



The following paper was originally published in the
Proceedings of the Twelfth Systems Administration Conference (LISA '98)
Boston, Massachusetts, December 6-11, 1998

Large Scale Print Spool Service

Ignacio Reguero, David Foster, and Ivan Deloose, CERN

For more information about USENIX Association contact:

1. Phone: 510 528-8649
2. FAX: 510 548-5738
3. Email: office@usenix.org
4. WWW URL: <http://www.usenix.org>

Large Scale Print Spool Service

Ignacio Reguero, David Foster, and Ivan Deloose – CERN

ABSTRACT

The paper describes a project to enhance the print service for CERN.

The printer infrastructure consists of over 1000 printers serving more than 5000 Unix users running on workstations of various brands as well as PCs running Linux. In addition, the infrastructure must serve more than 3000 PCs running Windows/95 and NT 4.

We support a large number of printer manufacturers, including HP, QMS, Tektronix, Xerox and Apple.

Lightweight print clients are provided for all the supported platforms and transparently distributed using the ASIS software repository and the NICE application architecture. They may be used as “drop-in” replacements of the standard vendor clients. Compatibility with older CERN lightweight print clients is provided. Printing with standard vendor clients is also possible.

Administrative tools are provided for the general management of print servers and in particular for replicating server configurations and monitoring spool file systems.

The service offers a high level of scalability and fault tolerance, since it has no single point of failure in the server back-end.

Background

CERN [3] is the Laboratory for Particle Physics Research on the Franco Swiss border near Geneva. It is funded by nineteen European countries with a strong participation from other countries like Israel, Japan, the Russian Federation, Canada, and the USA. Some 8000 physicist and engineers work on the CERN site at any one time.

The Printing Problem

The previous printing solution for CERN was based on a single, now obsolete print server (a DECstation running Ultrix). The print server was acting as a print spool server and Appletalk protocol converter for the whole site. This situation was leading to unacceptably poor availability and increasing printing delays. Furthermore the configuration was very complex as it used two different Appletalk stacks. One using CAP and a Shiva box to deliver IPTalk and another one with Netatalk to deliver EtherTalk. CAP was required to support the older Apple LaserWriter models like the Personal LaserWriter NT.

Whereas the printer server served mainly Unix clients, printing services for Windows PCs had to go mostly through Novell servers and sometimes through Windows NT servers thus making debugging and queue status monitoring very difficult. Some printers were being served simultaneously by a number of print servers and in some cases print jobs were passed between servers. This resulted into an extremely complex situation with many opportunities for problems.

Centralized management, Heterogeneity and Size of the Installation

CERN offers centralized computing services to a very varied community. This explains the fact that little control was possible on the model of printers which was bought, therefore a large number of different printer models, including those of HP, QMS, Tektronix, Xerox and Apple printers, is in use. These printers are accessed from a variety of Unix workstations such as Sun, HP, DEC, IBM and SGI, as well as PCs running Linux, in addition to the PCs running Windows/95 and NT 4.

For historical reasons CERN has a very large population of Appletalk based printers. These still constitute around 60% of the current total population of around 1200 printers. The previous printing service was needed to provide Appletalk conversion for printing from Unix hosts to Appletalk-based printers. The service was then expanded to support a lightweight print client that would eliminate the need for a spooling configuration on the Unix clients. The main problems with this service were the single point of failure, the lack of scalability of the server, and the complexity of managing the PC printing environment with Novell print servers, which provided parallel print services with interconnections.

At the end of 1997 a task force was set to define a new printing architecture for CERN that would be more manageable by simplifying printing operations, removing single points of failure, offering better scalability, and easing integration of new printing technologies.

Commercial Software and Commodity Computing

The more important requirements of the printing architecture are listed below:

- Reduction of the number of print servers servicing a particular printer to (ideally) one.
- Simplification of queue status monitoring.
- Reduction of the number of Netware print servers to one to service IPX-only printers, and the eventual deprecation IPX-based printing.
- Good Appletalk printer support.
- Support for both Windows and Unix clients.
- Preferably use a single protocol to interface to the server queueing back-end.
- Flexible architecture, allowing for easy inclusion of new technology, such as the Internet Printing Protocol.
- Automatic handling of all printer driver issues on the Windows clients.
- Lightweight clients, i.e., without spooling, to simplify client configuration.
- Use of mainstream software and commodity hardware.
- Easy system management.
- Elimination of single points of failure.
- Scalability for thousands of queues with a limited number of servers.

In summary, the aim is to reduce the complexity and diversity of the printing solutions currently being used while being able to support the diverse range of client systems.

Several products were reviewed, but were rejected due to their inability to support the large number of different printer types in use at CERN, or due to their inadequate client system support.

Most of the software solutions relied on Windows NT Print Servers. The scalability of these servers as specified in [17] is of the order of 40 queues per server. However, we have reports of Windows NT servers serving on the order of hundreds of printers. In any case this is not enough for our requirements. Also the support for Appletalk and IPX printers is of unknown quality and the NT printer server will not serve Windows 3.1 users. The inability to use filters is another serious limitation of Windows NT servers.

Finally, the task force arrived to a consensus based on the following conclusions:

- Enterprise printing is still dominated by custom solutions based on Unix servers.
- We cannot buy a complete packaged solution suited to our needs.
- The solution should run on a leading Unix implementation on commodity PC hardware.

The solution finally implemented relies heavily on Unix Public Domain products under GPL [10]. The high quality of products like the LPRng spooler [14, 13] has allowed us to build a very reliable system that is flexible enough to meet our scalability requirements and to support our heterogeneous environment.

Print Server Environment

Choice of Printing Protocol

RFC1179 [11] was chosen as the protocol for the print clients to interface to the server back-end. This is the protocol of the BSD `lpd` printer server. There are several reasons for this choice:

- RFC1179 is the only non-proprietary standard available today.
- The scalability of connection-oriented protocols like SMB is limited to less than hundred queues per server.
- The management of connection oriented protocols can be complex. In the past we often saw that Windows PCs on some network segments were losing their connections to Novell print servers, thus requiring a manual user intervention and reconnecting to his printer.
- It is available and easy to implement on all our supported platforms.

Appletalk Support

Although we have tried to migrate from Appletalk [1] to TCP whenever possible, the server back-end has to interface to both TCP and Appletalk printers. As more than 60 % of the printers still are on Appletalk this is an important requirement.

There are two main Appletalk protocol types on Ethernet: IPTalk, which is the transmission of Appletalk DDP packets inside IP UDP packets, and EtherTalk which are Appletalk protocols transmitted in specific type Ethernet packets.

Our goal was to support our whole Appletalk printer population with a single Appletalk stack running native EtherTalk phase 2. For efficiency reasons, we required a kernel-level implementation of Appletalk, such as in the Linux kernel DDP module.

We tried to avoid the older IPTalk protocol because it requires maintaining a static routing configuration and special software on a gateway, like a Shiva FastPath to translate IPTalk packets to/from EtherTalk or LocalTalk.

After we first installed the Netatalk package [15], we found that the PAP (Printer Access Protocol) client would not support the idiosyncratic behavior of old Apple LaserWriter printer models, like the Personal LaserWriter NT, hence we had a look at the Columbia Appletalk Package CAP [4].

CAP V6.0 with patch 198 supports EtherTalk using the Linux kernel DDP module and the papif CAP PAP driver successfully runs all our population of Apple LaserWriters including the old models.

However, in both cases, the amount of multicasts required to run Appletalk would break the driver of our Ethernet card. This was the Intel EtherExpress Pro 100B. More specifically, the `eepr100` driver generated timeouts and reseted itself before processing the multicast lists required for our Appletalk network, so

the driver was modified to disable hardware multicast filtering.

In the current version of the driver the author has added a module-settable variable “multicast_filter_limit.” It may be set to ‘0’ to disable the multicast filter and set the chip to Rx-all-multicast mode, whenever any multicast address is accepted.

Spooling Software

LPRng is our choice for print spooler software to support RFC1179 printing. It is mostly compatible with the lpd BSD print spooler while having a large number of enhancements and new features, some of which are exploited in our project. Features most relevant to us include:

- Lightweight print clients with both SysV and BSD flavors, requiring no databases or spooling setup in the clients.
- Better security, since no SUID root is required in the client software.
- Improved permission and authorization mechanisms: a set of rules determines the type of actions that a given user can perform.
- Improved filter model with support for parameters, and for separated banner page generation, and accounting filters.
- Support for dynamic redirection of queues.
- Excellent debugging facilities.
- Flexible customization model.

Filters

When printing using the BSD lpd model, the server passes the files through a filter that processes them according to a format specified in the control file and the type of device. The processing includes formatting, accounting and banner page generation.

LPRng supports a superset of the BSD printing filter model and allows greater flexibility. In particular, one can specify parameters for filters in the printcap database, and separate functions such as accounting, banner generation, etc. into different filtering programs. This feature was exploited mainly for the banner generation programs.

It should be noted that print jobs coming from Windows clients that use the “NICE Printer Wizard” go through the filters in literal mode, i.e., no formatting is required by the filters since it has already been done by the Windows driver. The filters still have to

do other tasks, such as handling communications with the printer, error reporting, accounting, etc.

apsfilter

We have chosen LPRng’s `apsfilter` as the base for our default input filter as it allows us to cleanly integrate different printer models within the spooling system.

The `apsfilter` is a modular script written in Perl [16], which allows us to have a generic behavior. It supports various communication protocols by using different drivers as back-end of the filter.

A `pstext()` function was implemented in `apsfilter` that uses other formatting programs, such as `a2ps`, `psf`, `ascii2ps` and `enscript` to emulate the formatting behavior of the former printing system. This was required for backwards compatibility. `pstext()` supports legacy formatting mechanisms, such as form codes and Fortran Control Characters.

Additional work was done to handle back-end filter errors by getting the printer status from the SNMP agent in the printer log and sending it by mail to the user that originated the print job, thus helping to debug the most common printer operational problems.

papif

We currently use the `papif` PAP driver of CAP as the back-end for Appletalk printers.

This filter was modified, to make the parameter list compatible with LPRng, and to improve status and error reporting.

CTI-ifhp

We make use of the `CTI-ifhp` filter distributed with LPRng to support not only HP-compatible printers, but most TCP printers.

This filter was slightly modified to use more portable tray selection PostScript commands. i.e., something like Listing 1 rather than Listing 2.

This modification was coordinated with logic added to the `apsfilter` so that some parameters are passed to the `CTI-ifhp` filter in order to change the default tray in the HP LaserJet 5Si Mx so that the default tray is the A4 tray, when having A3 and A4 paper sizes.

```
"<</DeferredMediaSelection true /MediaPosition 0>> setpagedevice\r\n"
```

Listing 1: More portable tray selection.

```
"statusdict begin 0 setpapertray end\r\n"
```

Listing 2: Less portable tray selection.

```
"<< /Policies << /Pagesize 2 >> >> setpagedevice\r\n"
```

Listing 3: Avoiding tray switching when paper is unavailable.

The logic in question also can select the printer tray according to the name used for the printer name as described in the *Multiple Tray Handling* section.

In order to avoid tray switching when no paper of the right size is available, we had to change the `PageSize` entry of the `Policies` dictionary by means of a command similar to that of Listing 3.

Unfortunately this is not always successful, since this sort of dictionary can always be overridden by further additions to the dictionary within the job. This is because the PPD files for many HP printers add a `<</DeferredMediaSelection true>>` dictionary entry that disables the `PageSize` policy.

The behavior of this filter may be customized by means of command line options in a very flexible way. For instance, some models such as the HP DeskJet 1600CM, do not support certain PJI commands. The `CTI-ifhp` filter may be configured to support them by means of the options in Listing 4.

These option settings disable the use of the PJI commands for status and page counts and use a PostScript command instead.

The minimal common denominator case may be supported by the `-Tstatus=off` option that treats the printer as write only device. This option may be also useful to support printers that are not bidirectional, when not getting page counts is not an issue.

phaserif

The `phaserif` filter by Al Marquardt et al. is used as back-end for Tektronix Phaser 450 and Phaser 300X models.

pfilter

We are currently testing the `pfilter` by Steve Wilson for Tektronix Phaser 560 support.

Multiple Tray Handling

Multiple tray printers are supported by means of `printcap` entries, an example is given in Figure 1.

The first entry resends the print jobs to the real device queue with a tag describing the characteristics

```
31-s-012-a3|31_s_012_a3|31-S-012-A3|31_S_012_A3:\
qq:lp=31-s-012@print1:force_queue_name=aps-PS_600-A3-auto-mono

31-s-012|31_s_012|31-S-012|31_S_012:\
qq:server:direct_read:mx#0:ps=status:pw#2:lp=137.138.34.47%9100:\
sd=/var/spool/lpd/31-s-012:\
if=/opt/lprng/lib/filters/apsfilter/apsfilter:\
of=/opt/lprng/lib/filters/ifhp -Tbanner=off:\
bp=/opt/lprng/bp/psbnrhplj5simx:\
vf=/opt/lprng/lib/filters/ifhp -c:
```

Figure 1: The `printcap` configuration file entries for multiple tray printers.

```
-Tsync=off, infostatus=off, pagecount=off, forcepagecount=on.
```

Listing 4: `CTI-ifhp` filter options for HP DeskJet 1600CM.

that are required. In this example, the tag is `aps-PS_600-A3-auto-mono`. This informs to the `apsfilter` which parameters should be passed to the back-end filter in order to select the correct tray. The second `printcap` entry describes the real device queue. All the jobs have to go through this same queue independently of page size or requested tray.

Figure 2 is an example of logic used to analyze the tag in order to determine the tray and pass it to the back-end filter.

```
if( $commprog eq
"/opt/lprng/lib/filters/ifhp" ){
# choose a paper tray
if ($opt_Q =~ /.A3/i ){
$opt_Z = "lower,fixtray";
$papersize = "A3";
}
elseif ( $papersize =~ /A4/i ){
$opt_Z = "upper,fixtray";
}
elseif ( $papersize =~ /A3/i ){
$opt_Z = "lower,fixtray";
}
else { $opt_Z = ""; }
}
```

Figure 2: `apsfilter` logic for multiple tray printers.

Banner Generation

The `-Tbanner=off` option has been specified in the output filter so that banner generation can be handled by an independent `bp` filter.

A `b1` entry has been set in `lpd.conf` as follows

```
b1=$-`H:$-`n Job: $-`J Date: $-`t
```

This entry determines the parameters passed to the banner programs.

Banner programs were generated to print the CERN logo, and user and job information in a way compatible with the former printing system. They

contain PostScript code to switch to the “yellow paper” tray, if it exists, so that the banner is printed on paper of a different color.

For Appletalk printers, the banner program writes out the banner into .banner in the spool directory for the papif input filter back-end, which can then later print it.

Accounting

Although no direct charging is done, accounting is enabled and a Perl script was written to process the accounting log files. It generates daily reports detailing usage, such as pages per user per printer, and pages per host per printer. It also generates summary reports of integrated usage for the last month.

Per printer information is accessible through a Web interface described in the *Web Interface* section.

Server Clustering

Clustering is required in the server back-end in order to guarantee scalability and resilience to failures.

Several alternative architectures were considered, for instance the one in which a “master” front-end machine distributes print jobs to an array of workers. This solution was rejected because it would still

present a single point of failure and it would complicate the management of print queues.

As a consequence a simpler architecture had to be developed with the following characteristics:

- An array of print servers which are configured with similar queue definition files (printcap, etc).
- Printer queues are distributed among the servers to balance the workload.
- A printer is served by a single server at any given time.
- A external naming service directs the print clients to the server assigned to each specific printer.

The advantages of this design are numerous:

- Queue management is easy as only one server sends jobs to the printer. Contention is not possible.
- In case of failure of one of the servers, a reconfiguration of the naming service database suffices to reallocate the queues served by the failed server onto active ones.

```
printername IN CNAME servername.cern.ch.
```

Listing 5: DNS configuration sample.

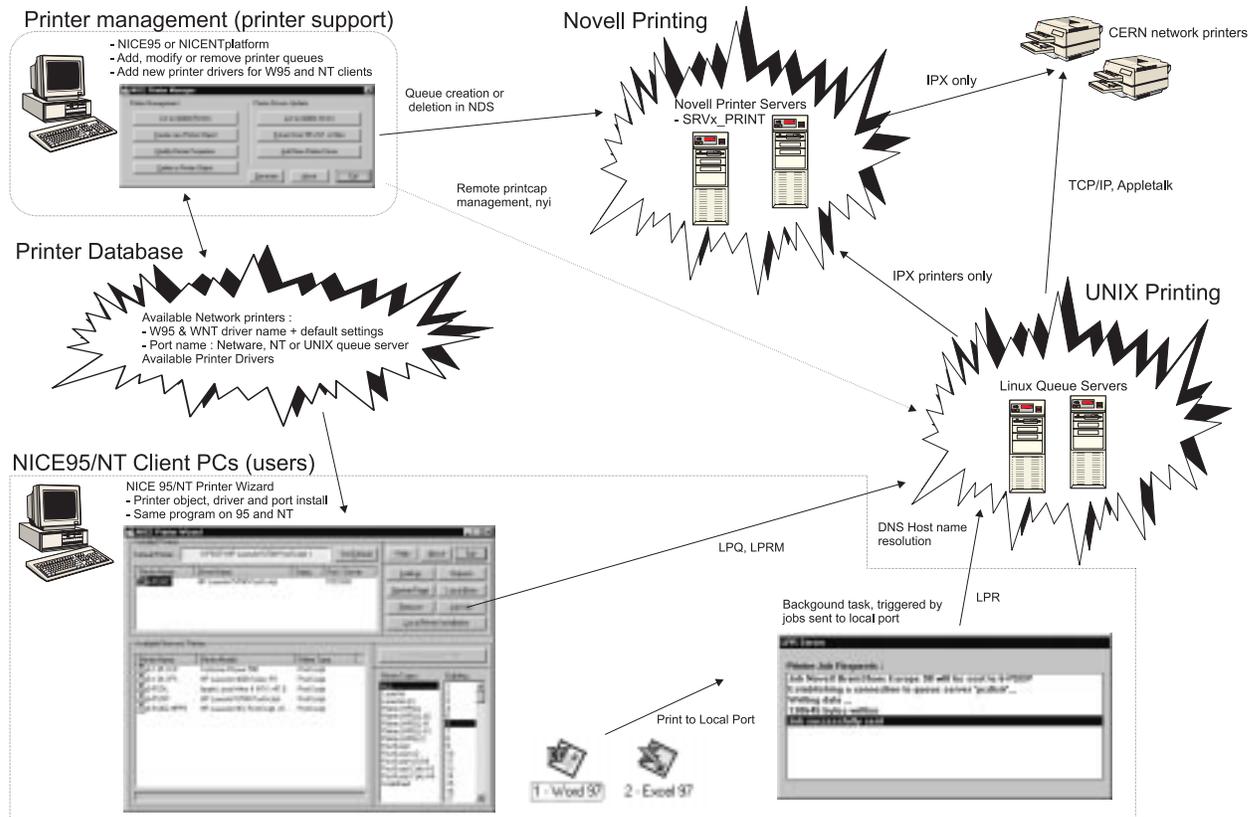


Figure 3: PC printing architecture overview.

Naming Service Setup

The Domain Name Service (DNS) has been chosen as dispatcher since it is easily accessible from the client software. Moreover a high availability DNS server infrastructure already exists at CERN.

A `print.cern.ch` DNS hierarchy was setup, such that for each printer, an alias of the form `printername.print.cern.ch` is available.

These aliases map printers to the server which serves their specific printer queue.

To generate the aliases, a DNS configuration file is required with entries as in Listing 5.

The caching lifetime for the entries was set to 10 minutes. The configuration is reloaded into the server also each 10 minutes. This determines the latency when reconfiguring the system.

A fall-back DNS configuration file is available for each server. In the fall-back file all the printers in the dead server are assigned to the other servers. Hence, when a server is down, the DNS is reconfigured by swapping the configuration file so that the queues can point to the surviving nodes.

Both Unix and Windows print clients that we support address the server through this DNS hierarchy as described in section named *The Unix Print Clients*.

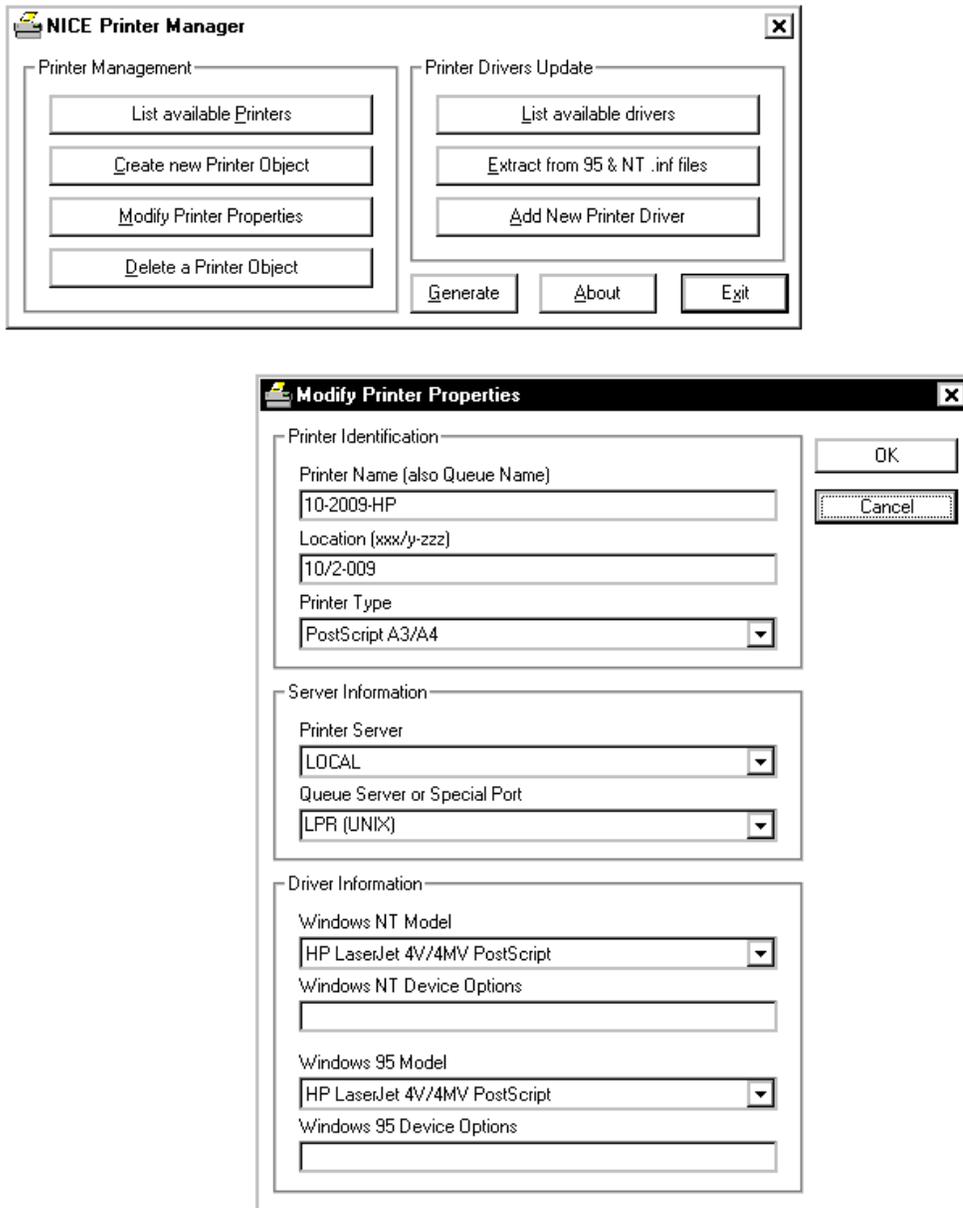


Figure 4: Printer Manager.

Scalability and Fail-over

The architecture implemented can support with two or three PC servers, our current printer population of around 1200 printers, for a client system population of some 2500 Unix workstations, servers and Linux PCs, 3000 Windows PCs and 1700 X-terminals.

The fail-over mechanism currently requires intervention of an operator and results in a slightly degraded mode, but the system is simple, robust and easy to manage.

The PC Printing Architecture

Introduction

From the PC/Windows point of view there is no commercial software available to integrate PC network printing into a Unix printing architecture. In order to avoid the deployment of platform dependent printing architectures (e.g., Netware and NT for the PC world) the CERN standard Unix printing server architecture has been adopted for the Windows clients.

The architecture for Windows 95 and NT clients can be grouped in the following building blocks:

- A common database and driver repository defining all CERN network printers (queue name, W95/WNT driver data,...) which is stored on a central application server.
- Administration tools: a set of programs used by the printer administrators to install, modify or delete network printers/drivers.
- Client software: Windows based programs running on the client PCs enabling the user to connect to any CERN network printer in an easy way.

The “NICE Printer Wizard” is the key application for printer installation and configuration on the PC platform.

The PC printing architecture is shown at Figure 3.

The printer database and repository

MS Access was chosen as the database medium because of its good integration with the PC world and the powerful development environment. The database contains the following information:

- **Supported printer drivers:** In addition to the standard printer drivers included in the W95 and WNT distribution CDs, the database holds information about third party drivers added by the administrator. For every driver, the database has a reference to the corresponding .inf and driver files. These files are stored in the printer data repository and used by the client software when a user connects to a printer.
- **Available network printers**, defined by:
 - **The name of the printer**, identical to its queue name on the Unix server
 - **The printer driver name**, the name of the .inf file as well as the path to the

driver files for the W95 and WNT platforms.

- **The printer type.** Selection is made from a predefined list in the database.
- **The default printers settings** as defined by the administrator.

Administration tools

The printer administration tool, called the “NICE Printer Manager,” has the following features:

- Possibility to add third party printer drivers. Any new printer driver which is not part of the standard W95/WNT distribution CDs can be added in an automated way. All information required to install this driver on a client PC is added to the database and the necessarily files are copied to the repository by the “NICE Printer Manager.”
- Management of the network printers. For every CERN network printer, the “NICE Printer Manager” allows the administrator to:
 - enter the printer name.
 - select the W95 and WNT driver name from the supported driver list.
 - choose the printer type.
 - define the default settings.

The data stored in the printer database by the “NICE Printer Manager” is extracted to binary data files after every modification. For performance reasons these files are then used by the client software instead of the database itself.

The Printer Manager is shown at Figure 4.

The Client Software

The client software is split into two components:

- **The “NICE Printer Wizard”:** a graphical interface which enables the user to connect to all network printers defined in the database. Connecting to a printer from W95 or WNT implies installing the driver files on the client PC, importing the default settings and creating the printer port. In this architecture, the printer port will point to a file on the local hard disk. The “NICE Printer Wizard” presents the available printers organized by building number and printer type, so the user knows exactly the capabilities and location of the nearest printers. Other options include the configuration of the printer settings and banner page. The “NICE Printer Wizard” is shown at Figure 5. On request, the printer wizard returns information about the current jobs for every connected printer with the possibility to delete them. Note that the job information is queried from the corresponding Unix printer server by implementing the lpq and lprm protocol inside the Printer Wizard. This gives the big advantage that a PC user has an overview of all current jobs on the connected printers, even sent by non PC platforms (e.g., Unix clients).

Another benefit in a heterogeneous client environment gained by having a single server (and hence queue) for a particular printer.

The Job Info window is shown at Figure 6.

- **The LPR Service module:** This task is permanently running in the background on every Windows 95 and NT PC and is the gateway between the local printer ports and the Unix printer server. It is triggered by the Microsoft Printer Change Notification Mechanism when the user prints a job from an application program. As defined by the “NICE Printer Wizard,” the printer port sends the data to a local file on the hard disk, containing the name of the printer. As part of the communication convention between the “NICE Printer Wizard” and the “LPR Service,” this background task

knows exactly where to find this file and will then transmit the contents to the Unix printer server using the LPR protocol.

The “LPR Service” window during a job transmission is shown at Figure 7.

We use the NICE [2] architecture to deliver and maintain the software so that it appears in the start menu of all our supported platforms by means of a number of replicated application servers to which all users connect.

The Unix Printing Architecture

The Unix Print Clients

The print clients supported are the LPRng clients which have been modified to allow the use of DNS to locate the print server. In particular, the modification

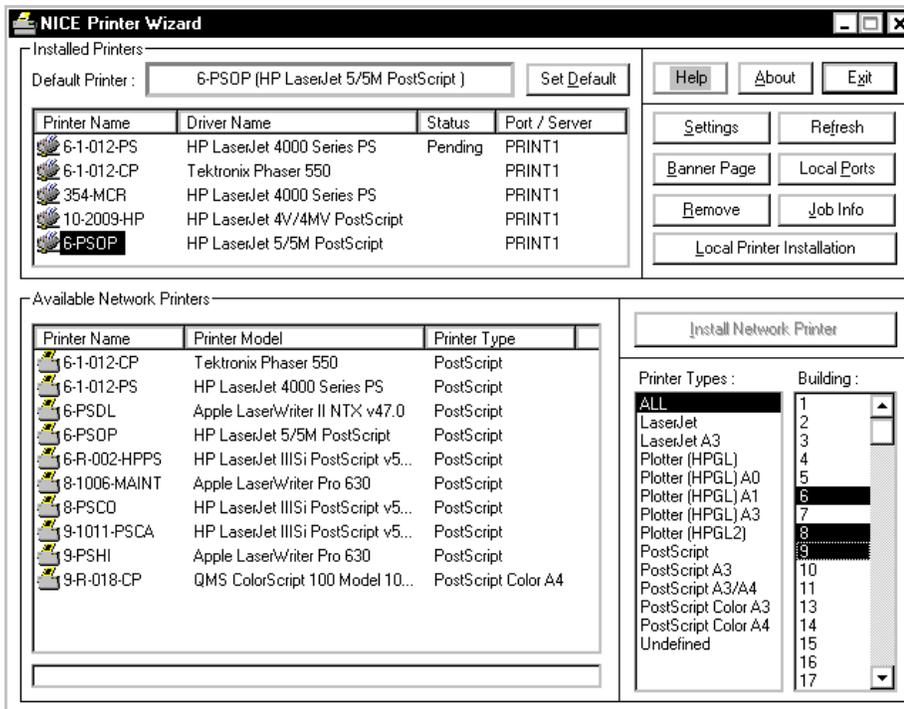


Figure 5: The “NICE Printer Wizard.”

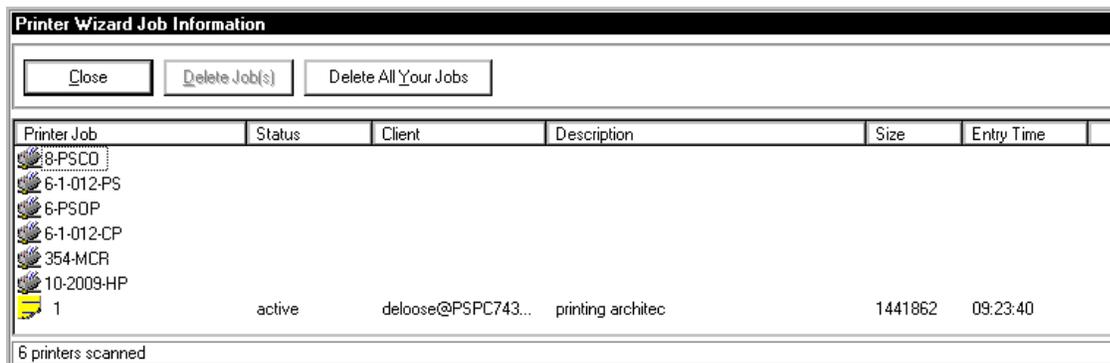


Figure 6: The Job Info window.

ensures that printer names that are specified as -Ppname are translated into names like pname@printername.print.cern.ch, thus making the print server back-end scalable in a way that is transparent to the print clients.

The DNS resolution mechanism is shown at Figure 8.

The LPRng clients have the interesting property of being lightweight. This means that they send the job directly to the print server, not requiring any spooling configuration on the client.

We use the ASIS [7, 8] repository to distribute the software so it appears in /usr/local/bin on

all supported platforms via a distributed file system such as AFS, NFS, or as a controlled local copy.

The supported Unix platforms are Sun Solaris, IBM AIX, HP HP-UX, DEC Unix, Linux on Intel and SGI Irix. We plan to replace the print commands distributed with these systems with the LPRng clients.

Support for Standard Vendor Clients

Although we do not recommend to use vendor specific print clients, LPRng in our servers has been configured with the “fix_bad_job” configuration option, so that it supports the widest spectrum of clients possible.

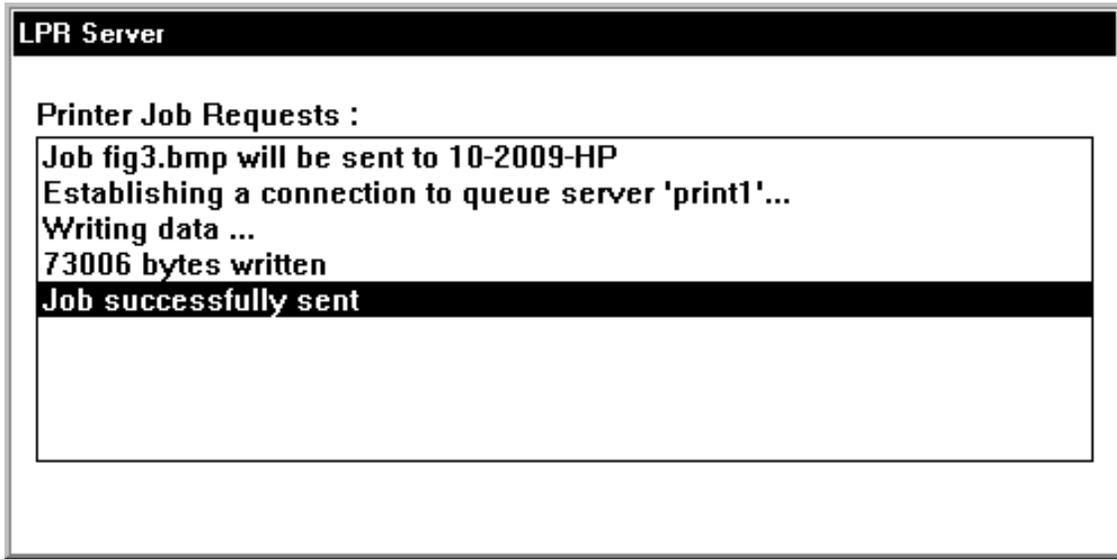


Figure 7: The “LPR Service” window during a job transmission.

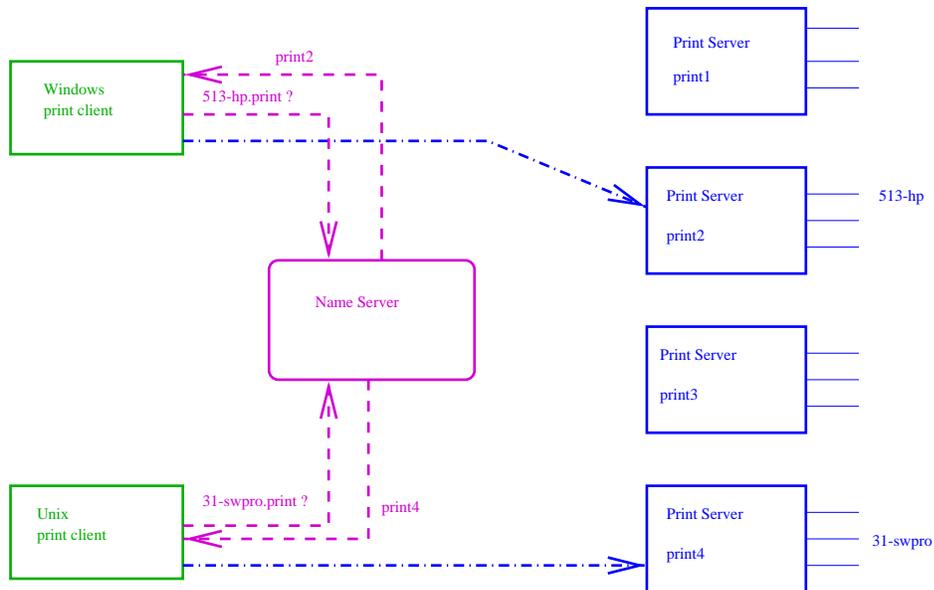


Figure 8: The printer resolution mechanism.

For instance a BSD `lpr` print client could access the print servers by using the following `/etc/printcap` entry:

```
printername:lp=:\  
rm=printername.print.cern.ch:\  
rp=printername:\  
sd=/var/spool/lpd/printername:mx#0:
```

Backwards Compatibility

The `xprint` lightweight print client has been used at CERN since more than five years. Since it uses its own protocol, it requires its own `xprintd` daemon machinery in the server to format the control file and spool the print jobs.

In order to simplify the printing system, it was decided to stop supporting the `xprint` code. Instead we use the interface in the LPRng print client to support different “personalities” of the client to implement a `xprint` like look and feel.

Some work was also required in the Perl script for `apsfilter` to support specific old formatting options, including forms and Fortran Control Characters as described in the *Filters* section.

Management

One of the main goals of the project was to simplify operations and management of the printing service. Therefore several tools were provided to ease the migration and the management of the service.

Migration Tools

The main migration tool is the `syncpcap` script that combines `printcap` queue configuration and Appletalk addresses from the old service to generate the LPRng server spooling configuration.

The generated configuration includes the `printcap` file, the spool directories, the configuration for `apsfilter`, the Appletalk address file `cap.printers`, and the DNS configuration.

Security

To reduce security risks, the LPRng client executables are installed as `non-setuid`. In the same line, access to the print servers is reduced to the strict minimum, the `ssh` secure shell has been installed and the `sudo` utility is used to delegate privileges to the operations team.

Authorization

The permissions for LPRng, both to print, manage print jobs, and to view their status, are controlled in great detail by the authorization rules of LPRng in `/etc/lpd.perms`.

The current rule set allows users to manage jobs submitted by themselves, but (unless they are members of the Printer Operations Group) they cannot control other jobs.

These rules are also being used to set access control lists in printers with expensive consumables, so that only certain users can print on them.

```
xterm
22-1-cor 4-2-002-320 594-r013-tek pisa-lw
22-cb-printer 4-2-11 599-uge pisa-lw1
2257-r-hp5ps 4-2-next 599-us70 pisa-lw2
23-1-022-ps 4-2021-cm 599-us96 pisa-lw3
23-1007-tek 4-3009 596-r020 pisa-lw640
23-kkg 4-3016-dm 596-r021-lw pollux-orsay
23-r-005 4-3019-f 596-r027-cm prtshop1
23-rolf 4-3022-f 599-hpc prttest
24-1-004 4-3024-f 599-ntcm ps4dr27f
24-1-8-hp4 4-3026-f 599-r000-lw-ps pythie-orsay
24-1018-tek 4-3029-lw 6-1-012-cp pythie-ps-orsay
24-e011-dc 4-3030-f 6-1-012-ps ssdd-161-1
24-tisdi 4-3cor-f 6-psdl t9-nonad
24-tiege 4-3cor-hp 6-psop unige-1
24-tisrp 4-dgc 6-r-002-hpps unige-2
25-r036-plt 4-doran 60-2001-cp unige-3
25-r038-cm 4-dtpp1 60-2005-cp unige-4
25-r038-cm-t 4-fi-ppp 60-2006-ls unige-5
252-1-hp4ps 4-ps205 60-2008-c unige-6
252-1-lw 4-qap-laser 60-2009-hp unige-7
252-1000-hp3ps 4-r-057-ess 60-2009-pb van6ps1
252-atrf 4-r005-ess 60-2011-cm van6ps2
26-1-003 4-r046-ess 60-2011-lw van8ps1
26-2001-cm 4-r05-cm 60-2014-cp van8ps2
26-2001-cm-t 4-r13-capro 60-2016-pb misc1
26-2001-hp 4-thcolor 60-2022-hp misc2
[sunadn] project/printing/luc > 31-1029-dci
31-1029-dci>lpq
Printer: 31-1029-dci@print1 '31-1029'
Queue: no printable jobs in queue
Status: server finished at 20:58:50
Filter_status: ifhp Initial page count 235378, final 235382, Total pages = 4, e
lapsed time 64 secs at Sep 20 20:58:50
31-1029-dci>status
Printer Printing Spooling Jobs Server Slave Redirect Status/Debug
31-1029-dci@print1 enabled enabled 0 none none
31-1029-dci>
```

Figure 9: Printer status with `lpc` extensions.

Queue Status Information

Operations are greatly simplified already by the fact that a single queue is used for all users of each printer, whether they submit from Windows or Unix. This fact also circumvents the large family of problems that may arise when several servers attempt to connect to a same printer simultaneously.

The standard tools provided with the print clients, such as `lpq`, `lprm` and `lpc`, can be used to manage the print queues.

Any of these commands can be used in any client independently of the location because they have been modified, in the same way as the print clients, to support our DNS print hierarchy.

The permissions are controlled by the authorization rules of LPRng as described in section named *Authorization*.

For compatibility with an older tool, some extensions to `lpc` have been implemented. For instance, a directory in the AFS distributed file system holds all the printer names available as links to the `lpc` program, so that running one of them directly calls `lpc` for that printer. The usage of this interface is shown at Figure 9.

In addition, a WWW interface is offered for these commands. This interface is described in the *Web Interface* section.

Printer Status Reporting with SNMP

A utility script has been developed using the CMU `snmpget` and `snmpwalk` distributed with Linux to retrieve the current status of a printer. This is particularly useful for problem determination.

The status is also made available through the WWW interface described in described in the *Web Interface* section.

Server Configuration

In order to ensure that the system configuration is exactly the same in all print servers, the installation of these systems has been completely automated.

The `kickstart` utility that is distributed with RedHat Linux is used to automate the basic system installation. This is then linked with the installation of SUE [6, 5]. SUE is a Unix automated system management framework done at CERN. SUE features for LPRng, CAP, and the configuration management scripts install and guarantee the consistency of these elements.

The configuration management scripts insure the correctness of the queuing configuration of the servers. Changes to the configuration must be securely propagated to all servers.

The queuing configuration includes the `print-cap` file, the spool directories, the `apsfilter` configuration and the Appletalk `cap.printers` address file.

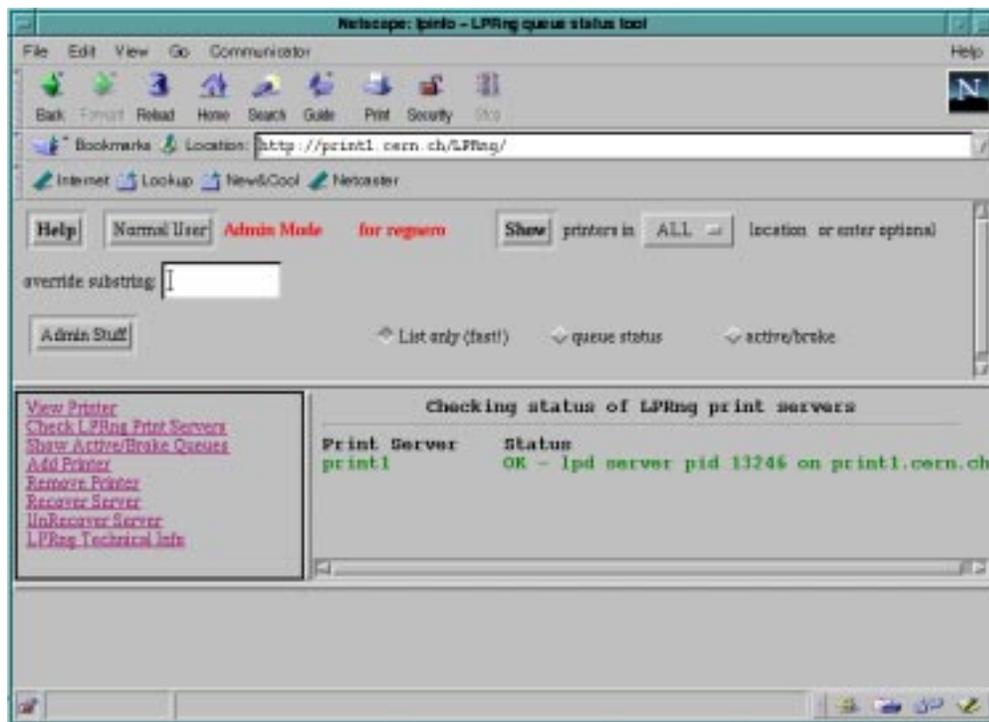


Figure 10: WWW administrative interface with lpinfo.

Server Monitoring

All the print servers are included in the CERN specific alarm system. In this system both a heartbeat signal and several other checks on processes and system resources are done. In particular spool space usage is periodically checked this way.

Web Interface

Users are provided a WWW interface to view the status of the printers. The same interface allows the Printer Operations team to actually control the printers. This functionality is based on the `lpinfo` CGI scripts [12].

The `lpinfo` Web server(s) and CGI scripts are run in the print servers.

The following features have been added to this interface:

- View queue status
- View server status
- View SNMP printer status
- View accounting for a printer
- Add printer
- Remove printer
- Move server out of the cluster
- Move server into the cluster

A view of this interface is shown at Figure 10.

Recovery Procedure

For convenience, the DNS server loads the configuration file from a directory in the AFS distributed file system.

Currently, the configuration file describing the print DNS hierarchy is read once every 10 minutes.

We have prepared alternate configuration files for failover situations. Therefore the recovery procedure consists of copying the specific alternate configuration file to the one in production, thus redistributing the queues in the broken server to other nodes of the print server array.

This operation is supported through the WWW interface described in the *Web Interface* section.

The Status of the Project

The PRINT service is currently running on two Digital Prioris PC servers with 256 Mbytes of RAM and 9 Gbytes of disk running the LPRng spooler under the Linux RedHat operating system.

The choice of PC hardware was based on price/performance considerations. Linux was chosen as a leading Unix implementation on PC hardware as it offers very good kernel-based Appletalk implementation, and an excellent level of system support from the network community. However, nothing prevents the implementation of our solution on other hardware or on a different flavor of Unix, such as Sun Solaris.

Deployment Plan and status

The new service is progressively replacing the old printing system.

Most of the PC users have been migrated to the new system, while the Unix clients are currently under test. The migration for the Unix clients will proceed during fall 1998.

Availability

CAP + LPRng + Linux

All these components are publicly available. Their integration is described in this article. If interested in specific details please contact Ignacio Reguero (i.reguero@cern.ch).

Printer Wizard

For information on the “NICE Printer Wizard,” please contact Ivan Deloose (ivan.deloose@cern.ch).

NICE

Information on the NICE architecture is available through <http://nicewww.cern.ch> or from David Foster (david.foster@cern.ch).

Future Developments

We are studying how to rationalize the dissemination of status information for Windows clients to reduce the number of `lpq` queries required to show the print queues in the printer wizard.

We would like to implement Web access to server back-end status, as well as user notification using *Zephyr* [9] messages.

We are considering the possibility of automatically triggering the recovery procedure from the alarm system when a print server is missing.

Author Information

Ignacio Reguero received the “Ingeniero Superior de Telecomunicacion” degree from the Polytechnic University of Madrid in 1985. He studied in parallel two curriculum areas: “Data Communication/Data Transmission” and “Computer Science/Data Transmission.” His graduation thesis was “Specification and Implementation of Interpersonal Messaging Protocol, Following the X.420 Standard.” He worked in 1986 and 1987 as system engineer for the IBM IN network. He worked in 1988 for “Banco De Santander” as communications systems engineer of a network of more than 2000 offices. Since 1989, he is working at CERN on various fields such as large systems performance and communications, network backup systems, automated system software installation and maintenance and consultancy for Unix system administrators. He can be reached at [<Ignacio.Reguero@cern.ch>](mailto:Ignacio.Reguero@cern.ch).

David Foster received the Bachelor of Science degree in Applied Physics and Electronics from Durham university in 1978. This was followed by the Doctor of Philosophy degree in 1982 for work in real-time multiprocessor control systems for atmospheric

physics experiments. Since 1981, he has worked at CERN on a variety of projects including compiler design for cross platform development, communications to IBM mainframes and management of enterprise PC systems. He has acted as an industry consultant in these fields and his work has been widely published. Currently he is the manager of the PC desktop support team at CERN and is completing a Master of Business Administration at Durham University Business School specializing in large scale IT infrastructure support for knowledge management. He can be reached at <David.Foster@cern.ch>.

Ivan Deloose received the “Industriële Ingenieur Elektronica, optie Telecommunicatie” degree from the Industrial High School of Kortrijk in 1987. His graduation thesis was based on the design of a Digital Video Effects Mixer. He worked in 1989 as development engineer for BARCO INDUSTRIES. In parallel, he developed the electronics for a large video projection system (video wall). Since the end of 1989, he worked at CERN in the sector of operations and controls of the particle accelerators. He was especially involved in a wide variety of software projects in the domain of accelerator controls. In 1995, he became the divisional responsible for office automation and network infrastructure. He is managing the design and architecture of PC based control system software inside the division and for some attached experiments. He can be reached at <Ivan.Deloose@cern.ch>.

References

- [1] Apple Computer, Inc. *Inside Macintosh: Networking*, <http://developer.apple.com/techpubs/mac/Networking/Networking-2.html>.
- [2] CERN. *NICE Architecture*, <http://nicewww.cern.ch/homepage/Documentation.htm>.
- [3] CERN, European Laboratory for Particle Physics. <http://www.cern.ch/>.
- [4] Columbia University, et al. *Columbia Appletalk Package*, <http://www.cs.mu.OZ.AU/appletalk/cap.html>.
- [5] Lionel Cons. *SUE Users Guide*. CERN, the European Laboratory for Particle Physics, 1996.
- [6] Unix Workstation Support CN Division. *SUE Definition Document*. CERN, the European Laboratory for Particle Physics, 1.00 edition, 1995.
- [7] Ph. Defert, et al. “Automated management of an heterogeneous distributed production environment.” In *First Conference on Freely Redistributable Software*, pages 1-8, 1996.
- [8] Ph. Defert, et al. “Managing and distributing application software.” In *LISA'96 Proceedings*, pages 213-226, 1996.
- [9] Tony Della Fera, et al. ‘Zephyr Notification Service’ *Project Athena Technical Plan Section E.4.1*, Massachusetts Institute of Technology, 1987.
- [10] GNU. *GNU General Public License*. Free Software Foundation, Inc., 1991.
- [11] L. McLaughlin III. *RFC 1179*. Network Printing Working Group, 1990.
- [12] Alek Komarnitsky. *lpinfo*, <http://www.komar.org/komar/alek/>.
- [13] Patrick Powell. Managing network printers and print spoolers. In *LISA XI*, Tutorial T15pm. Astart Technologies Inc., 1997.
- [14] Patrick Powell and Julian Mason. “Lprng – an enhanced printer spooler system.” In *LISA IX Proceedings*, pages 13-24. Usenix, 1995.
- [15] The University of Michigan. *Netatalk*, <http://www.umich.edu/rsug/netatalk/>.
- [16] Larry Wall, Tom Christiansen, and Randal Schwartz. *Programming perl*. O’Reilly and Associates, Inc., 1996.
- [17] *Microsoft Windows NT Server Resource Guide*, Microsoft Corporation, 1996.

