Universiteit
Leiden

# EssCS - Topic 4

# Introduction to Computer Networking

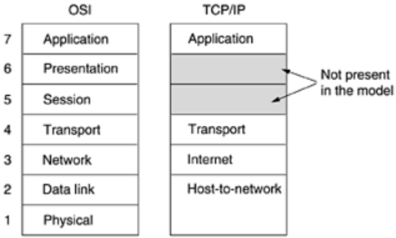
Lecture 14, 3.12.2024
Nuša Zidarič

# The Reference Models

- reference models: each layer has different functionality (with protocols and interfaces to layers above and below)
- what was just discussed is a small part of physical layer
- OSI = Open Systems Interconnection

| | | |
|---|---|---|
| 7 | Application | https, email, ... |
| 6 | Presentation | syntax and semantics of transmitted data |
| 5 | Session | establish/terminate communication sessions between host processes |
| 4 | Transport | ensures reliable transmission source → destination |
| 3 | Network | addressing, routing |
| 2 | Data Link | ensuring error-free transfer of data between the nodes |
| 1 | Physical | transmitting raw bits over a communication channel |

- TCP/IP (Transmission Control Protocol/Internet Protocol)

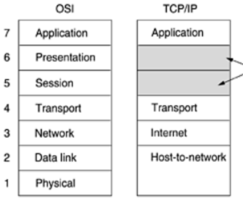


| | OSI | TCP/IP |
|---|---|---|
| 7 | Application | Application |
| 6 | Presentation | |
| 5 | Session | |
| 4 | Transport | Transport |
| 3 | Network | Internet |
| 2 | Data link | Host-to-network |
| 1 | Physical | |

Not present in the model

- we will roughly follow a hybrid in a bottom-up fashion

# The Reference Models



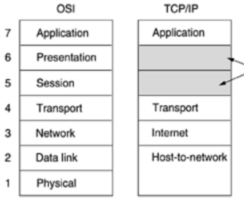
| | OSI | TCP/IP |
|---|---|---|
| 7 | Application | Application |
| 6 | Presentation | |
| 5 | Session | |
| 4 | Transport | Transport |
| 3 | Network | Internet |
| 2 | Data link | Host-to-network |
| 1 | Physical | |

- Recap: 3 main concepts are **interface**, **protocol** and **service**
- Recap: we covered (a part of) physical layer in last lecture and during tutorial (NRZ, bipolar RZ, Manchester encoding)
- Recap: we touched on Data Link Layer during tutorial with Hamming Code

- Data Link Layer:
  - provides a service to the Network layer
  - regulates data flow (flow control)
  - deals with transmission errors

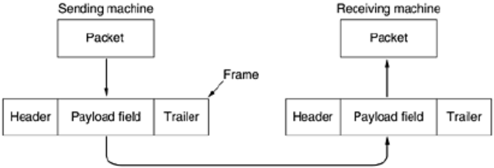
- Physical layer accepts raw bit-stream from the Data Link Layer and attempts to deliver it to the destination
- Data Link Layer uses this service from Physical layer

# The Reference Models



| | OSI | TCP/IP |
|---|---|---|
| 7 | Application | Application |
| 6 | Presentation | |
| 5 | Session | |
| 4 | Transport | Transport |
| 3 | Network | Internet |
| 2 | Data link | Host-to-network |
| 1 | Physical | |

- Data Link Layer breaks the bit-stream into discrete frames (and adds a checksum to each frame) - encapsulation
- Recap: Message format: payload with a header and trailer

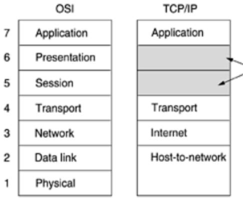


header can include: destination, source, packet length, sequence number, . . .

- Example of forming a frame: flagbyte 01111110
  to uniquely recognize the flagbyte the sender must add[1] a 0 after any sequence with five 1's: 11111

[1] receiver will remove the 0

## The Reference Models



- Data Link Layer services:
- unacknowledged connectionless service:

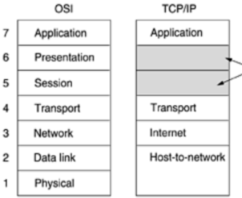
  no logical connection is established/released
- acknowledged connectionless service:

  no logical connection is established/released, however, each frame is acknowledged[2]
- acknowledged connection-oriented service:

  first, a logical connection is established, then each frame is sent/received in order and only once, and is acknowledged, finally, the logical connection is released
- to reduce the traffic: piggy-back the acknowledgement onto some outgoing frame
- metric: round-trip time (RTT): the time between sending a packet and receiving a response (usually in milli-seconds)
- flow control techniques: speed matching of transmitter and receiver with the intention to ensure that the receiver does not overflow: packets can be dropped from queues or even at the receiver

[2] confirmation: if no ACK, then retransmit-on-timeout

# The Reference Models



- Data Link Layer flow control example: Sliding Window Protocol

- naive method stop-and-wait: the transmitter waits for the ACK before sending the next packet
- optimization: transmitter and receiver maintain sliding windows[3]

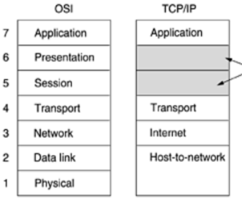
  the *sequence numbers* of the frames that have been sent, but not yet acknowledged, are kept in the *sending window*

  the *sequence numbers* of the frames that are expected, are kept in the *receiving window*

  note: the two sides do not need to have same-sized windows and the lower and upper bounds of the seq.no. can differ!

[3] transmitter also has buffers to store the frames that have not yet been acknowledged

# The Reference Models



- Data Link Layer flow control example: Sliding Window Protocol

- naive method stop-and-wait: the transmitter waits for the ACK before sending the next packet
- optimization: transmitter and receiver maintain sliding windows[4]

  the *sequence numbers* of the frames that have been sent, but not yet acknowledged, are kept in the *sending window*

  the *sequence numbers* of the frames that are expected, are kept in the *receiving window*

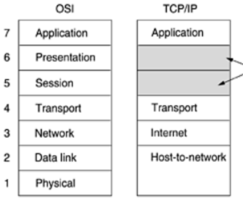
  note: the two sides do not need to have same-sized windows and the lower and upper bounds of the seq.no. can differ!

  ⇒ this allows the transmitter to continue sending Data[N+1] without ACK[N]
- if timeout: go-back-N option (retransmit multiple frames)
- if timeout: selective repeat (retransmit only lost or received with error)

- PAR (Positive Acknowledgment with Retransmission)
- ARQ (Automatic Repeat reQuest)

[4] transmitter also has buffers to store the frames that have not yet been acknowledged

# The Reference Models



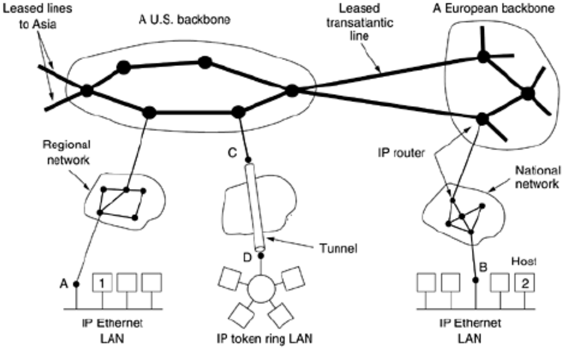
| | OSI | TCP/IP |
|---|---|---|
| 7 | Application | Application |
| 6 | Presentation | |
| 5 | Session | |
| 4 | Transport | Transport |
| 3 | Network | Internet |
| 2 | Data link | Host-to-network |
| 1 | Physical | |

- Data Link Layer: moving frames from transmitter to receiver
- Network Layer: moving packets from source to destination - with many hops inbetween (end-to-end transmission)
- topology awareness (nodes keep routing tables)
- routing algorithms
    - static methods:
    - (selective) flooding
    - shortest[5] path routing (Dijkstra)
    - adaptive methods:
    - distance vector routing (routing table with additional information, such as # of queued packets along the path, neighbors exchange routing tables)
    - link state routing (nodes exchange “link state packets”, each node builds a graph, then using algorithms such as shortest path)
- congestion control (monitor and adjust, e.g., packet discard policy)
- quality of service (resource reservation, expedited forwarding)

[5] can be based on distance, # hops, price, . . .

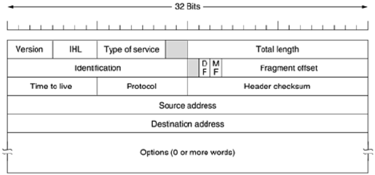
# Network Layer and IP

- Internet Protocol (Ipv4, Ipv6)
- universal routing and addressing protocol
- best-effort transport of data from source to destination
- multiple redundant links
- scalability
- fragmentation

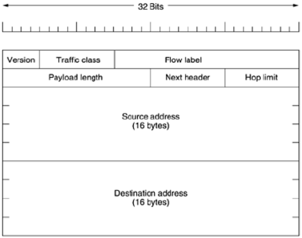


# Network Layer and IP

- Ipv4 header[6]



- IPv6 header



[6] figures: Tanenbaum: Computer Networks