

# TCP / UDP

**Capa de transport d'Ethernet**

***Jordi Binefa i Martínez***

Professor de cicles formatius a

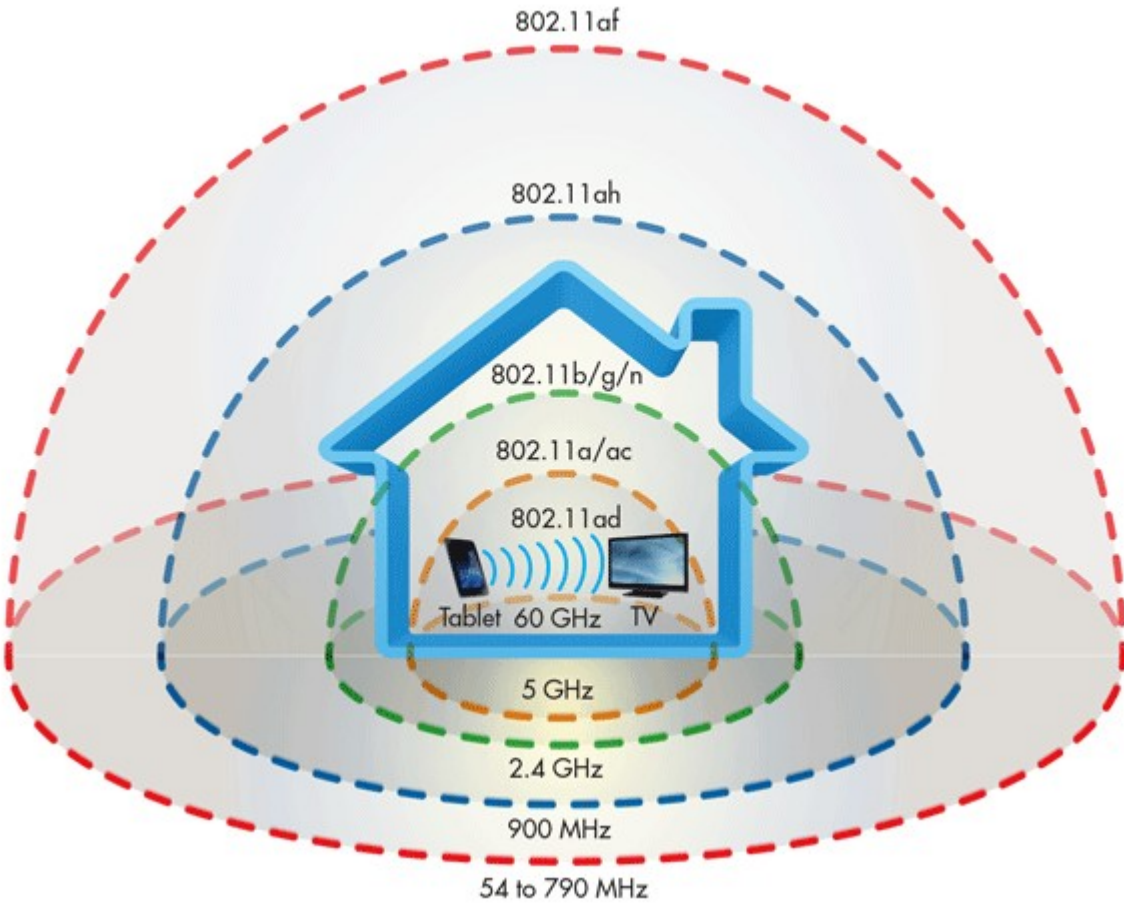


**JESUÏTES** El Clot  
Escola del Clot



# TCP / UDP

## WiFi

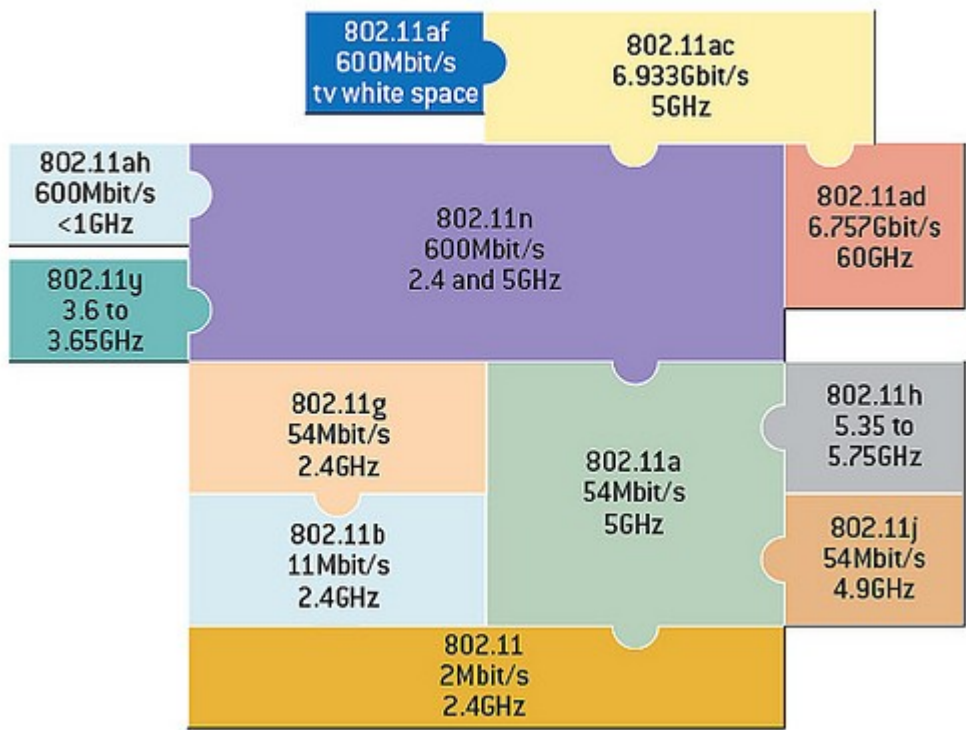




# TCP / UDP

## WiFi

Fig 1: How the various WI-FI standards relate to each other

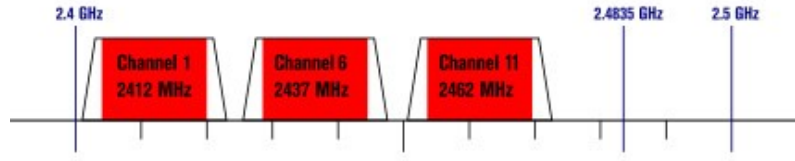


### Non-Overlapping Channels for 2.4 GHz WLAN

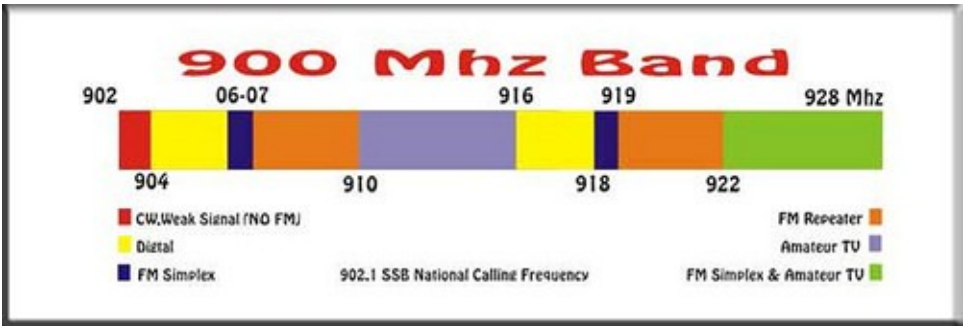
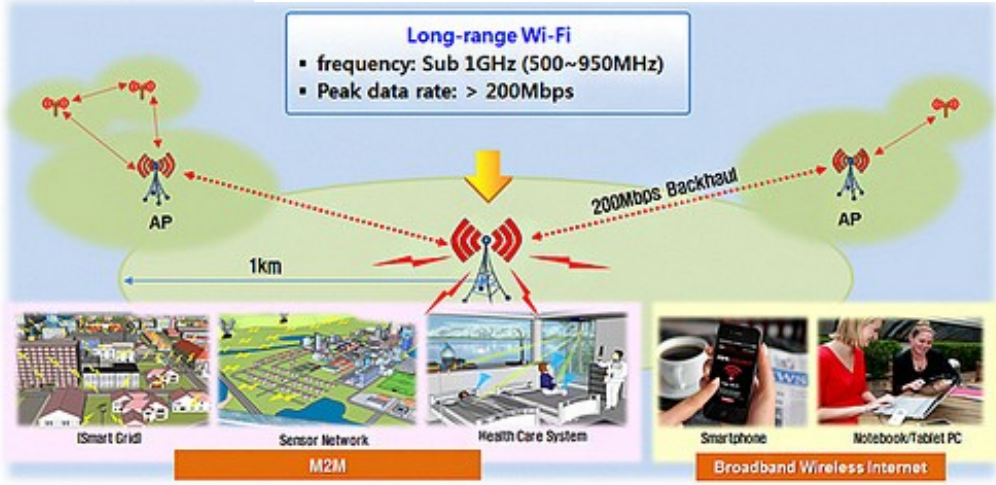
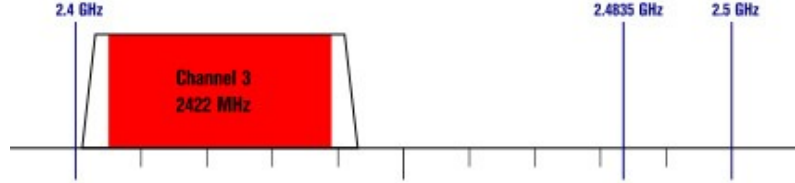
802.11b (DSSS) channel width 22 MHz



802.11g/n (OFDM) 20 MHz ch. width - 16.25 MHz used by sub-carriers



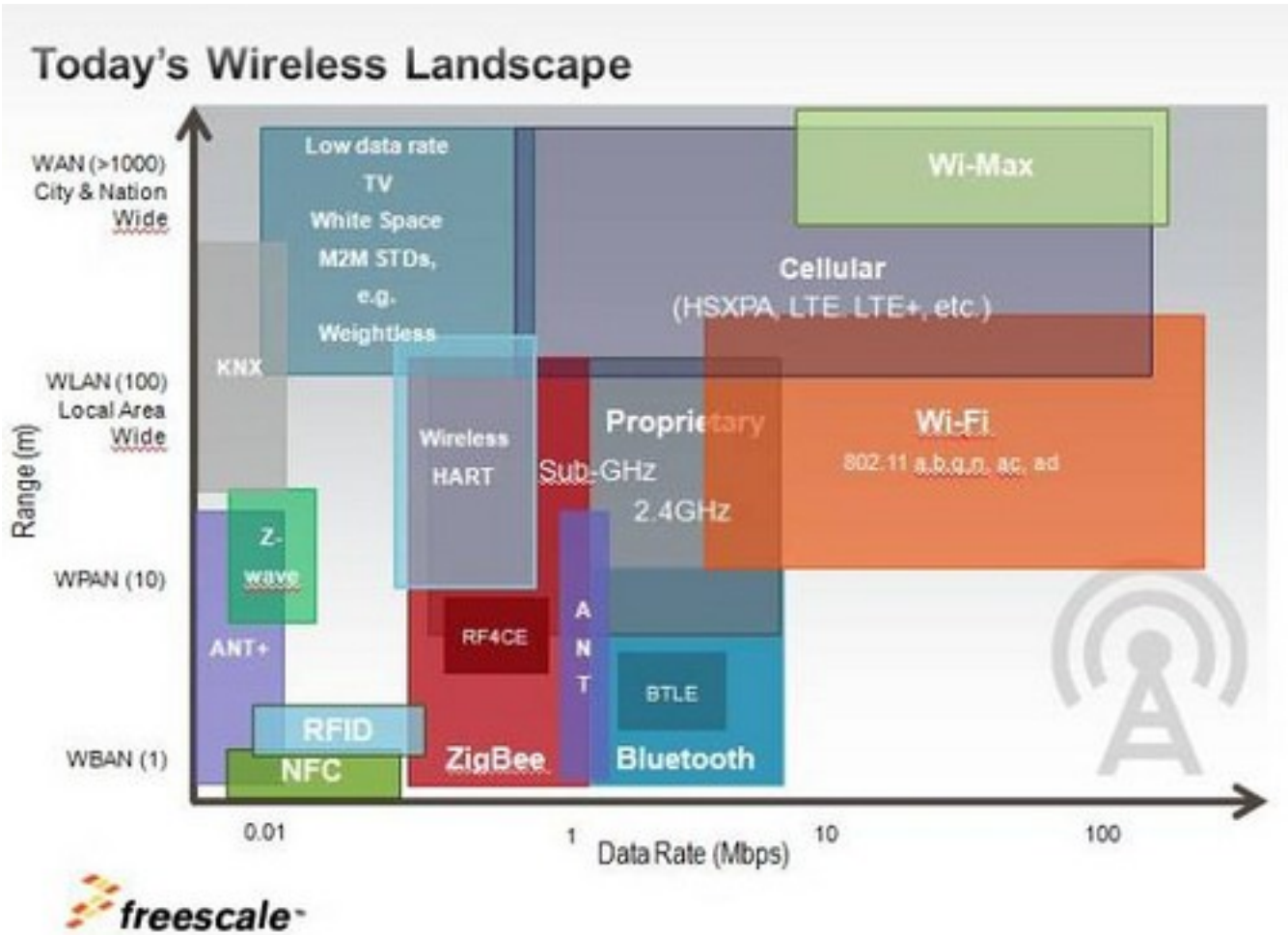
802.11n (OFDM) 40 MHz ch. width - 33.75 MHz used by sub-carriers



<http://www.dailywireless.org/2013/08/30/802-11ah-wifi-standard-for-900mhz/>



# TCP / UDP WiFi



<http://www.dailywireless.org/2013/08/30/802-11ah-wifi-standard-for-900mhz/>

# TCP / UDP

## Capa de transport d'Ethernet

UDP v/s TCP		
Characteristics/ Description	UDP	TCP
General Description	Simple High speed low functionality “wrapper” that interface applications to the network layer and does little else	Full-featured protocol that allows applications to send data reliably without worrying about network layer issues.
Protocol connection Setup	Connection less data is sent without setup	Connection-oriented; Connection must be Established prior to transmission.
Data interface to application	Message base-based is sent in discrete packages by the application.	Stream-based; data is sent by the application with no particular structure
Reliability and Acknowledgements	Unreliable best-effort delivery without acknowledgements	Reliable delivery of message all data is acknowledged.
Retransmissions	Not performed. Application must detect lost data and retransmit if needed.	Delivery of all data is managed, and lost data is retransmitted automatically.
Features Provided to Manage flow of Data	None	Flow control using sliding windows; window size adjustment heuristics; congestion avoidance algorithms
Overhead	Very Low	Low, but higher than UDP
Transmission speed	Very High	High but not as high as UDP
Data Quantity Suitability	Small to moderate amounts of data.	Small to very large amounts of data.



# TCP / UDP

## Capa de transport d'Ethernet

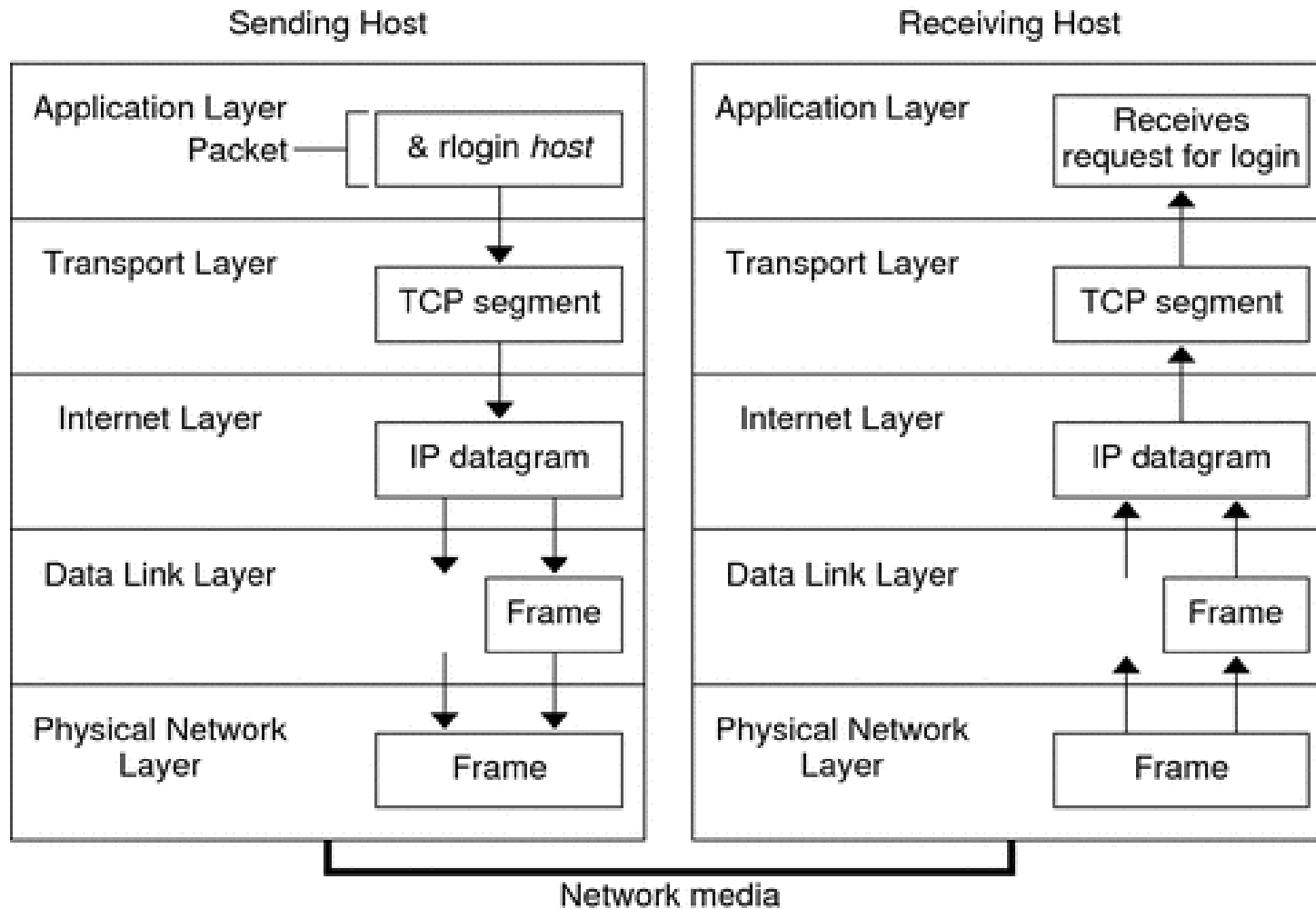
### OSI Model - Encapsulation

OSI Layer	Wrapper Name	Header Name	Frequency of usage
Application	N/A	Layer 7 Header	<i>Everytime (100%)</i>
Presentation	N/A	Layer 6 Header	<i>Everytime (100%)</i>
Session	N/A	Layer 5 Header	<i>Everytime (100%)</i>
Transport	Segment	TCP Header	<i>More common</i>
		UDP Header	<i>Less common</i>
Network	Packet	IPv4 Header	<i>Old Standard, still more common</i>
		IPv6 Header	<i>New Standard, still less common</i>
Data Link	Frame	Ethernet Type II Header	<i>More common</i>
		IEEE 802.2    802.3    802.3 SNAP	<i>Less common</i>
		Other Frame Headers	
Physical	Bits	N/A	<i>Everytime (100%)</i>

Original - JonathanCrosby.com

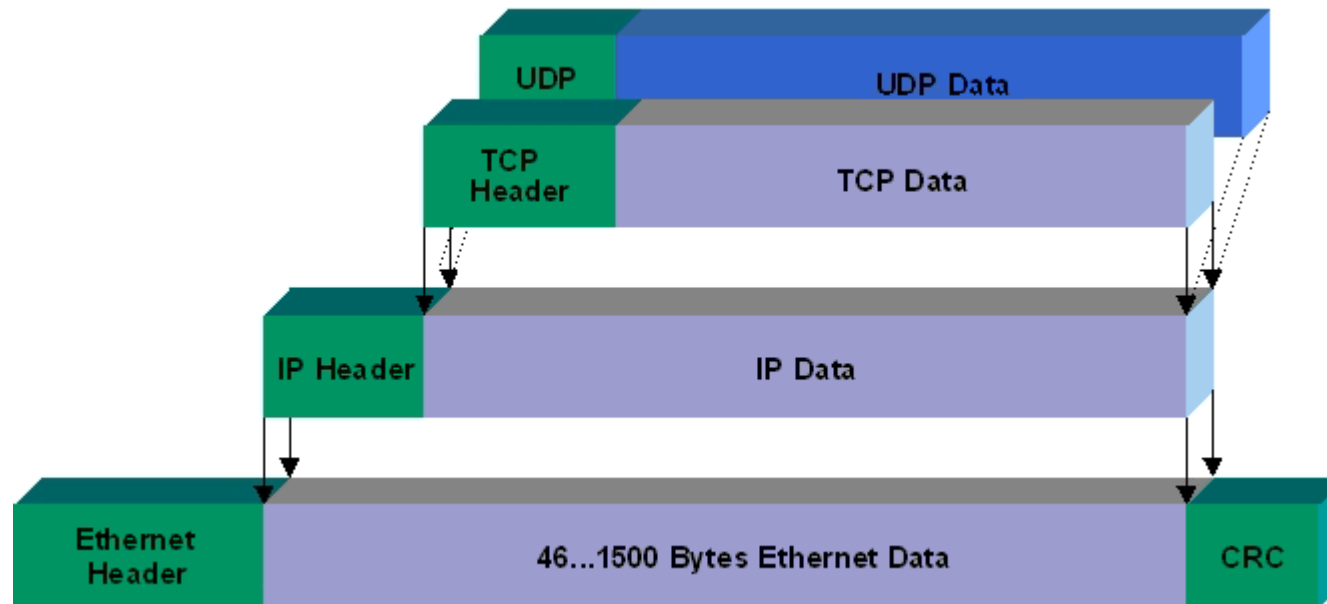
# TCP / UDP

## Capa de transport d'Ethernet



# TCP / UDP

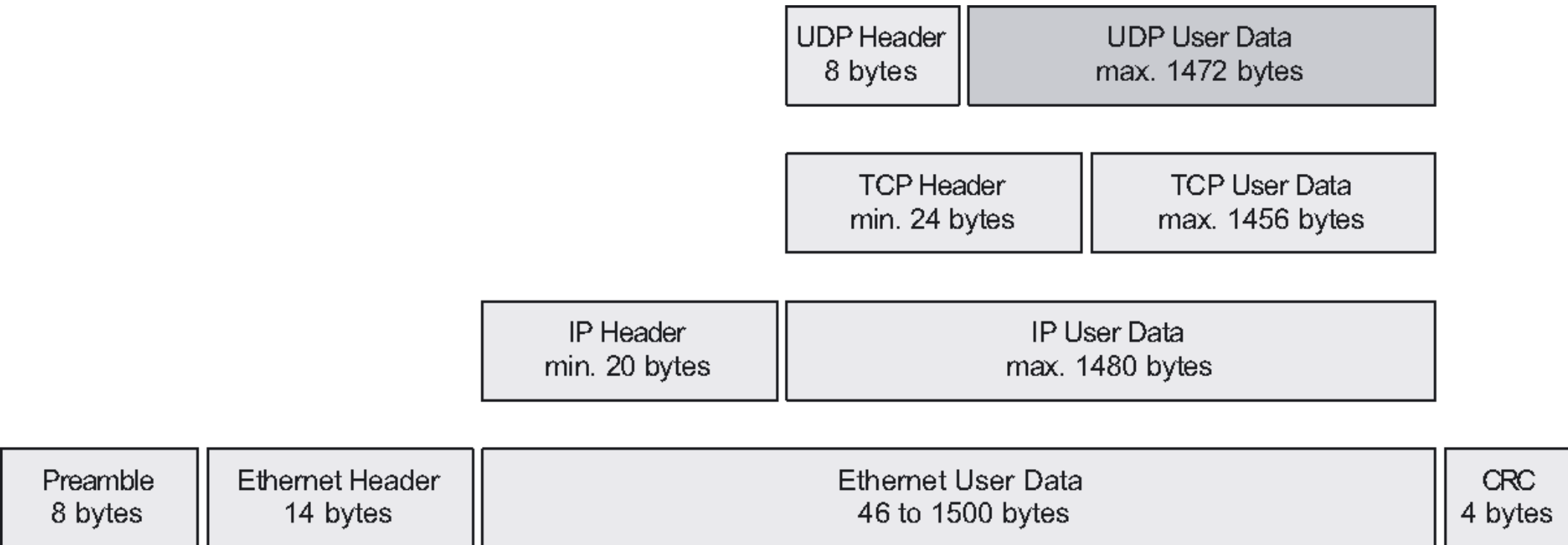
## Capa de transport d'Ethernet





# TCP / UDP

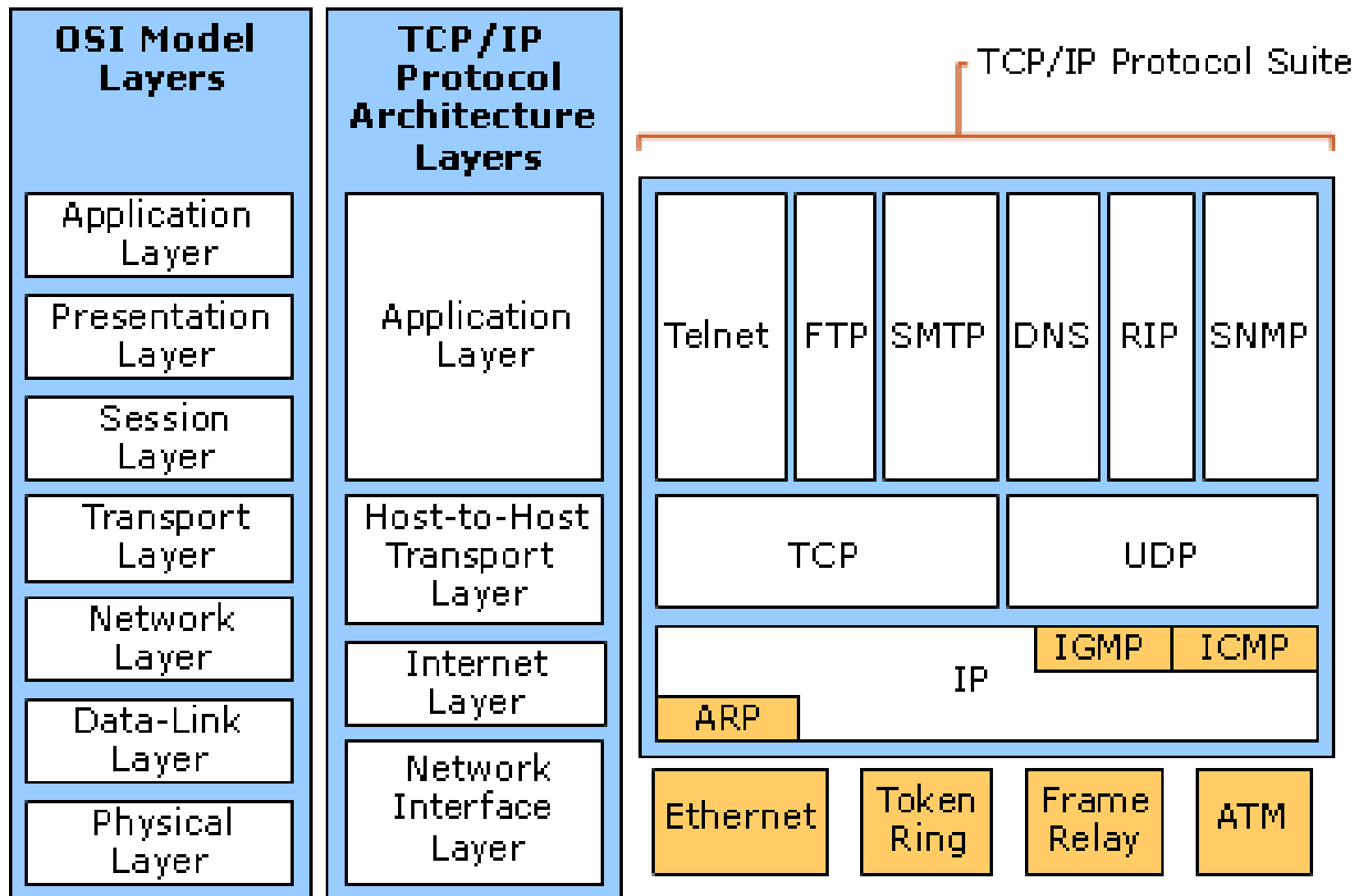
## Capa de transport d'Ethernet





# TCP / UDP

## Capa de transport d'Ethernet

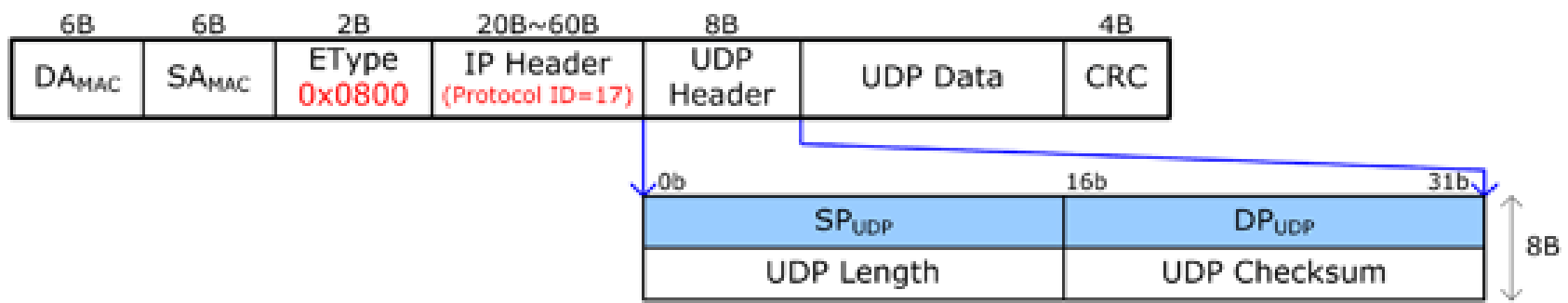




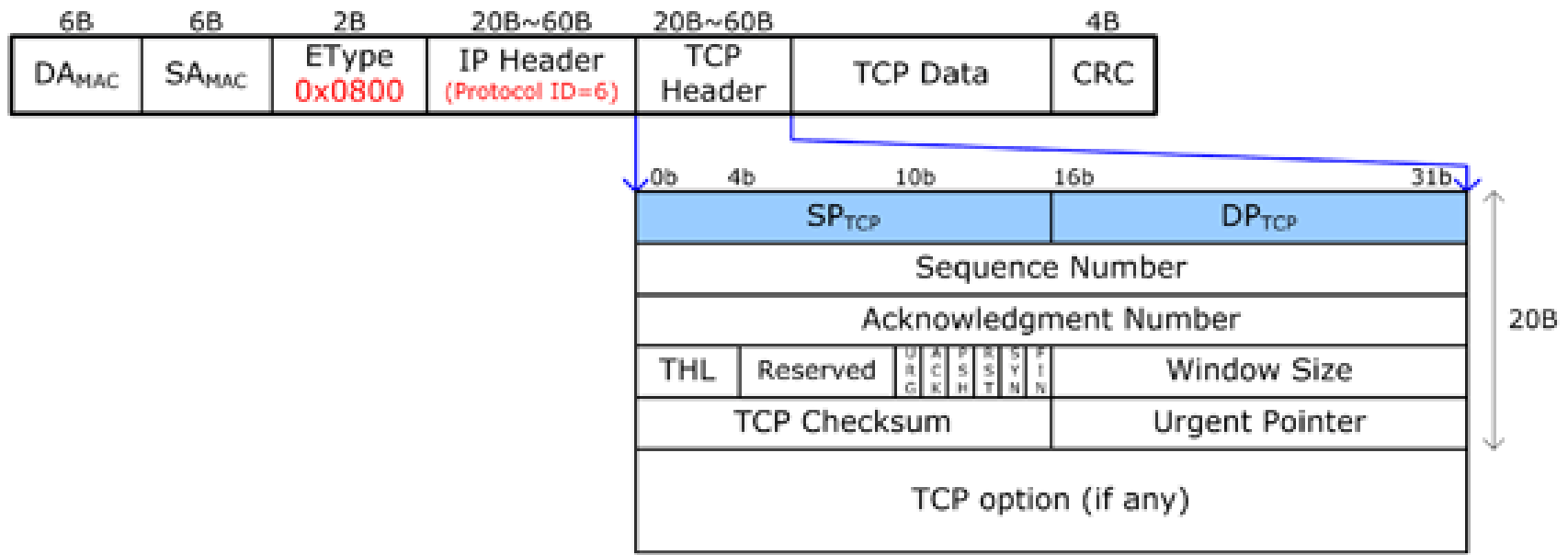
# TCP / UDP

## Capa de transport d'Ethernet

### UDP Packet



### TCP Packet



TCP Header (variable size, up to 60B, default 20B)



# TCP / UDP

## Capa de transport d'Ethernet

### TCP Segment Header Format

Bit #	0	7	8	15	16	23	24	31
0	Source Port				Destination Port			
32	Sequence Number							
64	Acknowledgment Number							
96	Data Offset	Res	Flags			Window Size		
128	Header and Data Checksum				Urgent Pointer			
160...	Options							

### UDP Datagram Header Format

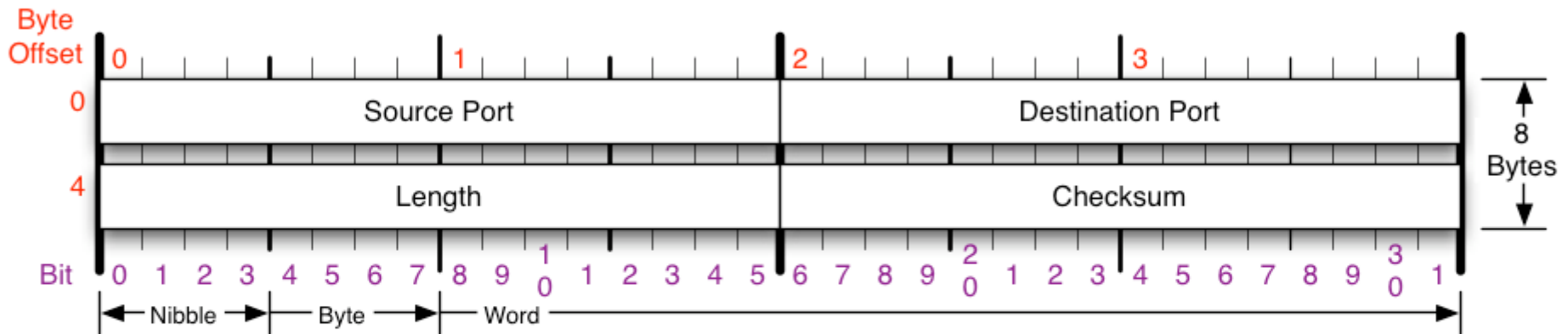
Bit #	0	7	8	15	16	23	24	31
0	Source Port				Destination Port			
32	Length				Header and Data Checksum			



# TCP / UDP

## Capa de transport d'Ethernet

### UDP Header



Checksum

Checksum of entire UDP segment and pseudo header (parts of IP header)

RFC 768

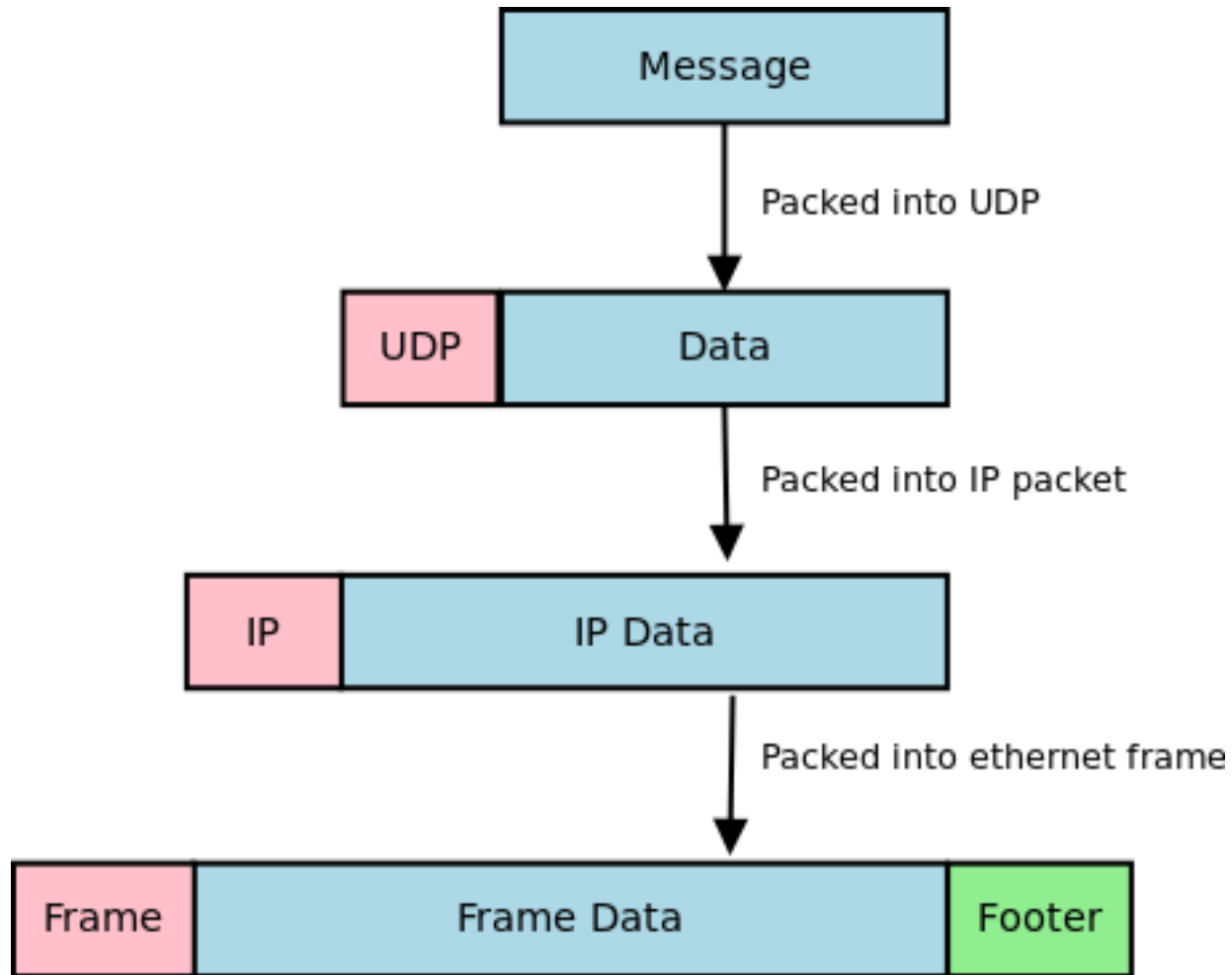
Please refer to RFC 768 for the complete User Datagram Protocol (UDP) Specification.

Copyright 2008 - Matt Baxter - [mjb@fatpipe.org](mailto:mjb@fatpipe.org) - [www.fatpipe.org/~mjb/Drawings/](http://www.fatpipe.org/~mjb/Drawings/)



# TCP / UDP

## Capa de transport d'Ethernet





# TCP / UDP

## Capa de transport d'Ethernet

### TCP, UDP and SCTP Comparison

Attribute	TCP	UDP	SCTP
Reliability	Reliable	Unreliable	Reliable
Connection Management	Connection-orientated	Connectionless	Connection-orientated
Transmission	Byte-orientated	Message-orientated	Message-orientated
Flow Control	Yes	No	Yes
Congestion Control	Yes	No	Yes
Fault Tolerance	No	No	Yes
Data Delivery	Strictly Ordered	Unordered	Partially Ordered
Security	Yes	Yes	Improved

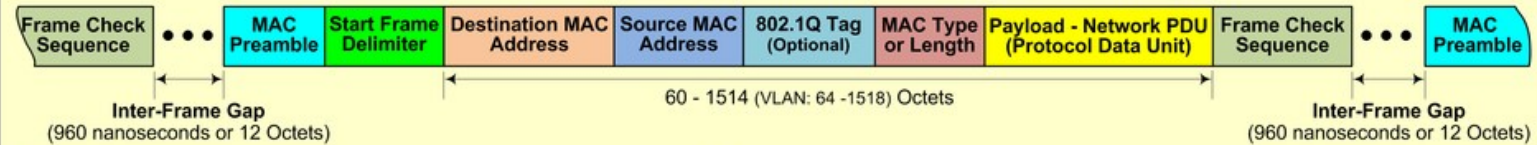


# TCP / UDP

## Capa de transport d'Ethernet

**Fast Ethernet (IEEE 802.3u) - UDP**  
 Maximum Ethernet frames and data throughput rate calculations.

**Fast Ethernet (IEEE 802.3u) Frame Structure with UDP Datagram**



Fast Ethernet Frame Component Size With UDP Datagram		
Frame Component	Component Size	
MAC Preamble	7 Octets of: 10101010	
Start Frame Delimiter	1 Octet of: 10101011	
Destination MAC Address	6 Octets	
Source MAC Address	6 Octets	
802.1Q VLAN TAG ID (Optional)	4 Octets (Optional)	
MAC Type or Length	2 Octets	
<b>MTU</b> (Maximum Transmission Unit)  Payload Network PDU Protocol Data Unit:	IP Header	20 Octets
	UDP Header	8 Octets
	Data/Padding	18 - 1472 Octets
	<b>Total:</b>	<b>46 - 1500 Octets</b> (Max: 1504 - VLAN)
Frame Check Sequence (CRC)	4 Octets	
Inter-Frame Gap	12 Octets (960 nanoseconds)	
<b>Total Physical Frame Size:</b>	<b>84 - 1538 Octets</b> (Max: 1544 -VLAN)	

Fast Ethernet Maximum Frame and Data Throughput Rate Calculation with UDP Datagram	
Rate Term	Value
Fast Ethernet Bit Rate	100 Mbit/sec -or- 100Mb/sec
Fast Ethernet Bit Time	10 nanoseconds (.0000001 seconds)
1 Octet (Byte)	8 Bits
<b>Max Octet Rate</b>	$(100\text{Mb/sec}) / (8 \text{ Bits}) =$ <b>12,500,000 Octets/sec</b>
<b>Max Frame Rate</b> (84 Octet Frames) Min Packet (60 Bytes + 4 Bytes CRC)	$(100\text{Mb/sec}) / ((8 \text{ Bits}) * (84 \text{ Octets/Frame})) =$ <b>148,810 Frames/sec (FPS)</b>
<b>Max UDP Data Rate</b> (84 Octet Frames) Min UDP Packet (60 Bytes + 4 Bytes CRC)	$(148,810 \text{ Frames/sec}) * (18 \text{ Bytes/Frame}) =$ <b>2,678,571 Bytes/sec</b>
<b>Max Frame Rate</b> (1538 Octet Frames) Max Packet (1514 Bytes + 4 Bytes CRC)	$(100\text{Mb/sec}) / ((8 \text{ Bits}) * (1538 \text{ Octets/Frame})) =$ <b>8,127 Frames/sec (FPS)</b>
<b>Max UDP Data Rate</b> (1538 Octet Frames) Max UDP Packet (1514 Bytes + 4 Bytes CRC)	$(8,127 \text{ Frames/sec}) * (1472 \text{ Bytes/Frame}) =$ <b>11,963,589 Bytes/sec</b>
<b>Max Fast Ethernet Frame Bandwidth</b> Max Packet (60 Bytes + 4 Bytes CRC) Max Packet (60 Bytes)	$(148,810 \text{ Frames/sec}) * (64 \text{ Bytes/Frame}) =$ <b>9,523,840 Bytes/sec</b> ( 9.082641 MiB/s) $(148,810 \text{ Frames/sec}) * (60 \text{ Bytes/Frame}) =$ <b>8,928,600 Bytes/sec</b> ( 8.514977 MiB/s)
<b>Max Fast Ethernet Frame Bandwidth</b> Max Packet (1514 Bytes + 4 Bytes CRC) Max Packet (1514 Bytes)	$(8,127 \text{ Frames/sec}) * (1518 \text{ Bytes/Frame}) =$ <b>12,336,786 Bytes/sec</b> (11.765276 MiB/s) $(8,127 \text{ Frames/sec}) * (1514 \text{ Bytes/Frame}) =$ <b>12,304,278 Bytes/sec</b> (11.734274 MiB/s)

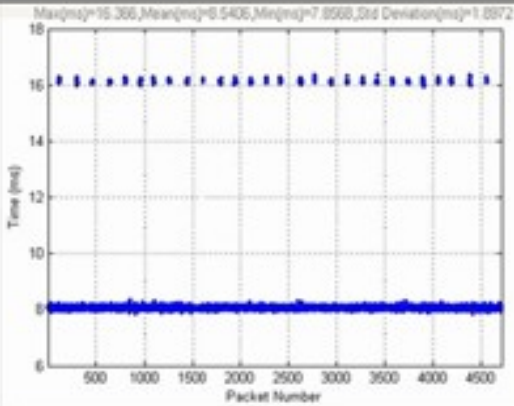
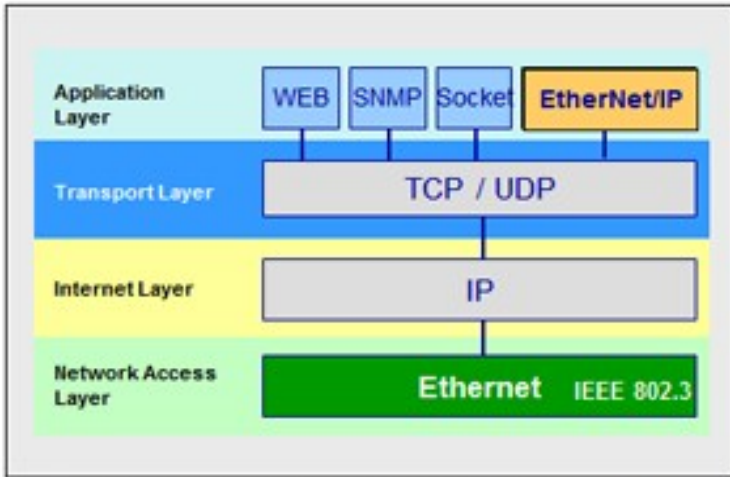
\*\*\* Note 1: Units - M: 1,000,000 Mi: 1,048,576





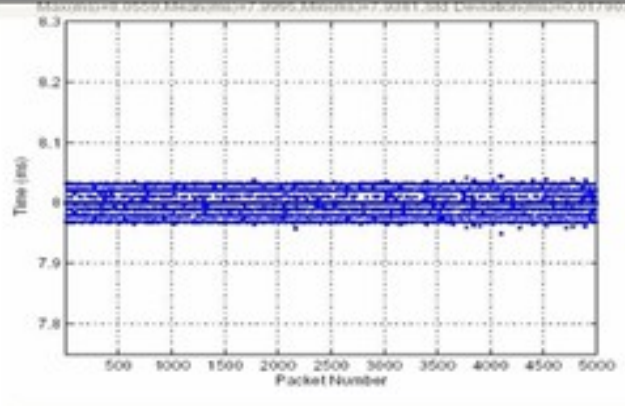
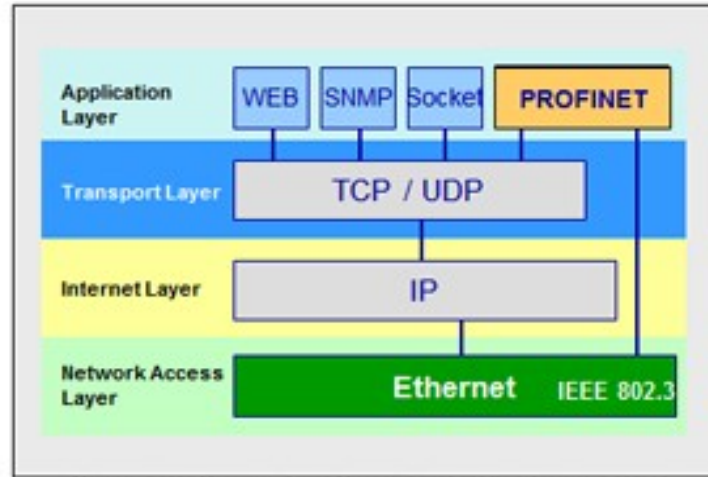
# TCP / UDP

## Capa de transport d'Ethernet



← 16 ms

← 8 ms

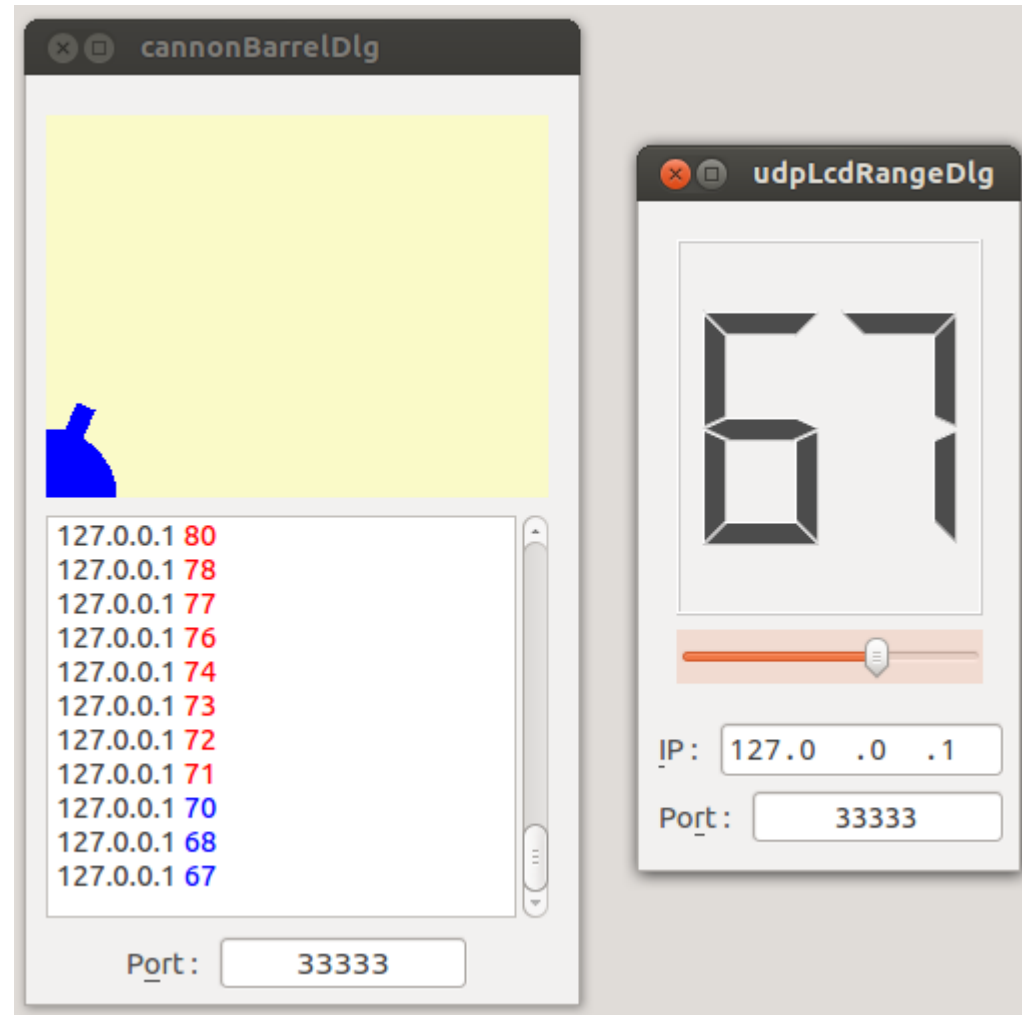


} Less than 100µs Jitter



# TCP / UDP

## Exemple amb UDP emprant Qt / C++



<http://binefa.cat/php/dam/m09uf3/20160120/cannonBarrel05.zip>

<http://binefa.cat/php/dam/m09uf3/20160120/lcdRangeUdp02.zip>



# TCP / UDP

## Exemple amb UDP emprant Qt / C++



<http://binefa.cat/php/dam/m09uf3/20160120/UDPbasic02.tar.gz>



# TCP / UDP

## Exemple amb UDP emprant Python

```
jordi@eCat: ~/Documents/elClot1516/damM09UF3/20160120/pyUdp
jordi@eCat:~/Documents/elClot1516/damM09UF3/20160120/pyUdp$ python udpServer.py
UDP server listening on port 6000
IP 192.168.1.13 , socket 50827 , said hola
IP 192.168.1.13 , socket 60876 , said hola

jordi@eCat: ~/Documents/elClot1516/damM09UF3/20160120/pyUdp
jordi@eCat:~/Documents/elClot1516/damM09UF3/20160120/pyUdp$ python udpClient02.py
Resposta del servidor : hola
jordi@eCat:~/Documents/elClot1516/damM09UF3/20160120/pyUdp$ python udpClient02.py
Resposta del servidor : hola
jordi@eCat:~/Documents/elClot1516/damM09UF3/20160120/pyUdp$
```

<http://binefa.cat/php/dam/m09uf3/20160120/pyUdp.tar.gz>

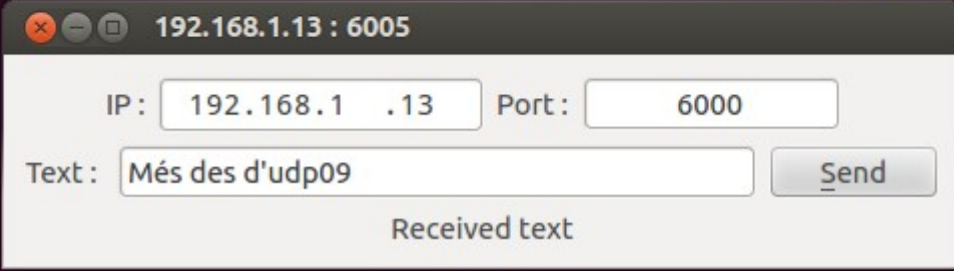


# TCP / UDP

## Exemple amb UDP emprant Qt / Python

```
jordi@eCat: ~/Documents/elClot1516/damM09UF3/20160120/pyUdp
jordi@eCat:~/Documents/elClot1516/damM09UF3/20160120/pyUdp$ python udpServer.py
UDP server listening on port 6000
IP 192.168.1.13 , socket 50827 , said hola
IP 192.168.1.13 , socket 60876 , said hola
IP 192.168.1.13 , socket 59835 , said Des d'udp09
IP 192.168.1.13 , socket 52101 , said Més des d'udp09

jordi@eCat: ~/Documents/elClot1516/damM09UF3/20160120/serialAndUdp
jordi@eCat:~/Documents/elClot1516/damM09UF3/20160120/serialAndUdp$ ./udp09.py
```



<http://binefa.cat/php/dam/m09uf3/20160120/pyUdp.tar.gz>

<http://binefa.cat/php/dam/m09uf3/20160120/serialAndUdp.zip>



# TCP / UDP

## Exemple amb UDP emprant Qt/Python i Qt/C++



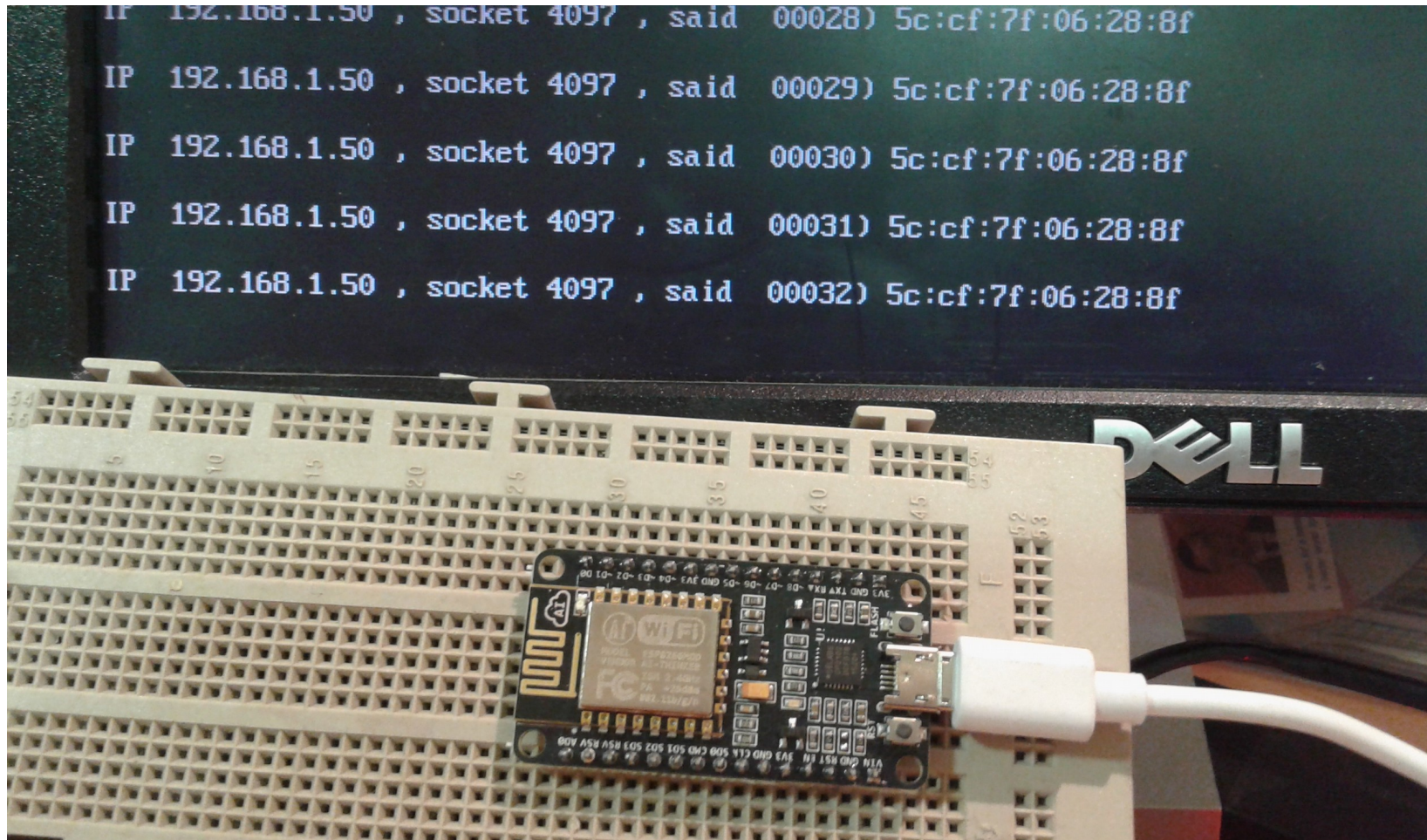
<http://binefa.cat/php/dam/m09uf3/20160120/serialAndUdp.zip>

<http://binefa.cat/php/dam/m09uf3/20160120/UDPbasic02.tar.gz>



# TCP / UDP

## Exemple amb UDP emprant ESP8266



<http://binefa.cat/php/dam/m09uf3/20160120/pyUdp.tar.gz>

<http://binefa.cat/php/dam/m09uf3/20160120/UDPClientMAC02/UDPClientMAC02.ino>