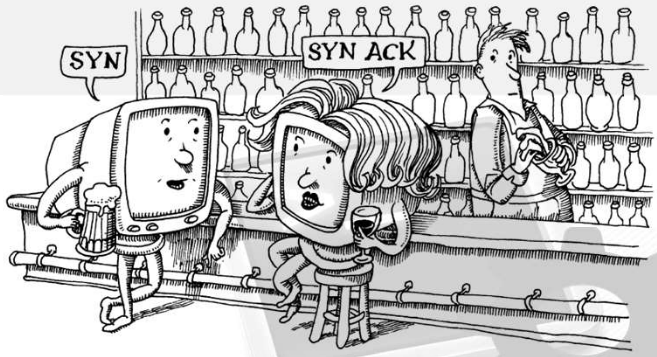
Linux Networking
(Fundamentals)



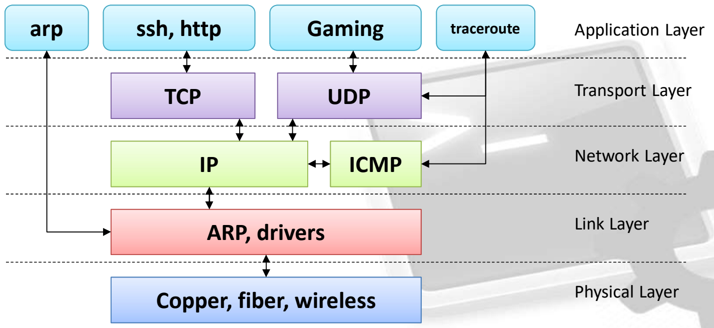
Reference Material:
[1] UNIX and Linux System Administration Handbook, Chapter 13 (TCP/IP Networking).

Index

- **Introduction (TCP/IP) (G661 + G662)**
- **Network Interface**
- **Link Layer**
- **Network Layer**

Intro (TCP/IP)

- Protocol "Suite", a set of protocols designed to implement interconnection networks.
  - Origin: research project of the USA defense department (ARPANET).
- Multiple components, arranged hierarchically (stack)

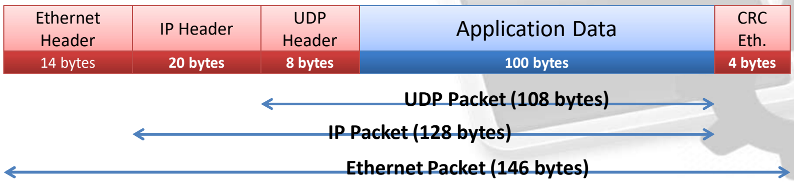


Intro (TCP/IP)

- Protocol "Suite", a set of protocols designed to implement interconnection networks.
  - Origin: research project of the USA defense department (ARPANET).
- Multiple components, arranged hierarchically (stack)
  - **UDP**, User Datagram Protocol, unverified, one-way data delivery
  - **TCP**, Transmission Control Protocol, reliable, full duplex, flow controlled, error corrected conversations.
  - **IP**, the Internet Protocol, routes data packets from one machine to another
  - **ICMP**, the Internet Control Message Protocol, provides low level support for IP: error messages, routing assistance, debugging
  - **ARP**, Address Resolution Protocol, translates IP addresses into HW address (MAC).

Intro (TCP/IP): Encapsulation

- Data travels on the network in the form of packets, bursts of data with a maximum length imposed by the link layer.
- Each packet consists of a header and a payload:
  - Header: includes Source-Destination and protocol information.
  - Payload: the information (Data).
- As a packet travels down the TCP/IP protocol stack, each protocol adds its own header information



Ethernet Header 14 bytes | IP Header 20 bytes | UDP Header 8 bytes | Application Data 100 bytes | CRC Eth. 4 bytes

UDP Packet (108 bytes)

IP Packet (128 bytes)

Ethernet Packet (146 bytes)

Intro (TCP/IP): Packet Addressing

- **HW** Addressing (link layer):
  - Each net interface has one MAC addr that distinguishes it in the physical network.
  - Ethernet Network: 6 byte direction (2-digit hex bytes: 00:50:8D:9A:3B:DF)
- **IP** Addressing: (IPv4: 216.58.211.196)
  - Identifies the network interface in internet. Unique at global level* (NAT)
  - Physical Address – IP address mapping: ARP protocol
- **Hostname** addressing:
  - Number-based directions hard to remember (216.58.211.196 ??). Name mapping.
  - File mapping (/etc/hosts) or DNS (world-wide Domain Name Server)
- **Ports**:
  - IP identifies the interface, How to identify active services? (multiple connections)
  - Extend IP address with port number: 16 bits identifying a communication channel.
  - Standard services (ssh, ftp, http) associated to pre-established ports. (/etc/services)

Intro (TCP/IP): IP Addressing

- **Types** of IPv4 addresses: (32 bits divided into 4 8-bit fields a.b.c.d)
  - Determines which portion identifies the network and which one the host.
  - Class A: (N.H.H.H) 1.x.x.x – 127.x.x.x (apple, AT&T, Ford, US DoD, …)
    - Network part=a, 126 nets
    - Host part=b.c.d, +16 millon hosts at each net.
  - Class B: (N.N.H.H) 128.x.x.x – 191.x.x.x
    - +16K nets, 65K hosts per net.
  - Class C: (N.N.N.H) 192.x.x.x – 233.x.x.x
  - Classes D and E: 234.0.0.0 – 255.x.x.x
    - Experimental networks and multicast addressing.
  - **Subnetting**: A & B oversized, break classes into subclasses
    - Part of the host identifier is employed to identify the network
    - Through the network mask (mapping)

0.0.0.0: My own Host (NO net connection)

0.x.x.x: One machine in our network

127.0.0.1: Loopback. Does not reach the NIC.

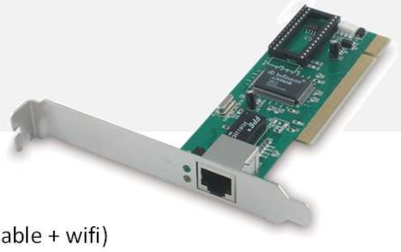
255.255.255.255: Bcast in local network.

x.x.x.255: Bcast in specified network.

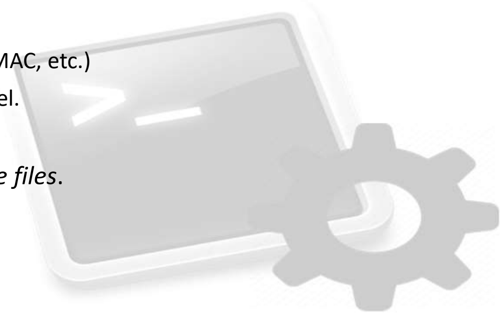
Index

- Introduction (TCP/IP)
- **Network Interface**
- Link Layer
- Network Layer

Network Interface



- **Host / Interface**:
  - Hosts are computers/individual systems.
  - Each host can have one or more network interfaces (NICs) (Cable + wifi)
    - Each interface represents a connection to a different network (different IP).
- Basic network equipment:
  - Hubs (level OSI-1): Only interconnects wires.
  - Switches (level OSI-2): Ethernet level management (ARP, MAC, etc.)
  - Routers (level OSI-3): IP packet management, network level.
  - Others: traffic balancing, firewalls, …
- Linux **does not** perform net management through *device files*.
  - ethX has no device file associated (/dev/ethX not found)
  - NICs are managed through kernel modules (drivers)



Network Interface Configuration

- How to see available interfaces?:
  - **ifconfig –a**
  - **ip link show**
- Configuration (Debian): file **/etc/network/interfaces**
  - Establishes the configuration of network interfaces.
  - Allows additional functionality: routes, alias, pre/post operations,…
  - Fields:
    - auto <interface>: activates the interface when the system boots up
    - iface <interface> <ip_addressing> <method>: interface configuration
      - ip_addressing: inet (IPv4) / inet6 (IPv6)
      - method: dhcp (automatic) / static (manual, requires additional lines for configuration)
  - Loopback interface:
    - Communication of network apps hosted in the same system.

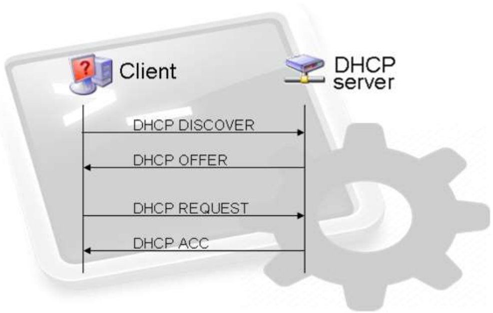
```
auto lo
iface lo inet loopback

allow-hotplug enp0s3
iface enp0s3 inet static
        address 192.168.1.132
        netmask 255.255.255.0
```

Network Interface Configuration

- **DHCP** (Dynamic Host Configuration Protocol)
  - The DHCP service performs **automatic network configuration** for the system.
    - “Renting” parameters from a server: IP, Gateway, DNS, etc.
    - “Safe” : allows forcing network configuration based on MAC address.
    - Easier: centralized management of the whole network.
    - Dynamic: information is only valid temporally.
    - Requires a “client” service at each host.
  - How to specify we want to use DHCP:
    - In /etc/network/interfaces:

```
iface  ethX  inet  dhcp
```

    - man dhclient
    - ifconfig eth0 up



Network Interface Configuration

- Interface configuration can be modified in a “running” system.
  - STEP 1, Take interface down. (ifdown)
  - STEP 2, Modification. Edit the file /etc/network/interfaces or command ifconfig
  - STEP 3, Restart. (ifup)
- Commands **ifup/ifdown (or ip link)**: power on/off a network interface.
  - Syntax: ifdown enp0s3 (power off enp0s3 card).
    - Ip link set dev enp0s3 down
- Command **ifconfig (or ip address)**: net parameter configuration.
  - Syntax: ifconfig <interface> <address> <options>
    - Example: ifconfig enp0s3 192.168.1.13 netmask 255.255.255.198 broadcast 192.168.1.191 up
    - Example: ip address add 192.168.1.13/255.255.255.198 dev enp0s3
    - ifconfig –a prints information about available interfaces.
  - Caution!! changes made with ifconfig are not permanent (do not modify interfaces file).

Index

- Introduction (TCP/IP)
- Network Interface
- **Link Layer**
- Network Layer

Link Layer

- The physical level in TCP/IP, almost always an ethernet network.
  - Each interface (NIC) has a unique MAC address.
  - Layer in charge of IP Frame <–> Ethernet Frame conversion.
    - Need to map IP address and MAC Address: ARP (Address Resolution Protocol).
  - **ARP Protocol**:
    - Search @MAC corresponding to a @IP in the local ARP table (translated address cache)
    - If not in the table, it performs a broadcast and the receiver informs. ARP table is updated for future connections.
  - Command **arp (ip neighbour)**: manipulation/display of ARP table.
  - Configuration/Modification of @MAC:
    - # ifconfig eth0 **hw ether** 00:02:B3:19:C8:21

Index

- Introduction (TCP/IP)
- Network Interface
- Link Layer
- **Network Layer**

Network Layer

- Through ARP only hosts in my net segment can be reached.
  - Cannot reach further than my hub/switch/router.
  - IP routes must be established for external addresses.
- **Route Tables**: Information about how to reach IP destinations
  - **Destination**: Identifies destination network.
  - **Gateway**: how to reach to Destination (* means no forwarding is required, the packet is already in that network.
  - **Genmask**: network mask (identifies the subnetwork).
  - **Iface**: network interface to reach destination network.

```
Destination     Gateway          Genmask          Flags  Metric  Ref  Use  Iface
192.168.10.0    *                255.255.255.0    U      0       0    0    eth1
127.0.0.0       *                255.0.0.0        U      0       0    0    lo
default         192.168.10.1     0.0.0.0          UG     0       0    0    eth1
```

Network Layer

- Manual configuration of route tables: Command **route (or ip route)**
  - #route –n : shows route tables.
  - Add a route for a network segment:
    - # route add –net 192.168.1.0 netmask 255.255.255.0 eth0
  - Add the link element to other subnetworks (default route)
    - # route add default gw 192.168.1.1 eth0
- Dynamic routes (automatic)
  - Static configuration of tables limits their functionality.
    - Valid for stable networks (not very large…)
    - Requires knowledge about network topology.
  - Complex environments: Dynamic Routes
    - Daemon "routed" o "gated". OSFP, RIP, BGP, …
    - Maybe one of the most complex aspects concerning network administration.

Network Layer

- **Name Resolution**
  - name<->IP translation, the network phonebook:
  - Option 1: through the file **/etc/hosts**:
    - Conventional way, editing the file manually or through the command addhost.
    - Reasonable for small and private networks. Not useful for the rest of cases.
      - Adding a new host requires to modify all the /etc/hosts files in the network.
    - Usually employed only for the values required during boot process (localhost, hostname, …)
    - Can add the IPs of relevant network servers or those providing essential network services.
  - Option 2: **Domain Name Service** (DNS)
    - Dedicated server in charge of performing the conversion.
    - Each host must be configured to make use of its corresponding name server.
    - The client is configured through the file /etc/resolv.conf

Network Layer

- **Name Resolution**: the file /etc/resolv.conf:
  - **search**: domain search order
    - When we try to connect to a host without suffix, it auto-completes
    - ssh si -> ssh si.localdomain
    - Priority from left to right (first atc.unican.es, then unican.es)
  - **nameserver**: name server
    - Try to resolve with the first one.
    - If it fails, keep on descending to lower lines

```
search localdomain
search atc.unican.es unican.es

nameserver 193.144.193.11
nameserver 193.144.193.22
nameserver 192.168.0.105
```

  - The content in /etc/resolv.conf can be provided by dhcp server
    - It can also be created through /etc/network/interfaces: (resolvconf package required)

```
iface enp0s3 inet static
        dns-nameservers 1.1.1.1 2.2.2.2
```

Checking Network Status

1. **Check Interface**:
   - **Take a look at the definition**: /etc/network/interfaces
     - Edit the file and correct possible errors, restart then (ifdown/ifup, ifconfig [dev] down/up)
   - **Check the interface status:** ifconfig
2. **Check Network Layer:**
   - Check the **routes table** (is the gateway correctly defined?)
     - Edit the table with route command if necessary.
   - Check the **name resolution**
     - /etc/hosts and /etc/resolv.conf (is DNS correctly defined)
3. **Always, check the status**
   - Command **netstat:** shows network status.
   - Command **ping**: packet ECHO_REQUEST (ICMP) to a host (warning, firewall & ICMP).
   - Command **traceroute**: route followed by a packet towards destination
   - Try some command that requires network access (**apt-get update**).