École Pour l'Informatique et les Techniques Avancées – EPITA

BSc L1 – 15 May 2024

Course: Introduction to Computer Networks

Course instructor: M Salman Nadeem
mohammad-salman.nadeem@epita.fr

Introduction to Computer Networks

Course schedule (tentative)

Date & Time	No.	Topics	Duration (hours)
Fri 19/04/24 – 10:00–13:00	1	Primer, Network protocols, types, topology, architecture	3
Fri 26/04/24 – 10:00–13:00	2	Network models, TCP/IP model, Packet switching	3
Sat 27/04/24 – 10:00–13:00	3	Physical Layer (Function, Signals, Modulation, Multiplexing, Transmission media & Hardware, Optical networks)	3
Sat 27/04/24 – 14:00–17:00	4	Data Link Layer (Function, Framing, Protocols, Flow control, Access control, Error correction, Hardware)	3
Fri 03/05/24 – 14:30–17:30	5	Network Layer (Function, IP addressing and subnets)	3
Sat 04/05/24 – 10:00–13:00	6	Network Layer (Routing algorithms and protocols), Internet Control Message Protocol	3
Tue 14/05/24 – 16:30–19:30	7	Network Layer (IGP & EGP), Autonomous System, Border Gateway Protocol	3
Wed 15/05/24 – 14:30–17:30	**8**	**Transport Layer (Function, Flow and congestion controls, Protocols)**	**3**
Thu 16/05/24 – 11:15–13:15	9	Application Layer (Function, Protocols)	2

Introduction to Computer Networks

Course schedule (tentative)

Date & Time	No.	Topics	Duration (hours)
Fri 17/05/24 – 14:00-17:00	10	Cross-layer process: Access Control Lists	3
Sat 18/05/24 – 14:00-17:00	11	Cross-layer process: Network Address Translation	3
Fri 24/05/24 – 10:00-13:00	12	Review / Open-session	3
		Total	***35***
Fri 31/05/24 – 14:30-15:30		EXAM	1

GRADING criteria :

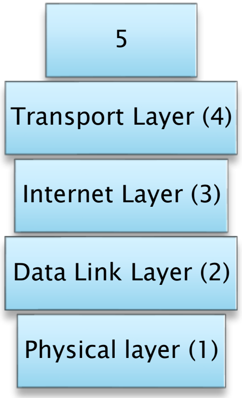
- Class participation comprising **attendance** & **reactivity**): 10%
- **Exercises** (practical work): 40%
- Final evaluation (**Quiz** & **Exercises**): 50%

ℹ **Check policies in the course outline**

Lecture 13 Outline

- **Transmission Control Protocol (TCP)**
  - **Introduction**
  - **Connection establishment (handshake)**
  - **Header and Version**
- User Datagram Protocol (UDP)
  - Implementation
  - Header
- Class exercise 13

Transport Layer (Function)



5

Transport Layer (4)

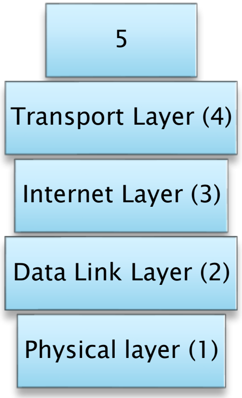
Internet Layer (3)

Data Link Layer (2)

Physical layer (1)

- Layer-4 of TCP/IP model
- Primary function is **host-to-host (or source to destination) data transfer** using segments or datagrams
- Provides:
  - Addressing is done using port numbers
  - TCP provides reliable data transfer with error detection (through checksum), correction (through retransmissions), and flow control

Transmission Control Protocol (TCP)

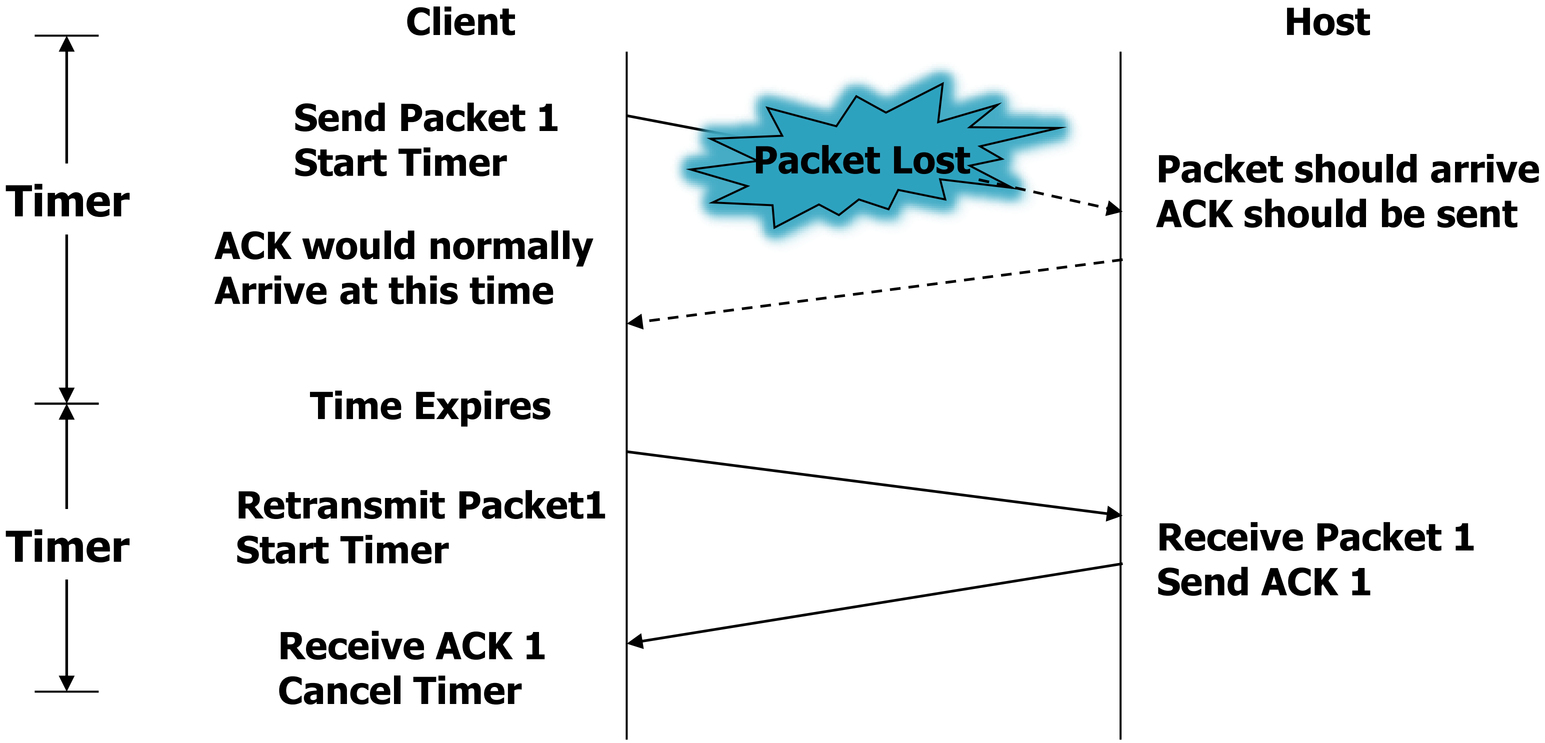


- Reliable and the backbone of most of today's protocols based on IP address, which make the communication possible
- Connections are:
  - Full-duplex
  - Stream-oriented (data can be delivered as stream of bytes)
  - Connection-oriented
- Protocols that use TCP:
  - HTTP, FTP, SMTP…

TCP operations

- Connections are established using a *three-way handshake* *(Client: Syn, Server: Syn/Ack, Client:Ack)*
- Data is divided into packets (payload) on OS/application level
- Packets are sequentially numbered, and received packets are acknowledged
- Connections are explicitly closed (or may abnormally terminate)

TCP: Data transfer with Timer



Client

Host

Timer

Send Packet 1
Start Timer

Packet Lost

Packet should arrive
ACK should be sent

ACK would normally
Arrive at this time

Time Expires

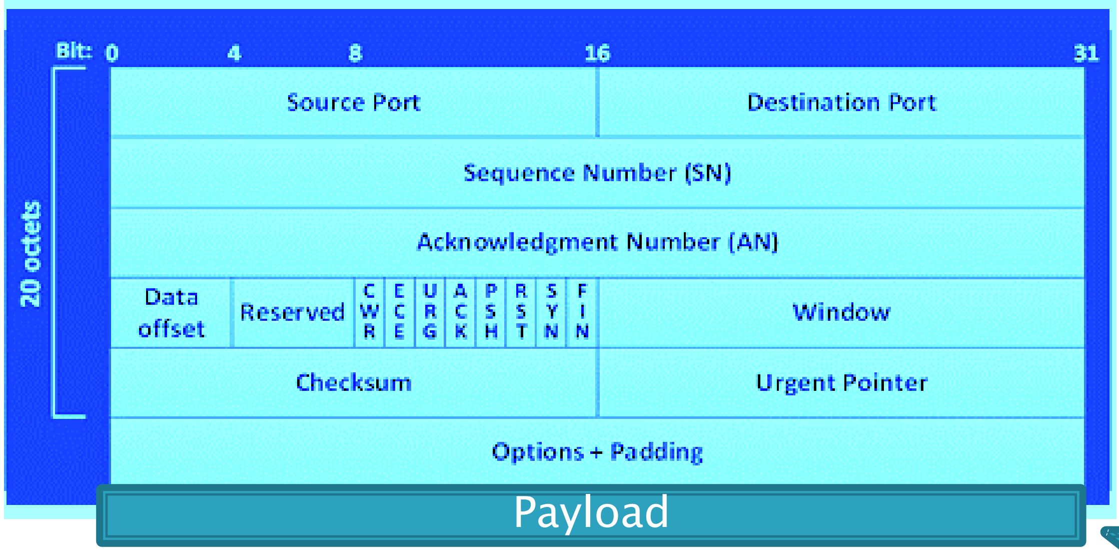
Timer

Retransmit Packet1
Start Timer

Receive Packet 1
Send ACK 1

Receive ACK 1
Cancel Timer

TCP Header (1/2)

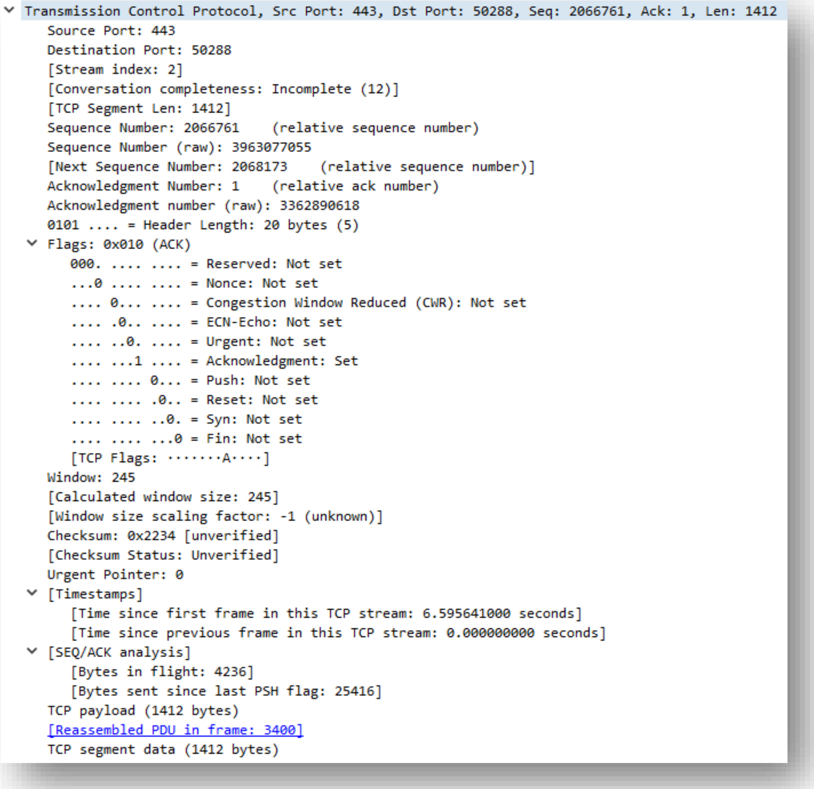


Max. size? (depends on the MTU length)

Field	**Purpose**
Source Port	Identifies originating application
Destination Port	Identifies destination application
Sequence Number	Sequence number of first octet in the segment
Acknowledgment #	Sequence number of the next expected octet (if ACK flag set)
Flags	TCP flags: [SYN, FIN, RST](connection man.), PSH (push now), ACK (valid), URG (urgent)
Window	Number of octets from ACK that sender will accept for flow control
Checksum	Checksum of IP pseudo-header + TCP header + data
Urgent Pointer	Pointer to end of “urgent data” (with URG flag), from the first byte, for priority transfer
Options	Special TCP options such as MSS and Window Scale / variable length
Payload	Transmitted data of variable length

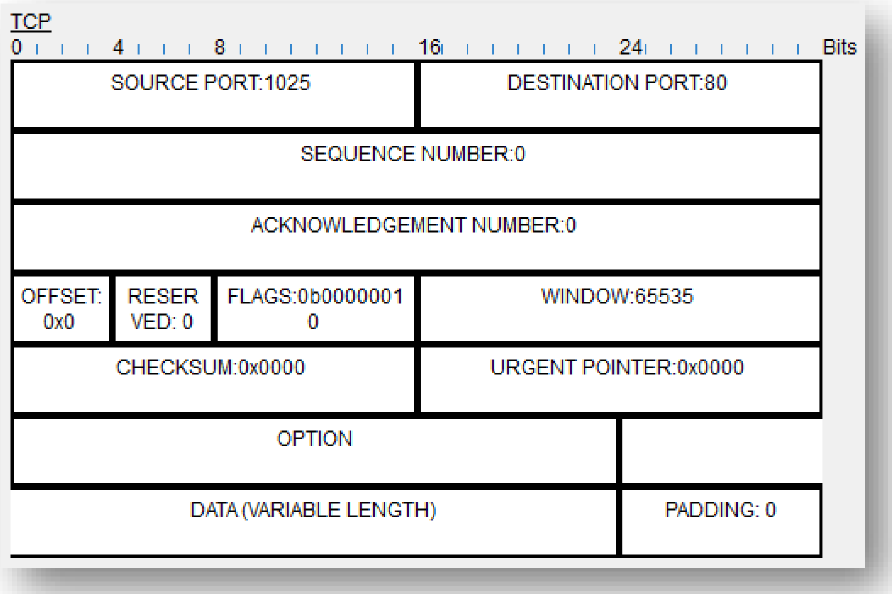
TCP Header (2/2)

Wireshark



```
Transmission Control Protocol, Src Port: 443, Dst Port: 50288, Seq: 2066761, Ack: 1, Len: 1412
   Source Port: 443
   Destination Port: 50288
   [Stream index: 2]
   [Conversation completeness: Incomplete (12)]
   [TCP Segment Len: 1412]
   Sequence Number: 2066761    (relative sequence number)
   Sequence Number (raw): 3963077055
   [Next Sequence Number: 2068173    (relative sequence number)]
   Acknowledgment Number: 1    (relative ack number)
   Acknowledgment number (raw): 3362890618
   0101 .... = Header Length: 20 bytes (5)
   Flags: 0x010 (ACK)
      000. .... .... = Reserved: Not set
      ...0 .... .... = Nonce: Not set
      .... 0... .... = Congestion Window Reduced (CWR): Not set
      .... .0.. .... = ECN-Echo: Not set
      .... ..0. .... = Urgent: Not set
      .... ...1 .... = Acknowledgment: Set
      .... .... 0... = Push: Not set
      .... .... .0.. = Reset: Not set
      .... .... ..0. = Syn: Not set
      .... .... ...0 = Fin: Not set
      [TCP Flags: ·······A····]
   Window: 245
   [Calculated window size: 245]
   [Window size scaling factor: -1 (unknown)]
   Checksum: 0x2234 [unverified]
   [Checksum Status: Unverified]
   Urgent Pointer: 0
   [Timestamps]
      [Time since first frame in this TCP stream: 6.595641000 seconds]
      [Time since previous frame in this TCP stream: 0.000000000 seconds]
   [SEQ/ACK analysis]
      [Bytes in flight: 4236]
      [Bytes sent since last PSH flag: 25416]
   TCP payload (1412 bytes)
   [Reassembled PDU in frame: 3400]
   TCP segment data (1412 bytes)
```

Cisco Packet tracer



SOURCE PORT:1025 | DESTINATION PORT:80

SEQUENCE NUMBER:0

ACKNOWLEDGEMENT NUMBER:0

OFFSET: 0x0 | RESERVED: 0 | FLAGS:0b00000010 | WINDOW:65535

CHECKSUM:0x0000 | URGENT POINTER:0x0000

OPTION

DATA (VARIABLE LENGTH) | PADDING: 0

TCP/IP versions

- TCP/IP Versions
  - Version 4 (curent)
  - Version 5
  - Version 6 (future)
- IPv6 has been slow to arrive
- IPv6 requires new software; IT staffs must be retrained; …
  - Per some experts, IPv6 will most likely coexist with IPv4 for years to come

Lecture 13 Outline

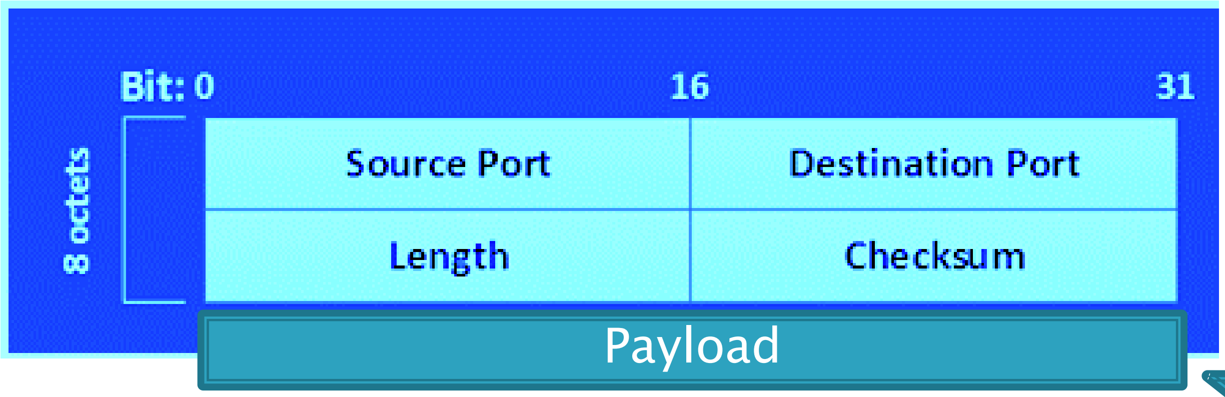
- Transmission Control Protocol (TCP)
  - Introduction
  - Connection establishment (handshake)
  - Header and Version
- **User Datagram Protocol (UDP)**
  - **Implementation**
  - **Header**
- Class exercise 13

User Datagram Protocol (UDP)

- UDP is connectionless, and does not establish any end-to-end connection manager to check on the received packets
  - Packet loss is better handled by application than the network stack (due to no need of unwanted connection setup overhead)
- It provides port information for application connection
- Protocols that use UDP:
  - DNS, SNMP, …

UDP Header (1/2)

- It provides source and destination port numbers for application connection
  - Length: Length field contains the length of the entire UDP segment (UDP header & data)



Bit: 0 16 31

8 octets

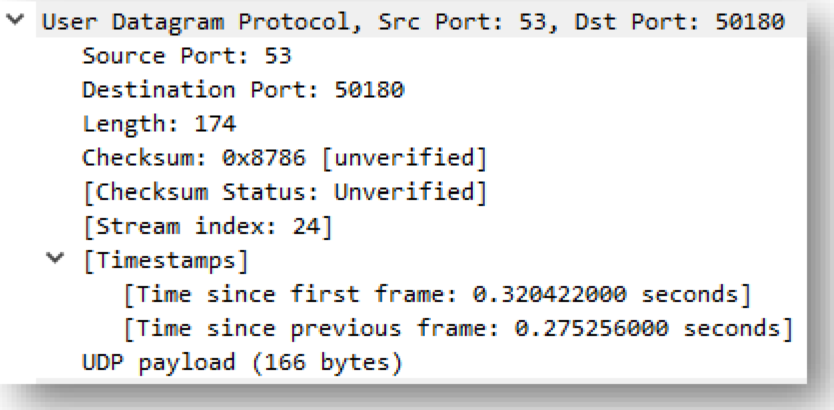
Source Port	Destination Port
Length	Checksum

Payload

Max. size? (depends on the MTU length)

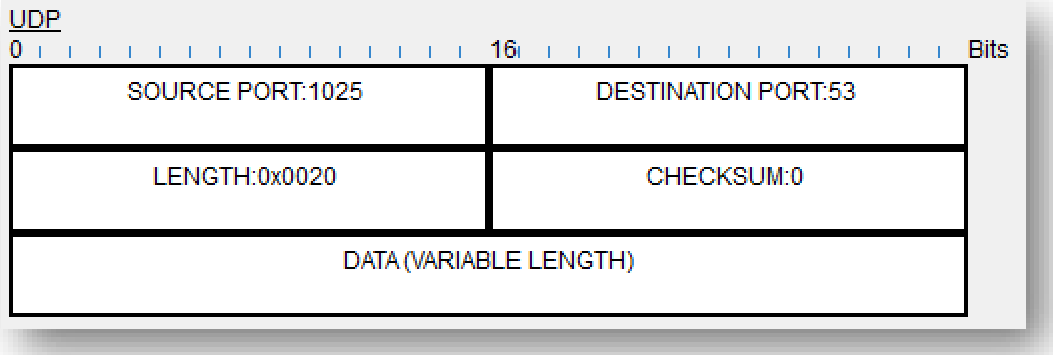
UDP Header (2/2)

Wireshark



```
▾ User Datagram Protocol, Src Port: 53, Dst Port: 50180
    Source Port: 53
    Destination Port: 50180
    Length: 174
    Checksum: 0x8786 [unverified]
    [Checksum Status: Unverified]
    [Stream index: 24]
  ▾ [Timestamps]
      [Time since first frame: 0.320422000 seconds]
      [Time since previous frame: 0.275256000 seconds]
    UDP payload (166 bytes)
```

Cisco Packet tracer



SOURCE PORT:1025	DESTINATION PORT:53
LENGTH:0x0020	CHECKSUM:0
DATA (VARIABLE LENGTH)	

UDP Header (cont.)

Checksum detects bit errors in the UDP header

- It uses the same error checking algorithm as used in TCP and IPv4 headers *i.e. One's complement*

  E.g. Bit stream: 11100110011001101101010101010101

  1.Divide bit stream into two parts of 16-bit each:

  1110011001100110 and 1101010101010101

  2.Add these two bit streams:

```
   1110011001100110
   1101010101010101 +
  ___________________
  11011101110111011 (wrap around)
   1011101110111100 (sum)
```

  3.Apply one's complement (inverse) to the result:

  0100010001000011 (checksum)

  4.Receiver will add the **sum** and **checksum** value to get all 1's

If an error is detected, the packet is discarded, and no error recovery action is taken

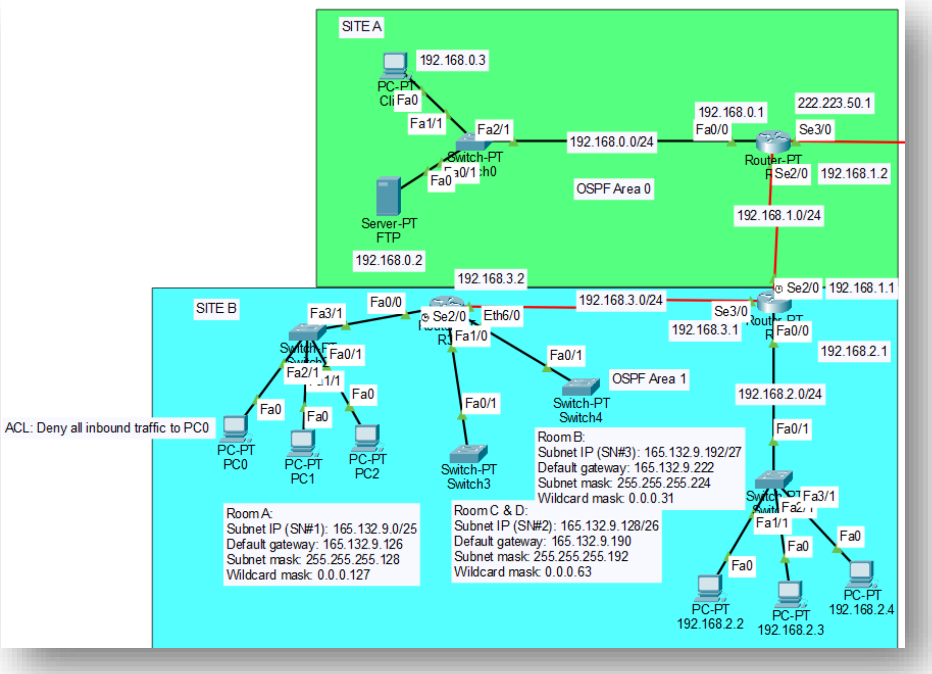
Use of the checksum field in UDP is optional (e.g., all bits can be set to zero or off when transmitting data over LAN, since transmission errors can be obtained from Ethernet protocol checksum)

Lecture 13 Outline

- Transmission Control Protocol (TCP)
  - Introduction
  - Connection establishment (handshake)
  - Header and Version
- User Datagram Protocol (UDP)
  - Implementation
  - Header
- **Class exercise 13**

Exercise 13: Practical work

- Use your existing cisco packet tracer file (from last class exercise):
  1. Generate following packets:
     1. TCP
     2. UDP
  2. View the network packet in simulation mode, take screenshots and place them in a document file including your observations [e.g., What do you see in a specific header field? Does the field value corresponds to a specific octet? If yes, which one is it (e.g., last, first?)]



ACL: Deny all inbound traffic to PC0

Room A:
Subnet IP (SN#1): 165.132.9.0/25
Default gateway: 165.132.9.126
Subnet mask: 255.255.255.128
Wildcard mask: 0.0.0.127

Room B:
Subnet IP (SN#3): 165.132.9.192/27
Default gateway: 165.132.9.222
Subnet mask: 255.255.255.224
Wildcard mask: 0.0.0.31

Room C & D:
Subnet IP (SN#2): 165.132.9.128/26
Default gateway: 165.132.9.190
Subnet mask: 255.255.255.192
Wildcard mask: 0.0.0.63

Alternatively, you can use Wireshark

Verify using PING and TRACERT/TRACEROUTE

Lecture 8 ends here

- Course Slides: Go to MS Teams:
  ‘Introduction to Computer Networks - Spring 2024 | BSc’
  -> Files section

- Send your questions by email:
  mohammad-salman.nadeem@epita.fr
  OR via direct message using MS Teams

- Thank You!