

**NSWI090: Computer Networks**

<http://www.ksi.mff.cuni.cz/~svoboda/courses/222-NSWI090/>

Lecture 1

# Paradigms

**Martin Svoboda**

[martin.svoboda@matfyz.cuni.cz](mailto:martin.svoboda@matfyz.cuni.cz)

2023

**Charles University**, Faculty of Mathematics and Physics

# Lecture Outline

## Introduction and motivation

- Basic concepts
- **Network paradigms**
- Telecommunication / computer networks

# Motivation

## Ultimate objective

- **Transmission of data**
  - I.e., **not processing**
    - Algorithms, data structures, time complexities, ...
  - As well as **not storing**
    - Database systems, logical models, query languages, ...

## Major requirement

- **Transmitted data should not change**
  - However...

## Relevant questions

- Distance, volume, velocity, latency, reliability, ...

# Basic Concepts

## Transmission **path**

- Path between the sender and recipient in a network
- Includes **any means of transmission**
  - Regardless of technical implementation
    - Wired (metallic, optical), wireless
  - At different levels of abstraction
    - Physical, virtual

## Transmission **channel**

- One-way transmission path in telecommunication networks

## Transmission **circuit**

- Two-way transmission path
  - Physical, emulated / virtual

# Basic Concepts

## Simplex mode

- Allows for transmissions in **one direction only**
  - E.g., television and radio broadcasting

## Half-duplex mode

- Allows for transmissions in **both directions**, but **not simultaneously**
  - Communicating entities need to take turns

## Full-duplex mode

- Allows for transmissions in **both directions** at the same time

# Basic Concepts

## Unicast transmissions

- One sender, one intended recipient
  - The most common situation

## Anycast transmissions

- Any node in a given predefined group

## Multicast transmissions

- Every node in a static / dynamically created group

## Broadcast transmissions

- Any reachable node (in a given network, ...)

# Basic Concepts

## Telecommunication / computer **network** / infrastructure

- System of nodes mutually interconnected by transmission paths formed by network elements

## Nodes

- Anything wanting to communicate
  - Servers, terminals, personal computers, laptops, ...
  - Printers, mobile phones, smart TVs, ...

## Network elements

- Active
  - Repeaters, switches, routers, gates, ...
  - Switchboards, controllers, transceivers, ...
- Passive

# Basic Concepts

## Networks provide **services** to their users

- Television and radio broadcasting
- Voice and video calls
- Video on demand
- Electronic mail
- Messaging platforms
- File transfers
- Web pages browsing
- ...



# Basic Concepts

## Services (cont'd)

- Provided at different levels of abstraction
  - Internal implementation details are concealed
  - Services can be built on top of other services
    - **HTTP** web page request at L7 → reliable connection-oriented  
**TCP** at L4 → best effort **IP** at L3 → **Ethernet** at L2
- **Different requirements and expectations**
  - Different principles, protocols, and technologies are needed

# Network Paradigms

Fundamental questions to be figured out

- **Stream vs. block transmissions**
  - *In what form will the data be transmitted?*
- **Circuit vs. packet switching**
  - *How will the data reach the intended recipient?*
- **Connection-oriented vs. connectionless transmissions**
  - *Do we need to agree on the communication in advance?*
- **Reliable vs. unreliable transmissions**
  - *What level of transmission reliability is required?*
- **Guaranteed vs. non-guaranteed services**
  - *Will sufficient resources be available all the time?*
- ...

# Data Transmission

*In what form will the data be transmitted?*

## **Stream transmissions**

- Data is sent as a continuous stream without any logical division

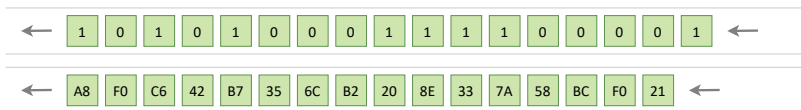
## **Block transmissions**

- Data is divided into reasonably large blocks, these blocks are then sent one by one

# Stream Transmissions

## Streams

- Data is transferred as a continuous unstructured **stream**
  - At the level of individual **symbols** (bits, bytes, words, ...)

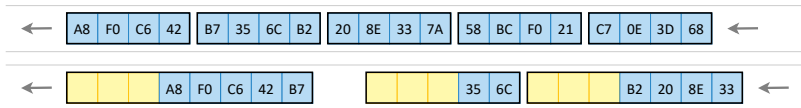


- Features
  - **Preserves order** (works as a FIFO queue)
  - Suitable for point-to-point connections
    - I.e., in case there is only one intended recipient
- Examples
  - L1: **Ethernet, Wi-Fi, ...**
  - L4: **TCP** (virtual streams for L7 over L3 blocks)

# Block Transmissions

## Blocks

- Data is divided into reasonably large units called **blocks**
  - **Fixed size**
  - **Variable size**
    - Restrictions on maximal / minimal lengths may be in effect, too



- Features
  - Blocks may not directly follow each other
    - Nothing is remaining to be sent at a moment
    - Intentional delays between the blocks may be needed
  - **Order may not be preserved**

# Block Transmissions

May (but also may not) be enriched with additional **metadata**

- **Sender / recipient addresses**
  - So that blocks can be correctly forwarded when a given transmission path is not direct
- **Transmission identification**
  - So that multiple different transmissions can be mutually distinguished on a shared path
- **Block ordinal numbers**
  - So that the original data can once again be reconstructed
- ...

# Block Transmissions

## Main tasks

- Internal structure
  - **Header, body, footer**
- Combinations of both the approaches
  - **Blocks over streams** (framing)
  - **Blocks over blocks** (encapsulation)
  - **Streams over blocks**

## Names of blocks

- Depend on particular layers / technologies / protocols
  - L2: Ethernet **frames**, ATM **cells**, ...
  - L3: IPv4 **datagrams**, IPv6 **packets**, ...
  - L4: TCP **segments**, UDP **datagrams**, ...
  - L7: HTTP **messages** (requests, responses), ...

# Connection Schemes

*Do we need to agree on the communication in advance?*

## Connection-oriented transmissions

- Both the communicating parties need to **establish, maintain, and eventually terminate a connection** with each other
- Order of data is / must be preserved

## Connectionless transmissions

- Communication is based on **sending separate messages**
- Order of data is not preserved



# Connection-Oriented Transmissions

## Connection-oriented approach

- **Connection is established**
  - This means that...
    - Both the parties really **exist**
    - They are able to locate each other
    - They **agree** with the communication
  - Communication **parameters can be negotiated**
  - Transmission path can be laid out, resources allocated, ...
- The actual data is then transferred
- Connection is **eventually terminated**
  - E.g., resources are returned, marked circuits are released, ...

# Connection-Oriented Transmissions

## Analogy

- Traditional **voice call** over a **telephone network**
  - Phone number is dialed → call is established → parties are talking to each other → call is hanged up and terminated

## Examples

- L2: ATM (Asynchronous Transfer Mode)
- L4: **TCP** (over a connectionless IP at L3)
  - Ordering of segments needs to be enforced
- L7: HTTP, SMTP, POP3, ...
  - I.e., anything over TCP from L4

# Connection-Oriented Transmissions

## Stateful operation

- Communicating parties **transition between the states**
  - E.g.: *closed, established, ...*
- Transitions must be **correct and coordinated**
  - Mutual misunderstandings must be avoided
  - Deadlocks must be avoided
- **Non-standard situations** must be detected and treated
  - Connection failures, ...

# Connectionless Transmissions

## Connectionless approach

- Separate messages called **datagrams are sent**
  - No connection is established
    - Recipient may not exist, may not want to communicate, ...
    - Nothing is established, no termination is required, ...
  - Stateless operation
  - **Datagrams are delivered independently on each other**
    - They may be routed differently
    - Their order cannot be guaranteed
  - Must contain full **recipient identification**

# Connectionless Transmissions

## Analogy

- Traditional **postal services**
  - Postcards, letters, parcels, ...

## Examples

- L4: **UDP**
- L3: **IP, ICMP**
- L2: **Ethernet**

# Switching Mechanisms

*How will the data reach the intended recipient?*

## Circuit switching

- Transmission circuit is created first, all data is then sent through this circuit
- Based on **reserved capacity**
  - Mainly in **telecommunication** networks

## Packet switching

- Each block of data is routed and forwarded separately
- Based on **shared capacity**
  - Mainly in **computer** networks
  - Very revolutionary idea in its time (in 1960s)
  - Aims at **robustness and resilience** against failures and outages

# Circuit Switching

## Basic mechanism

- Transmission **circuit is created on demand**
  - I.e., transmission path is **found** and **laid out**
    - Physically interconnected
    - Marked virtually at higher layers
- All **data is then sent directly** through this circuit

## Examples

- PSTN (traditional Public Switched **Telephone Network**)
  - Originally **manual** switching by operators → later on automated **electronic** switching → fully **digital** switching nowadays

# Circuit Switching

## Features and consequences

- Always **connection-oriented**
- Illusion of a **direct connection**
  - Though a circuit will most likely in reality be laid out across multiple network elements
- $\Rightarrow$  low and **constant latency**
- $\Rightarrow$  **preserves order**
- Effect of **reserved capacity**
  - No one else is entitled to use a given circuit
- $\Rightarrow$  enables **guaranteed** transmissions
- Supports both **streams and blocks**



# Packet Switching

## Switching node logic

- Each interface (path) has its own **inbound / outbound buffer**
- Whenever a new block is received
  - It is stored within a given inbound buffer
- **Waiting blocks are perpetually iterated over and processed**
  - Routing decision is made for a given block
  - It is then moved to the corresponding outbound buffer
  - Finally, it is sent as soon as it is possible

## Examples

- L2: Ethernet **switches**, ...
- L3: IP **routers**, ...

# Packet Switching

Only possible for **block transmissions**

⇒ the following pieces of metadata is necessary

- **Recipient** identification
  - So that the intended recipient can be located as well as the individual ongoing transmissions mutually distinguished
- **Sender** identification
  - So that error messages can be sent back in case of failures

# Packet Switching

Available **capacity is shared** and limited

- Transmission capacity of paths
- Computing (switching) capacity of switching nodes
  - Buffer sizes, processor throughput

⇒ significantly **higher, variable, and unpredictable latency**

- Which also **depends on the current load**
  - I.e., it can never be estimated in advance

⇒ **available capacity may be insufficient**

- Excessive blocks will then need to be discarded

# Packet Switching

**Packet switching** itself may be...

- Connection-oriented: **virtual circuits**
  - Transmission **path is laid out only virtually**
    - Suitable **Virtual Circuit Identifier (VCI)** is assigned
    - Individual network elements must note the actual path
  - Blocks are then forwarded based on these VCIs
    - Each is delivered via the same path
  - Example
    - L2: **ATM**
- Connectionless: **datagram service**
  - Example
    - L2: **Ethernet**

# Reliability Levels

*What level of transmission reliability is required?*

## Observations

- Transmissions are never ideal
  - It may always happen that the data will be damaged
    - Entire **blocks can be lost**
    - Actual **data can be corrupted**

## Reliable transmissions

- Sender considers their duty to take care of the remedy
- **Errors are detected and treated**

## Unreliable transmissions

- Errors are not detected, nor treated
- **Transmission simply goes on**

# Reliable Transmissions

## Detection mechanisms

- Parity, checksums, CRC, ...
  - Whatever particular approach is exploited, **it is impossible to detect all the possible errors and extents of damage**

## Remedy options

- Error correction codes
- **Repeated transmission**
  - **Positive acknowledgment** when received successfully
  - **Negative acknowledgment** otherwise
    - I.e., request for a retransmission

## Example

- L4: **TCP** (Go-Back-N or Selective Repeat ARQ)

# Reliable Transmissions

## Consequences

- Messages are slightly bigger
  - Because of the checksums
- **Higher number of messages**
  - Positive / negative acknowledgments and retransmissions
- Delivery **regularity is disrupted**
  - Significant delays occur
- **Higher usage of computing and transmission capacity**
  - Sender / recipient nodes
    - The entire mechanism needs to be deployed
  - Network: handling of extra messages
- **Reliability is never absolute**

# Unreliable Transmissions

In other words...

- Reliability is always **relative only**
  - While insufficient in one case, excessive in another
- It is always connected with a **non-zero overhead**

⇒ **unreliable transmissions** make sense as well

- Especially in case of **multimedia applications**
  - Audio / video, interactive / non-interactive
- **Regularity** of delivery is essential, **lower latency** may as well

Examples

- L4: **UDP**
- L3: **IP**
- L2: **Ethernet, ATM**



# Guarantee Options

*Will sufficient resources be available all the time?*

## Guaranteed transmission / service

- Provides such a guarantee for all the currently ongoing data transmissions
  - In terms of computing and transmission capacity
- Resources must be **reserved in advance**
  - During the connection process
- Realized via **circuit switching**

## Non-guaranteed transmission / service

- It may happen that sufficient resources will be missing
- Realized via **packet switching**

# Guaranteed Transmissions

## Exclusive capacity

- Solely for the purpose of the given communicating parties

## ⇒ disadvantages

- Must be high enough to cover the **maximum expected load**
- **Unused capacity** cannot be left to anyone else
  - It is forfeited uselessly
- The whole approach is **ineffective** and **expensive**
  - Everything must be dimensioned for the sum of maxima

# Non-Guaranteed Transmissions

## Shared capacity

- Cheaper and more efficient solution
  - Everything can be **dimensioned for the average load**

## ⇒ disadvantages

- When sufficient remaining resources are not available
  - Because of...
    - Completely filled buffers
    - Overloaded processor
    - Transmission capacity of individual paths
- Certain **packets will need to be discarded!**
  - Note that this is the only possible measure
- The question is which...

# Extent of Guarantees

## Best Effort principle

- *Maximum effort...*
  - All packets are delivered as long as it is possible
- *... but uncertain outcome*
  - Should packet loss be inevitable, **all data is treated equally**
    - I.e., there are no rules, no priorities, no criteria
- Examples
  - L3: **IP**
  - L2: **Ethernet**, ATM (but also various QoS alternatives)

## Quality of Service (QoS)

- Anything else compared to the best effort principle
  - The extent of particular **guarantees may vary**

# Quality of Service

## Relative QoS – principle of **prioritization**

- Better conditions are provided for certain kinds of data
  - Based on **different priorities**

## Absolute QoS – principle of **reservation**

- Guarantees the same conditions regardless of the current situation and load
- Resources must be **reserved in advance**
  - Reservation request must be **rejected when not attainable**
- Similar to circuit switching
  - And so has the same advantages, but disadvantages as well

# Different Worlds

## World of **telecommunication networks**

- Significantly older
  - Communication was considered a strategic interest
- **Smart network, dumb devices** paradigm

## World of **computer networks**

- **Dumb network, smart devices** paradigm

Both the worlds traditionally built their own **separate networks**

- They have always **differed greatly** in many aspects
- But **integration and convergence attempts** are intensifying

# Telecommunication Networks

## Smart network, dumb devices

- All intelligence is concentrated in the network
  - Network elements are **often single-purpose**
  - Easier and usually **central management**
  - **Expensive, cumbersome, inflexible**
- User devices can, therefore, be very simple and foolproof

## Preferred transmission characteristics

- **Circuit switching**
- **Connection-oriented**
- **Reliable**
- **Guaranteed (QoS)**

# Telecommunication Networks

## Additional observations

- Assumption of **insufficient resources**
  - There are not enough resources to satisfy everyone
  - Focusing on **exclusive resource allocation**
  - Afraid to sell unreliable services
- Considered a matter of **strategic importance**
  - High level of **regulation**
    - Directive decision-making by national governments
  - Gradually liberalized
    - Monopolies are becoming incumbents
- Network **owners and users are different** entities



# Computer Networks

## Dumb network, smart devices

- Intelligence is concentrated in user devices
- Network should be minimalist and as efficient as possible
  - The only goal is to **transfer data without understanding it**
  - Easily adaptable to changes in user behavior
  - **Cheaper, straightforward, flexible**

## Preferred transmission characteristics

- **Packet switching**
- **Connectionless**
- **Unreliable**
- **Non-guaranteed** (Best Effort)

# Computer Networks

## Additional observations

- **Availability of resources is not a major limiting factor**
  - I.e., resources are sufficient
  - Technical factors have higher importance than the commercial
- **Liberalized** from the very beginning
  - Bottom-up approach
  - **Standardization and coordination challenges**
    - In order to achieve compatibility and interoperability
- Network **owners and users are often** the same entities

# Management of Resources

## Moore's law

- Observation that the **number of transistors** in a dense integrated circuit **doubles about every 2 years**
  - Projection of a historical trend
  - In effect for more than 55 years...
  - Originally 1 year, later 1.5 years, now 2 years
- Implication
  - **Cost** of equivalent computing power **drops in half every 2 years**

## Gilder's law

- **Transmission capacity triples every 1 year**

## Disk law

- **Storage capacity doubles every 1 year**



# Lecture Conclusion

## Basic concepts

- **Stream / block** transmissions
- **Circuit / packet** switching
- **Connection-oriented / connectionless** transmissions
- **Reliable / unreliable** transmissions
- **Guaranteed / non-guaranteed** transmissions
  - **Best Effort / Quality of Service**