Universiteit Leiden

EssCS - Topic 4

Introduction to Computer Networking

Lecture 15, 10.12.2024

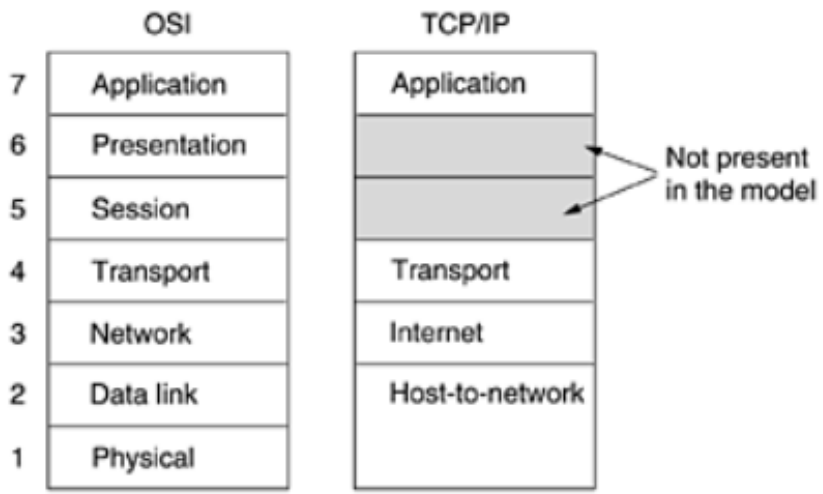
Nuša Zidarič

The Reference Models

- Recap: 3 main concepts are **interface**, **protocol** and **service**
- reference models: each layer has different functionality (with protocols and interfaces to layers above and below)
- OSI = Open Systems Interconnection

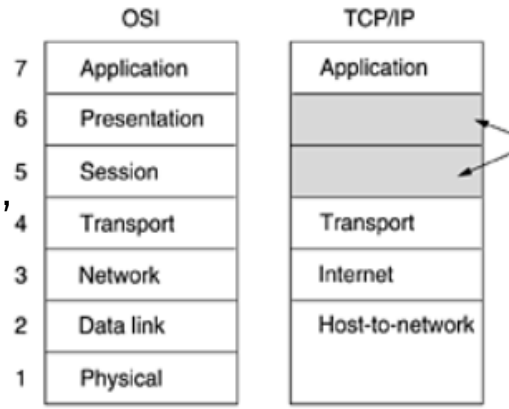
7	Application	https, email, ...
6	Presentation	syntax and semantics of transmitted data
5	Session	establish/terminate communication sessions between host processes
4	Transport	ensures reliable transmission source $\rightarrow$ destination
3	Network	addressing, routing
2	Data Link	ensuring error-free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

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



- bottom-up: layer N provides a service to layer N+1

Recap



- Message format: payload with a header and trailer
  typically header includes control information, such as destination,
  and trailer additional information, e.g., parity of the payload

header	payload (data)	trailer

  header and trailer are message overhead

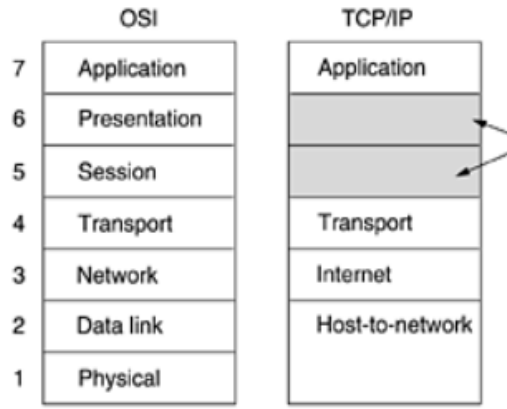
- Physical Layer:

- Physical layer accepts raw bit-stream from the Data Link Layer
  and attempts to deliver it to the destination
- Network media: twisted pair, coax, fiber optics, wireless communication
  all media suffer from attenuation and noise
- analog-to-digital conversion:

  analog signal $\rightarrow$ sampler $\rightarrow$ quantizer $\rightarrow$ encoder$\rightarrow$ binary signal (bit-stream)

- tutorial: NRZ, bipolar RZ, Manchester encoding

Recap

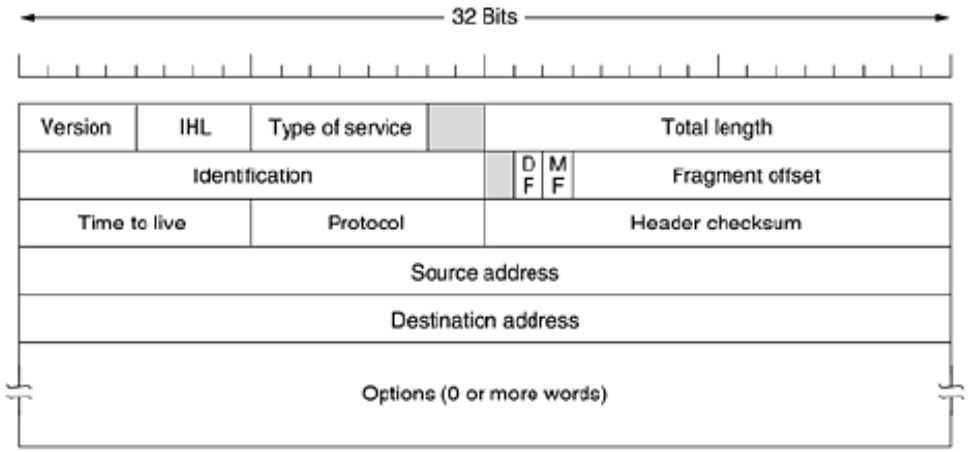


- Data Link Layer:
  - provides a service to the Network layer
  - encapsulation (H, payload, T), framing method example
  - keywords: transmitter (Tx), receiver (Rx), link (link between two nodes !)
  - flow control: sliding window protocol example (sequence numbers, ACK)
  - transmission errors (e.g., Hamming code)
- Data Link Layer is moving frames from transmitter to receiver!

- Network Layer: moving packets from source to destination - many hops inbetween
  - topology awareness (nodes keep routing tables)
  - routing algorithms:
    static methods (e.g., flooding), adaptive methods (e.g., link state routing)
  - Internet Protocol (IP):
    best-effort transport of data, redundancy, scalability, fragmentation

Network Layer and IP - IPv4 header

- Recap: lab 6 last week

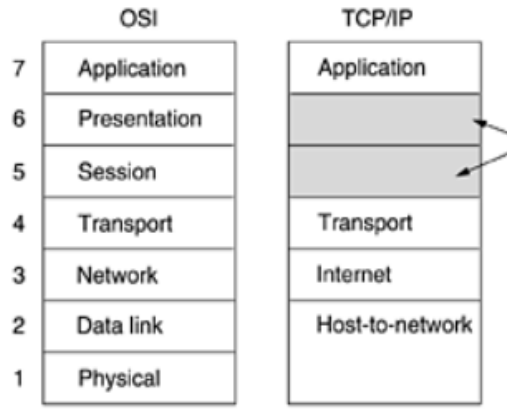


- Total Length (16 bits): maximum size 2^{16} bytes including the header
- Fragmentation - Identification field (16 bits): same for all fragments
- Fragment Offset - used to reassemble the original payload (MTU is Maximum Transfer Unit)

$$\text{A} \xrightarrow[\text{MTU}=1500]{} \text{B} \xrightarrow[\text{MTU}=400]{} \text{C} \xrightarrow[\text{MTU}=1500]{} \text{D}$$

- TTL (Time-To-Live): decrements by 1 at each node (8 bits)
  when 0 discard packet $\rightarrow$ maximum number of hops allowed on the path
- Protocol: among common examples are TCP and UDP

The Reference Models



- Network Layer:

  best-effort transport of data: no ACK (but it usually arrives)

  redundancy: packets can travel different routes

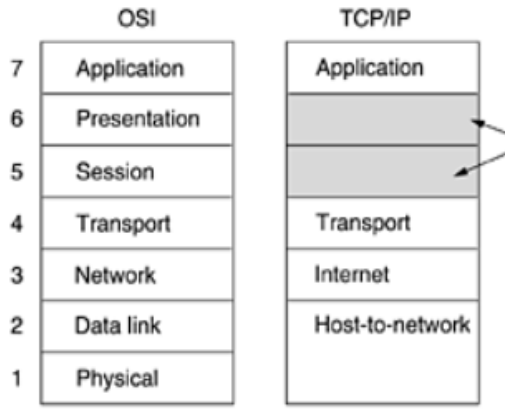
  ⇒ the layer above creates, maintains, and releases the connection, and ensures the packets are correctly reassembled[1] at the destination

- Transport Layer:
  - accepts the data from above, splits it up into smaller units (if needed), passes them down to the network layer, and ensures that all units arrive correctly to the other end
  - IP protocol only supports sending to a specific *host*. An application running on the host will listen to specified *ports*; together they are called *sockets*: (host,port) pairs
  - TCP = Transmission Control Protocol
  - UDP = User Datagram Protocol

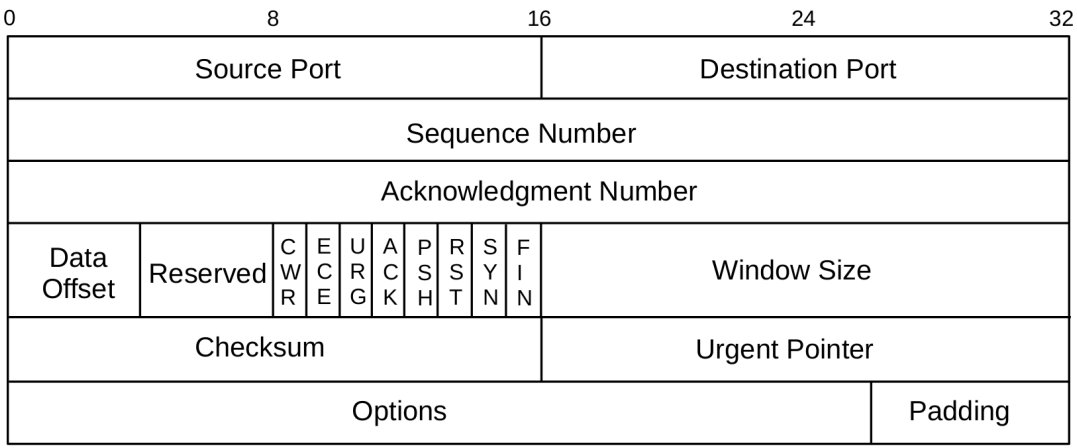
- Note: TCP/IP model is named after two protocols on the transport and on the network layer!

[1] we are **not** referring to fragmentation here

Transport Layer and TCP



- TCP = Transmission Control Protocol
  - simple connection-oriented transport service provides basic primitives connect, send, receive, listen, and disconnect
  - reliable connection:
    - packets are numbered and lost ones retransmitted
    - packets that arrive out-of-order are buffered and passed on to the upper layer (to the application) in the correct order
    - all packets are acknowledged, and the sender keeps a timer for ACK and will retransmit on timeout
  - sliding window protocols: to maximize throughput and avoid congestion
- TCP Header

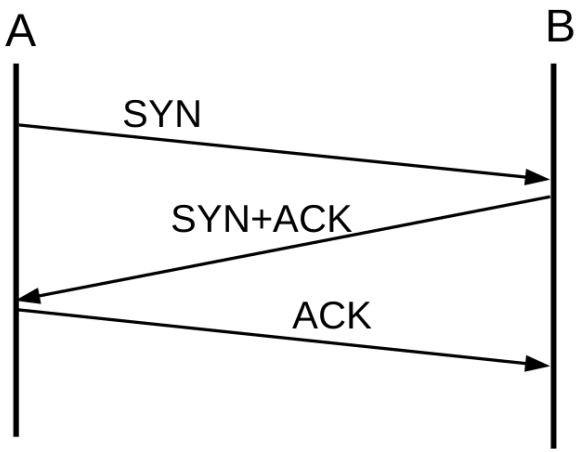


control flags SYN, ACK, FIN are used to establish and release the connection

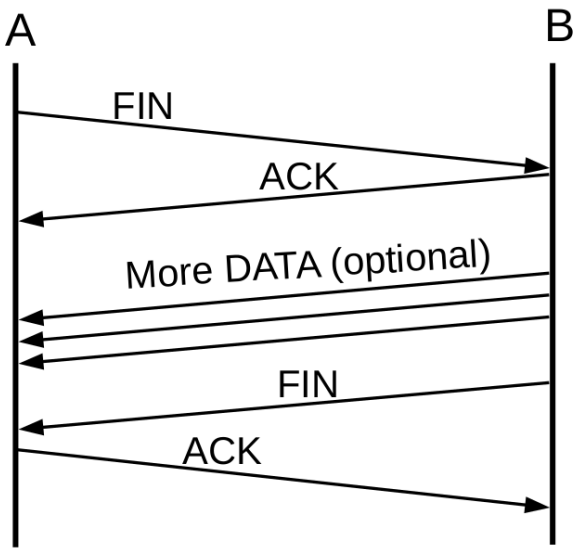
Transport Layer and TCP

- TCP three-way handshake to establish connection
- releasing the connection (B still has data to send)

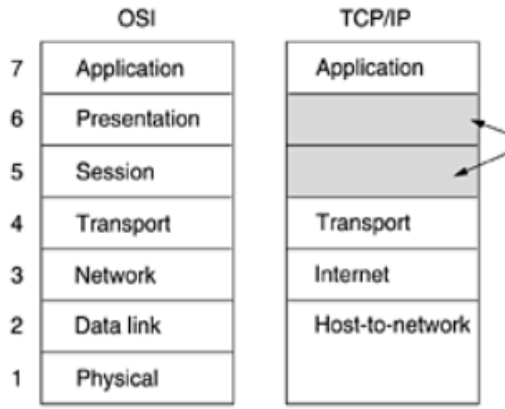
Establishing connection



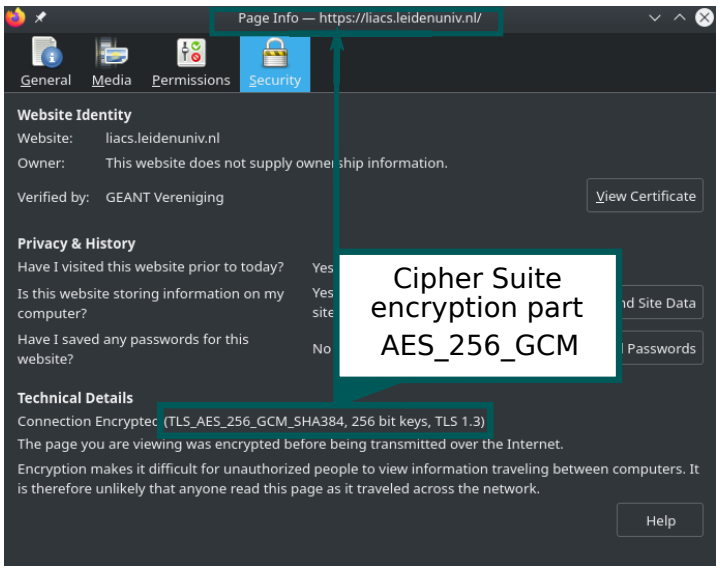
Releasing connection



The Reference Models



- Application Layer:
  - DNS = Domain Name System:
    resolves the (user-friendly) domain name into the corresponding IP address
    `liacs.leidenuniv.nl` → `132.229.137.15`
  - FTP = File Transfer Protocol
  - SMTP = Simple Mail Transfer Protocol
  - HTTP = Hyper Text Transfer Protocol
  - HTTPS = Hypertext Transfer Protocol Secure
    `https://liacs.leidenuniv.nl/`



Secure Communications

We wish to provide:

- **confidentiality:** keeping information secret from all but those who are authorized
- **data integrity:** ensuring the information has not been altered
- **authentication:** corroborating the source of information
- other objectives: identification, non-repudiation, ...

- encryption/decryption $\Rightarrow$ confidentiality
- message authentication codes, digital signatures $\Rightarrow$ integrity, authentication

- Secure communication protocols
  - handshake (negotiation and key exchange) $\rightarrow$ public key cryptography
  - encrypted and authenticated payload $\rightarrow$ symmetric key cryptography