Nelson Chung

nelson2\_chung@compalbn.com

Abstract

The document show the technique about xxx for project – GW7557

Technique IntrO

Software Technique Document

History

|  |  |  |
| --- | --- | --- |
| Version | Author | Release |
| 0.1 | Yintao\_ling | First release |

內容

[Background 3](#_Toc42175364)

[Technique 3](#_Toc42175365)

[Reference 3](#_Toc42175366)

[Abbreviation 3](#_Toc42175367)

# Background

After knowing the basic knowledge about computer network, let us take a look at how Linux manages the Network area.

# Technique

## Connecting to Internet

### Network Interface Card in Linux

Most of devices in Linux are described as file system names, for instance, /dev/hda represents master hardware of IDE1. But Network Interface Card in Linux is represented by module names, such as eth0, eth1…

### Driver for Network Interface Card

Most drivers are installed in Linux distributions. If you want to update the Network Interface Card that Linux is not supported, two approaches are shown in following.

1. Compiling new Linux kernel.
2. Compiling new module of Network Interface Card.

Note that every time kernel is updated, driver of Network Interface Card is also necessary to be updated.

### Information of Network Interface Card

If Network Interface Card attached in PCI, PCI-E or PCI-x is detected by your Linux system, it should be shown in bootup log:

[root@Device ~]# dmesg | grep -in eth

You can also check devices in PCI.

[root@Device ~]# lspci | grep -i ethernet

As the Network Interface Card is detected, its module should be loaded as well:

[root@Device ~]# lsmod | grep <module name>

The module information can be checked through this command:

[root@Device ~]# modinfo

### Configuration of Linux Networking

TCP/IP has some major parameters: IP, Netmask, Gateway, DNS IP.

|  |  |  |
| --- | --- | --- |
| Parameters | Config file | Description |
| IP, Netmask, DHCP, Gateway…… | /etc/sysconfig/network-scripts/ifcfg-eth0 | DEVICE=NIC name  BOOTPROTO=Enable DHCP  HWADDR=NIC MAC  IPADDR=IP address  NETMASK=netmask  ONBOOT=Enable by bootup  GATEWAY=gateway  NM\_CONTROLLED=other management software |
| Device name | /etc/sysconfig/network | NETWORKING=Enable network  NETWORKING\_IPV6=Support IPv6  HOSTNAME=device hostname |
| DNS IP | /etc/resolv.conf | nameserver DNS IP address |
| Hostname of private IP | /etc/hosts | Append data below origin ones |
| Network Service | /etc/services | Apply (Service, Port) if you want |
| Protocols | /etc/protocols | Definition of protocol packet |

Restart network of Linux

[root@Device ~]# /etc/init.d/network restart

Enable / Disable network interface

[root@Device ~]# ifup eth0

[root@Device ~]# ifdown eth0

### Debug Network setting

Checking interface setup

[root@Device ~]# ifconfig <IP/netmask>

Checking network

[root@Device ~]# route -n <gateway>

Checking DNS service

[root@Device ~]# dig <Domain name>

Checking Hostname

[root@Device ~]# hostname

[root@Device ~]# ping $(hostname)

### Wireless Network Interface Card

Checking Wireless NIC from USB

[root@Device ~]# lsusb

Checking WNIC information

[root@Device ~]# iwconfig

Searching for AP with WNIC

[root@Device ~]# iwlist <interface name> scan

## Linux Command about Network

### Network Setup

#### ifconfig

# Modify Network Interface

[root@Device ~]# ifconfig [interface] [options]

interface：eth0, eth1, ppp0 ......

options ：

up, down ：Enable or Disable network interface

mtu ：Change mtu value, e.g. mtu 1500 (bytes)

netmask ：Change Netmask

broadcast：Change broadcast address

#### ifup, ifdown

# Modify FILE of Network Interface

[root@Device ~]# ifup <interface>

[root@Device ~]# ifdown <interface>

#### route

# Routing of network

[root@Device ~]# route [-n|ee]

Options:

-n ：Not showing protocols or hostname, showing IP and port indeed；

-ee ：Showing in Detail

[root@Device ~]# route add <-net/-host> <IPs/hostname> netmask <mask> [gw|dev]

[root@Device ~]# route del <-net/-host> <IPs/hostname> netmask <mask> [gw|dev]

Options:

-net ：Routing an area of IP；

-host ：Routing a host；

netmask：Size of IP network；

gw ：Gateway IP address；

dev ：Routing with specific interface；

#### ip

[root@Device ~]# ip [option] <action> [order]

Options:

-s ：Statistics of device

Actions:

link ：Device configuration, e.g. MTU, MAC

addr/address：Additional protocols setup.

route ：Routing setup.

ip link:

[root@Device ~]# ip [-s] link show

Orders:

show ：Content of device.

[root@Device ~]# ip link set <device> [action]

Options:

device ：Device code name, eth0, eth1.

Actions:

up/down ：Enable or disable a device.

address ：Change Device MAC.

name ：Label a device with a name.

mtu ：MTU setup.

ip address:

[root@Device ~]# ip address show

Orders:

show ：Content of ip information.

[root@Device ~]# ip address <add/del> <IP parameter> <dev> [option]

Options:

IP parameter ：192.168.0.100/24.

dev ：eth0, eth1.

Options:

broadcast ：Setup broadcast address.

label ：Additional name of device.

scope ：Default is global.

scope global ：Allow all of source ip.

scope site ：Only IPv6, only localhost.

scope link ：Only self connecting by device.

scope local ：Only localhost.

ip route:

[root@Device ~]# ip route show

Orders:

show ：Content of routing table.

[root@Device ~]# ip route <add/del> <IP parameter> [via] <dev>

Options:

IP parameter ：192.168.0.1 or 192.168.0.100/24.

via ：Setup outgoing gateway.

dev ：Setup outgoing device.

Some usage examples:

[root@Device ~]# ip link show

[root@Device ~]# ip -s link show eth0

[root@Device ~]# ip link set eth0 up

[root@Device ~]# ip link set eth0 mtu 1000

[root@Device ~]# ip link set eth0 name LABEL

[root@Device ~]# ip link set eth0 address 01:23:45:67:89:ab

[root@Device ~]# ip address show

[root@Device ~]# ip address add 192.168.123.1/24 broadcast + \

> dev eth0 label eth0:LABEL

[root@Device ~]# ip address del 192.168.10.0/24 dev eth0

[root@Device ~]# ip route show

[root@Device ~]# ip route add 192.168.10.0/24 via 192.168.5.11 dev eth0

[root@Device ~]# ip route add default via 192.168.1.254 dev eth0

[root@Device ~]# ip route del 192.168.10.0/24

#### dhclient

Assign eth0 to go through DHCP

[root@Device ~]# dhclient eth0

### Network Debug Tools

#### ping

[root@Device ~]# ping [option] <IP addr>

Options:

-c <value> ：Number of packets.

-n ：Do not look up domain name. Use IP address directly.

-s <value> ：Size of packets, default is 56 bytes.

-t <value> ：TTL of packets, default is 255.

-W <value> ：Time for command execution.

-M [do/dont]：Restrict packet fragment or not.

#### traceroute

[root@Device ~]# traceroute [option] <IP addr>

[root@Device ~]# traceroute -w 1 -n -T tw.yahoo.com

Options:

-n ：Do not look up domain name. Use IP address directly.

-U ：Use UDP port 33434 for traceroute.

-I ：Use ICMP for traceroute.

-T ：Use TCP port 80 for traceroute.

-w ：Timeout for waiting reply, default is 5 seconds.

-port <value>：Manually set port number.

-i <device> ：Choosing interface, e.g. ppp0 or ppp1.

-g <IPaddr> ：Choosing gateway IP.

#### netstat

[root@Device ~]# netstat [option]

Routing Options:

-r ：Show routing rable.

-n ：Show as IP and port number instead of hostname and services.

IP Options:

-a ：Show all connection statements.

-t ：Show TCP connection only.

-u ：Show UDP connection only.

-l ：Show services in listening only.

-p ：Show PID and program name.

-c <value> ：Update information periodically in seconds.

#### host

[root@Device ~]# host [-a] <hostname> [ns]

[root@Device ~]# host -a tw.yahoo.com 168.95.1.1

Options:

-a ：Show hostname detailed setup.

-ns ：Using specific name server.

#### nslookup

[root@Device ~]# nslookup [-query=type] [hostname/IP]

[root@Device ~]# nslookup -query=cname www.google.com

Options:

-query=? ：Searching for specific query type.

#### nmap

[root@Device ~]# nmap [scanType] [scanParameter] [hosts]

scanType:

-sT ：Searching for connected TCP packets.

-sS ：Searching for TCP packets with SYN label.

-sP ：Searching with ping method.

-sU ：Searching in UDP.

-sO ：Searching hosts by IP protocol.

scanParameter:

-PT ：Searching with TCP ping method.

-PI ：Searching with ICMP ping method.

-p ：Setup port range.

hosts:

192.168.1.100 ：Single specific IP.

192.168.1.0/24 ：IP range by subnet.

192.168.\*.\* ：Regex-like input.

192.168.1.0-50,60-100,140-160 ：Multiple range input.

### Network Applications

#### telnet

[root@Device ~]# telnet <hostname/IP> <port>

[root@Device ~]# telnet localhost 25

#### ftp

[root@Device ~]# ftp <hostname/IP> <port>

[root@Device ~]# ftp example.com 318

#### links - text web browser

[root@Device ~]# links [option] <URL>

[root@Device ~]# links -dump http://tw.yahoo.com > yahoo.html

Options:

-annoymous [0|1] ：Login with annoymous.

-dump [0|1] ：Dump website data to standard output.

-dump\_charset <code> ：Change to specific charset, e.g. BIG5 is cp950.

#### wget

[root@Device ~]# wget [option] <URL>

Options:

--http-user <user> ：If website has authorization,

--http-password <passwd> ：add these two parameters.

--quiet ：Do not show detailed information through wget.

### Packets Listener

#### tcpdump

[root@Device ~]# tcpdump [option] [-i interface] [-w filename] [-c times] \

[-r filename] [filter]

Options:

-A ：Display in ASCII.

-e ：Show layer-2 (MAC) content.

-nn ：Display in IP and port number instead of hostname and services.

-q ：Display shorter packet information.

-X ：Show packet content with HEX and ASCII.

-i ：Filter the listening interface.

-w ：Save the grabbed packets to file.

-r ：Load file of tcpdump.

-c ：Upper bound of listening packets.

#### nc (netcat)

[root@Device ~]# nc [option] <IP/hostname> <port>

Options:

-l ：Open a new port for listening.

-u ：Use UDP for listening instead of TCP.

### Network Security

There are 3 types of Firewall: Netfilter, TCP wrappers and Proxy. Single host uses Netfilter and TCP wrappers. Area-controlled hosts use Netfilter and proxy.

#### TCP wrappers

Simply, it uses host.allow and host.deny to control the firewall rules.

# Abbreviation

|  |  |
| --- | --- |
| Name | Full information |
|  |  |
|  |  |
|  |  |
|  |  |

# Reference