Imperial College
London

Computer Networks and Distributed Systems

Tutorial – Networking Tools

Course 527 – Spring Term 2015-2016

Emil C Lupu and Daniele Sgandurra

e.c.lupu@imperial.ac.uk, d.sgandurra@imperial.ac.uk

Ifconfig / Ipconfig

If no arguments are given, ifconfig displays the **status of the currently active interfaces**. If a single interface argument is given, it displays the status of the given interface only; if a single **-a** argument is given, it displays the **status of all interfaces**, even those that are down. **Otherwise, it configures an interface**.

```
$ ifconfig
eth0      Link encap:Ethernet  HWaddr 00:50:56:a6:01:33
          inet addr:146.169.21.39  Bcast:146.169.21.255  Mask:255.255.255.0
          inet6 addr: fe80::250:56ff:fea6:133/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1323411468 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1072026139 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1337701283239 (1.3 TB)  TX bytes:1916628466104 (1.9 TB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:28388357 errors:0 dropped:0 overruns:0 frame:0
          TX packets:28388357 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:293202079215 (293.2 GB)  TX bytes:293202079215 (293.2 GB)
```

Remember: 48 bits, expressed as 12 hexadecimal num (4 bits each).

Remember: 32 bits, expressed as 4 num in range 0-255 (8 bits each).

Ifconfig / Ipconfig

To **configure** the interface, on your own machine, from root shell (or sudo):

```
# ifconfig eth0 192.168.1.1 netmask 255.255.255.0
```

This command assigns IP address 192.168.1.1 and netmask 255.255.255.0 to eth0 network interface.

To **enable**/**shutdown** a network interface:

```
# ifup eth0
# ifdown eth0
```

Route

Show/manipulate the **IP routing table**.

U=UP

G=Gateway

The 'distance' to the target (usually counted in hops). Not really used.

```
$ route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
default         vl421-gw.net.ic 0.0.0.0         UG    0      0        0 eth0
146.169.21.0    *               255.255.255.0   U     0      0        0 eth0
```

It shows the local routing table. In this case, the **default routing table** (e.g., 0.0.0.0/0) points to to **vl421-gw.net.ic.ac.uk.**

To see the numerical IP addresses:

```
$ route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         146.169.21.223  0.0.0.0         UG    0      0        0 eth0
146.169.21.0    0.0.0.0         255.255.255.0   U     0      0        0 eth0
```

Route

To add a **specific network** (root):

```
# route add -net 192.168.201.0 netmask 255.255.255.0 gw 192.168.200.254
```

It **adds a routing entry** to reach network 192.168.201.0/24 using gateway (router) 192.168.200.254.

To **add a default rule** to the routing table (root):

```
# route add default gw {IP-ADDRESS} [INTERFACE-NAME]
```

Where:

- IP-ADDRESS: router's ip address;
- INTERFACE-NAME: which interface to forward the packets.

Example:

```
# route add default gw 192.168.1.254
```

It sets the **default gateway** to router 192.168.1.254: all the packets are forwarded there, if not specified differently.

Ping

Ping - send **ICMP ECHO_REQUEST** to network hosts.

An IP header without options is 20 bytes. An ICMP ECHO_REQUEST packet contains an additional 8 bytes worth of ICMP header followed by an arbitrary amount of data. The default size of the payload of "dummy data" contained in the ICMP message is 56 bytes. 20+**8+56**=84

```
$ ping www.google.co.uk -c 5
PING www.google.co.uk (64.233.184.94) 56(84) bytes of data.
64 bytes from wa-in-f94.1e100.net (64.233.184.94): icmp_seq=1 ttl=40 time=8.79 ms
64 bytes from wa-in-f94.1e100.net (64.233.184.94): icmp_seq=2 ttl=40 time=8.81 ms
64 bytes from wa-in-f94.1e100.net (64.233.184.94): icmp_seq=3 ttl=40 time=8.64 ms
64 bytes from wa-in-f94.1e100.net (64.233.184.94): icmp_seq=4 ttl=40 time=8.70 ms
64 bytes from wa-in-f94.1e100.net (64.233.184.94): icmp_seq=5 ttl=40 time=8.86 ms

--- www.google.co.uk ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4004ms
rtt min/avg/max/mdev = 8.643/8.764/8.869/0.130 ms
```

ping uses the ICMP protocol's mandatory **ECHO_REQUEST** datagram to elicit an **ICMP ECHO_RESPONSE** from a host or gateway. ECHO_REQUEST datagrams (``pings'') have an IP and ICMP header, followed by a struct timeval and then an arbitrary number of ``pad'' bytes used to fill out the packet.

Ping

Common options:

```
ping [ -LRUbdfnqrvVaAB] [-c count] [-i interval] [-s packetsize]
[-t ttl] [ -w deadline] [-I interface] destination
```

-n: Numeric output only. No attempt will be made to lookup symbolic names for host addresses.

-c count: Stop after sending count ECHO_REQUEST packets.

-i interval: Wait interval seconds between sending each packet. The default is to wait for one second between each packet normally.

-s packetsize: Specifies the number of data bytes to be sent. The default is 56, which translates into 64 ICMP data bytes when combined with the 8 bytes of ICMP header data.

-t ttl: Set the IP Time to Live.

-w deadline: Specify a timeout, in seconds, before ping exits regardless of how many packets have been sent or received. In this case ping does not stop after count packet are sent, it waits either for deadline expire or until count probes are answered or for some error notification from network.

-I interface: Interface is either an address, or an interface name. If interface is an address, it sets source address to specified interface address. If interface in an interface name, it sets source interface to specified interface.

Traceroute / Tracert

Traceroute tracks the route packets taken from an IP network on their way to a given host. It utilizes the IP protocol's **time to live** (**TTL**) field and attempts to elicit an **ICMP TIME_EXCEEDED** response from each gateway along the path to the host.

Three probes (by default) are sent at each ttl setting and a line is printed showing the **ttl**, **address** of the gateway and **round trip time** of each probe. […] If the probe answers come from different gateways, the address of each responding system will be printed. If there is no response within a 5.0 seconds (default), an "*" (asterisk) is printed for that probe.

Traceroute / Tracert

```
$ traceroute www.google.co.uk
traceroute to www.google.co.uk (64.233.184.94), 30 hops max, 60 byte packets
 1  vl421-hsrp-slave.net.ic.ac.uk (146.169.21.253)  0.634 ms  0.658 ms  0.727 ms
 2  * * *
 3  rach-deck.net.ic.ac.uk (194.82.153.22)  0.959 ms  1.856 ms  1.810 ms
 4  srx-3600-1.net.ic.ac.uk (155.198.1.105)  0.746 ms  0.729 ms  0.680 ms
 5  rachael-untrust.net.ic.ac.uk (194.82.153.180)  1.867 ms  1.975 ms  2.035 ms
 6  ae10-3799.londic-rbr2.ja.net (146.97.136.89)  1.041 ms  1.136 ms  1.092 ms
 7  ae20-0.londpg-sbr1.ja.net (146.97.37.133)  1.368 ms  1.388 ms  1.396 ms
 8  ae29.londhx-sbr1.ja.net (146.97.33.1)  1.965 ms  1.888 ms  1.866 ms
 9  po1.lond-ban3.ja.net (146.97.35.106)  20.624 ms  20.591 ms  20.653 ms
10  72.14.196.137 (72.14.196.137)  1.912 ms  1.937 ms  1.994 ms
11  209.85.249.224 (209.85.249.224)  2.849 ms 216.239.54.159 (216.239.54.159)  2.037
ms 216.239.54.138 (216.239.54.138)  2.795 ms
12  72.14.239.231 (72.14.239.231)  3.248 ms 209.85.143.67 (209.85.143.67)  2.541 ms
72.14.239.231 (72.14.239.231)  3.169 ms
13  216.239.42.43 (216.239.42.43)  7.823 ms 216.239.41.218 (216.239.41.218)  8.491 ms
209.85.253.208 (209.85.253.208)  9.043 ms
14  72.14.238.215 (72.14.238.215)  8.683 ms 216.239.51.147 (216.239.51.147)  8.559 ms
66.249.95.252 (66.249.95.252)  8.276 ms
15  * * *
16  wa-in-f94.1e100.net (64.233.184.94)  8.680 ms  7.885 ms  8.797 ms
```

Nslookup

nslookup - query **Internet name servers** interactively.

```
$ nslookup www.imperial.ac.uk
Server:         155.198.142.8
Address:        155.198.142.8#53

Name:   www.imperial.ac.uk
Address: 155.198.140.14
```

Nslookup has two modes:

- **Interactive** mode allows the user to query name servers for information about various hosts and domains or to print a list of hosts in a domain.
- **Non-interactive** mode is used to print just the name and requested information for a host or domain.

Nslookup

```
$ nslookup
> www.google.com
Server:         155.198.142.8
Address:        155.198.142.8#53

Non-authoritative answer:
Name:   www.google.com
Address: 64.233.184.106
Name:   www.google.com
Address: 64.233.184.105
Name:   www.google.com
Address: 64.233.184.99
Name:   www.google.com
Address: 64.233.184.103
Name:   www.google.com
Address: 64.233.184.104
Name:   www.google.com
Address: 64.233.184.147
```

Response from one of the local configured nameservers

The list of authoritative nameservers for Google can be queried on https://www.internic.net/whois.html

Domain Name: GOOGLE.COM
Registrar: MARKMONITOR INC.
Sponsoring Registrar IANA ID: 292
Whois Server: whois.markmonitor.com
Name Server: NS1.GOOGLE.COM
Name Server: NS2.GOOGLE.COM
Name Server: NS3.GOOGLE.COM
Name Server: NS4.GOOGLE.COM
Updated Date: 20-jul-2011
Creation Date: 15-sep-1997
Expiration Date: 14-sep-2020

Or similarly:

```
$ host -t ns google.com
google.com name server ns3.google.com.
google.com name server ns4.google.com.
google.com name server ns1.google.com.
google.com name server ns2.google.com.
```

Nslookup

Running nslookup command against one of those servers, one will get the **authoritative answer**:

```
$ nslookup www.google.com ns1.google.com
Server:         ns1.google.com
Address:        216.239.32.10#53

Name:   www.google.com
Address: 64.233.184.105
Name:   www.google.com
Address: 64.233.184.104
Name:   www.google.com
Address: 64.233.184.106
Name:   www.google.com
Address: 64.233.184.147
Name:   www.google.com
Address: 64.233.184.99
Name:   www.google.com
Address: 64.233.184.103
```

/etc/hosts

The “hosts” file is a computer file used by an operating system to **map hostnames to IP addresses**:

```
$ cat /etc/hosts
127.0.0.1                localhost
::1                      ip6-localhost ip6-loopback
fe00::0                  ip6-localnet
ff00::0                  ip6-mcastprefix
ff02::1                  ip6-allnodes
ff02::2                  ip6-allrouters
ff02::3                  ip6-allhosts
146.169.1.25             tokaimura.doc.ic.ac.uk tokaimura
146.169.1.157            thoth.doc.ic.ac.uk thoth
146.169.1.169            tody.doc.ic.ac.uk tody
146.169.1.86             whirly.doc.ic.ac.uk whirly
146.169.1.156            ntp0.doc.ic.ac.uk ntp0
146.169.21.39            vm-shell1.doc.ic.ac.uk vm-shell1
```

You can add hostname and IP addresses to the file `/etc/hosts` **for static lookups**.

/etc/resolv.conf

“resolv.conf” is the name of a computer file used in various operating systems to configure the system's **Domain Name System** (**DNS**) resolver.

```
$ cat /etc/resolv.conf
# Dynamic resolv.conf(5) file for glibc resolver(3) generated by
resolvconf(8)
#     DO NOT EDIT THIS FILE BY HAND -- YOUR CHANGES WILL BE OVERWRITTEN
nameserver 155.198.142.8
nameserver 155.198.142.7
search doc.ic.ac.uk
```

Search list for hostname lookup. The search list is normally determined from the local domain name but it can be set to a list of domains.

Used for resolving short host-names - e.g. “test” is resolved to test.doc.ic.ac.uk.

In case of doubt there is always the possibility to use **the Google DNS servers** as your default DNS servers:

```
nameserver 8.8.8.8
nameserver 8.8.4.4
```

/etc/network/interfaces

Contains **network interface configuration information** for the ifup(8) and ifdown(8) commands.

```
$ cat /etc/network/interfaces
# Used by ifup(8) and ifdown(8). See the interfaces(5) manpage or
# /usr/share/doc/ifupdown/examples for more information.

auto lo
iface lo inet loopback
auto eth0
iface eth0 inet dhcp
```

To configure **a static IP**:

```
auto eth0
iface eth0 inet static
  address 192.168.1.14
  gateway 192.168.1.1
  netmask 255.255.255.0
  network 192.168.1.0 broadcast 192.168.1.255
```

For **wireless**:

```
auto wlan0
iface wlan0 inet dhcp
 pre-up /etc/init.d/wpa.sh
 start post-down /etc/init.d/wpa.sh stop
```

/etc/hostname

It stores the **computer’s hostname**.

```
$ cat /etc/hostname
vm-shell1.doc.ic.ac.uk
```

When the system **boots** it will automatically read the hostname from the file `/etc/hostname`.

ARP

Address Resolution Protocol: to find the media access control address of a network neighbor for a given IPv4.

Each complete entry in the ARP cache will be marked with the C flag. Permanent entries are marked with M and published entries have the P flag.

To display the **cache**:

```
$ arp
Address                  HWtype  HWaddress           Flags Mask            Iface
vm-svnuser02.doc.ic.ac.  ether   00:50:56:a6:19:0b   C                     eth0
vl421-gw.net.ic.ac.uk    ether   00:00:0c:9f:f0:00   C                     eth0
marabou.doc.ic.ac.uk     ether   00:50:56:a6:02:bb   C                     eth0
vl421-hsrp-slave.net.ic  ether   00:1d:71:83:8c:00   C                     eth0
vl421-hsrp-master.net.i  ether   00:1d:71:83:48:00   C                     eth0
vm-timetable.doc.ic.ac.  ether   00:50:56:ac:20:8c   C                     eth0

$ arp -n
Address                  HWtype  HWaddress           Flags Mask            Iface
146.169.21.53            ether   00:50:56:a6:19:0b   C                     eth0
146.169.21.223           ether   00:00:0c:9f:f0:00   C                     eth0
146.169.21.61            ether   00:50:56:a6:02:bb   C                     eth0
146.169.21.253           ether   00:1d:71:83:8c:00   C                     eth0
146.169.21.254           ether   00:1d:71:83:48:00   C                     eth0
146.169.21.6             ether   00:50:56:ac:20:8c   C                     eth0
```

Netstat

The possible states are as follows:

CLOSE_WAIT
CLOSED
ESTABLISHED
FIN_WAIT_1
FIN_WAIT_2
LAST_ACK
LISTEN
SYN_RECEIVED
SYN_SEND
TIMED_WAIT

netstat - Print **network connections**, routing tables, interface statistics, masquerade connections, and multicast memberships.

```
$ netstat -antu
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 0.0.0.0:53742           0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:111             0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:46676           0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:38165           0.0.0.0:*               LISTEN
tcp        0      0 192.168.122.1:53        0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:42774           0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:6010            0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:6011            0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:6012            0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:6013            0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:6016            0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:45121           0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:42693           0.0.0.0:*               LISTEN
tcp        0      0 146.169.21.39:57246     146.169.1.117:2049      TIME_WAIT
tcp        0      0 146.169.21.39:46207     216.58.213.173:443      ESTABLISHED
….
```

Netstat

Common options:

-a: (all): show all connections;

-n: (numeric): show numerical addresses;

-p: (pid): show the PID and name of the program to which each socket belongs;

-t: (tcp): show TCP connections;

-u: (udp): show UDP sessions;

-r: (route): show routing table.

```
$ netstat -r
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
default         vl421-gw.net.ic 0.0.0.0         UG        0 0          0 eth0
146.169.21.0    *               255.255.255.0   U         0 0          0 eth0
```

Tested Machine

Hostname: `shell1.doc.ic.ac.uk`

– alias: `vm-shell1.doc.ic.ac.uk`

IP: `146.169.21.39`

Try it:

```
ssh username@shell1.doc.ic.ac.uk
```