LBL-21440 c.2

# Lawrence Berkeley Laboratory
UNIVERSITY OF CALIFORNIA

RECEIVED
LAWRENCE
BERKELEY LABORATORY

JUN 30 1986

LIBRARY AND
DOCUMENTS SECTION

## Computing Division

Presented at the USENIX Association Conference,
Atlanta, GA, June 9-13, 1986

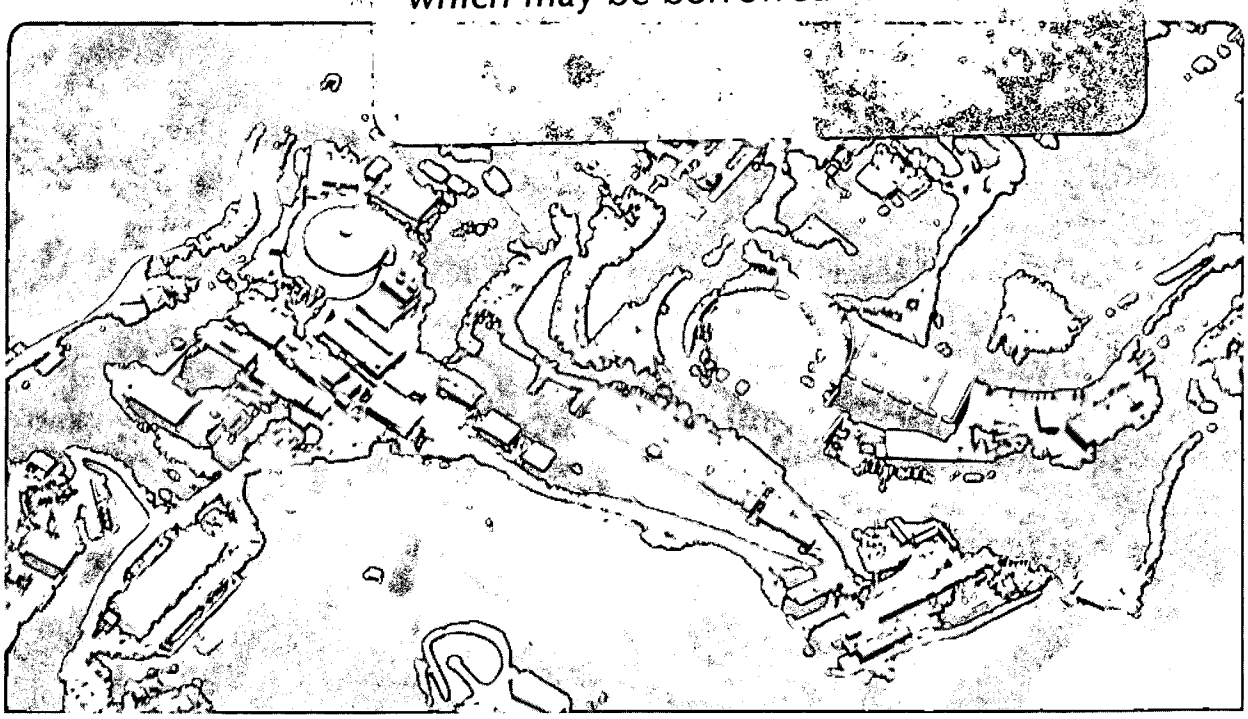
# UNIX BASED DISTRIBUTED PRINTING IN A DIVERSE ENVIRONMENT

W.E. Johnson and D.E. Hall

June 1986

TWO-WEEK LOAN COPY

This is a Library Circulating Copy
which may be borrowed for two weeks.



Prepared for the U.S. Department of Energy under Contract DE-AC03-76SF00098

LBL-21440 c.2

## DISCLAIMER

This document was prepared as an account of work sponsored by the United States Government. While this document is believed to contain correct information, neither the United States Government nor any agency thereof, nor the Regents of the University of California, nor any of their employees, makes any warranty, express or implied, or assumes any legal responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by its trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof, or the Regents of the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof or the Regents of the University of California.

LBL-21440

# UNIX Based Distributed Printing in a Diverse Environment

William E. Johnston

Dennis E. Hall

Advanced Development Projects
University of California
Lawrence Berkeley Laboratory
Berkeley, California 94720

June, 1986

This research was supported by the Applied Mathematics Sciences Research Program of the Office of Energy Research, U.S. Department of Energy under contract DE-AC03-76SF00098.

# UNIX Based Distributed Printing in a Diverse Environment

*William E. Johnston*
*Dennis E. Hall*

Advanced Development Projects
Lawrence Berkeley Laboratory
University of California
Berkeley, California, 94720
wejohnston@lbl.arpa, ...ucbvax!lbl-csam!johnston
dehall@lbl.arpa, ...ucbvax!lbl-csam!hall

*ABSTRACT*

This paper presents our experiences using the Berkeley UNIX* line printer spooler mechanism (*lpd*, et al) to provide distributed, mostly laser printer based, typesetting and graphics output to a geographically dispersed, heterogeneous, set of host computers and users. The user interface is the usual set of UNIX commands, though the methodology employed is somewhat different from the usual. The user's environment is resolved on the local machine and the tasks of document formatting and device driving are relegated to dedicated server systems in order to remove these compute intensive tasks from the timesharing client, or user, systems. The details of the system are described, together with an analysis of performance issues, the suitability of *lpd*, and operational aspects.

## 1. Introduction

The Advanced Development Projects group evaluates, tests, and installs new computing technologies for Lawrence Berkeley Laboratory's Computing Division. Our primary function is to bring promising new computing technology into the scientific computing environment. This article describes one such project, a distributed printing facility.

Lawrence Berkeley Laboratory (LBL) is a multi-purpose research facility with programs in physics, astrophysics, nuclear chemistry, materials science, biophysics, research medicine, electron microscopy, mathematics, computer science, earth sciences, and renewable resources. These programs are supported by computing facilities consisting of a central VMS cluster of five VAX-8600's and several 780's, four UNIX systems, and a significant part of a remote Cray XMP. In addition to these "central" facilities, various other departments at LBL operate another twenty to thirty VAX's, an ELXSI, at least one Pyramid, and an indeterminate number of workstations. Most of these systems (except the Cray, which is a remote facility) are directly connected to a site wide

---

* "UNIX" refers to 4.2bsd UNIX, unless otherwise noted.

The work presented in this paper is supported by the U.S. Department of Energy under contract DE-AC03-76SF00098. Any conclusions or opinions, or implied approval or disapproval of a company or product name are solely those of the authors and not necessarily those of The Regents of the University of California, the Lawrence Berkeley Laboratory, or the U.S. Department of Energy.

Ethernet, or have some other direct access to the LBL internet.

One thing that most LBL computer users need is document formatting and hard-copy graphics output. These needs have been met, in part, by establishing a distributed printing system that handles, in various degrees, ASCII, TEX, *troff*, Tektronix and UNIX *plot* format files. These various formats enjoy differing degrees of sophistication in their handling by the distributed printing system, and in the available output devices.

By way of review, recall that the batch mode text formatting typical of *troff* and TEX may be represented logically as:

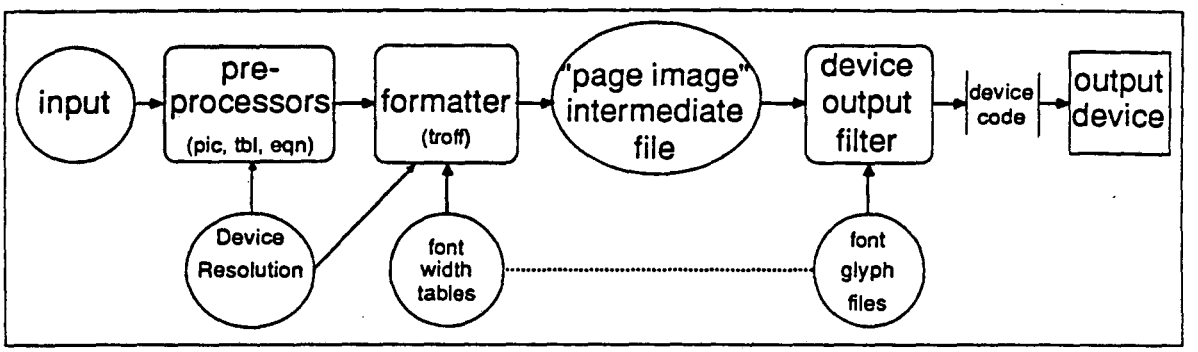



Figure 1

Most of the software development effort has gone into the remote *troff* service, whereby all user systems can spool *troff* input files to the back end servers. The *troff* input files have local file environment dependencies removed on the user system. The server systems are (typically) Integrated Solutions, 68020 based, UNIX systems dedicated to printing. The server systems execute *troff*, and its preprocessors (*eqn*, *tbl*, etc.), and any filters necessary to generate output device specific code. The server sends that code to the requested device and records accounting information. By far the most commonly used output devices are Imagen laser printers.

The universal availability of the distributed printing system within LBL depends on each user system being able to send input files to a server system, where it is received by *lpd*. In general this is done with an implementation of *lpr/lpd* on the user system, though variations have been tried. Access to output devices, which are located throughout LBL, is via Ethernet, long haul serial line, or some combination of these together with an AppleTalk network. (The site is a little more than one square mile of rough hillside which inhibits the use of local microware systems.)

*The system is successful, and has been in operation for about 18 months. There are six public Imagen 8/300 printers; three of these consistently output 30,000 pages per month each, two output 20,000 pages per month each, and one outputs 10,000-15,000 pages per month. Three more public printers have been recently installed. Additionally, there are about ten private printers that output approximately 25% the amount of the*

---

The following trademarks are acknowledged (apologies to anyone I missed): AT&T, Bell Laboratories: *UNIX*; Adobe Systems; *Transcript, Postscript*; American Mathematical Society: *TEX*; Apple Computer: *AppleTalk, LaserWriter, MacIntosh*; Cray Research: *Cray XMP*; Digital Equipment Corporation: *DEC, VAX, VMS, DECNET, Q-bus*; ELXSI; ISSCO: *TELL-A-GRAF, DISSPLA*; Imagen; *8/300, Inovator, imPRESS*; Integrated Solutions, Inc.; Motorola: *68010, 68020*; Mergenthaler-Linotype: *Linotronic-101*; Pyramid; Quality Micro Systems: *QMS, QUIC*; Sequent Computer Systems: *Balance 8000*; SUN Microsystems: *SUN*; Talaris Systems; Tektronix: *PLOT-10, TCS*; Versatec: *V-80*; Wollongong: *Eunice*; Xerox: *Ethernet, XNS*.

*public printers.*

## 2. The Printing System

### 2.1. Goals

The primary goal of the distributed printing system is to accept all common types of input files, and to print these files on any of the available output devices. It is recognized that a full connection of N_formats × M_output_devices is a goal that has to be tempered by the reality of available input and output filters, and the available software effort to generate new ones. We have been successful by applying the "90%" principle to determine what software should be written. That is, solving the problems so that 90% of the users are satisfied.

A second goal is to introduce individual users, departments, and the central facility to the use of modern, "high resolution", laser driven, Xerographic, hardcopy output devices. In this second goal we have been amazingly successful. While everybody will make use of central facility provided, "free" hardware, adding hardware to user owned computers requires spending money that might otherwise be used for research instrumentation. This is done with great reluctance. There are now more than twenty laser printers at LBL, two-thirds of which were purchased outside the central computing facility. Of those two-thirds, most are used with the distributed printing system.

A third goal is to make the system as operator-less as possible, a point to be discussed later.

### 2.2. The Mechanism for Remote *troff*

Of the several models of processing used by the various types of input files, the off loading of *troff* input to a back end server system, and output to Imagen printers, constitutes the major use of the distributed printing system. The mechanism for *troff* is described here. The mechanism for other input file types is described briefly in the section below on input types. In what follows, program and file names are typeset in italic in the text, and/or bold in the standouts. More detailed information may be found in reference [1].

The essential idea of the distributed printing system is to use *lpd*'s ability to route files to remote systems, and to use its flexible notion about output filters and the */etc/printcap* configuration mechanism, to send *troff* input files to a server system for processing and then to an output device.

The "user" or "client" system is the computer initiating a printing request. The "server" system is the computer that executes the programs needed to produce output. The user system may also be the server for a particular device, though usually not. There are normally many user system computers, and one server system for a given output device.

The distributed printing mechanism is just the sequence of processes through which data pass on their way from user systems to server systems. For remote *troff* targeted for an Imagen printer (*itroff*), the sequence is:

**itroff → lpr → (user spool directory) → lpd →**
**→ (network communication) → lpd →**
**→ (server machine spool directory) →lpd → itroffd →**
**→ icat.e → tbl | eqn | troff | catimp | ies**

A brief description of this sequence follows.

### 2.2.1. Itroff

*Itroff* is the user interface for *troff* typesetting on the Imagen printers. For example, invoking *itroff -Pip1 troff.file* causes *itroff* to accept the user (*troff*) input file, process it through *soelim* to eliminate environmental dependencies (e.g., user include files), and then pass the file to *lpr* with flags giving the requested output device (*ip1*), and input file type (*troff*).

### 2.2.2. *Lpr* and the Spool Directory

*Lpr* takes a file and device specification, sends the input data file (and an associated control file which *lpd* creates) to a spool directory. The control file contains a variety of information: what output filter should be used to process the file; the user name; the user computer system name; etc. *Lpr* uses the file */etc/printcap* to determine where the spool directory is located, which system is the server for the requested printer, and how the user file should be processed. For example

**lpr -t -Pip1 file.troff**

selects the following sets of lines from */etc/printcap* (abbreviated in this example):

On a server system:

```
ip1:
        :sd=/usr/spool/imagen/ip1:
        :tf=/usr/local/imagen/bin/itroffd.ip1:
```

Or, on a user system:

```
ip1:
        :rm=host2:
        :rp=ip1:
        :sd=/usr/spool/imagen/ip1:
        :mx#1000:
```

In the first case (on the server system) the spool (or queuing) directory is specified (*:sd*), together with the filter to be used to process the file (*:tf*, in the case of a *troff* file).

In the second case (on the user system) the specification indicates what remote system to send the file to (*:rm*), what device to use for output (*:rp*, which may, incidentally, be different from the user requested device), where to spool (enqueue) the file (*:sd*), and a limitation on the size of files so processed (*:mx*).

In both cases *lpr* builds the control file (with a name of the form *cf** in the spool directory) which it sends, along with the input file (now with a name of the form *df**), to the spool directory (queue) where *lpd* takes over.

### 2.2.3. *Lpd*

*Lpd* is the worker back end for *lpr*. *Lpd* is a "daemon" process that watches the queues (as defined by printcap) for files needing to be processed. When a job shows up (a *cf** and *df** file combination) *lpd* looks at the *cf** file, which contains a pointer to an */etc/printcap* entry. The *printcap* entry specifies an output filter to be invoked on the data (*df**) file. For the Imagens, the output filter is a shell script called *itroffd.ipX* ("X" indicating the specific Imagen).

For a user system, *lpr* places the *cf** and *df** files in the local spool directory. The files stay in the local spool directory only long enough for *lpd* to forward them to the server system. Unless queuing is disabled for the requested device on the server, the files are normally forwarded immediately. For a server system, the files remain in the spool

directory until output processing is complete.

### 2.2.4. Itroffd.ipX

*Itroffd.ipX* is the *lpd* output filter for the Imagens. *Itroffd* sets up the sequence where the work is done :

**tbl | eqn | troff | catimp | ies**

The first three are familiar. *Tbl* and *eqn* were included in the output processing mostly to minimize the support needed on the user systems. Both of these preprocessors need to know various things about the output device.

*Catimp* converts the *troff* output to imPRESS (the language of the Imagen), and manages the loading of the character bit maps into the Imagen. *Ies* manages the network interface, transmitting data (imPRESS instructions) to the Imagen over a TCP circuit, and receiving status information back via UDP packets. For a serial line connected printer, *ies* is replaced by *ips*, which implements a sequenced packet protocol over a serial line.

## 2.3. Limitations of this Mechanism

This whole mechanism is to a certain extent a "90%" solution. Users who make use of preprocessors other than *eqn* and *tbl* must have a somewhat better than average understanding of how the system works. (The average required understanding is, by design, near zero.) For example someone with the following in a *troff* input file could probably also be assumed to understand the sequence of operations done by the distributed printing system *troff* scripts:

```
.\"  this has to be run through soelim before refer
.
.
.so /u0/csam/johnston/util/troff.macros
'so /u3/graphics/biblio/refer.me        \" prevent soelim from expanding this
.
.
```

The same applies to the use of *pic* and *ideal*.

The mechanism for handling TEX input files will be somewhat different than the way the distributed printing system deals with *troff* files. The reasons are that 1) TEX produces more meaningful error messages than does *troff*, and 2) TEX provides an indirect referencing scheme that may require two or three passes over the input file in order to resolve the references. Neither of these are impossible to deal with in a batch environment. *Lpd* will mail messages back to the user, and *errorout* message analysis on the server could result in automatically invoking TEX multiple times. We have not done this yet, but probably will as TEX use increases owing to people from the VMS environment becoming familiar with the UNIX aspects of the distributed printing system.

## 2.4. Input to the Distributed Printing System

There are several file formats the printing system must be able to process. These formats are *troff* and *ditroff* input files, TEX *dvi*, Tektronix, UNIX *plot*, ASCII and device specific code.

### 2.4.1. *troff*

The use of *troff* is well established, but not universal at LBL. The four UNIX systems operated by the central facility are used primarily for *troff* based text processing,

the Technical Information Department uses *troff* to drive it's phototypesetter, and the Computer Science Research Department text processing is mostly *troff* based.

*Ditroff* is considered separate from *troff* because of the unfortunate circumstance of having a different set of available fonts. Most of the laser printer output is to Imagen printers, and the Imagen support for *ditroff* supplies a less sophisticated set of fonts than for *troff*. *Ditroff*, however, provides the features of landscape mode formatting, and a mono-spaced font. These are the main reasons for its preference over *troff*. The lack of some fonts, and diversity of other fonts create the one of the biggest nuisance factors in trying to provide reasonable typesetting services. There are now professionally designed fonts becoming available. These will help, but as yet they lack the diversity to replace the existing ad hoc fonts.

### 2.4.2. TEX

The TEX user community at LBL is smaller than the *troff* community, and is primarily VMS based. At the moment, the distributed printing system deals only with TEX *dvi* (device "independent" intermediate) file. The main issue in supporting *dvi* output is one of accumulating the fonts on the server that are supported by the various flavors of TEX on the user systems. The only problem in serving the TEX community is the unfortunate circumstance that TEX output device drivers can put in page margins unknown to the TEX formatter. This is done in some TEX environments and not others, so, naturally, both cases show up on our distributed printing system.

### 2.4.3. Tektronix

For historical reasons, the Tektronix 401x, Plot-10/TCS graphics format, like the Calcomp subroutine library interface, is ubiquitous and must be supported. (Parenthetically, the Tektronix format is not a bad choice for laser printers, being compact and providing most of the required functionality for graphics, except for a line width attribute.) The support for Tektronix files takes two forms. One is a filter that converts Tektronix code to UNIX *plot* format and then to device code. The second method is that most laser printers (including Imagens) do various degrees of emulation for Tektronix code. The Imagen emulator is reliable, but originally emulated a Tektronix 4010, a relatively low resolution device (750 $\times$ 1000). A 4014 emulator, which we have not tried, has recently become available. This new emulator emulates the 4014 w/enhanced graphics option (3000 $\times$ 4000), which is a much better resolution match with a 300 dot per inch (dpi) laser printer.

At LBL the largest source of Tektronix code is from the Cray XMP, where it is produced as a graphics metafile by most of the graphics subroutine libraries on the Cray. These files are sent to the central facility by an FTP-like program, and printed using either the distributed printing system or the central facility, Talaris-2400 laser printers.

### 2.4.4. Others

UNIX *plot* code is dealt with by a *plot*-to-imPRESS filter, and then handed off to the distributed printing system.

Output device specific code (e.g. imPRESS for the Imagen, QUIC for the Talaris, and Postscript for the LaserWriter) is typically generated by graphics packages. For example, the largest sources of imPRESS code at LBL (ignoring the text formatters) are ISSCO's TELL-A-GRAF, and Tektronix's GKS systems. Like most such commercial systems, they have device drivers for each specific output device. The result is that a moderate amount of device code is dealt with by the distributed printing system.

There are some ASCII files which are printed in line printer mode.

## 2.5. Output Devices

This section provides a description of output devices that participate in typesetting and graphics output.

The acquisition of hardware is sometimes based on rational decisions, and sometimes not. Especially with new types of hardware the wisdom or foolishness of a particular decision may not be realized until long after the fact. When we specified the first devices that would support the distributed printing system, we had three hard requirements. The device should have:

1) A minimum resolution of 300 dpi on the paper;

2) Memory to do full page bit maps, and;

3) An interface to Ethernet, using TCP/IP.

Subsidiary constraints were that the marking engine quality should be commensurate with the resolution, and that the print speed should be of approximately 10 pages per minute. (This last requirement is based on wanting a reasonable replacement for a Versatec, V-80 doing 5,000 pages a month.) It is worth commenting here that we were completely aware of differing needs for output "resolution". Our goal was to procure a relatively high resolution output device compared to those in use by computer users at the time. It was not our goal to convince professional graphics artists and typesetters that they should abandon their old 2,000+ dpi devices and use our new "high" resolution devices.

### 2.5.1. The Imagen 8/300 Printers

The requirements mentioned above resulted in an initial acquisition of one Imagen 8/300 printer with an Ethernet interface, followed six months later by two more. These devices are 300 dpi, eight page/minute, Xerographic laser printers. Conceptually, the architecture of the printer controller consists of a communication handler, a command processor, and five translators or emulators. The input file format consists of a control line giving various state information and the language of the following data. The printer gets the command line first, sets its state and invokes the appropriate translator for the remainder of the file. The translator converts the data file from one of several formats (imPRESS, line printer, Tektronix, Diablo 630 or raster image) to a page image bitmap. The bitmap scan line modulates a laser to produce a latent image on an electrostatically charged, selenium drum. This latent image is "developed" by applying a dry powder toner (usually black) to the drum. The toner adheres to the electrostatically represented latent image, and is subsequently transferred to plain paper where it is heat fused in place to produce the final output. The software architecture mentioned above is implemented on a Motorola 68000, disk based operating system using about 1.8Mbytes of memory, of which 1.1 MBytes are used for the bitmap. The processing in the controller can exhibit a fair amount of parallelism in its functioning (communication, translation and output). While this description is nominally of an Imagen printer, it applies to most of the currently available laser printers used for computer output.

The Imagen printers are the mainstay of the distributed printing system. There are about twenty printers scattered about the site, and the six original public printers each turn out 15,000 to 30,000 pages per month, or about 10 times the suggested duty cycle. Although the original requirement was for Ethernet connected printers, there are now about an equal number of serial line connected printers. The serial line printers are substantially less expensive than the Ethernet printers, are almost as fast (see "Operational

Issues", below), and can be located in places where the Ethernet does not reach.

### 2.5.2. The Talaris 2400 Printers and the Issue of Duty Cycle

A second type of output device was acquired primarily to support lineprinter and graphics output from the central VMS cluster, though these printers are also used by the distributed printing system. Based on our experience with the small laser printers it was decided to use similar devices on the cluster machines instead of impact printers. The requirement was to support 250,000 pages a month of lineprinter output and an additional 5,000-10,000 pages a month of graphics output, which can be done by a 75 page/minute output device running during prime time. In what may be shown to be folly instead of wisdom, we required that the needed 75 pages/minute should be provided by at least two printers for the sake of redundancy, and at most three printers since VMS could not feed multiple printers from one queue (at that time). The result of this was to purchase three 24 page/minute, Talaris printers.

The Talaris 2400 printers are based on a Xerox, XP-24 print engine. This print engine is rated at about 30,000 to 50,000 pages a month, though this is not immediately obvious until you look at the per page maintenance charges. What may prove to be our undoing on these printers is that the aggregate duty cycle for the three 24 page/minute printers is much less than for a single 75 page/minute printer like the Xerox 8700 (which has a duty cycle in excess of 500,000 pages/month). Not only is the aggregate duty cycle too low, but when one printer is down, the other two pick up the load, thereby further exceeding their duty cycle. While these XP-24 based printers (this same print engine is used by Imagen for their 24/300 printer, and by Xerox in their 3700) have settled down, in the first several months each one was down about 50% of the time owing to mechanical problems. Things got so bad at one point that the service person (who was coming in several times a week) refused to service the printers because we had put 30,000 pages through one printer in less than a week. Things have settled down, and the printers have been working well for six or eight months now. We are, however, paying substantially more than anticipated for the maintenance required to support the high duty cycle.

Beyond the print engine problems, the controllers and supporting software of Talaris does a reasonable job, and do not exhibit any problems out of the ordinary. The principle complaint with these controllers is the QMS QUIC language code. (Talaris is a software house that OEMs the QMS systems.) QUIC is the native language of the QMS controller. The problem with QUIC is that it uses a "human readable", ASCII format, and is therefore not compact. (PostScript, for all its good features is even worse in this regard.) By way of comparison, Tektronix format files typically undergo a five times expansion even when using a clever conversion algorithm to QUIC code. This expansion is a significant factor when the graphics print job has 200 complex frames that occupy 10-15 MBytes in Tektronix format, a not uncommon situation in super-computer output. Fortunately Talaris now provides a Tektronix 4014 emulator for it's controller, and this file expansion is unnecessary.

These printers primarily serve the VMS systems. They coexist with the UNIX/Eunice print spoolers by having the final *lpd* output filter place the file into the VMS print queue, where it is printed by VMS like any other job.

### 2.5.3. Apple LaserWriters and PostScript

The time of PostScript is comming. Religious arguments aside, the main advantages of PostScript are twofold. First, PostScript provides a measure of typesetting device independence that we have not come close to before. Second, PostScript,together

with more and more powerful personal workstations, is producting a boom in affordable page composition systems (e.g. MacIntosh and IBM PC). Already people at LBL format special documents (e.g., brochures and signs) that would be difficult, if not impossible for even an experienced *troff* user. They debug these using a LaserWriter, and get final copy by taking their MacIntosh diskette to the local commercial copy shop to use the 2500 dpi Linotronic-101 phototypesetter, which is a PostScript printer connected to a MacIntosh, like the LaserWriter.

In the more traditional context of the system described in this paper, PostScript printers offer a substantial advantage in price (partly owing to discounts given the University of California) and device independence. A modern PostScript phototypesetter could be connected to the distributed printing system *just like any other user printer*, eliminating the pain and delays of having to go through an in-house (traditional) printing department.

Apple LaserWriters and local AppleTalk networks exist in plenty, and this community wants to include their printer in the distributed printing system without impairing its use by the local MacIntoshs. We have taken two approaches to this. The people who gave us PostScript (Adobe Systems) have also given us TranScript software that outputs *troff* to PostScript printers. (TEX output filters are also available.) The connection to the distributed printing system is made by sending the PostScript *troff* file to a MacIntosh over a serial line. The MacIntosh is connected to an AppleTalk network and thus to a LaserWriter. There is a spooling program in the MacIntosh (which is dedicated to this function) which sends the PostScript code from the distributed printing system to the LaserWriter. Another approach to this connection is being done by developing an AppleTalk network to Ethernet gateway based on a Sun with an AppleTalk network interface board.

### 2.5.4. Electrostatic Printer/Plotters

The venerable Versatec, V-80's (200 dpi, wet toner, electrostatic printer-plotters) remain attached to several systems. These fan-fold paper devices are useful as a line-printer (the laser printers do line printing no faster than they typeset) when the output listing may be 100+ pages and you do not want to incur the wrath of other printer users. The same is true for long *vgrind* output. The extensive use of *troff* on Versatec output devices is one thing that prompted the development of the distributed printing system. We found that 20-30% of our VAX 780, UNIX system cpu time was used for *troff* and it's output filters for a Versatec like device. (This class of devices requires the rasterizing be done in the host system to supply the printers with bitmap images.) We immediately gained back this 20-30% of our user systems when the distributed printing system came on line, since most of the formatter processing was moved to a dedicated back end system.

## 2.6. Systems

This section describes the salient features of each of the "directly connected" systems, and comments on some aspects of integrating the Cray XMP.

The distributed printing system consists of server systems that drive the output devices, and client systems where input files originate. Directly connected client systems are those that have Internet access (TCP/IP on LBL Ethernet or ARPANET), and that run an *lpd* like file sender. Indirectly connected client systems handle input to the distributed printing system via file upload to a directly connected system. Output to indirectly connected server systems may be somewhat more automated. One case is described below.

### 2.6.1. The Server Systems

The client/server distinction among computer systems is not rigid, and those systems with unique hardware interfaces act as servers for their hardware, though in general the goal has been to affect a separation. The primary server systems are dedicated back end systems that normally do not have any users except for an occasional operator.

The servers for the distributed printing system consist of three Integrated Solutions, 4.2bsd UNIX systems. There are two 68020, VME bus systems supporting most of the printers, and one 68010, Q-bus system used mostly for development and debugging. The two 68020 systems are configured with 16 serial ports, 8 Mbytes of memory, an Ethernet interface and two 300 Mbyte disks. The serial ports support DMA output, making them suitable for attaching printers. The Q-bus system is similar but with less of everything and the addition of a 1/4" tape drive. The tape drive is used mainly for installing operating system updates. Most other tape access is via *rtar* or *rdump* using a tape drive on a VAX UNIX system.

The Integrated Solutions machines were chosen because of excellent cost/performance characteristics. These systems have proven reliable enough that we have not yet regretted the lack of a maintenance contract.

The two 68020 systems easily handle the nine public printers, and the dozen or so private printers. They will probably support almost that many again.

### 2.6.2. The Client Systems

The client operating systems include 4.2bsd UNIX, System V UNIX and VMS.

The 4.2bsd client systems work in the obvious way, via *lpr*. There are 8-10 such clients, including the central facility systems, several research systems, and several SUN workstations.

There are another 8-10 VMS clients that act as clients by virtue of running the Wollongong UNIX emulator.

The ELXSI system has a System V UNIX emulator, networking code from a third party, and a local version of *lpr* and *lpd*. This system can operate as a standard client, but most of its printing is done through a directly attached Imagen. The high speed of the ELXSI cpu significantly reduces *troff* turnaround, a definite advantage to users when the system has cpu to spare.

The Cray XMP is an indirectly connected client. There is an effort underway to establish an interprocess communication mechanism between the Cray and the central facility systems. Once this is done, it will not be too difficult to queue files from the Cray directly to the distributed printing system by developing an *lpd* look alike.

### 2.6.3. VMS Implementation Issues

Considerable effort has been put into the Wollongong UNIX emulator (Eunice) to support *lpr* and the DEC DEUNA Ethernet interface shared with DECNET. The result is that VMS systems now constitute a significant portion of the clients with much of their usage being TEX, graphics and some *troff*.

Eunice does not support all of 4.2bsd UNIX. *Lpr* was rewritten so not to use *select* (which did not exist when this was done), and not to use the AF_UNIX communication domain. The use of *syslog* was changed so not to require a daemon process. The use of *flock* was eliminated because it is redundant under VMS. To permit the use of VMS as a server system, an interface to the VMS print symbiont was implemented using the VMS SNDJBC function. Several new codes were added to */etc/printcap* to support the

VMS symbiont interface.

In general, UNIX style shell scripts do not do well under VMS. This is due in part to the higher overhead of starting up processes in VMS. The standard user interface functions of the distributed printing system that are implemented as scripts and invoke many processes, were rewritten to collapse the operation into one process.

There are some VMS systems that have only the UNIX networking code from Wollongong (not the full Eunice), and we have been experimenting with non-*lpd* originators. These systems have code that generates the control file, and makes *lpd* like connections to a server system.

## 2.7. Networks and Interconnections

Connections to output devices includes Ethernet, serial line, and parallel interface. The connections are determined by cost, location and network availability. The core of the distributed printing system is Ethernet based laser printers.

Internet connected client systems have access to the distributed printing system via TCP/IP on Ethernet, or ARPANET, or TCP/IP over point to point DECNET links using DBRIDGE. (DBRIDGE provides a mechanism for tunneling IP packets through a DECNET connection. Initially this was important, but is somewhat less so now that most systems have IP connections to the Ethernet. See reference [2].) Although not directly related, it is worth briefly describing the LBL Ethernet based internet, since this was the key to the rapid success and spread of the distributed printing system owing to it's relatively high bandwidth and wide availability throughout the site.

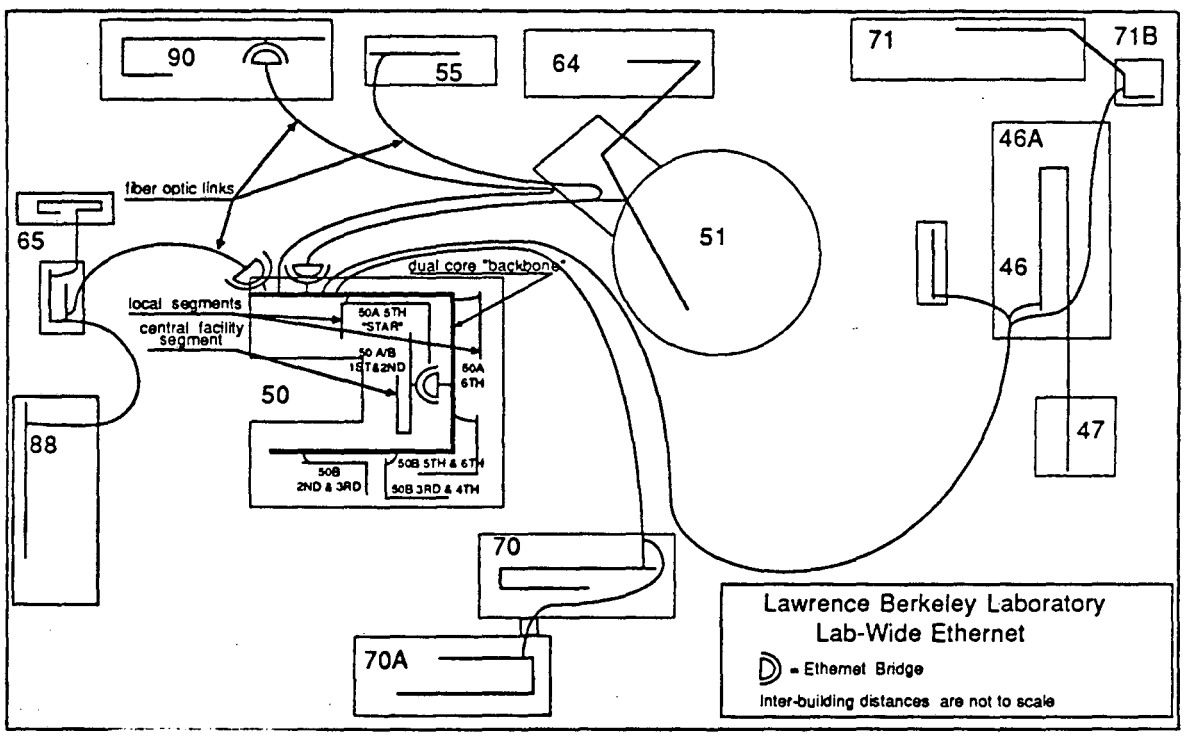



Figure 2

The LBL Ethernet consists of 2.5 Km of coaxial cable and multi-strand fiber optic links. The system is organized into core segments that are interconnected by DEC, LAN Bridge-100, protocol insensitive bridges. These bridges serve the important function of isolating the core segments by virtue of selective packet forwarding. The bridges

dynamically build destination tables indicating which packet destinations are on the "other side" of the bridge, and then only forward packets to the next core segment when the destination is located there. This is done at 15,000 packets/second, nearly the theoretical Ethernet bandwidth. These bridges are strategically located to isolate core segments with locally heavy traffic. They also serve as a "fire door" so that when one host or interface goes berserk it does not bring down the whole net.

The core segments (typically one per building) are connected to local segments (typically one per floor) via repeaters. The repeaters are used to electrically isolate the segments. They also reform the packets, but only at the signal level. As far as the Ethernet is concerned these are passive devices. See figure 2.

This large, contiguous Ethernet has several hundred attached systems running at least five different protocol suites: NSP (DECNET), TCP/IP, XNS, 3COM-XNS, and Intel, so far as we know (there could be others). Particularly as the local segments turn to "thin" Ethernet, the control of what gets attached becomes minimal.

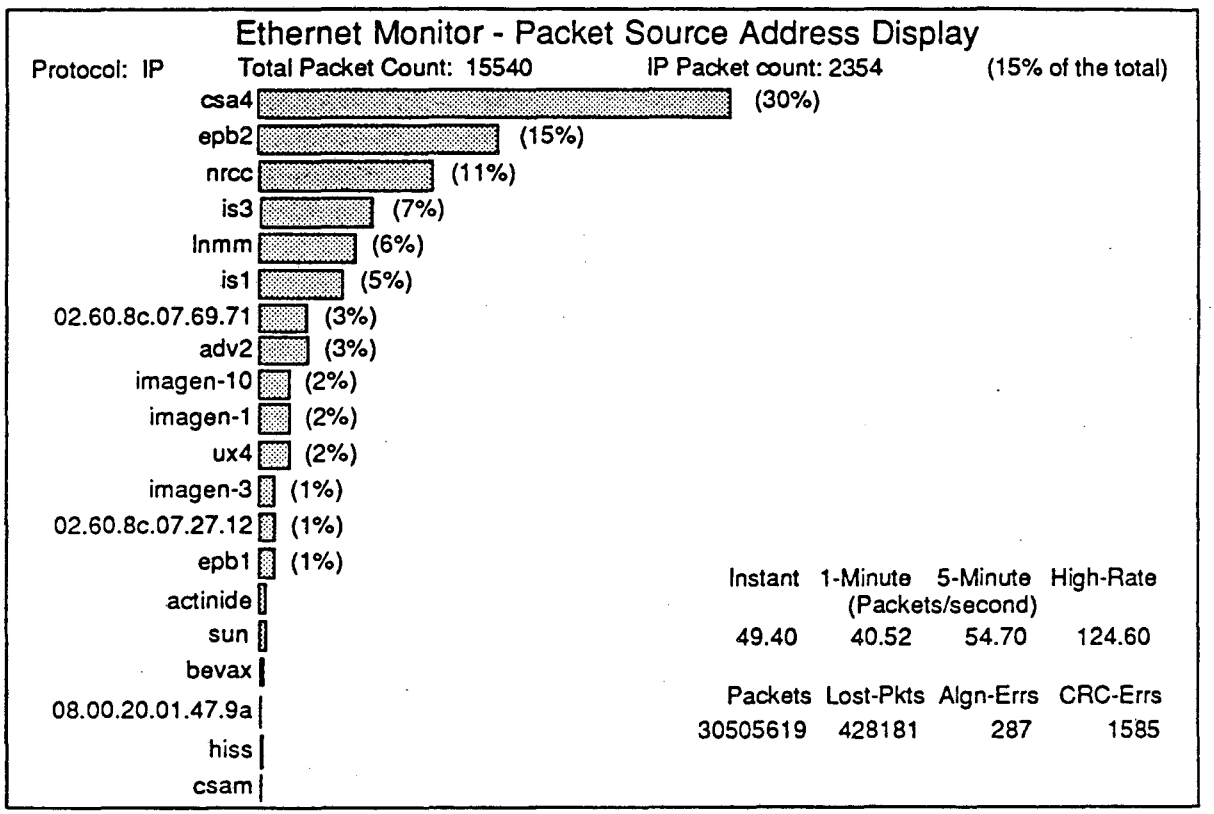



Figure 3

Figure 3 shows output from an Ethernet monitor connected to the central facility segment. The display shows sources of IP packets only, which were about 15% of the total at the time of the sample. Most of the rest of the traffic is from DECNET hosts. The IS1, IS3 and ADV2 hosts are distributed printing system server systems and Imagen-1, etc. are printers. As may be seen, printing accounts for about 20% of the IP traffic, or 3% of the total network traffic. Since the monitor is connected to the central facility segment which is connected by a bridge to the rest of the Ethernet, the graph shows only hosts on that segment, or hosts whose traffic destination is on that segment. The systems represented by their Ethernet address are of unknown identity.

## 3. Performance Issues

In acquiring the initial components of the distributed printing system we estimated that one 68010 system would support two or three Imagen printers running continuously. After assembling the first instantiation of the system we ran some cpu versus time-to-print-a-job benchmarks. We found that for a specific job running one, two and three simultaneous instances (job= *tbl* | *eqn* | *troff* | *catimp* | *ies*, as discused above) that the times to complete each job were about 1.0, 1.1, and 1.5 times one job, respectively. About 90% cpu utilization occurred with two jobs running. This benchmark has not been repeated for the currently used 68020 systems, but the one job case runs a little better than three times as fast. The other figures should scale similarly since I/O utilization is small compared to cpu, and there is adequate memory to run ten or twelve simultaneous jobs.

Other benchmarks show the overhead of unnecessarily running *eqn* and *tbl* on every *troff* job is less than 5%. We did find in early tests that we got a noticeable improvement in throughput by moving the font files off the system disk to somewhat balance disk access.

The disk space requirements of a print server are comparable to a multi-user system of the same cpu size. *Troff*, TEX, fonts and spooled files need about 100 MBytes on a four to six printer server system.

The speed of Ethernet printers and 9600 baud serial line printers is comparable, once a job starts printing. Ethernet printers are about 10-15% faster overall because of font cache reloading. The Imagen printers will cache fonts, but not a huge number. Since *troff*, *ditroff* and TEX all use different fonts, each of those jobs tend to invalidate the font cache. The reloading seems to be much faster over the Ethernet. The serial line printers now run at 19,200 baud, which somewhat narrows the difference.

Initially we had substantial concerns that running all the printers (four or five of which run almost continuously) on the Ethernet would degrade its preformance. Figure 3 in the network section shows that this has not been the case. The entire distributed printing system, including the user to server system file transfers and output to the Ethernet printers, is never more than a few percent of the total traffic. The total Ethernet traffic is almost never more than 10-15% of the 1,000 packet per second that we feel is the practical maximum for the Ethernet to function optimally. (We have no diskless workstations on the central Ethernet at this point.)

We have had some trouble with hardware related Ethernet problems, from both interface failures and incompatible Ethernet hardware. Certain combinations of interface boards, cable transceivers, and repeaters work well and certain ones don't work at all. Detecting this type of problem is difficult, and isolating the offending hardware is even harder. It requires perseverance from the technicians and systems programmers responsible for operations.

Since UNIX pipes are an expression of (course grained) parallelism, this application should do well on a multi-processor system like the Sequent, Balance 8000 (where we would love to try it). These machines schedule processes from a single run queue to the lowest priority processor, and handle interrupts in special hardware. The distributed printing system job (*tbl* | *eqn* | *troff* | *catimp* | *ies*) would naturally distribute itself over many processors.

## 4. Suitability of *lpd*

This section contains comments on changes that were made to *lpd*.

An accounting mechanism is now being added. Each control (*cf**) file must provide a valid account number that is recorded, together with page counts, by job. Following the philosophy that simple is best, charging is done only on a per printed page basis.

There were only two types of changes made to *lpd*. The first change was to increase the number of entries in the control file. These changes were made to accommodate accounting information and filter arguments, such as landscape mode flags and hard margin changes.

The second change to *lpd* was to prevent it from attempting to restart a job. The scripts that implement the distributed printing system transform the control file in a way that is not idempotent, and a second pass results in nonsense. Even without this circumstance, jobs of the type being discussed here almost never fail in a recoverable way and restarting just results in *lpd* looping. Many debugging facilities were also added to *lpd*.

## 5. Operational Aspects

Once the system is running, you have to find out if operators can run it, or, better yet, if it can run itself. The goal of this system is to have the printers be self service (user operated) up to, but not including, preventive maintenance, and not to require operators. Once the "sanity" checking was added (as described below) we approximated this.

A key to reducing operator participation (i.e. taking phone calls from irate users who know that the distributed printing system is running, but nothing has come out of the printer for an hour) was to add simple sanity checks to make sure that the *tbl* | *eqn* | *troff* | *output_filter* | *communication_filter* chain had not become wedged, and if it had, to notify someone automatically. At the moment this is done via the simple expedient of having a *crontab* triggered process check the printer status file to make sure that it changes periodically. If it does not, a message is sent to a central facility operator who can check the job manually. The rule of thumb that we use for *troff* jobs is that the cpu time (on a 68010 system) should be about one minute per five kilobytes of input file. Anything much beyond this probably means that *troff*, or it's preprocessors, is looping.

Initially there was a fair amount of educating the users. Users would look at the job queue (via *lpq*), observe that their job was active, and the printer was idle. What they failed to realize is that "active" means that *troff* is working on formatting the input, not that output is yet being sent to the printer. (The job pipe takes some time to fill up.) Also, users are now educated on how to add paper, change toner, clear jams, etc., thanks to the good design of the Canon print engine.

Operational tools have been provided to permit moving printers from one server to another, changing user systems to server systems, etc. These are described in reference [1].

The Canon print engine used in the Imagen 8/300 printer is rated at approximately 3,000 pages a month and a total lifetime of about 100,000 pages. From the beginning we have consistently put more than 5,000 pages a week through the public printers. The company that maintains the Imagens for us refused to maintain the print engines after discovering that the internal page counter on one printer was over 500,000. This turns out not to be a problem for two reasons. First, the Canon engines are reliable and a small amount of preventive maintenance done by the operators will keep them operating for months at a time. Secondly, as the printers age past their advertised lifetime, what usually goes wrong is that the lubrication on some of the internal working parts fails and they start squeaking and sometimes binding. Since we have more than fifty Canon print engines (LaserWriters, Laser Jets, Imagens, etc.) the local maintenance group had one

person trained in their maintenance. The squeaking and binding problem takes them a few hours to correct. The replacement Canon engines are about $1,000, but we have yet to replace one. As pointed out elsewhere, these conservative duty cycle ratings do not manifest themselves on faster printers, whose advertised duty cycle is much closer to reality.

In the issue of priority queues (or how do you keep the Director's secretary, who has just queued an important one page letter, from killing the graduate student who queued his 100 page thesis just before that), the solution has been mostly one of education. However we do impose software implemented page limits on some of the public printers. The best solution, though, is to buy more printers.

## 6. Acknowledgements

The authors were the primary architects and implementors of the distributed printing system and assisted in the transition from prototype to a production system. Maintenance, and many refinements, are carried out by Bob Rendler of the Computer Center. Marty Gelbaum and Wayne Graves were responsible for making the VMS end of things work with Eunice. Theresa Breckon wrote the MacIntosh spooler to get output to AppleTalk network connected LaserWriters, and Van Jacobson has been experimenting with Ethernet-AppleTalk gateways to this same end.

The quite reasonable UNIX software of Imagen made the initial effort manageable, and thanks goes to Dan Curtis of Imagen for his expert assistance with that software. The similarly good Adobe Systems, Transcript software made possible the attachment of the LaserWriters.

## 7. Bibliography

[1]

"An Administrator's Guide to the LBL Distributed Printing System,"
Johnston, W. E., LBL Report, PUB-3055, Lawrence Berkeley Laboratory, Berkeley (1986).

[2]

"4bsd UNIX TCP/IP and VMS DECNET: Experience in Negotiating a Peaceful Coexistence,"
Jacobson, Van, Craig Leres, Joseph Sventek, and Wayne Graves, *USENIX Summer Conference Proceedings*, (1984).

This report was done with support from the Department of Energy. Any conclusions or opinions expressed in this report represent solely those of the author(s) and not necessarily those of The Regents of the University of California, the Lawrence Berkeley Laboratory or the Department of Energy.

Reference to a company or product name does not imply approval or recommendation of the product by the University of California or the U.S. Department of Energy to the exclusion of others that may be suitable.

*LAWRENCE BERKELEY LABORATORY*
*TECHNICAL INFORMATION DEPARTMENT*
*UNIVERSITY OF CALIFORNIA*
*BERKELEY, CALIFORNIA 94720*