

# Network Computing courses

Maël Auzias

ENSIBS - UBS

October 2014



Figure : [teaching.auzias.net](http://teaching.auzias.net)

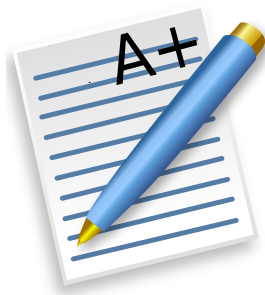
# Course details

## Objectives

- ▶ How do *computers* communicate?
- ▶ What are the mechanisms **under** an HTTP request or a telegram message?
- ▶ Networks are all around us, better study them!



# Course details



## Evaluation

- ▶ Short test at the beginning of every lesson (5 min) ?
- ▶ Project
- ▶ Final exam (1 hour)
- ▶ All same weighting

## Material

- ▶ Slides available at [teaching.auzias.net](https://teaching.auzias.net) (github too)

# Presentation Outline

Introduction

Physical

Data Link

Network

Transport

# Definitions and presentation

- ▶ **Network:** an **interconnected** group or system

# Definitions and presentation

- ▶ **Network:** an **interconnected** group or system
- ▶ **Internet:** world wide **interconnected system of networks**  
RFC791 (September 1981)

# Definitions and presentation

- ▶ **Network:** an **interconnected** group or system
- ▶ **Internet:** world wide **interconnected system of networks**  
[RFC791 \(September 1981\)](#)
- ▶ **IP:** Internet **Protocol** provides the functions necessary to deliver a package of bits from a source to a destination over a network

# Definitions and presentation

- ▶ **Network:** an **interconnected** group or system
- ▶ **Internet:** world wide **interconnected system of networks**  
[RFC791 \(September 1981\)](#)
- ▶ **IP:** Internet **Protocol** provides the functions necessary to deliver a package of bits from a source to a destination over a network
- ▶ **(world wide) Web: network** consisting of a collection of Internet websites using HTTP



# Definitions and presentation

- ▶ **HTTP:** Hypertext Transfer **Protocol**, application-level protocol for distributed, collaborative, hypermedia information systems [draft HTTP2 \(July 2014\)](#)

# Definitions and presentation

- ▶ **HTTP:** Hypertext Transfer **Protocol**, application-level protocol for distributed, collaborative, hypermedia information systems [draft HTTP2 \(July 2014\)](#)
- ▶ **FTP:** File Transfer **Protocol** promotes sharing of files, encourages the use of remote computers [RFC959 \(October 1985\)](#)

## Definitions and presentation

- ▶ **HTTP:** Hypertext Transfer **Protocol**, application-level protocol for distributed, collaborative, hypermedia information systems [draft HTTP2 \(July 2014\)](#)
- ▶ **FTP:** File Transfer **Protocol** promotes sharing of files, encourages the use of remote computers [RFC959 \(October 1985\)](#)
- ▶ **TCP:** Transmission Control **Protocol** is intended for use as a highly reliable host-to-host [RFC761 \(January 1980\)](#)

# Definitions and presentation

- ▶ **HTTP:** Hypertext Transfer **Protocol**, application-level protocol for distributed, collaborative, hypermedia information systems [draft HTTP2 \(July 2014\)](#)
- ▶ **FTP:** File Transfer **Protocol** promotes sharing of files, encourages the use of remote computers [RFC959 \(October 1985\)](#)
- ▶ **TCP:** Transmission Control **Protocol** is intended for use as a highly reliable host-to-host [RFC761 \(January 1980\)](#)
- ▶ **UDP:** User Datagram **Protocol** provides a procedure for application programs to send messages to other programs with a minimum of protocol mechanism [RFC768 \(August 1980\)](#)

# Definitions and presentation

- ▶ **HTTP:** Hypertext Transfer **Protocol**, application-level protocol for distributed, collaborative, hypermedia information systems [draft HTTP2 \(July 2014\)](#)
- ▶ **FTP:** File Transfer **Protocol** promotes sharing of files, encourages the use of remote computers [RFC959 \(October 1985\)](#)
- ▶ **TCP:** Transmission Control **Protocol** is intended for use as a highly reliable host-to-host [RFC761 \(January 1980\)](#)
- ▶ **UDP:** User Datagram **Protocol** provides a procedure for application programs to send messages to other programs with a minimum of protocol mechanism [RFC768 \(August 1980\)](#)
- ▶ **RFC:** Request For Comments (Internet Draft (ID), RFC, Internet Standard)

## Definitions and presentation

- ▶ **Router:** network **hardware** providing routing services

## Definitions and presentation

- ▶ **Router:** network **hardware** providing routing services
- ▶ **Routing:** **algorithm processed** to decide where to forward a packet

# Definitions and presentation

- ▶ **Router:** network **hardware** providing routing services
- ▶ **Routing:** **algorithm processed** to decide where to forward a packet
- ▶ **Forwarding:** **action** of moving a packet from one NIC to another



## Definitions and presentation

- ▶ **Router:** network **hardware** providing routing services
- ▶ **Routing: algorithm processed** to decide where to forward a packet
- ▶ **Forwarding: action** of moving a packet from one NIC to another
- ▶ **NIC:** Network Interface Card
- ▶ **Switch (hub):** network **hardware** connecting systems using packet switching

## Definitions and presentation

- ▶ **Router:** network **hardware** providing routing services
- ▶ **Routing:** **algorithm processed** to decide where to forward a packet
- ▶ **Forwarding:** **action** of moving a packet from one NIC to another
- ▶ **NIC:** Network Interface Card
- ▶ **Switch (hub):** network **hardware** connecting systems using packet switching
- ▶ **Packet switching:** forward-like method regardless of the content (destination-based)

## Definitions and presentation

- ▶ **Router:** network **hardware** providing routing services
- ▶ **Routing:** **algorithm processed** to decide where to forward a packet
- ▶ **Forwarding:** **action** of moving a packet from one NIC to another
- ▶ **NIC:** Network Interface Card
- ▶ **Switch (hub):** network **hardware** connecting systems using packet switching
- ▶ **Packet switching:** forward-like method regardless of the content (destination-based)
- ▶ **NAT:** Network Address Translation, router modifying IP address into another IP address.

## Definitions and presentation

- ▶ **Node (network):** any entity that can send packets to/receive packets from a network through a NIC

# Definitions and presentation

- ▶ **Node (network):** any entity that can send packets to/receive packets from a network through a NIC
- ▶ **Client: computer** able to send requests to a server

# Definitions and presentation

- ▶ **Node (network):** any entity that can send packets to/receive packets from a network through a NIC
- ▶ **Client: computer** able to send requests to a server
- ▶ **Request: application message** destined for a server (*order*)

# Definitions and presentation

- ▶ **Node (network):** any entity that can send packets to/receive packets from a network through a NIC
- ▶ **Client: computer** able to send requests to a server
- ▶ **Request: application message** destined for a server (*order*)
- ▶ **Server: computer** able to respond a client's requests

## Definitions and presentation

- ▶ **Node (network):** any entity that can send packets to/receive packets from a network through a NIC
- ▶ **Client: computer** able to send requests to a server
- ▶ **Request: application message** destined for a server (*order*)
- ▶ **Server: computer** able to respond a client's requests
- ▶ **Response: application message** destined for a client (*result*)



## Definitions and presentation

- ▶ **Node (network):** any entity that can send packets to/receive packets from a network through a NIC
- ▶ **Client: computer** able to send requests to a server
- ▶ **Request: application message** destined for a server (*order*)
- ▶ **Server: computer** able to respond a client's requests
- ▶ **Response: application message** destined for a client (*result*)
- ▶ **Fat client: application** where most functions are processed by the client itself

## Definitions and presentation

- ▶ **Node (network):** any entity that can send packets to/receive packets from a network through a NIC
- ▶ **Client: computer** able to send requests to a server
- ▶ **Request: application message** destined for a server (*order*)
- ▶ **Server: computer** able to respond a client's requests
- ▶ **Response: application message** destined for a client (*result*)
- ▶ **Fat client: application** where most functions are processed by the client itself
- ▶ **Thin client: application** where most functions are carried out on a central server

# Network classification

- ▶ **BAN:** Body Area Network

# Network classification

- ▶ **BAN:** Body Area Network
- ▶ **PAN:** Personal Area Networks

# Network classification

- ▶ **BAN:** Body Area Network
- ▶ **PAN:** Personal Area Networks
- ▶ **(W)LAN:** (Wireless) Local Area Networks (home, office, school or airport)

# Network classification

- ▶ **BAN:** Body Area Network
- ▶ **PAN:** Personal Area Networks
- ▶ **(W)LAN:** (Wireless) Local Area Networks (home, office, school or airport)
- ▶ **MAN:** Metropolitan Area Networks, can cover a whole city

# Network classification

- ▶ **BAN:** Body Area Network
- ▶ **PAN:** Personal Area Networks
- ▶ **(W)LAN:** (Wireless) Local Area Networks (home, office, school or airport)
- ▶ **MAN:** Metropolitan Area Networks, can cover a whole city
- ▶ **WAN:** Wide Area Networks cover a broad area (Internet)

# Topologies

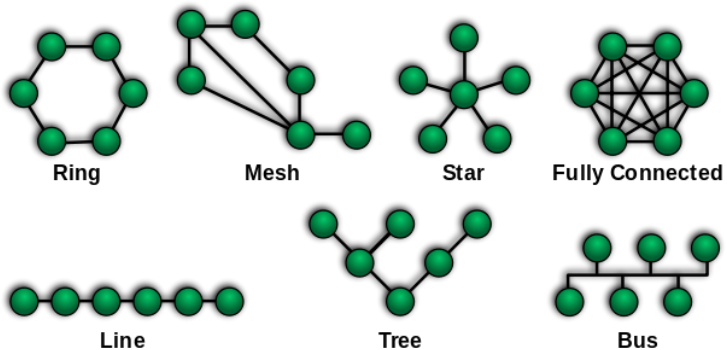


Figure : [upload.wikimedia.org](http://upload.wikimedia.org)



# Topologies

- ▶ **Point-to-point:** two entities directly connected to each other (tunnel).

# Topologies

- ▶ **Point-to-point:** two entities directly connected to each other (tunnel).
- ▶ **Ring:** data go around the ring, unidirectional way network.

# Topologies

- ▶ **Point-to-point:** two entities directly connected to each other (tunnel).
- ▶ **Ring:** data go around the ring, unidirectional way network.
- ▶ **Mesh:** all nodes cooperate in the distribution of data in the network<sup>1</sup>.

---

<sup>1</sup>Hong Kong protesters use a mesh network to organize 

# Topologies

- ▶ **Point-to-point:** two entities directly connected to each other (tunnel).
- ▶ **Ring:** data go around the ring, unidirectional way network.
- ▶ **Mesh:** all nodes cooperate in the distribution of data in the network<sup>1</sup>.
- ▶ **Star:** all messages go through the same central node, reducing network failure.

---

<sup>1</sup>Hong Kong protesters use a mesh network to organize 

# Topologies

- ▶ **Point-to-point:** two entities directly connected to each other (tunnel).
- ▶ **Ring:** data go around the ring, unidirectional way network.
- ▶ **Mesh:** all nodes cooperate in the distribution of data in the network<sup>1</sup>.
- ▶ **Star:** all messages go through the same central node, reducing network failure.
- ▶ **Fully connected:** all nodes are connected to all other nodes.

---

<sup>1</sup>Hong Kong protesters use a mesh network to organize 

# Topologies

- ▶ **Point-to-point:** two entities directly connected to each other (tunnel).
- ▶ **Ring:** data go around the ring, unidirectional way network.
- ▶ **Mesh:** all nodes cooperate in the distribution of data in the network<sup>1</sup>.
- ▶ **Star:** all messages go through the same central node, reducing network failure.
- ▶ **Fully connected:** all nodes are connected to all other nodes.
- ▶ **Line:** bidirectional link between two nodes. Node can only send packet going through its neighbors.

---

<sup>1</sup>Hong Kong protesters use a mesh network to organize 

# Topologies

- ▶ **Point-to-point:** two entities directly connected to each other (tunnel).
- ▶ **Ring:** data go around the ring, unidirectional way network.
- ▶ **Mesh:** all nodes cooperate in the distribution of data in the network<sup>1</sup>.
- ▶ **Star:** all messages go through the same central node, reducing network failure.
- ▶ **Fully connected:** all nodes are connected to all other nodes.
- ▶ **Line:** bidirectional link between two nodes. Node can only send packet going through its neighbors.
- ▶ **Bus:** all nodes are connected to the same media. Only one can send a packet at a time, which all others then receive.

---

<sup>1</sup>[Hong Kong protesters use a mesh network to organize](#) 

# Topologies

- ▶ **Point-to-point:** two entities directly connected to each other (tunnel).
- ▶ **Ring:** data go around the ring, unidirectional way network.
- ▶ **Mesh:** all nodes cooperate in the distribution of data in the network<sup>1</sup>.
- ▶ **Star:** all messages go through the same central node, reducing network failure.
- ▶ **Fully connected:** all nodes are connected to all other nodes.
- ▶ **Line:** bidirectional link between two nodes. Node can only send packet going through its neighbors.
- ▶ **Bus:** all nodes are connected to the same media. Only one can send a packet at a time, which all others then receive.
- ▶ **Tree:** hierarchical topology, such as a binary tree.

---

<sup>1</sup>[Hong Kong protesters use a mesh network to organize](#) 



# Bonus

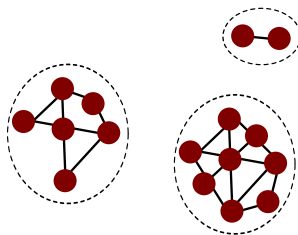


Figure : Disconnected MANET illustration [?]

# Bonus

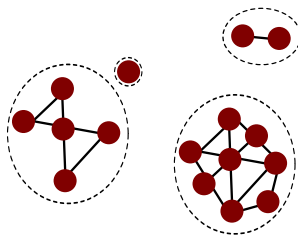


Figure : Store-carry-and-forward [?]

# Bonus

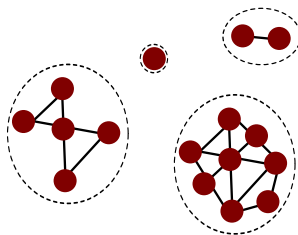


Figure : Store-carry-and-forward [?]

# Bonus

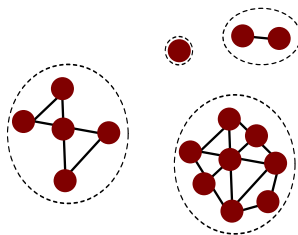


Figure : Store-carry-and-forward [?]

# Bonus

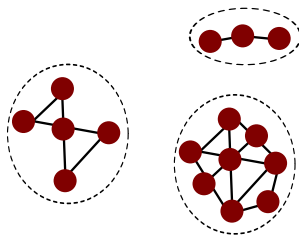


Figure : Store-carry-and-forward [?]

# HTTP request/response example

Enter [getbootstrap.com](https://getbootstrap.com) in your browser

# HTTP request/response example

Enter [getbootstrap.com](http://getbootstrap.com) in your browser

Source	Destination	Protocol	Length	Info
192.168.0.48	208.67.222.222	DNS	76	Standard query 0x4797 A getbootstrap.com
208.67.222.222	192.168.0.48	DNS	108	Standard query response 0x4797 A 192.30.252.154 A 192.30.252.153

Figure : DNS request/response

# HTTP request/response example

Enter [getbootstrap.com](http://getbootstrap.com) in your browser

Source	Destination	Protocol	Length	Info
192.168.0.48	208.67.222.222	DNS	76	Standard query 0x4797 A getbootstrap.com
208.67.222.222	192.168.0.48	DNS	108	Standard query response 0x4797 A 192.30.252.154 A 192.30.252.153

Figure : DNS request/response

Source	Destination	Protocol	Length	Info
127.0.0.1	127.0.0.13	TCP	74	36159 > http [SYN] Seq=0 Win=43690 Len=0 MSS=65495 SACK_PERM=1 TSval=12
127.0.0.13	127.0.0.1	TCP	74	http > 36159 [ACK] Seq=0 Ack=1 Win=43690 Len=0 MSS=65495 SACK_PERM
127.0.0.1	127.0.0.13	TCP	66	36159 > http [ACK] Seq=1 Ack=1 Win=43776 Len=0 TSval=122257 TSecr=12225
127.0.0.1	127.0.0.13	HTTP	356	GET /index.html HTTP/1.1
127.0.0.13	127.0.0.1	TCP	66	http > 36159 [ACK] Seq=1 Ack=291 Win=44800 Len=0 TSval=122259 TSecr=122
127.0.0.13	127.0.0.1	HTTP	354	HTTP/1.1 200 OK (text/html)
127.0.0.1	127.0.0.13	TCP	66	36159 > http [ACK] Seq=291 Ack=289 Win=44800 Len=0 TSval=122259 TSecr=1
127.0.0.1	127.0.0.13	HTTP	357	GET /favicon.ico HTTP/1.1
127.0.0.13	127.0.0.1	HTTP	565	HTTP/1.1 404 Not Found (text/html)
127.0.0.1	127.0.0.13	TCP	66	36159 > http [ACK] Seq=582 Ack=788 Win=45952 Len=0 TSval=122269 TSecr=1

Figure : HTTP request/response



# How do messages reach their destination?

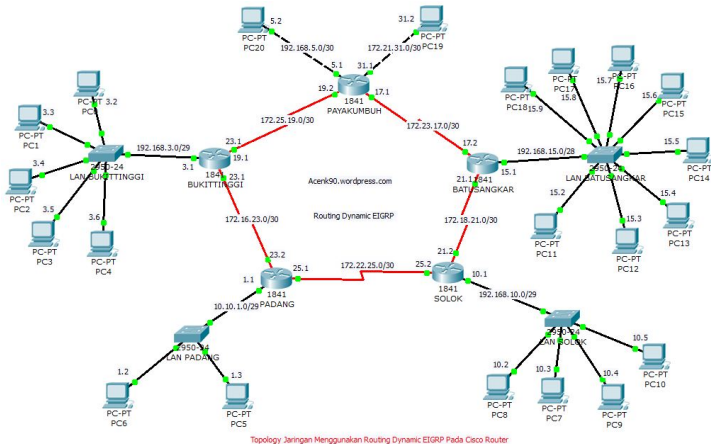
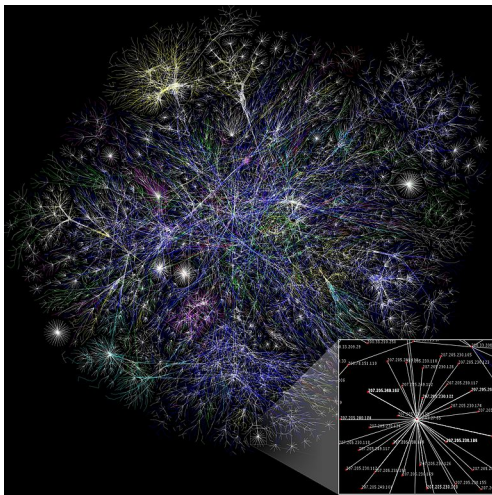


Figure : [acenk90.files.wordpress.com](http://acenk90.files.wordpress.com)

More like this...



## Models overview (OSI and TCP/IP)

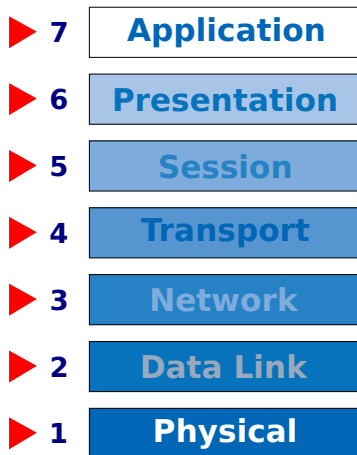
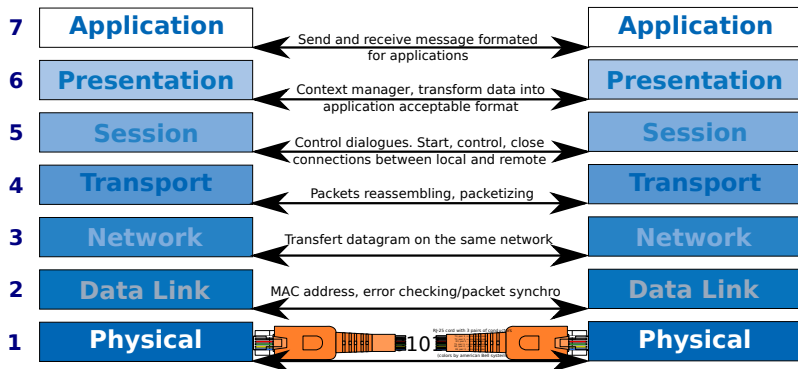
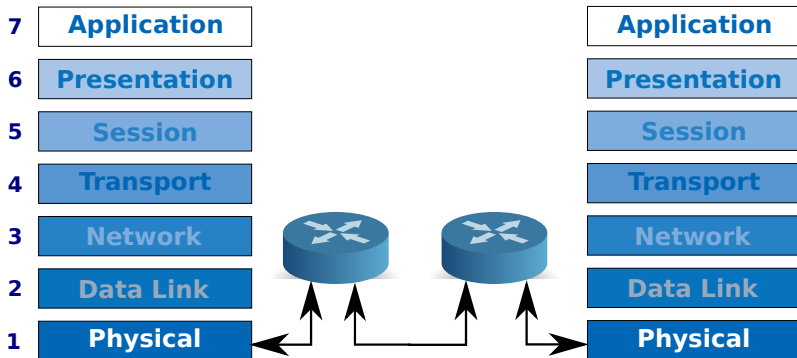


Figure : OSI model

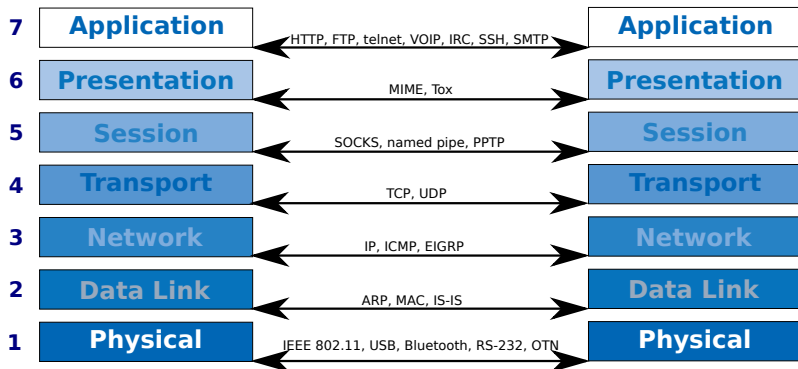
# $N^{\text{th}}$ layer communicate with $N^{\text{th}}$ layer..



.. thanks to 3<sup>th</sup> layers



# One single protocol, one single layer



# Encapsulation

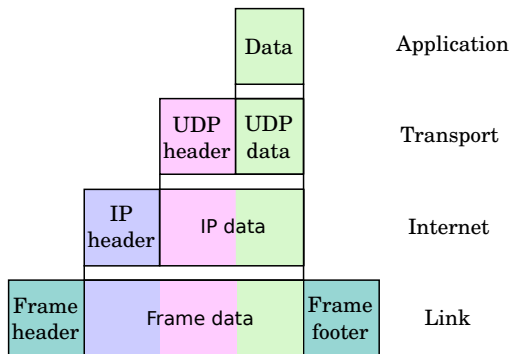


Figure : Encapsulation

# Presentation Outline

Introduction

Physical

Data Link

Network

Transport



# Aims

- ▶ Interface data link layer,

# Aims

- ▶ Interface data link layer,
- ▶ (De)Encode,

# Aims

- ▶ Interface data link layer,
- ▶ (De)Encode,
- ▶ Transmit: 1 after 0 (after 0 or 1, after 0... or 1)

## Hardware medium

- ▶ IEEE 802.3 (a.k.a. Ethernet):  $< 100\text{Gbit/s}$

## Hardware medium

- ▶ IEEE 802.3 (a.k.a. Ethernet): <100Gbit/s
- ▶ IEEE 802.11 (a.k.a. Wi-Fi): <50 Mbit/s (802.11ad goes up to 6.75 Gbit/s)

## Hardware medium

- ▶ IEEE 802.3 (a.k.a. Ethernet):  $< 100 \text{ Gbit/s}$
- ▶ IEEE 802.11 (a.k.a. Wi-Fi):  $< 50 \text{ Mbit/s}$  (802.11ad goes up to  $6.75 \text{ Gbit/s}$ )
- ▶ IEEE 802.15.1 (a.k.a. Bluetooth):  $< 1 \text{ Mbit/s}$

## Hardware medium

- ▶ IEEE 802.3 (a.k.a. Ethernet): <100Gbit/s
- ▶ IEEE 802.11 (a.k.a. Wi-Fi): <50 Mbit/s (802.11ad goes up to 6.75 Gbit/s)
- ▶ IEEE 802.15.1 (a.k.a. Bluetooth): <1 Mbit/s
- ▶ IEEE 802.15.4 (a.k.a. ZigBee): <250 kbit/s

## Hardware medium

- ▶ IEEE 802.3 (a.k.a. Ethernet):  $< 100 \text{ Gbit/s}$
- ▶ IEEE 802.11 (a.k.a. Wi-Fi):  $< 50 \text{ Mbit/s}$  (802.11ad goes up to  $6.75 \text{ Gbit/s}$ )
- ▶ IEEE 802.15.1 (a.k.a. Bluetooth):  $< 1 \text{ Mbit/s}$
- ▶ IEEE 802.15.4 (a.k.a. ZigBee):  $< 250 \text{ kbit/s}$
- ▶ IEEE 802.16 (a.k.a. Wi-Max):  $< 40 \text{ Mbit/s}$



## Hardware medium

- ▶ IEEE 802.3 (a.k.a. Ethernet): <100Gbit/s
- ▶ IEEE 802.11 (a.k.a. Wi-Fi): <50 Mbit/s (802.11ad goes up to 6.75 Gbit/s)
- ▶ IEEE 802.15.1 (a.k.a. Bluetooth): <1 Mbit/s
- ▶ IEEE 802.15.4 (a.k.a. ZigBee): <250 kbit/s
- ▶ IEEE 802.16 (a.k.a. Wi-Max): <40 Mbit/s
- ▶ IEEE 1394 (a.k.a. Firewire): <3200 Mbit/s

## Hardware medium

- ▶ IEEE 802.3 (a.k.a. Ethernet): <100Gbit/s
- ▶ IEEE 802.11 (a.k.a. Wi-Fi): <50 Mbit/s (802.11ad goes up to 6.75 Gbit/s)
- ▶ IEEE 802.15.1 (a.k.a. Bluetooth): <1 Mbit/s
- ▶ IEEE 802.15.4 (a.k.a. ZigBee): <250 kbit/s
- ▶ IEEE 802.16 (a.k.a. Wi-Max): <40 Mbit/s
- ▶ IEEE 1394 (a.k.a. Firewire): <3200 Mbit/s
- ▶ USB, serial port such as RS-232...

## Hardware medium: IEEE 802.3 (Ethernet)



Figure : RJ45 connector

## Hardware medium: IEEE 802.15.1 (Bluetooth)

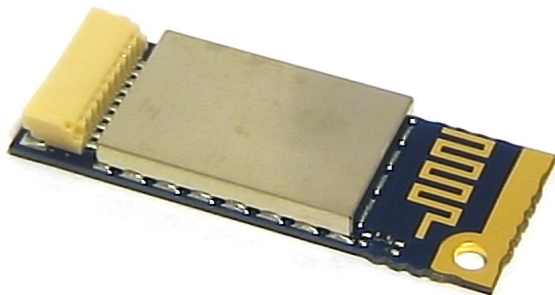


Figure : Bluetooth card

## Hardware medium: IEEE 802.15.4 (ZigBee)

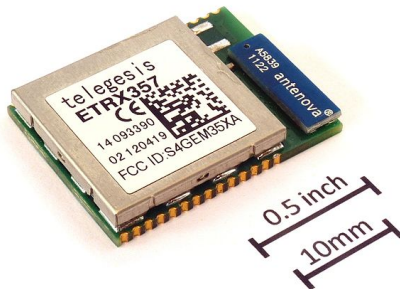


Figure : ZigBee card

## Hardware medium: IEEE 802.16 (Wi-Max)



Figure : Wi-Max antenna

## Hardware medium: IEEE 1394 (Firewire)



Figure : Firewire connector

# Encoding

- ▶ **MLT3 (Multi-Level Transmit):** state change for 1s over 3 levels, stay in the same state for 0s



# Encoding

- ▶ **MLT3 (Multi-Level Transmit):** state change for 1s over 3 levels, stay in the same state for 0s
- ▶ **AMI (Alternate Mark Inversion):** state 0 for 0s, state  $+/-1$  for 1s

# Encoding

- ▶ **MLT3 (Multi-Level Transmit):** state change for 1s over 3 levels, stay in the same state for 0s
- ▶ **AMI (Alternate Mark Inversion):** state 0 for 0s, state  $\pm 1$  for 1s
- ▶ **Manchester:** voltage transition (rising/falling edge mean 1/0)

# Encoding

- ▶ **MLT3 (Multi-Level Transmit):** state change for 1s over 3 levels, stay in the same state for 0s
- ▶ **AMI (Alternate Mark Inversion):** state 0 for 0s, state  $+/-1$  for 1s
- ▶ **Manchester:** voltage transition (rising/falling edge mean 1/0)
- ▶ **BMC (Biphase Mark Code):** change its state for 1s, stay on the same state for 0s

# Encoding

- ▶ **MLT3 (Multi-Level Transmit):** state change for 1s over 3 levels, stay in the same state for 0s
- ▶ **AMI (Alternate Mark Inversion):** state 0 for 0s, state  $+/-1$  for 1s
- ▶ **Manchester:** voltage transition (rising/falling edge mean 1/0)
- ▶ **BMC (Biphase Mark Code):** change its state for 1s, stay on the same state for 0s
- ▶ and so on...

## Encoding: Multi-Level Transmit

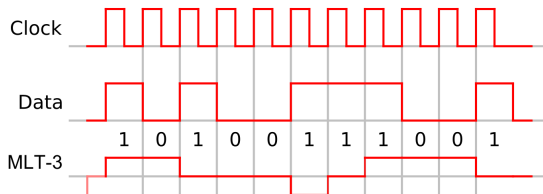


Figure : Multi-Level Transmit

## Encoding: Alternate Mark Inversion

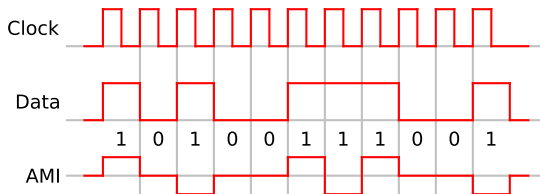


Figure : Alternate Mark Inversion

# Encoding: Manchester

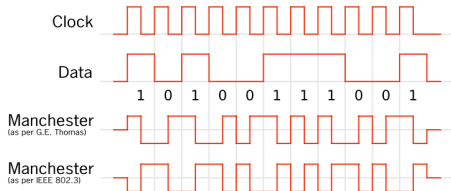


Figure : Manchester

## Encoding: Biphase Mark Code

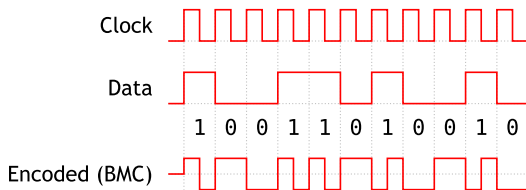


Figure : Biphase Mark Code



# Transmitting

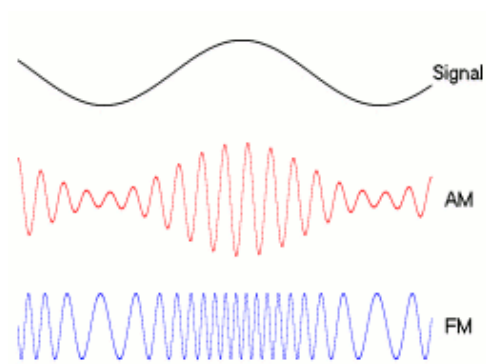


Figure : Amplitude and phase modulation

# Error detection

- ▶ Repetition (hum...)

# Error detection

- ▶ Repetition (hum...)
- ▶ Parity (XOR)

# Error detection

- ▶ Repetition (hum...)
- ▶ Parity (XOR)
- ▶ Checksum

# Error detection

- ▶ Repetition (hum...)
- ▶ Parity (XOR)
- ▶ Checksum
- ▶ CRC (Cyclic redundancy check): with a polynomial division

# Error detection

- ▶ Repetition (hum...)
- ▶ Parity (XOR)
- ▶ Checksum
- ▶ CRC (Cyclic redundancy check): with a polynomial division
- ▶ Hash

## Error detection

- ▶ Repetition (hum...)
- ▶ Parity (XOR)
- ▶ Checksum
- ▶ CRC (Cyclic redundancy check): with a polynomial division
- ▶ Hash
- ▶ and so on...

# Error correcting

- ▶ Repetition (again)



# Error correcting

- ▶ Repetition (again)
- ▶ Hamming

# Error correcting

- ▶ Repetition (again)
- ▶ Hamming
- ▶ MDPC (Multidimensional parity-check code)

## Correction: MDPC

Raw data to send: 0x01 02 03 04

0x01	0x02	0x03
0x03	0x04	0x07
0x04	0x06	

Figure : Data received with MDPC

Data sent (with MDPC): 0x01 02 03 03 04 07 04 06

# Presentation Outline

Introduction

Physical

Data Link

Network

Transport

# Aims

- ▶ Interface network layer,

# Aims

- ▶ Interface network layer,
- ▶ Delivery to unique(?) hardware addresses,

# Aims

- ▶ Interface network layer,
- ▶ Delivery to unique(?) hardware addresses,
- ▶ Framing,

# Aims

- ▶ Interface network layer,
- ▶ Delivery to unique(?) hardware addresses,
- ▶ Framing,
- ▶ Data transfer



## Layer composition (of its two sublayers)

1. Logical Link Control (LLC):
  - ▶ end to end flow control
  - ▶ end to end error control
  - ▶ (transmitting/receiving) protocols, over MAC sublayer, multiplexing

## Layer composition (of its two sublayers)

1. Logical Link Control (LLC):
  - ▶ end to end flow control
  - ▶ end to end error control
  - ▶ (transmitting/receiving) protocols, over MAC sublayer, multiplexing
2. Media Access Control (MAC):
  - ▶ physical (hardware) addressing
  - ▶ collision detection and retransmission
  - ▶ data packet scheduling (and queuing)
  - ▶ QoS
  - ▶ VLAN

# Carrier Sense Multiple Access with Collision Avoidance

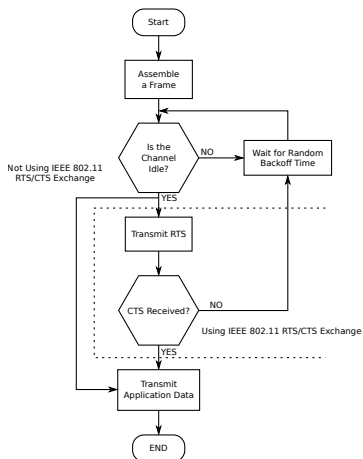


Figure : CSMA CA

## Layer 2 Ethernet packet

MAC dest. (6)	MAC src. (6)	VLAN tag* (4)	Ethertype (2)
Payload (42-1500)		Frame check sequence (4)	

Figure : Layer 2 Ethernet packet

optional, Content (size in bytes)

Ethertype 0x	Protocol
0800	IPv4
0806	ARP
0842	Wake-on-LAN
86dd	IPv6

Figure : Data received with MDPC

## ARP example

<b>0000</b>	ff	ff	ff	ff	ff	ff	fa	ba	00	ab	ab	af	08	06	00	01
<b>0010</b>	08	00	06	04	00	01	fa	ba	00	ab	ab	af	ac	11	22	37
<b>0020</b>	00	00	00	00	00	00	ac	11	00	f9	00	00	00	00	00	00
<b>0030</b>	00	00	00	00	00	00	00	00	00	00	00	00				

Figure : ARP request

MAC address destination MAC address source Ethertype Hardware  
 type Protocol type OpCode (1 request, 2 reply) IP address source  
 IP address destination

# ARP example

<b>0000</b>	ff	ff	ff	ff	ff	ff	fa	ba	00	ab	ab	af	08	06	00	01
<b>0010</b>	08	00	06	04	00	01	fa	ba	00	ab	ab	af	ac	11	22	37
<b>0020</b>	00	00	00	00	00	00	ac	11	00	f9	00	00	00	00	00	00
<b>0030</b>	00	00	00	00	00	00	00	00	00	00	00	00				

Figure : ARP request

MAC address destination MAC address source Ethertype Hardware  
 type Protocol type OpCode (1 request, 2 reply) IP address source  
 IP address destination

## ARP example

<b>0000</b>	fa	ba	00	ab	ab	af	be	be	00	00	eb	eb	08	06	00	01
<b>0010</b>	08	00	06	04	00	01	be	be	00	00	eb	eb	ac	11	00	f9
<b>0020</b>	fa	ba	00	ab	ab	af	ac	11	22	37	00	00	00	00	00	00
<b>0030</b>	00	00	00	00	00	00	00	00	00	00	00	00				

Figure : ARP reply

MAC address destination MAC address source Ethertype Hardware  
 type Protocol type OpCode (1 request, 2 reply) IP address source  
 IP address destination

# ARP example

<b>0000</b>	fa	ba	00	ab	ab	af	be	be	00	00	eb	eb	08	06	00	01
<b>0010</b>	08	00	06	04	00	01	be	be	00	00	eb	eb	ac	11	00	f9
<b>0020</b>	fa	ba	00	ab	ab	af	ac	11	22	37	00	00	00	00	00	00
<b>0030</b>	00	00	00	00	00	00	00	00	00	00	00	00				

Figure : ARP reply

MAC address destination MAC address source Ethertype Hardware  
 type Protocol type OpCode (1 request, 2 reply) IP address source  
 IP address destination



# Presentation Outline

Introduction

Physical

Data Link

**Network**

Transport

# Aims

- ▶ Interface transport layer,

# Aims

- ▶ Interface transport layer,
- ▶ Host addressing,

# Aims

- ▶ Interface transport layer,
- ▶ Host addressing,
- ▶ End-to-end packet transmission (data link? Connectionless? Switch? Router?),

# Aims

- ▶ Interface transport layer,
- ▶ Host addressing,
- ▶ End-to-end packet transmission (data link? Connectionless? Switch? Router?),
- ▶ Routing, load balancing

# Concepts

- ▶ IP addressing fundamentals,
- ▶ Classfull IP addressing,
- ▶ Subnet and VLSM (Variable length subnet masks),
- ▶ CIDR (Classless inter-domain routing),
- ▶ Routing,
- ▶ IPv6.

# IP addressing fundamentals

## IP address

32 bits (4x4 bytes)

mask	
Networks part	Host part

Figure : IP address parts

# IP addressing fundamentals

## Masks

- ▶ Separates **network** and **host** bits,



# IP addressing fundamentals

## Masks

- ▶ Separates **network** and **host** bits,
- ▶ MSB **always** are ones and then zeros! 255.254.255.0 is not possible,

# IP addressing fundamentals

## Masks

- ▶ Separates **network** and **host** bits,
- ▶ MSB **always** are ones and then zeros! 255.254.255.0 is not possible,
- ▶ Indicates how many bits are used for the **network** part:
  - ▶ A 8-bit **mask** leaves 24 bits for the **hosts**,
  - ▶ A 16-bit **mask** leaves 16 bits for the **hosts**,
  - ▶ A 24-bit **mask** leaves 8 bits for the **hosts**,
  - ▶ A N-bit **mask** leaves  $32-N$  bits for the **hosts**.

# IP addressing fundamentals

## Masks

- ▶ Separates **network** and **host** bits,
- ▶ MSB **always** are ones and then zeros! 255.254.255.0 is not possible,
- ▶ Indicates how many bits are used for the **network** part:
  - ▶ A 8-bit **mask** leaves 24 bits for the **hosts**,
  - ▶ A 16-bit **mask** leaves 16 bits for the **hosts**,
  - ▶ A 24-bit **mask** leaves 8 bits for the **hosts**,
  - ▶ A N-bit **mask** leaves 32-N bits for the **hosts**.
- ▶ Two different **masks** (differences seen further):
  - ▶ Network **mask**,
  - ▶ Subnet **mask**.

# IP addressing fundamentals

## IP address

32 bits (4x4 bytes)

Networks part	Host part

Figure : IP address parts and mask

# IP addressing fundamentals

## IP address

32 bits (4x4 bytes)

ones mask	zeros mask
Networks part	Host part

Figure : IP address parts and mask

# IP addressing fundamentals

Is that an address?

- ▶ Network address,

# IP addressing fundamentals

Is that an address?

- ▶ Network address,
- ▶ Hosts,

# IP addressing fundamentals

Is that an address?

- ▶ Network address,
- ▶ Hosts,
- ▶ Broadcast address.



# IP addressing fundamentals

## Is that an address?

- ▶ Network address,
- ▶ Hosts,
- ▶ Broadcast address.

## Within the same network

- ▶ All addresses have the same **network** bits,

# IP addressing fundamentals

## Is that an address?

- ▶ Network address,
- ▶ Hosts,
- ▶ Broadcast address.

## Within the same network

- ▶ All addresses have the same **network** bits,
- ▶ Network address has zeros for **host** bits:  $x.x.x.0^*$ ,

# IP addressing fundamentals

## Is that an address?

- ▶ Network address,
- ▶ Hosts,
- ▶ Broadcast address.

## Within the same network

- ▶ All addresses have the same **network** bits,
- ▶ Network address has zeros for **host** bits:  $x.x.x.0^*$ ,
- ▶ All **hosts** have different **host** bits:  $x.x.x.[0-1]^*$ ,

# IP addressing fundamentals

## Is that an address?

- ▶ Network address,
- ▶ Hosts,
- ▶ Broadcast address.

## Within the same network

- ▶ All addresses have the same **network** bits,
- ▶ Network address has zeros for **host** bits:  $x.x.x.0^*$ ,
- ▶ All **hosts** have different **host** bits:  $x.x.x.[0-1]^*$ ,
- ▶ Broadcast address has ones for **host** bits:  $x.x.x.1^*$ .

# IP addressing fundamentals

Mask /24 254 hosts	255 11111111	255 11111111	255 11111111	0 00000000
Network address	192 11000000	168 10101000	1 00000001	0 00000000
First host	192 11000000	168 10101000	1 00000001	1 00000001
Last host	192 11000000	168 10101000	1 00000001	254 11111110
Broadcast address	192 11000000	168 10101000	1 00000001	255 11111111

Figure : IP address example 1

# IP addressing fundamentals

Mask /16 65.534 hosts	255 11111111	255 11111111	0 00000000	0 00000000
Network address	172 10101100	64 01000000	0 00000000	0 00000000
First host	172 10101100	64 01000000	0 00000000	1 00000001
Last host	172 10101100	64 01000000	255 11111111	254 11111110
Broadcast address	172 10101100	64 01000000	255 11111111	255 11111111

Figure : IP address example 2

# IP addressing fundamentals

**Formula:** how many **hosts** with a N-bit mask?

$$2^{32-N} - 2$$

# IP addressing fundamentals

**Formula:** how many **hosts** with a N-bit mask?

$2^{32-N} - 2$ , the  $-2$  moves out network and broadcast addresses which are not **hosts**.



# IP addressing fundamentals

**Formula:** how many **hosts** with a N-bit mask?

$2^{32-N} - 2$ , the  $-2$  moves out network and broadcast addresses which are not **hosts**.

- ▶ 24-bit **mask**:  $2^{32-24} - 2 = 2^8 - 2 = 254$  **hosts**

# IP addressing fundamentals

**Formula:** how many **hosts** with a N-bit mask?

$2^{32-N} - 2$ , the  $-2$  moves out network and broadcast addresses which are not **hosts**.

- ▶ 24-bit **mask**:  $2^{32-24} - 2 = 2^8 - 2 = 254$  **hosts**
- ▶ 16-bit **mask**:  $2^{32-16} - 2 = 2^{16} - 2 = 65.534$  **hosts**

# IP addressing fundamentals

**Formula:** how many **hosts** with a N-bit mask?

$2^{32-N} - 2$ , the  $-2$  moves out network and broadcast addresses which are not **hosts**.

- ▶ 24-bit **mask**:  $2^{32-24} - 2 = 2^8 - 2 = 254$  **hosts**
- ▶ 16-bit **mask**:  $2^{32-16} - 2 = 2^{16} - 2 = 65.534$  **hosts**
- ▶ 8-bit **mask**:  $2^{32-8} - 2 = 2^{24} - 2 = 16.777.214$  **hosts**

# IP addressing fundamentals

## Public addresses

- ▶ Most of IP addresses

# IP addressing fundamentals

## Public addresses

- ▶ Most of IP addresses
- ▶ Registered ISP and large organizations inherit blocks of public addresses from IANA<sup>2</sup>

---

<sup>2</sup>Internet Assigned Numbers Authority

# IP addressing fundamentals

## Public addresses

- ▶ Most of IP addresses
- ▶ Registered ISP and large organizations inherit blocks of public addresses from IANA<sup>2</sup>
- ▶ Usage of not registered public addresses is forbidden.

## Private addresses

- ▶ Private addresses are A, B and C classes (not all, see after)

---

<sup>2</sup>Internet Assigned Numbers Authority

# IP addressing fundamentals

## Public addresses

- ▶ Most of IP addresses
- ▶ Registered ISP and large organizations inherit blocks of public addresses from IANA<sup>2</sup>
- ▶ Usage of not registered public addresses is forbidden.

## Private addresses

- ▶ Private addresses are A, B and C classes (not all, see after)
- ▶ No registration needed

---

<sup>2</sup>Internet Assigned Numbers Authority

# IP addressing fundamentals

## Public addresses

- ▶ Most of IP addresses
- ▶ Registered ISP and large organizations inherit blocks of public addresses from IANA<sup>2</sup>
- ▶ Usage of not registered public addresses is forbidden.

## Private addresses

- ▶ Private addresses are A, B and C classes (not all, see after)
- ▶ No registration needed
- ▶ Not routed across the Internet

---

<sup>2</sup>Internet Assigned Numbers Authority



# IP addressing fundamentals

## Public addresses

- ▶ Most of IP addresses
- ▶ Registered ISP and large organizations inherit blocks of public addresses from IANA<sup>2</sup>
- ▶ Usage of not registered public addresses is forbidden.

## Private addresses

- ▶ Private addresses are A, B and C classes (not all, see after)
- ▶ No registration needed
- ▶ Not routed across the Internet
- ▶ Proxy, NAT and private addresses solved IPv4 shortage.

---

<sup>2</sup>Internet Assigned Numbers Authority

# Classful IP Addressing

Class	A	B	C
First octet	1 - 126	128 - 191	192 - 223
First octet 0b	0*	10*	110*
Network mask	255.0.0.0 /8	255.255.0.0 /16	255.255.255.0 /24
IP addresses range	1.0.0.0 126.0.0.0	128.0.0.0 191.255.0.0	192.0.0.0 223.255.255.0
Private range	10.0.0.0 10.255.255.255	176.16.0.0 176.31.255.255	192.168.0.0 192.168.255.0
Number of hosts	16.777.214	65.534	254

Figure : Three main classes

Where did 127.0.0.0/8 go ?!

# Classful IP Addressing

## Class D

- ▶ First octet: 224 - 239

# Classful IP Addressing

## Class D

- ▶ First octet: 224 - 239
- ▶ First octet pattern: 1110\*

# Classful IP Addressing

## Class D

- ▶ First octet: 224 - 239
- ▶ First octet pattern: 1110\*
- ▶ These IP addresses are multicast addresses.

# Classful IP Addressing

## Class D

- ▶ First octet: 224 - 239
- ▶ First octet pattern: 1110\*
- ▶ These IP addresses are multicast addresses.

## Class E

- ▶ Everything left

# Classful IP Addressing

## Class D

- ▶ First octet: 224 - 239
- ▶ First octet pattern: 1110\*
- ▶ These IP addresses are multicast addresses.

## Class E

- ▶ Everything left
- ▶ Experimental class.

# Classful IP Addressing

## Reserved addresses

- ▶ 0.0.0.0 used in routing (seen further)



# Classful IP Addressing

## Reserved addresses

- ▶ 0.0.0.0 used in routing (seen further)
- ▶ 127.0.0.0/8: loopback addresses (127.0.0.1 - 127.255.255.254).

## Classful IP Addressing

- ▶ Class A (16 m-addresses) and B (65 k-addresses) are too large!

## Classful IP Addressing

- ▶ Class A (16 m-addresses) and B (65 k-addresses) are too large!
- ▶ Class C (254 addresses) is manageable. A and B are not, and then not fully utilized... That's a waste of IP addresses!

## Classful IP Addressing

- ▶ Class A (16 m-addresses) and B (65 k-addresses) are too large!
- ▶ Class C (254 addresses) is manageable. A and B are not, and then not fully utilized... That's a waste of IP addresses!

Means to limit the number of nodes on a network (regardless of the class) and, thus, improve the manageability, are needed. Three means for it:

- ▶ Subnet,

## Classful IP Addressing

- ▶ Class A (16 m-addresses) and B (65 k-addresses) are too large!
- ▶ Class C (254 addresses) is manageable. A and B are not, and then not fully utilized... That's a waste of IP addresses!

Means to limit the number of nodes on a network (regardless of the class) and, thus, improve the manageability, are needed. Three means for it:

- ▶ Subnet,
- ▶ VLSM (Variable Length Subnet Mask),

## Classful IP Addressing

- ▶ Class A (16 m-addresses) and B (65 k-addresses) are too large!
- ▶ Class C (254 addresses) is manageable. A and B are not, and then not fully utilized... That's a waste of IP addresses!

Means to limit the number of nodes on a network (regardless of the class) and, thus, improve the manageability, are needed. Three means for it:

- ▶ Subnet,
- ▶ VLSM (Variable Length Subnet Mask),
- ▶ CIDR (Classless Inter-Domain Routing).

# Subnet and VLSM

- ▶ Class A (16 m-addresses) and B (65 k-addresses) are too large!

# Subnet and VLSM

- ▶ Class A (16 m-addresses) and B (65 k-addresses) are too large!
- ▶ Class C (254 addresses) is manageable. A and B are not, and then not fully utilized... That's a waste of IP addresses!



## Subnet and VLSM

Mask /16 65.534 hosts	255 11111111	255 11111111	0 00000000	0 00000000
Network address	172 10101100	64 01000000	0 00000000	0 00000000
First host	172 10101100	64 01000000	0 00000000	1 00000001
Last host	172 10101100	64 01000000	255 11111111	254 11111110
Broadcast address	172 10101100	64 01000000	255 11111111	255 11111111

Figure : IP address example 2

## Subnet and VLSM

Mask /12 1.048.574 hosts	255 11111111	240 11110000	0 00000000	0 00000000
Network address	172 10101100	64 01000000	0 00000000	0 00000000
First host	172 10101100	64 01000000	0 00000000	1 00000001
Last host	172 10101100	79 01001111	255 11111111	254 11111110
Broadcast address	172 10101100	79 01001111	255 11111111	255 11111111

Figure : IP address example 3

## Subnet and VLSM

Mask /10 4.194.302 hosts	255 11111111	192 11000000	0 00000000	0 00000000
Network address	172 10101100	64 01000000	0 00000000	0 00000000
First host	172 10101100	64 01000000	0 00000000	1 00000001
Last host	172 10101100	127 01111111	255 11111111	254 11111110
Broadcast address	172 10101100	127 01111111	255 11111111	255 11111111

Figure : IP address example 4

## Subnet and VLSM

Mask /31 0 host	255 11111111	255 11111111	255 11111111	254 11111110
Network address	172 10101100	64 01000000	0 00000000	254 11111110
First host	172 10101100	64 01000000	0 00000000	? 1111111?
Last host	172 10101100	64 01000000	255 00000000	? 1111111?
Broadcast address	172 10101100	64 01000000	255 00000000	255 11111111

Figure : IP address example 5

## Subnet and VLSM

Mask /30 2 hosts	255 11111111	255 11111111	255 11111111	252 11111100
Network address	172 10101100	64 01000000	0 00000000	252 11111100
First host	172 10101100	64 01000000	0 00000000	253 11111101
Last host	172 10101100	64 01000000	255 00000000	254 11111110
Broadcast address	172 10101100	64 01000000	255 00000000	255 11111111

Figure : IP address example 6

	Netmask	CIDR	hosts
255.255.255.255	11111111.11111111.11111111.11111111	/32	single address
255.255.255.254	11111111.11111111.11111111.11111110	/31	Unusable
255.255.255.252	11111111.11111111.11111111.11111100	/30	2
255.255.255.248	11111111.11111111.11111111.11111000	/29	6
255.255.255.240	11111111.11111111.11111111.11110000	/28	14
255.255.255.224	11111111.11111111.11111111.11100000	/27	30
255.255.255.192	11111111.11111111.11111111.11000000	/26	62
255.255.255.128	11111111.11111111.11111111.10000000	/25	126
255.255.255.0	11111111.11111111.11111111.00000000	/24	254
255.255.254.0	11111111.11111111.11111110.00000000	/23	510
255.255.252.0	11111111.11111111.11111100.00000000	/22	1.022
255.255.248.0	11111111.11111111.11111000.00000000	/21	2.046
255.255.240.0	11111111.11111111.11110000.00000000	/20	4.094
255.255.224.0	11111111.11111111.11100000.00000000	/19	8.190
255.255.192.0	11111111.11111111.11000000.00000000	/18	16.382
255.255.128.0	11111111.11111111.10000000.00000000	/17	32.766
255.255.0.0	11111111.11111111.00000000.00000000	/16	65.534
255.254.0.0	11111111.11111110.00000000.00000000	/15	131.070
255.252.0.0	11111111.11111100.00000000.00000000	/14	262.142
255.248.0.0	11111111.11111000.00000000.00000000	/13	524.286
255.240.0.0	11111111.11110000.00000000.00000000	/12	1.048.574
255.224.0.0	11111111.11100000.00000000.00000000	/11	2.097.152
255.192.0.0	11111111.11000000.00000000.00000000	/10	4.194.302
255.128.0.0	11111111.10000000.00000000.00000000	/9	8.388.606
255.0.0.0	11111111.00000000.00000000.00000000	/8	16.777.214
254.0.0.0	11111110.00000000.00000000.00000000	/7	33.554.430
252.0.0.0	11111100.00000000.00000000.00000000	/6	67.108.862
248.0.0.0	11111000.00000000.00000000.00000000	/5	134.217.726
240.0.0.0	11110000.00000000.00000000.00000000	/4	268.435.454
224.0.0.0	11100000.00000000.00000000.00000000	/3	536.870.910
192.0.0.0	11000000.00000000.00000000.00000000	/2	1.073.741.822
128.0.0.0	10000000.00000000.00000000.00000000	/1	2.147.483.646
0.0.0.0	00000000.00000000.00000000.00000000	/0	IP space

# CIDR

## Classless Inter-domain Routing?

# CIDR

## Classless Inter-domain Routing?

- ▶ Wait! What is routing?



## Routing Principles

Algorithm processed to decide where to forward a packet

### Any router must

- ▶ know where any packet should be directed
- ▶ send directly the packets to the packet's destination if the router and the destination are on the same network

### Any node

- ▶ on any network can communicate directly with all the nodes within the same network
- ▶ can connect to any node using its gateway
- ▶ needs to be aware of its gateway to communicate with nodes on other networks

# Routing Principles

## Route

- ▶ Destination
- ▶ Gateway
- ▶ Masks
- ▶ Metric

# Routing Principles

## Route

- ▶ Destination
- ▶ Gateway
- ▶ Masks
- ▶ Metric

```
>sudo route -n
Kernel IP routing table
Destination      Gateway          Genmask          Flags Metric Ref    Use Iface
0.0.0.0          192.168.0.254   0.0.0.0          UG    0      0        0 eth0
192.168.0.0      0.0.0.0         255.255.255.0    U     0      0        0 eth0
```

Figure : Routing table

# Routing Principles

```
>sudo route -n
Kernel IP routing table
Destination      Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0          192.168.0.254  0.0.0.0         UG    0      0        0 eth0
192.168.0.0      0.0.0.0        255.255.255.0   U      0      0        0 eth0
```

Figure : Routing table

0.0.0.0 ?

- ▶ Default address
- ▶ Default route
- ▶ Default gateway

# Routing Principles

## Example

What would the routing table of this router will look like?

# Routing Principles

Static or dynamic ?

# Routing Principles

Static or dynamic ?

We will see this later

# CIDR

Combine 2+ networks' into one bigger to facilitate routing.



# CIDR

Combine 2+ networks' into one bigger to facilitate routing.

## Classless Inter-domain Routing?

- ▶ Does a routing table having both (192.168.0.0/24, E0), (192.168.1.0/24, E0), (10.0.0.0/8, S0) can be shorten?

# CIDR

Combine 2+ networks' into one bigger to facilitate routing.

## Classless Inter-domain Routing?

- ▶ Does a routing table having both (192.168.0.0/24, E0), (192.168.1.0/24, E0), (10.0.0.0/8, S0) can be shorten?
- ▶ Does a routing table having both (192.168.0.0/24, E0), (192.168.1.0/24, E0), (192.168.8.0/24, E0), (10.0.0.0/8, S0) can be shorten?

# CIDR

Combine 2+ networks' into one bigger to facilitate routing.

## Classless Inter-domain Routing?

- ▶ Does a routing table having both (192.168.0.0/24, E0), (192.168.1.0/24, E0), (10.0.0.0/8, S0) can be shorten?
- ▶ Does a routing table having both (192.168.0.0/24, E0), (192.168.1.0/24, E0), (192.168.8.0/24, E0), (10.0.0.0/8, S0) can be shorten?
- ▶ Does a routing table having both (192.168.0.0/24, E0), (192.168.4.0/24, E0), (192.168.1.0/24, E1), (10.0.0.0/8, S0) can be shorten?

# Routing Protocol

- ▶ RIP: Routing Information Protocol

# Routing Protocol

- ▶ RIP: Routing Information Protocol
- ▶ OSPF: Open Shortest Path First

# Routing Protocol

- ▶ RIP: Routing Information Protocol
- ▶ OSPF: Open Shortest Path First
- ▶ EIGRP: Enhanced Interior Gateway Routing Protocol

# Routing Protocol

## RIP v1

- ▶ Classful routing

# Routing Protocol

## RIP v1

- ▶ Classful routing
- ▶ Periodic updates (30 sec) ..



# Routing Protocol

## RIP v1

- ▶ Classful routing
- ▶ Periodic updates (30 sec) ..
- ▶ ..by broadcasting (!)

# Routing Protocol

## RIP v1

- ▶ Classful routing
- ▶ Periodic updates (30 sec) ..
- ▶ ..by broadcasting (!)
- ▶ Metric is hop-count (max = 15, infinite = 16)

# Routing Protocol

## RIP v1

- ▶ Classful routing
- ▶ Periodic updates (30 sec) ..
- ▶ ..by broadcasting (!)
- ▶ Metric is hop-count (max = 15, infinite = 16)
- ▶ Timer (180 sec) to tag route as invalid (metric = 16)

# Routing Protocol

## RIP v1

- ▶ Classful routing
- ▶ Periodic updates (30 sec) ..
- ▶ ..by broadcasting (!)
- ▶ Metric is hop-count (max = 15, infinite = 16)
- ▶ Timer (180 sec) to tag route as invalid (metric = 16)
- ▶ no subnet, no VLSM, no CIDR, no router authentication

# Routing Protocol

## RIP v2

- ▶ Classless routing

# Routing Protocol

## RIP v2

- ▶ Classless routing
- ▶ Multicast (224.0.0.9)

# Routing Protocol

## RIP v2

- ▶ Classless routing
- ▶ Multicast (224.0.0.9)
- ▶ VLSM support

# Routing Protocol

## RIP v2

- ▶ Classless routing
- ▶ Multicast (224.0.0.9)
- ▶ VLSM support
- ▶ Route summarization



# Routing Protocol

## RIP v2

- ▶ Classless routing
- ▶ Multicast (224.0.0.9)
- ▶ VLSM support
- ▶ Route summarization
- ▶ "Authentication" (MD5)

# Routing Protocol

## RIP v2

- ▶ Classless routing
- ▶ Multicast (224.0.0.9)
- ▶ VLSM support
- ▶ Route summarization
- ▶ "Authentication" (MD5)

RIPng is the next RIP version for support of IPv6

# Routing Protocol

1. Router getting online broadcasts Request message

---

<sup>3</sup>not always all the routing table

# Routing Protocol

1. Router getting online broadcasts Request message
2. RIP Router send broadcasts Response message with their routing table

---

<sup>3</sup>not always all the routing table

# Routing Protocol

1. Router getting online broadcasts Request message
2. RIP Router send broadcasts Response message with their routing table
3. When Update timers (from other routers) expire routing table<sup>3</sup> is sent again

---

<sup>3</sup>not always all the routing table

# Routing Protocol

1. Router getting online broadcasts Request message
2. RIP Router send broadcasts Response message with their routing table
3. When Update timers (from other routers) expire routing table<sup>3</sup> is sent again
4. When Invalid timer expires, the metric of the route is set to 16 (unreachable)

---

<sup>3</sup>not always all the routing table

# Routing Protocol

1. Router getting online broadcasts Request message
2. RIP Router send broadcasts Response message with their routing table
3. When Update timers (from other routers) expire routing table<sup>3</sup> is sent again
4. When Invalid timer expires, the metric of the route is set to 16 (unreachable)
5. When Flush timer expires, the 16-metric routes are removed from the routing table

---

<sup>3</sup>not always all the routing table

# Routing Protocol

1. Router getting online broadcasts Request message
2. RIP Router send broadcasts Response message with their routing table
3. When Update timers (from other routers) expire routing table<sup>3</sup> is sent again
4. When Invalid timer expires, the metric of the route is set to 16 (unreachable)
5. When Flush timer expires, the 16-metric routes are removed from the routing table
6. When a new router (or new metric) is sent, a Hold-down timer is started to stabilize the network.

---

<sup>3</sup>not always all the routing table



# Routing Protocol

## OSPF

- ▶ Classless

# Routing Protocol

## OSPF

- ▶ Classless
- ▶ IPv4 and IPv6

# Routing Protocol

## OSPF

- ▶ Classless
- ▶ IPv4 and IPv6
- ▶ VSLM

# Routing Protocol

## OSPF

- ▶ Classless
- ▶ IPv4 and IPv6
- ▶ VSLM
- ▶ CIDR

# Routing Protocol

## OSPF

- ▶ Classless
- ▶ IPv4 and IPv6
- ▶ VSLM
- ▶ CIDR
- ▶ Build a topology of the network

# Routing Protocol

## OSPF

- ▶ Classless
- ▶ IPv4 and IPv6
- ▶ VSLM
- ▶ CIDR
- ▶ Build a topology of the network
- ▶ Dijkstra

# Routing Protocol

## OSPF

- ▶ Classless
- ▶ IPv4 and IPv6
- ▶ VSLM
- ▶ CIDR
- ▶ Build a topology of the network
- ▶ Dijkstra
- ▶ Metric =  $f(\text{hop-count, bandwidth, link reliability})$

# Routing Protocol

## OSPF

- ▶ Classless
- ▶ IPv4 and IPv6
- ▶ VSLM
- ▶ CIDR
- ▶ Build a topology of the network
- ▶ Dijkstra
- ▶ Metric =  $f(\text{hop-count, bandwidth, link reliability})$
- ▶ Subdivided into area (a 32-bit number)



# Routing Protocol

## OSPF

- ▶ Classless
- ▶ IPv4 and IPv6
- ▶ VSLM
- ▶ CIDR
- ▶ Build a topology of the network
- ▶ Dijkstra
- ▶ Metric =  $f(\text{hop-count, bandwidth, link reliability})$
- ▶ Subdivided into area (a 32-bit number)
- ▶ Multicast

# Routing Protocol

## OSPF

- ▶ Classless
- ▶ IPv4 and IPv6
- ▶ VSLM
- ▶ CIDR
- ▶ Build a topology of the network
- ▶ Dijkstra
- ▶ Metric =  $f(\text{hop-count, bandwidth, link reliability})$
- ▶ Subdivided into area (a 32-bit number)
- ▶ Multicast
- ▶ Authentication support (update only from trusted routers)

# Routing Protocol

## EIGRP

- ▶ Enhanced IGRP (to support classless routing)

# Routing Protocol

## EIGRP

- ▶ Enhanced IGRP (to support classless routing)
- ▶ IPv4 and IPv6

# Routing Protocol

## EIGRP

- ▶ Enhanced IGRP (to support classless routing)
- ▶ IPv4 and IPv6
- ▶ VSLM

# Routing Protocol

## EIGRP

- ▶ Enhanced IGRP (to support classless routing)
- ▶ IPv4 and IPv6
- ▶ VSLM
- ▶ CIDR

# Routing Protocol

## EIGRP

- ▶ Enhanced IGRP (to support classless routing)
- ▶ IPv4 and IPv6
- ▶ VSLM
- ▶ CIDR
- ▶ Build a topology of the network

# Routing Protocol

## EIGRP

- ▶ Enhanced IGRP (to support classless routing)
- ▶ IPv4 and IPv6
- ▶ VSLM
- ▶ CIDR
- ▶ Build a topology of the network
- ▶ Dijkstra



# Routing Protocol

## EIGRP

- ▶ Enhanced IGRP (to support classless routing)
- ▶ IPv4 and IPv6
- ▶ VSLM
- ▶ CIDR
- ▶ Build a topology of the network
- ▶ Dijkstra
- ▶  $\text{Metric} = f(\text{bandwidth, load, delay, reliability})$

# Routing Protocol

## EIGRP

- ▶ Enhanced IGRP (to support classless routing)
- ▶ IPv4 and IPv6
- ▶ VSLM
- ▶ CIDR
- ▶ Build a topology of the network
- ▶ Dijkstra
- ▶  $\text{Metric} = f(\text{bandwidth, load, delay, reliability})$
- ▶ Authentication support

# IPv6 - Aims

- ▶ Support billions of hosts (even with inefficient IP addressing)

## IPv6 - Aims

- ▶ Support billions of hosts (even with inefficient IP addressing)
- ▶ Reduce routing table size

## IPv6 - Aims

- ▶ Support billions of hosts (even with inefficient IP addressing)
- ▶ Reduce routing table size
- ▶ Simplified protocol to allow routers to process packets faster

## IPv6 - Aims

- ▶ Support billions of hosts (even with inefficient IP addressing)
- ▶ Reduce routing table size
- ▶ Simplified protocol to allow routers to process packets faster
- ▶ Better security

# IPv6 - Aims

- ▶ Support billions of hosts (even with inefficient IP addressing)
- ▶ Reduce routing table size
- ▶ Simplified protocol to allow routers to process packets faster
- ▶ Better security
- ▶ Better real-time QoS

# IPv6 - Aims

- ▶ Support billions of hosts (even with inefficient IP addressing)
- ▶ Reduce routing table size
- ▶ Simplified protocol to allow routers to process packets faster
- ▶ Better security
- ▶ Better real-time QoS
- ▶ Better multicast diffusion (scope)



# IPv6 - Aims

- ▶ Support billions of hosts (even with inefficient IP addressing)
- ▶ Reduce routing table size
- ▶ Simplified protocol to allow routers to process packets faster
- ▶ Better security
- ▶ Better real-time QoS
- ▶ Better multicast diffusion (scope)
- ▶ Being able to move, without changing IP address

# IPv6 - Aims

- ▶ Support billions of hosts (even with inefficient IP addressing)
- ▶ Reduce routing table size
- ▶ Simplified protocol to allow routers to process packets faster
- ▶ Better security
- ▶ Better real-time QoS
- ▶ Better multicast diffusion (scope)
- ▶ Being able to move, without changing IP address
- ▶ Make the protocol able to evolve

# IPv6 - Aims

- ▶ Support billions of hosts (even with inefficient IP addressing)
- ▶ Reduce routing table size
- ▶ Simplified protocol to allow routers to process packets faster
- ▶ Better security
- ▶ Better real-time QoS
- ▶ Better multicast diffusion (scope)
- ▶ Being able to move, without changing IP address
- ▶ Make the protocol able to evolve
- ▶ Make the protocol able to coexist with newer version

# IPv4 vs IPv6

- ▶ not compatible

## IPv4 vs IPv6

- ▶ not compatible
- ▶ IPv4 address: 4 octets, IPv6: 16 octets ( $2^{128} = 3 \times 10^{38}$ )

# IPv4 vs IPv6

- ▶ not compatible
- ▶ IPv4 address: 4 octets, IPv6: 16 octets ( $2^{128} = 3 \times 10^{138}$ )
- ▶ Packet Header, IPv6: 7 fields, IPv4:13 (faster to process)

# IPv4 vs IPv6

- ▶ not compatible
- ▶ IPv4 address: 4 octets, IPv6: 16 octets ( $2^{128} = 3 \times 10^{38}$ )
- ▶ Packet Header, IPv6: 7 fields, IPv4: 13 (faster to process)
- ▶ IP options: some required options are now optionals (faster to process)

# IPv4 vs IPv6

- ▶ not compatible
- ▶ IPv4 address: 4 octets, IPv6: 16 octets ( $2^{128} = 3 \times 10^{138}$ )
- ▶ Packet Header, IPv6: 7 fields, IPv4: 13 (faster to process)
- ▶ IP options: some required options are now optionals (faster to process)
- ▶ Notation:
  - ▶ 8000:0000:0000:0000:0123:4567:89AB:CDEF



# IPv4 vs IPv6

- ▶ not compatible
- ▶ IPv4 address: 4 octets, IPv6: 16 octets ( $2^{128} = 3 \times 10^{138}$ )
- ▶ Packet Header, IPv6: 7 fields, IPv4: 13 (faster to process)
- ▶ IP options: some required options are now optionals (faster to process)
- ▶ Notation:
  - ▶ 8000:0000:0000:0000:0123:4567:89AB:CDEF
  - ▶ 8000::0123:4567:89AB:CDEF

# IPv4 vs IPv6

- ▶ not compatible
- ▶ IPv4 address: 4 octets, IPv6: 16 octets ( $2^{128} = 3 \times 10^{138}$ )
- ▶ Packet Header, IPv6: 7 fields, IPv4: 13 (faster to process)
- ▶ IP options: some required options are now optionals (faster to process)
- ▶ Notation:
  - ▶ 8000:0000:0000:0000:0123:4567:89AB:CDEF
  - ▶ 8000::0123:4567:89AB:CDEF
  - ▶ ::192.168.2.3

## IPv4 vs IPv6

- ▶ not compatible
- ▶ IPv4 address: 4 octets, IPv6: 16 octets ( $2^{128} = 3 \times 10^{138}$ )
- ▶ Packet Header, IPv6: 7 fields, IPv4:13 (faster to process)
- ▶ IP options: some required options are now optionals (faster to process)
- ▶ Notation:
  - ▶ 8000:0000:0000:0000:0123:4567:89AB:CDEF
  - ▶ 8000::0123:4567:89AB:CDEF
  - ▶ ::192.168.2.3
- ▶ Unicast address format:

bits	48 (or more)	16 (or fewer)	64
field	routing prefix	subnet id	interface identifier

Figure : Unicast IPv6 address format

# IPv6 adoption

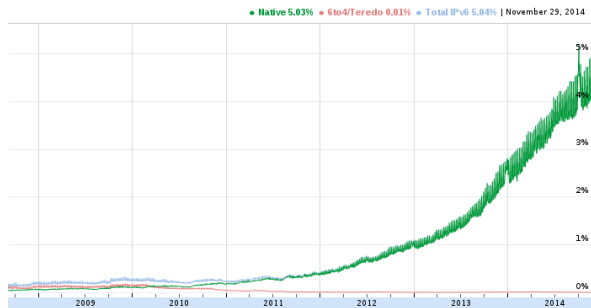


Figure : IPv6 adoption (among Google users)<sup>4</sup>

Belgium: 28%, USA and Germany: 11%

<sup>4</sup><https://www.google.com/intl/en/ipv6/statistics.html>

# Presentation Outline

Introduction

Physical

Data Link

Network

Transport

# Aims

- ▶ Interface session layer,

# Aims

- ▶ Interface session layer,
- ▶ Reliability end-to-end communication,

# Aims

- ▶ Interface session layer,
- ▶ Reliability end-to-end communication,
- ▶ Order and reassemble received packets,



# Aims

- ▶ Interface session layer,
- ▶ Reliability end-to-end communication,
- ▶ Order and reassemble received packets,
- ▶ Flow control,

# Aims

- ▶ Interface session layer,
- ▶ Reliability end-to-end communication,
- ▶ Order and reassemble received packets,
- ▶ Flow control,
- ▶ Congestion avoidance,

# Aims

- ▶ Interface session layer,
- ▶ Reliability end-to-end communication,
- ▶ Order and reassemble received packets,
- ▶ Flow control,
- ▶ Congestion avoidance,
- ▶ Multiplexing

# Application identification

## Socket address

- ▶ Node identification is made by IP address,

# Application identification

## Socket address

- ▶ Node identification is made by IP address,
- ▶ Application identification is made by node identification..

# Application identification

## Socket address

- ▶ Node identification is made by IP address,
- ▶ Application identification is made by node identification..
- ▶ .. and a port. Number between 0 and 65535. (1-1024: root privilege)

# Application identification

## Socket address

- ▶ Node identification is made by IP address,
- ▶ Application identification is made by node identification..
- ▶ .. and a port. Number between 0 and 65535. (1-1024: root privilege)
  - ▶ ip.ad.dr.ess:port

Port	Protocol
21	FTP
22	SSH
23	Telnet
25	SMTP
465	SMTPS
80	HTTP
443	HTTPS
3128 - 8080	Web Proxy
9418	git

Figure : Default port for well known protocol



# TCP header

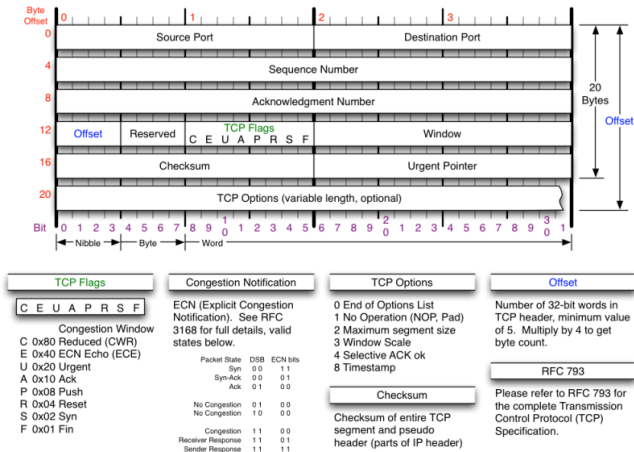


Figure : nmap.org: TCP header

# UDP header

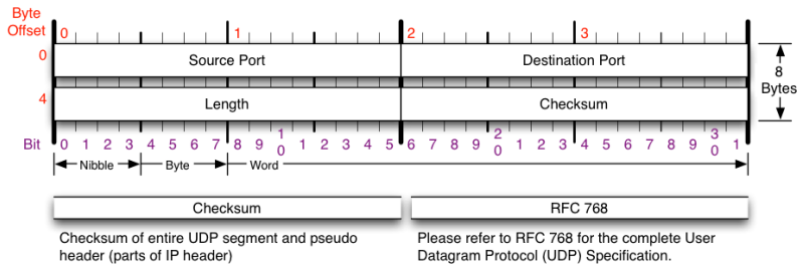


Figure : nmap.org: UDP header

## Socket Primitives (TCP)

Order	Primitive	Meaning
1	SOCKET	Creates a new communication endpoint
2	BIND	Links local IP address to the socket
3	LISTEN	Signs up for incoming connections
4	ACCEPT	Blocking call till a connection attempt occurs
-	CONNECT	<b>Tries</b> to connect to another communication endpoint
-	SEND	Sends data through the established connection
-	RECEIVE	Receives data through the established connection
last	CLOSE	Releases the connection

Figure : TCP primitives

A socket does not have an IP address until it is bound, just an allocation in the transport entity. A server must listen before any client is able to connect.

## What are theses?

- ▶ **Frame:** Physical layer representation
- ▶ **Datagram:** UDP<sup>5</sup> or IP packet (IP datagram, UDP datagram)
- ▶ **Segment:** TCP data unit
- ▶ **PDU:** Protocol Data Unit, generic term.
- ▶ **Fragment:** Any data unit **fragmented**

---

<sup>5</sup>User **Datagram** Protocol

# Hope you liked it and learnt about networking!



Figure : [teaching.auzias.net](http://teaching.auzias.net)

## Take a look:

- ▶ "Computer Networks" by A Tanenbaum, Andrew S., G ISBN 013162959X
- ▶ <http://nmap.org/book/toc.html>
- ▶ and many many other resources on the Internet freely available. If you can read it, knowledge is reachable!