# The Transport Layer

Lecture given by Emmanuel Lochin

ISAE-SUPAERO

Original slides from A. Carzaniga (Univ. Lugano)
Extended/modified by E. Lochin (ISAE-SUPAERO) with author permission

Textbook Chap. #3 Sections 3.1 to 3.3

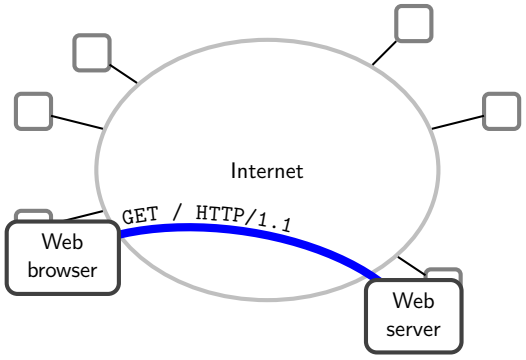


## Outline

- Basic concepts in transport-layer protocols
- Multiplexing/demultiplexing
- UDP message format
- Reliable transfer



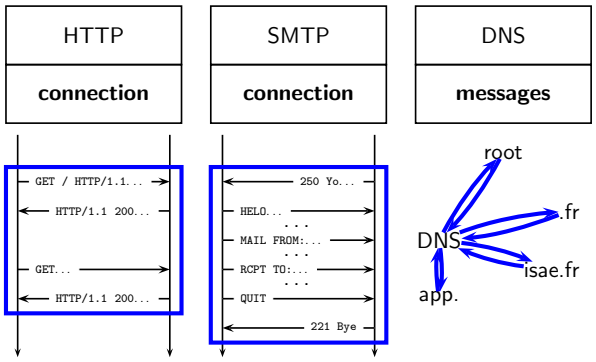
## Transport Layer




Primitive communication between applications



## Type of Service






## Transport Layer in the Internet

- Transport Control Protocol (TCP)
  - conntection-oriented (i.e., "connections")
- User Datagram Protocol (UDP)
  - connectionless (i.e., "messages")
- Terminology
  - transport-layer packets are called segments
- Basic assumptions on the underlying network layer
  - every host has one unique **IP address**
  - best-effort delivery service
    - ⋆ no guarantees on the integrity of segments
    - ⋆ no guarantees on the order in which segments are delivered

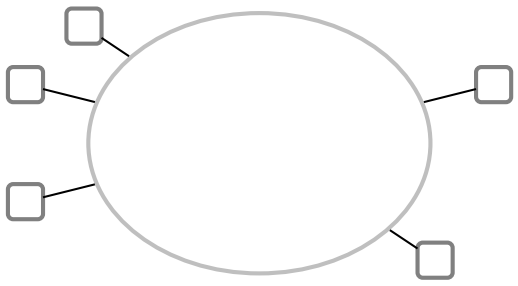


## Transport-Layer Value-Added Service

- Transport-layer multiplexing/demultiplexing
  - i.e., connecting applications as opposed to hosts
- Reliable data transfer
  - i.e., integrity and possibly ordered delivery
- Connections
  - i.e., streams
  - can be seen as the same as ordered delivery
- Congestion control
  - i.e., end-to-end traffic (admission) control so as to avoid destructive congestions within the network

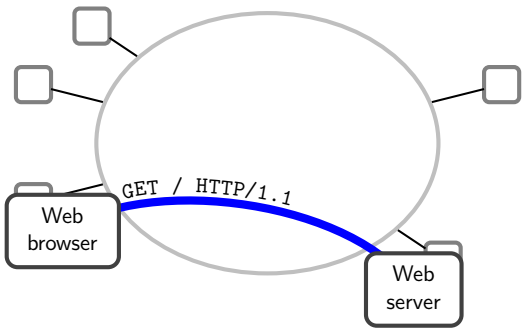


## Multiplexing/Demultiplexing

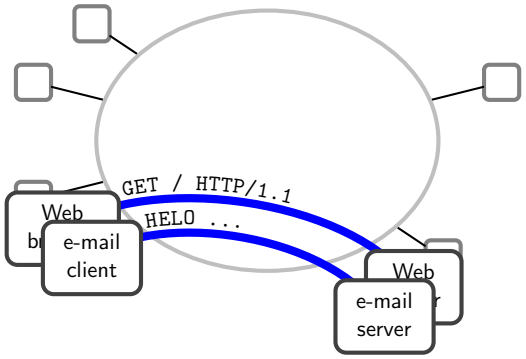




## Multiplexing/Demultiplexing



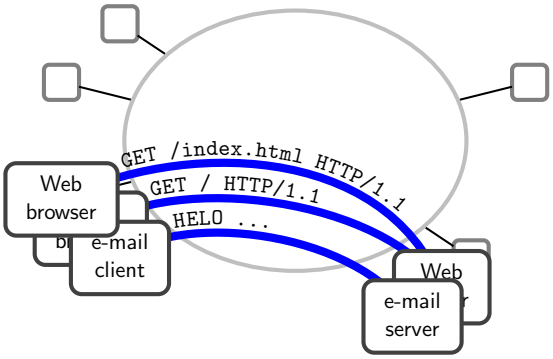
## Multiplexing/Demultiplexing




How do we distinguish all these "connections" ?



## Multiplexing/Demultiplexing




How do we distinguish all these "connections" ?

Lecture given by Emmanuel Lochin The Transport Layer ISAE-SUPAERO 7 / 12

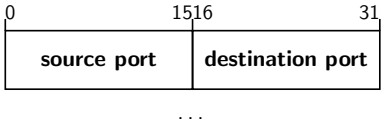
## Ports

- Each application running on a host is identified (within that host) by a unique port number
  - port numbers are simply cross-platform process identifiers
- How do we identify a "connection" ?
  - two pairs of **host** and **application** identifiers
  - i.e., two pairs **(IP-address, port)**
- How do we find out which application (host and port number) to connect to ?
  - outside the scope of the definition of the transport layer
  - but of course we can have "well-known" service numbers



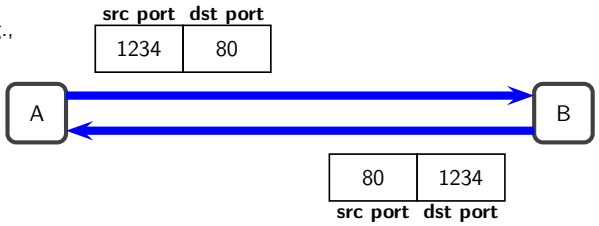
## Ports

- The message format of both UDP and TCP starts with the source and destination port numbers




...

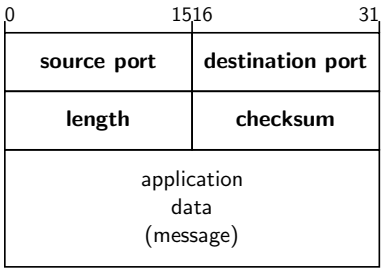
- E.g.,




Lecture given by Emmanuel Lochin The Transport Layer ISAE-SUPAERO 9 / 12

## UDP Packet Format

- The UDP message format is very simple




Lecture given by Emmanuel Lochin The Transport Layer ISAE-SUPAERO 10 / 12

## UDP Features

- UDP provides only the two most basic functionalities of a transport protocol
  - application identification (multiplexing/demultiplexing)
  - integrity check by means of a CRC-type checksum
- What if there is no application at the other end ?
- Checksum ensures that data is free of errors (within the checksum algorithm's limits)
  - UDP and TCP use 1's complement for their checksums (same for IP)
- What should happen when the checksum doesn't check ?

Lecture given by Emmanuel Lochin The Transport Layer ISAE-SUPAERO 11 / 12

## Checksum computation example

- Assume an IP header starting by `0x4500003044224000800600008C7C19ACAE241E2B`
- Compute the one's complement sum `4500 + 0030 + 4422 + 4000 + 8006 + 0000 + 8C7C + 19AC + AE24 + 1E2B = 2BBCF`
- Next fold the result into 16 bits by adding the carry to the result `2 + BBCF = BBD1`
- Final step is to compute the one's complement : `ones(BBD1) = 442E`
- Validation is done using the same algorithm : `2BBCF + 442E = 2FFFD` then `2 + FFFD = FFFF` and one's complement of `FFFF` is zero