# Remote booting via PXE


## Iain Rae
### Division of Informatics

**iainr@dcs.ed.ac.uk**


Some notes on medialess booting.

# 1. Overview

Most new bought PC's come with a managed Network card, usually supporting PXE. e.g. 3com's managed boot agent cards (http://www.3com.com/products/software/dynamicaccess/dyn_mba.html). This allows for PC's to be installed completely over the network, without the need for boot media. In the case of labs or a large cluster of similar computers it also allows the possibility of unnattended batch installations. The same technique could be used as a basis for diskless clients, or conceivably to run hardware updates which are only available via DOS such as, ironically enough, upgrading the pxe client stored in flash memory on the network card.

The administrator can control the installation via lcfg and by manipulating the **pxegrub**, **dhcp** and **install** objects which should (if necessary) rebuild the dhcpd.conf file, restart the server, create the grub .lst files, create the appropriate symbolic links, set up nfs exports etc.

The PC can then be rebooted to a menu which allows you to choose to perform a standard install as with the dcs boot disk.

It is also possible to flag a PC to (re-)install without user intervention the next time it is rebooted.

# 2. Server side configuration

## 2.1. Software requirements

Service requirements are fairly simple with DHCP and TFTP servers being all that is necessary to get the bootloader up and running and an nfs exported filesystem containing installroot in order to run the install. Because grub only supports a bootp client the dhcp server must allow bootp booting. The PXE spec states that the boot server must provide tftp and should provide mtftp (this may be redirected). Since we are using dhcp/bootp multiple times (once via pxe, once via grub and at least once via linux) it is probably easier to run the dhcp, installroot and tftp services on the same server. It is possible to store the RPM's on a separate nfs server.

### 2.1.1. DHCPD server

The recommended server is v3 of the ISC (http://www.isc.org/products/DHCP/) server, at the time of writing this was still in Beta, however there have not been any issues and I have used the v3 dhcp server before without any problems.

With some NIC's (usually those supporting PXE versions < 2.2) it is necessary for the dhp server to pass option 60 "vendor-class-identifier "PXEClient" ", if the DHCP server doesn't support this (the v3 one does) you need to setup a proxy DHCP server (such a server is bundled with redhat 6.2) though this does introduce another point of failure and is messy.

#### 2.1.1.1. The dhcp object.

The dhcpd v3 server is packaged as dhcp-3*.rpm to distinguish it from the v1 server currently deployed, similarly the dhcp object is packaged as obj-dchp-3.0-*.rpm. Both objects use the same dhcp.def file in order to minimise maintenance. Where v3 specific options are used they are flagged as such in the .def file and the doc method.

The following methods have been added to the v3 object

- **run**. Restart the dhcp daemon (effectively get it to reibuild the dhcpd.conf file).

- **restart**. Restart the dhcp server without rebuilding the dhcpd.conf file. It is intended the this should be used in conjunction with the **addhost** and **rmhost** methods in scripting.
- **hasentry** *`hostname`*. Check to see if *`hostname`* has an entry in the dhcpd.conf file.
- **addhost** *`hostname`*. Add host to the dhcpd.conf file (but will not restart the daemon)
- **rmhost** *`hostname`* remove an individual host from the dhcpd.conf file.

#### 2.1.1.2. dhcpd.conf file

NB the dhcpd.conf file syntax has changed slightly between v2 and v3.

The next section contains some of the grubby details of what's happening under the hood of the dhcp object, if you're just looking for the overview it's probably enough to know that **filename***`pxegrub`* defines the file that PXE will download and run (in this case the second stage grub bootloader) and that each host has an **option-150** entry which points grub at the correct menu configuration file to download.

The first mildly annoying thing is that we need to define the ddns method to be used (v3 supports dynamic dns) so we need to set **ddns-update-style ad-hoc;**. We also define some options to make things a little easier to read.

**Example 1. dhcpd.conf entries for mtftp**

```
option space PXE;
option PXE.mtftp-ip    code 1 = ip-address;
option PXE.mtftp-cport code 2 = unsigned integer 16;
option PXE.mtftp-sport code 3 = unsigned integer 16;
option PXE.mtftp-tmout code 4 = unsigned integer 8;
option PXE.mtftp-delay code 5 = unsigned integer 8;
class "pxeclients" {
        match if substring (option vendor-class-
identifier, 0, 9) = "PXEClient";
        option PXE.mtftp-ip 225.0.0.1;
        option PXE.mtftp-cport 76;
        option PXE.mtftp-sport 75;
        option PXE.mtftp-tmout 1;
```

```
        option PXE.mtftp-delay 2;
        vendor-option-space PXE;
}
```

This defines a class of hosts “pxeclient” which all get the PXE. options passed to them. In this case we are passing the multicast address of the file (225.0.0.1 which is defined elsewhere to be /tftpboot/pxegrub), the mtftp port for the client (76) and the server (75) and some timeouts.

* *NB I am dammned if i can get the mtftp stuff to work, either the NIC’s are not sending out requests or slapin is ignoring them.*

For each client host we have an entry which looks like the following code fragment

**Example 2. dhcpd.conf entry for a typical client**

```
host minibw5 {
                hardware ethernet 00:b0:d0:9b:c4:e3;
                fixed-address 129.215.58.147;
                option option-
150 "/tftpboot/grub_hosts/129.215.58/minibw5.lst";
                option host-name "minibw5";
        }
```

Each host has an individual entry for option-150 which grub uses to indicate which file should be used as the boot menu, for more details see the grub and obj-install sections.

### 2.1.1.3. Setting up a new server

Multiple dhcp servers should be able to co-exist on the same wire and the dhcp objects should not be able to define entries on multiple servers. Note however if you change the **host.install.bootserver** entry you must remove the dhcpd.conf entry from the old boot server, either by running **rmhost** or **stop/start**. If this is the first server on a wire then you will have to define some defaults, use the existing entries in the dhcp.def file as a guide

There are no special steps involved in running the dhcp daemon on the server, beyond ensuring that the rpms are installed. An **om *server*.dhcp start** will reconfigure the server and start it.

### 2.1.2. Tftpd server

One thing which became quickly apparent was that there are a number of variations on the tftp "standard" (see RFC1350, RFC2090, RFC2347, RFC2348 and RFC2349), mostly revolving round support for mtftp and for larger than 512 byte data blocks. Some bootloaders will only work with tftp daemons which support better than 512 byte block and others will only work with specific daemons. Grub seems to work with most and in the first instance we are using it with the supplied redhat 6.2 daemon. The current version of grub does not support mtftp however given the size of the menu files I don't think that's ever going to be an issue.

#### 2.1.2.1. Configuring the tftp server

Tftp has to be an allowed service for the inet object and suitable entres for /etc/inetd.conf and hosts.allow need to be generated.

### 2.1.3. Grub

Grub is the GNU GRand Unified Bootloader (http://www.gnu.org/software/grub/) which supports a fairly large number of operating systems. Whilst it was designed for installation on the local disk (like lilo) the second stage grub loader (pxegrub) can be downloaded via tftp and works well with our PXE NICs. By default grub provides a command line enviroment like the Sparc PROM which can be used to perform some basic diagnostics and to boot the Operating System. It is possible to define a simple menu based interface using a configuration file.

In the case of pxegrub the location of this file is passed via dhcp as option-150 and should be the absolute path to the file. Both the command line and menu interfaces are available via a serial console if required.

In most cases the default configuration is to chainload the boot loader on the master boot record of the first hard disk, with the standard dcs linux install available as a menu option. Where **install** has been used to flag a hands off install of the PC the default is to boot the dcs installroot.

More information on grub can be found at the website (http://www.gnu.org/software/grub/) or via the grub manual (http://www.dcs.ed.ac.uk/home/iainr/documentation/grub/grub_toc.html)

### 2.1.4. The Linux install object.

This is a re-write of the Solaris **install** object and at the moment is limited to manipulating the dhcp object and maintaining the client grub menu files.

#### 2.1.4.1. The /tftpboot directory

As we are chainloading the bootloader installed on the PC there are in fact far fewer Grub configurations than there are hosts. For a standard dcs 6.2 client there are basically two options, boot from the hard disk or boot the installroot in preparation for an install. In most cases a boot menu allowing the PC to boot from the hard disk by default and providing a clean installation as a menu option is enough.

In some cases we may want to install a PC without any prompting, when installing a large number of PC's at once, a lab or a Beowulf perhaps, or when we don't have access to the console. So we need a configuration which will only boot the dcs installroot. In fact this is slightly trickier as once the initial install has finished (at the end of the dcsrc script) we need to switch the configuration back to booting off the hard disk.

The third most obvious option is for diskless clients which currently we don't use but would be similar to booting the installroot (probably you'd need to override the root option in dhcp for each host or class of host).

The grub configuration files are stored in /tftpboot/grub_files and are built from lcfg by the grub object using the install.system_type, install.console and install.zapable keys to determine which files to create.

* *Well they will be when I write it.*

The configuration files for individual hosts are generated by the install object as symbolic links, partly to make it easy to change the configuration file for a class of hosts and partly to minimise any changes to the dhcpd.conf file.

In order to flag a PC for a hands off installation we run **om** *`installserver`*.install zaphost *`installhost`* this will check to see if "zapping" the host is allowed and set up the appropriate symbolic link.

* *Once the install has finished the dcsrc script will perform some magic to tell the install server to switch the symbolic links back.*

## 2.2. Client side configuration

The first decision to make is how dependant do we want, or need, to be on the dhcp server in order to boot the PC normally. The boot order (and what happens in the event of failure) is also partly dependant on the PC BIOS and how the NIC behave. Needless to say different cards behave in different ways. Of the 3C90X's we have there are at least two distinct versions.

The more up to date cards support MBA v4.x and PXE v2.x. This allows PXE to be chosen as the primary boot method and if the PXE client times out it will fall though to the next boot method. Just before the initial PXE client start the user is given the option of bypassing PXE by hitting the H key. The PC will fail to boot only if the PXE client cannot tftp the pxegrub file, the user can always force booting off of the hard disk by hitting the H key.

The second group of cards (MBA 3.x PXE 0.9X-1.X) can be configured to boot either off the hard disk or the network and will prompt the user to override the default, these clients will not "fall thought" in the event of failure and they will hang.

In both cases if a DHREPLY is recieved the PXE agent will assume that an attempt to boot from the network has been made and will either try again or reboot the PC. The general trend in the spec seems to be away from just supporting diskless clients and they are becoming more configurable.

3Com provide flash updates for the MBA agent and PXE client but these are minor updates only (cannot update a 3.x MBA to a 4.x one).

Suggested configurations are to default both sets of cards to boot off of hard disks with the "boot from network" prompt and configure PC's which are going to be installed "hands off" to boot from the network by default.

### 2.2.1. Installing via a pxe NIC

On the client configure the NIC's boot agent the way you want it, the details vary from card to card but will probably involve switching on the management functions and setting the boot order in the bios. In terms of the management agents themselves it's usually just a matter of selecting PXE.

Within lcfg set the install.bootserver appropriately, note that if the PC is set up with a serial console the grub menu will be set to display on the serial console, this can

be overloaded with **+install.console xterm**

Run **om slapin.install add *hostname*** or **om slapin.install run** if you're installing a lot of PC's.

Reboot the PC, if everything goes ok you should see a screen something like the following.

**Example 3. GRUB Menu.**

```
   GRUB  version 0.5.96  (640K lower / 128696K upper memory)

 +-------------------------------------------+
 | Linux                                     |
 | DOS                                       |
 |                                           |
 |                                           |
 |                                           |
 |                                           |
 |                                           |
 |                                           |
 +-------------------------------------------+
      Use the ^ and v keys to select which en-
try is highlighted.
      Press enter to boot the selected OS, 'e' to edit the
      commands before booting, or 'c' for a command-line.
```