Communication in Electronic Systems

Lecture 7: Layered Protocols. Network Infrastructure, Topologies, and Architectures

Lecturer: Petar Popovski

TA: Junya Shiraishi, João H. Inacio de Souza

email: petarp@.es.aau.dk

AALBORG UNIVERSITY
DENMARK

Connectivity

Course Overview: Part 2. Communication and Networking

- MM5: Introduction to Communication Systems
- MM6: Simple Multiuser Systems
- **MM7: Layered System Design. Network Topology and Architecture**
- MM8: Networking and Transport Layers
- MM9: Introduction to Security
- Guest lecture
- MM10: Packets and Digital Modulation
- MM11: Communication waveforms
- MM12: Workshop on modulation and link operation

outline

- recap
- layering
- protocol stack
- network systems
- network topologies
- examples

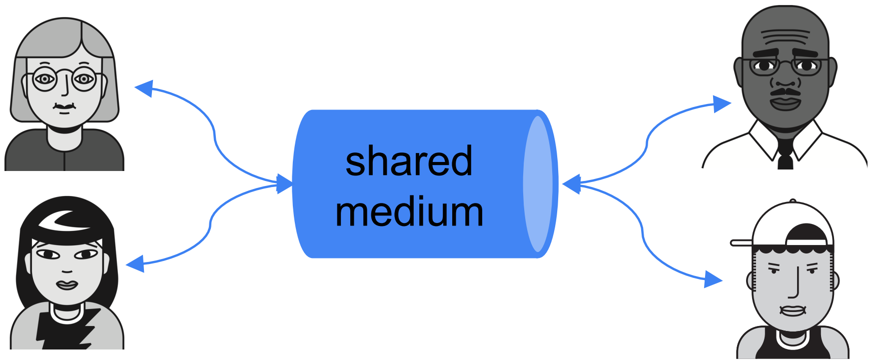
recap: lecture 1

- **lecture 1:** point-to-point communications

- what is a message
- what is a communication channel and how to model it
- some principles behind point-to-point protocols – frame design
  - how/when to transmit data
  - error-detection
  - feedback (ACK/NACK)

recap: lecture 2

- **lecture 2:** multiaccess communications



- how do we communicate when the communication channel is shared?
  - coordinated and uncoordinated methods
- coordinated:
  - scheduling, token-based
- uncoordinated:
  - random access methods
- we need to design systems that consider that the medium is being shared

recap: measuring data rate performance

- throughput

$$\frac{total\ number\ of\ correctly\ transmitted\ bits}{total\ time}$$

- goodput

$$\frac{total\ number\ of\ correctly\ transmitted\ USEFUL\ bits}{total\ time}$$

- useful bits from a perspective of a user

communication modules and the idea of layering

why layering?

- the task of a communication network is
  to provide data flows between source and destination points.

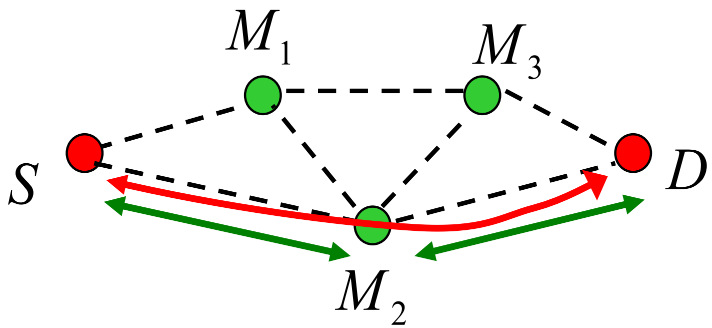
- each flow may have different requirements
  depending on the service implemented by that flow.

- two generic types of services:
  - guaranteed services (e. g. delay for real-time services)
  - best-effort services (e. g. e-mail or FTP)

- layering provides a structured or modular way to design network protocols

task decomposition

- task: provide a reliable data communication flow between S and D



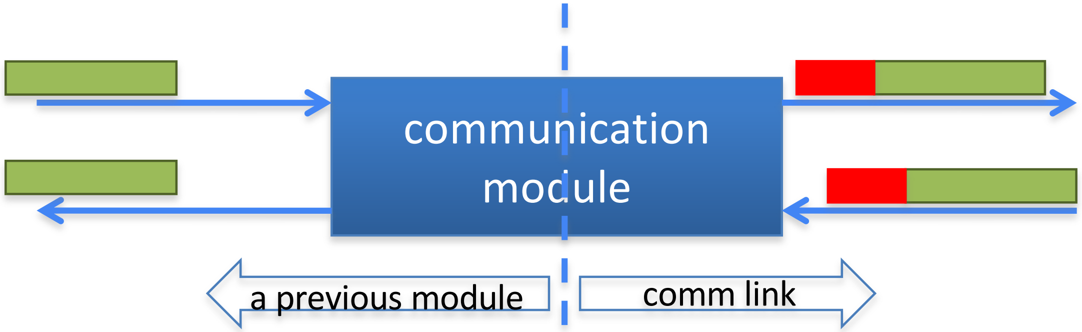
- this task is carried out as follows:
  - find a set of links that make a connection between S and D
  - make sure that the data is reliably received at the D, by sending feedback to S
  - each link should take care on its own for the reliable transfer of the data

communication modules and the idea of layering (1)

a communication system is comprised of many processes that occurs simultaneously
e.g., multiple access, error correction/detection

Q1. how can we design each one of the process in an independent manner?
Q2. how can we make this process standardizable?

- **layering:** see the communication system in an abstract, hierarchical perspective
- **modularity:** different functionalities of the entire communication system can be performed by several **modules**
- each module is seen as black box that support the overall function of the system by performing a service



communication modules and the idea of layering (2)

- note that frame is not visible to the module user
  - more precisely, the frame headers are not visible
- module functions we have not described
  - link initialization
  - timeout mechanisms (e. g. link breakage)
  - interface towards the module user
- the module can be treated as a black box
  - provides certain service to a user,
    but hides the unnecessary details
  - for example, the module user does not know
    how many times certain data packet has been sent until received successfully

communication modules and the idea of layering (3)

- how can Alice use the module to *e.g.* send a file?

1000 “pure data” bytes

- the file cannot be passed on to the module, as the module accepts max 64 bytes
- Alice adds a header to the 1000 bytes
    - it is assumed that a file can be max 65536 bytes long
    - Alice adds 2 bytes to tell the length of the pure data
    - Alice adds 16 CRC bits for error detection
        - this is to capture any error that may have slipped through the module

file length	file CRC	1000 “pure data” bytes

➡ but all this is 1004 bytes of Data for the module!

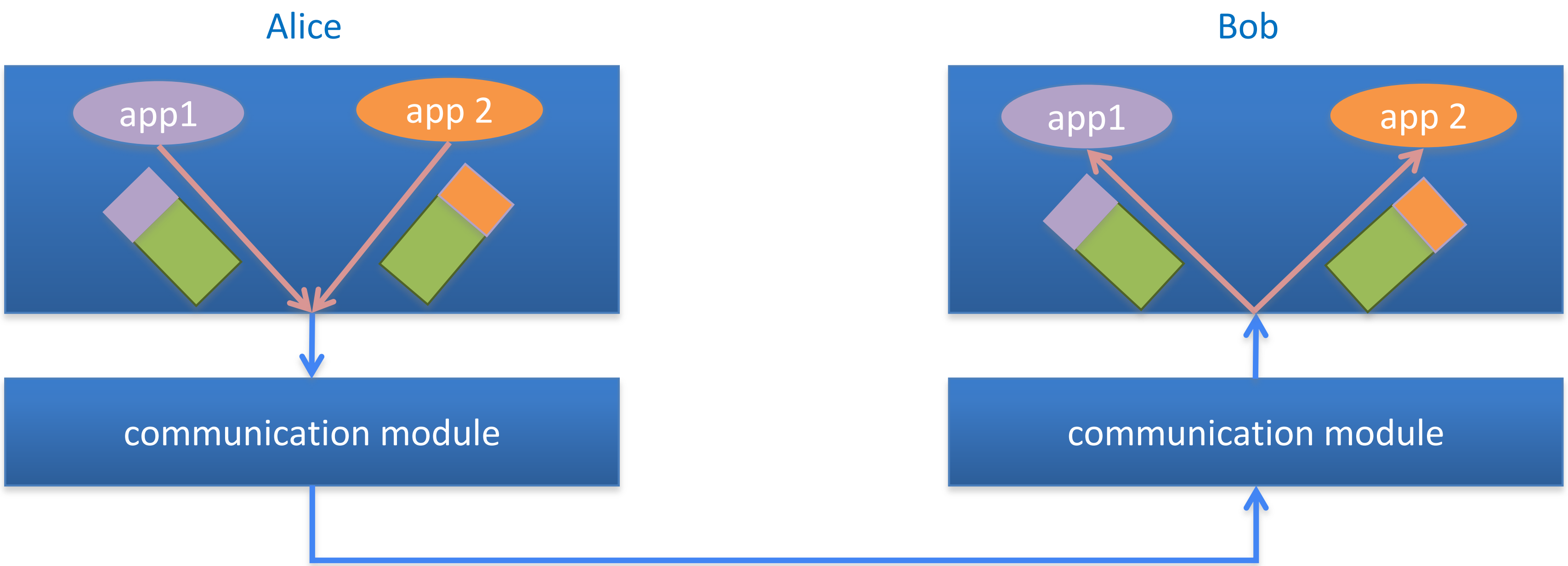
multiple apps over the same module (1)

- the described module is oblivious with respect to the content of the data it carries

- it is universal
  - can be used by different applications that require communication

- example
  - assume Alice has a device with two different apps
  - each app is communicating with the corresponding app in Bob's device
    - peer-to-peer communication
  - both apps are using the described communication module
  - the question is: how can the communication of one app be multiplexed with the communication of the other app by using the same communication module?

multiple apps over the same module (2)



multiple apps over the same module (3)

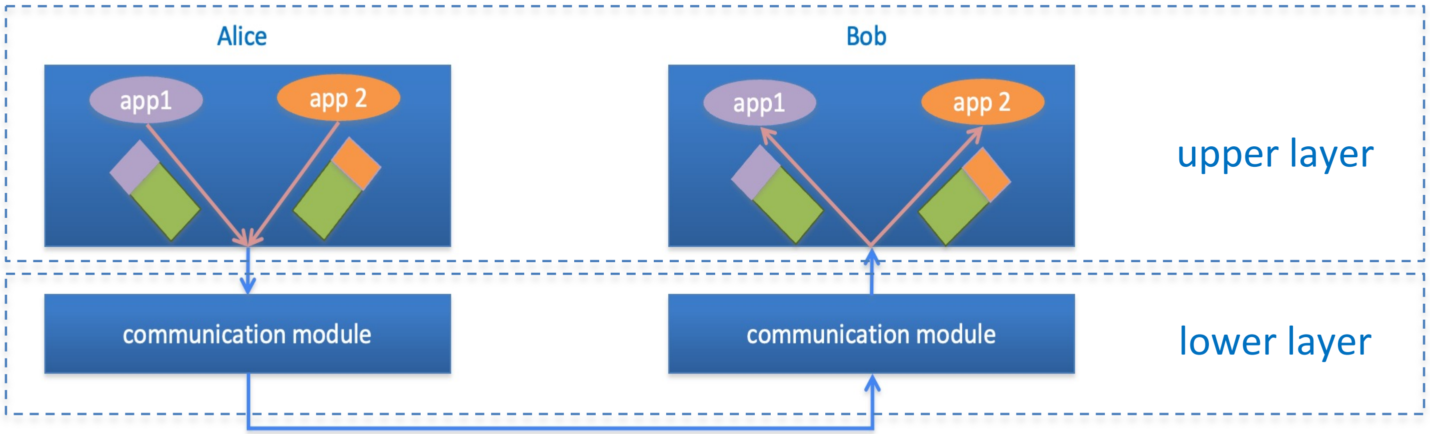
- each packet from a particular app has a label or port that corresponds to that app
  - this label is not recognized by the communication module

- we can speak that app1 of Alice establishes a peer-to-peer virtual connection with the app1 of Bob

- this leads us to a concept of a layered communication system

communication modules and the idea of layering (1)

- two communication peers can be thought of as residing in one communication layer
- in the simplest case there are two layers only
  - the upper layer uses the communication services of the lower layer
  - the lower layer provides communication service



communication modules and the idea of layering (2)

- we can even decompose the communication module into layers

- the simple case of two-layers
    - the lower layer understands only bits, does not understand headers or packet structure
    - in other words, the lowest layer provides an unreliable bit pipe for the upper layer
    - the upper layer uses that service to create reliable packet transmission

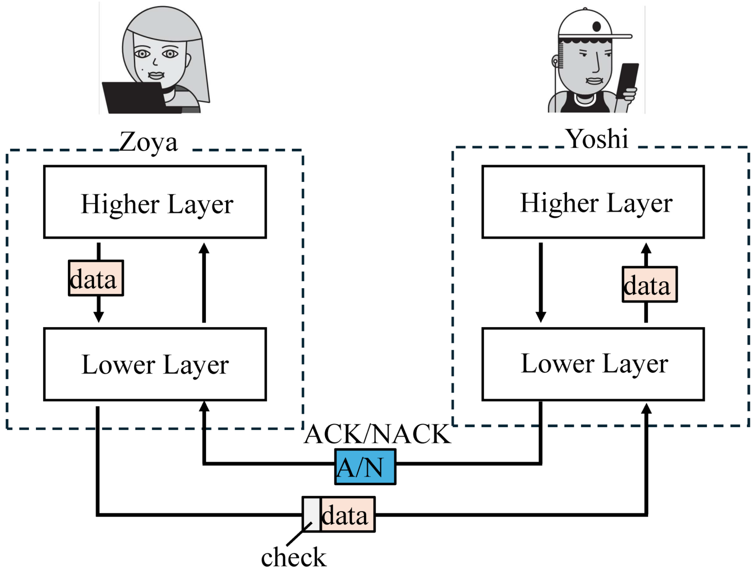
- note that the reliability feature can be shifted across layers or even duplicated

packet exchange over a layered system (1)

- Zoya sends data packets to Yoshi

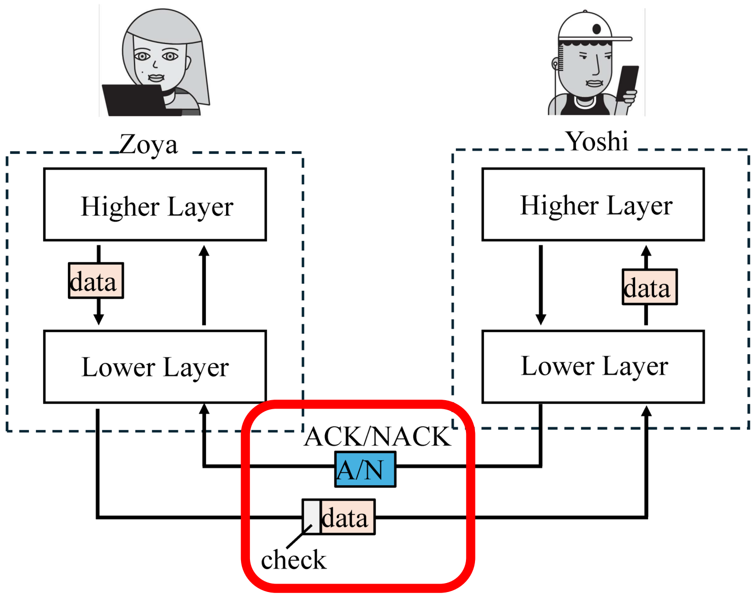
- lower layer as a service for the higher one
  - HL: higher layer
  - LL: lower layer

- protocol **reliable** packet transmission
  - integrity check information/error detection
  - acknowledgment (ACK/NACK)
  - automatic repeat request (ARQ)



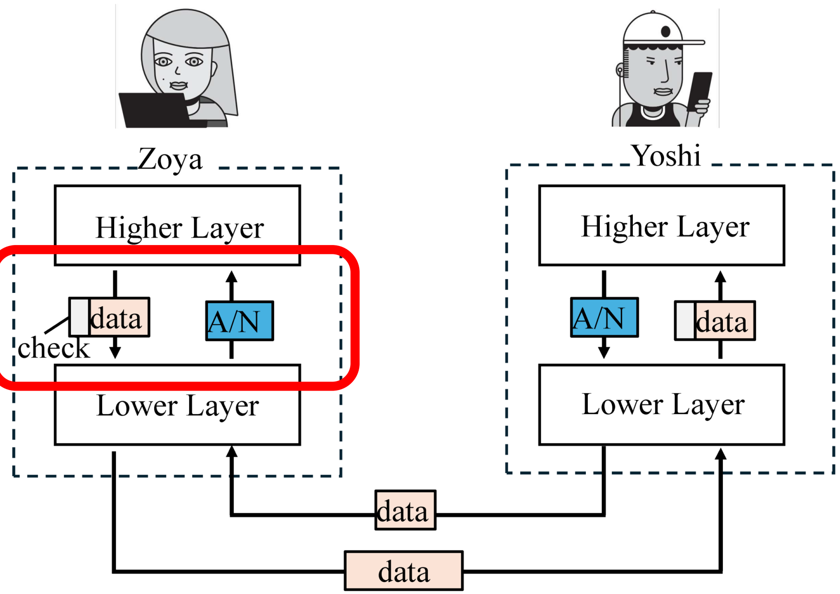
packet exchange over a layered system (2)

- the LL is a **reliable** service
  - the LL does error detection, ACK/NACK, and ARQ
- transparent packet transmission for the HL
- LL control information
  - integrity check information
  - ACK/NACK



packet exchange over a layered system (3)

- the LL is an **unreliable** service
  - no error detection, ACK/NACK, and ARQ at the LL
- the HL ensures reliable packet transmission
  - the HL does error detection, ACK/NACK, and ARQ
  - control information for the HL
- encapsulation

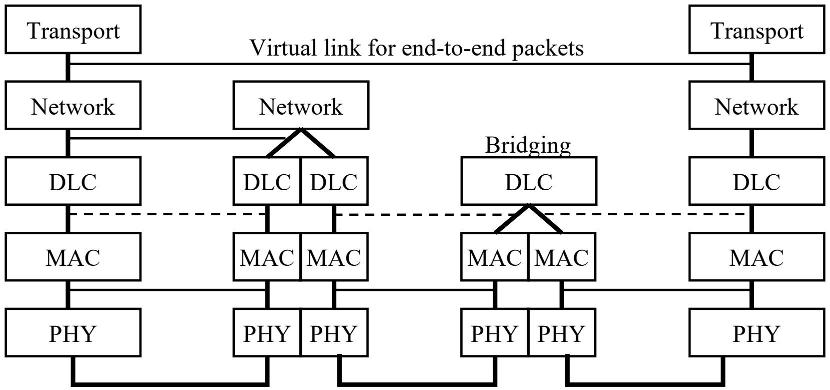


standard layers

recall the task decomposition

- we have basically decomposed the task into subtasks
  - the overall task is carried out by a protocol stack, which represents the set of algorithm run at a particular node in order to support the communication flow
  - each subtask is assigned to a particular layer

- protocol data unit is used for peer-to-peer communication between the entities at the same layers



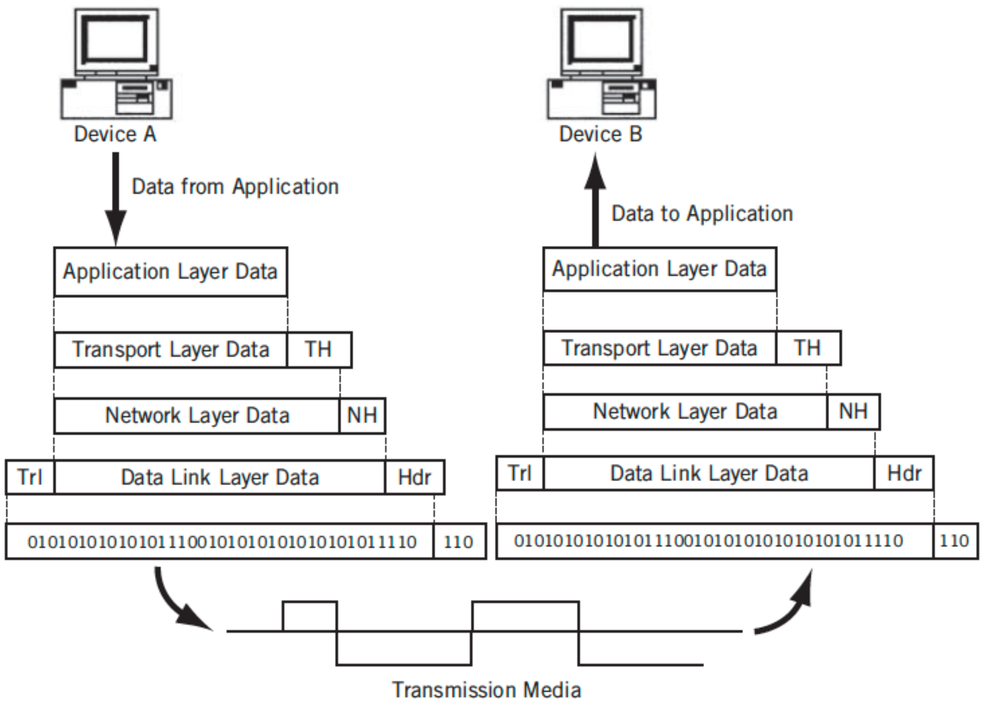
OSI vs TCP/IP layering

- two widely used layering models

OSI model	TCP/IP model
7-Application layer	Application
6-Presentation layer	
5-Session layer	
4-Transport layer	Transport
3-Network layer	Internet (IP)
2-Data link (DL) layer	Network interface
1-Physical layer (PHY)	

how does it look from a message point of view?

- A message is sent from one application to another. This is done as a packet: A sequence of bits.
- **Encapsulation**
- In the different layers, different kinds of control information (headers and/or tails) are added. This control information can for example be addresses or different kinds of protocol information.
- There can also be limitations to packet sizes, so packets may be split up.
- This also means that in addition to the message being sent, each layer is putting additional information (or overhead): How much is added on each layer depends on the protocols being used.

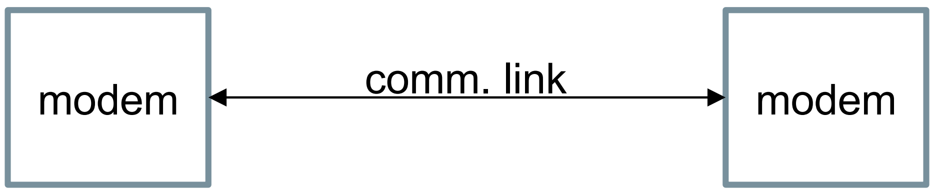


AALBORG UNIVERSITY

layer 1 – physical layer

functions:

- receives a packet of information bits with the headers of the upper layers
- at the sender, it provides unreliable bit pipe for the bits passed down
  - should do all the coding/decoding, processing,
    detection to support transmission / reception of the information bits



  - modem: digital data MOdulator and DEModulator

layer 2 – data link control layer

functions:

- provide an error-free packet link to higher layers
- in charge of error detection and correction
- delineate the beginning and ending of frames

the MAC sublayer: provide multiaccess communication capabilities

$$\underbrace{\boxed{\text{header}}\boxed{\text{packet: bit string}}\boxed{\text{trailer}}}_{\text{frame}}$$

layer 3 – network layer

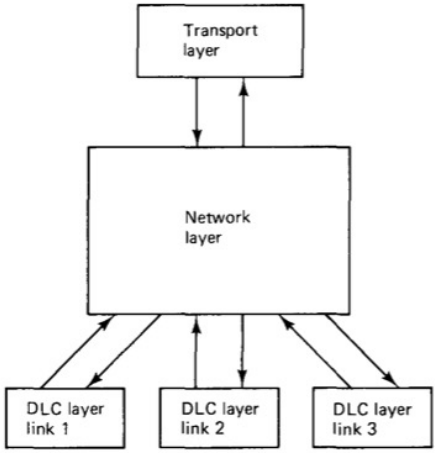
functions:

- routing: specify how a packet needs to travel over the network in order to reach its destination

- work with the transport layer

  to prevent or mitigate congestion
  - congestion control: refers to the the state of the network
  - flow control: refers to a specific pair of sender and receiver

- most complex: all peer process need to work together, that is, the network layer of all devices connected by it need to work in a consistent way



layer 4 – transport layer

functions:

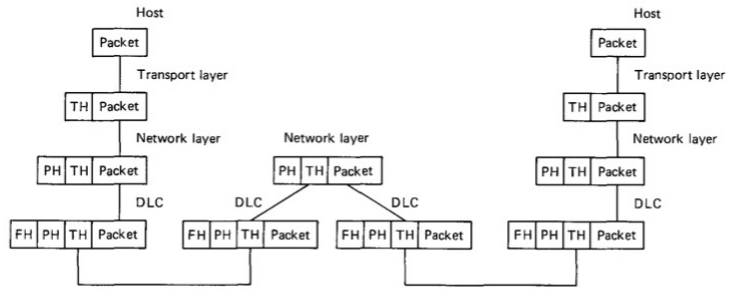
1. packet reassemble: put packets in order to be transmitted/when they are received

2. multiplex sessions

3. split sessions

4. guarantee reliability when network layer is unreliable

5. end-to-end flow control: work with the network layer



recall the onion analogy

layer 5 – session layer

functions

"session layer handles the interactions between the two end points in setting up a session"

- information to set up sessions
- define access rights
  - who pays for a given service

Dimitri Bertsekas and Robert Gallager. 1992. Data networks (2nd ed.). Prentice-Hall, Inc., USA.

layer 6 – presentation layer

functions:

- data encryption: security 
- data compression: reduce number of bits (often done before encryption)
- code conversion: some terminals might use ASCII code, others EBCDIC code

layer 7 – application layer

- what is left: things that are specific to the particular application
- for example: email programs, web browsers, file explorers
- example of functions: file sharing, message handling, database access
- common protocols: HTTP, FTP, SMTP

takeaways

- OSI is just the most classical layering scheme
- however, there are many others and the functions of the layers may change
- the important thing is that layering ease the implementation of the network

OSI model	TCP/IP model
7-Application layer	Application
6-Presentation layer	
5-Session layer	
4-Transport layer	Transport
3-Network layer	Internet (IP)
2-Data link (DL) layer	Network interface
1-Physical layer (PHY)	

network systems

why do we need networking?

computer networks: crucial for personal, business, and governments

- enabled new computing paradigms
- enabled new services for users and customers of commercial enterprises

historically, the development of computer networks occur in two fronts:

- network protocols: define the methods and mechanisms necessary to achieve reliable communication between two parties
  - mechanisms (flow/error control), syntax (headers/fields), system design and implementation

- network systems: realize the implementation of network protocols

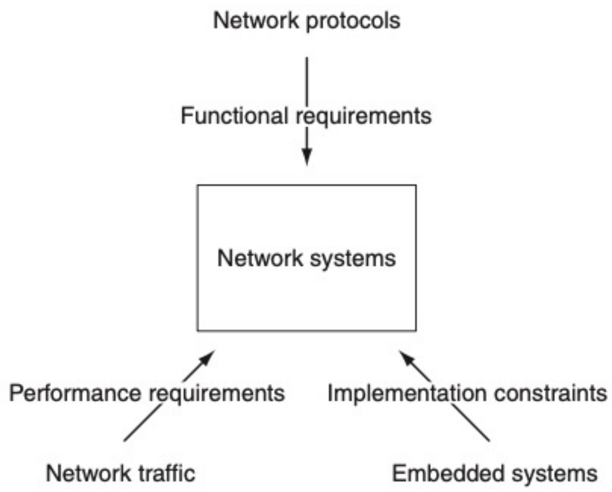
what are network systems?

- **definition:** hardware and software that realize the implementation of the protocols given performance requirements and resource limitations

a first way to classify network systems:

1. infrastrucure systems:
ensure operation of the core network

2. end systems: system visible to end users (mobile phones, modems)



protocol stacks: recalling the OSI model

- each node has its protocol stack that abstract its operation
- layer-by-layer equivalence
- openess: independent development
- flexibility: stacking protocols together

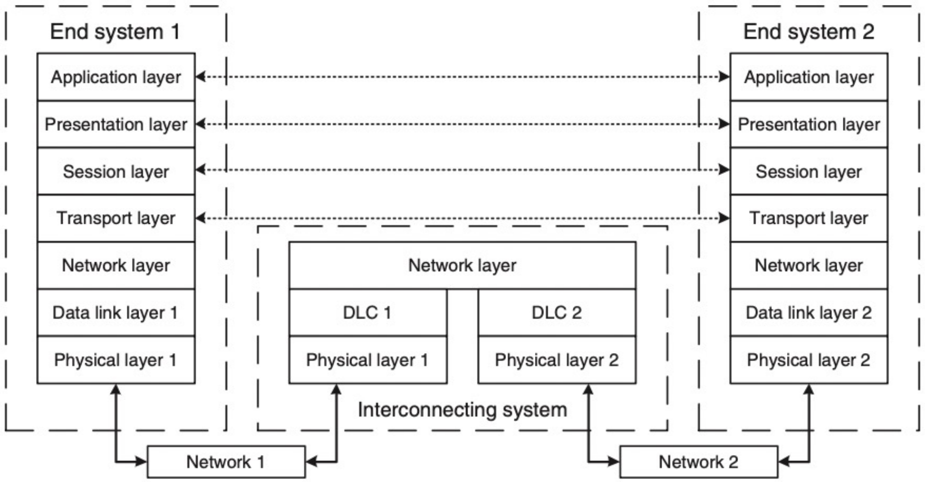


Figure X. Node-to-node **heterogeneous** communication

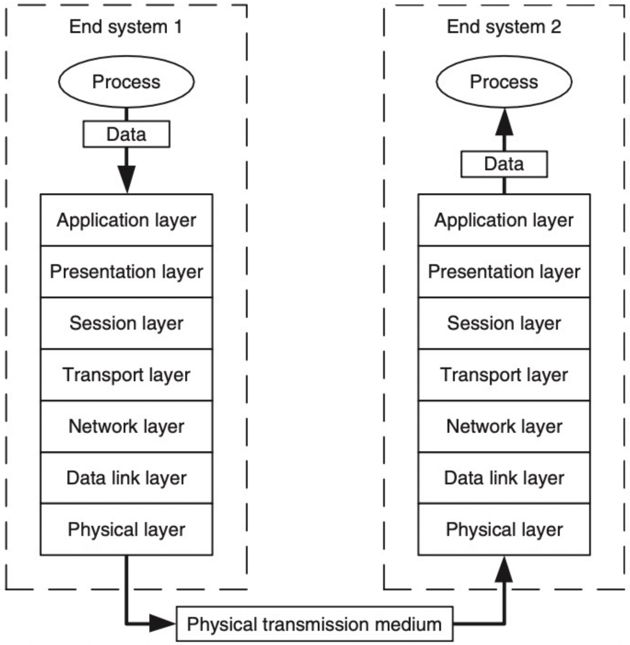
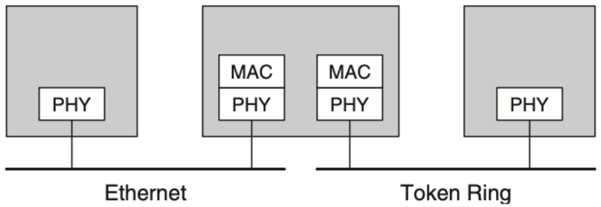


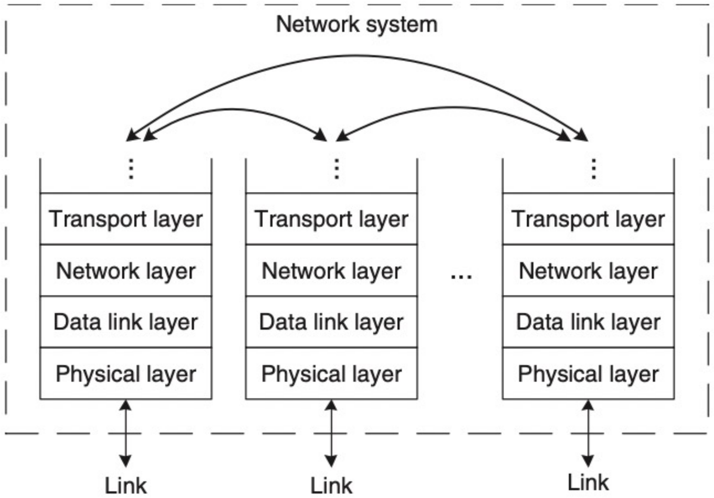
Figure X. Node-to-node **homogeneous** communication

a more formal definition of network systems

- **formal definition:** a network system is a computation system that executes protocol stacks and switches data among the protocol stacks

special case: for end systems there is only a single protocol stack

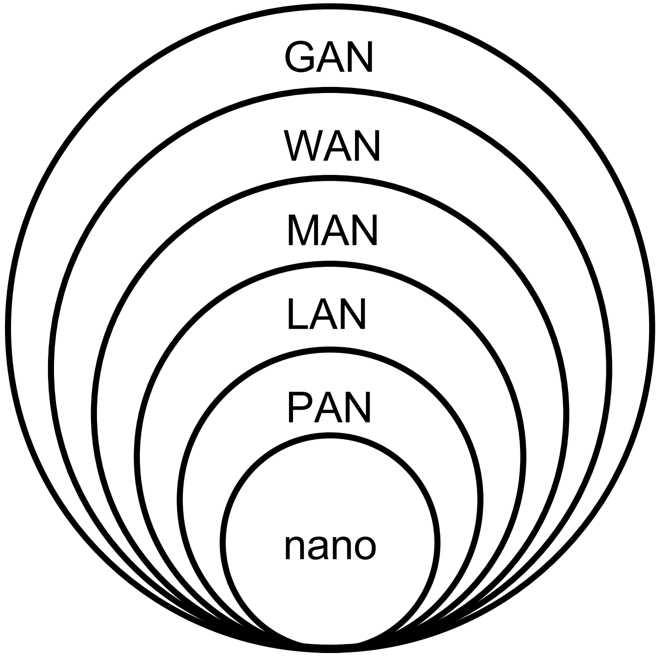




each node in the network has its own protocol stack

network types: a geographic classification

- **nanoscale:** hardware scale, actuators, sensors
- **personal (PAN):** USB, Bluetooth
- **local (LAN):** home, school, office
- **metropolitan (MAN):** city level
- **wide (WAN):** cities, country
- **global (GAN):** arbitrary number of LANs



*AN – area network

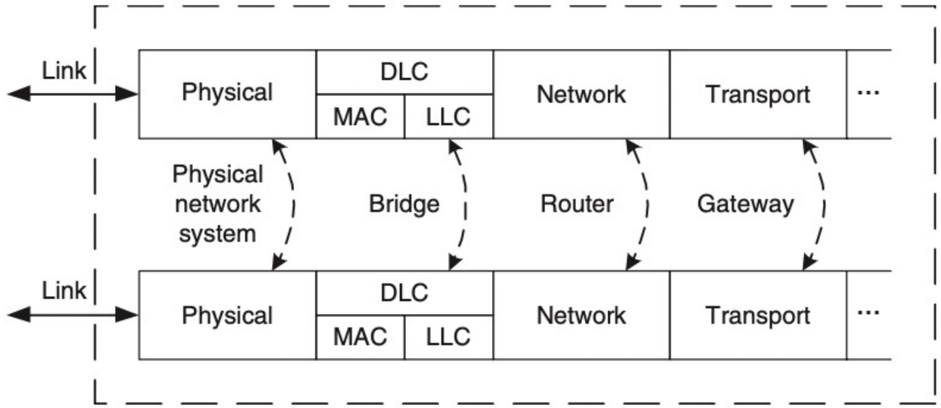
network topology

- topology: how the nodes of a network system are arranged
- the network topology relies on the following elements:
  - network nodes
  - network links
- the network topology can also be seen from different perspectives
  - we will start with the physical one,
    where nodes represent physical entities and connections are physical as well

network topology: nodes

classification based on which layer the network system switches data:

1. **physical:** PHY layer (repeater/hub)
2. **bridges/switches:** DLC layer data
3. **routers:** network layer
4. **gateways:** transport or higher layers



- network interfaces/modems

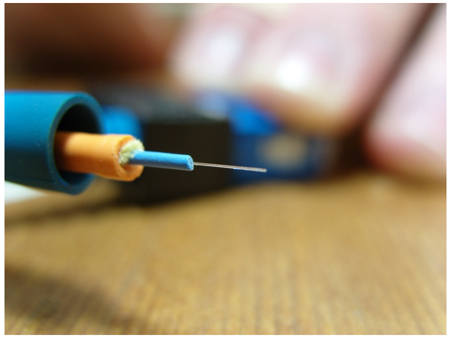
- firewalls

network topology: physical links

how the systems are connected to each other can be classified according to the transmission media/transmission technology (layers 1 and 2)

wired

- coaxial cable
- twisted pair
- optical cable
- phone/power lines

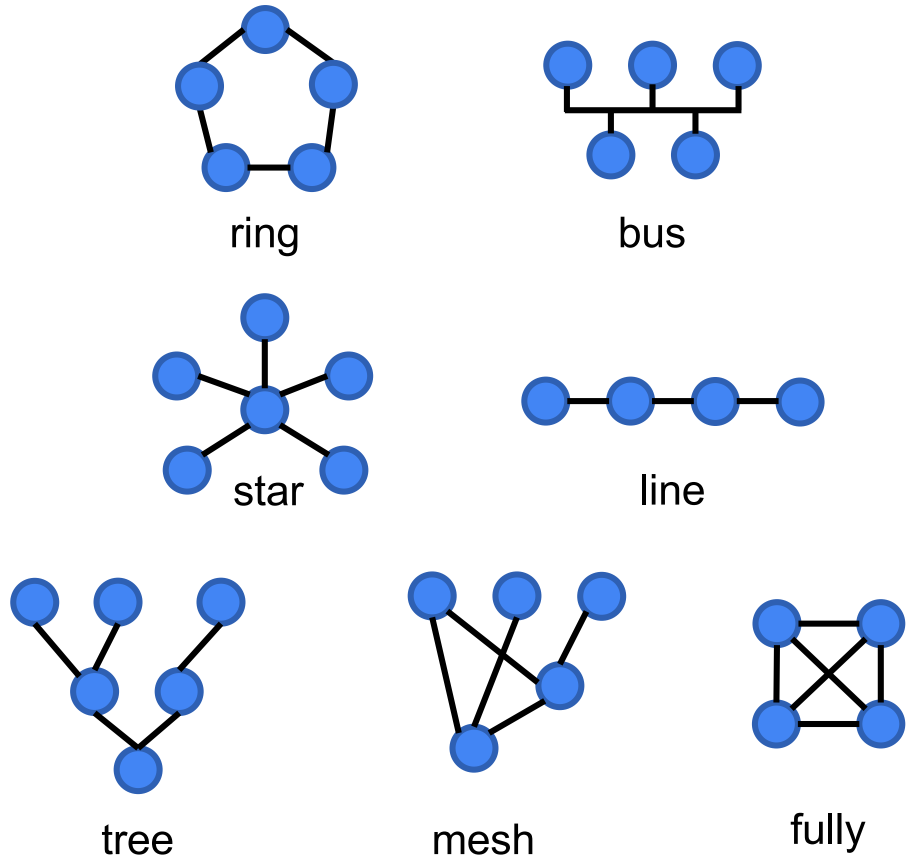


wireless

- terrestrial microwave
- satellites
- cellular network
- free-space optical
- wireless local networks (Wi-Fi)

network topologies: some classifications

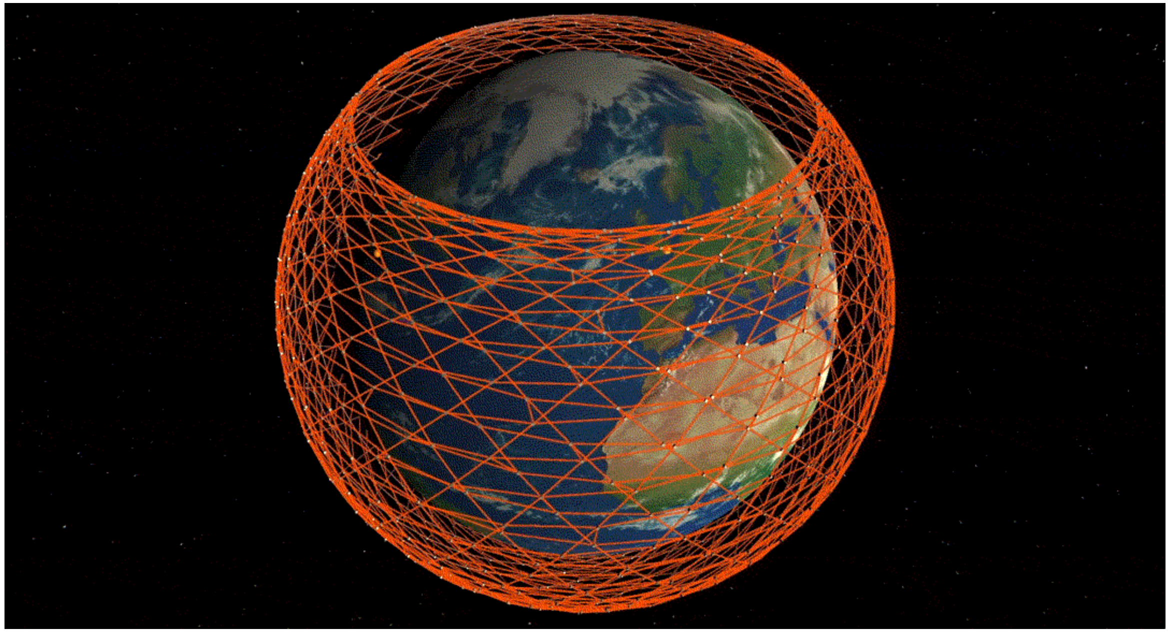
- how to connect the network nodes?
- why is this important?
  - throughput and reliability
- common layouts:
  - bus, star, ring, mesh, fully, tree



overlay network:
virtual network built on top of another network

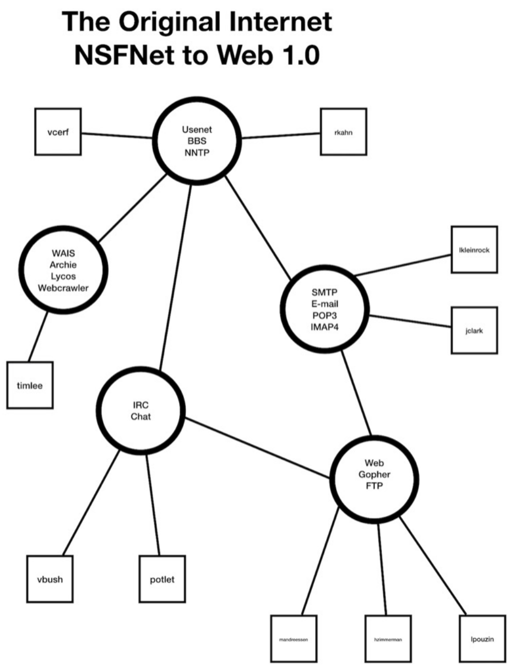
internet is a network of networks

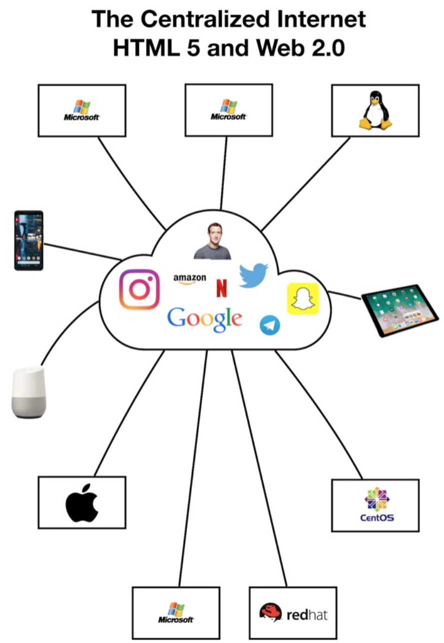
infrastructure topology does not need to be static!



Starlink animation by Mark Handley
check animation at: https://www.youtube.com/watch?v=AdKNCBrkZQ4

internet: decentralized vs. centralized





source:
https://hackernoon.com/the-evolution-of-the-internet-from-decentralized-to-centralized-3e2fa65898f5

cloud computing

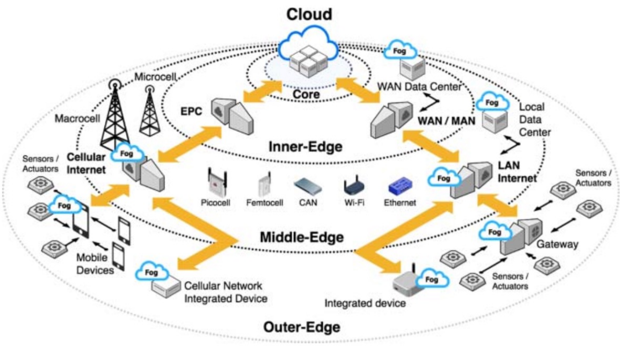
- **concept:** on-demand availability of computer system resources
  - data storage, computing power
- pay-as-you-go: if you need more, more resources are going to be allocated to you
- example: Amazon Web Services (AWS)

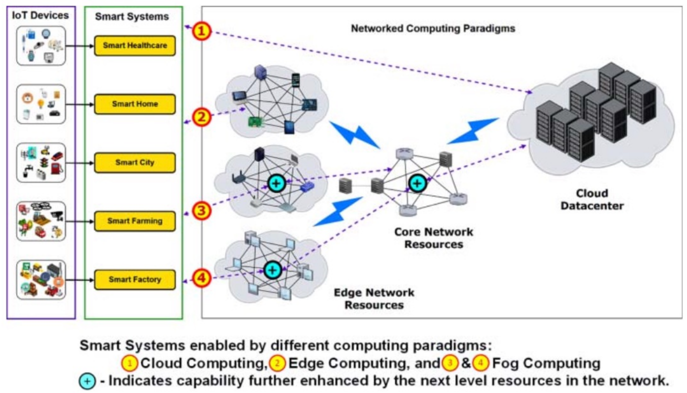
AWS data center in Sweden

inside AWS

edge computing

- **concept:** bring computation and storage close to the sources of data
  - reduce response time
  - save bandwidth
- example of factory automation and private 5G networks
- key technology to enable autonomous cars (!!!)





Buyya R., Srirama, S. N., Fog and Edge Computing - Principles and Paradigms, 2019.

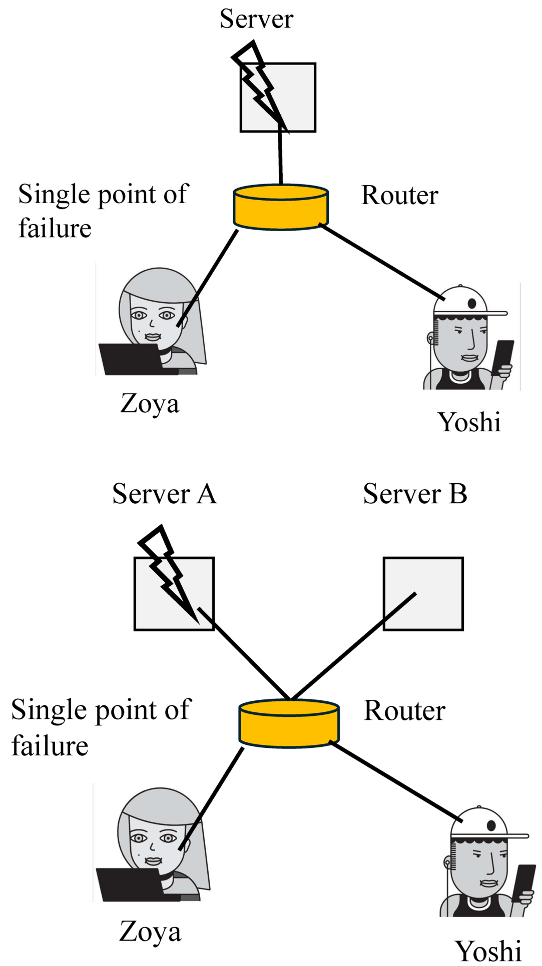
examples

AALBORG UNIVERSITY
DENMARK

Connectivity

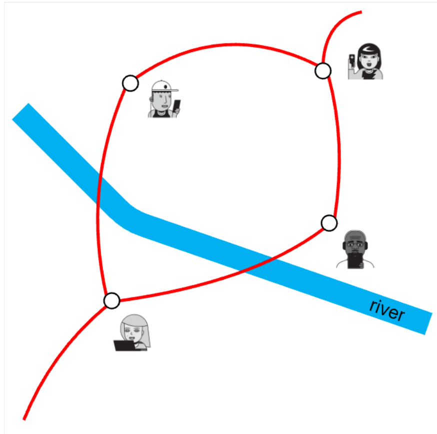
single point of failure

- single point of failure (SPOF)
  - a single failure can stop entire systems
- resilient topology
  - robustness based on redundancy
  - price of redundancy
- at the application level
- at the communication level

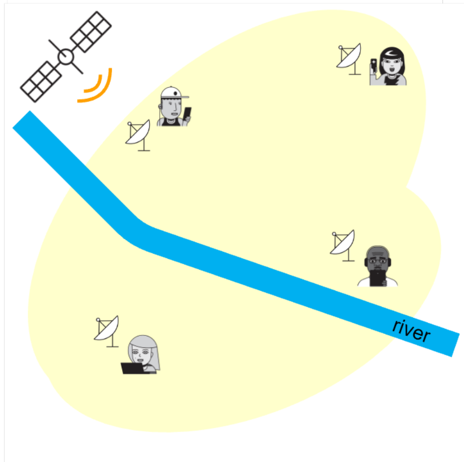


network topology for a rural community

- characteristics of the communication service
  - large area to be covered
  - small number of customer
  - low throughput per area/connection density



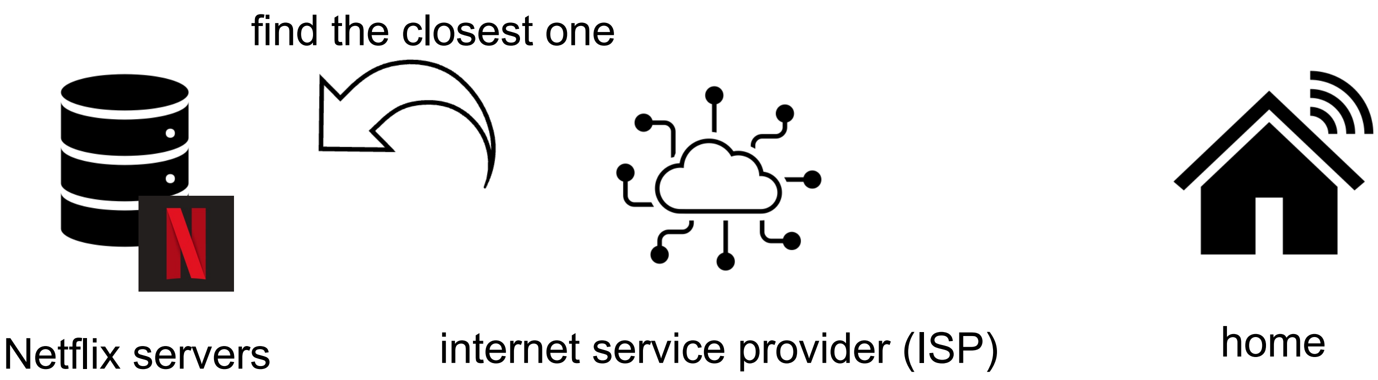
- optical fiber network
  - cable infrastructure might be expensive
  - optical network terminal (ONT)
  - point of presence (PoP)



- satellite network
  - wide coverage
  - very-small aperture terminal (VSAT)
  - deployment of many satellites with a single launch
  - good option for this case!

an example: Netflix – video stream

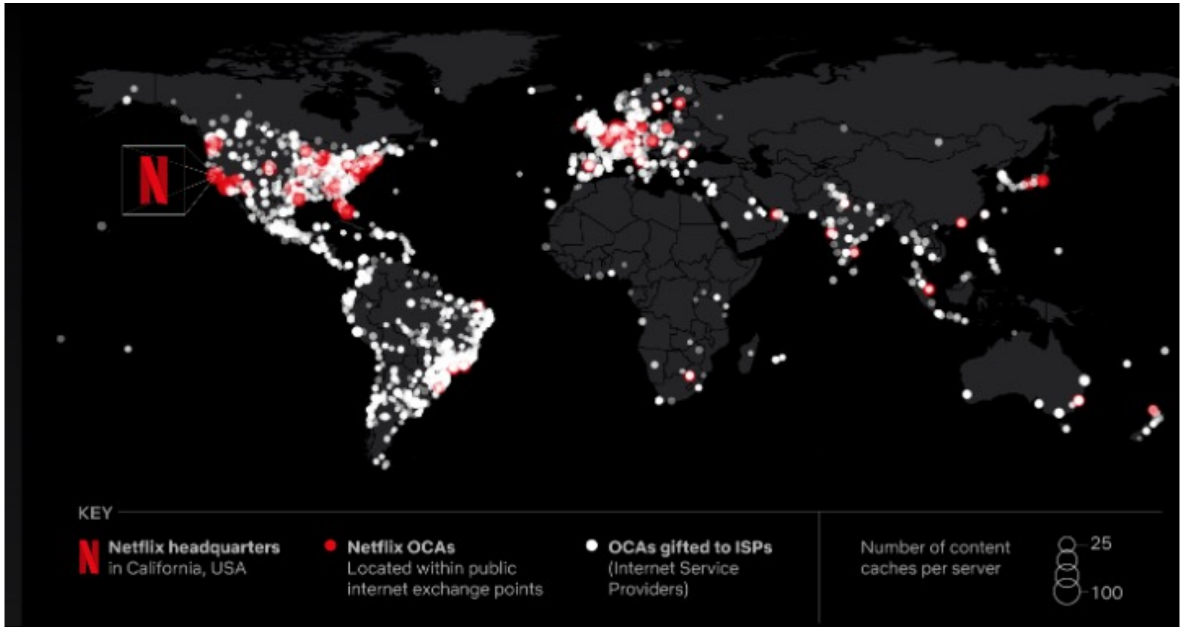
- travelling through multiple paths:



Content available on Netflix website

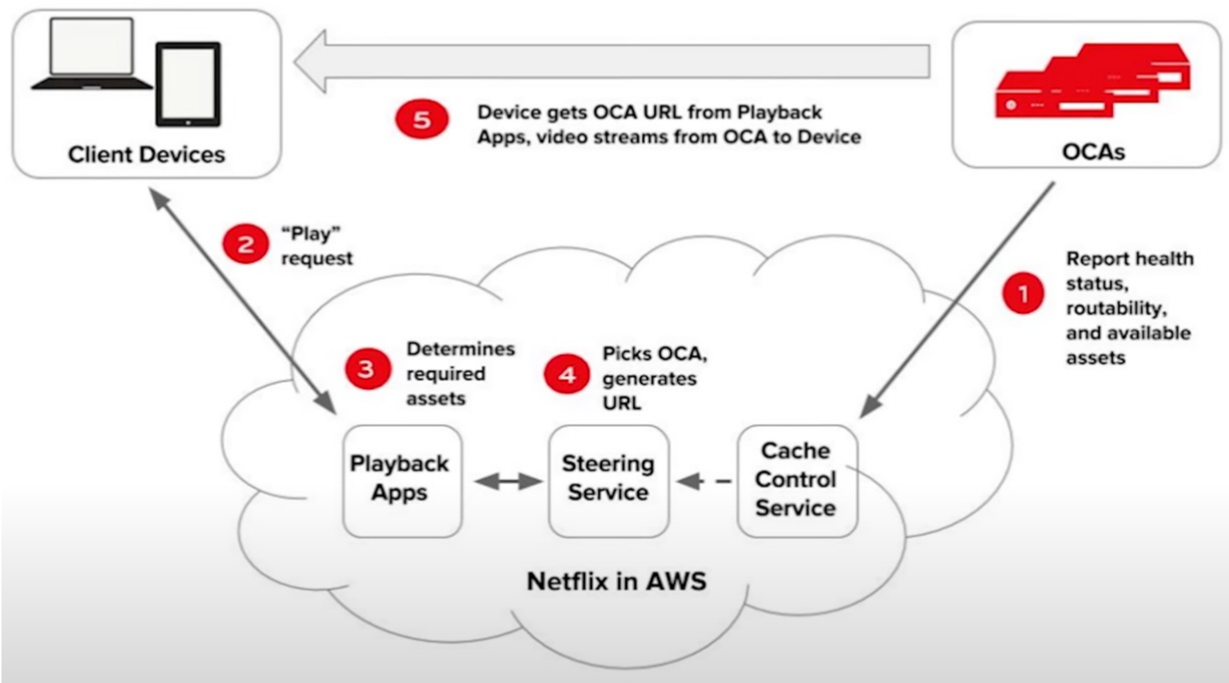
content deliver network (CDN)

- Netflix installs redundant servers over the entire world
- Netflix makes contracts with ISPs to install their own hardware via OCAs (open connect appliance)
- These OCAs can be equipped with 280 TB of data, which can store most of Netflix relevant content
- OCAs have the capacity of going to 90 Gbits/s of data (13,000 people watching at once)



amazon web services (AWS)

- Netflix stores information of the users using the cloud servers from Amazon



summary and outlook

- basic ideas behind the layered communication systems
- layering offers a structured way to build communication protocols
  - enables scaling across various topologies
- we have discussed the functionality of different layers
  - OSI vs. TCP/IP system
- network systems and topologies
  - simple topologies, network of networks
- cloud and edge computing