NSWI090: **Computer Networks**

http://www.ksi.mff.cuni.cz/~svoboda/courses/202-NSWI090/

Lecture 3

Network Models and Architectures

Martin Svoboda
svoboda@ksi.mff.cuni.cz

15. 3. 2021

Charles University, Faculty of Mathematics and Physics

Lecture Outline

Layered network models

- Basic principles
  - Layers
  - Horizontal / vertical communication
  - Protocols
- Models and architectures
  - **OSI** model
  - **TCP/IP** architecture
- **Layers** and tasks they are supposed to solve

Layered Models

Motivation and objectives

- (Computer) **networks are very complex**
  - It make sense to decompose the problem into smaller parts
    - Similarly as in case of implementation of large software systems
- We will work with **hierarchy of layers**
  - I.e., instead of assuming arbitrary modules, our layers are strictly vertically ordered
- Each layer focuses on particular tasks
  - Different layers at **different levels of abstraction**
    - Sending of individual bits at the lowest physical layer…
    - … usage of complex services at the highest application layer
  - Lower layer offers services to the higher one
  - Higher layer uses services of the lower one

Layered Models

Basic principles of layers

- Only **public interface** is defined
- Internal details are intentionally hidden
  - ⇒ layers can be **independent on each other**
    - As for the implementation, though exceptions exist
  - ⇒ **alternative approaches** can be deployed
  - ⇒ flexibility is increased
    - Since each layer can be treated and solved separately
    - And new implementations can appear seamlessly

Questions

- How many layers do we actually need?
- What tasks should they perform?
- What interface should they provide?

Horizontal Communication

Horizontal communication = within a layer / across nodes

- I.e., communication of the **corresponding entities** at **the same layer** across **different nodes** or active network elements
  - E.g.: network interfaces at L2, nodes at L3, processes at L7, …

Observations

- Can never happen across layers
  - Appropriate counter-party entity is always at the same layer
- **Multiple transmissions** can occur at the same time
- Involved entities must follow **rules** defined by **protocols**

Horizontal Communication

Observations (cont'd)

- **Asynchronous** character
  - Individual bits / blocks of data are send by the sender
  - We then need to wait for the response
- Usually **virtual** character only
  - Just `L1` physical layer actually transfers something!
  - All **higher layers** provide only **illusion** (though very good) of such **direct horizontal communication**
    - In reality, **vertical communication** takes place…

Vertical Communication

Vertical **communication** = within a node / across layers

- I.e., communication of **different layers** within **the same node** or active network element

Principle

- **Sender** perspective
  - Data to be sent is prepared and **passed to the lower layer**
- **Recipient** perspective
  - Received data is unpacked and **passed to the higher layer**

Observations

- Individual layers cannot be skipped
  - Only **directly adjacent layers** can communicate with each other

Communication Protocols

Communication protocol

- Specification according to which two or more entities communicate with each other
  - Must be given in advance, implementation independent, …
- At least the following must be defined…
  - **Public interface**
    - For the purpose of the **vertical** communication
  - **Communication rules**
    - For the purpose of the **horizontal** communication
    - Define **permitted / expected actions** of the individual entities in situations that can occur
    - Technically using **state diagrams** or verbal descriptions
  - **Data format**
    - **Internal structure and semantics** of the individual components

Communication Protocols

Observations

- Always **within a single layer**
  - Never span two or more layers at a time
- **Multiple protocols** often exist within a given layer
  - Can be mutually **alternative**…
    - I.e., perform the same tasks differently
    - E.g.: TCP and UDP at `L4`
  - Can be mutually **complementary**…
    - I.e., perform different tasks
    - E.g.: SMTP and HTTP at `L7`
- They can even be **used concurrently** within a given layer
  - We must be able to distinguish between them

Communication Protocols

Protocol Data Unit (PDU)

- Unit of data transmitted among the peer entities
  - **Different names** (frames, cells, packets, …)
- Internal structure
  - **Header**: sender / recipient address, …
  - **Body**: useful data (**payload**) provided by the higher layer
  - Sometimes also **footer**
- **MTU** (**Maximum Transmission Unit**)
  - Maximal permitted payload size
- Questions and tasks
  - Necessary **metadata pieces** in header / footer
  - Minimal / maximal block sizes
  - Coexistence of multiple protocols side by side

Models and Architectures

Network model

- Conceptual model describing how a network should operate
  - Number of **layers**
  - **Tasks** to be solved
    - And assignment of these tasks to the individual layers
  - **Services** to be provided
    - Connection-oriented / connectionless, reliable / unreliable, …

Network architecture

- Particular implementable and implemented network model
  - Everything above + the following…
  - Definition of **protocols**

Models and Architectures

ISO **OSI** **model** (Open Systems Interconnection Model)

- Also denoted as Reference Model
- Originated in the **world of communications**
  - Preference of connection-oriented and reliable transmissions with QoS support
- **7 layers**

TCP/IP **architecture** (Internet Protocol Suite / Stack)

- Originated in the **world of computers**
  - Preference of connectionless and unreliable transmissions over the Best Effort principle
- **4 layers**

…

ISO OSI Model

Lower layers

- L1: **Physical Layer**
  - Sending of individual bits through a physical medium
- L2: **Data Link Layer**
  - Delivery of blocks of data within a local network
- L3: **Network Layer**
  - Routing and forwarding of packets across a system of networks

Adaptation layer

- L4: **Transport Layer**
  - End-to-end communication of individual entities within nodes

ISO OSI Model

Higher layers

- L5: **Session Layer**
  - Management of sessions and organization of data exchange
- L6: **Presentation Layer**
  - Automatic conversions and serialization of structured data
- L7: **Application Layer**
  - Sending of messages and usage of user-oriented services

TCP/IP Architecture

Network layers

- **Network Interface Layer** (Link Layer)
  - L1+L2
- **Network Layer** (Internet Layer)
  - L3
- **Transport Layer**
  - L4
- **Application Layer**
  - L7 and selected aspects of L5 and L6

Physical Layer

`L1`: **Physical layer**

- Main task: **transmission** of individual **bits** through a given physical **medium**
  - **Does not understand the content being transmitted**
    - Treats all bits equally, cannot distinguish between them

Transmission **media**

- Available options (guided / unguided)
  - **Metallic**: twisted pairs, coaxial cables
  - **Optical**: optical fibers
  - **Wireless**
- Real-world paths are not optimal
  - Attenuation, distortion, interference, …
  - $\Rightarrow$ **transmission potential is always limited**

Physical Layer

Signal transmission

- Certain **analog quantity** is transmitted in all the cases
  - Metallic: electrical signal
  - Optical: light
  - Wireless: radio electromagnetic waves
- **Interpretation** can be **analog** / **digital**

Other aspects to be solved

- Coding, modulation, timing, synchronization, bandwidth, …

Data Link Layer

L2: **Data Link layer**

- Main task: sending of **blocks** of data between network **interfaces** of particular **nodes** within a **local network**
  - Illusion of a **direct path** between the sender and recipient
    - I.e., all the nodes are mutually visible and reachable
  - Reality can be different, though
    - I.e., even a local network can have a complex internal structure
    - However, sender does not need to be aware of it
  - Everything is / can be and typically will be sent to everyone

Data Link Layer

Internetworking

- Active network elements
  - Bridges, **switches**
  - Controllers in bigger networks may also be needed
- Internal mechanisms
  - Store&Forward, Cut-Through

Logical topologies

- Describe the internal **logical structure of a network**
  - Defines how data flows within a network
  - May differ from the physical topology at `L1`
- Approaches
  - Bus, star, ring, mesh, hypercube, …

Data Link Layer

Addresses and addressing

- **Physical address** (MAC / HW / hardware address)
  - Allow for the **identification of the intended recipient**
    - So that the recipient can be found and data delivered
    - So that the recipient can actually recognize its data
  - Must be **unique within a given network**
- Questions
  - Internal structure
  - Assignment mechanisms
- Example: **Ethernet**, **Wi-Fi**, **Bluetooth**, …
  - **EUI-48** (originally MAC-48) or newer EUI-64
  - E.g.: `FC:77:74:19:41:1E`

Data Link Layer

Filtering and forwarding

- Mechanisms allowing to find and reach the intended recipient
  - Otherwise everything would need to be sent in all directions
- Implemented in bridges and switches

Ensuring **transparency**

- **Control signals** (metadata) need to be separated from the useful **payload**
- Techniques
  - Escaping, framing, stuffing

Data Link Layer

Enabling **block** transmissions (**framing**)

- **Sender** perspective
  - Constructed PDU (e.g., Ethernet frame) is simply passed to `L1`
  - **MTU** depends on the particular technology
- **Recipient** perspective
  - **Stream of bits** (or other symbols) is received at `L1`
  - Frames need to be correctly **recognized and interpreted**
    - **Start** of a block
    - **end** / **length** of a block

Cooperation of `L1` and `L2`

- Extra bits (bytes, …) may intentionally be added to help with **synchronization** or other aspects solved at `L1`

Data Link Layer

Shared medium **access methods**

- **Shared medium**
  - Multiple nodes share the same transmission path
  - **Only one participant can transmit at a moment**
- Access control methods
  - Determine particular rules
    - Based on a competition, …

Data link layer decomposition

- Shared media were not originally assumed by ISO OSI
- Solution
  - Lower **MAC sublayer** (Media Access Control)
  - Higher **LLC sublayer** (Logical Link Control)

Network Layer

L3: **Network layer**

- Main task: **hop-to-hop routing and forwarding** of packets across a **system of interconnected networks** to the target **node** of the **final** intended recipient
  - We are aware of the **existence of multiple networks** as well as the way they are **mutually interconnected**
    - Or at least to a certain extent
    - Even the sender itself must think about the first steps of routing
  - Packets are delivered through individual **routers**, one by one

Internetworking

- Active network elements
  - **Routers**, …

Network Layer

Addresses and addressing

- Requirement
  - Each node must have a **globally unique address**
  - All nodes within a network must share the same prefix
    - Since routing can only work at a level of whole networks
- Questions and tasks
  - Internal structure
    - Allowing to easily resolve membership of a node in a network
  - **Assignment of blocks of addresses** to networks as a whole
  - **Assignment of individual addresses** to nodes inside a network
- Example: **IPv4** addresses
  - E.g.: 213.46.172.38

Network Layer

Lack of IPv4 addresses

- Techniques
  - Subnetting, supernetting, CIDR, private addresses and NAT, IPv6 addresses

Sending of packets

- Direct **delivery**
  - IP address of the intended recipient belongs to our network
  - $\Rightarrow$ packet is sent locally via L2 directly to the **target node**
- Indirect **delivery**
  - Otherwise…
  - $\Rightarrow$ packet is first sent locally via L2 to our **router** which then takes care of further routing and forwarding of this packet

Network Layer

Local L2 **delivery**

- Particular node within our network must be reached
  - Final recipient in case of the direct L3 delivery
  - Router otherwise
- **Encapsulation**
  - IP packet is inserted as a payload into a constructed L2 frame
- This frame is then sent… but to whom?
  - **IP** $\rightarrow$ **HW** **address resolution** is required

Network Layer

Routing

- Process of **selecting an optimal transmission path**
  - Path = **sequence of routers** allowing to reach the final recipient
  - Combinatorial problem of **searching for the shortest paths**
    - In a (weighted) multi-graph
  - At least a certain knowledge of L3 **topology** is necessary
    - Expressed using **routing tables**
- Strategies
  - Dynamic / static
  - Isolated / centralized / distributed
  - …
- Particular protocols (RIP, OSPF, …), policies, …

Network Layer

Forwarding

- Process of **sending of packets** based on the already resolved routing paths
  - Using **forwarding tables**
- May / may not be executed by the same device (router) as in case of routing itself

Routing domains

- It is not possible to maintain routing tables in **large systems**
  - Not just due to their size…
- $\Rightarrow$ they need to be **decomposed into smaller parts**
  - E.g.: **Autonomous Systems** (AS) in Internet

Network Layer

Fragmentation of blocks

- Each L2 technology has its own **MTU**
  - Size of IP packets can be higher than these MTUs
- $\Rightarrow$ **fragmentation is needed**
  - Who shall be responsible
  - How **de/fragmentation** should technically be performed

Transport Layer

L4: **Transport layer**

- Main task: **end-to-end** communication of particular **entities** within the **sender / recipient nodes**
  - Lower layers (L1 – L3) always treat nodes at atomic units
    - I.e., they are unable to distinguish the individual communicating entities inside these nodes
  - L4 and higher layers are **only implemented in end nodes**
    - I.e., the highest layer implemented in routers is L3
    - And so L4 does not occur in typical network elements at all

Transport Layer

Addresses and addressing

- Individual entities must be **mutually distinguishable**
- Requirements
  - **Unique** within a given node
  - **Static** = fixed and known in advance
    - So that we are able to determine the address of the recipient
  - **Abstract** = independent on a particular platform
  - **Implicit** = independent on the current situation
- Example
  - **Port numbers** in TCP/IP
    - 25 (SMTP), 80 (HTTP), …
- Questions and tasks
  - Rules of usage for both **outgoing and incoming directions**

Transport Layer

De/multiplexing

- **Several communications** can take place **concurrently**
  - However, we have **only one transmission path** at L3
- **Multiplexing**
  - Merging of several separate transmissions by the sender
- **Demultiplexing**
  - Reverse decomposition by the recipient

Sockets

- **Data structure** allowing applications to send / receive data
  - Created on demand
  - Dynamically bound with particular ports

Transport Layer

Adaptation layer

- L4 offers various ways of **adapting** the **expectations of higher layers** to the **possibilities of the lower layers**
  - Lower layers: L1 – L3
    - Focus on transmissions themselves
    - E.g.: **IP**: blocks, connectionless, unreliable, Best Effort
  - Higher layers: L5 – L7
    - Focus on applications needs
- In particular…
  - **Streams** over blocks
  - **Connection-oriented** transmissions over connectionless
  - **Reliable** transmissions over unreliable
  - **Quality of Service** over the Best Effort principle

Transport Layer

Other optional services

- **Flow control**
  - Preventing **slower recipients** to be overwhelmed by **faster senders**
- **Congestion control**
  - Preventing the **whole network** to be overwhelmed by the overall traffic generated by senders

Session Layer

L5: **Session layer**

- Main task: provide mechanisms for opening, closing and managing **sessions** and organizing **dialog** (data exchange) between the communicating entities
  - Originally very broad functionality
    - Meaningful concepts and ideas
    - But not needed by everyone
  - Usually completely omitted nowadays
    - Implemented within L7 only when really needed

Session Layer

Session management

- One L5 session over multiple L4 transport connections
  - One session over **multiple concurrent connections**
    - Achieving higher overall transmission capacity (bonding)
  - One session over **more consecutive connections**
    - Ensuring session continuity after a transport connection failure
- More L5 sessions over one L4 transport connection
  - **More consecutive sessions** over one connection
    - Minimizing the number of established transport connections
  - **Multiple concurrent sessions** over one connection
    - Multiplexing several separate sessions at the same time

Session Layer

Synchronization on entities

- Illusion of **synchronous communication** over asynchronous L4
  - Similarly as understood by RPC (Remote Procedure Call)
- Simplex / half-duplex / **full-duplex** communication
- **Deadlock** prevention

Transaction support (atomicity and consistency)

- **Checkpointing** and **recovery**
  - Restoration points are created on demand / on a regular basis
  - Allows for recovery in case of failures
    - Only data after the last checkpoint needs to be resent
- **Two-phase commit** protocol

Session Layer

Identification

- Including **counter-party localization**
  - Similarly as in SIP (Session Initiation Protocol) in VoIP

Security

- **Authentication**
  - Verification of user identity
- **Authorization**
  - Mechanism of determining access levels or privileges
- **Encryption**
  - Ensuring confidentiality of transmitted data

Presentation Layer

L6: **Presentation layer**

- Main task: provide mechanisms for automatic **serialization** of data enabling its transmission and **conversions** ensuring that its **semantics** is preserved on different platforms
  - I.e., it is not entirely true that data should always be received completely unchanged
    - Simply because **sender / recipient** nodes can run on different **hardware platforms** and **operating systems**, as well as can assume different **localizations**
  - **Translation between application and network data formats** is thus needed
- Reality
  - Usually completely omitted once again
    - Implemented within L7 only when really needed

Presentation Layer

Conversion of atomic values

- Different **text encodings**
  - ASCII, ISO 8859-2, Windows-1250, UTF-8, UTF-16, …
- Different **byte order conventions**
  - **Big Endian**
    - The most significant byte is provided as the first one
  - **Little Endian**
    - The least significant byte is provided as the first one
- Different **number formats**
  - Various mantissa and exponent sizes for floating point numbers

Presentation Layer

Serialization of simple structures

- E.g.: arrays, records, sets, …
- Transformation is (relatively) straightforward

Serialization of complex structures

- E.g.: high-dimensional matrices, objects with pointers, …
- Issues
  - Higher **dimensions**
    - Transmission path is always **one-dimensional**
    - I.e., only flat sequences of bytes can be transmitted
  - **Pointers**
    - Sender / recipient nodes have their own address spaces

Presentation Layer

Serialization strategies

- Proprietary
  - Specific approach proposed by a given protocol / application
- Generic
  - **Abstract syntax**
    - Structure of data is first described using a suitable language
    - E.g.: ASN.1 (X.680 Abstract Syntax Notation One)
  - **Transfer syntax**
    - Data is then serialized into a particular serialization format
    - E.g.: BER (X.690 Basic Encoding Rules)
- Real-world examples
  - Contemporary solutions used by NoSQL databases
    - Apache Thrift, Protocol Buffers, …

Application Layer

L7: **Application layer**

- Main task: provide access to the communication interface and so allow applications to send and receive **messages** via which they can provide or use **services**
  - Original idea
    - L7 should contain entire applications
    - That would require their full standardization
  - Reality
    - Contains only communication essentials,
      not user interface, business logic, or other parts of applications

Application Layer

Addresses and addressing

- **Identification** of communication partners an objects
  - **IRI** (Internationalized Resource Identifier)
    - E.g.: URL: https://www.mff.cuni.cz/
- **Localization** of such partners
  - **DNS** (Domain Name System)
    - **Hierarchical system for domain names**
    - Translation of these names to IP addresses

Particular **communication protocols**

- SMTP (Simple Mail Transfer Protocol)
- HTTP (Hypertext Transfer Protocol)
- …

Lecture Conclusion

Layered network models

- ISO OSI model
- TCP/IP architecture

Layers

- L7 **application** layer
- L6 **presentation** layer
- L5 **session** layer
- L4 **transport** layer
- L3 **network** layer
- L2 **data link** layer
- L1 **physical** layer