512MB. The buffer size is 128kB - the standard RFIO buffer size. Since both the SP/2 nodes are single CPU machines an interesting factor is the CPU consumption when running multiple concurrent streams. The results presented below are a geometrical mean of several measurements of transfers between shift7 and sp023. Tests performed in the other direction show very similar results.

#streams	speed (MB/s)	CPU consumption (% tot)		
		shift7	sp041	sp023
1	10.4	28.5	27.3	40.5
2	14.2	43.3	46.5	72.0
3	14.8	47.3	52.5	90.0

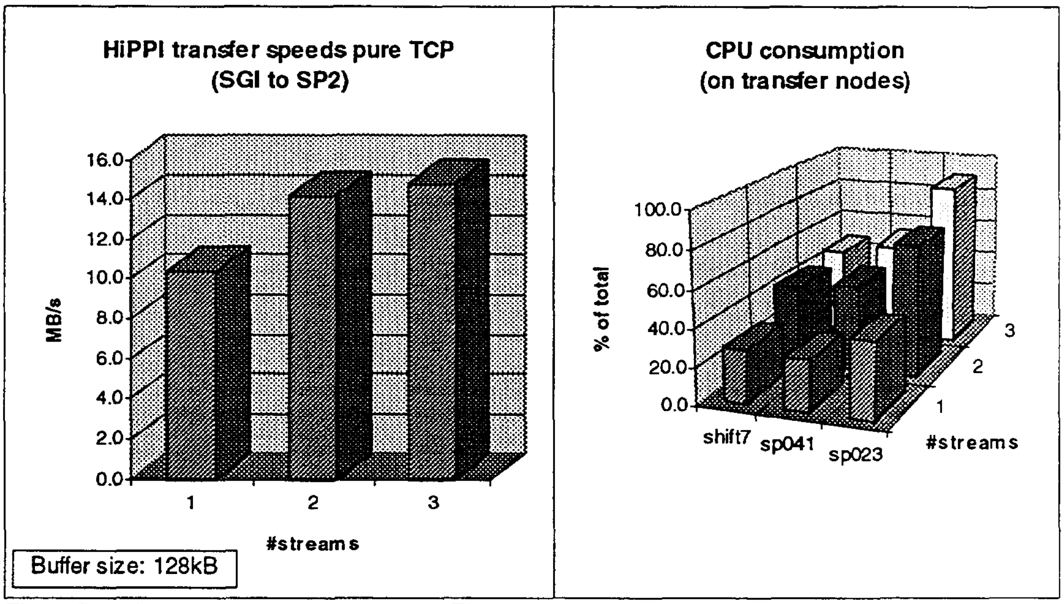

Fig. 7 Results of memory-to-memory RFIO test over HIPPI to tape server

The graphs show clearly that the bottleneck resides in the tape server node. When processing three parallel HIPPI streams the CPU consumption on sp023 reaches 90%, thus almost saturating the node. It can be noted that the gateway node supports the load somewhat better.

Other RFIO tests were performed to verify the buffer size sensitivity. The following is a RFIO writing test between shift7 and sp023. The size of the transferred file was 200MB, which is a typical size of staged files. The transfers were done using a single stream. Once again tests performed in the other direction show very similar results.

Buffer size	speed (MB/s)	CPU time (s)	
		shift7	sp023
1024	7.60	20.5	26.2
512	6.74	16.8	23.1
256	6.39	16.6	23.1
128	6.52	16.2	23.2

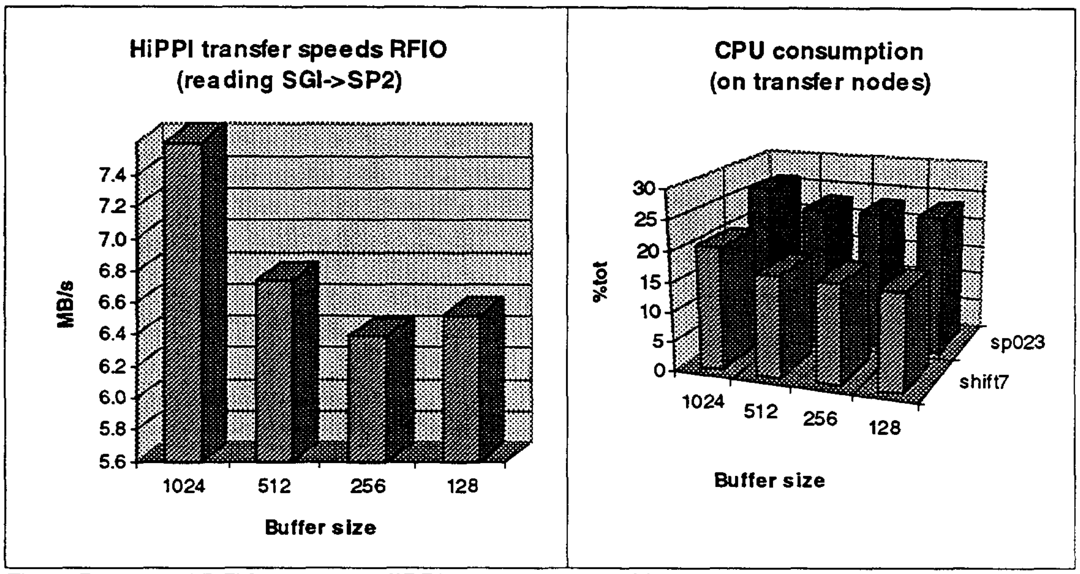

Fig. 8 Results of RFIO test over HIPPI to tape server with changing buffer size

This test confirms again the typical RFIO buffer size sensitivity. It can also be noted that CPU consumption on the involved nodes does not change greatly with changing buffer size.

5. IBM SP/2 internal switch tests

All the nodes in the IBM SP/2 machine are connected using a fast, proprietary, internal switch. For a time the switch was ridden by many problems. During tests performed on the switch we have got strange results that we could not explain. Later with the help of IBM engineers we have found out that the problems were caused by:

- physical problems with the switch itself
- problems during transition from AIX version 3 to version 4
  (nodes with different AIX versions could not communicate through the switch)
- problems with the choice of the IP window size
  (too small gives bad performance, too big causes crashes because of MBUF exhaustion)
- problems in the RFIO software caused by too small client buffer size for the switch
  (128kB gives very bad performance when reading, 1MB fixes the problem)
- problems caused by the "Nagle Algorithm" in TCP/IP
  (while trying to increase the amount of data sent inside large MTU packets, it introduces a latency of 200ms for small messages such as RFIO ACK's)

The correction of all these problems is not yet finished, but some tests performed with improved software show that the combination of HIPPI and IBM SP/2 internal switch outperforms FDDI. These tests were done writing data to NTP tapes and then reading it back again. The NTP tape servers are IBM SP/2 nodes attached to both the internal switch and FDDI. The disk server used in the tests was an SGI Challenge XLIS (shift2), which has both HIPPI and FDDI attachments. By switching the routes we could either use FDDI or the combination of HIPPI and the IBM SP/2 internal switch. The performance could also be improved by increasing the "Tape blocksize" from 32kB to 128kB. Below is a summary of the results:

Direction	Media	Tape blocksize	Transfer rate
read	FDDI	32kB	2.3MB/s
read	HIPPI & SP/2 switch	32kB	2.9MB/s
write	HIPPI & SP/2 switch	128kB	4.6MB/s
read	HIPPI & SP/2 switch	128kB	3.3MB/s

Fig.9 NTP test results

6. Introduction of a GigaRouter

Later on in the HIPPI project at CERN a new device was introduced, namely a Netstar GigaRouter. The GigaRouter can route between different media, including FDDI and HIPPI. It uses a high speed, non-blocking, media independent internal crossbar switch which allows to route for example a HIPPI connection onto multiple FDDI rings.

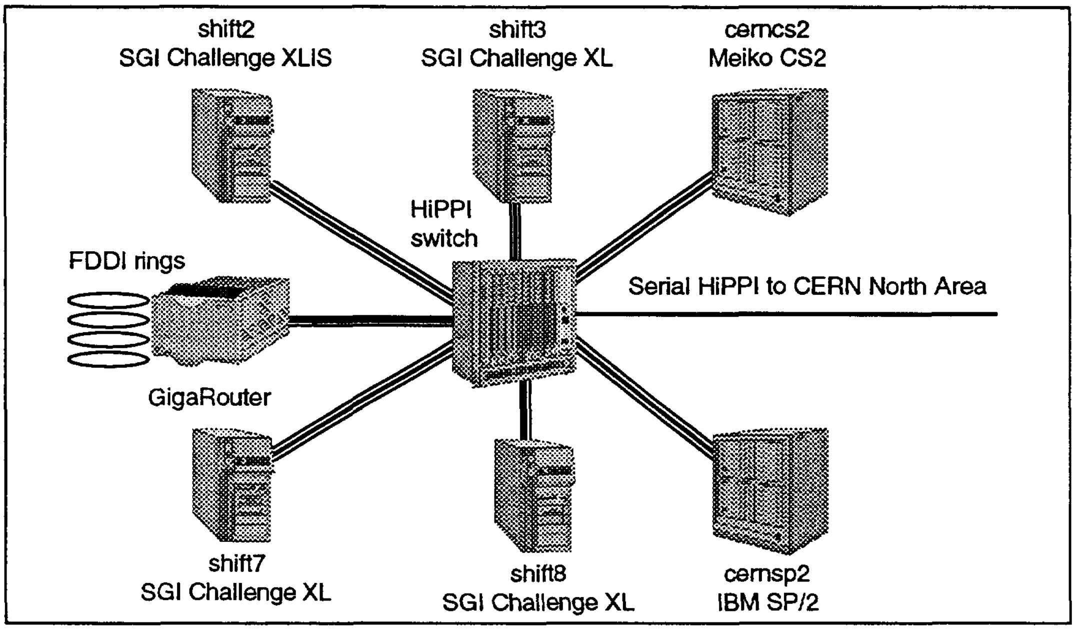

Fig.10 HIPPI setup in the CERN Computer Centre with addition of a GigaRouter

The idea behind the use of this first CERN GigaRouter was to allow powerful multiprocessor machines like the SGI Challenges to access through HIPPI a large set of service machines in CORE that only have interfaces on FDDI. Since a HIPPI interface allows an eight times higher total bandwidth than FDDI (800Mb/s vs 100Mb/s) a multiprocessor machine can thus run many more concurrent streams to FDDI attached machines than would be possible via its direct FDDI interface to CORE.

This idea is currently used in the CERN Computer Centre to allow its most powerful machines like the SGI Challenge and DEC TurboLaser to access a set of about 16 DLT Tape Servers (themselves connected only via FDDI) through HIPPI. The DLT Servers are distributed evenly over 4 independent FDDI rings, which are subnets of the main Class C CORE FDDI network. The design ensures that no FDDI bandwidth bottlenecks are created even when all DLT tape drives on all Servers are active. Of course other client machines without HIPPI interfaces must also be able to access the DLT Servers. This was taken into account by providing the DLT Tape Servers with dual access: to HIPPI via the GigaRouter and directly to FDDI clients via the standard connection (the CORE DEC GigaSwitch). Below is the logical design of the interconnect (Figure 11).

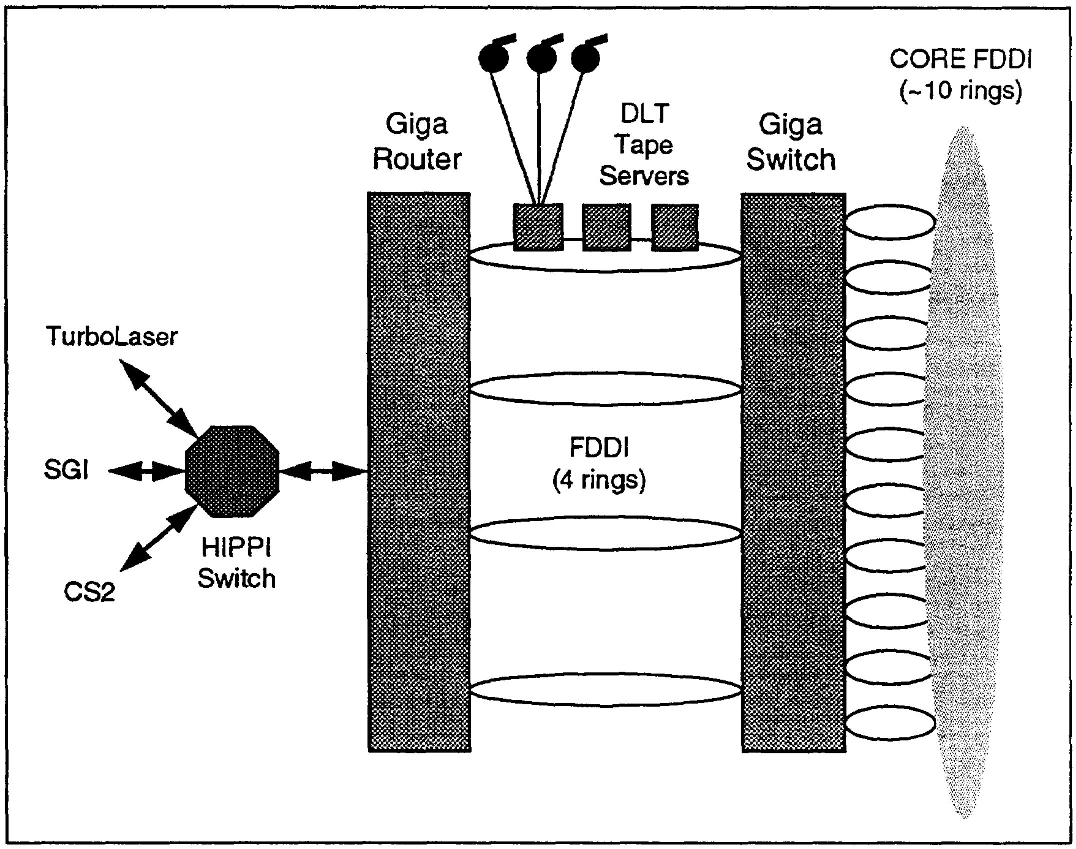

Fig.11 Logical interconnect for DLR access through HIPPI and FDDI.

Note that (although not shown in Fig.11) the HIPPI-attached machines are also connected directly to CORE FDDI, and can thus access the DLT Servers either through HIPPI/FDDI (GigaRouter) or directly via FDDI (GigaSwitch). This arrangement gives us a fallback solution for these powerful machines in the case HIPPI or the GigaRouter is unavailable.

7. HIPPI connection to the CERN North Area

Recently a connection to the CERN North Area has been achieved using serial HIPPI (running over a single pair of monomode fibres). As a leading part of the Central Data Recording project, this link allows NA48 physicists to transfer large amounts of on-line data from their experimental and test beam areas to the Meiko CS2 machine in the Computer Centre for processing and archiving. Data generated on the NA48 FDDI-attached "Front-End Workstations (FEWs)" is transferred through parallel FDDI rings to a second CERN GigaRouter. From there it is sent down the serial HIPPI connection, over 10 km long, to the Computer Center where it can enter the CS2 (or other CORE machines) either by HIPPI or via another set of 4 dedicated FDDI rings that have been added to the first GigaRouter for this purpose. The required sustained data rate in 1997 is 20 MB/s, but considerably less in 1996. (Currently the CS2's HIPPI performance is not adequate to handle the necesssary data rate, so the FDDI solution is used, connecting to 4 independent FDDI interfaces on the CS2). This setup is described in the figure below (Figure 12).

Initial tests of the link throughput have shown that 19 MB/s can be achieved today using 4 parallel TCP streams distributed among 3 FEWs and 2 CS2 nodes. This corresponds to approximately 5000 4KB packets/s, which is known to be the current limit for the current GigaRouter HIPPI implementation. A fourth FEW and 2 more CS2 nodes will soon be connected, and the link throughput is expected to reach 40 MB/s when firmware improvements are made to the GigaRouter HIPPI interfaces. The NA48 experiment is already using this link in production mode, with real data rates between 5 and 10 MB/s being produced for extended periods.

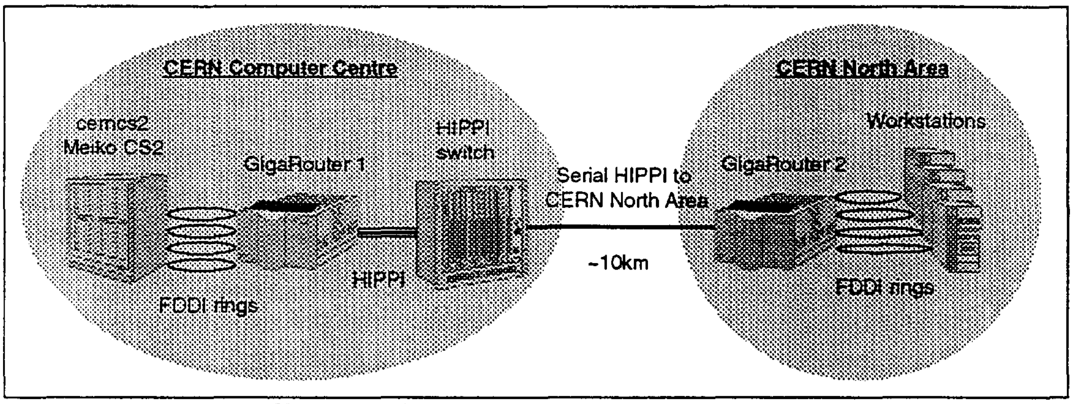

Fig.12 Connection to CERN North Area through serial HIPPI.

8. Conclusions

The HIPPI tests described in this paper have been a success. It has been shown that an aggregate transfer rate of 91.5MB/s can be achieved over HIPPI between SGI hosts using pure TCP, which is only 8.5MB/s lower than the nominal maximum transfer speed for the medium, and by far higher than the highest transfer rate achieved over UltraNet.

It has also been shown that acceptable speeds can be obtained over HIPPI using RFIO, without any modifications to the software. A buffer size sensitivity of RFIO has however been discovered, believed to be an unfortunate interaction of the Nagle algorithm and the RFIO protocol when ACK packets are used; fortunately, at the buffer size at which RFIO normally operates, this is not extremely serious. Nevertheless, the RFIO protocol is currently being modified to mimimise its effects.

Furthermore, it has been demonstrated that tape transfers using HIPPI and the IBM SP/2 internal network can be performed, achieving good transfer rates. The bottleneck has been found to be CPU limitation of the tape server node and an upper limit of three concurrent transfer streams has been established.

New developments, like the addition of two GigaRouters and a 10km link to the CERN North Area through serial HIPPI, have been introduced into the HIPPI project in the CERN Computer Centre, expanding its network capabilities. This allows us to integrate HIPPI tightly with the other CORE network media, making it a full member of CORE's "Hybrid Backbone Network".

9. Acknowledgements

The authors wish to acknowledge the essential contributions of Arie van Praag and Patrick Donnat (CERN-ECP) in setting up the HIPPI infrastructure, both in the CERN Computer Centre and for the 10 km serial link to the North Area. We also thank Felix Hassine (CN-PDP) and Tim Bell (IBM) for their help in understanding and overcoming RFIO problems encountered in the HIPPI and SP2 switch environments. Finally we thank Bernd Panzer (CN-PDP) for help in running tests between NA48 and the Meiko CS2.