

CMS Services for Micros - Supporting the IBM PC/XT at SLAC *

Joseph W. Wells

Stanford Linear Accelerator Center
P.O. Box 4349
Stanford, CA 94305

August 1984

INTRODUCTION

SLAC has been providing mainframe services to micros on a limited basis for over 10 years. The services were primarily intended to aid program development for the micros which were to be used in a variety of control functions at the Lab. These services had little impact on the central system and were only used by a few people. The growth in the availability, capability, and versatility of the personal computer in the last few years has dramatically changed that. SLAC now has over 120 IBM PC/XT's. Even though some of these PC's are primarily used stand alone, all are connected to the SLAC network and have the capability to use CMS services.

SLAC has been integrating the PC into the CMS computing environment both through the use of existing services and the introduction of special services for the PC. This session describes the services needed by a PC, compares some of the products evaluated at SLAC, and discusses what services SLAC currently provides. A detailed discussion of file transfer experiences is given since data interchange is a prime part of the services needed.

* Work supported by the Department of Energy, contract DE-AC03-76 SF00515.
Invited talk presented at SHARE 63, Miami, Florida, August 13-17, 1984.

COMPUTING ENVIRONMENT AT SLAC

The central computing services at SLAC are provided on an IBM 3081-K24 and an IBM 3033-U16, both running VM/CMS with no other operating systems. The CP portion is VM/SP Release 2 with HPO Release 3.0, while the CMS portion is VM/SP Release 3. Interactive and batch work are done on the 3081-K, while the 3033 is essentially dedicated to batch. The batch system is a highly modified INTEL CMS batch system with a full tape setup system added. On average, over 80% of the CPU cycles go to batch and 600-1000 tape mounts per day are serviced. SLAC provides 24 hour, 7 day a week service, and all processors typically average almost 100% CPU utilization.

CMS provides personal computing to the SLAC user community. There are about 1600 users served by the facility with a peak of 260 connected at a time. Peak period expansion factors normally range from 7-9 and trivial response averages .25 seconds. About 40% of the connected users are on local 3278-4 compatible terminals (43 lines by 80 columns); 40% use Yale ASCII full-screen emulation through the Series 1 (split 60%-40% between 3278-2 (24 lines by 80 columns) and 3278-4 emulation); and 20% use line mode connections through the IBM 3705. SLAC is on BITNET. Offsite access to SLAC includes 300/1200 baud dialup, TYMNET, and a few statistical multiplexors on leased lines.

SLAC has 180 3278-compatible terminals, over 800 ASCII terminals, and over 120 IBM PC/XT's. Onsite ASCII terminals and PC's are connected at 9600 baud async either to a Micom digital switch or to an ethernet using Bridge terminal servers. Users on Bridge or Micom can choose to connect to CMS via full-screen emulation or line mode. They also can connect to various other services or computers at SLAC.

CONNECTING PC'S TO CMS

The IBM/PC users primarily make use of a SLAC developed terminal emulator for accessing CMS through the Series 1 in 3270 emulation mode at 9600 baud onsite or 1200 baud dial-up. The async connection is the least expensive, provides the most flexibility (given the async network at SLAC), and provides excellent full screen access to CMS services. Surprisingly, we have found 9600 baud service to be comparable in speed with the Irma and Forte coax connections through a 3274 controller. In addition, Yale University also provides a basic file transfer facility and Series 1 support to allow file transfer through the Series 1 connection. SLAC file transfer facilities are an improved version of the initial work done at Yale.

Coax connections

All the coax connections have the apparent advantage of speed if they are connected to locally attached 3274 control units. However, all the connections we have seen or have onsite actually run much slower than a real 3278 terminal and have the extremely annoying features of locked keyboards (no type ahead), typing errors caused by nulls,

loss of the left bracket, right bracket, and carat characters from the ASCII character set, and frustrating keyboard sequences. In addition, these coax connections require one of the long slots for the driver board. Our version of the IRMA board updates the screen in a very annoying manner (not top to bottom and not very fast). There may be a fix to this available now. The Forte board worked better, but file transfer was extremely slow (100-200 cps) and was unreliable. We don't know if they have fixed file transfer.

Local PC Networks

None of the local PC networks include connection to CMS on a mainframe. SLAC does have a small network of PC's connected via PCNET. These PC's are running a multi-PC application, and the network is dedicated to that application. We also have a few 3-COM Ethernet interfaces, but they do not allow the PC to connect to CMS.

ASCII Asynchronous Terminal Emulators

A number of good terminal emulators are available and SLAC uses some of them for accessing other systems. The CROSSTALK product is the best product we have seen, but it does not support file transfer through the Series 1 nor does it support Tektronix 4013 graphics emulation. We use KERMIT, TYMCOMM, and CROSSTALK to access various other computer services. The only product that we know that supports file transfer through the Series 1 is the YTERM emulator from Yale. We have used Version 1.0 of YTERM with the old emulator code in the Series 1. Unfortunately, transferring files to CMS (Upload) was extremely slow because there was no DMA transparent input. SLAC now has the latest code for the Series 1 from Yale, and it provides a fast transparent output and input. We have modified, improved, and extended the file transfer program on CMS to use the new facilities. Yale has a version of YTERM that also uses transparent input. YTERM does not support Tektronix emulation nor does it have many of the features of CROSSTALK such as execs. Although Yale distributes the full source for the CMS facilities, it does not distribute source for YTERM nor does it provide a way to extend YTERM through exits.

The AMBASS Terminal Emulator at SLAC

The terminal emulator developed and used at SLAC is called AMBASS as it emulates many of the features of the Ann Arbor Ambassador terminal. In addition to emulating the features required for full screen service, it also has Tektronix 4010 emulation, full printer support, CMS initiated DOS commands, and effective, efficient file transfer services at 9600 baud. The emulator does not perform screen capture to a file, does not allow keyboard re-definition, nor does it have communication execs. It is written in a mixture of compiled BASIC and Assembler support routines.

CMS SUPPORTING SERVICES

Supporting services fall into 3 general categories: 1) enhancing the PC environment, 2) enhancing the CMS environment, and 3) cooperative processing. Facilities and services can either be developed such that the user primarily remains in the PC-DOS environment with services on CMS controlled by the PC, or the user can be in the CMS environment with services on the PC controlled by CMS. Most early links adopted the style of the PC controlling the function. CROSSTALK, KERMIT, IRMA, FORTE, and IBM products like the PC-3270 series all control special services and file transfer from the PC. YTERM and AMBASS are designed for CMS to control the services. The primary advantage of CMS control is that the process can be easily automated and extended by the user with CMS EXECs written in REXX. Wherever possible, SLAC has tried to use existing CMS services for the PC rather than developing special ones. Let's consider each of the three service areas in more detail.

Enhancing the PC Environment

At SLAC, the PC environment is enhanced by providing access to various central services used by other CMS users as well. These facilities include messages, mail, 3800 high speed printing, news, use of BITNET, and file archiving. In addition, we use CMS for program distribution to PC's when licensing allows it. An online PC conference is used to make announcements and exchange ideas. We have 2 special minidisks for the PC community. Any user may install files on the disks. One disk (called PERCOM for PERSONAL COMPUTING) contains documentation and announcements about the PC while the other (PERCOM2) contains programs and PC files for general use. SLAC has a number of cross compilers for the 808x, 68000, and LSI 11 series of processors, but we do not offer cross compilers compatible with the PC-DOS system.

Enhancing the CMS Environment

Most of the services to enhance the CMS environment are actually implemented in the PC or make use of a product. The services include printer support, graphics support, color support, and automated logon. AMBASS provides all these services except for the automated logon. On the CMS side, 2 assembler programs have been written to facilitate communication with the PC. One program, WRTASCI, can send data from CMS to the PC either using the transparent interface on the Series 1 or the CP line mode support through the 3705. REXX execs have been written to support printing files, controlling the PC printer, and displaying Tektronix graphics files at the PC. The second program, PCTRANS, is an enhanced version of the PCTRANS program from Yale. It also uses either the transparent interface on the Series 1 or CP line mode support. PCTRANS provides error-checked file transfer services to the PC. A number of our PC users prepare CMS files using editors on the PC such as the Personal Editor or the Professional Editor. In addition, Lotus 1-2-3 is used for spreadsheet work.

SLAC does not have any of the 3270-PC products, but the windowing support and the multiple-session support on these products are good examples of enhanced CMS services implemented in the PC without host support.

Cooperative Processing

Cooperative processing is the least understood area of PC use, yet it is the key to providing substantial improvement to the user. The prime distinction between cooperative processing and the other services is that an application in CMS communicates with an application in the PC to accomplish a task, and both sides know and take advantage of the intelligence on the other side. The interface can be designed to give maximum performance and high reliability, something that is very hard to do in the typical communication where CMS would simply think the PC was a standard terminal with a fast typist using it. The recent announcement of the 3270-PC/G and PC/GX versions of the PC were an important step towards cooperative processing. The graphics interface from GDDM to these PC's is unique to the PC's and much more efficient than the old 3270 programmed-symbols graphics. Much industry effort is going into providing data base extraction services and improved menu operations by cooperative processing between the PC and host applications. As these cooperative processing products appear, the effectiveness of the PC will increase substantially.

There are various products that support a virtual disk (d:) in a PC local net. A natural extension of this would be to support a VM minidisk as a virtual disk for DOS. I know some work has been done on this, but I don't think any product is available. I believe a more useful function would support the PC disk as a virtual disk on CMS (accessed at any file mode). Just as experience showed with ASP and JES3 that the largest processor should control the smaller processor, CMS on the mainframe should control the interactive session on the PC. I know of no one working on this "reverse" implementation. PC XT/370 implemented the remote disk on the mainframe, control remains on the PC XT/370, and performance is significantly worse than normal response on the mainframe.

At SLAC, we have only just begun to explore some of the cooperative processing issues. We now have the capability to write REXX EXECs that contain PC-DOS commands. The command is sent to the PC for execution, and the EXEC waits for the command to complete before continuing. A key part of the facility is that it is a programmed interface that does not require (though it does allow) a user at the PC to type any keys. We still do not have any application-to-application capabilities.

FILE TRANSFER EXPERIENCES AT SLAC

The ability to transfer files between the PC and CMS is a fundamental service required to provide many of the services. At the SHARE 62.5 Technical Session on the PC in Denver last May, Walt Christensen described the situation very well: "Communicating between a MICRO and a MICRO is EASY; Communicating between a MICRO and a MAINFRAME is HARD." Even when the file transfer works, it is quite slow. At SLAC, I have been studying the behavior and performance of file transfer. SLAC needs an effective, efficient, reliable, high speed (>10K cps) file transfer system between the PC and CMS. To date, no systems have been found to provide 1K cps, even if the link speed is very high (1 Mbits/sec). We decided that if file transfer is to be that slow, then we should minimize our cost of connecting PCs by using asynchronous

communication at 9600 baud. In addition, we needed to support file transfer at 1200 baud async for dial-in from offsite.

When we started looking for a product a year ago, there were very few that worked to the IBM environment. We chose to look at KERMIT (a package from Columbia University), YTERM (a package from Yale University), IRMA (from Decision Support Interface), and FORTE. The tests and results reported are based upon PC-KERMIT and CMS KERMIT used in line mode, YTERM 1.0 and Yale PCTRANS used in line mode, the IRMA and FORTE boards connected as a 3278-2 on a local 3274 controller, and SLAC's own AMBASS 5.0 with SLAC PCTRANS used through the new version of the Yale Series 1 code (Version 11.1.2-A). The decision to develop the SLAC version of PCTRANS grew out of our initial experiences with YTERM and Yale PCTRANS. Line mode is used for YTERM results because we understand that Yale now has a new version of YTERM that performs as well on the Series 1 for upload as for download (upload is PC to CMS; download is CMS to PC) when the latest Series 1 support code is used. I think the results reported here for YTERM line mode are comparable to the current results obtainable on the Series 1 by the new version.

From our experiences, we now have the following recommendations for file transfer requirements.

1. Error checking must be done at both ends. Systems that rely on input to an editor and listing do not work as production facilities. Error checking using Cyclic Redundancy Checking (CRC) is preferred as it is sensitive to bit and byte order as well as value. Table driven schemes for CRC are almost as simple as checksum techniques so the code complexity does not increase to provide a CRC.
2. Retry on detected errors should be done as a part of the protocol.
3. Character set conversion from ASCII to EBCDIC or EBCDIC to ASCII must be done for character files. The ability to transfer any PC file in binary without conversion should also exist.
4. Sets of files should be transferrable in one user operation. This facility greatly helps programs distribution and file backup.
5. A file transfer operation should have low impact on the host CMS system and other CMS users.
6. File transfers should be high speed and/or efficiently use the communication path.

Plot 1 vividly demonstrates the need for higher speeds. The plot shows the transfer time for various line speeds by size of file. The dashed lines show the time if the full bandwidth were used. Even at full 9600 baud bandwidth, transferring a complete floppy (368KB) takes .1 hours (6 minutes) and an entire hard disk (10000KB) needs almost 3 hours. Note how the efficiency varies between KERMIT and AMBASS and how both degrade significantly after 2400 baud. Plot 2 shows the efficiency differences more clearly and shows it for upload and download for all three products. Note that

YTERM upload is significantly worse than YTERM download, but KERMIT and AMBASS uploads are better than their respective downloads. I don't show IRMA or FORTE on these charts because we had trouble getting their versions of file transfer to work. We had the PCCOM software for IRMA from NPM, but we were unable to use it because it required disassembly of the PC/XT to change a chip on the motherboard. We used the XEDIT interfaces provided by IRMA and FORTE instead. Many tests failed, but the ones that succeeded only achieved 600-700 cps for IRMA and 200 cps for FORTE. NPM now provides PCCOM without requiring disassembly, but we have not had time to order and test it.

Plot 3 shows the CPU utilization (TTIME) on a 3081-K for these file transfer systems. Since file transfers can last for minutes at a time, it is very important to have low overhead on the CMS system. Since we would like to support multiple PC's doing file transfer at the same time, the low overhead of AMBASS is a major asset. One would expect the file transfer speed and CPU utilization to depend on a number of factors including

1. Packet size per transaction,
2. Speed of the link,
3. Speed of the PC,
4. Encoding and protocol,
5. Control unit overhead,
6. Reliability of the link.

Encoding and protocol could be important because the communication link has only 7 data bits per character to transmit binary data that has 8 bits per character. In addition, not all 7 data bits can be used because some values are control codes that affect the transmission. Reliability of the link could be important because any retries take additional time and CPU. These tests were done with a normal ASCII file. KERMIT, YTERM, and AMBASS use very similar methods for encoding ASCII data so that would not show a difference between them. No errors occurred during any of the transmissions.

Communicating with an IBM mainframe is essentially a half duplex protocol particularly for line mode on a 3705 and transparent mode on the Series 1. Thus each of the products send a record and wait for an acknowledgement before sending the next record.

The differences noted in CPU utilization can be explained entirely by the difference in packet sizes. Tests were run using the AMBASS PCTRANS protocol where the only variable was the packet size for communication. The results are shown in Plot 4 with a linear scale and Plot 5 with a log-log scale. Plot 6 is simply a copy of Plot 5 where each protocol is plotted using its packet size and CPU utilization. The difference in CPU utilization for the protocols is totally explainable by the differences in packet sizes.

One of the modifications SLAC made to Yale PCTRANS was to increase packet size to 1900 for both upload and download. When we first tried to make the modification, elapsed time at 9600 baud started to increase again for upload as the packet size rose above 500 bytes. The PC is only a 200 kips machine. We were trying to build the entire packet before starting to transmit any characters. This was changed to transmit each character as it is added to the packet. On download, the protocol acknowledges a packet as soon as it has been received without error. If the packet cannot be processed correctly, the error is reported on the next packet.

I have not yet been able to explain all the extra loss of efficiency in bandwidth at 9600 baud. I believe most of it is due to the half duplex nature of the protocols and the delay time before a response is received due to interrupt handling in the 3081 and processing in the Series 1.

Clearly, any protocol should try to have as large a packet as possible. As the speed goes up, the packet size must increase also. The interesting result so far is that we have not had to reduce the packet size for dial use. YTERM must increase their packet size for upload before their performance would be acceptable for 9600 baud use. The KERMIT protocol cannot allow a packet size greater than 95 bytes. For that reason we only recommend using KERMIT at 1200 baud.

CONCLUSIONS

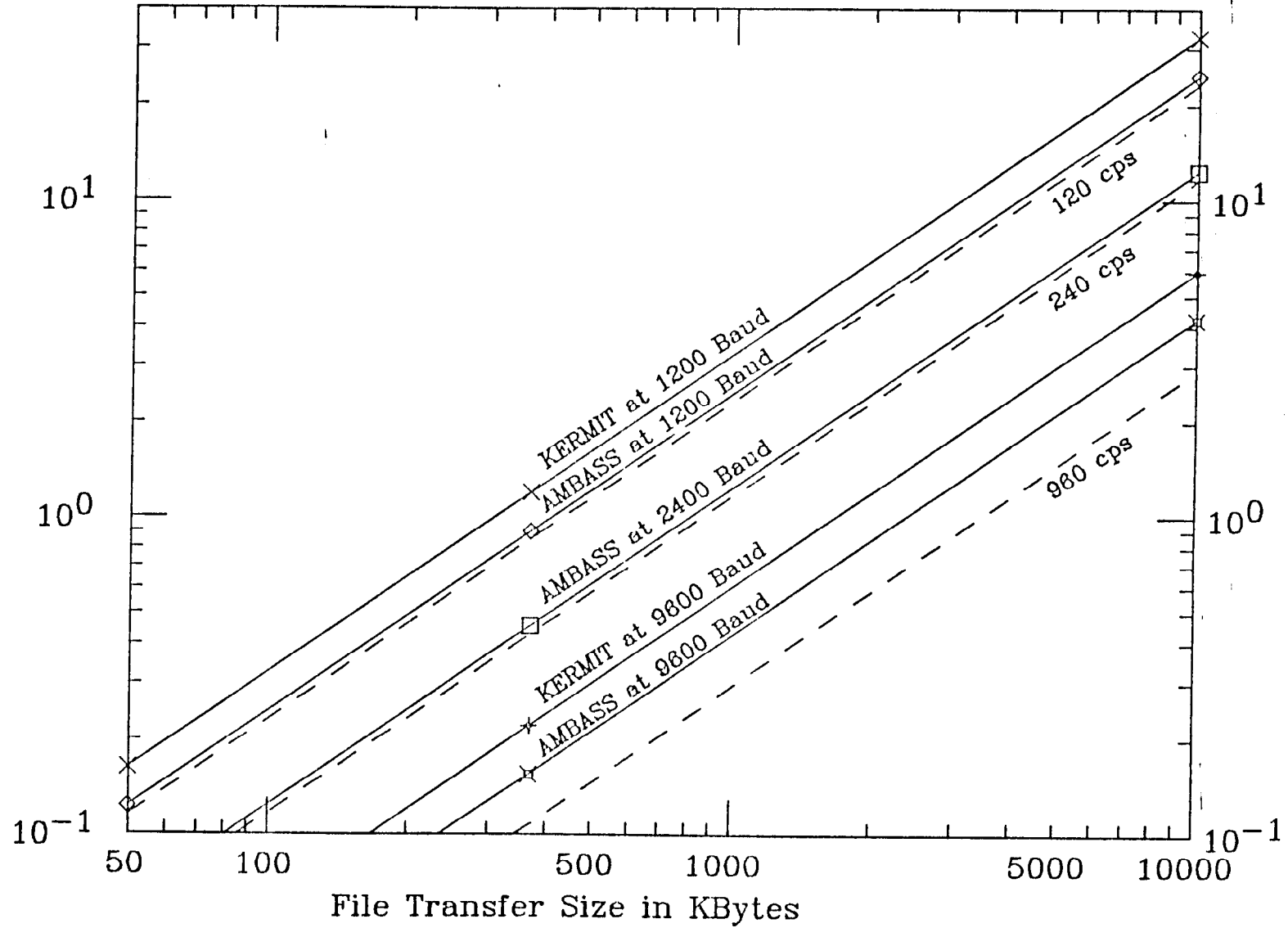
Use of the IBM PC at SLAC with CMS has demonstrated that a wide variety of services are needed and used by the PC user. File transfer services between the PC and CMS work very well and are very reliable, but a speed of 600 cps is not fast enough. The speed of the PC processor is one of the limiting factors. Much higher rates will occur only when the packets can be exchanged as a block without requiring each character in the file to be separately processed.

SLAC expects to continue consolidating and integrating its services. We also expect to see more integrated services coming from the vendors. However, the era of simple emulation on a PC is now being replaced by designed cooperation between the PC and the Host. As easy-to-use, cooperative products become more and more available, the user will be able to use more and more processing to achieve his results. We believe that the human will continue to be able to absorb increased processing power faster than we can supply it. The easier we can make it for the user, the faster his demands will grow.

File Transfer Time vs Size

Plot 1

File Transfer Time in Hours

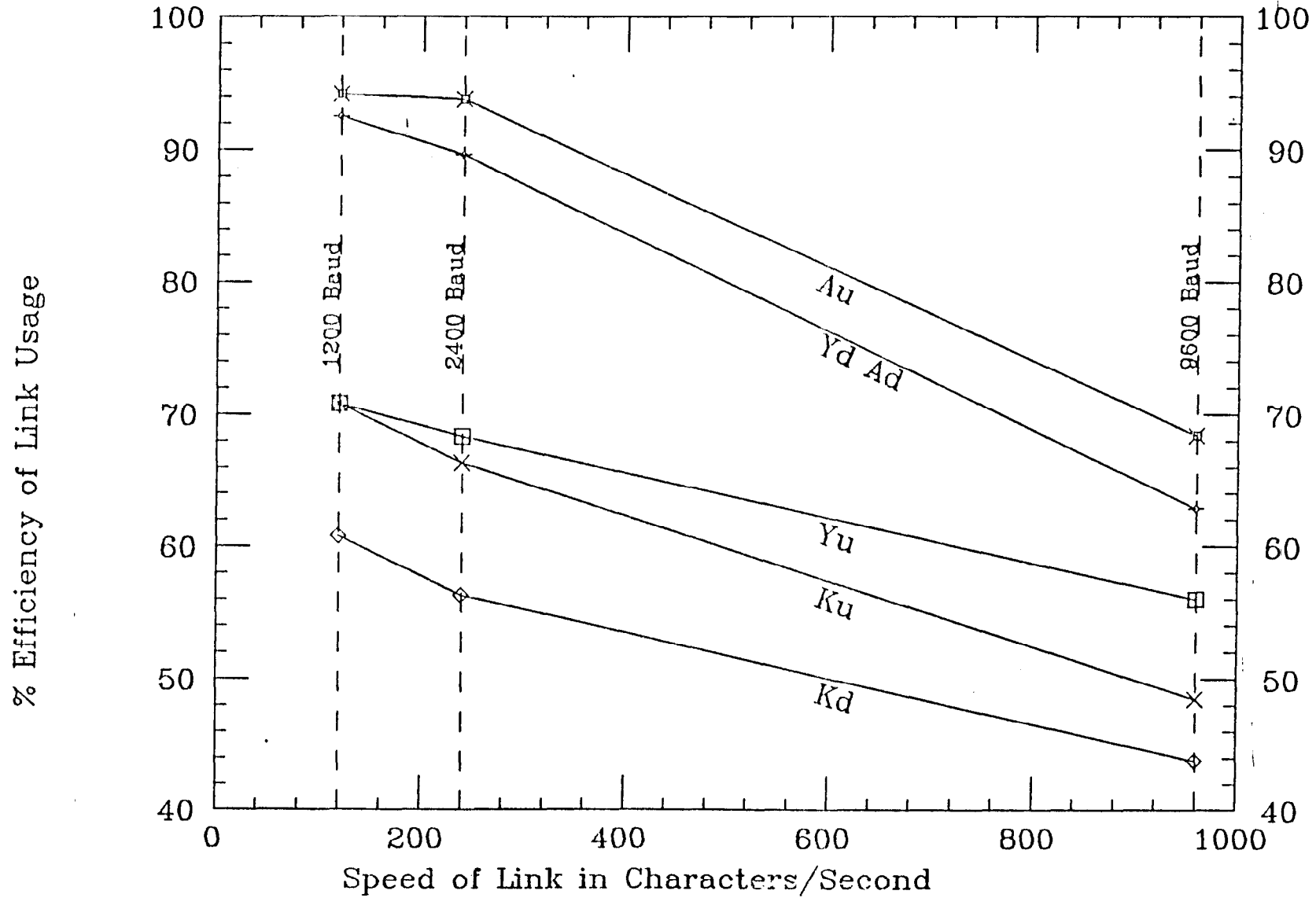


8/ 5/84

JWW

Efficiency of File Transfer Protocols

Plot 2



Ku=Kermit up, Kd=Kermit down, Yu=YTerm up,
 Yd=Yterm down, Au=Ambass up, Ad=Ambass down

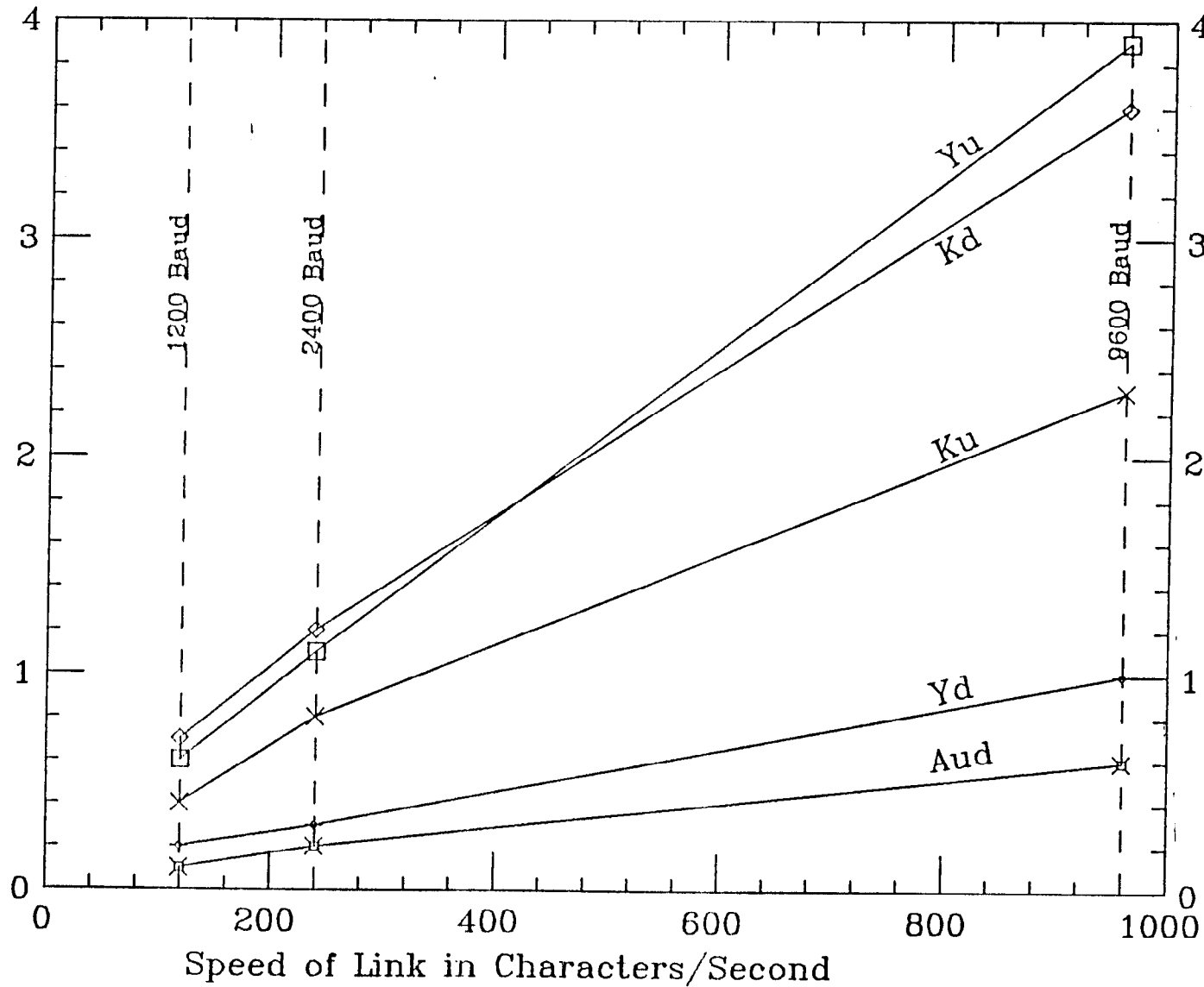
8/ 6/84

JWW

VM CPU Utilization of File Transfer Protocols

Plot 3

% CPU Utilization of an IBM 3081-K



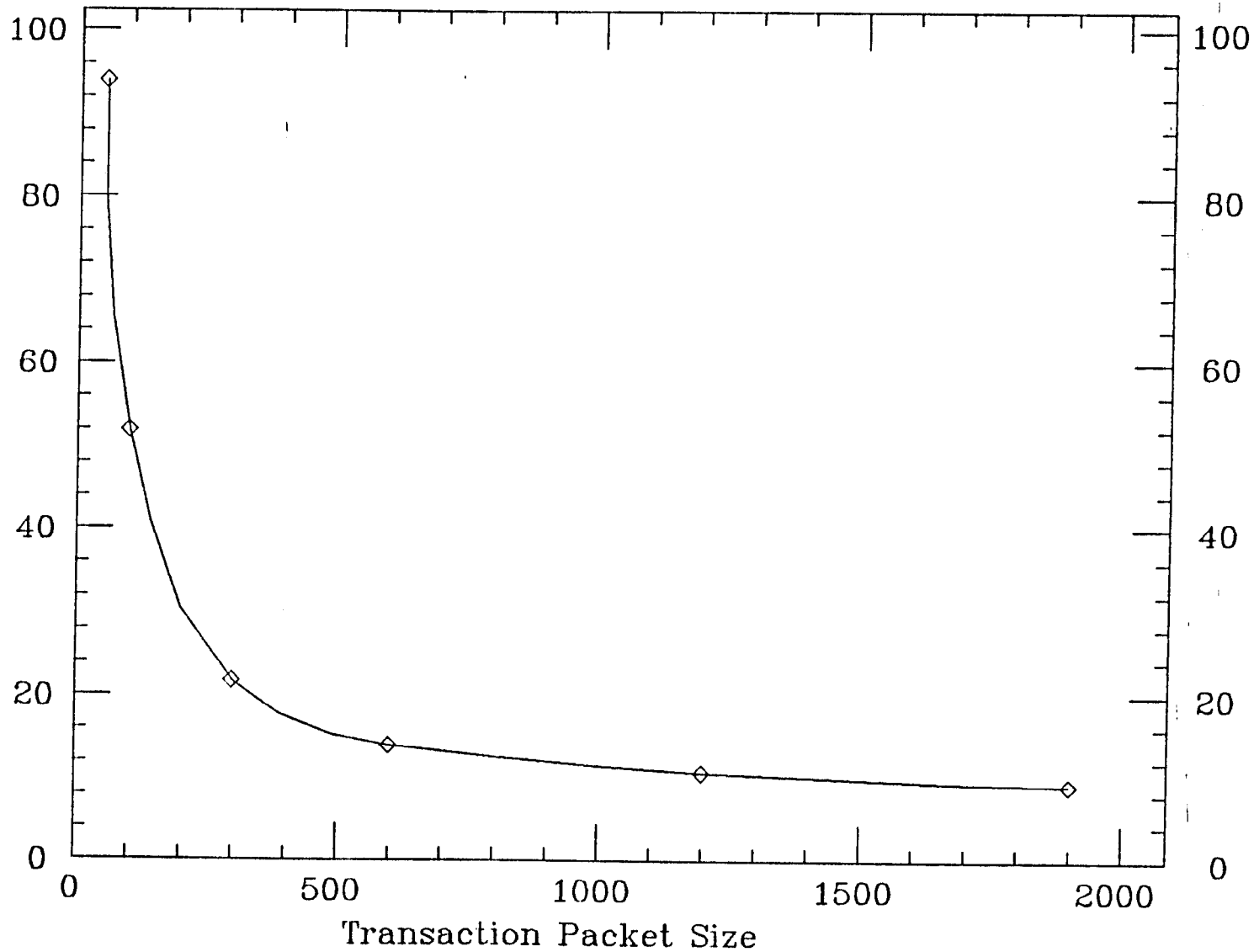
Ku=Kermit up, Kd=Kermit down, Yu=YTerm up,
Yd=Yterm down, Aud=Ambass up or down

8/ 6/84
JWW

VM CPU vs Packet Size for File Transfer (LINEAR)

Plot 4

IBM 3081-K VM TTIME in msec/1000 bytes

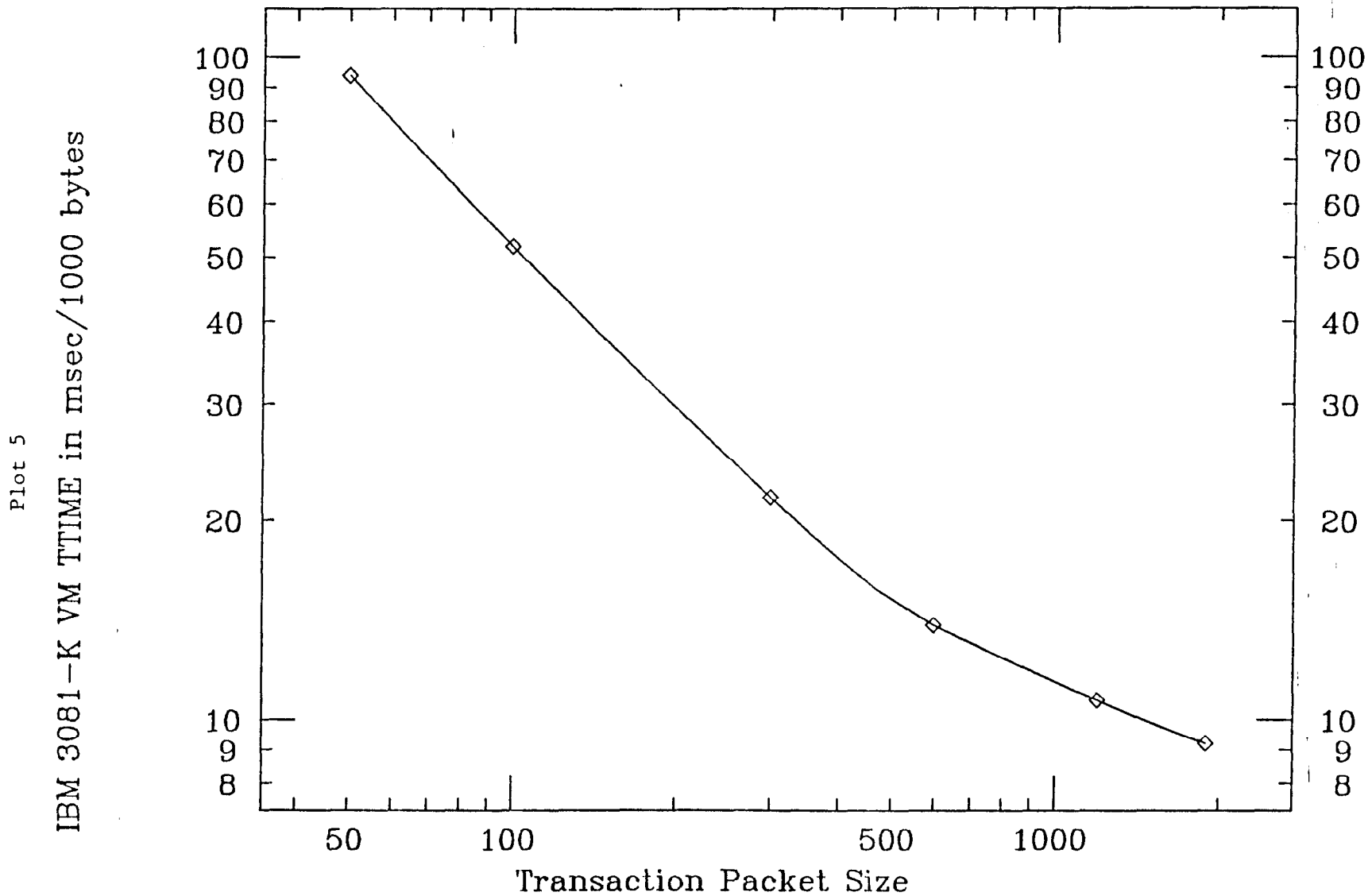


Using AMBASS/PCTRANS with only packet size changed

8/ 6/84

JWW

VM CPU vs Packet Size for File Transfer (LOG-LOG)



Using AMBASS/PCTRANS with only packet size changed

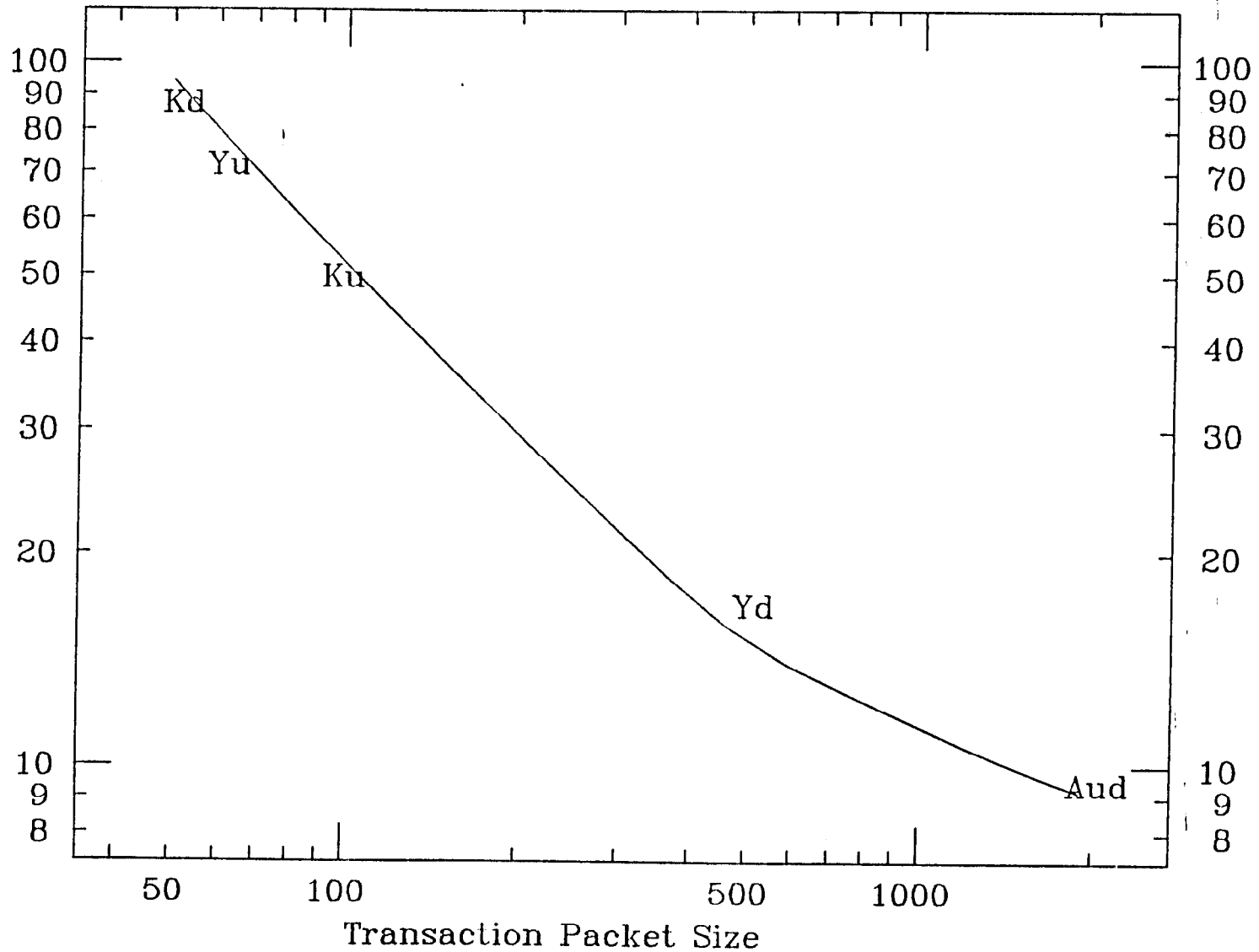
8/ 6/84

JWW

VM CPU vs Packet Size for File Transfer (LOG-LOG)

Plot 6

IBM 3081-K VM TTIME in msec/1000 bytes



Ku=Kermit up, Kd=Kermit down, Yu=YTerm up,
Yd=Yterm down, Aud=Ambass up or down

8/ 6/84
JWW