Preface

This document mainly focuses on how to configure Raspberry Pi 3's and 4's to boot over the network without the need for an SD card. If use of an SD card is not an issue, a (minimal) Plan 9 kernel can be loaded from the SD card, which will fetch the full Plan 9 kernel via tftp and copy it to /dev/reboot. This method works even for older pi models (pi1 and pi2).

Most of the critical information in this document come from various postings by Richard Miller to 9fans and direct emails from him. Richard's contrib repository on 9p.io is the authoritative source for Plan 9 for the Raspberry Pi (Zero, through 4).

Configure RPi 2 for PXE Boot

The configuration described for Raspberry Pi 3B/3B+ also works for configuring the Raspberry Pi 2, with one exception. Whereas the Raspberry Pi 3 will not require an SD card once it is configured, the Raspberry Pi 2 requires an SD card with the **bootcode.bin** file, but nothing else. *However, see the note above about using a minimal Plan 9 kernel instead of bootcode.bin*.

Configure RPi 3B for PXE Boot

To configure a RPi 3B for PXE booting (i.e. without an SD card), follow the instructions in the **Client Configuration** section given in the following: (*Note: **RPi 3B+ are already configured***)

https://www.raspberrypi.org/documentation/hardware/raspberrypi/bootmodes/net_tutorial.md

After completing the steps, but before shutting the device down, execute "**`cat /proc/cpuinfo`**" and record the *serial number* of the device. It is also a good time to record the ethernet MAC address of the device.

Configure RPi 4 for PXE Boot

The following link shows how to enable PXE booting for RPi 4. Note that the instructions direct you to copy the EEPROM image from `/lib/firmware/raspberrypi/bootloader/` to a location (e.g. `pieeprom.bin`). There are several versions of the firmware. :

https://www.raspberrypi.org/documentation/hardware/raspberrypi/bcm2711_bootloader_config.md

After completing the steps, be sure to record the serial number by either looking in `/proc/cpuinfo` or `/sys/firmware/devicetree/base/serial-number`

Plan 9 Server Setup

Plan 9 servers (file server, dhcpd/tftpd environment) will serve all the required content. You'll need to add configuration data for the RPi to /lib/ndb/local and /lib/tftpd. Verify that you are starting `ip/dhcpd` and `ip/tftpd` for the cpurc of the server (usually in `/cfg/<cpuname>/cpurc`)

Plan 9 DHCPD Modifications

If you only use RPi4's, you can ignore the modifications described here and instead set the value of the TFTP_IP configuration variable in Pi EEPROM to the IPv4 address of the TFTP server.

Plan 9 DHCP server (`dhcpd`) doesn't handle general configuration/sending of Vendor Option 43. This DHCP reply option is required for RPi3 and RPi4. Additionally, RPi4 asks for the "TFTP Server Name" in the Parameter Request List (Option 66). This patch allows hdcpd to detect RPi 3 and Rpi 4 DHCP requests and provide the correct information in the Offer reply to satisfy RPi's.

Apply `/n/sources/patch/rpi-netboot-dhcpd` patch from 9p.io, or manually apply the diffs noted in this link to `/sys/src/cmd/ip/dhcpd/dhcpd.c`:

https://gist.github.com/9nut/5a3829356b6d0ce3ae47e2f390df7442

Then "`mk install`" in `/sys/src/cmd/ip/dhcpd` directory.

Place the Newest Raspberry Pi Firmware in /lib/tftpd

Be sure to get the latest versions of the following files and place them in the TFTP root directory (/lib/tftpd); previous versions are known to have problems with PXE loading:

- `bootcode.bin`
- `start.elf`
- `fixup.dat`
- `start4.elf`
- `fixup4.dat`
- `start_cd.elf`
- `fixup_cd.dat`
- `start4cd.elf`
- `fixup4cd.dat`
- `bcm2708-rpi-0-w.dtb`
- `bcm2708-rpi-b-plus.dtb`
- `bcm2708-rpi-b.dtb`
- `bcm2708-rpi-cm.dtb`
- `bcm2709-rpi-2-b.dtb`

- `bcm2710-rpi-3-b-plus.dtb`
- `bcm2710-rpi-3-b.dtb`
- `bcm2710-rpi-cm3.dtb`
- `bcm2711-rpi-4-b.dtb`

Note: if you intend to only PXE boot Plan 9 on RPi3, you will not need start.elf and fixup.dat files. Plan 9 doesn't use the GPU, and the "cut down" versions (start_cd.elf, fixup_cd.dat) are sufficient. When Plan 9 boots on RPi, it will ask for the appropriate Device Tree Blob depending on the hardware model.

The latest RPi firmware are found here:
https://github.com/raspberrypi/firmware/tree/master/boot

Place the Newest Non-free Firmware in /sys/lib/firmware

Copy all nonfree firmware from `https://github.com/RPi-Distro/firmware-nonfree/raw/master/brcm` to `/sys/lib/firmware`. For example, run the following script and copy and paste the resulting script to rc (i.e. run the output script):

```
for (i in (brcmfmac43430-sdio.bin brcmfmac43430-sdio.txt brcmfmac43455-sdio.bin
brcmfmac43455-sdio.clm_blob brcmfmac43455-sdio.txt)) {
	echo 'hget https://github.com/RPi-Distro/firmware-nonfree/raw/master/brcm/'^$i '>
'^/sys/lib/firmware/$i
}
```

Build and Install Plan 9 for RPI (BCM)

Copy the Plan 9 Pi kernel sources from `/n/sources/contrib/miller/9/bcm` to `/sys/src/9/bcm`. Build one of the PI kernels; for example run 'mk' to build the pi2 kernel. Copy the output (e.g. **9pi2**) to **/lib/tftpd/arm** directory.

Add an Entry to NDB

Per usual, you'll need to create a DNS entry for the new machine in /lib/ndb/local. You'll need the minimal set of attributes: ip, ether, sys and dom. Use the ether (MAC address) that was recorded in the first step. For example:

```
# Raspberry Pi model 3 terminal; pxe boot capable.
ip=172.16.1.58 ether=cafedeadc0de sys=piterm
	dom=piterm.example.com
```

Configure TFTPD Root Content

The state of RPi netbooting a bit of a mess. The structure and location of configuration files differ between what RPi 2 and RPi 3 need and what RPi 4 needs.

Configuration for RPi 2 and 3

RPi2s and 3s expect to find the following files in the TFTPD root directory (/lib/tftpd):

- `config.txt`
- `bootcode.bin`
- `start.elf`
- `fixup.dat`
- `start_cd.elf`
- `fixup_cd.dat`
- `bcm2708-rpi-0-w.dtb`
- `bcm2708-rpi-b-plus.dtb`
- `bcm2708-rpi-b.dtb`
- `bcm2708-rpi-cm.dtb`
- `bcm2709-rpi-2-b.dtb`
- `bcm2710-rpi-3-b-plus.dtb`
- `bcm2710-rpi-3-b.dtb`
- `bcm2710-rpi-cm3.dtb`

Even though there can be multiple RPi 3 devices, there is only one **config.txt** in `/lib/tftpd`; this is how bootcode.bin expects things.

Because a single config.txt contains the configuration for multiple RPi3's, it uses the device serial number (8 digit, hex) as *Conditional Filters* to create sections for each device; see Conditional Filters description here:

https://www.raspberrypi.org/documentation/configuration/config-txt/README.md

For example, if there is an RPi 3 with serial number `000000baadc0de`, then **config.txt** file will look something like this:

```
[0xbaadc0de]
start_file=start_cd.elf
fixup_file=fixup_cd.dat
cmdline=baadc0de/cmdline.txt
kernel=arm/9pi2
gpu_mem=16
hdmi_group=2
hdmi_mode=82
core_freq=250
```

The content of the config.txt file above gives a hint about the directory hierarchy in /lib/tftpd. Note that values for **kernel** and **cmdline** include relative paths of each file. This will let us customize the boot parameters for each device, as needed.

In this case, we created a directory for arm kernel binaries and a directory based on the serial number of the device where cmdline.txt is stored. For this installation, the layout looks like this:

```
/lib/tftpd
├── arm
│   └── 9pi2
├── bootcode.bin
├── config.txt
├── baadc0de
│   └── cmdline.txt
├── fixup_cd.dat
├── fixup.dat
├── start_cd.elf
└── start.elf
```

Configuration for RPi 4 and RPi 400

RPi 4 network boot expects its configuration files to be in a directory named by its serial number (i.e. <serialnumber>/ directory, relative to /lib/tftpd). This requires that each RPi 4 have a directory `/lib/tftpd/<serialnumber>/` that contains its **config.txt** and its **cmdline.txt** files.

It is important to explicitly ignore loading of Device Tree Blob (i.e. 'device_tree=') and to explicitly enable the Generic Interrupt Controller (i.e. 'enable_gic=1') in config.txt.

The following files can be placed anywhere in the /lib/tftpd hierarchy so long as their relative path is given in the appropriate entry.

- `start4.elf`
- `fixup4.dat`
- `start4cd.elf`
- `fixup4cd.dat`

For example, for an RPi 4 with serial number 0xbaadc0de, there must be a directory `/lib/tftpd/baadc0de` that contains config.txt that looks like:

```
start_file=../start4cd.elf
fixup_file=../fixup4cd.dat
cmdline=cmdline.txt
```

```
kernel=../arm/9pi4
gpu_mem=16
hdmi_group=2
hdmi_mode=82
core_freq=250
enable_gic=1
device_tree=
```

This is an example of what `/lib/tftpd` might look like if you're netbooting an RPi3B+ with serial number **baadbaad** and an RPi4 with serial number **baadc0de**:

```
/lib/tftpd
├── baadc0de
│   ├── bcm2711-rpi-4-b.dtb
│   ├── cmdline.txt
│   └── config.txt
├── baadbaad
│   └── cmdline.txt
├── arm
│   ├── 9pi2
│   └── 9pi4
├── bcm2710-rpi-3-b-plus.dtb
├── bcm2711-rpi-4-b.dtb
├── bootcode.bin
├── config.txt
├── fixup4cd.dat
├── fixup4.dat
├── fixup_cd.dat
├── fixup.dat
├── LICENSE.broadcom
├── start4cd.elf
├── start4.elf
├── start_cd.elf
└── start.elf
```

Booting

With the above steps completed, power the RPi 3 and monitor /sys/log/ipboot to trace DHCP discovery and offer messages and subsequent tftp downloads of the files. It will look something like this:

```
supermic Nov 27 14:58:32 Discover(0.0.0.0->172.16.1.47)
xid(26f30339)flag(0)ea(cafedeadc0de)id(hwa01_cafedeadc0de)need(vendorinfo vendorclass bf 128 129
130 131 132 133 134 135 tftpserver ) broadcast
```

```
supermic Nov 27 14:58:32 Offer(172.16.1.47-172.16.1.58supermic Nov 27 14:58:32
Discover(0.0.0.0->172.16.1.47)
xid(26f30339)flag(0)ea(cafedeadc0de)id(hwa01_cafedeadc0de)need(vendorinfo vendorclass bf 128 129
130 131 132 133 134 135 tftpserver ) broadcast
)id(hwa01_cafedeadc0de)ci(0.0.0.0)gi(0.0.0.0)yi(172.16.1.58)si(172.16.1.47)
typ(2)leas(1800)sid(172.16.1.47)mask(/120)router(172.16.1.1)host(piterm.9netics.com)vendorinfo(Ra
spberry Pi Boot   )vendorinfo(43 20 82 97 115 112 98 101 114 114 121 32 80 105 32 66 111 111 116
32 32 32)
supermic Nov 27 14:58:34 tftp 104108 connection from 172.16.1.58!49152 dir /net/udp/3
supermic Nov 27 14:58:34 tftpd 104108 send file 'bootcode.bin' octet to 172.16.1.58!49152
supermic Nov 27 14:58:34 tftpd 104108 sent file 'bootcode.bin' octet to 172.16.1.58!49152
supermic Nov 27 14:58:35 tftp 104109 connection from 172.16.1.58!49153 dir /net/udp/3
supermic Nov 27 14:58:35 tftp 104110 connection from 172.16.1.58!49154 dir /net/udp/3
supermic Nov 27 14:58:35 tftp 104111 connection from 172.16.1.58!49155 dir /net/udp/4
supermic Nov 27 14:58:35 tftpd 104111 send file 'config.txt' octet to 172.16.1.58!49155
supermic Nov 27 14:58:35 tftpd 104111 sent file 'config.txt' octet to 172.16.1.58!49155
supermic Nov 27 14:58:35 tftp 104112 connection from 172.16.1.58!49156 dir /net/udp/3
supermic Nov 27 14:58:35 tftp 104113 connection from 172.16.1.58!49157 dir /net/udp/4
supermic Nov 27 14:58:35 tftpd 104113 send file 'start_cd.elf' octet to 172.16.1.58!49157
supermic Nov 27 14:58:36 tftp 104114 connection from 172.16.1.58!49158 dir /net/udp/3
supermic Nov 27 14:58:36 tftpd 104113 sent file 'start_cd.elf' octet to 172.16.1.58!49157
supermic Nov 27 14:58:36 tftpd 104114 send file 'fixup_cd.dat' octet to 172.16.1.58!49158
supermic Nov 27 14:58:36 tftpd 104114 sent file 'fixup_cd.dat' octet to 172.16.1.58!49158
supermic Nov 27 14:58:36 tftp 104115 connection from 172.16.1.58!49153 dir /net/udp/3
supermic Nov 27 14:58:36 tftp 104116 connection from 172.16.1.58!49154 dir /net/udp/4
supermic Nov 27 14:58:36 tftpd 104116 send file 'config.txt' octet to 172.16.1.58!49154
supermic Nov 27 14:58:36 tftpd 104116 sent file 'config.txt' octet to 172.16.1.58!49154
supermic Nov 27 14:58:36 tftp 104117 connection from 172.16.1.58!49155 dir /net/udp/3
supermic Nov 27 14:58:36 tftp 104118 connection from 172.16.1.58!49156 dir /net/udp/3
supermic Nov 27 14:58:36 tftp 104119 connection from 172.16.1.58!49157 dir /net/udp/4
supermic Nov 27 14:58:36 tftpd 104119 send file 'config.txt' octet to 172.16.1.58!49157
supermic Nov 27 14:58:36 tftpd 104119 sent file 'config.txt' octet to 172.16.1.58!49157
supermic Nov 27 14:58:36 tftp 104120 connection from 172.16.1.58!49158 dir /net/udp/3
supermic Nov 27 14:58:36 tftp 104121 connection from 172.16.1.58!49159 dir /net/udp/4
supermic Nov 27 14:58:36 tftpd 104121 send file 'e18671d6/cmdline.txt' octet to 172.16.1.58!49159
supermic Nov 27 14:58:36 tftpd 104121 sent file 'e18671d6/cmdline.txt' octet to 172.16.1.58!49159
supermic Nov 27 14:58:36 tftp 104122 connection from 172.16.1.58!49160 dir /net/udp/3
supermic Nov 27 14:58:36 tftpd 104122 send file 'arm/9pi2' octet to 172.16.1.58!49160
supermic Nov 27 14:58:36 tftpd 104122 failed to send file 'arm/9pi2' octet to 172.16.1.58!49160
supermic Nov 27 14:58:36 tftp 104123 connection from 172.16.1.58!49161 dir /net/udp/4
supermic Nov 27 14:58:36 tftp 104124 connection from 172.16.1.58!49162 dir /net/udp/3
supermic Nov 27 14:58:36 tftp 104125 connection from 172.16.1.58!49163 dir /net/udp/4
supermic Nov 27 14:58:36 tftp 104126 connection from 172.16.1.58!49164 dir /net/udp/3
supermic Nov 27 14:58:36 tftp 104127 connection from 172.16.1.58!49165 dir /net/udp/4
supermic Nov 27 14:58:36 tftpd 104127 send file 'arm/9pi2' octet to 172.16.1.58!49165
supermic Nov 27 14:58:37 tftpd 104127 sent file 'arm/9pi2' octet to 172.16.1.58!49165
```