ESP
Network Configuration

5/13/22 Brent Roman brent@mbari.org

Linux Serial Console

- RS-232 Serial 3-wire interface
  - 115200 baud, 8 bits 1-stop bit, no parity
  - Not suitable for transfer of long data files
- ESP bootloader’s shell is only available on console
- Connect to it with any terminal emulator
- Only used as a last resort for:
  - reconfiguring basic networking parameters
  - debugging an ESP that will not boot
  - capturing a kernel crash dump
  - updating the Linux kernel
    - via Ethernet or SDcard

ESP Networking Hardware

- Built-in Wired Ethernet
- WiFi
- Cellular Radio Modem
- Satellite Radio Modem
- Point-of-Presence (PPP) wired serial
- Wired Ethernet via USB
- Freewave Radio via PPP
- Acoustic Modem

Built-in Wired Ethernet

- Most reliable ESP networking interface
  - Bootloader uses built-in Ethernet to upload new kernels
- Supports only 10 or 100Mbit/s – nothing faster
  - Usually runs at only 10Mbit/s to save 300mW of power
- Network configuration
  - via Dynamic Host Configuration Protocol (DHCP)
  - manually when local net does not support DHCP
    - typically requires net configuration via serial console
  - Always responds on backup IP address of 10.10.10.10
    - this can provide ‘direct’ Ethernet access w/o DHCP

ESP WiFi

- A few older USB WiFi dongles are supported
  - primarily in 2.4Ghz band
- Usually operates as a WiFi client
  - with most common WiFi security schemes
- May alternatively operate as WiFi Access Point
  - with very limited features
- Must configure by editting text file

  `/etc/wpa_supplicant.conf`

  `https://linux.die.net/man/5/wpa_supplicant.conf`

Cellular LTE modems

- supports quite a variety of USB cell modems
  - Most require sim card and service contract
  - Beware ESP cannot read text messages
  - also cannot enter password on a webpage
    - some cell providers do not support M2M
      - (Machine-2-Machine)
  - Best service in US is from Verizon
    - Verizon offers LTE-M specialized for M2M
  - T-Mobile LTE is also quite good

Satellite Data Modem

- Only Iridium model 9555 satphone tested
  - others should also work
- Iridium satphone xfers data @ 2400baud
  - Billed $1 USD / minute
    - at best this is $1 / 12kilobytes !
  - Too slow & expensive to send images
    - or fully detailed logs
- Experimental ESP config limits data
  - Typically used with Long Mission Mode

PPP Serial Data Link

- Used within power limited vehicles
  - like MBARI's Low Power AUVs
- 5 wire RS232 serial interface
  - Runs at up to 430kBaud
  - Emulates a network interface
  - Provides full internet access
    - sort of like a faster dial-up internet

Additional Network Interfaces

- More Ethernet, modem and WiFi interfaces
  - can be connected on the ESP’s USB port
  - may require addition of powered USB hub
- Many USB devices automatically recognized
  - but require some custom configuration

Network Interface Aliases

- Physical Devices
  - `wired` #builtin Ethernet
  - `WiFi` #any WiFi dongle operating as a client
  - `AP/WiFi` #any WiFi dongle operating as access point
  - `iridium` #Iridium 9555 satphone
  - `me910` #Telit model ME910 LTE-M modem
  - `platform` #PPP serial data link to LRAUV
- Virtual Devices
  - `direct` #always available IP address 10.10.10.10
  - `wg2shore` #Wireguard VPN to ESPshore.mbari.org
  - `shore` #Obsolete PPtP VPN to ESPshore

Network Interface Names

- Interface names are assigned by the Linux kernel
  - tend to be less mnemonic than the corresponding alias
  - list all network names having aliases with:

```
$ cd /etc/sysconfig
$ ls -l `find -type l`
```

    - example: ifcfg-eth0 → ifcfg-wired
    - ‘eth0’ is the kernel’s name for the ‘wired’ interface
  - use aliases (like wired) with ifup and ifdown
  - some lower level network tools require kernel names
    - /sbin/ifconfig eth0 #for instance

Network Interface Commands

- `# ifup interfaceAlias|Name`
  - Brings up interface (ignored if already UP)
  - Must be run as root user

- `# ifdown interfaceAlias|Name`
  - Shuts down interface
  - Must be run as root user

- Take care not to shut down important links

Network Interface Status

- `$ /sbin/ifconfig {-a} {interfaceName}`
  - shows configuration for interface
  - all UP interfaces if name omitted
  - all interfaces (UP or DOWN) if -a specified

- `$ /sbin/route {-n}`
  - displays networking routes

- Both commands may be run as normal user

Network Configuration Files

```
/etc/sysconfig/    #configuration directory
    if-default  #default network interface configuration
    ifcfg-*     #custom configuration for interface *
    if-order    #start/stop interfaces on boot/shutdown
    gateway.priority    #determines default network route

/etc/inittab                    #select services to start on boot
/etc/wpa_supplicant.conf  #WiFi security/passwords,etc.
```

- Only these files may be altered
- It is quite easy to break networking if one isn't careful
  - Be prepared to access ESP via its serial console port!

/etc/sysconfig/if-default

- Configures interfaces lacking a specific ifcfg-*

```
BOOTPROTO="dhcp-b"  #dhcp in background
DHCPNAME=`hostname`

#VPN=$ESPshore/shore   #server IP / VPN interface

hosts() {    #add lines before EOS
local name=$IFNAME
[ "$IFALIAS" ] && name=`basename $IFALIAS`
  cat <<EOS
$(netIfIP $IFNAME $DHCPNAME-$name)
EOS
}
```

/etc/sysconfig/ifcfg-wired

```
. $syscfg/if-default
IFALIAS=wired
IFNAME=eth0
BOOTPROTO="dhcp-b -m1800"  #dhcp in background, renew every 30 min
#BOOTPROTO=static   #uncomment for static IP addresses
IPADDR=192.168.0.50 #assign this IP address if DHCP fails
#NETMASK=255.255.255.0
#GATEWAY=192.168.0.1
#ignoreDNS=yes   #uncomment to ignore DNS servers from DHCP

if $ignoreDNS; then
resolv_conf() {
  cat <<END
search shore.mbari.org
nameserver  8.8.8.8   #Google's public DNS
nameserver  8.8.4.4
END
}
fi

ifPost() {  #function called after main interface is UP
  ifup direct  #bring up static alias
}
```

/etc/sysconfig/ifcfg-direct

```
#last resort method of contacting host if network is misbehaving

#is to configure another host on the same physical LAN

#with the address 10.10.10.11

#Then, from that host,

#  telnet to the IPADDR configured below.

#NOTE:  You must ensure no other hosts on the LAN share these
IPADDReses!


IFALIAS=direct

IFNAME=eth0:10

BOOTPROTO=static


IPADDR=10.10.10.10

NETMASK=255.255.255.0
```

/etc/sysconfig/if-order

```
# Define interface start and stop order on boot & shutdown
#  Interfaces in START list are started before NFS and login allowed
#  Interfaces in STOP list are stopped in sequence before others
#  Interfaces in START are stopped last in reverse order

START='lo wg2shore'
STOP='wg2shore shore'
```