

STR9 Ethernet SpeedWay



Networking and Ethernet

2007

STM EMMA Prague
enrico.marinoni@silica.com

Content

- ▣ Networking

 - ▣ ISO OSI model – described layer by layer

- ▣ Ethernet

 - ▣ History

 - ▣ Standards Around Ethernet

 - ▣ Media Access Control (MAC) IEEE 802.3

Networking

STR9 Ethernet SpeedWay

Enrico Marinoni

2007



ISO OSI Model

	Layer	Function
7	Application	Network process to application
6	Presentation	Data representation and encryption
5	Session	Inter-host communication
4	Transport	End-to-end connections and reliability
3	Network	Path determination and logical addressing
2	Data link	(Physical) addressing
1	Physical	Media; signal and binary transmission

Physical Layer

Application
Presentation
Session
Transport
Network
Data link
Physical

- Media – connectors, cables
 - Metal: Coaxial cable, Twisted-Pair
 - Optical: Optical fiber
 - Radio: Wi-Fi, GSM, Bluetooth
 - Infra Red (IR)
- Signal
 - Modulation
 - Line coding
- Standards: RS-232, V.35, ISDN, 10BASE-T, 100BASE-FX, SONET, xDSL, 802.11b

Data Link Layer

Application
Presentation
Session
Transport
Network
Data link
Physical

▣ Framing:



▣ Header: Destination and Source (Link) **addresses**, Control information

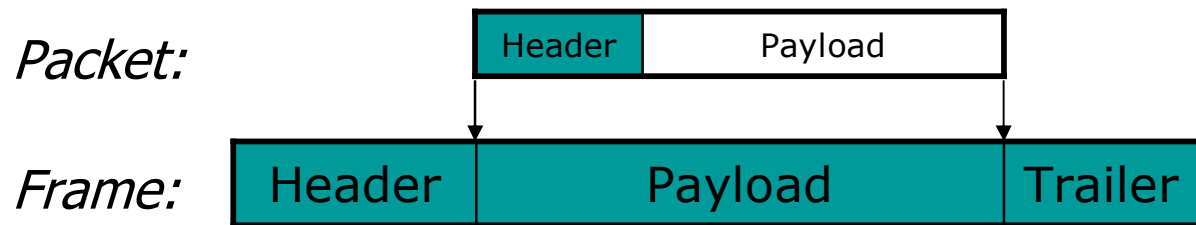
▣ Trailer: Checksum

▣ Standards: **Ethernet**, 802.11 (WiFi), token ring, FDDI, PPP, HDLC, Frame Relay, ATM, Fibre Channel

Network Layer

Application
Presentation
Session
Transport
Network
Data link
Physical

“Packetization”:



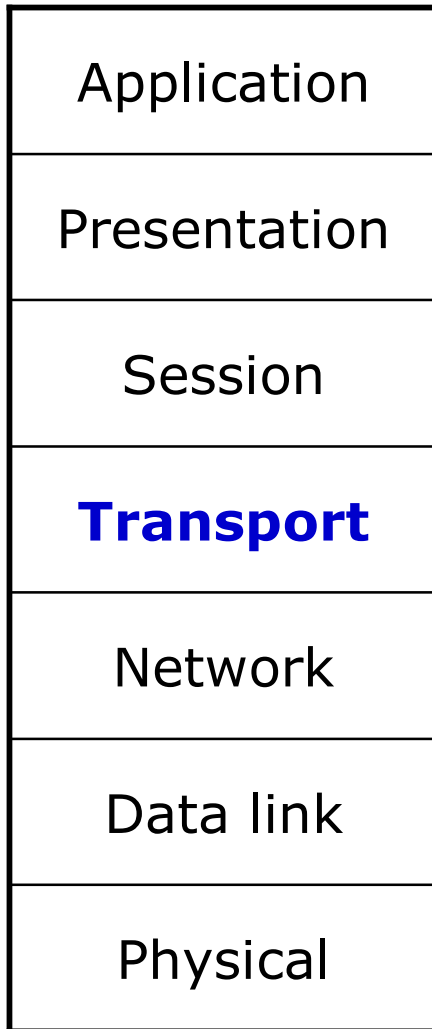
Path determination

network routing, flow control, network segmentation (e.g. by bridge) and error control functions

Logical addressing

Standards: **IP**, IPv6, IPSec, IPX, X.25

Transport Layer



☐ “Segmentation”:

Segment:



Packet:



Frame:



☐ Virtual Connections, Same Order Delivery, Reliable Data, Congestion Avoidance

☐ Standards: **TCP**, **UDP**, SPX, ATP

Session Layer

Application
Presentation
Session
Transport
Network
Data link
Physical

- ❏ Managing the dialogue between end-user application processes.
- ❏ Often unused, but there are exceptions e.g. web conferencing – to ensure audio and video match up
- ❏ Standards: Named Pipes (RPC), NetBIOS



Presentation Layer

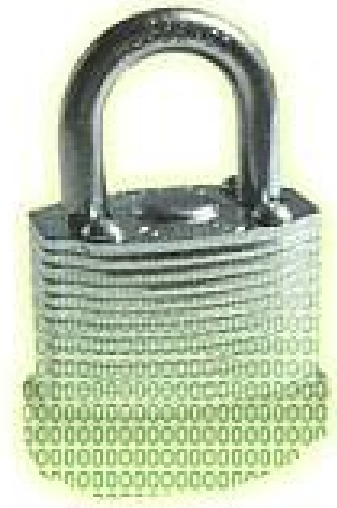
Application
Presentation
Session
Transport
Network
Data link
Physical

▣ Data formatting and encryption

```

P E R C U B B A 9 0 8 7 6 5 4 3 2 1 0
I I I I I I I I I I I I I I I I I I
H H H H H H H H H H H H H H H H H
Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y
K K K K K K K K K K K K K K K K
L L L L L L L L L L L L L L L L
J J J J J J J J J J J J J J J J
H H H H H H H H H H H H H H H H
Z Z Z Z Z Z Z Z Z Z Z Z Z Z Z Z
. . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . .
L O R D S A M I T H I S I S A M I T H
S E R V E R S A R E I N T H E C L O U D
N O W

```



▣ Standards: ASCII, Unicode, RSA, DES, GIF, MPEG



Application Layer

Application
Presentation
Session
Transport
Network
Data link
Physical

Network process to application: e.g. virtual file, virtual terminal and job transfer and manipulation protocols

Standards: **HTTP**, SMTP, SNMP, FTP, Telnet, SSH



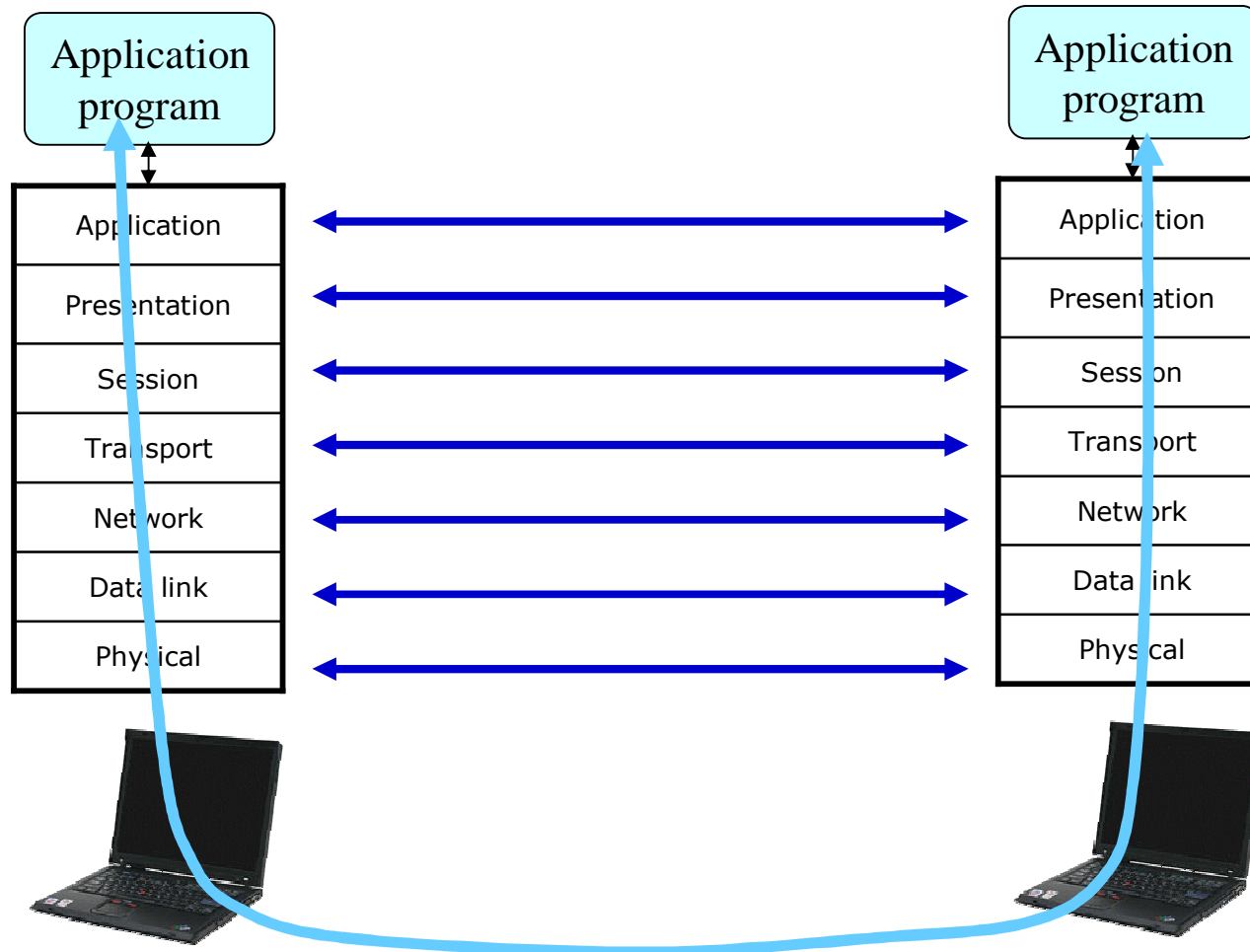
```
X hgaver@onella:~
[hgaver@onella hgaver]$ ssh isin
hgaver@isin's password:
Last login: Tue Apr  9 13:22:36 2002 from hyndla.admc.com
Welcome to TERMINAL SERVER MASTER 'heimdall'
TERM MASTER PORT MENU.  Enter one of the following.

example      format      resulting action
=====      =====      =====
Hit ENTER key
/.           /.           List all labelled target devices
stellar      portlabel   Connect to target device with given label
/dumbo      /regex     List target devices with matching labels
server/142   termserver/number Connect to port on given terminal server
=           =           List available Terminal Servers

(To connect with a non-default speed, you may append @speed to a port number
or portlabel specification. Examples: stellar@9600, server/142@9600).

PortLabel | Termserver/PortNumber | /Reg-Exp [QUIT]:
```

OSI Layers



Ethernet

STR9 Ethernet SpeedWay

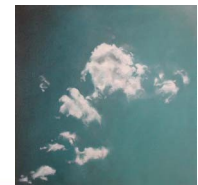
Enrico Marinoni

2007



History of Ethernet

- ❏ Ethernet was invented in the period of 1973-1975 by **Robert Metcalfe** and David Boggs at **Xerox** PARC. Patented in 1975 the experimental Ethernet ran at 3 Mbit/s and had 3-bit addressing.
- ❏ Metcalfe left Xerox in 1979 and founded **3Com**. He convinced DEC, Intel and Xerox to promote Ethernet as **standard**, the so-called "DIX" standard with 10 Mbit/s speed and 48-bit address.
- ❏ Thanks to J.H.Saltzer's paper suggesting that token-ring architectures were theoretically superior to Ethernet-style technologies, computer manufactures decided not to include Ethernet in their products, which allowed 3Com to build business around selling add-in Ethernet network cards.
- ❏ Ethernet originally worked on coaxial cable. It required a method to deal with collisions on the media – the **CSMA/CD** method was introduced in Ethernet.
- ❏ The *word "Ethernet"*: Searching a word to describe a medium that would be everywhere and serve for the propagation of electromagnetic waves into data packet, the word "**ether**", fallen in disuse in 1900s, was chosen.



STR9 Ethernet SpeedWay

Industrial Ethernet



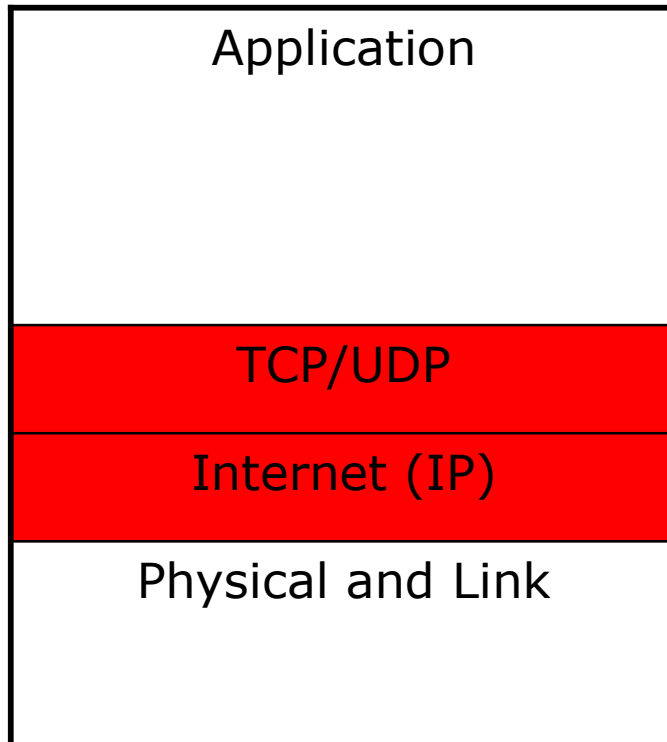
2007

STM EMMA Prague
enrico.marinoni@silica.com

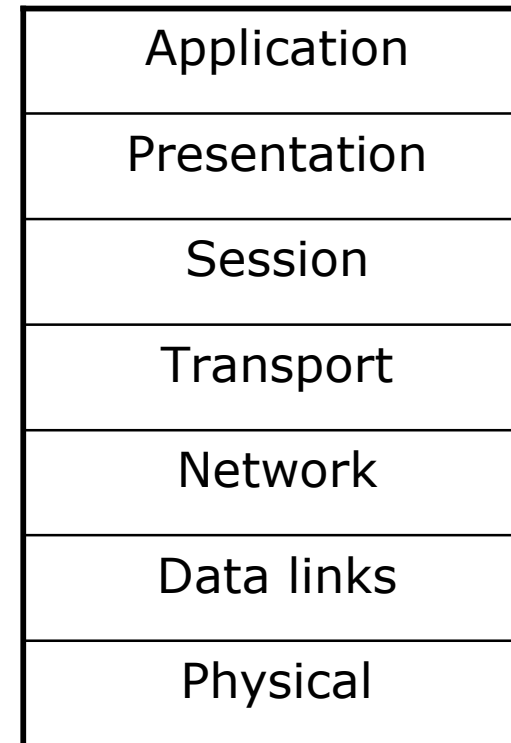
ISO OSI vs. TCP/IP

Ethernet

TCP/IP

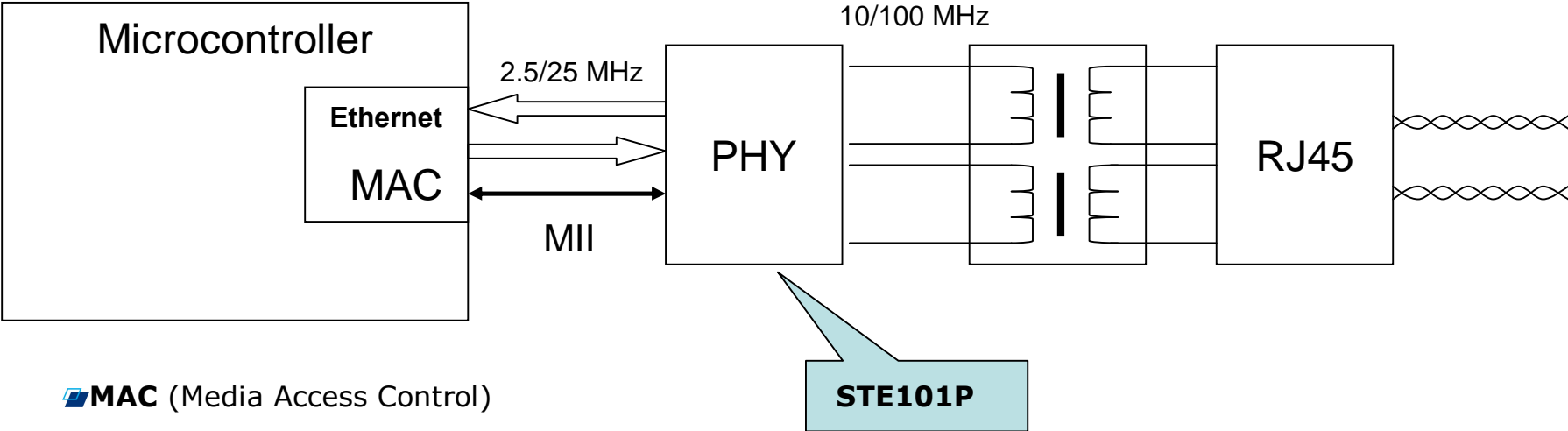


ISO7498/ OSI model



Ethernet

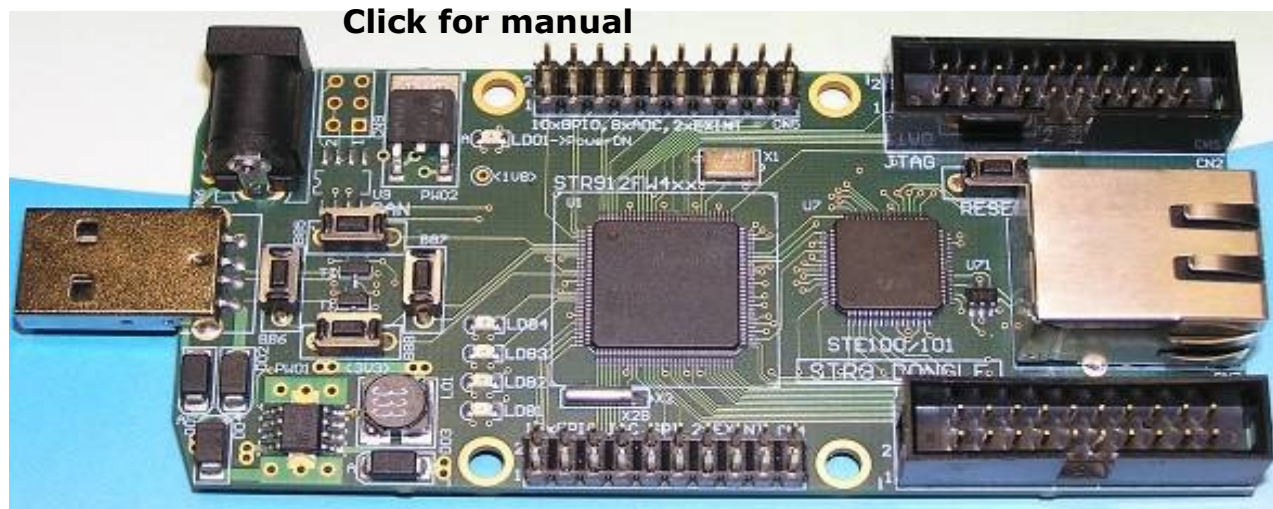
Ethernet



MAC (Media Access Control)

Ethernet TCP/IP Example

Ethernet



STR9 Ethernet SpeedWay

Enrico Marinoni

2007



ISO OSI vs. Real Time Ethernet

RT Ethernet

RT protocol

Application	
TCP/UDP	RT Ethernet
IP	
Custom Data link, Physical	

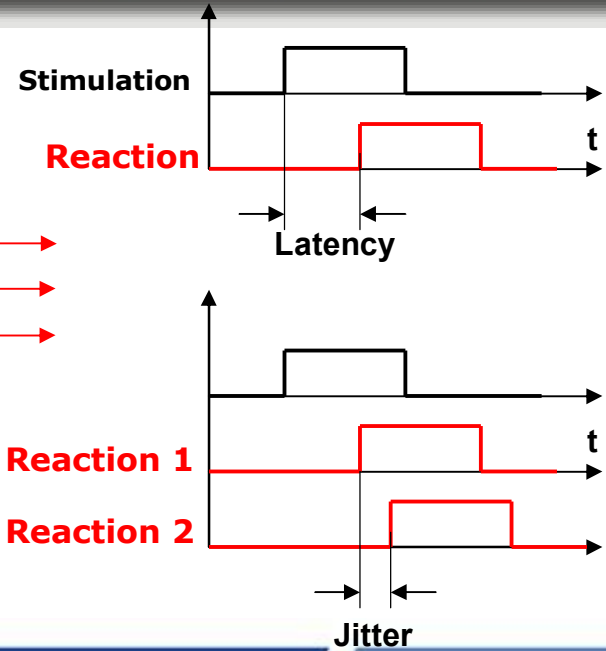
ISO7498/
OSI model

Application
Presentation
Session
Transport
Network
Data links
Physical



What means „Realtime“ ?

RT Ethernet



- 3 Parameters of Realtime:
- 1.) minimize Latency
 - 2.) maximize Data Throughput
 - 3.) Simultaneity: to minimize Jitter

What means „Ethernet Realtime“ ?

RT Ethernet

Hard Real Time

- Powerlink
- Sercos III
- Ethernet/IP with CIPsyn
- Profinet IRT
- EtherCAT

Soft Realtime

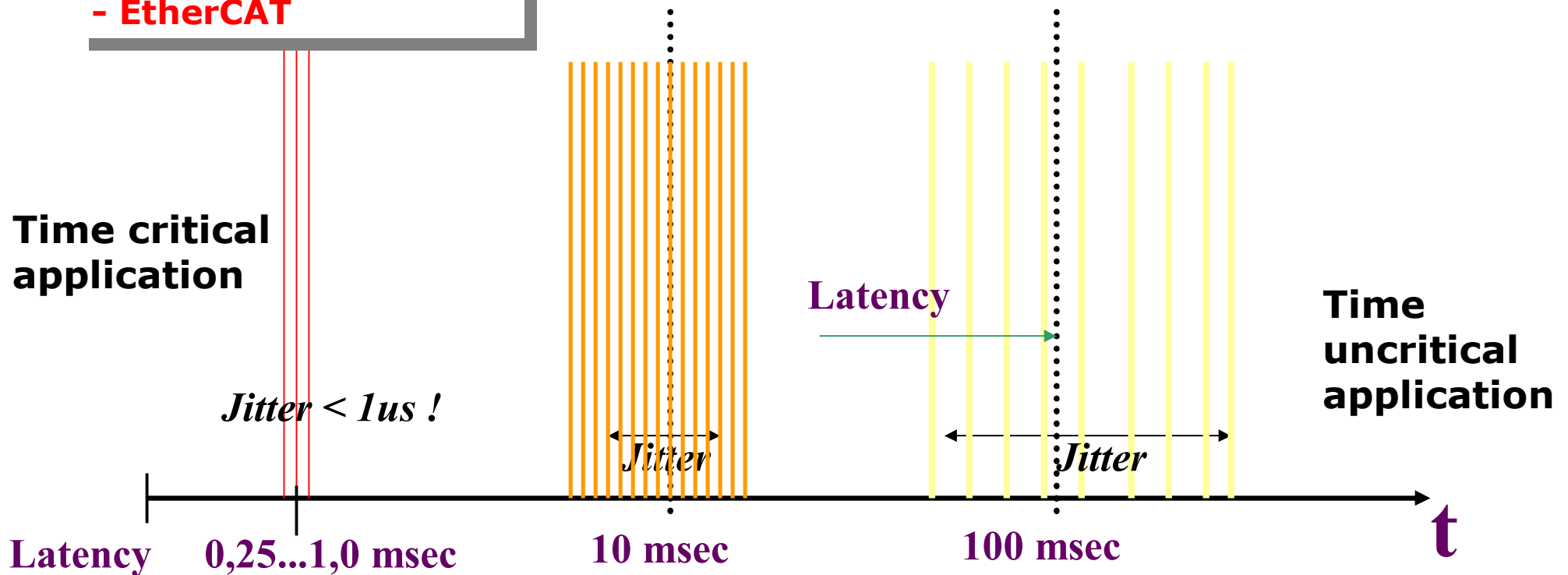
- Ethernet/IP
- Modbus/TCP
- Profinet SRT

Non Real Time

- all TCP/IP

Time critical application

Time uncritical application



STR9 Ethernet SpeedWay

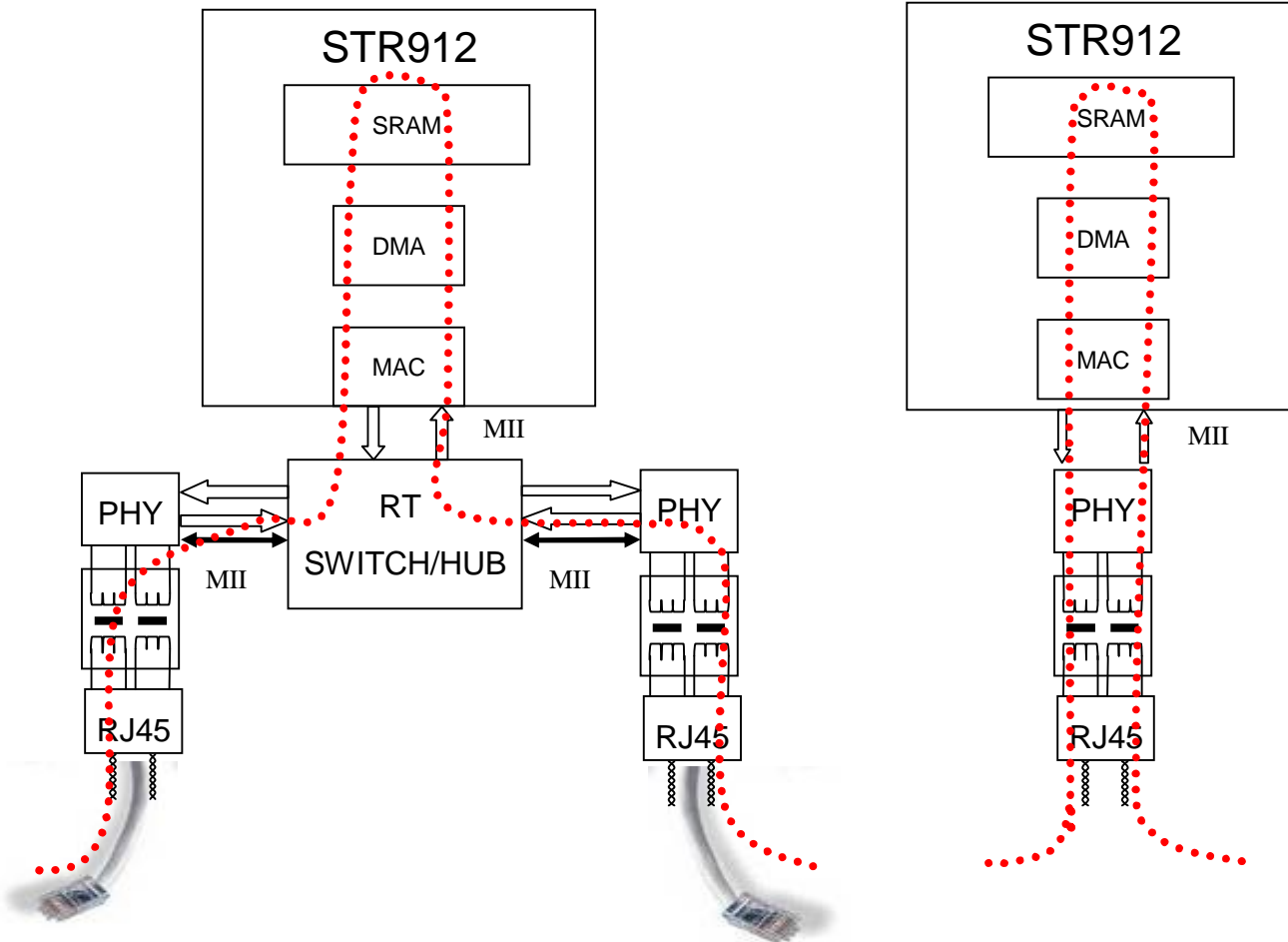
Enrico Marinoni

2007



RT Ethernet node

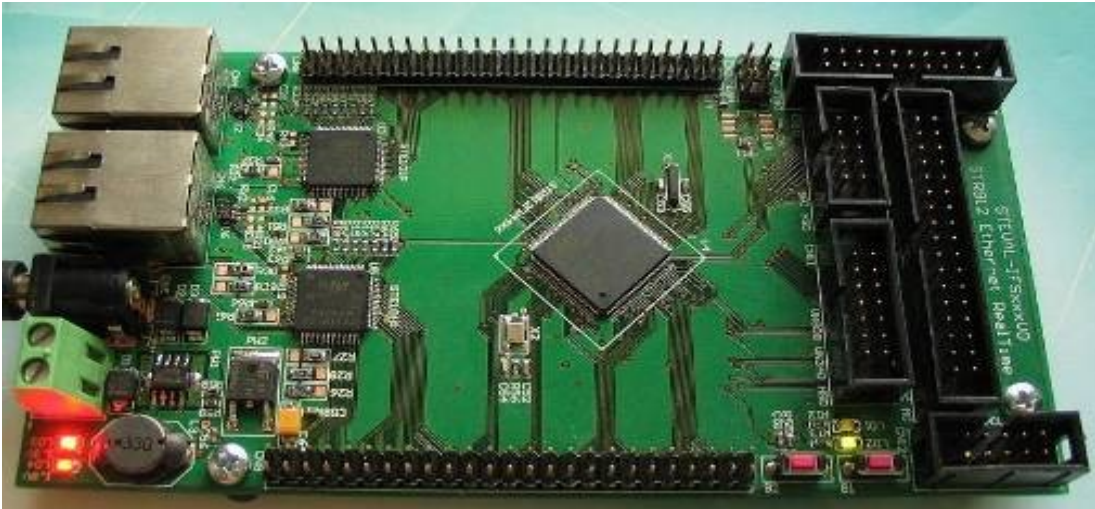
RT Ethernet



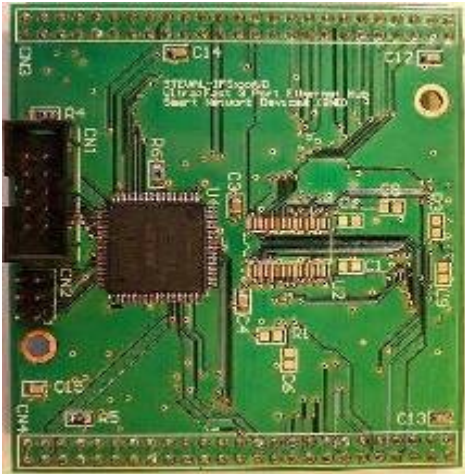
- Typical requirements:
- ◆ Fast response time (data read/write)
 - ◆ Minimized throughput delay of the data frame
 - ◆ Minimized Jitter
 - ◆ Daisy chain connectivity – no collisions on the medium

RT Ethernet node example

RT Ethernet



Dual PHY
Carry board



RT IP
extension

STR9 Ethernet SpeedWay

TCP/IP

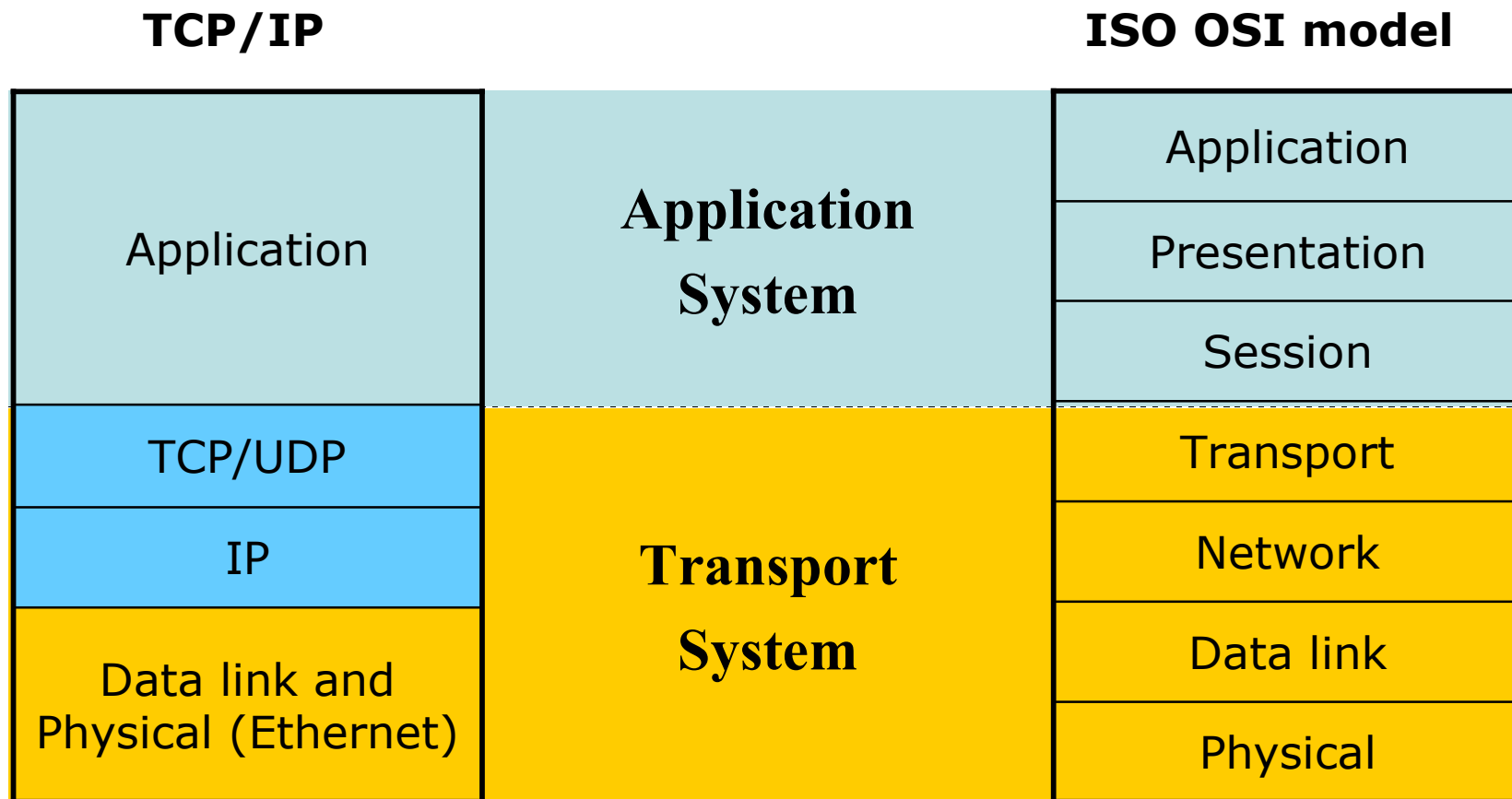


2007

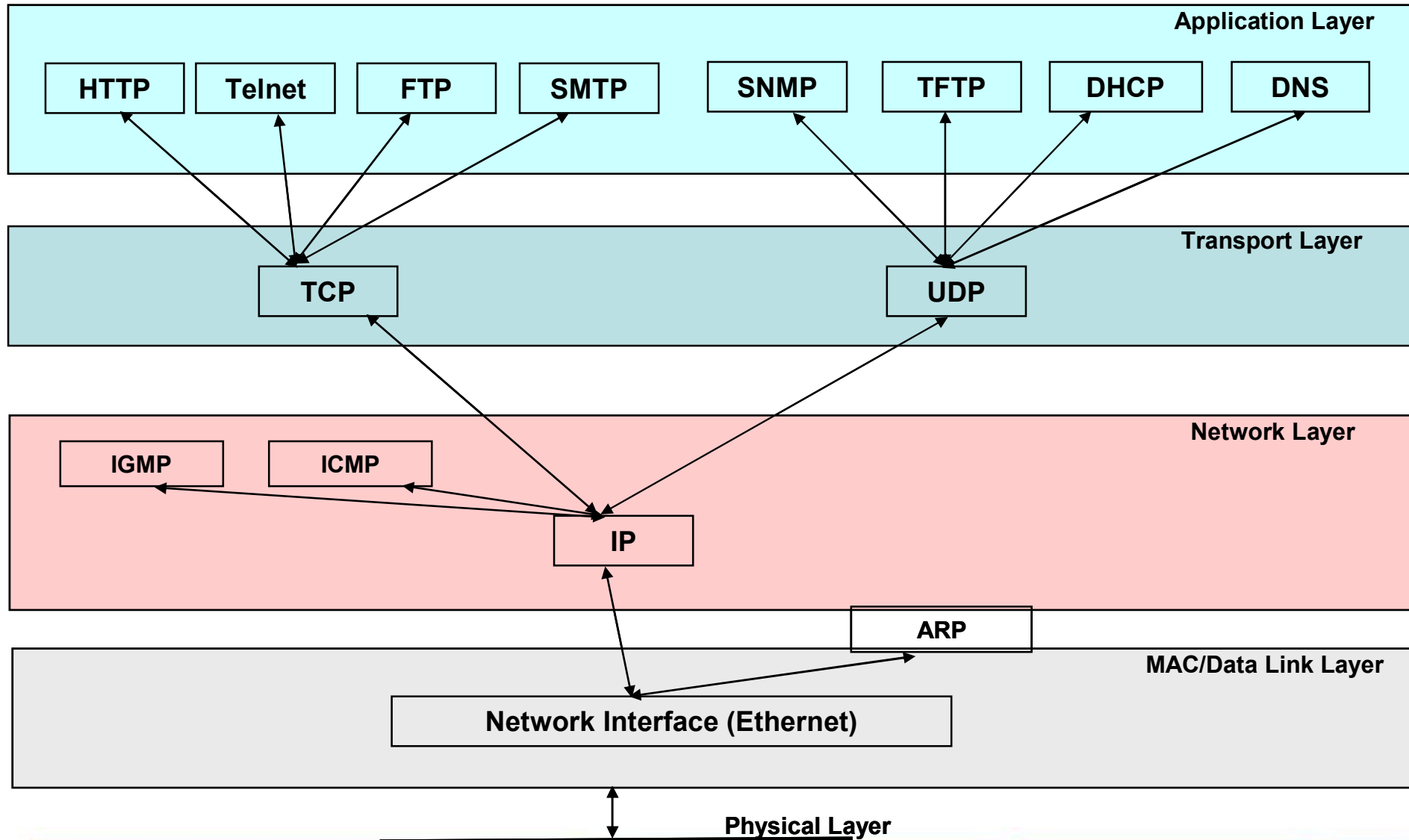
STM EMMA Prague
enrico.marinoni@silica.com

TCP / IP

TCP/IP vs. OSI



TCP/IP Stack



IP

Internet Protocol

IP Protocol

- ❏ The **Internet Protocol (IP)** is a data-oriented [protocol](#) used for communicating data across a [packet-switched internetwork](#).
 - ❏ “Packetization” – encapsulation of data
 - ❏ Fragmentation – breaking packets into fragments small enough to pass over a link with smaller frame size
 - ❏ Routing – Addressing and Path determination
 - ❏ Reliability – IP is **not reliable**, protocols of upper layers have to be used
- ❏ **IPv4 Internet Protocol version 4** is the fourth iteration of the [Internet Protocol](#) (IP) and it is the first version of the protocol to be widely deployed.
IPv4 is the dominant [network layer](#) protocol on the [Internet](#) and apart from [IPv6](#) it is the only protocol used on the [Internet](#).

IP Packet Structure

+	Bits 0-3	4-7	8-15	16-18	19-31
0	Version specifies if it's an IPv4 or IPv6 packet	Header length the length of the header	Type of Service also referred to as Quality of Service (QoS) - describes what priority the packet should have (used in streaming multimedia applications such as voice over IP)	Total Length	
32	Identification helps reconstruct the packet from several fragments			Flags e.g. a flag that says if the packet is allowed to be fragmented or not	Fragment Offset a field to identify which fragment this packet is attached to
64	Time to Live the number of hops (router, computer or device along a network) the packet is allowed to pass before it dies (for example, a packet with a TTL of 16 will be allowed to go across 16 routers to get to its destination before it is discarded)		Protocol TCP, UDP, ICMP, etc...	Header Checksum	
96	Source IP Address				
128	Destination IP Address				
160	Options (optional)				
160 or 192+	Data				

Maximal data size = Max. Ethernet Payload - IP Header = 1500 - 20 = 1480 bytes

TCP

Transmission Control Protocol

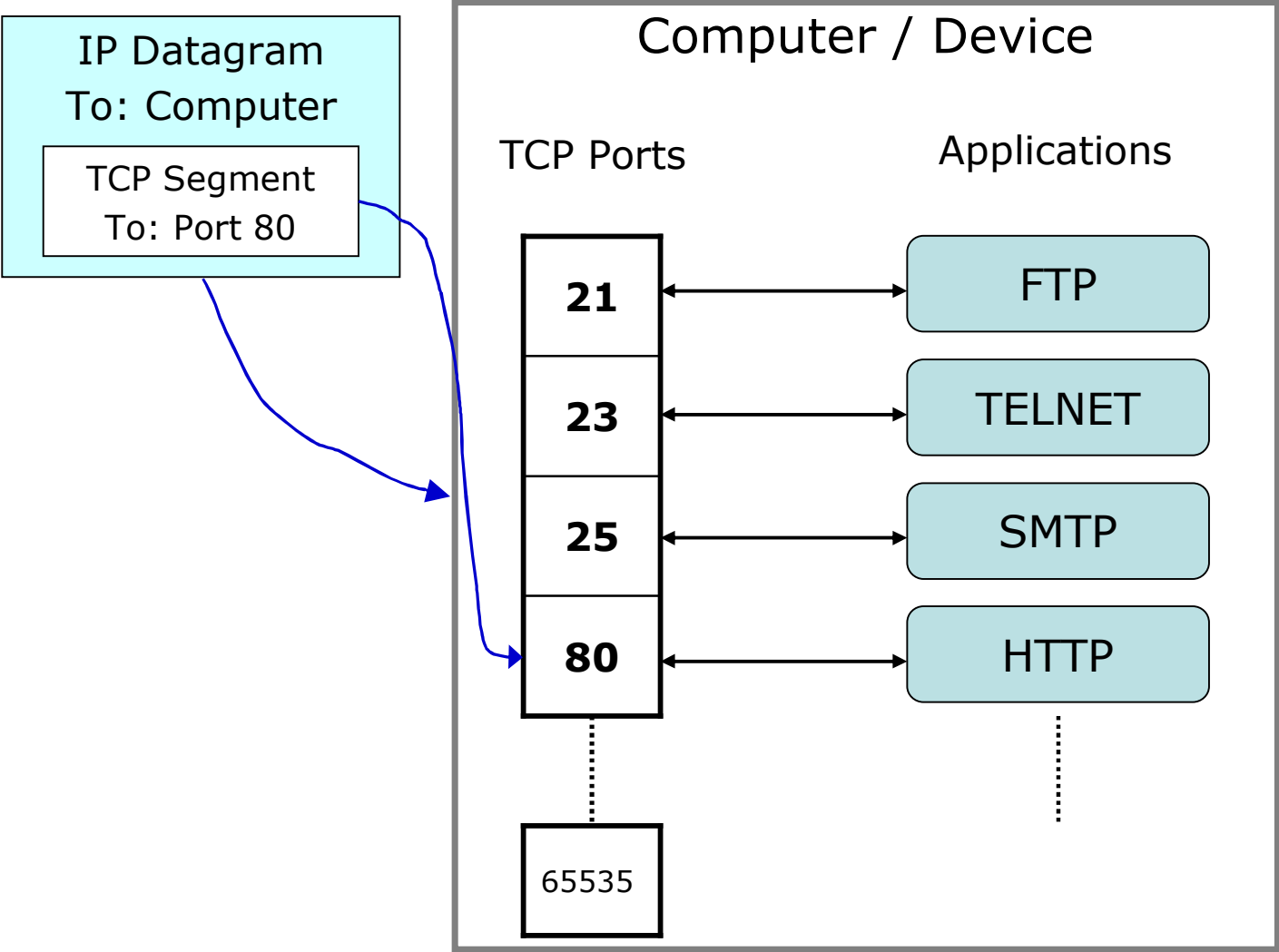
TCP Protocol

- ❏ The **T**ransmission **C**ontrol **P**rotocol is one of the core protocols of the [Internet protocol suite](#), often simply referred to as [TCP/IP](#). Using TCP, applications on networked hosts can create *connections* to one another, over which they can exchange streams of data using [Stream Sockets](#).
- ❏ The protocol guarantees reliable and in-order delivery of data from sender to receiver.
- ❏ TCP protocol avoids **congestion** using **window** mechanisms

TCP Segment Structure

+	Bits 0–3	4–7	8–15	16–31
0	Source Port the sending port		Destination Port the receiving port	
32	Sequence Number the sequence number of the first data byte in TCP segment			
64	Acknowledgment Number the sequence number the sender expects next, it says to the other side how many data bytes were already received			
96	Data Offset specifies the size of the TCP header in 32-bit words. The minimum size header is 20 bytes and maximum 60 bytes	Reserved	Flags ACK (Acknowledgement), PSH (Push function – application data), RST (Reset/refuse the connection), SYN (Synchronize sequence numbers), FIN (No more data from sender), URG (Urgent pointer field – see below – is significant)	Window the number of bytes the sender is willing to receive starting from the acknowledgement field value – used for Congestion avoidance
128	Header Checksum		Urgent Pointer if the URG flag is set, then this 16-bit field is an offset from the sequence number indicating the last urgent data byte – used by routers	
160	Options (optional)			
160/ 192 +	Data			

TCP Ports



UDP

User Datagram

Protocol

UDP

✚ The **User Datagram Protocol (UDP)** is one of the core protocols of the [Internet protocol suite](#). Using UDP, programs on networked computers can send short messages sometimes known as [datagrams](#) to one another. UDP is sometimes called the **Universal Datagram Protocol** or Unreliable Datagram Protocol.

✚ **UDP does not provide the reliability and ordering that [TCP](#) does. Datagrams may arrive out of order, appear duplicated, or go missing without notice.**

Without the overhead of checking whether every packet actually arrived, **UDP is faster and more efficient** for many lightweight or time-sensitive purposes. Also, its stateless nature is useful for servers that answer small queries from huge numbers of clients.

UDP vs. TCP

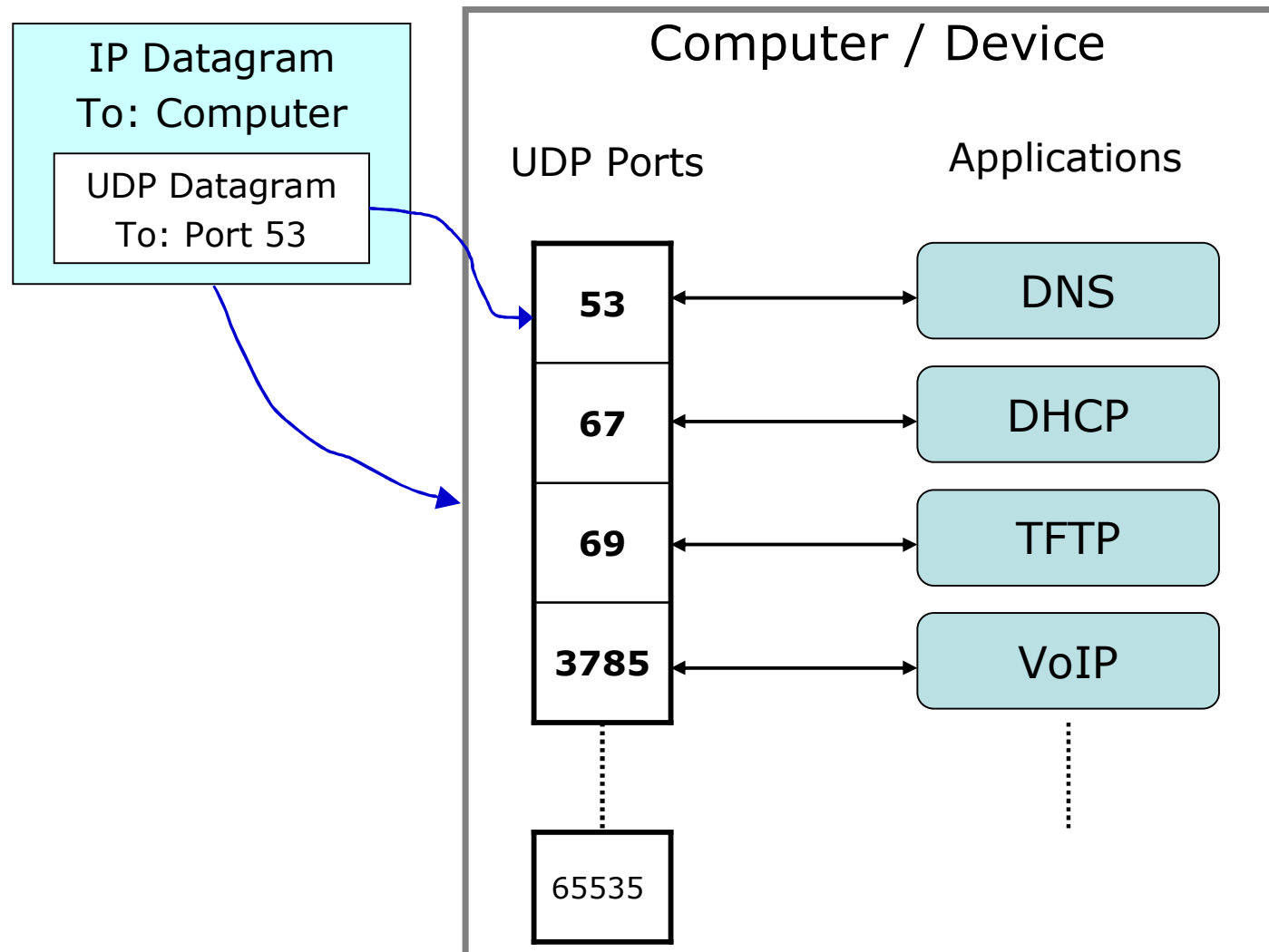
UDP	TCP
Unreliable – sender does not know if packet was received, when he sends a packet it can be lost on the way	Reliable – if sender sends a packet, he knows it arrived, unless the connection has failed, if the packet is lost server will re-request it
Not ordered – when sender sends two packets he does not know if they will be received in the same order	Ordered – packets are always received in the right order
Lightweight – there is no overhead caused by connection tracking no ordering, it is faster	Heavyweight – reliability, congestion avoidance etc. take time to process
Datagrams – packets are sent individually	Streaming – packets are sent as a continuous "stream"

UDP Datagram

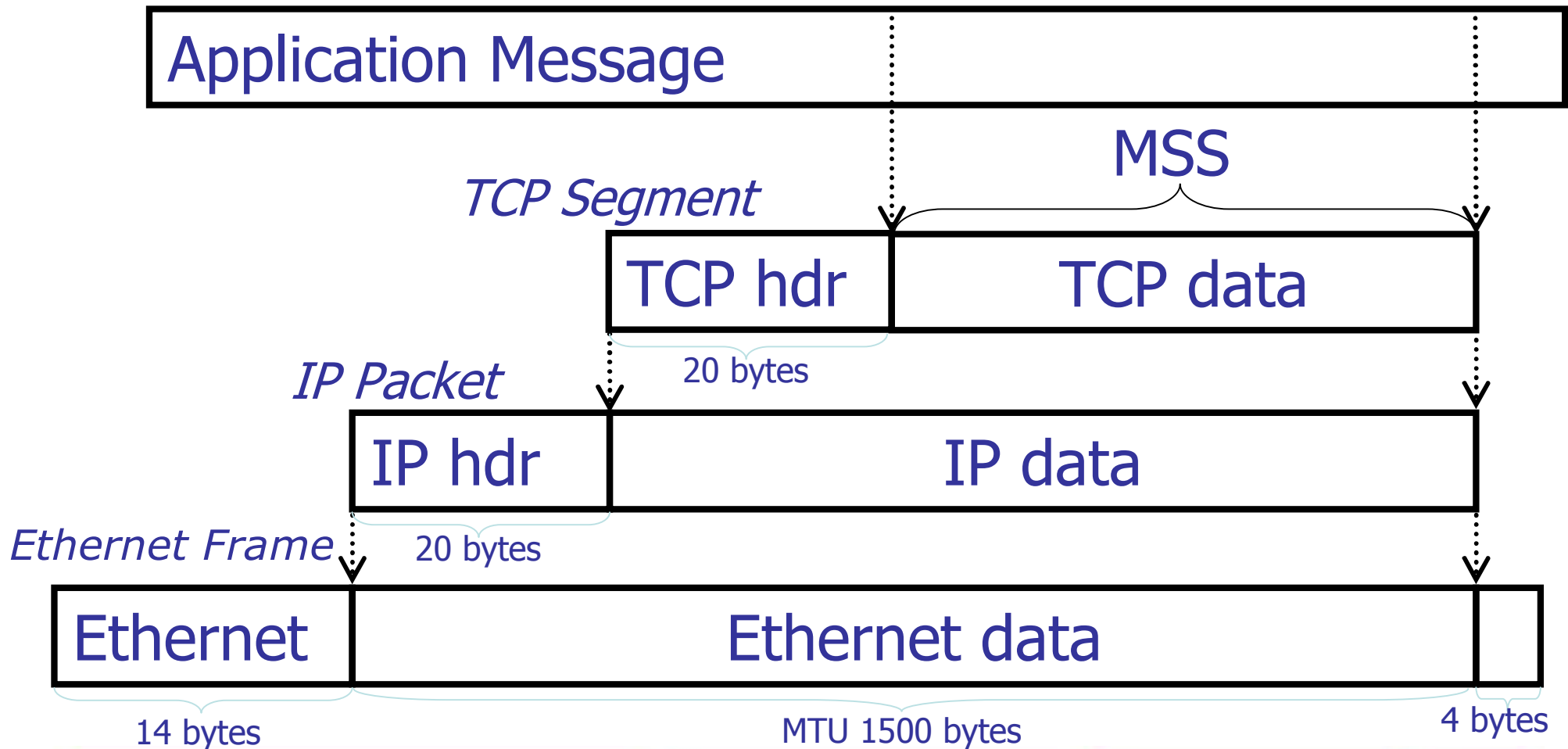
+	Bits 0 - 15	16 - 31
0	Source Port	Destination Port
32	Length	Checksum
64	Data	

- ❏ **Source port** – identifies the sending port when meaningful and should be assumed to be the port to reply to if needed. If not used then it should be zero.
- ❏ **Destination port** – field identifies the destination port and is required.
- ❏ **Length** – the length in bytes of the entire datagram: header and data. The minimum length is 8 bytes since that's the length of the header.
- ❏ **Checksum** – the checksum field is used for error-checking of the header and data.

UDP Ports



Packet Terminology Overview



TCP/IP Stacks

	uCOS	uIP	LWIP	TCPnet	easyWEB (it is just a demo)
Flash memory footprint	20076	5664	13830	26464	8019
RAM memory footprint	13954	988	9825	21800	2584
Pros	<ul style="list-style-type: none"> • Well defined interface • Well structured code • Good support 	<ul style="list-style-type: none"> • The best in code size • Very good for 8/16 bit microcontrollers • Doesn't cost a penny 	<ul style="list-style-type: none"> • Well defined interface • Doesn't cost a penny • Fast enough 	<ul style="list-style-type: none"> • Well defined interface • Suitable to run standalone or with RTOS • Good support 	<ul style="list-style-type: none"> • Well defined interface • Well structured code • Small footprint
Cons	<ul style="list-style-type: none"> • Licensing costs • Code size more • Needs the support of a RTOS 	<ul style="list-style-type: none"> • No support • If we want to develop few apps we have to have a better understanding of the code since it doesn't have a good interface 	<ul style="list-style-type: none"> • No support • If we want to develop few apps we have to have a better understanding of the code since it doesn't have a good interface • Needs the support of a RTOS 	<ul style="list-style-type: none"> • Requires Keil's RT library => Licensing costs 	<ul style="list-style-type: none"> • Only one webpage • Only one connection at a time • No pictures

STR91x Firmware and Libraries

Reference	Description
www.stmcu.com	
STR910-EVAL	STR91x demonstration firmware
STR91M	STR91x MAC/DMA controller (ENET) firmware Library
STR91S	STR91x standard firmware Library
STR91U	STR91x USB firmware library