B(its) T(o the) U(ser): a Communication Benchmark Proposal

Kurt Maly[1] Ajay Gupta Satish Mynam Sanjay Khanna[2]
Department of Computer Science
Old Domininion University
Norfolk, Virginia 23529-0162
June, 1995.

Abstract

In this paper we propose a communication benchmark suite: BTU(Bits To the User). Our benchmark does not test the maximum performance of a workstation in a homogeneous network under ideal circumstances. Rather, we take a vendor supplied workstation running UNIX[3] operating system, install our benchmark program and connect the workstation to a blackbox testbed which emulates a LAN connected to a WAN. The benchmark run will submit the workstation to a carefully designed combination of tests. The result is an indicator of what the user, at the application level, can expect in terms of bits sent to or received from a remote host. Our benchmark, in contrast to maximum performance tests, takes into account concurrent activities, such as CPU and I/O activities, which compete for resources on the test machine and concurrent activities on the network which will interfere with the test machine's communication. On the one hand we intend these results for the workstation buyer who can use the data together with the configuration specification and the list price to make a well reasoned decision if the workstation meets the needs when compared to the results of other workstations. We also believe these results can provide insights to the architects as to the location and nature of the communication bottlenecks of a particular workstation. We present the results of applying the BTU benchmark to four workstations from different vendors which shows that neither price nor SPECmarks are a good predictor for communication performance as measured by BTUs. We conclude this paper with a discussion of our plans for having the community accept this proposal as a communication benchmark

Keywords: Communication Benchmark, TCP/IP, Ethernet, ATM, FDDI, SPECmarks.

1 Introduction

Who needs yet another benchmark suite? First, there is no benchmark suite which assesses the effectiveness of the way high performance workstations and PCs communicate with other hosts. Second - and not all users realize that - more and more computing involves access of a network. That has long been true for Unix workstations but is becoming increasingly true for PCs. The client/server computing paradigm is becoming pervasive, distributed computing's importance is increasing, and information accessed over the Internet is becoming a dominant application for workstations and PCs(for the rest of the paper we shall use the term workstation to mean both).

What should we measure in a communication benchmark geared to the user? Since the objective of our proposed benchmark is to measure user satisfaction, we propose to rank workstations on how fast they deliver bits to a user application when those bits' sources or destinations are external hosts.

What conclusions can a user draw from a benchmark ranking and what not? The most important thing to observe is that communication time depends on two factors: bit network flow rate on the physical channel and bit host processing rate inside the host from the time bits arrive at the interface and are handed over to the application. The first factor depends on whether the physical network is a 9.6K baud modem line, an Ethernet, an 155 Mb/s ATM network, how many routers are between source and destination and similar characteristics; all are totally independent of the the workstation. The second factor depends on such things as the protocol stack used, the bus architecture , and the operating system. If the network flow rate is small compared to the bit host processing rate the user will probably see not much difference in total communication time for

[1] Technical Contact: Kurt Maly, (804) 683-4817, Fax:(804) 683-4900, Internet: maly@cs.odu.edu.
[2] IBM,NC
[3] TM of USL, Inc.

different workstations even if they are ranked differently by the benchmark. If however the network flow rate is sufficiently high (say > T1) than the host processing time becomes the dominant aspect of the communication time and the ranking will be relevant. Consider workstation A rated at 2.5 BTUs (2.5 Mb/s to the user) and workstation B rated at 1.6 BTUs, both executing Mosaic (installed on a server in an Ethernet LAN) to access a page in England (with a 56Kb link somewhere in between). Bringing up a Mosaic window involves intensive network communication between the server and the client and might take 20s by workstation A but will take about 30s on workstation B. However there will be not much difference in retrieving the page from England because the dominant factor will be the slow network flow rate. Another example would be FTP over a T1, one would expect workstation A to do significantly better because host processing is the dominant factor.

Workstations exhibit ever increasing performance when measured in benchmark tests such as SPEC and these workstations commonly connect to the network which have a bandwidth of 10Mb/s and more. Considerable effort is being spent on designing architectures, both software and hardware, which transfer bits to and from workstation memory from and to the network efficiently. However, no benchmark tests exist which enable us to judge the effectiveness of these developments. A number of studies do exist which measure maximum performance of individual workstations. Typically these studies use ttcp and experiment with various system parameters such as buffer management [10] and window sizes. Invariably, these studies are designed to elicit under which parameter configuration the workstation can push the most data to the network. Although quite useful in themselves these studies do not tell us what we can expect in terms of data in and out of the machine at the user level when competing processes are present on the workstation and the network is loaded with contending traffic.

In this paper we address the lack of such a communication benchmark by proposing a benchmark suite with the following properties:

- it will provide a single number reflecting the performance of a number of applications in typical network environments with typical concurrent activities on a host machine
- it provides a specification for the testbed together with source code for replication of the test results
- it provides summary information on performance for sets of applications(characterized by their message size used in communication) together with resultant penalties on concurrent activities caused by communication activities
- it provides detailed data on the performance of individual components of the benchmark test for the knowledgeable user and software and hardware architects.

In Section 2 of this paper we give some of the background on benchmarking in general and performance testing; in Section 3 we outline our approach to the design of our proposed benchmark test suite . Section 4 describes the implementation and the automation process we developed.. Section 5 provides a discussion of the statistical methods involved and the results of applying our benchmark to a set of machines of various vendors with varying operating systems and configurations(SUN,DEC[26, 36, 37, 38],SGI, PC) over a fixed network(Ethernet [25]). We conclude with a discussion and outline of future work(ATM, FDDI, T based 100 Ethernet) [5, 18] and possible models for operating communication benchmarking in Section 6.

2 Background

The Parallel Kernels and Benchmarks Committee, PARKBENCH, was founded at SuperComputing 1992 with an objective to establish a comprehensive set of parallel benchmarks which is generally accepted by vendors and users. This set of parallel benchmarks focuses on avoiding duplication of efforts and sets a standard for benchmark technology and makes results of benchmarks tests freely available in the public domain [2]. The PARKBENCH concentrates mainly on CPU performance and includes only one component on network performance. It is a test of how well a machine can send data to another host while varying the message length.

We identified a number of programs and suites [9, 43, 41, 42, 40] of MACH OS [17] which emphasized either low level system calls or studied CPU intensive applications like SPECINT [39], CINT 2.0 and CFP 2.0 [8] and as such are only relevant as far as general benchmark principles are concerned. Other such general studies are Benchmark Characterization [16, 12, 15]. A great deal of work has been done in determining the performance of networking system. Netperf [33] is a public domain database containing networking performance data of various computer hardware and software. Also, there have been results reported for transport protocols (e.g. TCP) [24, 14, 13, 35, 6] over various media (e.g. Ethernet, ATM [5]) and tests on Network File Servers (NFS)

[20]. Some of the work reported in the above articles was done when there were frequent complaints of slow performance of networks systems, some were reported by the vendors. All of these used and defined a large number of inconsistent terms. RFC 1242 [7] attempts to define a set of terminology that vendors can use to measure and report performance of network systems. This should provide users with comparable data from different vendors. The emphasis in all these performance studies is on maximum performance under ideal conditions and no correlation is given with background network, CPU [1, 21, 22], and IO activities.

3 Approach

The difficulty with designing a benchmark suite for communication is the many factors which influence overall performance of interprocess communication. For example, communication is influenced by the media access protocol, the bus architecture of the workstation, the implementation of the higher level protocols, the operating system and most importantly by activities occurring concurrently on the host machine and the network. No benchmark can test all the combinations of these factors and we were forced to make choices in our design. First and foremost we decided on a most common situation in the networking environment: UNIX workstations running TCP/IP in a LAN with connections to other hosts in a MAN or WAN. Secondly, we decided early on to concentrate our first effort on a widely distributed network, albeit slow speed: Ethernet. Our intention is to extend the benchmark suite to work for highspeed networks(ATM,FDDI, Fast Ethernet) as well. Thirdly, since this is a proposal for a communication benchmark we limited our study to a small number of well known vendors in the workstation arena: SGI, DEC, SUN, and Intel processor based personal computers. For applications competing for resources we considered ftp, telnet, rlogin, X windows, ttcp, CPU intensive and disk I/O intensive applications. Applications which are CPU intensive we considered are software compressed video and realtime audio applications. Bulk Data transfer applications such as client/server distributed databases, storage and retrieval of multimedia streams from storage devices come under I/O intensive and network intensive applications.

No judgement or explanation for the result is meant to be given; it is to be an unbiased look at a workstation; clearly the configuration(OS, memory,..) will influence the test results. We intend to serve the user of a workstation and provide data on how a particularly configured workstation can be expected to perform in a realistic network environment. The information should enable the users to make a reasonable judgement when acquiring a workstation with a specific configuration within certain cost constraints. Thus the results of the benchmark test are given together with a detailed specification of the test machine and its list price. The second audience group we are addressing are the hardware and software designers of workstations which can use our results to fine tune their operating system and identify hardware and software bottlenecks in the architecture of their workstations. To satisfy both audiences we provide two sets of results, one, a greatly abstracted set: the bit rate a user(BTU) can expect to communicate in a realistic setting for various message lengths and the penalty incurred by combining net with host activities; the second set is a detailed tabulation of each component's execution performance.

Briefly, our approach is to have workstations from various vendors, preloaded with their implementation of UNIX operating systems and TCP/IP, subjected to our benchmark one by one. The benchmark will create a controlled, fair and replicable test environment. It connsists of two major components: the Benchmark Testbed and the Benchmark Suite.

The benchmark testbed: it is used to emulate a communications environment. To emulate a communication environment with a slow link somewhere in its route, the testbed will introduce pre-computed packet delays. Similarly, to emulate an environments which loses packets , the testbed will drop packets with a pre-computed probability.

The internals of our testbed are shown in Figure 1. It consists of the following major components -

1. Test Workstation (W_{test}): This is the workstation which is being tested and monitored to assess its performance. Workstations from various vendors will be used here.

2. Active Black Box Workstation (W_{active}): This workstation has its operating system kernel modified such that it emulates various communication environments including packet loss and delay. Network traffic is generated between W_{active} and W_{test} in each of the cases specified in Benchmark Suite section. This Black Box is called active because of its inherent functionality namely, generating load between itself and W_{test}. Almost all the network activity of the benchmark is caused by W_{active} except for the two Black

Boxes participating in the tests together(as described in cases 5 and 6 of benchmark suit in Section 4). At the end of the test W_{active} post-processes the raw results to produce the various tables.

3. Passive Black Box Workstation ($W_{passive}$): This workstation is used to generate background network traffic between W_{test} and itself and is used to emulate simultaneous sending and receiving from two competing workstations to and from W_{test}. $W_{passive}$ also has its kernel modified to allow for WAN emulation for the multiple connections to W_{test}.

   Each of W_{active} and $W_{passive}$ is called a "Black Box" because these workstations present a uniform and replicable communications environment. The only component that is varied in this testbed is W_{test}. As a result, all the workstations under test will undergo the same level of test. All modifications to emulate a new communications environment is done in the black boxes ($W_{passive}$ and W_{active}) while W_{test} is not changed at all to reflect the real communication efficiency of the test machine.

4. Network between W_{active}, $W_{passive}$ and W_{test} which for this feasibility study of a communication benchmark is Ethernet.

In the implementation section we shall describe the details on the communication between W_{active}, $W_{passive}$, and W_{test} necessary to obtain a fair and fully automated benchmark test.

The benchmark suite: it is the second important concept in the design of this benchmark. We have categorized the activities we want to benchmark as "Network Activities" and those which compete for resources on W_{test} as "Host Activities".

Network Activities: These are the core activities of the entire benchmark. Measurements made during these activities are used to compute the BTU number for W_{test}. These activities are classified according to: Type of data transfer - bulk and interactive, reliable/unreliable protocol - TCP and UDP , and direction of data transfer - half duplex and full duplex.

Host Activities: Host activities include CPU, I/O and communication intensive applications such as FTP which compete with network activities. The competition is for shared resources such as bus, memory, media and disk. These activities are a major factor in evaluating the performance of W_{test} in a concurrent environment.

In summary, the benchmarking process consists of the following steps:

1. initiate network to be emulated;
2. execute Network, CPU, I/O, and FTP activities separately to obtain baselines;
3. execute Network and CPU activities concurrently;
4. execute Network and I/O activities concurrently;
5. execute Network and FTP activities concurrently;
6. compute BTU as the normalized appropriate average of all Network activities;
7. compute compound penalty on all activities.

We have automated this process so that we can take a workstation with a UNIX operating system, standard TCP/IP implementation, and an Ethernet connection, install our benchmark program on the machine to be tested, hook it up to the testbed, run the benchmark and get the two main results(BTU plus penalty) as well as detailed tables of the performance of individual components. This process is described in more detail in Section 4.3.

4 The Implementation

In this section, we describe our implementation of the benchmark we discussed in the previous section. First, we shall describe the technical details of the testbed followed by a description of the components which make up the benchmark suite. Finally, we provide details on the automation of the benchmark process. This latter part proved to be very important because of the large number of runs which go into the obtaining of the BTUs for a particular machine.

	Type	1K/1K	1K/4K		4K/1K	4K/4K		8K/4K	8K/8K
1	SEND	4.94	4.97		5.33	4.45		4.50	4.38
2	REC	4.68	4.79		4.79	4.64		4.78	4.52
3	SEND+DL	8.12	8.23		6.54	6.23		6.85	6.14

Table 1: Communication time for different combinations of application buffer sizes, SS10, in sec

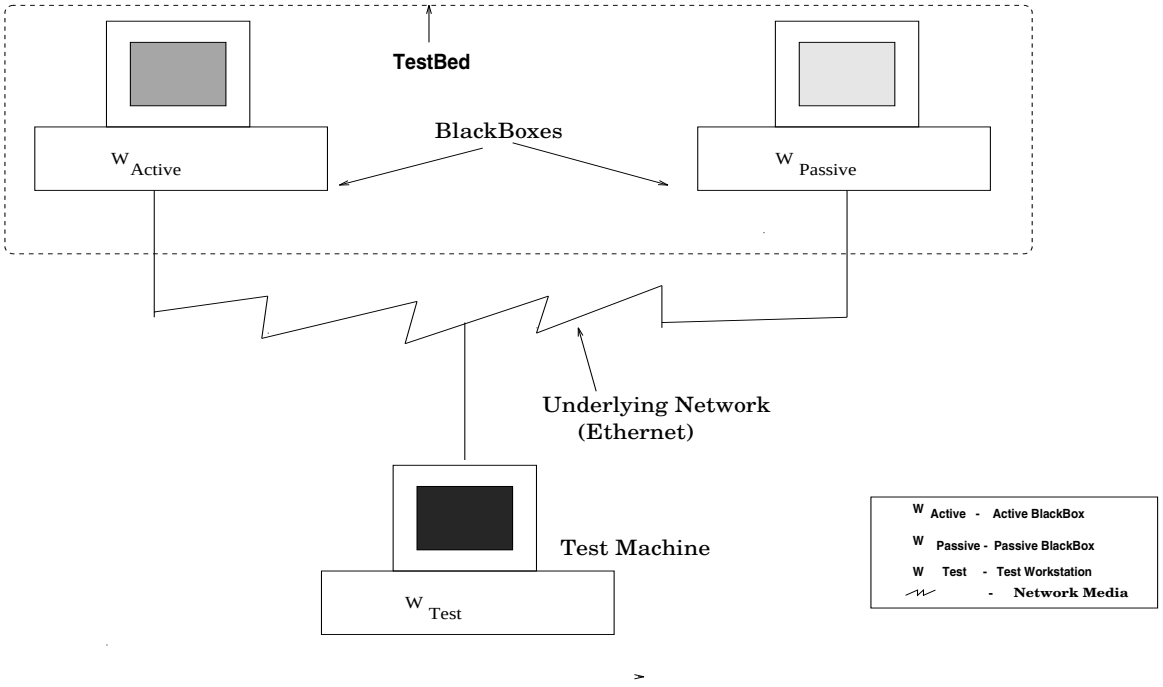

Figure 1: The Benchmark Testbed

4.1 TestBed Description

Our current testbed is shown in Figure 1. It consists of the three machines namely W_{active}, $W_{passive}$ and W_{test}. W_{active} is a Sun Sparc Station 10 with 1.2GB secondary storage capacity and 32MB RAM, $W_{passive}$ is a Sun Sparc Station 2 with 32MB RAM and secondary storage the same as for W_{active}. Both of these workstations run SunOs 4.1.3 UNIX operating system. TCP, UDP, IP and Ethernet are as the communication protocols in W_{active} and $W_{passive}$. All the workstations are connected to an Ethernet in this testbed.

Modifications at the IP layer have been made in W_{active} and $W_{passive}$ to emulate various communications conditions. Packet delay(DL) in IP, and packet drop (DR) are used to simulate a LAN and WAN environment. When packet delay and drop are considered they are applied to only the incoming packets of the IP layer. The ipintr() routine of the kernel is modified to incorporate these changes. We experimented with the proper selection for values of the delay and the drop probability to cause various parameters of TCP(Silly Window Syndrome, delayed acknowledgement, slow start, Nagle's algorithm) to be executed. The range of values considered was from 0.1ms to 1s for DL and from 0.0001 to 0.01 for DR, the probability of a packet being dropped. The values used for the components 5 and 6, described in Section 4.2, are 1ms and 0.01 respectively.

In this paper we include the results of our benchmark tests on workstations from four vendors for W_{test}. The selection of these test workstations and their operating systems was based on their availability in our networking lab and their being representative of currently existing LANs. The configuration of these tested workstations is listed in Table 4.

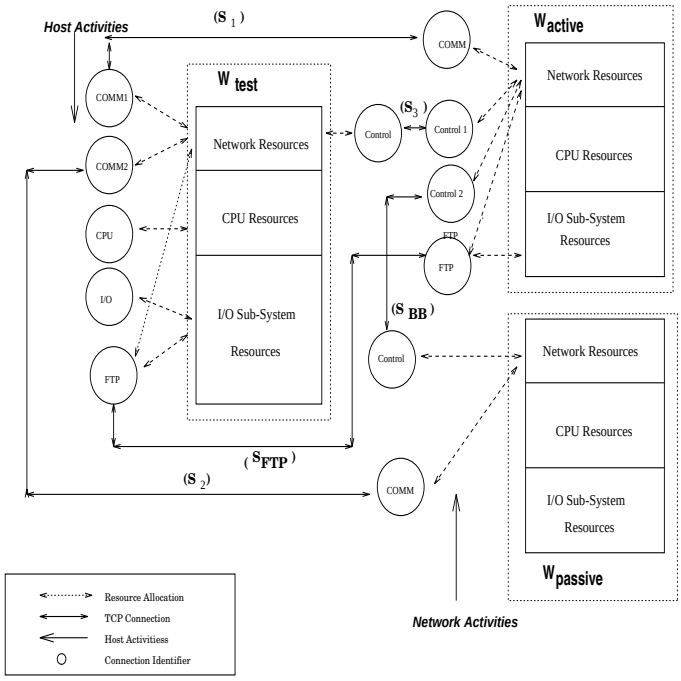

Figure 2: Connectivity in Automated Benchmark

4.2 Benchmark suite

The causes for bandwidth limitation and latency at the user level are twofold: network latencies and host latencies. Network latencies are host independent but dependent on the network load situation and the speed of the physical medium of the network. Host latencies are independent of the network but dependent on the host's CPU speed and the operating system. Various factors influence the overhead incurred by a message moving upstream/downstream in the system. They include: copying of data, buffer management, protocol processing, interrupt handling, system call handling, unpredictable overheads obtained due to different message sizes, internal TCP path taken [46]. All these cause reduction in CPU cycle availability for other applications.

Host activities: as discussed in the approach section the network activities to be measured are run concurrently along with other host activities such as CPU, I/O and other network activities. On W_{test}, a CPU activity is a process which keeps the CPU busy 100 percent of the time. This job has two parts built into it. The first part performs complex arithmetic calculations and the second part allocates and dellocates chunks of memory, copying the results of the complex calculations in the first part to the additionally allocated memory. The first part acts as a basic CPU intensive step and the second part puts a burden on the MMU (memory management unit) and occupies a significant bandwidth on the bus interface. This two steps are repeated over a number of times so that the overall baseline CPU time is comparable with the baseline network activity's time.

The I/O activity constantly reads a large file and copies it into another file which is repeated a number of times, again to make its time comparable to the baseline network activity's time. Disk I/O contends with network I/O for bus, memory and context switches. One should note here that in the host activities I/O commonly refers to both network I/O and disk I/O. We divided these into disk I/O and network activity, thus we refer to disk I/O as I/O and network I/O as network activity.

Network activities: They are choosen such that bytes are transferred between W_{test}, $W_{passive}$ and W_{active} workstations. Some examples of the networking activities are ftp and TTCP which do real file transfer and simulated data transfer over the network respectively. TTCP has been used by many researchers to simulate network traffic for performance studies. FTP can be considered as a compound application which uses both network and disk I/O. For that reason, we have selected FTP as one of the applications competing for resources with the benchmark network activities. At the core of all our components of the benchmark suite is a modified TTCP to act as the sending and receiving agent for the user on W_{test}.

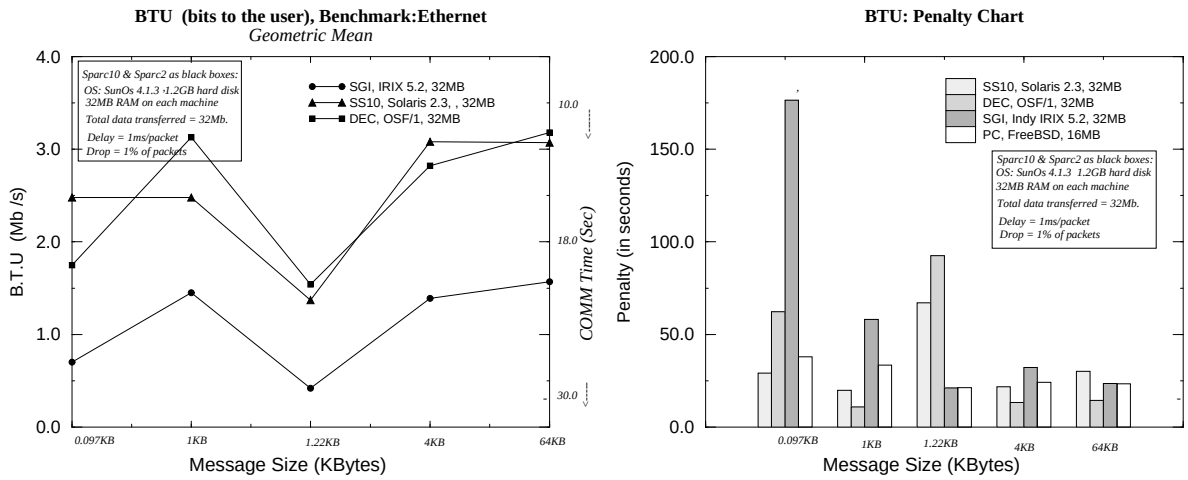

Figure 3: a)BTU Benchmark: Ethernet b)BTU Penalty Chart

We performed a number of experiments with TCP window sizes on W_{active} and $W_{passive}$. Initial results showed that a send window size of 16KBytes on W_{active} and $W_{passive}$ good throughput for the W_{test} workstations. But due to SWS (Silly Window syndrome)[27] effects, there was a marked difference between the test machines of different operating systems and architectures. In order to emulate the heterogeneous nature of networks and the likelihood for a test machine to encounter communication hosts with small window sizes we kept the window sizes on the blackboxes at 4K. The next parameter that we changed was the user submitted buffer size also called message size. In the initial phase we used a combination of different send and receive message sizes (eg.2K, 4K,.. 64K). In Table 1 we list times for four components of our benchmark suite(explained below) for different combinations of send/receive buffer sizes. We confirmed what has been observed before that best throughput occurs at the same send and receive message sizes. Since most applications follow this dictum we used in all our components equal sized buffers. We have selected five classes of application buffer sizes(100Bytes, 1024Bytes, 1250Bytes, 4KBytes, 64Kbytes) each representing a specific class of application such as an interactive database application(100Bytes), ftp or Mosaic or any typical application which wants to use standard 1K buffer size, a typical 16KHZ audio sample data captured for every 1/8th of a second (1250Bytes) or a compressed 160 x 120 sized video frame(or infact any application generating odd message sizes), a typical intelligent application which uses 4K to optimize the buffer management or make buffer sizes equal to the Block Size of the disk, some bulk data unintelligent application which want to transfer huge data sizes(64K). When messages are submitted at smaller lengths more processing overhead and context switches are undergone by the kernel to supply the data from the user buffers to the kernel TCP buffers[32]. The seemingly odd size of 1.22KB (1250Bytes) was added because we found applications with odd message sizes which caused great variations in performance.

Benchmark suite components: From the various experimentations and considerations we selected six components which represent a large percentage of common net activities and which exercise the various factors identified in the above discussion. As indicated the core network activity is based on a modified TTCP program(we shall use the term COMM to refer to the network transfer program and use SEND and RECV for specific components of this program). The modifications were made to implement the benchmark components we selected; these components are as follows:

1. Execute the network activity with W_{test} sending the data to W_{active}($SEND$);
2. Execute the network activity with W_{test} receiving the data from W_{active}($RECV$);
3. Execute the network activity with delay added to the IP of W_{active} and W_{test} sending the data ($SEND$+DL);
4. Execute the network activity with two connections to W_{test} from W_{active} one receiving and one sending ($SEND\_RECV$);
5. Execute the network activity with two connections from W_{test}, one to W_{active} and another to $W_{passive}$ along with delay added to the IP of the blackboxes. The test machine is receiving data from both the

Sparc 10	Message Size	$COMM_{cpu}$	CPU_{comm}	$COMM_{io}$	IO_{comm}	$COMM_{ftp}$	FTP_{comm}
	100 Bytes	6.98	4.98	5.39	4.42	2.55	4.77
	1024 Bytes	6.71	2.43	4.85	2.07	1.34	2.43
	1250 Bytes	21.71	2.43	19.67	1.82	19.34	2.14
	4K Bytes	6.35	2.21	4.93	1.68	4.17	2.48
	64K Bytes	6.69	2.98	8.42	2.87	5.69	3.46
DEC	Message Size	$COMM_{cpu}$	CPU_{comm}	$COMM_{io}$	IO_{comm}	$COMM_{ftp}$	FTP_{comm}
	100 Bytes	2.93	4.41	46.34	1.11	2.62	4.94
	1024 Bytes	0.19	3.62	0.93	0.67	2.28	3.18
	1250 Bytes	0.13	0.88	85.37	0.41	2.80	2.97
	4K Bytes	2.94	0.61	3.17	0.43	2.38	3.71
	64K Bytes	1.76	2.68	4.62	0.86	1.84	2.56
SGI	Message Size	$COMM_{cpu}$	CPU_{comm}	$COMM_{io}$	IO_{comm}	$COMM_{ftp}$	FTP_{comm}
	100 Byte	8.32	11.65	56.54	12.52	86.67	0.72
	1024 Bytes	11.70	4.59	12.53	7.38	4.58	17.31
	1250 Bytes	3.78	3.17	4.93	6.32	1.87	1.03
	4K Bytes	8.05	2.90	9.38	6.74	3.91	1.22
	64K Bytes	6.34	2.46	4.78	3.46	2.88	3.56
PC	Message Size	$COMM_{cpu}$	CPU_{comm}	$COMM_{io}$	IO_{comm}	$COMM_{ftp}$	FTP_{comm}
	100 Byte	5.23	8.23	9.38	7.51	5.03	2.53
	1024 Bytes	3.77	5.09	8.04	7.22	6.99	2.35
	1250 Bytes	3.57	4.57	2.40	2.97	6.48	1.37
	4K Bytes	3.59	3.76	7.10	2.85	5.07	1.76
	64K Bytes	3.51	4.69	6.66	1.98	4.68	1.86

Table 2: Penalties on Activities for Various Machines

	Type	New Testbed (BTU'95)	Previous Testbed (BTU'94)
1	SPARC 10	1.37	2.8
2	SGI Indy R4000	0.42	1.56

Table 3: Comparision of the Results of the BTU'95 and BTU'94 Testbed.

blackboxes. ($2 \times RECV$+DL);

6. Execute the network activity with two connections from W_{test}, one to W_{active} and another to $W_{passive}$ along with packet dropping added to the IP of the blackboxes. The test machine is sending data to the blackboxes($2 \times SEND$+DR);

4.3 Automation of the Benchmark Process

We have automated the entire benchmark processing software including data collection and post-benchmark processing. The user has to start the server on the two blackboxes and a client program on the test machine. After 1 to 2 hours of processing the W_{active} provides the necessary results by collecting the data from the test machine and post-processing the data. In order to do this five connections are made between the three machines namely W_{active}, $W_{passive}$ and the test machine. The automation connections which are shown in Fig. 2, are:

1. Connection between W_{active} and W_{test} (S_1)
2. Connection between $W_{passive}$ and W_{test} (S_2)
3. A connection between W_{active} and W_{test} to get control information from W_{test} (S_3)
4. A connection between W_{active} and $W_{passive}$ to get control information (S_{BB})

Sl.No.	Machine Architecture	Operating System	Memory Spec	Disk Spec.	CPU Spec.	Network Spec.	File System Spec.	Price ($)
1	SPARC 10	Solaris 2.3 (SunOs 5.3)	32MB RAM	1.2GB DSU. SCSI	SuperSparc10, SBus, 60Mhz 20/16 cache	Lance Ethernet (10mbps card)	UFS	18,0000
2	DEC ALPHA 4/233 Model:200	OSF/1 V 3.0	32MB RAM	1.2 GB SCSI	21064A-6 233MHz 512KB cache, PCI Bus	DECchip 21040-AA: Revision: 2.3 DEC TULIP Ethernet (10mbps card)	AdvFS, UFS	14,5000
3	SGI Indy	IRIX 5.2	32MB RAM	1.2 GB SCSI	MIPS R4000 100MHz 8/8 Data/Inst. Cache	Integral Ethernet Ver. 1.0 (10mbps card)	UFS	8,495
4	AMD DX2	FreeBSD Ver 2.0	16MB RAM	1.037GB Enhanced IDE	80486DX2 80MHZ	SMC Elite 16 (10mbps card)	UFS	7,500

Table 4: Test Machine Configurations

5. A connection between W_{active} and W_{test} to obtain FTP control (S_{FTP})

Each process interacts with the other processes on the other machines. Processes are concurrently started on the test machine whenever load (CPU,I/O,FTP activities) is introduced on the net activity. The network and host activities have been choosen so that the baseline execution times of these activities are comparable to nearest seconds so that the impact of one on the other can be clearly seen. It is clear that when a host load activity like a CPU job is run concurrently with network activity both of them finish later than they did in the baseline case due to CPU cycle sharing, competition for interrupt and context switches, MMU and bus bandwidth. As shown in Fig. 2 a minimum of five connections is made between the machines in the testbed and the latencies can also be viewed. Connection S_1 is for transferring data between W_{active} and W_{test}. Connection S_3 is for transferring the control information which makes the whole automation process to resume smoothly. Similarly, a connection is made between W_{active} and $W_{passive}$ for control information. From time to time data is collected from the test machine to W_{active}. The state diagram in Fig. 2 explains the connection mechanism between all the machines in various situations of connection activities. We have also built in a restart capability which became necessary when some machines stopped a component due to some abnormal situations. The W_{active} benchmark process is started with the test machine name as an argument, which then makes connection S_3 and S_{BB} with $W_{passive}$. These two connections synchronize the benchmark process till the end of the test. Whenever FTP load is applied on W_{test}, S_{FTP} is used and remains idle in the remaining span of time. At each step of the test (SEND,RECV, ...) the results of the activities are collected from W_{test} using S_3, S_{FTP} and from $W_{passive}$ using S_{BB}. A simple token mechanism is used for the proper synchronization of the testbed which occupies a minor amount of network bandwidth. This amount is negligible since the synchronization packets are sent only after each test is performed.

Sl.No.	Machine Architecture	SPEC int92	BTU'95	BTU Rank
1	SPARC 10	65.2[3]	2.40	I
2	DEC ALPHA 4/233 Model:200	157.7[4]	2.375	II
3	SGI Indy	36.4	0.987	IV
4	AMD DX2	32.2	2.077	III

Table 5: Final BTU Ranking

5 Experiments and Results

BTU and statistics calculation: For this paper we include the results of testing workstations from four different vendors. The specific configuration of each machine is given in Table 4 [4] [5] , also the specint '92 [45] are specified in Table 5 along with our BTU'95 results to aid in rating the machines on overall performance. The bits to the user chart shown in Fig. 3a represents the effective bits delivered by a system after all the activities are measured. The net activity durations for three loads on the host(CPU, I/O, FTP) are taken after averaging the net durations for various combinations of net activities as specified in section 4.3 per each message size. Since the data from which these averages are calculated show a large variation(see Tables A5 through A16 in the Appendix) we decided against the arithmetic mean as the basis for BTUs. Instead of the arithmetic mean, the geometric mean of the three resultant net duration times is taken as the final value to calculate BTUs(the BTU values from the current testbed are called BTU'95).

The x-axis of Fig. 3a represents various message sizes; on the left size of the diagram we measure BTUs in Mb/s(this should be compared to the 10 Mb/s capacity of Ethernet); this number is computed as the geometric mean of the amount of data communicated divided by the time taken for all components of the benchmark suite to complete. On the right side we provide the absolute time an average activity takes measured in seconds. This measure is useful when reading the penalty chart, Fig. 3b, where we show the combined penalties(sum of penalties on three average activities) measured in seconds. Results obtained using a test bed different from the latest bed(BTU'95) consisting of a Sparc 10 as W_{active} and Sparc ELC has $W_{passive}$ blackbox showed similar results of ranking as the one used to create BTU'95 (Table 5). Due to the different message sizes used then, we show only (in Table 3) BTU results for common message size. We discuss some of the issues related

[4] SPARCstation 10 Model 51 (128MB) rating may have a minor variation with the Sparc10 we tested.
[5] AlphaStation 200 4/233(112MB RAM) rating may slightly differ the 32MB Alpha Station tested.

to an evolving testbed in the next section.

Tables A5 through A17 in the appendix provide the raw results of running six benchmark components concurrently with some other activity as a function of the message size. We provide three means: arithmetic, geometric, harmonic. Each table contains two parts: the top lists the times for a benchmark component when another activity is run concurrently, the bottom part gives the times for these competing activities when a benchmark component runs concurrently. We have structured the tables by vendors, for example, Tables A5 through A7 give all the data for a DEC workstation. We have italicized entries which are clearly out of the normal range. To obtain a single BTU number which can be identified with a particularly configured machine we have combined(geometric mean) the various values for different message sizes in Fig. 3a. For comparison, we have included BTU calculations using other means(arithmetic - AM, geometric - GM, harmonic - HM). Details of this calculation are given in the Appendix : Table A17. We have also added the BTU number as the last column of Table 5 so that easy comparisons can be made with the results of Spec int92.

Penalty Calculations: We calculate the penalty on network activities by measuring a COMMunication activity when run by itself(Baseline) and when run with a competing activity.

$$Penalty_{COMM} = COMM_X \text{ - } COMM_{Baseline},$$

where X represents a component of the benchmark suite. The baseline figures for the various components are given in the Appendix, Tables A1 through A4 for the tested machines; the results are given in Table 2.

Similarly, we calculate the penalty on host activities by running the host activities separately by themselves(Baseline) and concurrently with the network activities.

$$Penalty_{Host} = Host_X \text{ - } Host_{Baseline}$$

where Host stands for CPU, I/O, and FTP activities and X again represents execution of the benchmark suite components. The baseline figures are given in the Appendix, Tables A1 through A4, and the individual penalties in Table 2. For example, CPU_{comm} means the penalty on the CPU job when the COMMunication benchmark is run concurrently. To achieve a single number for the penalty graph in Fig. 3b, we add all the penalties together although machines can be compared also by comparing individual penalties.

$$TotalPenalty = \sum Penalty_{COMM} + \sum Penalty_{Host}$$

Key results: the BTUs for each of the workstation and the corresponding penalties are given in Figures 3a, 3b ,and Table 2. Figure 3a shows that the machines behave differently for different application categories. Though the Sparc10 is robust and has good overall performance, our results showed that a PC outperforms a Sparc10 in some aspects because we used a 4.4 BSD version for the PC which is more enhanced compared to a System V stream implementation of networking in the Sparc10. The DEC machine shows a good processor performance as all the instructions are 64bit. But due to a lack of optimization of the I/O subsystem overall performance of the networking is poor. The SGI has somewhat less processing power compared to all the other three and since TCP the implementation is quite different it has a low performance compared to others. One abnormality shown in Table A6 is the COMM time for the 1250B buffer case ($COMM_s$, $COMM_{sr}$). Execpt for PC, all the other machines have significant changes in performance for 1250 bytes application buffer length. This can be attributed to the implementation techniques used in the operating system for copying the user to kernel buffers and the buffer allocation schemes used (eg. mbufs [28], streams [44]).

From these figures it can be seen that although some workstations are in the same prize categories (Sparc Station 10 and DEC) they deliver considerable different number of bits to the user on the average[23]. Comparing the send case with the receive case of test machine, the send case takes more time. This can be ascribed to the different window sizes of the test machine and the blackbox and the implementation differences (eg. mbufs and streams) along with the well known transport level effects like SWS, delayed acknowledgements, Nagle's algorithm and slow start. The Sparc Station 10 (Solaris 2.3) has a window size of 8K and the black box (SunOs 4.1.3) has 4K. Due to this effect and as the receiving side controls the sending side in the TCP flow control, the test machine behaves differently for send and receive cases.

6 Conclusion and Future Plans

As the results from testing four separate vendor's workstations indicate this benchmark produces clearly differentiable data sets which are not necessarily related to CPU speeds. Neither the speed of the CPU(measured in either Mhz or SPECmarks) nor the price was a predictor of communication performance as measured in BTUs. Thus, we believe we have made the case that this proposal satisfies the points we made in the introduction. In particular, this benchmark is immediately useful for a very large installed base of workstations in current Ethernet based LANs.

We are now in the process of submitting a number of other machines with varying operating systems to the benchmark test and will shortly provide a comprehensive study. At the same time we are adapting the design of the benchmark such that it will enable us to test a machine's performance over several networks of different speed(ATM, FDDI, and T based 100 Ethernet). By the time these highspeed networks become prevalent we feel confident that we have such a benchmark suite.

Several issues relating to the testbed remain unresolved to a certain degree. One issue yet to be resolved is the inclusion of UDP applications. Due to large losses in UDP datagrams when the network and the CPU are loaded, no measurement would be complete without taking the amount of the loss into account. On another issue: we performed a number of experiments to see the dependence of the results on the actual configuration of the testbed. We used various workstations as testbed machines and compared the results. In general, the actual numbers were indeed different but they did not affect the ranking of the tested workstations. Neither did the problem cases for the tested workstations change: the same components for the same message size produced results outside the normal range. This supports our contention that the type of the testbed machines is not important as long as we keep them representative of current installed bases. It leads though to the problem we are facing: How should such a benchmark suite process be run? We see two clearly different models: one is similar to the testing procedure used by [47] to evaluate network routers and switched internetworking products and the second would emulate SPECmarks. In the first model, an independent organization such as NIST, MITRE or OSF would maintain an up-to-date testbed and periodically publish rankings of machines supplied by vendors for testing. In the second model, we would explicitly specify in great detail the current testbed and provide the software throughout the Web for anybody who can duplicate the testbed. These individuals or organizations could then run the benchmark test themselves. This latter process implies that the testbed would be certified by some central organization.

References

[1] Rafael H. Saavedra-Barrera, "*CPU Performance Evaluation and Execution time Prediction Using Narrow Spectrum Benchmarking,*" Technical Report, Computer Science Division, University of Calfornia, Berkeley, CA, February 1992.

[2] Hockney, R. and Berry M., "*Public International Benchmarks for Parallel Computers: PARKBENCH Committee Report-1,*" Scientific Programming, vol3 (2), 101-146, 1994.

[3] Jack J. Dongarra, "*Performance of Various Computers Using Standard Linear Equations Software,*" Simulation (ISSN 0037-5497) v49, pp. 51 - 62, August 1994.

[4] "*High Performance TCP/IP for OSF/1 Alpha AXP Workstations,*" White Paper , Digital Equipment Corporation, March 1993.

[5] Allyn Romanow, "*Preliminary Report of Performance Results for TCP over ATM with Congestion,*" Technical Report, SunMicrosystems Inc. , July 1993.

[6] "*Nettest a network Benchmarking tool ,*" Technical Report, Cray Research Inc., 1992.

[7] S. Bradner, "*RFC 1242, Benchmarking Terminology for Network Interconnection Devices,*" Harvard University , July 1991.

[8] Roger W.Hockney, "*Computational Similarity,*" Technical Report, Department of Computer Science, Warwick University, August 1994,

[9] Ishfaq Ahmad, Min-You Wu, Jaehyung Yang, and Arif Ghafoor, "*An Experimental Assessment of Experess Parallel Programming Environment.*" Technical Report, Department of Computer Science, State University of New York, Buffalo. NY, May 1993.

[10] Luis Felipe Cabrera, "*The Impact of Buffer Management on Networking Software Performance in Berkeley UNIX 4.2BSD: A Case Study,*" CSRG, Department of Computer Science, Universityof California, Berkeley, April 1990.

[11] "*RFC 813, Window And Acknowledge Strategy in TCP,*" MIT Laboratory for Computer Science, CSCG, July, 1982.

[12] Rafael H. Saavendra, and Alan Jay Smith, "*Analysis of Benchmark Characteristics and Benchmark Preformance Prediction.*" Technical Report, Computer Science Division, University of Calfornia, Berkeley, CA, September 1992.

[13] David D. Clark, "*Modularity and Efficiency in Protocol Implementation,*" RFC 817, NIC, July 1982.

[14] "*RFC793, Transmission Control Protocol, DARPA,*" September 1981.

[15] Arup Mukherjee, and Daniel P. Siewiorek,"*Measuring Software Dependability by Robustness Benchmarking,*" CMU-CS-94-148, May 1994.

[16] Thomas M.Conte, and Wen-mei W. Hwu,"*Benchmark Characterization,*" IEEE Computer, p48-55, January 1991.

[17] David Finkel Robert , and Aju Bradford Somesh Rao, "*Developing Benchmarks to Measure the Performance of the Mach Operating System,*" Proceedings of the USENIX MACH Workshop, P83-100, 1990.

[18] Douglas E.Comer, and John C. Lin,"*TCP Buffering And Performance Over An ATM Network,*" Purdue Technical Report CSD-TR 94-026, Department of Computer Science, Purdue University, March 16, 1994.

[19] David D. Clark, Van Jacobson, John Romkey, and Howard Salwen "*An Analysis of TCP Processing Overhead,*" IEEE Communications Magazine (ISSN 0163-6804) v27, pp. 23 - 9, June 1994.

[20] Technical staff,"*A Summary of Network Performance Benchmarking of Sun 4/490 File Servers,*" Computer Science Department , Brown University, December 1990 .

[21] Kurt Maly, S.Khanna ,R.Mukkamala, and C.M.Overstreet, "*Parallel TCP/IP for Multiprocessor Workstations,*" C1.1-C1.16, Liege, HPN'92 , December 1992.

[22] Kurt Maly, D.Sudheer, R.Mukkamala, C.M.Overstreet, and D.E.Keyes, "*An Application-oriented Analysis of TCP/IP in High Speed LANs,*" 561-566, Australian Telecommunication Networks and Applications conference(ATNAC), December 1994.

[23] John Ousterhout, "*Why Aren't Operating Systems Getting Faster As Fast As Hardware?,*" WRL, Technical Note TN-11, Western Research Laboratory, Digital Equipment Corporation, Palo Alto, CA, October, 1989.

[24] Jonathan Kay, and Joseph Pasquale,"*A Performance Analysis of TCP/IP and UDP/IP Networking Software for the DEC station 5000,*" Technical Report , Computer Science Laboratory, Department of Computer Science and engineering, University of California, San Diego, March 1993.

[25] David R.Boggs, Jeffrey C.Mogul, and Christopher A.Kent, "*Measured Capacity of an Ethernet: Myths and Reality,*" Proceedings of SIGCOMM '88, pp. 222 - 234, August 1988.

[26] Technical Staff, "*Alpha AXP Personal Computer Performance Primer,*" Technical Report Digital Equipment Corporation , Palo Alto , CA , January, 1994.

[27] W.Richard Stevens , "*TCP/IP Illustrated Volume 1 ,*" ISBN 0-201-63354-X, pp. 325 - 330; pp. 263 - 285, Addison-Wesley Professional Computer Series, November, 1994.

[28] W.Richard Stevens, and Gary R.Wright, "*TCP/IP Illustrated Volume 2 ,*" ISBN 0-201-63346-9, pp. 31 - 60, Addison-Wesley Professional Computer Series, January, 1995.

[29] W.Richard Stevens , "*Advanced Programming in the UNIX Environment ,*" ISBN 0-201-56317-7, Addison-Wesley Professional Computer Series, June, 1993.

[30] W.Richard Stevens , "*UNIX Network Programming ,*" ISBN-0-87692-749-5,Prentice-Hall International,INC, June, 1993.

[31] Lawrence H.Miller, and Alexander E.Quilici ,"*The Joy of C ,*" ISBN-0-471-513351333-4, John Wiley and Sons, Inc., June, 1993.

[32] Jefffrey C.Mogul ,"*Observing TCP Dynamics in Real Networks ,*" WRL Research Report 92/2, Western Research Laboratory, Digital Equipment Corporation , Palo Alto , CA, April, 1992.

[33] Rick Jones, "*Network Performance Home Page,*" http://www.cup.hp.com/netperf/NetperfPage.html.

[34] Jonathan Kay, and Joseph Pasquale, "*Measurement,Analysis and Improvement of UDP/IP Throughput for the DEC station 5000 ,*" Winter USENIX Conference, 1993. Computer Science Laboratory, Department of Computer Science and engineering, University of California, San Diego.

[35] Lawrence S.Brakmo,Sean W.O'Malley, and Larry L.Peterson,"*TCP Vegas: New Techniques for Congestion Detection and Avoidance,*" Technical Report , Department of Computer Science , University of Arizona, Tucson AZ December, 1994.

[36] Allan Small,"*TUTBOCHARGING your UNIX environment with DEC OSF/1 ,*" Technical Exchange Program 9, Alpha Interoperability, Customer Support Center, Sydney, June 1992.

[37] Linda Simon ,"*OSF/1: Digital Equipment Corporation's Strategy for UNIX ,*" Technical Report EC-N0368-43, DEC, March 1991.

[38] Technical Staff ,"*High Performance TCP/IP for OSF/1 Alpha AXP Workstations,*" White Paper, , DEC, Networks Engineering TCP/IP Programming Office, MA , March 1993.

[39] "*Benchmark Results ,*" SPEC Newsletter, Vol. 2, No. 2, Spring 1990.

[40] D.L. Black, R.F. Rashid, D.B. Golub, C.R. Hill, and R.V. Baron, "*Translation Look-aside Buffer Consistency: A Software Approach ,*" Digest of Papers, COMPCON Spring '89, Thirty-Fourth IEEE Computer Society International Conference: Intellectual Leverage, (1989), pp. 184 - 190.

[41] H.J. Curnow and B.A. Wichmann, "*A Synthetic Benchmark ,*" The Computer Journal, 19 (1976), pp. 43 - 49.

[42] J. Dongarra, J.L. Martin and J. Worlton, "*Computer Benchmarking: Paths and Pitfalls ,*" IEEE Spectrum, pp. 38 - 43, July 1987.

[43] D. Golub, R.Dean, A. Forin and R. Rashid, "*UNIX as an application Program ,*" Proceedings of the USENIX Summer Conference, June, 1990, pp. 87 - 95.

[44] "*STREAMS Programmer's Guide ,*" SunOS 5.4, Sun Microsystems Inc.,Proceedings of the USENIX Summer Conference, June, 1990, pp. 87 - 95.

[45] "*PDS: The Performance Database Server,*" http://performance.netlib.org/performance/html/spec.html.

[46] Peter A. Steeenkiste, Brain D. Zill, et al., "*A Host Interface Architecture for High-Speed Networks ,*" Proceedings of the 4th IFIP conference on High Performance Networking (HPN'92), December, 1992, pp. A3-1 to A3-16.

[47] Scott Bradner, "*Strategic Networks Consulting Services,*" http://www.snci.com/netrpt.htm.

Appendix: Details for Ethernet Benchmark.

A1. Baseline Cases:

	Type	100 Bytes	1024 Bytes	1250 Bytes	4KBytes	64KBytes		Type	Real Time
1	SEND	6.21	5.13	4.51	4.45	4.74	1	CPU	22.15
2	RECV	6.17	4.90	**42.00**	4.69	4.62	2	IO	18.07
3	SEND+DL	9.19	8.62	7.76	6.42	7.03	3	FTP	13.32
4	SEND_RECV	9.45	7.83	**23.86**	7.04	7.69			
5	2 X RECV+DL	9.84	8.59	41.99	8.08	7.12			
6	2 X SEND+DR	15.53	12.41	12.98	10.25	11.20			

Table A1: Baseline Network and Host Activities of Sparc Station 10 (in sec)

	Type	100 Bytes	1024 Bytes	1250 Bytes	4KBytes	64KBytes		Type	Real Time
1	SEND	5.45	4.44	4.38	4.18	4.21	1	CPU	3.51
2	RECV	5.67	4.78	4.64	4.67	4.57	2	IO	6.11
3	SEND+DL	8.52	7.12	7.10	6.68	6.62	3	FTP	11.20
4	SEND_RECV	8.72	7.46	7.23	7.34	7.47			
5	2 X RECV+DL	10.01	8.96	8.46	8.31	7.25			
6	2 X SEND+DR	43.25	43.31	44.45	41.81	44.48			

Table A2: Baseline Network and Host Activities of DEC (in sec)

	Type	100 Bytes	1024 Bytes	1250 Bytes	4KBytes	64KBytes		Type	Real Time
1	SEND	11.36	5.58	5.37	5.15	5.34	1	CPU	14.69
2	RECV	7.67	8.42	**335.18**	**20.40**	6.31	2	IO	17.53
3	SEND+DL	11.08	8.56	8.41	7.81	7.71	3	FTP	10.95
4	SEND_RECV	16.59	9.11	**170.06**	13.26	8.39			
5	2 X RECV+DL	13.67	11.66	**334.35**	20.13	7.77			
6	2 X SEND+DR	47.17	43.78	43.45	42.29	44.33			

Table A3: Baseline Network and Host Activities of SGI (in sec)

	Type	100 Bytes	1024 Bytes	1250 Bytes	4KBytes	64KBytes		Type	Real Time
1	SEND	7.17	5.39	5.44	4.89	4.91	1	CPU	12.62
2	RECV	5.97	5.54	5.94	5.41	5.11	2	IO	23.75
3	SEND+DL	9.63	8.40	8.04	7.69	7.60	3	FTP	7.59
4	SEND_RECV	11.16	8.33	8.26	8.43	8.54			
5	2 X RECV+DL	10.28	8.75	10.14	8.62	8.57			
6	2 X SEND+DR	45.85	42.33	44.25	43.37	43.35			

Table A4: Baseline Network and Host Activities of PC (in sec)

A2. Results for DEC Workstation :

	Type	100 Bytes	1024 Bytes	1250 Bytes	4KBytes	64KBytes
1	SEND	21.96	4.74	4.62	4.52	4.82
2	RECV	6.36	5.26	4.82	**20.27**	5.78
3	SEND+DL	8.48	7.27	7.07	6.72	6.69
4	SEND_RECV	8.76	7.67	7.15	7.50	7.52
5	2 X RECV+DL	8.71	8.69	8.65	8.32	7.34
6	2 X SEND+DR	43.61	42.22	44.63	43.32	44.52
7	Arithmetic Mean	43.61	42.22	44.63	43.32	44.52
8	Geometric Mean	16.31	12.64	12.82	15.11	12.78
9	Harmonic Mean	12.57	8.94	8.70	10.89	8.78
1	SEND	21.96	4.73	4.62	4.5	4.82
2	RECV	5.42	20.33	4.52	3.66	4.31
3	SEND+DL	4.57	4.21	4.15	3.97	3.99
4	SEND_RECV	5.56	4.67	4.52	4.52	4.48
5	2 X RECV+DL	5.63	5.12	4.77	4.36	4.39
6	2 X SEND+DR	4.39	3.74	3.76	3.69	3.70
7	Arithmetic Mean	7.92	7.13	4.39	4.12	4.28
8	Geometric Mean	6.49	5.75	4.38	4.10	4.27
9	Harmonic Mean	5.80	5.11	4.36	4.08	4.25

Table A5: Impact of CPU on COMM of DEC and Impact of COMM on CPU of DEC (in sec)

	Type	100 Bytes	1024 Bytes	1250 Bytes	4KBytes	64KBytes
1	SEND	7.74	6.68	6.69	6.90	7.27
2	RECV	**144.09**	7.52	**322.10**	**20.10**	5.81
3	SEND+DL	8.85	7.41	7.49	6.97	6.97
4	SEND_RECV	**145.73**	7.77	**153.06**	7.65	7.60
5	2 X RECV+DL	9.86	8.53	54.67	8.53	7.43
6	2 X SEND+DR	42.13	43.11	44.24	41.87	43.32
7	Arithmetic Mean	59.73	13.50	98.04	15.34	13.07
8	Geometric Mean	29.02	10.10	42.60	11.76	9.47
9	Harmonic Mean	15.74	8.74	17.99	9.84	8.08
1	SEND	7.73	6.64	6.69	6.77	7.12
2	RECV	6.67	6.76	5.79	5.96	6.48
3	SEND+DL	7.28	6.75	6.69	6.56	6.57
4	SEND_RECV	7.37	7.24	6.36	6.75	6.64
5	2 X RECV+DL	7.69	7.19	6.74	6.74	6.77
6	2 X SEND+DR	6.58	6.11	6.49	6.32	6.24
7	Arithmetic Mean	7.22	6.78	6.46	6.52	6.64
8	Geometric Mean	7.21	6.77	6.45	6.51	6.63
9	Harmonic Mean	7.19	6.76	6.44	6.50	6.63

Table A6: Impact of I/O on COMM of DEC and Impact of COMM on I/O of DEC (in sec)

	Type	100 Bytes	1024 Bytes	1250 Bytes	4KBytes	64KBytes
1	SEND	8.05	6.57	6.51	6.36	6.33
2	RECV	9.47	8.51	4.67	7.01	7.20
3	SEND+DL	10.72	9.87	9.59	9.79	9.71
4	SEND_RECV	12.18	8.07	10.99	10.59	10.95
5	2 X RECV+DL	13.30	12.77	12.80	11.71	11.68
6	2 X SEND+DR	43.62	43.93	48.47	41.25	50.34
7	Arithmetic Mean	16.22	14.95	15.51	14.45	16.03
8	Geometric Mean	13.39	11.65	11.21	11.43	11.91
9	Harmonic Mean	11.92	10.07	9.07	9.90	10.06
1	SEND	14.30	13.20	13.32	13.42	14.20
2	RECV	15.10	14.30	13.21	13.45	15.67
3	SEND+DL	20.22	18.77	18.62	18.43	17.65
4	SEND_RECV	16.22	12.33	12.32	14.62	15.32
5	2 X RECV+DL	17.21	16.23	15.23	16.77	16.32
6	2 X SEND+DR	13.77	11.44	12.31	12.79	12.77
7	Arithmetic Mean	16.14	14.38	14.17	14.91	15.32
8	Geometric Mean	16.00	14.17	14.01	14.78	15.24
9	Harmonic Mean	15.87	13.98	13.87	14.66	15.16

Table A7: Impact of FTP on COMM of DEC and Impact of COMM on FTP of DEC (in sec)

A3. Results for SGI Workstation :

	Type	100 Bytes	1024 Bytes	1250 Bytes	4KBytes	64KBytes
1	SEND	26.05	17.84	17.61	15.64	18.02
2	RECV	13.71	20.46	**335.07**	25.88	19.48
3	SEND+DL	25.62	18.06	17.62	14.83	17.91
4	SEND_RECV	22.74	22.40	170.17	21.68	18.58
5	2 X RECV+DL	20.53	34.30	**333.82**	37.03	28.59
6	2 X SEND+DR	48.84	44.26	44.54	41.90	43.38
7	Arithmetic Mean	26.25	26.22	**153.13**	26.16	24.33
8	Geometric Mean	24.35	24.65	80.05	24.22	22.92
9	Harmonic Mean	22.77	23.37	40.58	22.47	21.86
1	SEND	26.02	17.79	17.58	15.59	17.97
2	RECV	21.66	14.85	13.37	12.75	16.66
3	SEND+DL	25.59	18.02	17.58	14.79	17.85
4	SEND_RECV	24.39	18.62	17.13	18.34	18.90
5	2 X RECV+DL	23.24	19.59	15.13	17.89	18.90
6	2 X SEND+Dr	28.90	18.57	18.16	17.42	17.25
7	Arithmetic Mean	24.97	17.91	16.49	16.13	17.92
8	Geometric Mean	24.86	17.84	16.40	16.00	17.90
9	Harmonic Mean	24.76	17.77	16.30	15.87	17.88

Table A8: Impact of CPU on COMM of SGI and Impact of COMM on CPU of SGI (in sec)

	Type	100 Bytes	1024 Bytes	1250 Bytes	4KBytes	64KBytes
1	SEND	27.26	21.37	21.15	20.67	20.52
2	RECV	**263.63**	23.26	**335.19**	27.20	20.92
3	SEND+DL	27.59	21.15	20.96	20.01	19.98
4	SEND_RECV	24.45	22.24	169.93	19.39	15.29
5	2 X RECV+DL	54.15	30.50	**334.36**	**34.78**	27.05
6	2 X SEND+DR	49.69	43.40	44.67	43.28	44.60
7	Arithmetic Mean	74.46	26.99	154.38	27.56	24.73
8	Geometric Mean	48.52	26.01	85.0	26.27	23.26
9	Harmonic Mean	38.41	25.22	46.43	25.15	22.11
1	SEND	27.22	21.32	21.12	20.62	20.47
2	RECV	18.67	18.84	17.63	18.93	19.42
3	SEND+DL	27.55	21.1	20.92	19.98	19.97
4	SEND_RECV	26.22	21.14	19.61	20.42	20.26
5	2 X RECV+DL	24.00	20.88	17.81	20.10	20.05
6	2 X SEND+DR	31.38	20.93	20.74	20.29	19.73
7	Arithmetic Mean	25.84	20.70	19.64	20.06	19.98
8	Geometric Mean	25.52	20.68	19.58	20.05	19.98
9	Harmonic Mean	25.18	20.66	19.53	20.04	19.98

Table A9: Impact of I/O on COMM of SGI and Impact of COMM on I/O of SGI (in sec)

	Type	100 Bytes	1024 Bytes	1250 Bytes	4KBytes	64KBytes
1	SEND	18.52	7.78	8.14	8.21	7.75
2	RECV	**276.16**	13.68	**335.12**	21.10	13.05
3	SEND+DL	15.13	8.59	8.32	11.54	11.11
4	SEND_RECV	**123.78**	16.15	**169.97**	16.57	13.63
5	2 X RECV+DL	**136.16**	17.73	**334.40**	24.89	15.47
6	2 X SEND+DR	57.78	50.63	51.84	50.20	50.25
7	Arithmetic Mean	104.59	19.09	151.3	22.08	18.54
8	Geometric Mean	64.99	15.38	63.71	18.60	15.11
9	Harmonic Mean	38.35	13.15	21.88	15.94	13.23
1	SEND	13.84	12.84	12.88	13.00	13.09
2	RECV	11.24	11.08	12.68	11.92	13.63
3	SEND+DL	11.62	11.81	16.77	16.43	16.90
4	SEND_RECV	13.87	**113.87**	13.58	13.13	14..38
5	2 X RECV+DL	13.83	13.50	14.56	14.75	15.42
6	2 X SEND+DR	15.28	11.45	11.18	11.48	11.65
7	Arithmetic Mean	11.06	29.79	13.48	13.37	13.98
8	Geometric Mean	7.26	18.53	13.38	13.26	13.88
9	Harmonic Mean	11.99	15.02	13.30	13.16	13.78

Table A10: Impact of FTP on COMM of SGI and Impact of COMM on FTP of SGI (in sec)

A4. Results for PC :

	Type	100 Bytes	1024 Bytes	1250 Bytes	4KBytes	64KBytes
1	SEND	21.07	17.42	17.45	17.16	17.61
2	RECV	6.36	5.56	5.97	5.33	5.26
3	SEND+DL	22.85	17.45	17.07	16.92	16.73
4	SEND_RECV	14.16	8.50	8.56	8.37	8.87
5	2 X RECV+DL	10.80	8.84	10.22	8.65	8.51
6	2 X SEND+DR	46.22	43.56	43.18	42.86	43.20
7	Arithmetic Mean	13.40	10.56	13.53	12.04	16.68
8	Geometric Mean	12.07	9.52	12.12	11.17	13.09
9	Harmonic Mean	10.70	8.65	10.75	10.31	14.56
1	SEND	21.03	17.40	17.43	17.14	17.53
2	RECV	18.85	17.77	18.18	17.54	17.92
3	SEND+DL	22.82	17.43	17.05	16.87	16.67
4	SEND_RECV	20.04	17.82	17.94	17.59	17.71
5	2 X RECV+DL	18.98	18.03	18.07	17.44	17.71
6	2 X SEND+DR	20.85	17.71	17.19	16.38	16.87
7	Arithmetic Mean	18.32	17.84	18.39	17.9	16.51
8	Geometric Mean	18.28	17.82	18.29	17.97	13.03
9	Harmonic Mean	18.23	17.81	18.20	17.94	14.06

Table A11: Impact of CPU on COMM of PC and Impact of COMM on CPU of PC (in sec)

	Type	100 Bytes	1024 Bytes	1250 Bytes	4KBytes	64KBytes
1	SEND	30.41	30.02	26.66	26.28	26.04
2	RECV	7.27	6.30	6.60	6.06	5.72
3	SEND+DL	31.53	26.62	26.34	25.95	26.07
4	SEND_RECV	15.72	8.92	9.00	9.15	9.40
5	2 X RECV+DL	12.83	10.39	10.70	10.18	9.30
6	2 X SEND+DR	48.60	44.73	41.94	43.09	43.27
7	Arithmetic Mean	18.43	14.73	18.40	15.80	19.96
8	Geometric Mean	15.61	12.03	15.43	13.81	15.64
9	Harmonic Mean	12.92	10.14	12.79	12.11	16.72
1	SEND	30.33	30.00	26.64	26.24	25.97
2	RECV	28.05	26.58	26.91	26.96	26.35
3	SEND+DL	31.50	26.60	26.33	25.93	26.02
4	SEND_RECV	29.22	26.92	26.78	26.25	26.43
5	2 X RECV+DL	27.91	26.61	26.74	26.60	26.77
6	2 X SEND+DR	31.26	30.97	26.72	26.60	26.63
7	Arithmetic Mean	27.40	27.29	27.29	27.00	26.495
8	Geometric Mean	27.37	27.25	27.23	26.98	26.36
9	Harmonic Mean	27.33	27.22	27.17	26.96	26.38

Table A12: Impact of I/O on COMM of PC and Impact of COMM on I/O of PC (in sec)

	Type	100 Bytes	1024 Bytes	1250 Bytes	4KBytes	64KBytes
1	SEND	14.81	12.45	12.38	12.25	12.34
2	RECV	12.01	12.81	13.27	13.29	12.65
3	SEND+DL	17.14	15.67	15.49	15.01	14.89
4	SEND_RECV	22.08	10.70	19.13	23.22	14.84
5	2 X RECV+DL	28.55	25.09	10.10	17.46	21.61
6	2 X SEND+DR	50.88	43.95	50.55	48.44	51.43
7	Arithmetic Mean	15.86	16.0	15.09	15.93	21.29
8	Geometric Mean	15.02	15.45	14.55	15.63	18.36
9	Harmonic Mean	14.39	14.95	14.07	15.35	16.65
1	SEND	13.12	11.23	8.9	9.1	9.4
2	RECV	8.30	8.42	8.17	11.13	8.34
3	SEND+DL	14.32	9.28	10.44	10.33	9.84
4	SEND_RECV	14.22	9.64	10.22	10.12	8.82
5	2 X RECV+DL	9.23	10.13	10.32	13.34	13.69
6	2 X SEND+DR	10.12	9.94	8.96	9.35	8.49
7	Arithmetic Mean	10.49	9.76	10.32	10.49	9.76
8	Geometric Mean	10.41	9.70	10.16	10.24	9.61
9	Harmonic Mean	10.34	9.63	10.03	10.02	9.49

Table A13: Impact of FTP on COMM of PC and Impact of COMM on FTP of PC (in sec)

A5. Results for Sparc Station 10 :

	Type	100 Bytes	1024 Bytes	1250 Bytes	4KBytes	64KBytes
1	SEND	27.79	24.87	24.63	24.58	24.82
2	RECV	6.57	4.93	**42.00**	4.76	4.70
3	SEND+DL	28.0	24.75	24.63	24.16	24.33
4	SENd_RECV	10.51	8.65	24.05	7.13	7.89
5	2 X RECV+DL	9.85	8.60	**42.01**	8.13	7.08
6	2 X SEND+DR	14.57	15.35	13.88	10.46	10.99
7	Arithmetic Mean	16.22	17.43	28.53	13.20	13.30
8	Geometric Mean	14.06	20.32	26.64	10.94	10.97
9	Harmonic Mean	12.26	16.50	24.75	9.22	9.19
1	SEND	27.77	24.85	24.60	24.55	24.80
2	RECV	26.15	24.44	23.64	24.19	24.05
3	SEND+DL	26.93	24.73	24.89	24.44	24.74
4	SENd_RECV	26.93	24.64	24.89	24.44	24.74
5	2 X RECV+DL	26.20	24.51	24.98	24.25	24.10
6	2 X SEND+DR	27.74	24.83	24.77	24.61	24.86
7	Arithmetic Mean	27.13	29.6	24.58	24.36	24.48
8	Geometric Mean	27.12	28.6	24.58	24.36	24.47
9	Harmonic Mean	27.11	27.4	24.57	24.36	24.47

Table A14: Impact of CPU on COMM of SS10 and Impact of COMM on CPU of SS10 (in sec)

	Type	100 Bytes	1024 Bytes	1250 Bytes	4KBytes	64KBytes
1	SEND	23.13	20.32	19.92	19.87	20.05
2	RECV	6.25	5.01	**41.99**	4.75	4.64
3	SEND+DL	23.26	20.34	19.97	19.57	19.75
4	SENd_RECV	10.65	8.28	23.99	7.22	8.16
5	2 X RECV+DL	9.89	8.58	**42.08**	8.10	7.08
6	2 X SEND+DR	14.10	13.13	11.01	12.01	13.17
7	Arithmetic Mean	14.55	12.61	26.49	11.92	12.14
8	Geometric Mean	13.07	11.15	23.88	10.44	10.57
9	Harmonic Mean	11.72	12.43	21.40	9.13	9.15
1	SEND	23.12	20.30	19.9	19.85	20.03
2	RECV	21.36	19.67	19.13	19.42	19.35
3	SEND+DL	23.25	20.30	19.95	19.55	19.73
4	SENd_RECV	22.3	20.14	20.02	20.09	19.93
5	2 X RECV+DL	21.68	19.91	20.27	19.57	19.49
6	2 X SEND+DR	23.23	20.49	20.08	19.99	20.54
7	Arithmetic Mean	22.49	20.14	19.89	19.75	19.84
8	Geometric Mean	22.48	20.13	19.89	19.74	19.84
9	Harmonic Mean	22.46	20.16	19.88	19.74	19.84

Table A15: Impact of I/O on COMM of SS10 and Impact of COMM on I/O of SS10 (in sec)

	Type	100 Bytes	1024 Bytes	1250 Bytes	4KBytes	64KBytes
1	SEND	9.15	4.75	5.51	5.80	6.51
2	RECV	9.56	7.31	**45.86**	6.00	7.99
3	SEND+DL	11.13	11.32	10.54	8.32	10.13
4	SENd_RECV	12.73	9.84	27.62	8.38	10.06
5	2 X RECV+DL	12.77	11.59	**50.74**	22.6	10.37
6	2 X SEND+DR	16.37	16.68	16.72	16.34	15.92
7	Arithmetic Mean	11.95	10.25	26.17	11.24	10.16
8	Geometric Mean	11.72	9.53	19.92	9.82	9.78
9	Harmonic Mean	11.50	8.78	14.50	8.77	9.43
1	SEND	16.32	12.43	14.57	15.63	15.33
2	RECV	16.21	14.32	13.23	13.34	17.63
3	SEND+DL	23.32	20.21	18.64	18.78	18.24
4	SENd_RECV	17.73	14.63	14.77	14.99	14.73
5	2 X RECV+DL	18.63	17.46	16.65	17.52	17.63
6	2 X SEND+DR	16.34	14.55	14.82	14.55	16.32
7	Arithmetic Mean	18.09	15.60	15.45	15.80	16.65
8	Geometric Mean	17.94	15.41	15.35	15.70	16.60
9	Harmonic Mean	17.80	15.22	15.26	15.60	16.54

Table A16: Impact of FTP on COMM of SS10 and Impact of COMM on FTP of SS10 (in sec)

	Type	AM	GM	HM
1	SPARC 10	2.496	2.40	2.286
2	DEC ALPHA Station	2.48	2.375	2.266
3	SGI Indy R4000	1.106	0.987	2.077
4	486 DX2 PC	2.154	2.077	1.982

Table A17: Ethernet BTU AM, GM and HM Ratings for various vendor machines