

IP Basics

How the computer transport
system works

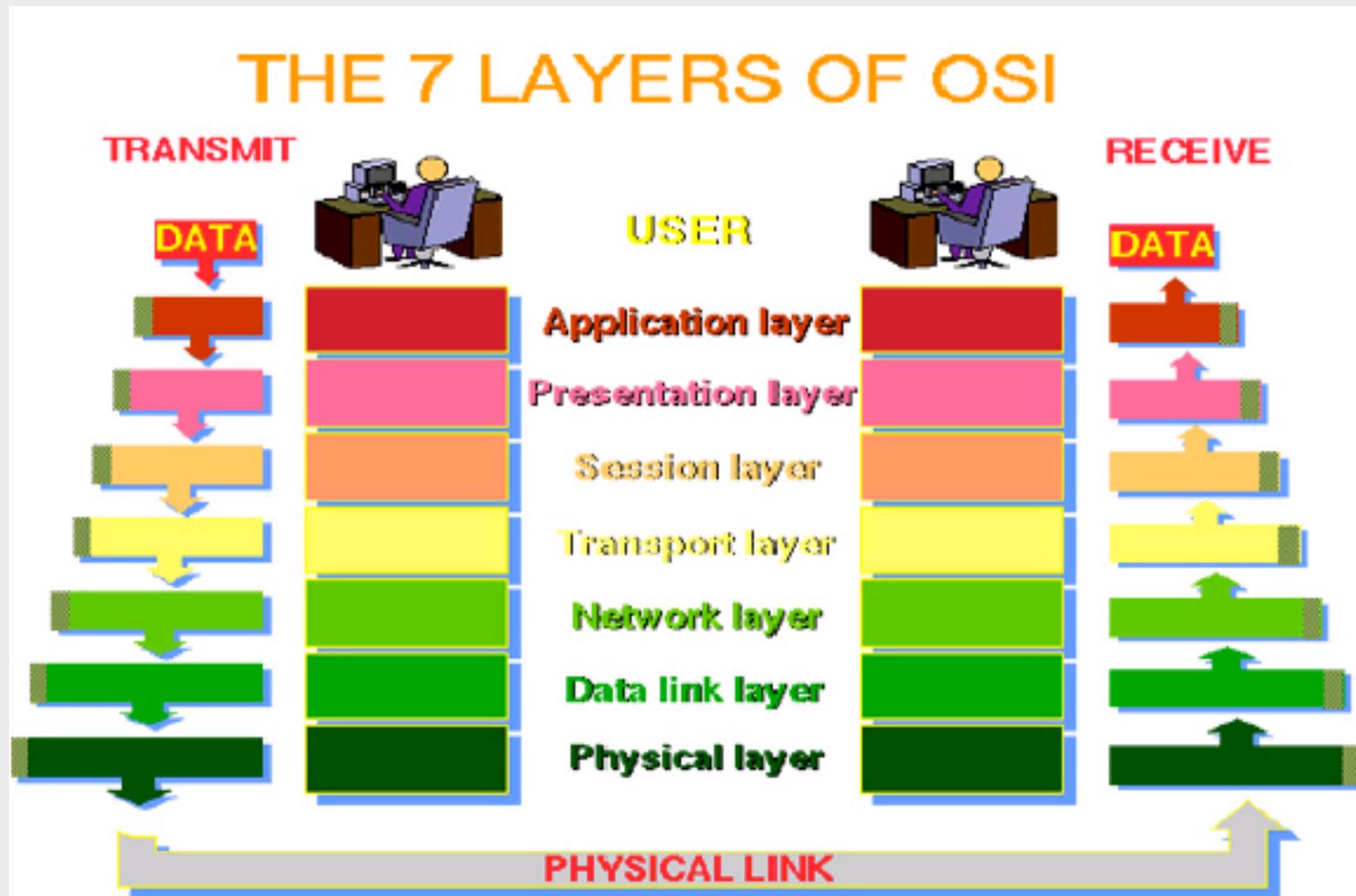
AFNOG X
Cairo, Egypt
May 2009



Layers

- Complex problems can be solved using the common divide and conquer principle. In this case the internals of the Internet are divided into separate layers.
 - Makes it easier to understand
 - Developments in one layer need not require changes in another layer
 - Easy formation (and quick testing of conformation to) standards
- Two main models of layers are used:
 - OSI (Open Systems Interconnection)
 - TCP/IP

OSI Model



OSI

- Conceptual model composed of seven layers, developed by the International Organization for Standardization (ISO) in 1984.

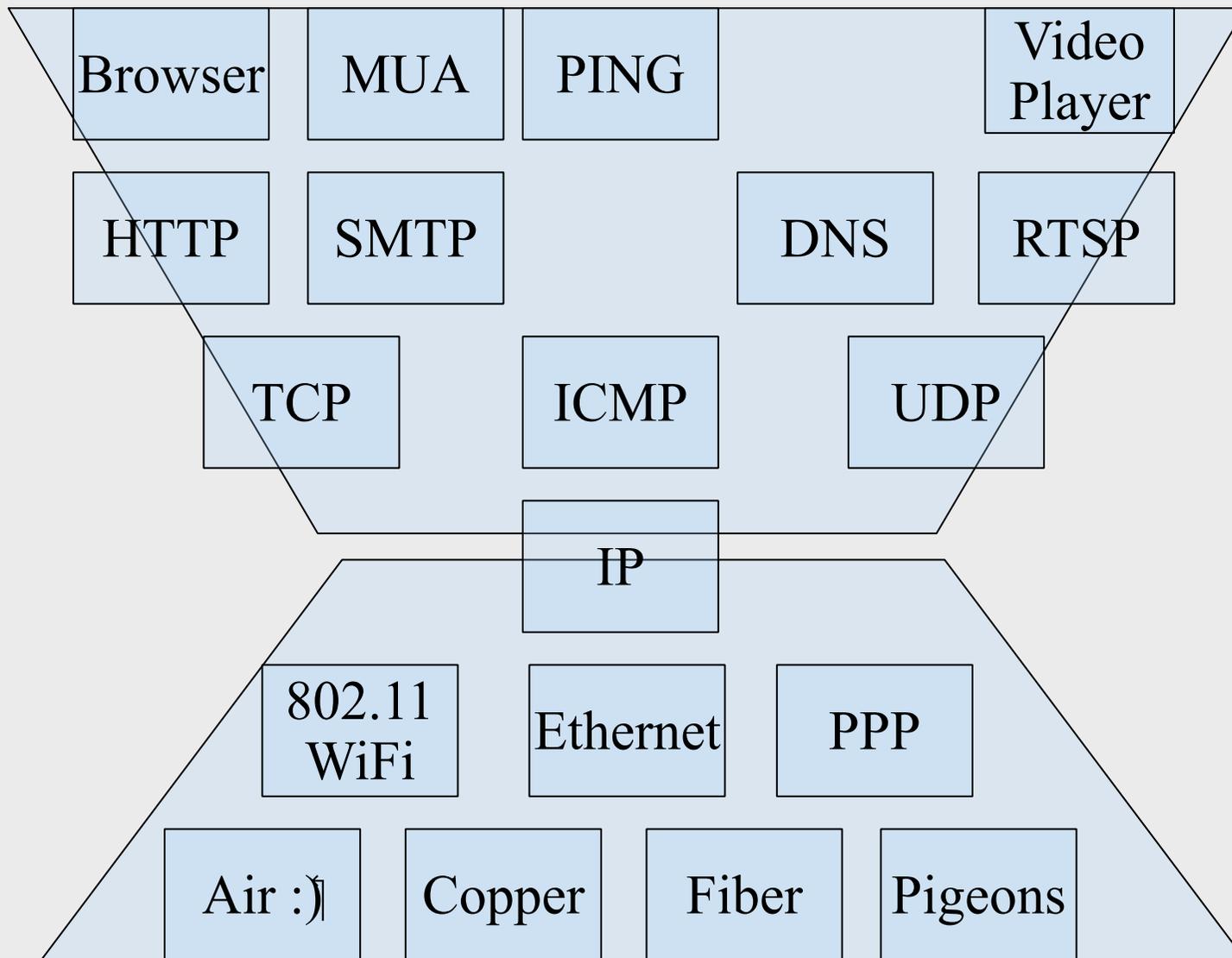
- Layer 7 – Application (servers and clients etc web browsers, httpd)
- Layer 6 – Presentation (file formats e.g pdf, ASCII, jpeg etc)
- Layer 5 – Session (conversation initialisation, termination,)
- Layer 4 – Transport (inter host comm – error correction, QOS)
- Layer 3 – Network (routing – path determination, IP[x] addresses etc)
- Layer 2 – Data link (switching – media acces, MAC addresses etc)
- Layer 1 – Physical (signalling – representation of binary digits)

- Acronym: All People Seem To Need Data Processing

TCP/IP

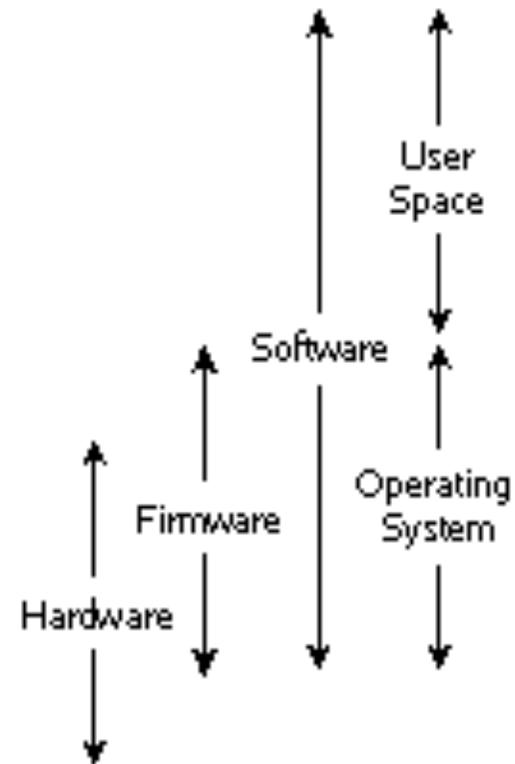
- Generally, TCP/IP (Transmission Control Protocol/Internet Protocol) is described using three to five functional layers. We have chosen the common DoD reference model, which is also known as the Internet reference model.
 - Process/Application Layer consists of applications and processes that use the network.
 - Host-to-host transport layer provides end-to-end data delivery services.
 - Internetwork layer defines the datagram and handles the routing of data.
 - Network access layer consists of routines for accessing physical networks.

TCP/IP model – the “hourglass”



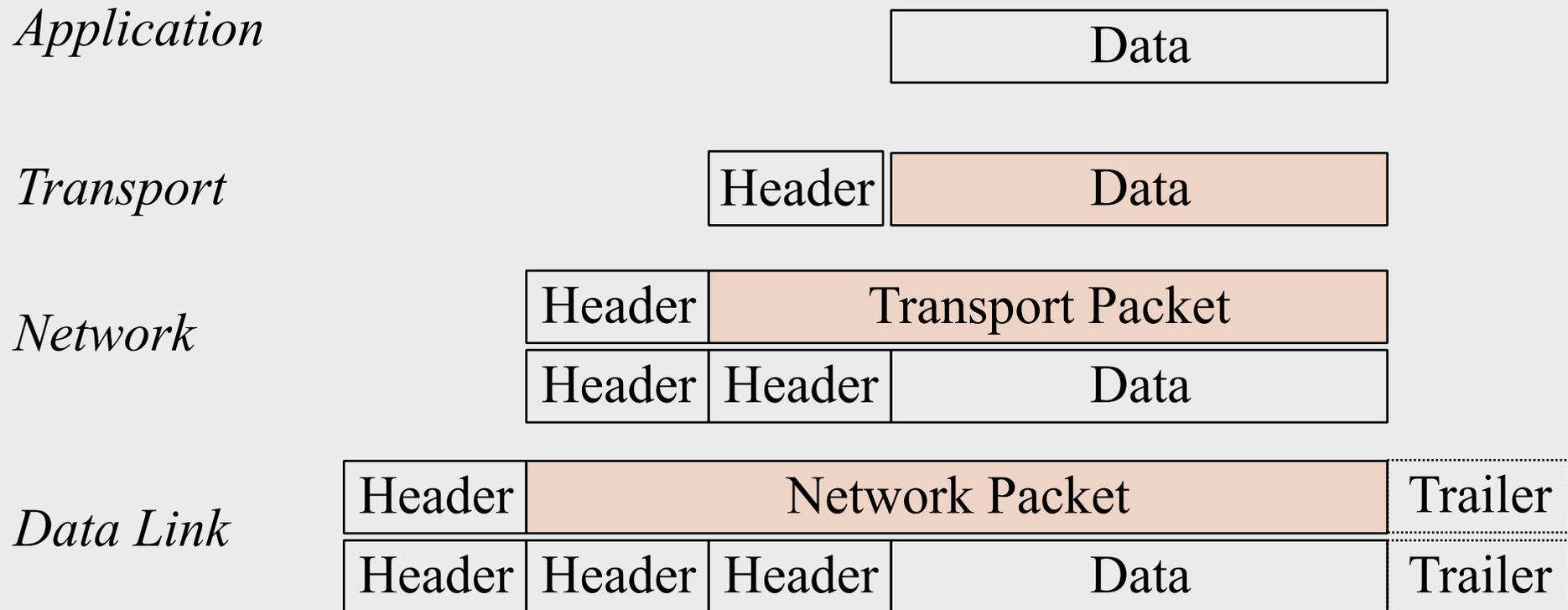
OSI and TCP/IP

TCP/IP	OSI
Application	Application
	Presentation
	Session
Trasport Host to host	Trasport
Internet	Network
Physical	Data link
	Physical



Encapsulation & Decapsulation

- Lower layers add headers (and sometimes trailers) to upper layers packets



Frame, Datagram, Segment, Packet

- Different names for packets at different layers
 - Ethernet (link layer) frame
 - IP (network layer) datagram
 - TCP (transport layer) segment
- Terminology is not strictly followed
 - we often just use the term “packet” at any layer

Summary

Networking is a problem approached in layers.

- OSI Layers

- TCP/IP Layers

Each layer adds headers to the packet of the previous layer as the data leaves the machine (encapsulation) and the reverse occurs on the receiving host (decapsulation)

So what is an IPv4 address anyway?

- 32 bit number (4 octet number) can be represented in lots of ways:

133	27	162	125
-----	----	-----	-----

10000101	00011011	10100010	01111101
----------	----------	----------	----------

85	1B	A2	7D
----	----	----	----

More to the structure

- Hierarchical Division in IP Address:

- Network Part (Prefix)

- describes which network

- Host Part (Host Address)

- describes which host on that network

205	.	154	.	8		1
11001101		10011010		00001000		00000001
Network					Mask	Host

- Boundary can be anywhere

- used to be a multiple of 8 (/8, /16/, /24), but not usual today

Network Masks

- **Network Masks** help define which bits are used to describe the **Network Part** and which for **hosts**
- Different Representations:
 - decimal dot notation: 255.255.224.0 (128+64+32 in byte 3)
 - binary: 11111111 11111111 111 00000 00000000
 - hexadecimal: 0xFFFFE000
 - number of network bits: /19 (8 + 8 + 3)
- Binary AND of 32 bit IP address with 32 bit **netmask** yields network part of address

Sample Netmasks

- 137.158.128.0/**17** (netmask **255.255.128.0**)

1111 1111	1111 1111	1	0000 0000	0000 0000
1000 1001	1001 1110	1	0000 0000	0000 0000

- 198.134.0.0/**16** (netmask **255.255.0.0**)

1111 1111	1111 1111		0000 0000	0000 0000
1100 0110	1000 0110		0000 0000	0000 0000

- 205.37.193.128/**26** (netmask **255.255.255.192**)

1111 1111	1111 1111	1111 1111	11	00 0000
1100 1101	0010 0101	1100 0001	10	00 0000

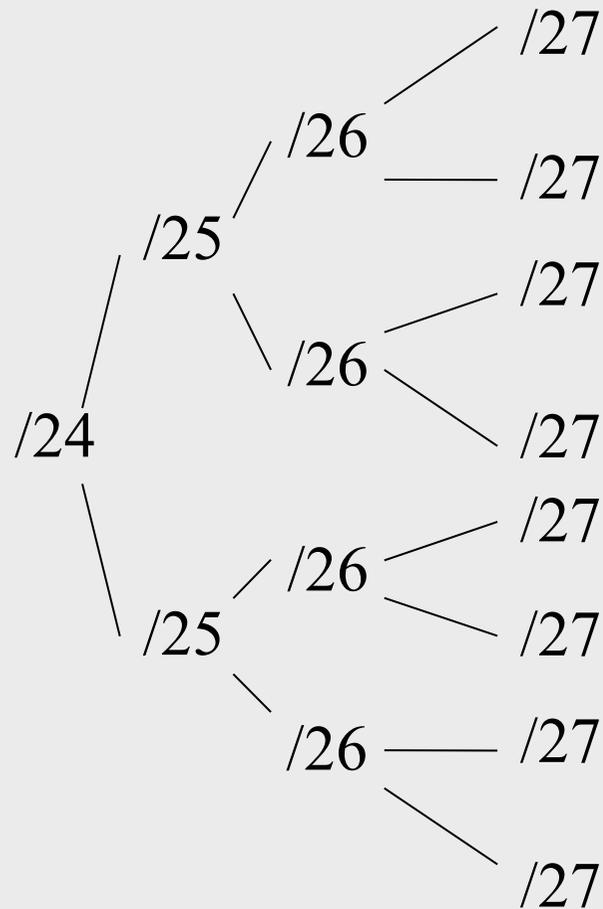
Allocating IP addresses

- The subnet mask is used to define size of a network
- E.g a subnet mask of 255.255.255.0 or /24 implies $32-24=8$ host bits
 - 2^8 minus 2 = 254 possible hosts
- Similarly a subnet mask of 255.255.255.224 or /27 implies $32-27=5$ hosts bits
 - 2^5 minus 2 = 30 possible hosts

Special IP Addresses

- All 0's in host part: Represents Network
 - e.g. 193.0.0.0/24
 - e.g. 138.37.128.0/17
 - e.g. 192.168.2.128/25 (WHY ?)
- All 1's in host part: **Broadcast** (all hosts on net)
 - e.g. 137.156.255.255 (137.156.0.0/16)
 - e.g. 134.132.100.255 (134.132.100.0/24)
 - e.g. 192.168.2.127/25 (192.168.2.0/25) (WHY ?)
- 127.0.0.0/8: **Loopback** address (127.0.0.1)
- 0.0.0.0: Various special purposes (DHCP, etc.)

Networks – super- and subnetting

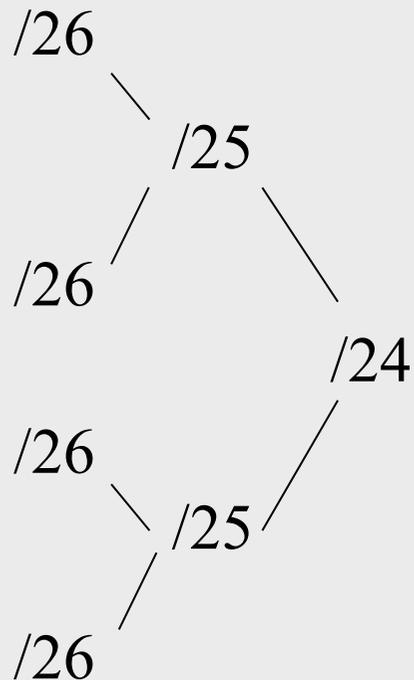


By adding one bit to the netmask, we subdivide the network into two smaller networks. This is *subnetting*.

i.e.: If one has a /26 network ($32 - 26 = 6 \Rightarrow 2^6 \Rightarrow 64$ addresses), that network can be subdivided into two subnets, using a /27 netmask, where the state of the last bit will determine which network we are addressing ($32 - 27 = 5 \Rightarrow 2^5 \Rightarrow 32$ addresses). This can be done recursively (/27 \Rightarrow 2 x /28 or 4 x /29, etc...).

Example: 192.168.10.0/25 (.0 - .127) can be subnetted into 192.168.10.0 / 26 and 192.168.10.64 / 26

Networks – super- and subnetting



Inversely, if two networks can be “joined” together under the same netmask, which encompasses both networks, then we are *supernetting*.

Example:

Networks 10.254.4.0/24 and 10.254.5.0/24 can be “joined” together into one network expressed: 10.254.4.0/23.

Note: for this to be possible, the networks must be *contiguous*, i.e. it is not possible to supernet 10.254.5.0/24 and 10.254.6.0/24

Numbering Rules

Private IP address ranges (RFC 1918)

-10/8 (10.0.0.0 – 10.255.255.255)

-192.168/16 (192.168.0.0 – 192.168.255.255)

-172.16/12 (172.16.0.0 – 172.31.255.255)

- Public Address space available from AfriNIC
- Choose a small block from whatever range you have, and subnet your networks (to avoid problems with broadcasts, and implement segmentation policies – DMZ, internal, etc...)

FreeBSD IP related settings

- `ifconfig_em0="196.200.219.x"`
- `defaultrouter="196.200.219.254"`
- `hostname="pcx.sse.mtg.afnog.org"`

Reaching hosts on the local net

- If you want to talk to other computers on the same network (e.g: within the same IP subnet, not necessarily the same physical network!), this is automatically possible the moment you assign an IP address to your network card.
- We will see this later with the hands-on

Reaching hosts on other networks

- If a computer isn't on your subnet, to reach it packets must be sent via a “gateway” connected to your network (“next hop”).
- If not explicit route (“direction”) is given on how to reach a particular network you want to talk to, then the computer will try a last resort “default gateway” for all packets that are not local
- defaultrouter option in `/etc/rc.conf` sets the default gateway for *this* system

Forwarding packets

- Any UNIX-like (and other) operating system can function as gateway (e.g.: forwarding packets from one interface to another)
- IP forwarding on a FreeBSD box turned on with the `gateway_enable` option in `/etc/rc.conf`
- Without forwarding enabled, the box will not forward packets from one interface to another: it is simply a host with multiple interfaces.

Packet Routing Exercise