MHPC
Master in High Performance Computing

Moreno Baricevic

CNR-IOM DEMOCRITOS
Trieste, ITALY

INTRO TO NETWORKING

PART 2: LINUX commands
(short)

SISSA
ma per seguir virtute e canoscenza
Scuola Internazionale Superiore di Studi Avanzati

ICTP
The Abdus Salam International Centre for Theoretical Physics

Agenda

- Network Interfaces
- LINUX command line utilities
  - Hardware Diagnostic
  - Configuration
  - Software Diagnostic
  - Clients Applications
  - Benchmarking

Network Interfaces

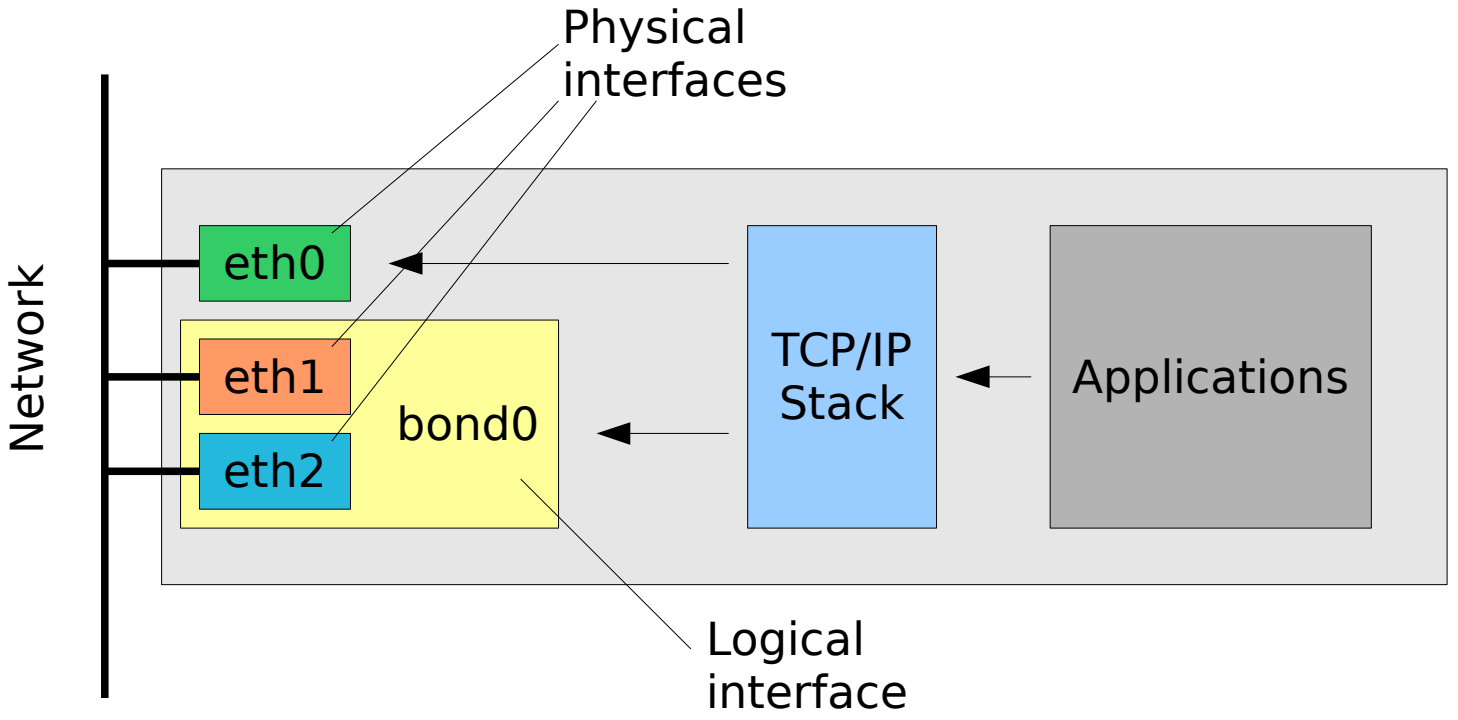
Main network interfaces:

- **lo**: loopback virtual interface for internal networking (provides networking stack to applications). NEVER play with this interface.
- **ethX (eth0, eth1, ...)**: physical Ethernet interfaces
- **ethX:LABEL (eth0:foo, eth0:10, ...)**: virtual interface, in case two or more IP addresses/networks are needed on the same physical interface
- **wlanX** or **iwX (wlan0, ...)**: wireless interface

Interfaces for specific uses:

- **bondX (bond0)**: bonding interface (link aggregation, load balancing), enslave 2 or more interfaces
- **brX (br0)**: ethernet bridging interface (layer 2 forwarding), enslave 2 or more interfaces
- **tunX/tapX (tun0/tap0)**: user-space logical interfaces (virtual machines, tunnels, ...)
- **sit0**: virtual interface which tunnels IPv6-in-IPv4
- (pppX, slipX, bnepX and many many more...)

LINUX Network Stack (example)



eth0: has it's own MAC and IP address, configured as usual

bond0:
- forces the **same** MAC address on both the slaves (**eth1** and **eth2**);
- the MAC address used is the one of the first interface enslaved;
- the IP address belongs to bond0, not eth* (ifconfig bond0 ...);
- depending on the bonding mode adopted, additional configuration may be required on the switch.

Some command line utilities

mii-tool, **ethtool**:	HW diagnostic/configuration
ifconfig, **ip**, **route**:	SW configuration
netstat, **lsof**:	report network resources status
{arp,}ping, **{tcp,}traceroute**:	diagnostic tools
telnet:	simple TCP client
nmap, **nc** (netcat):	TCP/IP swiss army knives
ssh, **scp**, **sftp**:	SSH clients
wget, **curl**:	web downloader (http, ftp, tftp)
tftp, **ftp**:	TFTP and FTP clients
dhclient, **dhcpcd**, **udhcpc**, **pump**:	DHCP clients
nslookup, **host**, **dig**:	DNS clients
tcpdump, **{wire,t}shark**:	network sniffers
iptables, **iptables-save**:	firewall configuration

Hardware Diagnostic

- **mii-tool**: this utility checks or sets the status of a network interface's Media Independent Interface (MII) unit. The default short output reports the negotiated link speed and link status for each interface.

  # mii-tool eth0

  # mii-tool -w

- **ethtool**: display or change ethernet card settings. Is used for querying settings of an ethernet device and changing them. With a single argument specifying the device name prints current setting of the specified device.

  # ethtool eth0
  # ethtool -i eth0

Configuration

- **ifconfig**: is used to configure the kernel-resident network interfaces. It is used at boot time to set up interfaces as necessary. After that, it is usually only needed when debugging or when system tuning is needed.

```
# ifconfig
# ifconfig eth0 192.168.0.2 netmask 255.255.255.0 up
# ifconfig eth0 down
```

- **ip**: show / manipulate routing, devices, policy routing and tunnels

```
# ip addr
# ip link show eth0
# ip monitor link
# ip neigh
```

- **route**: manipulates the kernel's IP routing tables. Its primary use is to set up static routes to specific hosts or networks via an interface after it has been configured with the **ifconfig** program.

```
# route add default gw 192.168.0.1
# route -n
```

Software Diagnostic

- **ping**: uses the ICMP protocol's mandatory ECHO_REQUEST datagram to elicit an ICMP ECHO_RESPONSE from a host or gateway
  # ping 127.0.0.1
  # ping 192.168.0.1
  # ping -c 1 -w 10 www.google.com
- **arp**: manipulate the system ARP cache
  # arp -n
- **arping**: send ARP REQUEST to a neighbor host
  # arping 192.168.0.1
  # arping -c 1 -I eth2 192.168.0.1
- **traceroute**: utilizes the IP protocol 'time to live' field and attempts to elicit an ICMP TIME_EXCEEDED response from each gateway along the path to some host
  # traceroute www.google.com
- **tcptraceroute**: **traceroute** implementation using TCP packets
  # tcptraceroute www.google.com

Clients Applications

- **telnet**: user interface to the TELNET protocol, but can be used to open a TCP connection to any port (useful for testing/diagnostic)
  ```
  # telnet switch01
  # telnet www.google.com 80
  ```
- **netcat/nc**: TCP/IP swiss army knife
  ```
  # nc -h
  ```
- **ssh/scp/sftp**: OpenSSH clients (secure shell for remote login, remote file copy and and secure file transfer)
  ```
  # ssh user@ssh.somedomain.com
  # ssh -l user ssh.somedomain.com
  # scp /home/foo/file1 user@hostX.somedomain.com:/tmp/
  ```
- **ftp/tftp**: file transfer programs, FTP and TFTP clients
  ```
  # ftp ftp.somedomain.com
  # tftp -v master.hpc -c get /pxe/pxelinux.0 ./pxelinux0
  ```

Clients Applications

- **wget**: network downloader
  # wget http://www.google.com
  # wget -r -l0 -t0 -np -nc -p -k www.somedomain.com/foo/
- **curl**: transfer data from/to a server using one of the supported protocols (HTTP, HTTPS, FTP, TFTP, DICT, TELNET, LDAP or FILE)
  # curl www.google.com
  # curl tftp://master.hpc/pxe/pxelinux.0 -o /tmp/foo.0
- **links/lynx/w3m**: text-based Web browsers and pages
  # w3m www.google.com

DNS Clients

- **nslookup**: is a program to query Internet domain name servers (uses /etc/resolv.conf for default domain names and servers)

```
# nslookup 192.168.0.1
# nslookup www.google.com
# nslookup www.google.com dns.somedomain.com
```

- **host**: a simple utility for performing DNS lookups. It is normally used to convert names to IP addresses and vice versa.

```
# host 192.168.0.1
# host www.google.com
# host -t MX gmail.com
```

- **dig**: (domain information groper) is a flexible tool for interrogating DNS name servers. DNS administrators use dig to troubleshoot DNS problems because of its flexibility, ease of use and clarity of output. Other lookup tools tend to have less functionality than dig.

```
# dig -x 192.168.0.1
# dig www.google.com
# dig +search www
# dig -t AXFR somedomain.com
```

DHCP clients

- **dhclient**: the Internet Systems Consortium DHCP Client provides a means for configuring one or more network interfaces using the Dynamic Host Configuration Protocol, BOOTP protocol, or if these protocols fail, by statically assigning an address.
  ```
  # dhclient eth0
  # dhclient -n eth0
  ```
- **dhcpcd**: is a DHCP client daemon
  ```
  # dhcpcd eth0
  # dhcpcd -R -N -t 60 eth0
  ```
- **pump**: yet another DHCP client (debian/ubuntu/knoppix specific)
- **udhcpc**: micro DHCP client, provided by **busybox**
  ```
  # udhcpc -i eth0 -f -n -q
  ```

Network Resources Status

- **netstat**: print network connections, routing tables, interface statistics, masquerade connections, and multicast memberships
  ```
  # netstat -p -u -t -a -n
  # netstat -rn
  ```
- **lsof**: list open files and sockets (and a lot of other things)
  ```
  # lsof -nP -i TCP -a -c ssh
  # lsof -nP -i UDP
  ```
- **fuser**: identify processes using files or sockets
  ```
  # fuser -v -n tcp 22
  ```
- **ss**: yet another utility to investigate sockets
  ```
  # ss -4 -n -a
  ```

Network Sniffing and Monitoring and...

- **tcpdump**: dump traffic on a network (sniffer)
  ```
  # tcpdump -i eth0 -nn
  # tcpdump -i any -qtep port bootpc and ip broadcast
  # tcpdump -i any -e arp or icmp
  ```
- **tshark/wireshark**: dump and analize network traffic (providing also a graphic interface)
  ```
  # wireshark &
  # tshark -i eth0 -V arp
  ```
- **ettercap**: sniffing of live connections, content filtering, active and passive dissection of many protocols
- **arpwatch**: keep track of ethernet/ip address pairings (logs activity and reports certain changes via e-mail)

Firewall Configuration and Testing

- **iptables**: administration tool for IPv4 packet filtering and NAT

```
# iptables -A INPUT -m state \
  –state ESTABLISHED,RELATED -j ACCEPT
# iptables -A INPUT -p tcp --dport 22 --syn -j ACCEPT
# iptables -A INPUT -j DROP
# iptables -nvL INPUT
```

- **iptables-save/iptables-restore**: show, save and restore iptables configuration

```
# iptables-save | grep '\-A INPUT' | nl
# iptables-save > ./iptables.conf
# iptables-restore < ./iptables.conf
```

- **nmap**: network exploration tool and security / port scanner

```
# nmap -sP 192.168.0.0/24
# nmap -sS -p 22,25,80,443,8080 hostX
```

- **netcat/nc, telnet, ping, arping, hping2, tcptraceroute, ...**: file transfer programs, FTP and TFTP clients

Some network benchmarking tools

- **iperf**
  - http://iperf.sourceforge.net/
- **netperf**
  - http://www.netperf.org/
- **netpipe**
  - http://www.scl.ameslab.gov/Projects/NetPIPE/
- **IMB (Intel MPI Benchmark)**
  - https://software.intel.com/en-us/articles/intel-mpi-benchmarks

That's All Folks!


```
( questions ; comments ) | mail -s uheilaaa baro@democritos.it

( complaints ; insults ) &>/dev/null
```

REFERENCES AND USEFUL LINKS

SOFTWARE:

- Linux Kernel http://www.kernel.org
- Netfilter http://www.netfilter.org

- nmap http://www.insecure.org/nmap/
- hping http://www.hping.org/
- netcat http://netcat.sourceforge.net/
- iptstate http://www.phildev.net/iptstate/
- ss http://linux-net.osdl.org/index.php/Iproute2
- lsof ftp://lsof.itap.purdue.edu/pub/tools/unix/lsof/
- netstat http://www.tazenda.demon.co.uk/phil/net-tools/
- tcpdump http://www.tcpdump.org
- wireshark http://www.wireshark.org
- ethereal http://www.ethereal.com (vedi wireshark)
- iptraf http://iptraf.seul.org/
- ettercap http://ettercap.sourceforge.net
- dsniff http://www.monkey.org/~dugsong/dsniff/
- tcptraceroute http://michael.toren.net/code/tcptraceroute/
- (telnet, traceroute, ping, ...)

DOC:

- IPTables HOWTO http://www.netfilter.org/documentation/HOWTO/
- IPTables tutorial http://iptables-tutorial.frozentux.net/
- Having fun with IPTables
  http://www.ex-parrot.com/~pete/upside-down-ternet.html
- Denial of Service http://www.cert.org/tech_tips/denial_of_service.html
- IPv4 Address space
  - http://www.cymru.com/Documents/bogon-bn.html
  - http://www.iana.org/assignments/ipv4-address-space
  - http://www.oav.net/mirrors/cidr.html
  - http://en.wikipedia.org/wiki/IPv4
  - IANA http://www.iana.org
  - RIPE http://www.ripe.net
  - RFC 3330 http://www.rfc.net/rfc3330.html
- SANS: http://www.sans.org/reading_room/whitepapers/firewalls/
  http://www.sans.org/reading_room/

RFC: (http://www.rfc.net)

- RFC 791 – Internet Protocol (IPv4)
  http://www.rfc.net/rfc791.html
- RFC 793 – Transmission Control Protocol (TCP)
  http://www.rfc.net/rfc793.html
- RFC 768 – User Datagram Protocol (UDP)
  http://www.rfc.net/rfc768.html
- RFC 792 – Internet Control Message Protocol (ICMP)
  http://www.rfc.net/rfc792.html
- RFC 1180 – A TCP/IP Tutorial
  http://www.rfc.net/rfc1180.html
- RFC 1700 / IANA db – Assigned Numbers
  http://www.rfc.net/rfc1700.html
  http://www.iana.org/numbers.html
- RFC 3330 – Special-Use IPv4 Addresses
  http://www.rfc.net/rfc3330.html
- RFC 1918 – Address Allocation for Private Internets
  http://www.rfc.net/rfc1918.html
- RFC 2196 – Site Security Handbook
  http://www.rfc.net/rfc2196.html
- RFC 2827 – Network Ingress Filtering
  http://www.rfc.net/rfc2827.html
- RFC 2828 – Internet Security Glossary
  http://www.rfc.net/rfc2828.html
- RFC 1149 – Transmission of IP Datagrams on Avian Carriers
  http://www.rfc.net/rfc1149.html
- Unofficial CPIP WG
  http://www.blug.linux.no/rfc1149/
- RFC 2549 – IP over Avian Carriers with Quality of Service
  http://www.rfc.net/rfc2549.html
- Firewalling the CPIP
  http://www.tibonia.net/
  http://www.hotink.com/wacky/dastrdly/

Some acronyms...

IP – Internet Protocol
TCP – Transmission Control Protocol
UDP – User Datagram Protocol
ICMP – Internet Control Message Protocol
ARP – Address Resolution Protocol
MAC – Media Access Control

OS – Operating System
NOS – Network Operating System
LINUX – LINUX is not UNIX

PING – Packet Internet Groper

FTP – File Transfer Protocol – (TCP/21,20)
SSH – Secure SHell – (TCP/22)
TELNET – Telnet – (TCP/23)
SMTP – Simple Mail Transfer Protocol – (TCP/25)
DNS – Domain Name System – (UDP/53)
NTP – Network Time Protocol – (UDP/123)
BOOTPS – Bootstrap Protocol Server (**DHCP**) – (UDP/67)
BOOTPC – Bootstrap Protocol Server (**DHCP**) – (UDP/68)
TFTP – Trivial File Transfer Protocol – (UDP/69)
HTTP – HyperText Transfer Protocol – (TCP/80)
NTP – Network Time Protocol – (UDP/123)
SNMP – Simple Network Management Protocol – (UDP/161)
HTTPS – HyperText Transfer Protocol over TLS/SSL – (TCP/443)
RSH – Remote Shell – (TCP/514,544)

ISO – International Organization for Standardization
OSI – Open System Interconnection

TLS – Transport Layer Security
SSL – Secure Sockets Layer

RFC – Request For Comments

ACL – Access Control List

PDU – Protocol Data Unit

TCP flags:
- **URG**: Urgent Pointer field significant
- **ACK**: Acknowledgment field significant
- **PSH**: Push Function
- **RST**: Reset the connection
- **SYN**: Synchronize sequence numbers
- **FIN**: No more data from sender

RFC 3168 TCP flags:
- **ECN**: Explicit Congestion Notification
- (**ECE**: ECN Echo)
- **CWR**: Congestion Window Reduced

ISN – Initial Sequence Number