

Network Computing courses

Maël Auzias

ENSIBS - UBS

October 2014



Figure: teaching.auzias.net

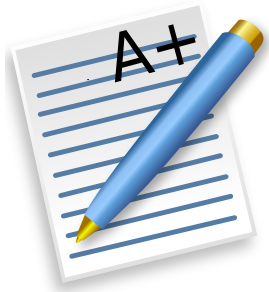
Course details

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 (github too)

Presentation Outline

- 1 Introduction
 - Definitions and presentation
 - Network classification
 - HTTP request/response example
 - Models overview (OSI and TCP/IP)
- 2 Layers
 - Physical
 - Data Link
 - Network
 - Transport
 - Session
 - Presentation
 - Application

Definitions

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

Definitions

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

Definitions

- **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

- **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

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

Definitions

- **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

- **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

- **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

- **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

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

Definitions

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

Definitions

- **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

- **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

- **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

- **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

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

Definitions

- **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

- **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

- **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

- **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

- **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

- **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

What kind of network is it?

- **BAN:** Body Area Network

What kind of network is it?

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

What kind of network is it?

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

What kind of network is it?

- **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

What kind of network is it?

- **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

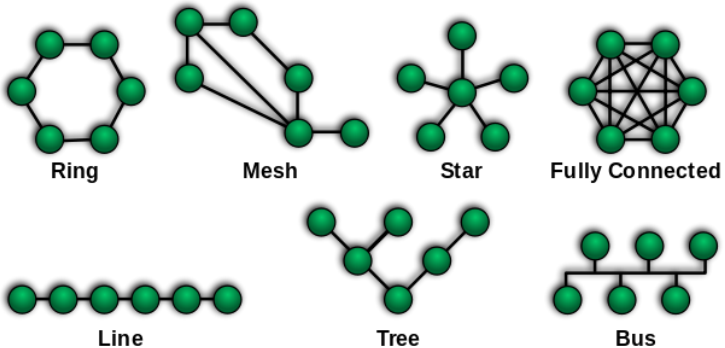


Figure: upload.wikimedia.org

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 network¹.

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 network¹.
- **Star:** all messages go through the same central node, reducing network failure.

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 network¹.
- **Star:** all messages go through the same central node, reducing network failure.
- **Fully connected:** all nodes are connected to all other nodes.

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 network¹.
- **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.

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 network¹.
- **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.

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 network¹.
- **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.

Bonus

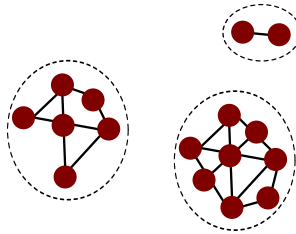


Figure: Disconnected MANET illustration [1]

Bonus

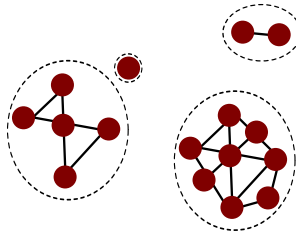


Figure: Store-carry-and-forward [1]

Bonus

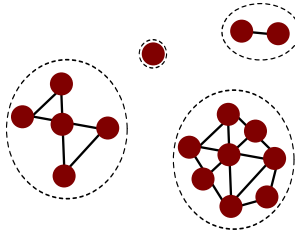


Figure: Store-carry-and-forward [1]

Bonus

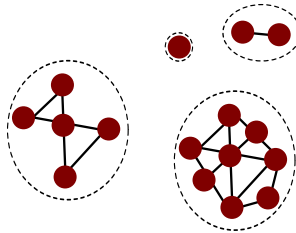


Figure: Store-carry-and-forward [1]

Bonus

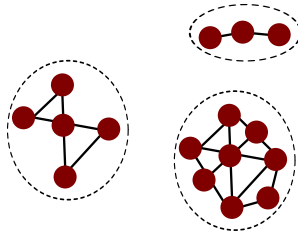


Figure: Store-carry-and-forward [1]

HTTP request/response example

Enter getbootstrap.com in your browser

HTTP request/response example

Enter 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 in your browser

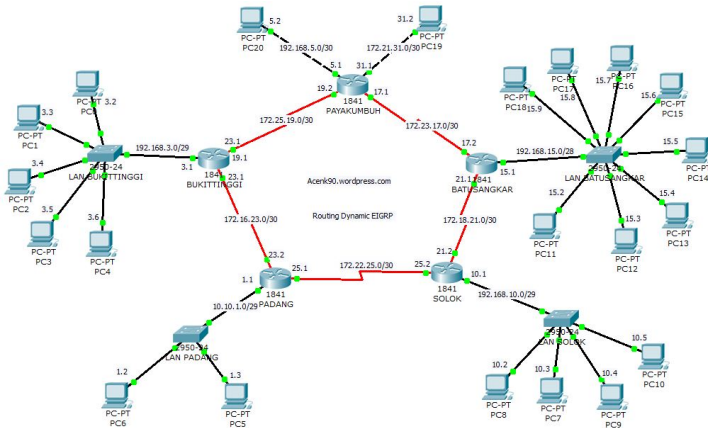
Source	Destination	Protocol	Length	Info
192.168.0.48	208.67.222.222	DNS	76	Standard query 0x4797 A getbootstrap.com
208.67.222.222	192.168.0.48	DNS	108	Standard query response 0x4797 A 192.30.252.154 A 192.30.252.153

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 [SYN, 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

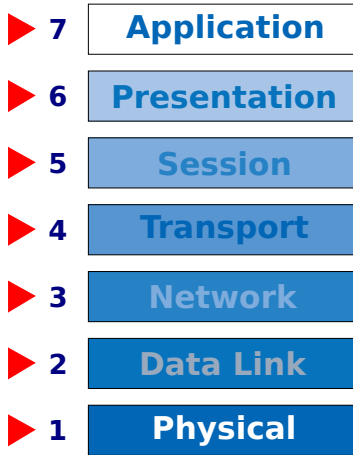
Figure: HTTP request/response

How do messages reach their destination?

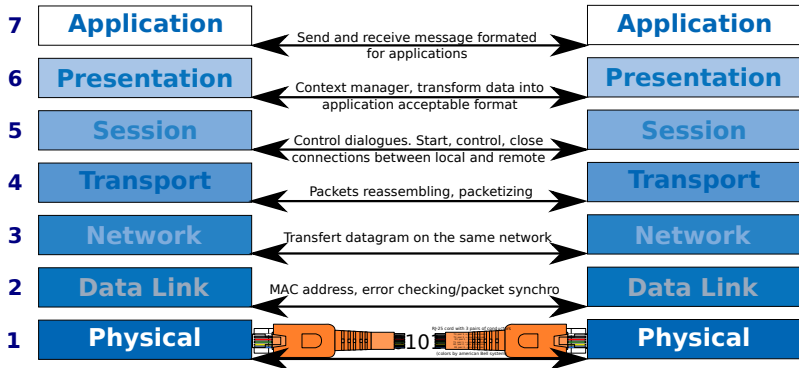


Topology Jaringan Menggunakan Routing Dynamic EIGRP Pada Cisco Router

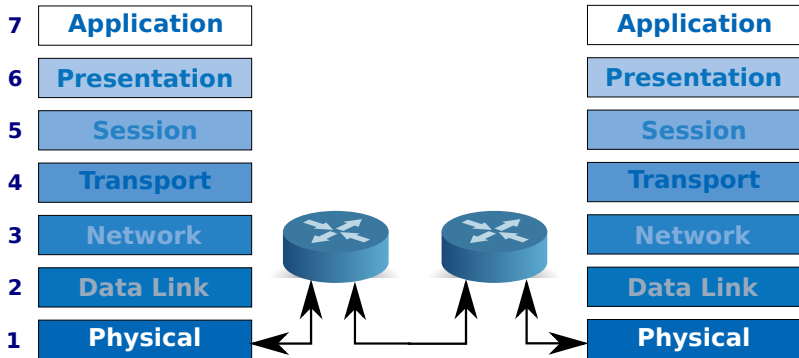
How does it work? From signal to application...



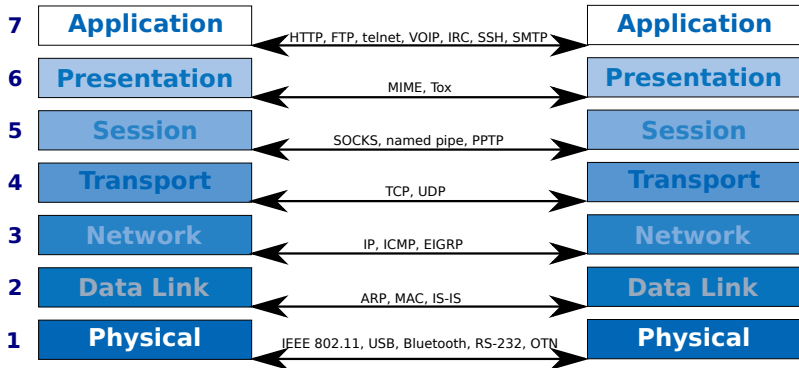
N^{th} layer communicate with N^{th} layer..



.. thanks to 3th layers



One single protocol, one single layer



Encapsulation

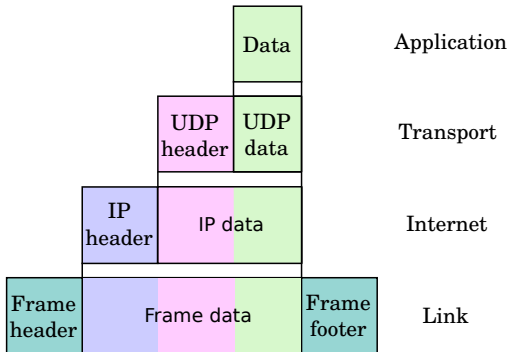


Figure: Encapsulation

Presentation Outline

- 1 Introduction
 - Definitions and presentation
 - Network classification
 - HTTP request/response example
 - Models overview (OSI and TCP/IP)
- 2 Layers
 - Physical
 - Data Link
 - Network
 - Transport
 - Session
 - Presentation
 - Application

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): <100Gbit/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): <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

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): <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

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)

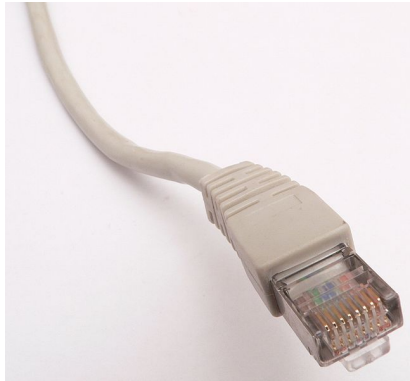


Figure: RJ45 connector

Hardware medium: IEEE 802.15.1 (Bluetooth)

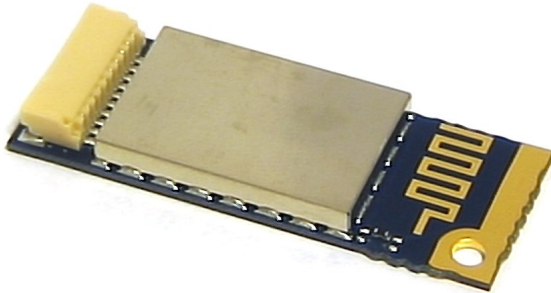


Figure: Bluetooth card

Hardware medium: IEEE 802.15.4 (ZigBee)

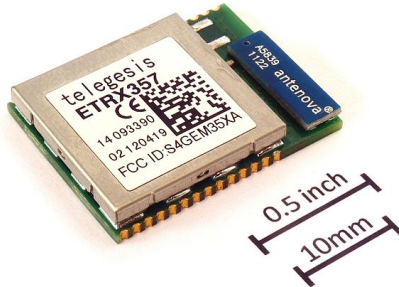


Figure: ZigBee card

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 $+/-1$ 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

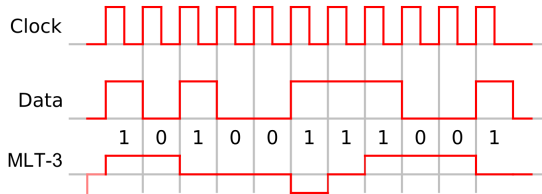


Figure: Multi-Level Transmit

Encoding: Alternate Mark Inversion

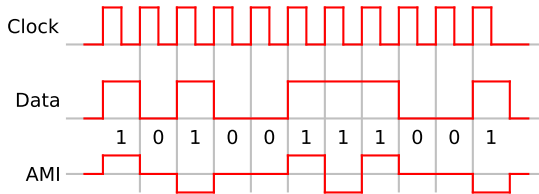


Figure: Alternate Mark Inversion

Encoding: Manchester

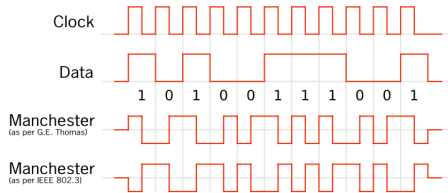


Figure: Manchester

Encoding: Biphase Mark Code

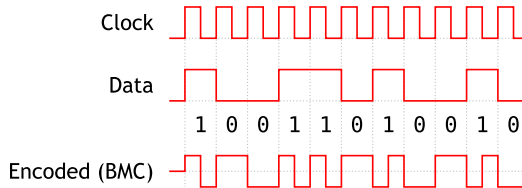


Figure: Biphase Mark Code

Transmitting

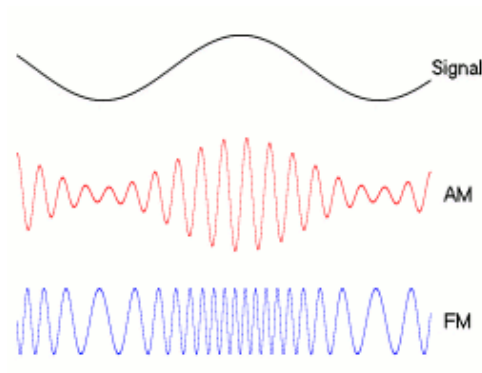


Figure: Amplitude and phase modulation

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): state change for 1s over 3 levels, stay in the same state for 0s

Error correcting

- Repetition (again)
- Hamming
- MDPC (Multidimensional parity-check code): state change for 1s over 3 levels, stay in the same state for 0s

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

Course details

Course details

Course details

Course details

Course details

Course details

References



Maurice J. Khabbaz, Assi Chadi M., and Fawaz Wissam F.
Disruption-Tolerant Networking: A Comprehensive Survey on
Recent Developments and Persisting Challenges.
IEEE communications surveys and tutorials, 2012.