

GPXE

Network Booting
for the 21st Century

Matthew Walster
matthew.walster@ixreach.com

UKNOF18
2011 / 01 / 20

Network Booting

Traditionally, ISPs and businesses have used PXE:

- Integrated into many motherboard BIOS chips
- Pulls a Network Bootstrap Program via TFTP
- Often used in conjunction with PXELINUX
- Requires modifying DHCP config (Option 60/43)
- TFTP doesn't work through NAT, cheap firewalls etc

Enter GPXE...

GPXE brings a whole host of new features:

- Media booting (Floppy, ISO, GRUB, ELF etc)
- Additional protocol support
 - NFS, HTTP, FTP, iSCSI, ATAoE, (HTTPS)
- WiFi support - if driver available
- Easy scriptable interface for boot-time configuration
- A "curses-like" configuration interface
- Able to boot MS Windows (XP via AoE, 2003 via iSCSI)

n.b. HTTPS isn't "secure" at present, due to lack of boot-time entropy source, but present as "proof-of-concept".

```
QEMU BIOS - build: 02/08/07
$Revision: 1.174 $ $Date: 2006/10/17 16:48:05 $
Options: apmbios pcibios eltorito rombios32

ata1 master: QEMU CD-ROM ATAPI-4 CD-Rom/DVD-Rom

Booting from Floppy...
Loading ROM image.....

gPXE 0.5.1 -- Open Source Boot Firmware -- http://etherboot.org
Features: HTTP HTTPS DNS TFTP iSCSI AoE bzImage Multiboot PXE
net0: 52:54:00:12:34:56 on PCI00:03.0 (open) TX:0 TXE:0 RX:0 RXE:0
DHCP (net0 52:54:00:12:34:56)... ok
net0: 10.254.254.31/255.255.255.0 gw 10.254.254.2
Booting from filename "http://chipmunk.tuntap/images/uniboot/uniboot.php"
http://chipmunk.tuntap/images/uniboot/uniboot.php... ok
/kernels/vmlinuz-2.6.17-14mdv... ok
initrd.img... ok
/kernelmods/2.6.17-14mdv/kernel/drivers/net/mii.ko.gz... ok
/kernelmods/2.6.17-14mdv/kernel/drivers/net/8139too.ko.gz... ok
/kernelmods/2.6.17-14mdv/kernel/net/packet/af_packet.ko.gz... ok
http://rom.etherboot.org/share/mcb30/bootsplash-800x600... _
```

GPXE in action

gPXE option configuration console

```
priority..... <not specified>  
root-path..... iscsi:10.0.0.6::::iqn.2006-05.home.dolphin:win2k3_  
password..... <not specified>  
username..... <not specified>  
hostname..... <not specified>  
ip..... 10.0.0.12
```

root-path (Text string) - NFS/iSCSI root path

Ctrl-S - save configuration

Configuring GPXE at boot-time

Enter configuration ID to boot (eg, 3005)



GPXE asking a boot-time question

Username :

dotwaffle_____

Password :

*****_____

GPXE requesting authentication

Getting started

Try out <http://netboot.me> - a nice example of a network boot:

- Boots from Floppy/CD/USB, NIC ROM or via PXE
 - Regular PXE image, requires TFTP server, then chainloads GPXE
- Contacts <http://www.netboot.me/menu.gpxe>:
 #!gpxe
 chain menu.c32 premenu.cfg
- Standard SYSLINUX menu allowing you to boot one of:
 - Pre-configured Linux/BSD installers or utilities
 - Your own kernels

"PXE forces me to do x though..."

Traditionally, you had to select a kernel, then an initrd etc.

```
#!gpxe
imgfree
kernel -n img http://archive.ubuntu.com/ubuntu/dists/lucid/main/
    installer-i386/current/images/netboot/ubuntu-installer/i386/linux
    vga=normal -- quiet
initrd http://archive.ubuntu.com/ubuntu/dists/lucid/main/
    installer-i386/current/images/netboot/ubuntu-installer/i386/initrd.gz
boot img
```

Just use memdisk-iso!

```
#!gpxe
imgfree
kernel -n img http://static.netboot.me/memdisk-iso iso
initrd -n img http://archive.ubuntu.com/ubuntu/dists/lucid/main/
    installer-i386/current/images/netboot/mini.iso
boot img
```

"Thin Clients" in seconds

```
#!gpxe
imgfree
kernel -n img http://boot.vpn/ltsp/vmlinuz ro quiet splash
        nbdroot=172.16.0.24:2000
initrd http://boot.vpn/ltsp/initrd.img
boot img
```

At home

- ADSL line (5Mbit) with Be
- *TWO* layers of NAT


Server in Netherlands

- 100Mbit connection
- 4GB RAM, typically 5 simultaneous users



ubuntu

Username

 Preferences

Itsp15 (10.0.2.15) // 11:31

Less than 60 seconds later...

What's so amazing about using HTTP?

TFTP is a file transfer protocol - you request a file, it sends it.

HTTP can be scripted, so performs based on:

- the IP address of the client
- the MAC address of the client (if sent in the request)
- the date/time of day
- whether that client has been seen before/authorised

Hence, you can set everyone to boot the same URL, and set your backend to make sure the next time a particular machine reboots, it gets re-imaged with the latest software.

"Customising the boot process is hard"

If you want to create a custom GPXE image, you could alter the source files, or you could just use the handy generator at:

- www.rom-o-matic.net

You can choose:

- Which NIC drivers to include (or all of them)
- Whether to output over a serial port
- Include iSCSI boot user/pass (or burn into ROM)
- Whether to include WEP/WPA/WPA2 support
- To embed a script, to remove DHCP dependency

GPXE sounds great, why not GPXE?

- Development is *slow* - so much that a fork has been made
 - <http://ipxe.org> (includes FCoE and Infiniband)
- Very limited available RAM (1MB)
 - TCP Window Size is only 4kB
 - Netbooting across the Atlantic can take up to 5 mins
- DHCP is inherently insecure
 - Though you can fix that with physical security

Where to go next...

- <http://etherboot.org/>
 - The GPXE website
- <http://rom-o-matic.net/>
 - Build your own GPXE ROMs
- <http://netboot.me/>
 - A community Network Booting site (configurable)
- <http://boot.kernel.org/>
 - Another Network Booting site (with iSCSI)
- <http://ipxe.org/>
 - The GPXE fork "bleeding edge"

Questions?

Matthew Walster
matthew.walster@ixreach.com