PXE
Installation of SME 6.0 on a VIA Mini-Itx

© RF-232 & Michel-André Robillard
webmestre@micronator.dyndns.org

Mars 2004

1.0 GOAL

Using PXE to instal SME 6.0 on a VIA Mini-Itx small form factor computer with no CDROM nor floppy drive.

2.0 IN A FEW WORDS

We install TFTP, DHCP, and NFS on a Linux server (RHL 7.3) and test them with a standard workstation. We copy the latest version of SME to our NFS shared directory and we adjust "install.cfg" configuration file.

We attach our future SME server directly to our Red Hat 7.3 portable computer through a crossover cable, check its BIOS settings then make it boot from the LAN. We accept the licence and after a few minutes the SME is automatically installed.

We use a crossover cable because we don't want to change the setting of the DHCP server that is already on our LAN. If you want to use your regular DHCP just make sure to set the proper boot method of the future SME server to **boot from disk** instead of **boot from LAN** after the installation so it won't loop back to installation again.

3.0 TFTP

3.1 tftp-server-0.28-2.i386.rpm & tftp-0.28-2.i386.rpm

Install the server and the client. The client is used to check the server installation for problems.

3.2 installation on the server

```
bash# rpm -Uvh tftp-server-0.28-2.i386.rpm
Preparing...           ########################################### [100%]
   1:tftp-server       ########################################### [100%]
```

Instating the server creates a configuration file: **/etc/xinetd.d/tftp**

```
# default: off
# description: The tftp server serves files using the trivial file transfer
#       protocol. The tftp protocol is often used to boot diskless
#       workstations, download configuration files to network-aware
#       printers, and to start the installation process for some operating
#       systems.
service tftp
{
        socket_type             = dgram
        protocol                = udp
        wait                    = yes
```

```
        user                    = root
        server                  = /usr/sbin/in.tftpd
        server_args             = -s /tftpboot
        disable                 = yes
        per_source              = 11
        cps                     = 100 2
}
```

3.2.1 server_args = -s /tftpboot

Indicates what is the tftp-server **s**ource directory that will be made available for the tftp clients. We have to create this directory manually and apply the proper permissions:

```
bash# mkdir /tftpboot
bash# chmod 755 /tftpboot
```

3.2.2 disable = yes

We need to change the **disable = yes** to **disable = no** then check if it was done:

```
bash# chkconfig tftp on
bash# chkconfig --list|grep tftp
   tftp:   on
```

3.3 Hiearchy of the directory: /tftpboot

```
/tftpboot/
   pxelinux.0
   initrd-everything.img
   initrd.img-6.0-sme
   vmlinuz
   vmlinuz-s-6.0-sme
   pxelinux.cfg/
      default
```

This root directory, that we have to create manually, need the same name as specified in the configuration file **/etc/xinetd.d/tftp**.

3.4 pxelinux.0

Go to http://www.kernel.org/pub/linux/utils/boot/syslinux/, and download the last version of <syslinux>. In our case, we downloaded **syslinux-2.06.tar.gz** in the **/tmp** directory. Untar it and copy **pxelinux.0** in the tftp-server root directory:

```
bash# cd /tmp
bash# tar -xvzf syslinux-2.06.tar.gz
bash# cd syslinux-2.06
bash# cp pxelinux.0 /tftpboot/
```

3.5 initrd-everything.img & vmlinuz

Those 2 files are on the SME CDROM under **/mnt/cdrom/images/pxeboot**. You have to copy those 2 files in the tftp-server directory and rename them if you have more than one version of the distribution. ***Make sure that the names you used is not ending with a ".0". The best is to end with an alphabetic character.*** The other restriction is that the names have to be the same as those used in **/tftpboot/pxelinux.cfg/default** (see below). There are 2 initrd files. One is a standard inird and the other one, **initrd-everything.img**, has support for all install methods and drivers supported for installation of SME. Since space and download speed is not important, it is better to use

the later one. Because we might have many versions in our tftp root directory, while we copy the files we also rename them to reflect the version we are installaing.

```
bash# mount /dev/cdrom
bash# cp /mnt/cdrom/images/pxeboot/initrd-everything.img /tftpboot/
initrd.img-6.0-sme
bash# cp /mnt/cdrom/images/pxeboot/vmlinuz /tftpboot/vmlinuz-6.0-sme
bash# eject
```

3.6 pxelinux.cfg directory

We have to create this **pxelinux.cfg** directory under the tftp-server root directory i.e. the directory that contains **pxelinux.0**. After the future SME server has used TFTP to get **pxelinux.0** it will look into the tftp root directory to find **pxelinux.cfg** subdirectory. This subdirectory has to be absolutely named **pxelinux.cfg**. The future SME server will then try to locate, in this subdirectory, a file corresponding to its MAC address and if it doesn't find it will take the file named **default**. We keep it simple and only use **default**.

3.7 default

This file is a kind of bootloader that looks like the standard **grub.conf** file. It contains parameters for the client station (the future SME sever) to be able to configure its boot:

```
label linux
  kernel vmlinuz-6.0-sme
  append ksdevice=eth0 \
         load_ramdisk=1 \
         initrd.img-6.0-sme \
         network ks=nfs:192.168.0.173:/PXE/SME/6.0/install.cfg
```

NOTE: The <\> at the end of the lines are used only for readability, don't put them in, all the append's parameters have to be on the same line.

3.7.1 kernel vmlinuz-6.0-sme

The name of the kernel file (vmlinuz) that we previously copied from the SME distribution CDROM. It is in the tftp-server root directory which also contains **pxelinux.0**. Avoid having a name ending with a digit (especially 0).

3.7.2 append

One of the main keyword of the bootloader file.

3.7.3 ksdevice=eth0

The client station (future SME sever) will go through **eth0** to get the kickstart file (install.cfg). If we want to use **eth1** we will have to modify **anaconda** and recompile it.

3.7.4 load_ramdisk=1

If we want our bootloader to default to the secondary operating system, we have to modify the default parameter to 1.

3.7.5 initrd=initrd.img-6.0-sme

The name of initrd file (initrd-everything.img) in the tftp-server root directory i.e. the file we copied from the SME distribution CDROM. As with **vmlinuz** avoid having a name ending with a digit (especially 0).

3.7.6 network ks=nfs:192.168.0.173:/PXE/SME/6.0/install.cfg

Specify that the installation will be done through **network**, the name of the kickstart "**ks**" file will be on the **nfs** server which have the IP address of **192.168.0.173** (our Red Hat server), it will be in the directory **/PXE/SME/6.0/** and its name will be **install.cfg**.

If we want to use a different name for the kickstart file we just have to change the name of **install.cfg** with another name as for example: **ks_install-6.0-sme.cfg**.

3.8 Testing TFTP

We go to a standard workstation, install the tftp client **tftp-0.28-2.i386.rpm**, cd to a test directory (/temp), tftp to our server with the -v (verbose) switch, tell the server to go to binary mode, try to download a file (in our case we tried to get **pxelinux.0** the "0" is zero), quit the tftp, and finally long list the directory:

```
bash# rpm -Uvh tftp-0.28-2.i386.rpm
Preparing...           ########################################### [100%]
   1:tftp              ########################################### [100%]
bash# cd /temp
bash# /usr/bin/tftp -v 192.168.0.173
Connected to 192.168.0.173 (192.168.0.173), port 69
tftp> binary
mode set to octet
tftp> get pxelinux.0
getting from 192.168.0.173:pxelinux.0 to pxelinux.0 [octet]
Received 11304 bytes in 0.0 seconds [inf bits/sec]
tftp> quit
[root@via700 pxe]# ll
total 12
-rw-r--r--    1 root     root        11304 Jan 24 22:30 pxelinux.0
```

4.0 DHCP

4.1 The DHCP server: dhcp-2.0p15.8.i386.rpm

Download dhcp-2.0p15.8.i386.rpm and install it on our Red Hat 7.3 server:

```
bash# rpm -Uvh dhcp-2.0p15.8.i386.rpm
```

This will install the dhcp server and create a few configuration files.

4.2 The DHCP server configuration file: /etc/dhcpd.conf

```
option domain-name     "micronator.dyndns.org";
          # An arbitrary domain name.
allow bootp;
          # The bootp flag is used to tell dept. whether or not to
          # respond to bootp queries.
          # Bootp queries are allowed by default.
allow booting;
          # The booting flag is used to tell dept. whether or not to
          # respond to queries from a particular client.   This
          # keyword only has meaning when it appears in a host declare-
          # tion. By default, booting is allowed, but if it is disabled
          # for a particular client, then that client will not be able to
          # get and address from the DHCP server.
subnet 192.168.0.0 netmask 255.255.255.0 {
  range 192.168.0.125 192.168.0.130;
          # The IP segment and the range of IP addresses that the server
          # will give in response to a dhcp query.
          # Note: We take a small range only.
```

```
  next-server 192.168.0.173;
          # The name of your TFTP server.
          # The next-server statement is used to specify the host address
          # of the server from which the initial boot file (specified in
          # the filename statement) is to be loaded. Server-name should
          # be a numeric IP address or a domain name. If no next-server
          # parameter applies to a given client, the DHCP server's IP
          # address is used.
  filename "pxelinux.0";
          # Name of the bootloader program in the tftp-server root
          # directory i.e. <192.168.0.173:/tftpboot/pxelinux.0>
          # The filename statement can be used to specify the name of the
          # initial boot file which is to be loaded by a client. The
          # filename should be a filename recognizable to what-ever file
          # transfer protocol the client can be expected to use to load
          # the file.
}         # END of subnet.
```

4.2.1 option domain-name "micronator.dyndns.org"

This is our domain name. Choose anything you like.

4.2.2 subnet 192.168.0.0 netmask 255.255.255.0 {

Replace with your subnet and mask for your particular LAN. Don't forget the "**{**" at the end of the line

4.2.3 range 192.168.0.125 192.168.0.130;

The first DHCP client will receive the IP 192.168.0.125 and the last one 192.168.0.130.

Please remember that you can have only one DHCP server per LAN segment. If you cannot install one on your segment because there is already one, then you can just connect the Red Hat server and the future SME server directly with a crossover RJ-45 cable.

4.2.4 next-server 192.168.0.173;

The is the IP address of our TFTP server i.e. Red Hat 7.3

4.2.5 filename "pxelinux.0";

This is the file that the future SME server will try to get using TFTP. It is some kind of "MBR" like file that tells the PXE client how to get the real boot loader configuration file i.e. /tftpboot/pxelinux.cfg/default.

4.3 The DHCP client lease database: /var/lib/dhcp/dhcpd.leases

When DHCP is first installed, there is no lease database. However, dhcpd requires that a lease database be present before it will start. To make the initial lease database, just create an empty file called **/var/lib/dhcp/dhcpd.leases**.

```
bash# touch /var/lib/dhcp/dhcpd.leases
```

This file will contains some data about the IP addresses given after clients queries. It is a good place to look to see if the dhcpd daemon is working properly. Another place to look for errors if the client didn't receive an address is **/var/log/messages**. It will tell you what is going on with the DHCP.

4.4 /etc/hosts

While doing the installation the server will try to resolve machine name. If it cannot do it,

it will stop with an error message. Since in our case the IP segment has only two stations, our portable and the future SME server, the best way to solve this problem is to put the name of the client station along with its IP address in the TFTP server's **/etc/hosts** file. If you are doing more than one client at the same time, put all the client IP and addresses in this hosts file. The dhcp-server will always give name starting with pc-xxxx. Where xxxx correspond to the IP address it gives to the client.

```
# Do not remove the following line, or various programs
# that require network functionality will fail.
127.0.0.1       dorgee.micronator.dyndns.org   pc-0173   dorgee   localhost

192.168.0.173   pc-0173   # NFS, TFTP, & DHCP server (our RHL 7.3)

192.168.0.125   pc-0125   # our first  client using PXE
192.168.0.126   pc-0126   # our second client using PXE
192.168.0.127   pc-0127   # our third  client using PXE
192.168.0.128   pc-0128   # our fourth client using PXE
192.168.0.129   pc-0129   # our fifth  client using PXE
192.168.0.130   pc-0130   # our sixth  client using PXE
```

4.5 Starting the DHCP daemon

At the DHCP server console just type the following command:

```
bash# service dhcpd start
Starting dhcpd:              [  OK  ]
```

Verify that the daemon is running:

```
bash# service dhcpd status
dhcpd (pid 1379) is running...
```

4.6 The DHCP client: dhcpcd-1.3.22pl1-7.i386.rpm

This RPM **dhcpcd-1.3.22pl1-7.i386**. is the client RPM as denoted by the **c** in the name. Install it on a test station on the same ethernet segment as the DHCP server to verify that your server is working properly. Copy the DHCP client RPM and install it by typing the following:

```
bash# rpm -Uvh dhcpcd-1.3.22pl1-7.i386.rpm
Preparing...           ########################################### [100%]
   1:dhcpcd            ########################################### [100%]
```

Make a backup copy of the file **/etc/sysconfig/network-scripts/ifcfg-eth0** on that same test station:

```
bash# cp /etc/sysconfig/network-scripts/ifcfg-eth0 /etc/sysconfig/network-
scripts/ifcfg-eth0.original
```

Make sure that the file **/etc/sysconfig/network-scripts/ifcfg-eth0** contains only the following:

```
DEVICE=eth0
BOOTPROTO=dhcp
ONBOOT=yes
```

We need to restart the network on the test station for the new configuration of **eth0** to apply:

```
bash# service network restart
Shutting down interface eth0:                [  OK  ]
Shutting down loopback interface:            [  OK  ]
Setting network parameters:                  [  OK  ]
Bringing up loopback interface:              [  OK  ]
```

```
Bringing up interface eth0:                [  OK  ]
```

The last line means that everything is working fine. Another way to check that the dhcp-server is running as it should is to first kill the IP we just received then ask for a new IP address:

```
bash# dhcpcd -k
bash# dhcpcd -n
```

Now we check if we received an address:

```
bash# ifconfig eth0
eth0 Link encap:Ethernet  HWaddr 00:40:63:CC:40:15
     inet addr:192.168.0.125  Bcast:192.168.0.255  Mask:255.255.255.0
     UP BROADCAST NOTRAILERS RUNNING MULTICAST  MTU:1500  Metric:1
     RX packets:3142 errors:0 dropped:0 overruns:0 frame:0
     TX packets:2102 errors:4 dropped:0 overruns:0 carrier:2
     collisions:795 txqueuelen:100
     RX bytes:3763618 (3.5 Mb)  TX bytes:170336 (166.3 Kb)
     Interrupt:10 Base address:0xe800
```

If you don't have this after typing the ifconfig command then recheck your setup it should work. When satisfied don't forget to put back on the test station the original **`/etc/sysconfig/network-scripts/ifcfg-eth0`**.

```
bash# cp /etc/sysconfig/network-scripts/ifcfg-eth0.original /etc/sysconfig/
network-scripts/ifcfg-eth0
```

5.0 The PXE directory containing the distribution files

Copy the complete CDs of the SME distribution on the server in **`/PXE/SME/6.0/`** directory.

```
bash# mkdir -p /PXE/SME/6.0
bash# mount /dev/cdrom
bash# cp -R /mnt/cdrom/* /PXE/SME/6.0
```

6.0 NFS

6.1 nfs-utils-0.3.3-6.73.i386.rpm & portmap-4.0-41.i386.rpm

Download **`nfs-utils-0.3.3-6.73.i386.rpm`** & **`portmap-4.0-41.i386.rpm`** or newer ones and install both RPM starting by portmap then nfs-utils:

```
bash# rpm -Uvh portmap-4.0-41.i386.rpm
Preparing...     ##################################################
portmap          ##################################################
bash# rpm -Uvh nfs-utils-0.3.3-6.73.i386.rpm
Preparing...     ##################################################
nfs-utils        ##################################################
```

This will install all the necessary files for the NFS server and create a few configuration files.

6.2 NFS configuration files: /etc/exports

Modify the /etc/exports file to look like the following:

```
# 29 décembre 2003
# We share: </PXE/SME/6.0>
/PXE/SME/6.0   192.168.0.0/255.255.255.0(ro,no_root_squash)
```

6.2.1 /PXE/SME/6.0

This is the directory we want to share under NFS.

6.2.2 192.168.0.0/255.255.255.0

We want to share with everybody on the IP segment **192.168.0.0** using the mask of **255.255.255.0**. Note that there is no space between the last <...255.0> and the next <(ro,...>. If you put one, it doesn't means the same thing at all, so beware.

6.2.3 ro

We want to share the directory in read only mode.

6.2.4 no_root_squash

When the user <root> will use the NFS server, it will do so under its real user name and power of **root** and will not be switch to user **nobody** when connecting.

6.3 Exporting the directory

To export the directory we have to type the following command:

```
bash# exportfs -rav
exporting 192.168.0.0/255.255.255.0:/PXE/SME/6.0
```

We want to see if the directory is exported:

```
bash# exportfs -v
/PXE/SME/6.0     192.168.0.0/255.255.255.0(ro,async,wdelay,no_root_squash)
```

Everything looks ok. The directory **/PXE/SME/6.0** is exported. Everybody on IP segment **192.168.0.0** with the mask of **255.255.255**. can see it. The directory is in read only mode and user <root> will be **root**. To let in user <root> as **root** is a security risk. Be aware.

6.4 Starting the NFS daemon

First we check the status of the NFS daemon:

```
bash# service nfs status
rpc.mountd is stopped
nfsd is stopped
rpc.rquotad is stopped
```

Then we start the NFS daemon:

```
bash# service nfs start
Starting NFS services:                                     [  OK  ]
Starting NFS quotas:                                       [  OK  ]
Starting NFS mountd:                                       [  OK  ]
Starting NFS daemon:                                       [  OK  ]
```

Again we check the status of the NFS daemon:

```
bash# service nfs status
rpc.mountd (pid 5633) is running...
nfsd (pid 5642 5641 5638) is running...
rpc.rquotad (pid 5628) is running...
```

If we modify **/etc/exports**, we have to export it again using **exportfs -rav**. Please re-export every time you modify something.

6.5 Testing NFS from a client side

We can check the NFS server using a standard station. Just make sure the station has the nfs-utils RPM installed. If it doesn't install the RPM.

```
bash# rpm -qa | grep nfs-utils
nfs-utils-0.3.3-6.73
```

Create the mount point on that station to receive the mount:

```
bash# mkdir /mnt/PXE
```

6.6 The mount command

Now we mount the exported directory. Always login as <**root**> to mount something.

NOTE: The <\> at the end of the 2 first line is used only for readability, don't put them on the command line. Also, the command and all the parameters have to be on the same line.

```
bash# mount -t nfs -o soft,bg,timeo=120 \
            192.168.0.173:/PXE/SME/6.0 \
            /mnt/PXE
```

6.6.1 -t nfs

The argument following the -t is used to indicate the file system type.

6.6.2 -o

Options are specified with a -o flag followed by a comma separated string of options.

6.6.3 soft

This option allows the kernel to time out if the nfs server is not responding for some time. The time can be specified with timeo=time. This option might be useful if your nfs server sometimes doesn't respond or will be rebooted while some process tries to get a file from the server. If it takes too long it means you have a problem in the command line itself or something is wrong on the server side.

6.6.4 bg

If the first NFS mount attempt times out, retry the mount in the background. After a mount operation is backgrounded, all subsequent mounts on the same NFS server will be backgrounded immediately, without first attempting the mount. A missing mount point is treated as a timeout, to allow for nested NFS mounts.

6.6.5 timeo=120

This option allows the kernel to time out if the nfs server is not responding for some time. Expressed in seconds

6.6.6 /mnt/PXE

Our mount point on the machine we are using.

Now we check if the directory is mounted on our standard station:

```
bash# df -h

Filesystem            Size  Used Avail Use% Mounted on
/dev/hda3              28G  491M   25G   2% /
/dev/hda1             197M   14M  173M   8% /boot
```

```
none                  121M     0  120M   0% /dev/shm
192.168.0.173:/PXE/SME/6.0     32G   25G  5.1G  83% /PXE
```

6.6.7 192.168.0.173:/PXE/SME/6.0

Here is the exported directry **/PXE/SME/6.0** from our NFS server **192.168.0.173**.

Now we check inside the mounted directory:

```
[root@via700 etc]# ll /PXE/SME/6.0/
total 44
dr-xr-xr-x    4 root     root         4096 Jul 21  2003 dosutils
dr-xr-xr-x    4 root     root         4096 Jul 21  2003 e-smith
dr-xr-xr-x    8 root     root         4096 Feb  7 15:33 images
-rwxr-xr-x    1 root     root         2181 Dec 26 03:49 install.cfg
-rwxr-xr-x    1 root     root         2181 Feb  2 20:53 ks.cfg
-r--r--r--    1 root     root        20249 Dec 11 10:19 LICENSE.txt
-r--r--r--    1 root     root          200 Dec 11 10:48 TRANS.TBL
```

We can even see the standard kickstart file **install.cfg**.

To see some satistics we can always use the command **/usr/sbin/nfsstat** on the server. A good place to look for error is **/var/log/messages**. If something goes wrong this is the first place to look.

Total victory on the NFS front.

6.6.8 Troubleshooting

If you get a message like: **mount: 192.168.0.173:/PXE/SME/6.0 failed, reason given by server: Permission denied**. then have a look at /var/log/messages:

```
bash# tail /var/log/messages

Dec 28 16:38:00 dorgee nfs: Starting NFS services:  succeeded
Dec 28 16:38:01 dorgee nfs: rpc.rquotad startup succeeded
Dec 28 16:38:01 dorgee nfs: rpc.nfsd startup succeeded
Dec 28 16:38:01 dorgee nfs: rpc.mountd startup succeeded
Dec 28 16:38:46 dorgee rpc.mountd: refused mount request from 192.168.0.125
(pc-0125) for /PXE/SME/6.0 (/PXE/SME/6.0): no DNS forward lookup
```

Since we don't have DNS services on our network, we modify **/etc/host** on the Red Hat server to have an entry in it for the test station (192.168.0.125).

```
bash# cat /etc/hosts

# Do not remove the following line, or various programs
# that require network functionality will fail.

127.0.0.1       dorgee.micronator.dyndns.org   pc-0173   dorgee   localhost

192.168.0.173   pc-0173   # NFS, TFTP, & DHCP server (our RHL 7.3)

192.168.0.125   pc-0125   # our first client using PXE
```

7.0 INSTALL.CFG

```
#-----------------------------------------------------------------------
# kickstart file for SME Server V5
# Copyright (c) 2001 Mitel Networks Corporation
#-----------------------------------------------------------------------
# NOTE: The only supported customizations to this file are:
```

```
#         keyboard
#         timezone
# To modify the default settings for these parameters, please refer
# to the HOWTO document on http://www.e-smith.org/docs/howto/
#-----------------------------------------------------------------------
#
#     - Michel-Andre Robillard 11 octobre 2003 @ 19:36:
#     - Voir insatll.cfg.original pour l'original
#
#-----------------------------------------------------------------------

#System language
lang en_US

#Language modules to install
langsupport en_US fr_CA

#System keyboard
keyboard us

#System mouse
#####mouse none
mouse --emulthree genericps/2

#System timezone
#timezone --utc America/New_York
timezone --utc America/Montreal

#Root password
#rootpw --iscrypted $1$GÅûÓøÔèê$jWK/s8WgbjQJRyQUS3B0d1
#Root password   "password"
rootpw --iscrypted $1$X4zN/vO1$I8.vXRAo2gh9GqomUHgM1.

#Reboot after installation
#reboot

#Use text mode install
#text

#System bootloader configuration
bootloader --useLilo --linear --location=mbr

#instead or upgrade
install

#Install method
#cdrom
#url --url http://allspice/development/files/projects/e-smith-releases/
6.0alpha5+SL/cdrom.image/

#####nfs --server 192.168.0.173 --dir /PXE/MITEL/6.3-b/
nfs --server 192.168.0.173 --dir /PXE/SME/6.0/

#Clear the Master Boot Record
zerombr yes

#Clear all partitions from the disk
clearpart --all --initlabel

#Disk partitioning information
part  swap --size 256 --ondisk 0
part /boot --fstype ext3 --size 200 --ondisk 0
part / --fstype ext3 --size 200 --grow --ondisk 0

#Use DHCP networking
network --bootproto dhcp

#System authorization information
auth  --useshadow  --enablemd5

#Firewall configuration
firewall --disabled
```

```
#Do not configure the X Window System
skipx

%packages

%packages
@ Base

%post

# set up initial configuration
/sbin/e-smith/signal-event post-install

# add comment to /etc/motd
echo "Welcome to the Mitel Networks SME Server." > /etc/motd
```

8.0 TROUBLESHOOTING

The worse problem is from IP address and name resolution. You will notice that the name DHCP gives to the station is sometime **`pc-1234`** and sometime **`pc-12345`**. **`pc-1234`** before the installation and **`pc-12345`** after the installation. In the **`/etc/hosts`** file we always use p**`c-1234`**. The alias for the name of our server also follows the same rule i.e. **`pc-1234`**. The best way to see this is by looking at the **`/var/log/messages`** files that will give you the most verbose hints for the resolution of the problem.

After trying PXE on different hardwares I noticed that sometime the on-board NIC doesn't seems to work with PXE but the add-on PCI NIC card has no problem to use PXE. So I tried to use the add-on PCI NIC.

The add-on PCI NIC after receiving the IP will connect to the PXE server for the TFTP transfer. The station will receive the initrd and the vmlinuz files and load them. At that moment the kernel will re-initialized the network.

The next step is to get the kickstart file **`install.cfg`**. The station tries to reconnect to the PXE server. I suspect that it is trying to resolve the station and server names because it tries two time to resolve names. Eventually it will time out on both tentatives and then complains that it cannot find our kicstart file **`install.cfg`** that he now names **`ks.cfg`**.

This particular problem is found maily with motherboard with on board NIC with PXE DISPLAY MESSAGE turned off so you just see the displayed message of the second card i.e. the add-on PCI NIC which of course you set to boot using PXE.

After receiving its IP address from the DHCP the add-on PCI NIC will try to find the boot file in the /tftpboot/pxelinux.0 directory. If its MAC adddress is 00:04:3C:AB:F1:F4 it will first look for the 01-00-04-3c-ab-f1-f4. Notice the leading 01 don't ask me why. You will see this file name appearing right after the station received its IP address. If this file is not found it will look for a file corresponding to its received IP address in some kind of hexadecimal form. After every unsuccessful attempt it will take out a digit at a time from its IP address and look again for a corresponding file. Finally if still without success it will take the default file which name is **`default`**. In our case we only use the **`default`** name for the boot configuration file. All of the above takes only one or two seconds.

. Our future SME server will get the **`default`** boot file, open it and start processing it. It

will load the kernel which will initialize all the NIC and of course the on-board one will be **`eth0`** and the one we just used to connect to the PXE server will be eth1, the add-on PCI NIC. There are two solutions for our problem.

First solution: turn off the on-board NIC but still tell the BIOS to boot from the LAN and we will use our PCI NIC to connect to the PXE server. After installation, when the SME will reboot for the first time, we will go back to the BIOS and enable the on-board NIC and take out the **`boot from LAN`**.

Second solution: find the way to show the PXE display for the on-board NIC and use it for connecting to the PXE server. You motherboard manual should explain how to enable the PXE display.

If at that moment you tail the /var/log/messages on the PXE server you will notice the MAC address of the station to be different than the one displayed when at the beginning of the PXE loading it is looking for the