Bell Laboratories

Cover Sheet for Technical Memorandum

The information contained herein is for the use of employees of Bell Laboratories and is not for publication (see GEI 13.9-3)

Title- Real-Time Measurements of the Prestandard Automatic Trouble Analysis (ATA) System

Date- March 15, 1975

TM-75-5161-1
5161-750315.01TM

Other Keywords- UNIX Operating System
SCC

Author(s)	Location and Room	Extension
J. L. Barberio	HO 2F-508	4390

Charging Case-

Filing Case- 40979
39764

ABSTRACT

The Automatic Trouble Analysis (ATA) System is part of a Bell Laboratories development program to help improve the maintenance of crossbar systems. The ATA system receives trouble card data from remote central office sites, thresholds the data and issues exception reports back to the appropriate sites. The UNIX-based prestandard ATA system runs on a PDP-11/40 minicomputer with an 80K core memory, an RP03 moving head disk and an RF fixed head disk. A real-time analysis of the prestandard ATA system has determined the present capacity and has identified some problem areas.

The prestandard ATA system capacity with an 80K word core memory is reached at about 700 trouble cards input per hour, which is equivalent to approximately seven heavy load sites (marker groups). The real-time analysis has defined problem areas including RP disk overload and high UNIX overhead. Immediate improvements such as increasing core memory to 96K words will increase system capacity considerably. The improvements will be used in the Phase II field trial and the prestandard installation at Patchogue, Long Island. The Patchogue system will have 10 remote sites with an expected total busy hour load of 500 cards per hour which should be well within the capacity of the system.

Pages Text 15 Other 16 Total 31

No. Figures 8 No. Tables 3 No. Refs. ____

Address Label

SEE REVERSE SIDE FOR DISTRIBUTION LIST

BELL TELEPHONE LABORATORIES
INCORPORATED

TM 75 - 5161 - 1

DISTRIBUTION
(REFER G. E. I. 13. 9-3)

COMPLETE MEMORANDUM TO
CORRESPONDENCE FILES:

OFFICIAL FILE COPY (FORM E-7770) —
PLUS ONE WHITE COPY FOR EACH ADDITIONAL FILING CASE REFERENCED

DATE FILE COPY (FORM E-1328)

REFERENCE COPIES (10 IF UNCLASSIFIED, 3 IF CLASSIFIED)

All Members of Department 5161
All Members of "SCC Group"
All Supervision Center 516
R. M. Averill
C. W. Brown
J. H. Carran
R. B. Curtis
W. A. Depp
T. N. Lowry
J. F. Maranzano
V. S. Mummert
R. C. Nance
H. N. Seckler
W. B. Smith
W. D. Snider
C. A. Strohecker
F. S. Vigilante
R. M. Wolfe

COVER SHEET ONLY TO
CORRESPONDENCE FILES:
FOUR COPIES (NONE IF CLASSIFIED)

B. A. Alberts
D. L. Alt
J. J. Amoss
G. W. Arnold
S. L. Arnold
D. L. Bayer
M. J. Billington
I. B. Biren
M. Blum
R. B. Brandt
C. W. Brown
T. A. Burrows
D. E. Bzowy
R. H. Canaday
W. M. Cardoza
H. V. Carlson
J. H. Carran
C. W. Christ
J. E. Cloutier
E. U. Coke
L. M. Cole
T. J. Cook
D. H. Copp
W. P. Coston
J. R. Coulter
L. L. Crume
S. J. Cunningham
D. S. DeJager
T. A. Dolotta
L. J. Donofrio
D. Edelson
T. W. Edmunds
D. L. Eitelbach
H. I. Fleischer
M. J. Fortney
H. G. Frank
M. I. Freedman
R. D. Freeman
B. Freidenreich
L. R. Gardiner
G. W. Gates
E. W. Geer
A. Gerard
A. L. Glasser
L. D. Goldsmith
R. F. Graveman
H. J. Greenbaum
M. P. Gruchowski
R. C. Haight
M. L. Hall
P. Hamilton

E-1932-D (2-74)

2

Bell Laboratories

subject: Real-Time Measurements of the Prestandard Automatic Trouble Analysis (ATA) System - Cases 40979 and 39764

date: March 15, 1975

from: J. L. Barberio
5161-750315.01TM

MEMORANDUM FOR FILE

INTRODUCTION

Automatic Trouble Analysis (ATA) is part of an effort to improve the maintenance of crossbar systems.* Trouble card data is transmitted by an on-site unit via dedicated data facilities to the SCC, where the data is analyzed by ATA software. The ATA system thresholds the data and issues exception reports both centrally and at the appropriate on-site locations. A prestandard version of ATA was field trialed in August, 1974 and standard development of an ATA package is underway. The ATA system is run under the UNIX operating system on a PDP-11/40. The UNIX version used is fully compatible with No. 2 SCCS. System performance during the field trial caused us to question the real-time capacity of the system. An analysis of the real-time consumption has been made to identify the prestandard ATA system's present capacity and to pinpoint the problem areas.

SUMMARY

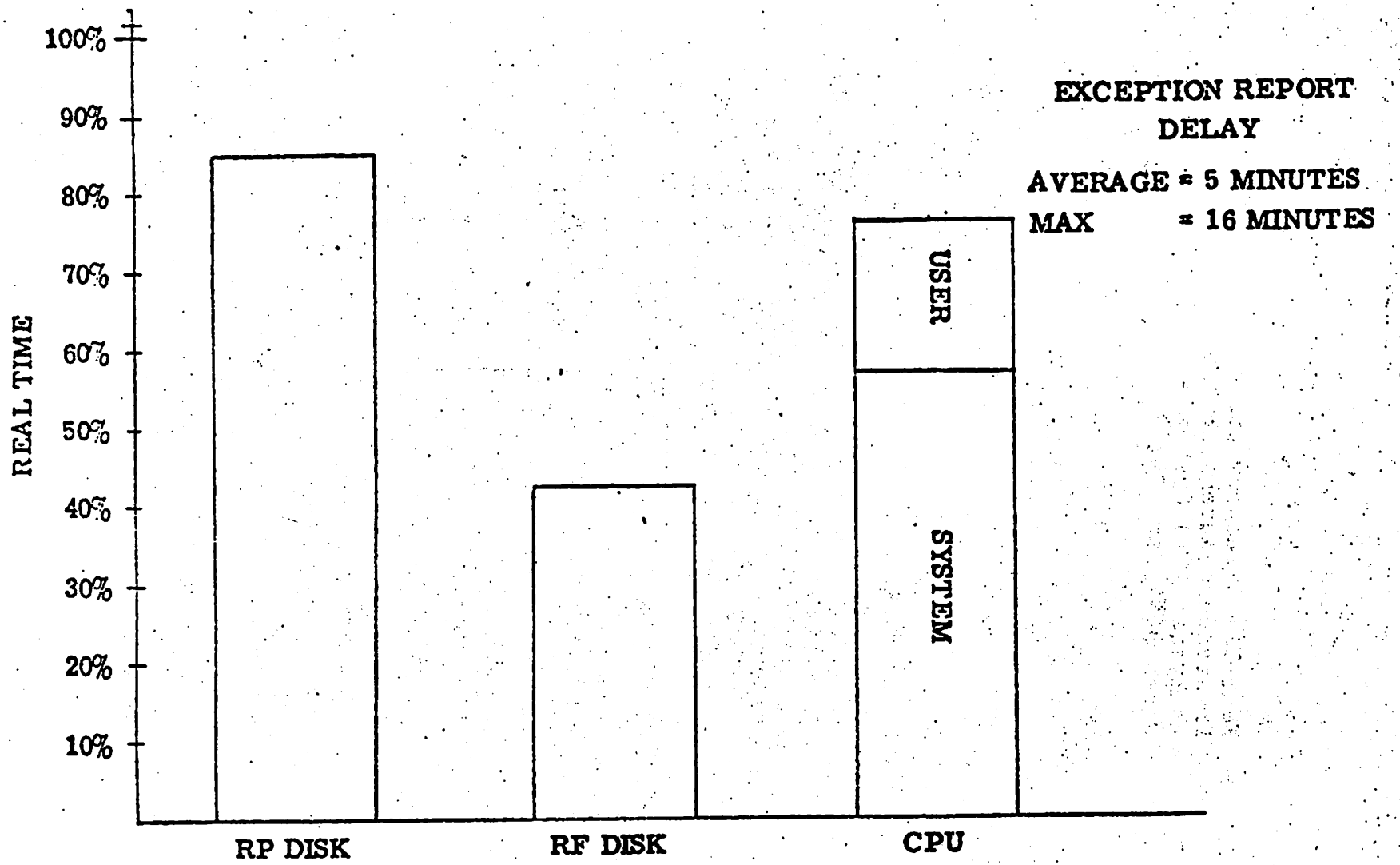
The real-time measurements made have determined the capacity of the prestandard ATA system with 80K words of core memory to be 700 cards per hour, which is equivalent to about seven heavy load marker group sites. At the 700 card rate the system reached an overload condition where it began to lose trouble card data. Figure 1 summarizes the measurement data at the 700 card rate. The average exception report delay at this rate was 5 minutes which is considered acceptable. A minimal amount of craft interaction can be added to these results which would substitute for the overhead of simulation data transmission which is included in all the measurements.

---

* A. F. Rehert, "Electromechanical Switching Maintenance Planning - An Introduction to the 1974 Automatic Trouble Analysis (ATA) Field Trial," Case 40979, April 17, 1974, 5161-740417.01MF.

FIGURE 1

ATA SYSTEM REAL TIME CAPACITY MEASUREMENTS
INPUT TO SYSTEM - 700 CARDS/HOUR



The real-time measurements have determined areas where improvements can be made. The RP03 disk is the most capacity-limiting device on the prestandard system and all efforts, whether in hardware or in software, should be made to decrease disk activity.

The most significant improvement is the addition of more core memory. A 16K word increase in core memory to 96K decreases RP usage by 20 percent which will significantly improve system capacity. A new in-core common area will eliminate some disk activity. A 96K core memory and the new in-core common will be used in the prestandard installation at Patchogue, Long Island. Expanded use of direct disk I/O and larger buffer areas will also lower RP disk usage. Limiting craft interaction especially during peak hours will help further.

The standard ATA system presently being developed will overcome many of the capacity problems encountered in the prestandard version. The standard system will use the new RP04 moving head disk and the new RJ fixed head disk which are both faster than their predecessors. The standard system will use "immediate action" exception reporting which will improve ER response where needed. Off prime-time processing of certain trouble bins will be used to lighten the load on the system during busy hours.

CPU usage is another capacity problem area. Possibly more efficient application code would help. UNIX overhead is consuming a good portion of the available real-time. This may indicate improper use of UNIX by the application code. UNIX was designed as a time-sharing system and the addition of some real-time system features would improve capacity. The implementation of large contiguous files and of asynchronous I/O features in UNIX would be advantageous in the ATA system. The measurements have attempted to resolve the UNIX overhead measurement. At best they have scratched the surface with the software sampling technique. More precise real-time measurements are needed to further analyze CPU usage. Future work planned by Hal Pierson and Bill Snider of Department 5222 will enable more precise measurements of UNIX. If UNIX software cannot be economically improved to reduce overhead, the faster 11/70 CPU may be needed.

Original estimates of ATA system capacity were made in April, 1973. A comparison of the original estimates and the present capacity results is shown in Appendix 1.

ATA SOFTWARE STRUCTURE

The ATA system accepts data from remote office sites in the form of a trouble card record. The record or card is put on a spooling buffer by the Logger program.* A work list is passed from the

---

* ATA User's Manual, Issue July 30, 1974.

logger to the Analysis I (AI) program which puts each card through a trouble decision tree. Analysis I then places the card in the appropriate category or bin. Each bin has a threshold which, when exceeded, initiates a program called Analysis II (AII). Analysis II performs matching of trouble records and calls the Exception Report Generator (ERG) to form an exception report. The Output Distributor (OD) prints the exception report on the appropriate output device. The ATA system has a set of input-output messages for craft-machine interaction. There is also a measurement software package (MEAS) which prints measurement summaries every hour for craft personnel and management.

ATA HARDWARE STRUCTURE

The UNIX-based ATA field trial system was run on a PDP-11/40 processor with 80K words of core memory. There are two DJ-11 multiplexers for communication with remote sites. The pre-standard system has two disks: an RF 256K word disk used to store pointer record files, and an RP03 20 million word disk used to store all other data as well as program storage. The RP disk is also used for swapping programs from core memory.

REAL-TIME MEASUREMENT TECHNIQUES

Real-time measurements in a UNIX-based time-sharing environment is a new area of interest and few real-time measurement facilities existed. Techniques had to be developed to enable various real-time measurements.

Several hardware and software techniques were devised to measure the real-time capacity of the ATA system. Hardware measurements were made by triggering electronic counters with bit-matcher panels attached to the PDP-11/40 UNIBUS.* The bit-matchers can be set to match on specific program addresses and data.

An example of the use of this type of measurement is CPU idle time. The UNIX operating system has an idle state initiated by a WAIT instruction where the processor is not executing any code. Only an interrupt such as a clock interrupt can take the CPU out of idle. To measure the idle time, one bit-matcher is set to the address of the WAIT instruction which signals the beginning of idle. The second bit-matcher is set to the address in the interrupt handler signaling the end of the CPU idle. The electronic counters monitor the signals from the bit-matchers. One counter simply counts the number of occurrences of CPU idle

---

* I. D. Buck, "EADAS - PDP-11/20 Debugging Match Circuit," 5222-720802.01MO, August 2, 1972

while the other counter time interval averages the duration of tne CPU idle. Multiplying the two results gives the amount of idle time over some period and a percent idle time can then be calculated.

Another similar hardware measurement that was used was to directly connect the counters to the computer backplane hardware. An example of the use of this technique is the User mode measurement. The 11/40 processor has two modes of operation: the Kernel mode and the User mode. The UNIX operating system code is executed in Kernel mode and the ATA applications code is run in User mode. The User mode measurement is mady by connecting the counters directly to the User signal lead in the CPU hardware. The counters are used in the same manner as described in the idle measurement to yield the percent of real-time spent in the User mode.

The complexity of the ATA system requires more complicated measurements which are very difficult to do with simple hardware. There is one device called the hardware monitor which can do more difficult measurements. The hardware monitor has been used on a large IBM system and will be used by the UNIX Support Group on a PDP-11/45 in the near future. The monitor was not available for use on the ATA system.

Software techniques were used to measure the more complicated areas of the ATA system which could not be measured by hardware. The inherent problem with software measurements is the effects of the measurement software itself on the system being measured. The effect must be understood and minimized to insure accurate results. One software technique developed was the use of the UNIX system call TIMES for every ATA process created. The TIMES call gives the total time spent running a process over some time duration. When each process is killed from the console TTY, the resulting time summary is printed. This was done for each ATA process, Logger, AI, AII, etc. The result was the total time spent in each process for Kernel mode and for User mode. This technique gave a good measurement of the ATA application codes real-time consumption. It, however, did not measure the UNIX overhead which includes scheduling, swapping, and interrupt handling.

Another software technique called Profile was devised to sample the operating system routines and provide a histogram of all the operating system code. A UNIX PROFILE utility exists to sample the application code.* It, however, does not sample the operating system. To sample the operating system, a subroutine was added to

---

* K. Thompson and D. M. Ritchie, "UNIX Programmer's Manual," June, 1974, Fifth Edition.

the clock interrupt handler. The subroutine determines which, if any, operating system routine was interrupted by the clock and pegs an entry in one of two tables of operating system pointer addresses. One table is for ATA initiated calls (process id $\neq$ 0) such as disk reads and writes. The other table is for UNIX initiated calls (process id = 0) such as swapping on the disk and idle. The Profile tables are printed as a histogram of events in the operating system.

The Profile technique provided a useful amount of information about the operating system. The technique, however, yielded some inaccurate results (explained later) because it is driven off the clock interrupt.

An obvious solution to the problem of driving Profile from the clock interrupt is to create a new interrupt clocked at some reasonable rate and dedicated to driving Profile. Another more reasonable approach is being implemented by Hal Pierson and Bill Snider of Department 5222. They are preparing to make real-time measurements on their UNIX-based No. 2 SCCS system using a second minicomputer and a DR-11 interface to the No. 2 SCCS system. This scheme will transmit sampling data from the SCCS system to the second mini, thus producing only a small load on the SCCS system. This measurement technique would be useful for future real-time analysis of the ATA system.

ATA INPUT TROUBLE CARD SIMULATION

A simulation scheme was developed to load the ATA system and proved to be a valuable tool in the analysis of real-time capacity of the ATA system.* The simulation scheme provided for variable simulation loads on the system of 100 to 1000 cards per hour total from all sites. A 100 card-per-hour load is considered to be a heavy per site load. The only load on the system that could not be simulated with this scheme was craft interaction on TTY terminals.

A simulation run was made on the system in the following manner.

1. The Main Record Store (MRS) and the pointer record files were loaded from magnetic tape with about 2 weeks worth of actual trouble card data from three sites. This was done to insure the system was reasonably settled before a simulation run was made.

2. ATA was started without initialization.

---

* J. L. Barberio, "Automatic Trouble Analysis (ATA) Input Trouble Card Simulation," 5161-74031.01EN, October 31, 1974.

3. Trouble card data collected previously from actual field test sites was transmitted into the simulated site data lines at the appropriate card rates.

4. After a 10-minute buildup period, the real-time measurements were begun.

5. The simulation run typically transmitted 1000 cards and lasted an hour or more depending on the selected card rate. Real-time measurements were made during the simulation run.

Identical simulation runs could be run as many times as needed by repeating the above steps. Simulation provided a flexible way of loading the ATA system for real-time measurements.

ATA REAL-TIME MEASUREMENTS

Measurements of the ATA system were made with two goals in mind. One was to define the real-time capacity of the present ATA system. The second goal was to determine if any immediate improvements in the software could be made for deployment in the Phase II field trial and the Patchogue prestandard installation in Long Island.

The measurements that follow were taken with simulation data from two or three sites coming into the system with the total card rate as a variable. Craft interaction was absent during the measurement runs; it was considered separately. The overhead of transmitting the simulation data is included in all the measurements. That overhead was itself measured and is small. Included in the ATA system is a software measurement package (MEAS) which prints hourly summaries for TelCo craft and management. The MEAS package is a part of ATA and is not to be confused with the real-time measurements. The MEAS package was suspected to be a large consumer of real-time on the system. Therefore, many of the real-time measurements were made with and without the MEAS package and its associated program interfaces. The measurement results shown in the figures are labeled "NORMAL" and "MEAS OFF" appropriately.

The usefulness of the RF disk was another question raised. Therefore, some of the measurements were made without an RF disk. These measurements are labeled "RP ONLY".

The first measurement made was the Exception Report (ER) delay. The delay indicates how long the system took from the time the trouble card that generated an ER was received to the time the

system responded by issuing an ER. ER delay is one of the prime indications of system load in the prestandard system. Figure 2 shows the average ER delay for various card rates. The NORMAL system had a 5 minute average delay at the 700 card-per-hour rate. At the 700 card rate the prestandard system with an 80K core memory began to lose cards due to overload. Therefore, the prestandard system capacity is reached at the 700 card rate. Assuming the heavy load site sends 100 cards per hour during peak hours, the prestandard system capacity will be reached at seven sites. The effects of the MEAS package is shown by the MEAS OFF curve. The RP ONLY curve clearly shows the need for the RF disk. Figure 3 illustrates the maximum ER delays that can be expected at various card rates. With the NORMAL system at 700 cards per hour, maximum delays of 16 minutes can be expected. Cards input to the system beyond the 700 card rate were lost by the system due to overload. At the 1K card rate, 200 cards were lost thus making the effective card rate 800 cards per hour. The actual input card rate of 1K cards is shown in Figures 2 and 3 rather than the effective card rate.

The RF (fixed head) disk is used to store the pointer records for the main record store (MRS) on the RP disk. The RF has a 256K word capacity which is almost totally consumed with the pointer information. Figure 4 shows that RF usage is approaching 50 percent which is no problem. Since the MEAS package does not access the RF disk, the "NORMAL" and "MEAS OFF" results are on one curve. The need for the RF disk is verified throughout the measurement results by comparing the NORMAL system curves with the RP ONLY disk curves.

A measurement of the UNIBUS* usage was made by connecting the electronic counters directly to the bus signal leads. The beginning of bus usage is indicated by the "SACK" signal and the end of the bus usage by the "MSYN" signal. At the heavy load of 1000 cards per hour, the UNIBUS usage was only about 30 percent for the prestandard system. A rough calculation of UNIBUS usage will verify the 30 percent result. The UNIBUS is not a capacity limiting medium in the prestandard system.

Another useful direct hardware measurement made on the ATA system was the processor KERNEL/USER mode usage. The lower portion of Figure 5 shows the real-time spent in the USER mode under various loads. The ATA application code is the only code executed in

---

* "PDP-11 Peripherals Handbook," Digital Equipment Corporation, Copyright 1973.

FIGURE 2

AVERAGE ER DELAY VS CARD RATE

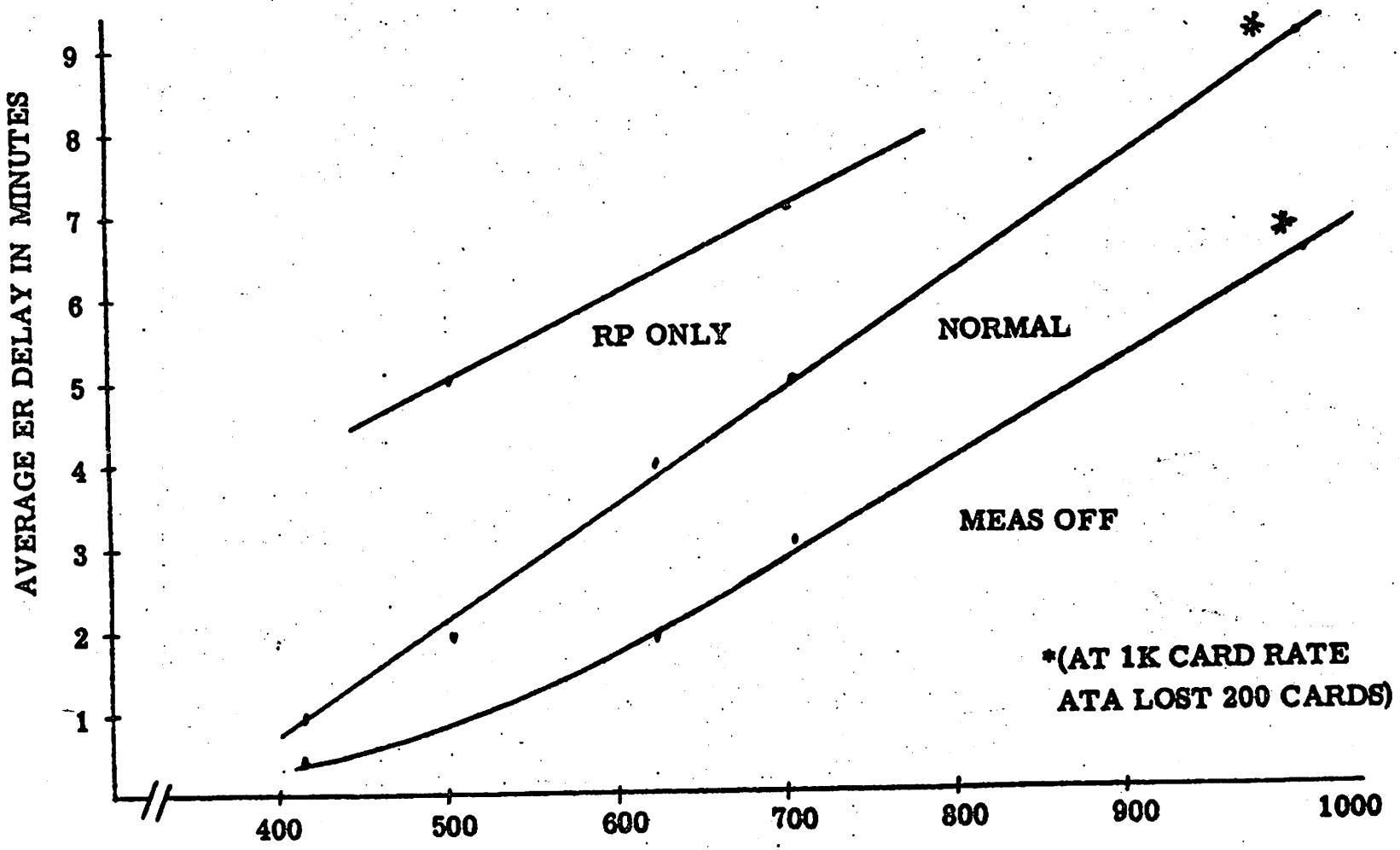


FIGURE 3

MAX ER DELAY VS CARD RATE

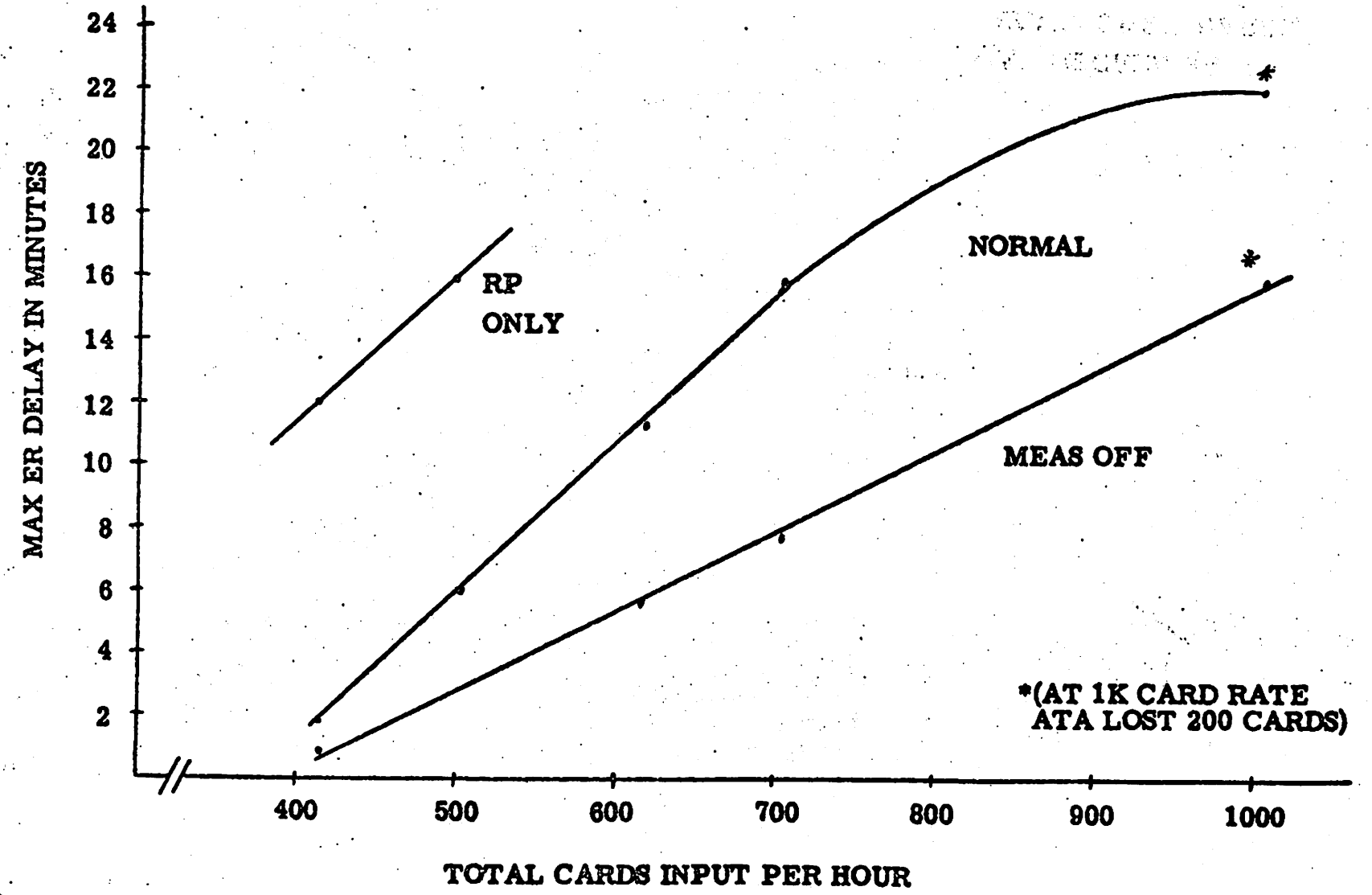
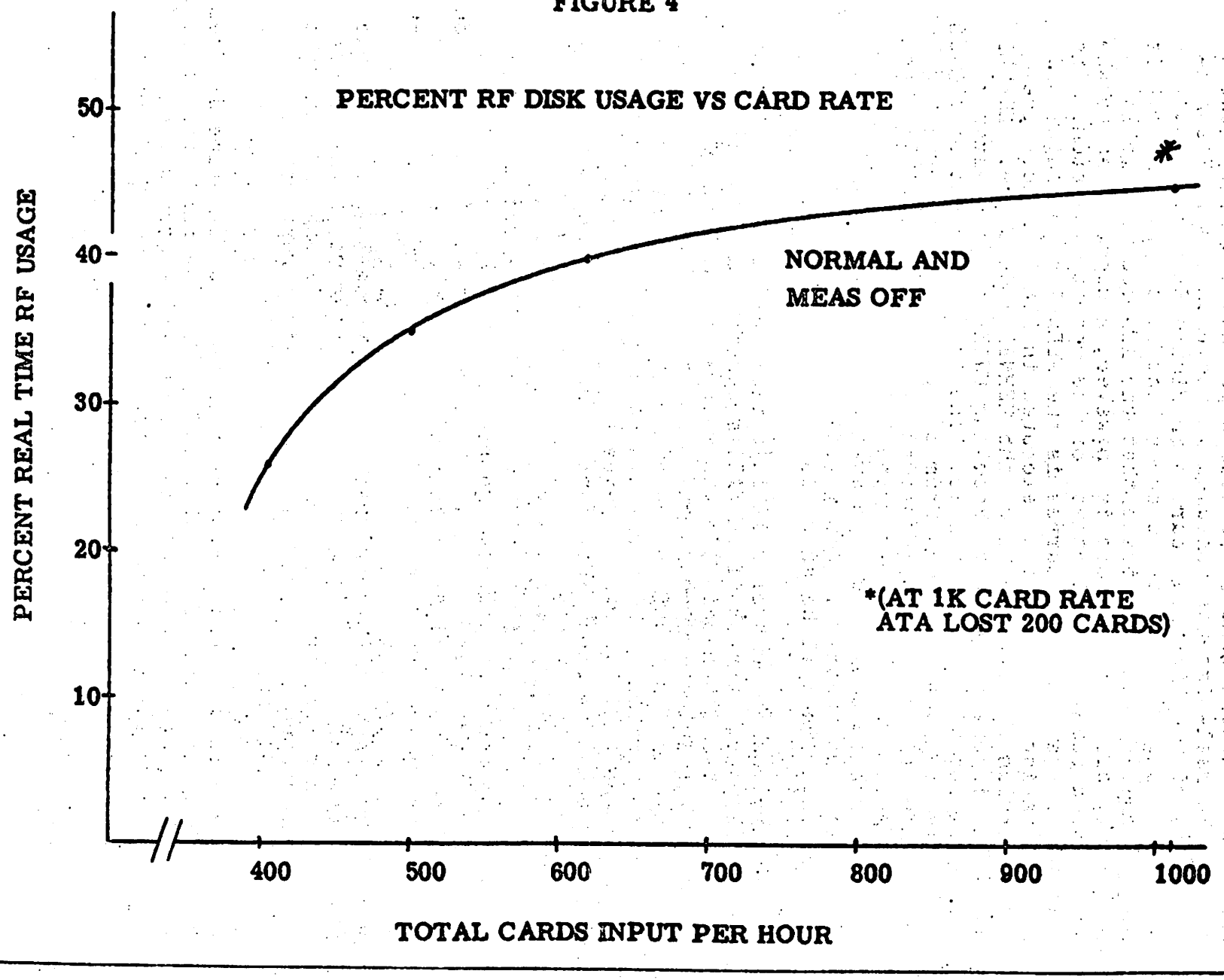


FIGURE 4


PERCENT RF DISK USAGE VS CARD RATE
PERCENT REAL TIME RF USAGE
50
40
30
20
10
NORMAL AND
MEAS OFF
*
*(AT 1K CARD RATE
ATA LOST 200 CARDS)
400
500
600
700
800
900
1000
TOTAL CARDS INPUT PER HOUR

USER mode. At the load of 700 cards per hour, only 16 percent of the CPU real-time was consumed in the USER mode. The non-linearity of the USER mode curves in Figure 5 in the 700-1K card rate range is due to the cards lost by the system in this range. These results indicate that the ATA code itself is reasonably efficient. There is, however, a question as to the application code's efficient use of the operating system. The inverse of the USER mode curves is the KERNEL mode usage. All UNIX operating system code is executed in KERNEL mode. The upper portion of Figure 5 shows the KERNEL mode usage. KERNEL mode time is the time spent executing UNIX operating system code plus the time spent at idle. The idle component of KERNEL mode time was subtracted out from the graph. Therefore, the KERNEL mode time shown was consumed executing operating system code. These results show that the UNIX operating system is using a high percentage of the available CPU real-time. This UNIX usage was further investigated with software measurement techniques which will be shown later.

A hardware measurement of UNIX idle time was made. Idle is the time UNIX places the CPU in a WAIT mode where no code is executed. The CPU remains in the wait mode until an interrupt occurs. The idle time could exist for two reasons: absence of work (true idle) and waiting on a device such as a disk operation to complete. With the system completely idle (no ATA running) the CPU idle was 95 percent. The 5 percent was consumed by the UNIX clock interrupt. The overhead of transmitting simulation data was determined with the rest of the system idle. At the 420 card rate, 1 percent of the CPU was used; at the 1K card rate, 5 percent of the CPU was used. This overhead is low and can for the most part be ignored. Figure 6 shows the CPU idle time under various loads. The true idle and the waiting on a device components of CPU idle are difficult to separate. At heavy card rates, the "waiting on the RP disk" component is most likely a good percentage of the idle time, since the RP is so heavily in use. At low card rates the inverse is probably true. While the system is at idle the RF or RP disk may be transferring data directly into core memory (DMA). If the DMA finishes while the system is at idle, then a disk interrupt upon DMA completion will take the processor out of idle.

At the 1K card rate only 15 percent of the CPU real-time is free. Considering that the effective card rate is 800 cards per hour, the 1K input card rate makes the CPU real-time even worse. These results show that the 11/40 processor usage is approaching maximum capacity.

FIGURE 5

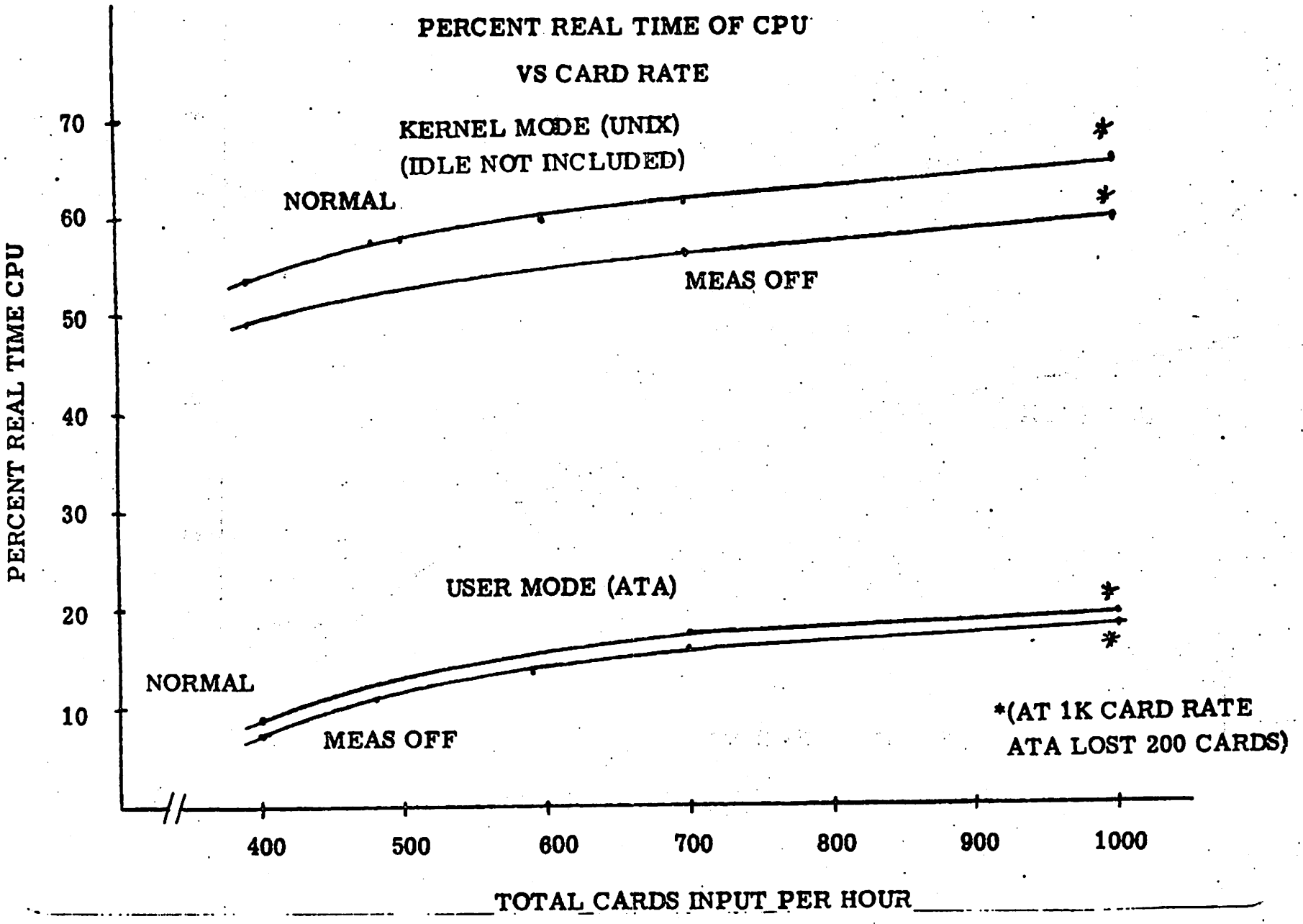
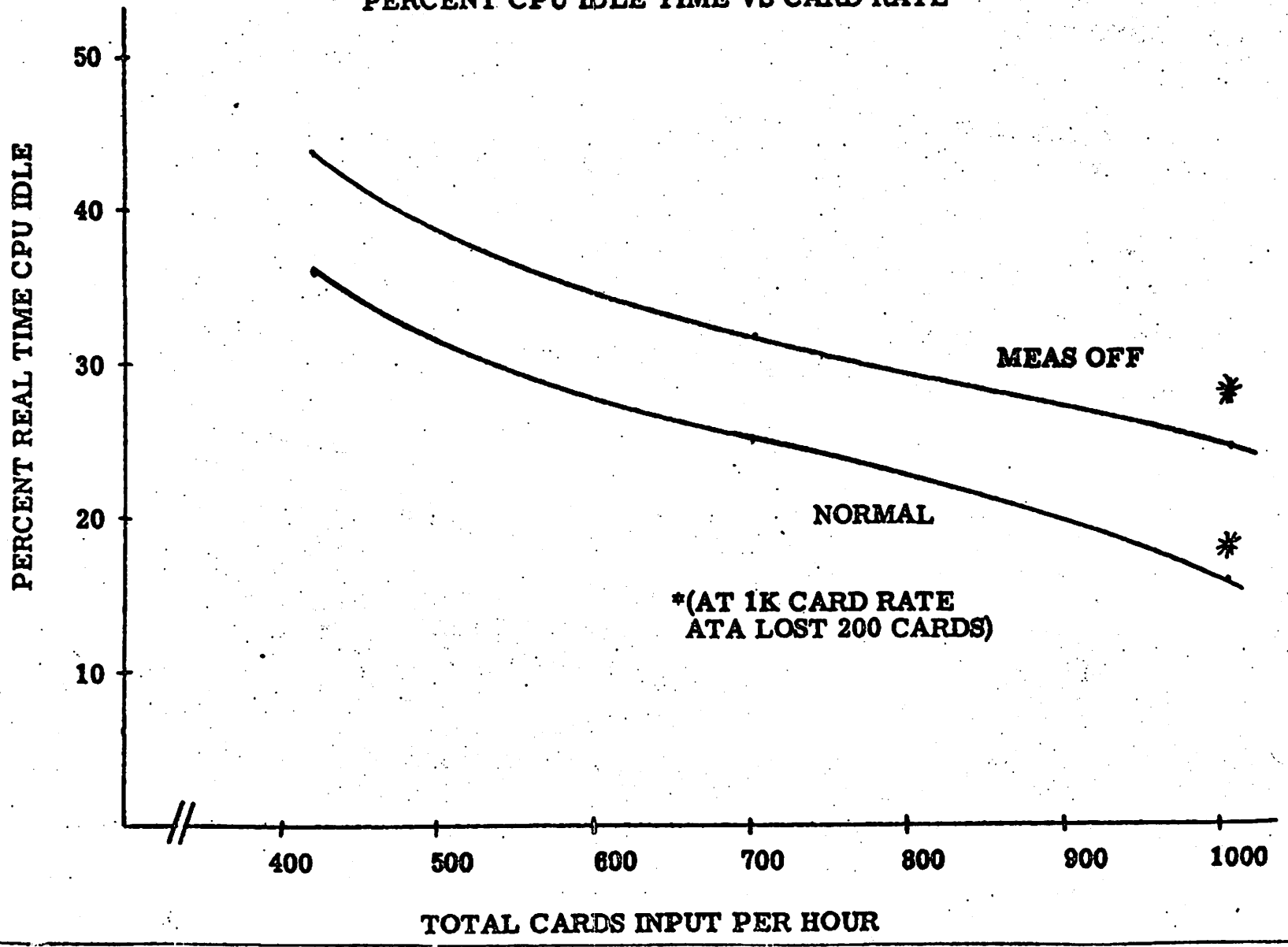

PERCENT REAL TIME OF CPU
VS CARD RATE
KERNEL MODE (UNIX)
(IDLE NOT INCLUDED)
NORMAL
MEAS OFF
USER MODE (ATA)
NORMAL
MEAS OFF
*(AT 1K CARD RATE
ATA LOST 200 CARDS)
PERCENT REAL TIME CPU
70
60
50
40
30
20
10
400
500
600
700
800
900
1000
TOTAL CARDS INPUT PER HOUR

FIGURE 6

PERCENT CPU IDLE TIME VS CARD RATE



The final measurement made with hardware techniques was the RP (moving head) disk usage. The RP disk contains the main record storage (MRS) for the ATA system. All data except for the pointer records on the RF disk are kept in the MRS on the RP. The RP is also used for program swapping to and from core memory. Since the ATA programs are together larger than a core memory size of 80K words, they have to be swapped into core.

Some initial measurements of the RP were made with the system at idle (no ATA running). The RP disk at idle was used 1 percent of the time. This usage is accounted for by the UNIX (SYNC) function which updates the RP every 30 seconds. The overhead of transmitting simulation data for the measurement run was also determined with the rest of the system at idle. The RP usage ranged from 1 percent at the 420 card-per-hour rate to 6 percent at the 1K card-per-hour rate. The simulation overhead on the RP is low and can for the most part be ignored.

The RP disk usage under the various load conditions is shown in Figure 7. The RP disk is the most critical capacity limiting device on the system. RP disk usage reached saturation (100 percent) at the 1K card-per-hour rate. At the 1K card rate, 200 cards were lost by the system between the Logger and AI. At this rate the RP disk usage was at saturation and 200 of the 1K cards input were thrown on the floor. The effective card rate was thus 800 cards per hour. At the other card rates, the RP disk was not saturated and there were an insignificant number of cards lost. The benefits of the RF disk are again shown comparing the RP ONLY curve with the NORMAL system curve.

A breakdown of RP disk usage was attempted to determine how much usage was due to swapping overhead. An experiment was made where all swapping was moved to an RK disk. With all swapping removed from the RP disk, its usage decreased by about 30 percent indicating that swapping has a high overhead on the RP disk. A further experiment was made by increasing core memory from 80K to 96K words.

With 96K of core the Logger, AI, AII, and the ERG can be kept in core simultaneously almost all of the time. Present program sizes are given in Table 1. At the 700 card rate, the RP usage went from 85 percent to 65 percent with the 96K core memory. A further increase in core memory size should decrease the RP swapping overhead even more.

The software measurement technique using the UNIX TIMES call was used to produce the results in Figure 8. The time the CPU spent in each of the ATA functions is shown. Each process

FIGURE 7

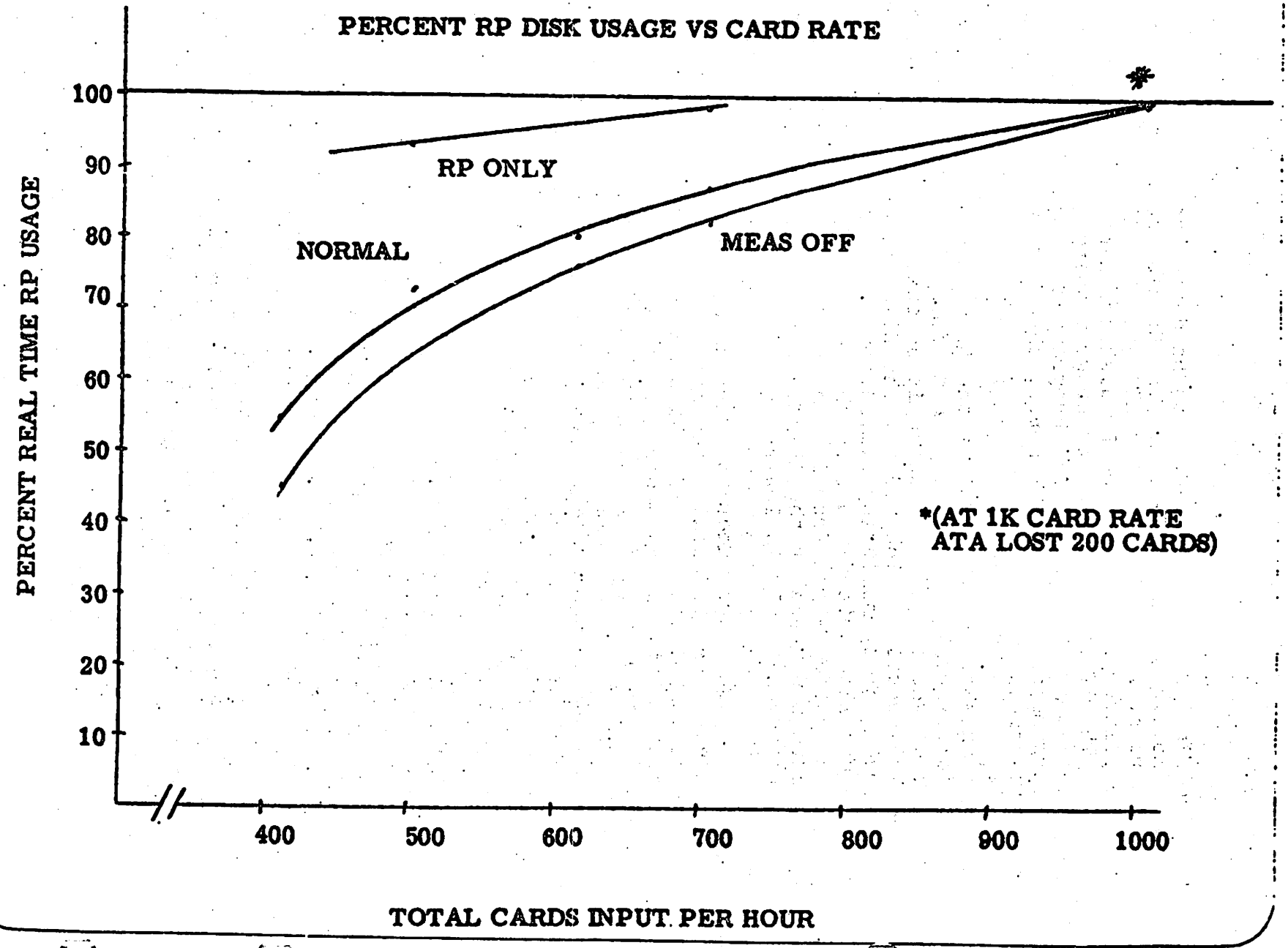
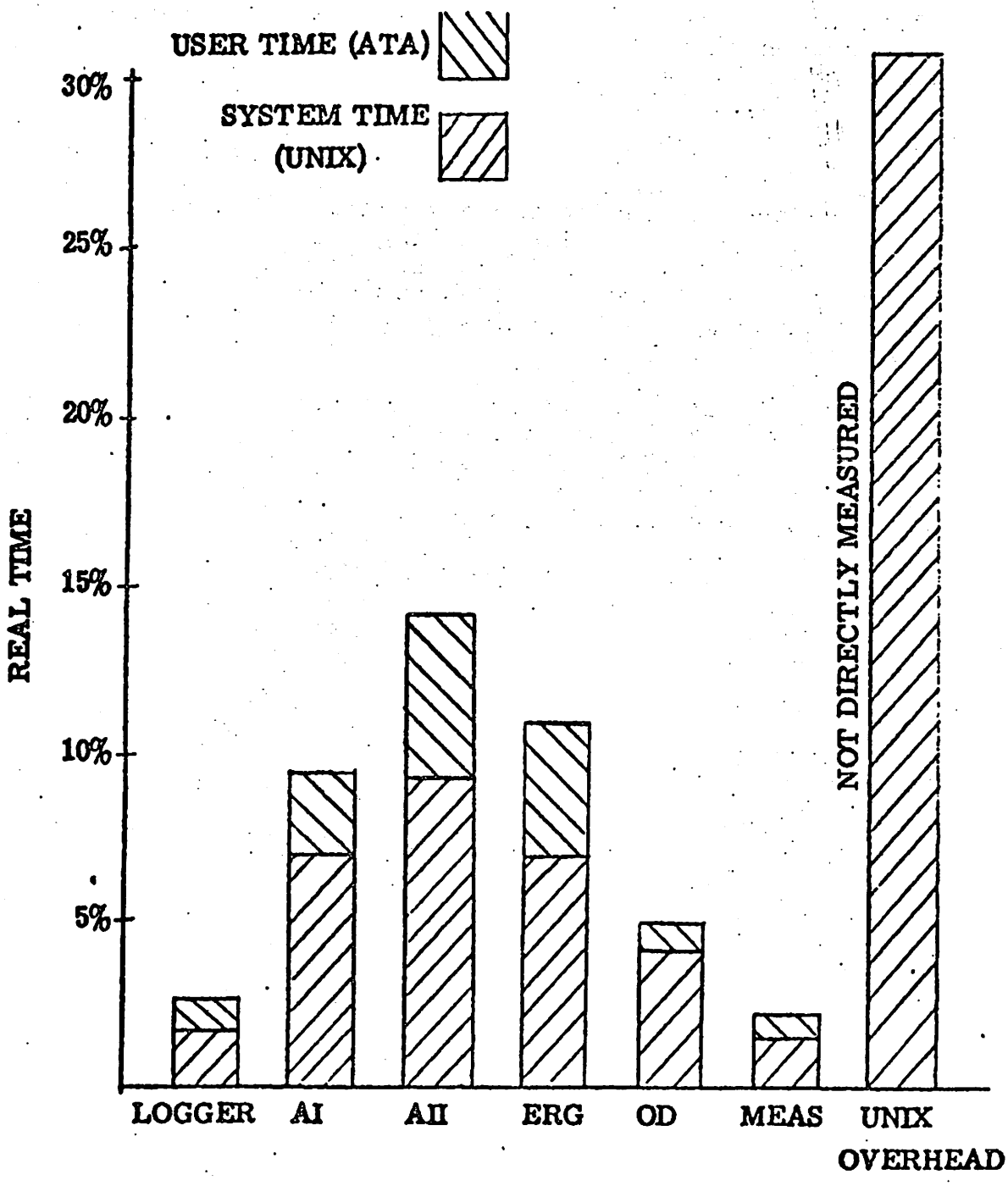

PERCENT RP DISK USAGE VS CARD RATE
PERCENT REAL TIME RP USAGE
100
90
80
70
60
50
40
30
20
10
RP ONLY
NORMAL
MEAS OFF
*(AT 1K CARD RATE
ATA LOST 200 CARDS)
400
500
600
700
800
900
1000
TOTAL CARDS INPUT PER HOUR

FIGURE 8

ATA SYSTEM REAL TIME MEASUREMENTS

INPUT TO SYSTEM - 700 CARDS/HOUR



time is separated into USER mode time and SYSTEM mode time. The USER time was consumed executing ATA application software while the SYSTEM mode time was used executing UNIX operating system software in behalf of ATA. The SYSTEM time includes such UNIX operations as disk reads and writes. The sum of the real-time consumed by ATA (42 percent) plus the CPU idle time (25 percent) leaves 33 percent of CPU time which has been labeled UNIX overhead. This overhead includes UNIX scheduling, interrupts, and swapping.

With the apparently high UNIX overhead, measurement efforts turned to investigating UNIX software rather than ATA software. A sampling technique called Profile was devised. Table 2 shows the resulting histogram printed by the Profile software after a 700 card-per-hour simulation run. The first column of the histogram is the name of all the UNIX system routines. The next two columns are the sum of the peg counts taken at clock interrupt by the Profile sampling subroutine for user (ATA) and system (UNIX SCHEDULER) initiated work. The peg counts indicate the total number of times that the clock interrupted the indicated UNIX routine during the 1 hour test run.

Because Profile was driven by the clock interrupt, some problems were encountered. On the particular measurement run shown in Table 2, Profile indicates a 40 percent CPU idle time while a hardware measurement made concurrently with Profile measured 25 percent CPU idle.

The inflated idle measurement obtained with Profile is explainable. The clock interrupts every 16.6 milliseconds and schedules events in the operating system such as swapping, wake up on time sleeps and I/O character delays. The tasks scheduled by the clock interrupt may complete before the next clock interrupt occurs. Since the Profile sampling is made at the beginning of the clock interrupt, the execution of these tasks scheduled by the clock interrupt may not be seen by Profile. Also, since the CPU is at idle a good percentage of time, the probability of a clock interrupt occurring during the idle period is high. The result is that Profile will never see some of the clock interrupt scheduled tasks and will inflate the CPU idle measurement. To further illustrate this assume that the system is completely idle. The only interrupt taking place is the clock interrupt. The actual CPU idle time would be about 95 percent with the remaining 5 percent being used by

the clock interrupt to determine that there is no work to do and to return the CPU to idle. The Profile technique would show 100 percent CPU idle since the 5 percent usage is never seen by Profile. Driving Profile off the clock interrupt distorts the idle measurement and may blind Profile from some of the tasks scheduled by the clock interrupt.

The other measurements obtained with profile are more reasonable than the inflated idle measurement and provide a good amount of information about UNIX. Up to this point, the CPU idle has always been considered as time spent in the WAIT with no code executing. Looking into the UNIX software, however, it can be shown that a significant amount of time is spent locking for work to do. If, for instance, the system is completely idle, the clock interrupt takes about 5 percent of the available real-time looking for work to do. This is indicated in the Profile histogram in the '_switch' routine count. The '_switch' routine scans through the process table looking for work to schedule. Some of the 5 percent real-time spent looking for work to do could be regained had there been work to do and some of the 5 percent could be considered idle time.

The Profile 'trap' and 'call' counts show as expected that there are many system calls. The 'rsave' and 'rreturn' counts indicate lots of subroutine calls and subroutine call overhead. The counts '_nami', '_readi', '_writei' and the counts from '_bread' to '_clrbuf' show a heavy use of path names possibly indicating an overuse of pathnames. The '_ttyoutp' count indicates time spent transmitting to online and offline TTYs. The overhead of transmitting to TTYs removed from service (off-line) should be eliminated. This can be done by modifying TTY software to ignore TTYs removed from service. A lot of disk I/O is indicated as expected. The '_rprintr' count indicates that most RP interrupts occur while in the UNIX scheduler (SYSTEM Column). The '_sched' count, however, is low indicating that when the system is in idle it is usually waiting for disk I/O. This shows that a faster disk would help system response.

The 'notused' counts indicate addresses out of the Profile range were interrupted. The out of range addresses include interrupt vectors and special UNIX scheduling in user space.

The Profile results indicate that a faster disk is needed, that less use of pathnames is needed and that the overhead of transmitting to TTYs removed from service should be eliminated. Unfortunately, due to the inaccuracy encountered

by driving Profile off the clock interrupt, very little can be learned about UNIX overhead. The measurement of UNIX planned by Hal Pierson and Bill Snider of Department 5222 will enable more accurate measurements of UNIX.

CRAFT INTERACTION ON THE ATA SYSTEM

An important function of ATA is to allow craft interaction with the system via TTYs at both remote and central sites. Craft personnel can use TTY I/O to obtain more exception report data, to update an exception report and to monitor and change various status information. Craft interaction is difficult to simulate and measure since it can vary tremendously. An attempt was made to measure the effect of craft interaction on the system with a load of 700 cards per hour. To simulate craft interaction the OP:ER:ALL message was inputed to the system every 5 minutes during a measurement run. The RP disk usage increased 7 percent to 45 percent. CPU idle was 16.6 percent; a decrease of 8 percent. The USER mode measurement was 7 percent higher at 23 percent. The ER delays were 20 minutes maximum; an increase of 4 minutes, and 6 minutes average; an increase of 1 minute. Craft interaction was also measured for three typical messages entered on an idle system (no ATA running) with the results shown in Table 3.

The OP:ER message is used by craft when additional information on an Exception Report is needed. The load placed on the system if this message is entered once every 5 minutes is shown. The load is large but the message should be infrequently used. The UPD:ER message is used by craft to update an Exception Report. This message will be entered frequently by the craft and the load on the system, especially on the RP disk, is significant. The VER:ER message is used to obtain additional information on an ER. The load placed on the system with this message is high, but it also should be infrequently used.

The measurements of craft interaction show that the load they place on the ATA system is large enough to seriously affect system capacity. Unnecessary craft interaction during peak loads should be limited to prevent possible overloading of the system.

ACKNOWLEDGEMENTS

D. Eitelbach and F. Six of my department and W. Snider and H. Pierson of Department 5222 have assisted in the development of the real-time measurement techniques and in the analysis of the measurement results.

J. L. Barberio

HO-5161-JLB-vaw

Atts.
Appendix 1
Tables 1 through 3
Figures 1 through 8

APPENDIX 1

Original ATA Estimates vs. Present Real-Time Measurements

Original estimates made before the ATA development was begun indicated that more than 30 sites could be handled with the 11/40 processor, RP and RF disk hardware configuration. Presently the capacity of the prestandard system with 80K of core memory is about seven sites. A general discussion about the estimates can more reasonably explain the discrepancies than a lengthly analytical analysis.

The original estimates consider only disk swapping and access. No estimates of CPU real-time were made. A comparison of original estimated program sizes and present sizes are given in Table 3.

The present ATA software size is almost three times the original estimate. Swapping had to be moved from the RF to the RP disk because of the growth in program size. The Logger is the only program which is presently at its estimated size. The ERG, OD, and MEAS software were not considered originally. The number of RP disk accesses per card, excluding swapping, was estimated at 12 maximum assuming a bin threshold of one. Today the number is 27. The addition of the ERG, OD, and MEAS software accounts for the present number of RP disk accesses.

The original estimates were made in April, 1973, which was very early in the ATA prestandard development. Many additional features were added to the system along the way making the original estimates obsolete.

TABLE 1

ATA PROGRAM SIZES

	ORIGINALLY ESTIMATED	PRESENT
Logger	2K words	1.9K words
AI	6K words	21.6K words
AII	2K words	23.9K words
ERG	0K words	23.7K words
OD	0K words	2.8K words
UNIX	23.3K words	23.3K words
TOTAL	33.3K words	97.2K words

TABLE 2

PROFILE SAMPLING

UNIX SYSTEM ROUTINES	USER PEG COUNT	SYSTEM PEG COUNT
trap /	559	0
call	1223	845
_display	0	0
_getc	370	182
gethalt	413	191
_putc	598	111
puthalt	512	122
_backup	0	0
_fubyte	417	0
_subyte	338	0
_fuword	1207	0
_suword	3255	0
_copyin	561	0
_copyout	1949	0
_idle	0	76904
_savu	818	30
_retu	518	545
_spl0	813	1297
_spl4	0	0
_spl5	546	431
_spl6	99	13
_spl7	0	0
_copyseg	759	0
_clearse	159	0
_dpadd	225	0
_dpcmp	177	0
start	0	0
_ldiv	82	0
_lrem	80	0
_lshift	141	0
rsave	4315	697
rretrn	4457	544
_swtch	154	0
_main	0	0
_sureg	4540	311
_estabur	6	0
_nseg	0	0
_iinit	0	0
_alloc	5	0
_free	1	0
_ialloc	0	0
_ifree	0	0
_getfs	4	0
_update	10	0
_iget	959	0
_iput	31	0
_iupdat	103	0
_itrunc	6	0

TABLE 2 (Con't)

_klwrite	0	0
_klxint	0	1
_klrint	0	0
_klsgtty	0	0
_logopen	0	0
_logclos	0	0
_logread	8	0
_passrec	162	0
_logsatt	0	0
_minput	113	155
_xbinput	210	293
_xbsgtty	0	0
_xbread	0	0
_xbwrite	0	0
_lwrite	0	0
_ttstrt5	0	0
_essinpu	0	0
_esssttty	0	0
_esswrit	0	0
_gtty	0	0
_stty	0	0
_sgtty	0	0
_wflusht	2	0
_cinit	0	0
_flushtt	1	0
_canon	7	0
_ttyinpu	8	11
_ttyoutp	1058	10
_cdelay	11	0
_ttrstrt	0	0
_ttstart	62	81
_ttread	1	0
_ttwrite	264	0
_lpopenl	0	0
_lpopen	0	0
_lpclose	0	0
_lpwrite	33	0
_lpcanon	120	0
_lpstart	63	20
_lpint	4	1
_lpoutpu	93	0
_tmopen	0	0
_tmclose	0	0
_tcomman	0	0
_tmstrat	0	0
_tmstart	0	0
_tmintr	0	0
_tmread	0	0
_tmwrite	0	0
_tmphys	0	0
_rpstrat	483	39
_rpstart	51	130
_rpintr	59	120
_rpread	0	0
_rpwrite	0	0
_rpphys	1	0

TABLE 2 (Con't)

_maknode	0	0
_wdir	1	0
_printf	0	0
_printn	0	0
_putchar	0	0
_iopanic	0	0
_panic	0	0
_readi	933	0
_writei	215	0
_max	5	0
_min	210	0
_iomove	551	0
_sleep	42	3
_wakeup	2026	2176
_sched	0	51
_swtch	558	25379
_newproc	26	0
_expand	20	0
_bmap	417	0
_passc	583	0
_cpass	786	0
_nodev	0	0
_nulldev	0	0
_bcopy	520	0
_xswap	2	44
_xfree	0	0
_xalloc	1	0
_xccdec	0	11
_trap	3272	0
_trapi	159	0
_nosys	0	0
_nullsys	0	0
_signal	0	0
_psignal	0	0
_issig	315	0
_psig	0	0
_core	0	0
_exec	19	0
_rexit	1	0
_exit	25	0
_wait	64	0
_fork	4	0
_sbreak	1	0
_read	62	0
_write	17	0
_rdwr	591	0
_open	6	0
_creat	0	0
_openi	17	0
_close	3	0
_seek	258	0
_link	0	0
_mknod	0	0
_sslep	1	0
_getswit	0	0
_gtime	17	0

TABLE 2 (Con't)

_stime	0	0
_setuid	0	0
_getuid	0	0
_setgid	0	0
_getgid	0	0
_sync	0	0
_nice	0	0
_unlink	0	0
_chdir	0	0
_chmod	0	0
_chown	0	0
_fstat	0	0
_stat	0	0
_stat1	28	0
_dup	0	0
_smount	0	0
_sumount	0	0
_getmdev	0	0
_ssig	3	0
_kill	0	0
_times	5	0
_namei	658	0
_schar	0	0
_uchar	62	0
_getf	241	0
_closef	12	0
_closei	7	0
_openi	15	0
_access	29	0
_owner	0	0
_suser	0	0
_ufalloc	26	0
_falloc	168	0
_event	80	0
_clock	0	0
_timeout	5	6
_ms_prfl	0	0
_dcopen	0	0
_dcclose	0	0
_dcread	2	0
_dcwrite	3	0
_dcxint	42	80
_dcrint	2	7
_dcsgtty	1	0
_djenabl	2	0
_djopen	4	0
_djclose	0	0
_djread	0	0
_djwrite	4	0
_djrint	289	392
_djsgtty	0	0
_djxint	453	564
_djstart	25	0
_klopen	0	0
_klclose	0	0
_klread	0	0

TABLE 2 (Con't)

_bread	326	0
_bwrite	38	0
_bdwrite	31	0
_bawrite	2	0
_brelse	688	23
_getblk	2055	0
_iowait	47	0
_notavai	505	0
_iodone	33	70
_clrbuf	70	0
_binit	0	0
_devstar	70	136
_swap	0	24
_bflush	0	0
_physio	1	0
_geterro	28	0
_mmread	9	0
_mmwrite	4	0
_malloc	8	106
_mfree	15	111
_pipe	0	0
_readp	3	0
_writep	5	0
_plock	0	0
_prele	36	0
_rfstrat	0	0
_rfstart	0	0
_rfintr	0	0
notused	0	0
notused	0	0
notused	0	0
>notused	69	143
notused	0	0

TABLE 3

ATA INDIVIDUAL CRAFT I/O MESSAGE USAGE*

OP:ER:ALL (entered once in a 5 minute measurement interval)

RP = 8%

RF = 4%

CPU Usage = 9%

USER Mode = 6%

UPD:ER (entered once in a 1 minute measurement interval)

RP = 11%

RF = 2.5%

CPU USAGE = 2%

USER Mode = 2.7%

VER:ER (entered once in a 1 minute measurement interval)

RP = 12.6%

RF = 15%

CP U Usage = 15%

USER Mode = 5%

*NOTE: These results are the real-time used by the indicated message. The idle system usages have been subtracted out.