IN2140:
Introduction to Operating Systems and Data Communication



Data Communication:

Network structures

Programming the network

```
import urllib.request

contents = urllib.request.urlopen(
    urllib.request.Request(
        "https://www.uio.no/studier/emner/matnat/ifi/IN2140/v25/index.html")
    ).read()

print(contents.decode('utf-8'))
```

Programming the network

```
import u

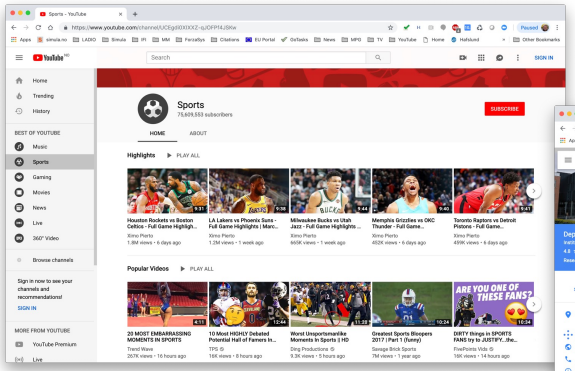
contents
    urll

    ).re

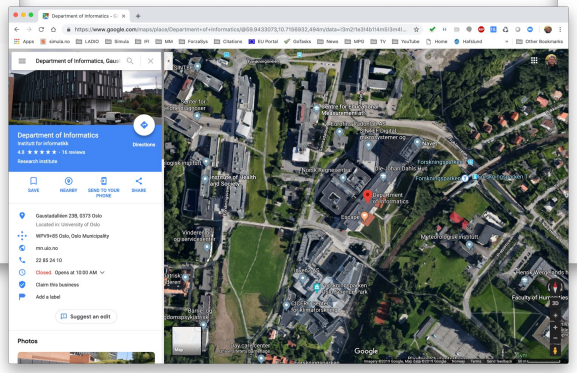
print(co
```

```
<!DOCTYPE html>
<html lang="no">
  <head>
    <meta charset="utf-8" >
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta id="viewport" name="viewport" content="width=device-width, initial-scale=1"
/>
    <meta name="format-detection" content="telephone=no">
    <meta name="generator" content="Vortex" />
      <title>Semesterside for IN2140 -  - Universitetet i Oslo</title>
      <meta property="og:title" content="Semesterside for IN2140 -  - Universitetet i
Oslo" />
    <meta name="twitter:card" content="summary" />
    <meta name="twitter:site" content="@unioslo" />
    <meta name="twitter:title" content="Semesterside" />
      <meta name="twitter:description" content="Les denne saken på UiOs nettsider." />
          <meta name="twitter:image"
content="https://www.uio.no/vrtx/dist/resources/uio2/css/images/social/uio-twitter-
share-1000x1000.jpg" />
      <meta name="twitter:url"
content="https://www.uio.no/studier/emner/matnat/ifi/IN2140/v25/index.html" />
    <meta property="og:url"
content="https://www.uio.no/studier/emner/matnat/ifi/IN2140/v25/index.html" />
    <meta property="og:type" content="website" />
        <meta property="og:description" content="Les denne saken på UiOs nettsider."
/>
            <meta property="og:image"
content="https://www.uio.no/vrtx/dist/resources/uio2/css/images/social/uio-share-
1200x630.jpg" />
            <meta property="og:image:width" content="1200" />
            <meta property="og:image:height" content="630" />
            <meta property="og:updated_time" content="1741591281" />
    <link rel="shortcut icon"
href="/vrtx/dist/resources/uio2/css/images/favicon/favicon.png">
```

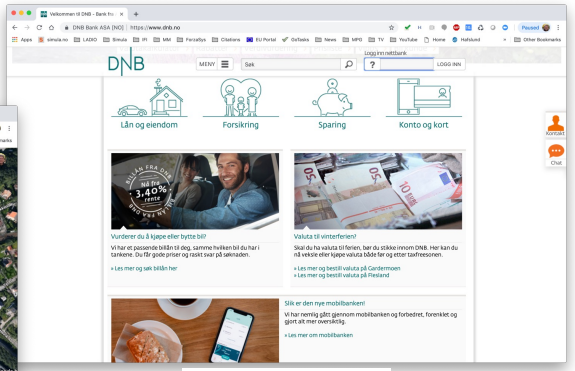
Web-based software



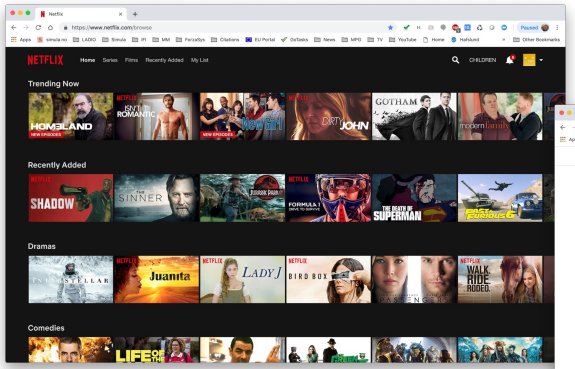
Video



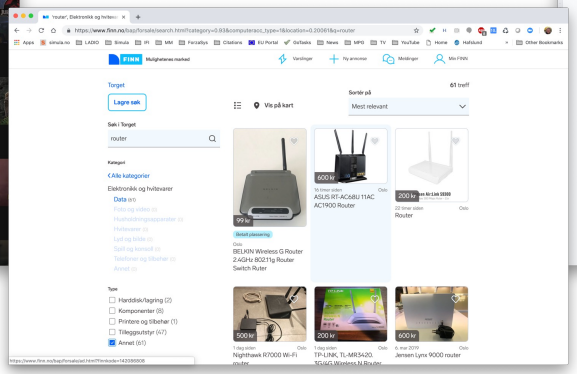
Navigation



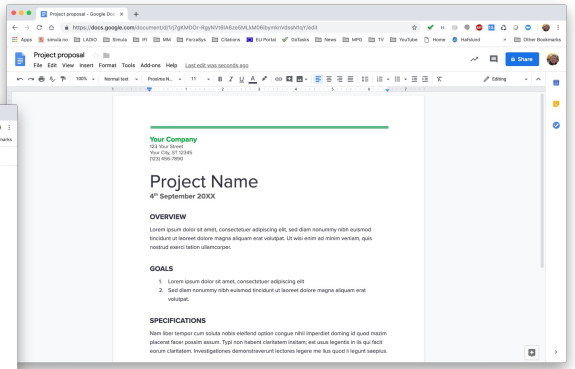
Banking



TV



Shopping



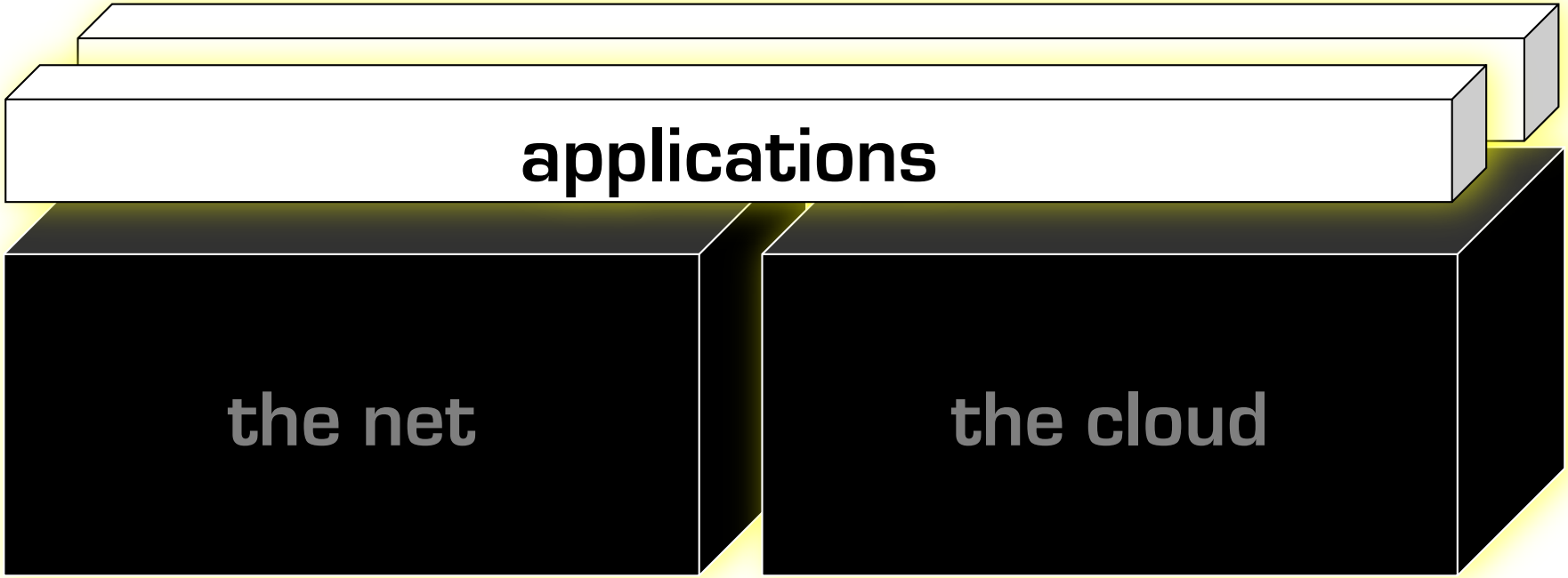
Office tools

Isn't this enough?

In everyday live,

the network and network functions are black boxes,

language is very inaccurate ... and that is only fair



But we – as software designers, architects, developers or researchers – must understand more

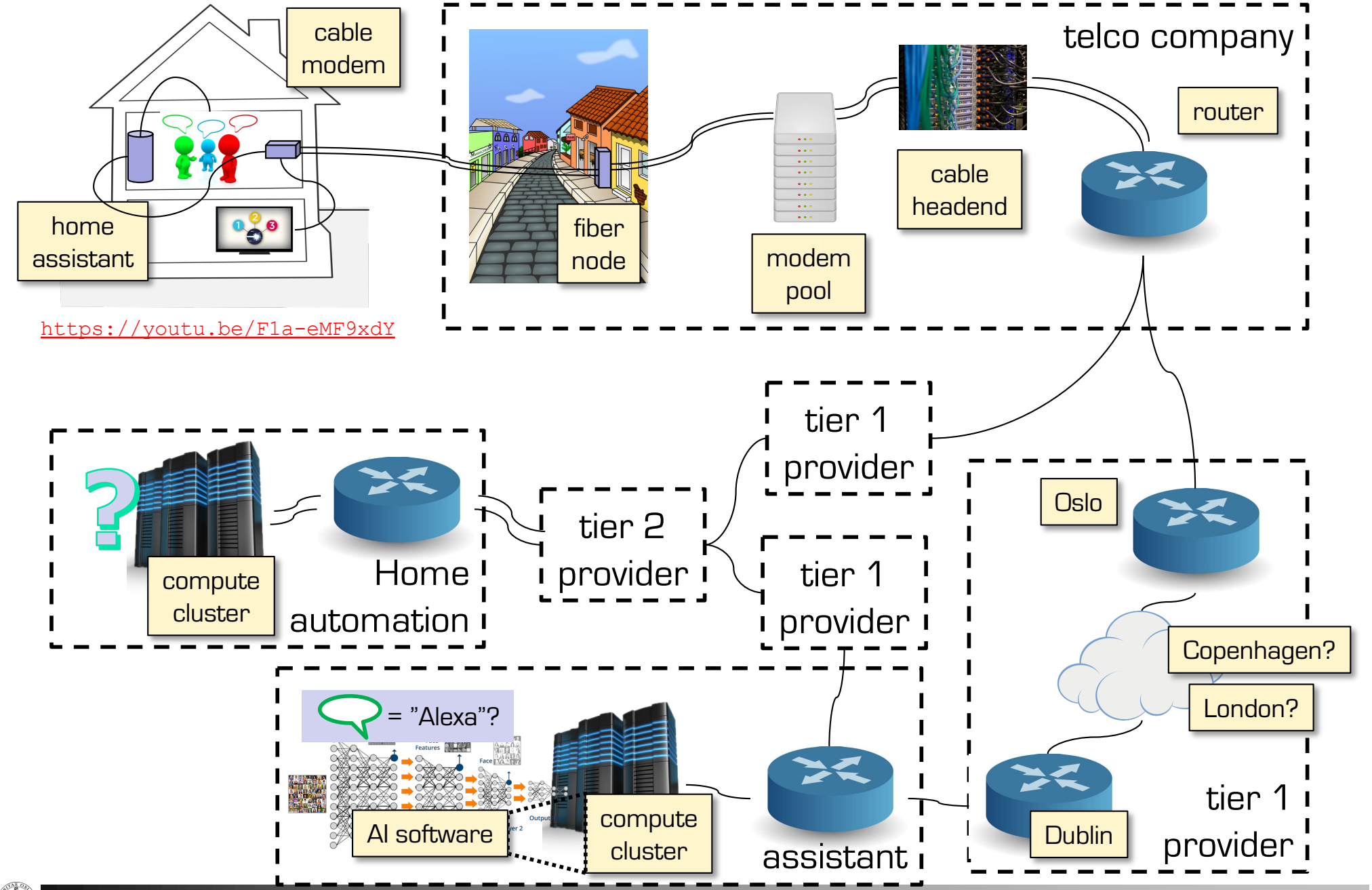
Is this the whole story?



How do you get from a cable in the ground to a virtual home assistant?



A possible communication path

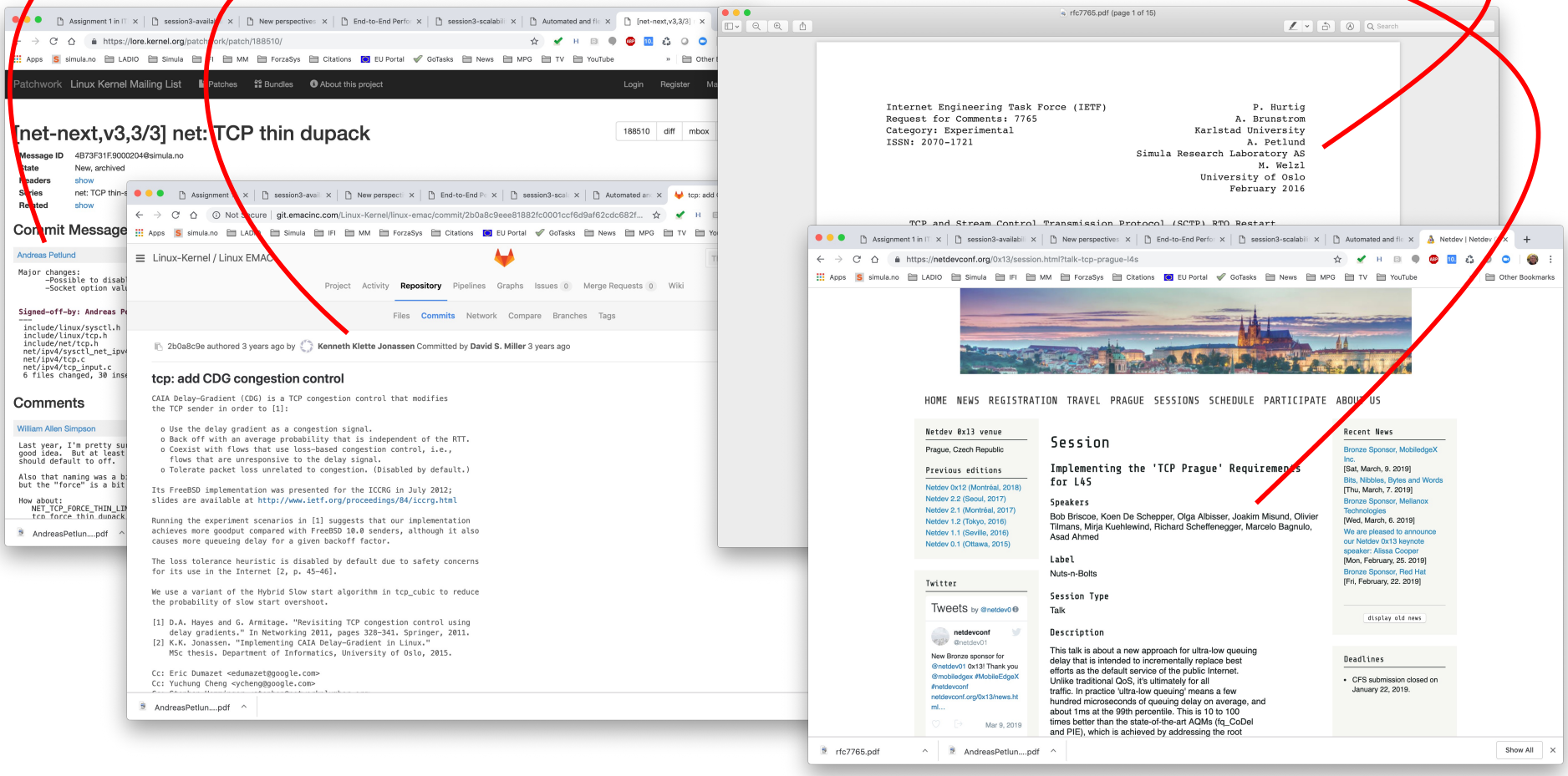


But does it matter if *we* understand it?

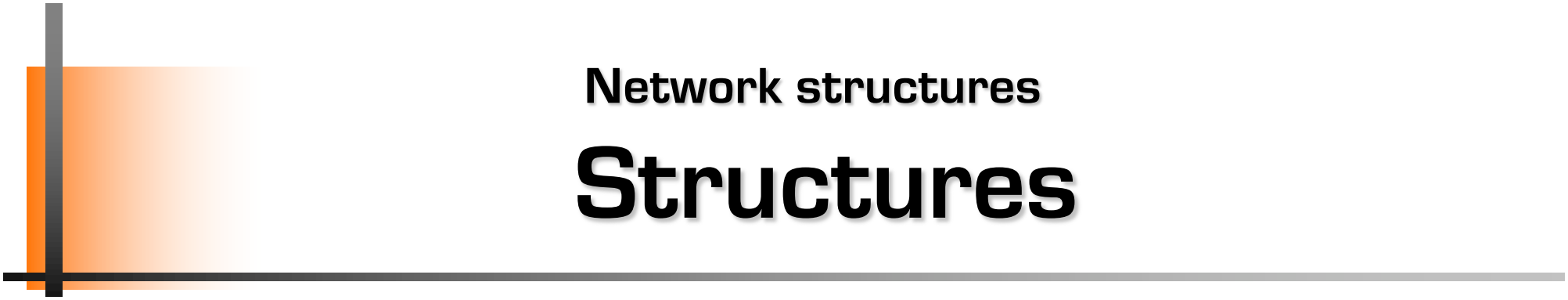
Linux contributions



Standards contributions



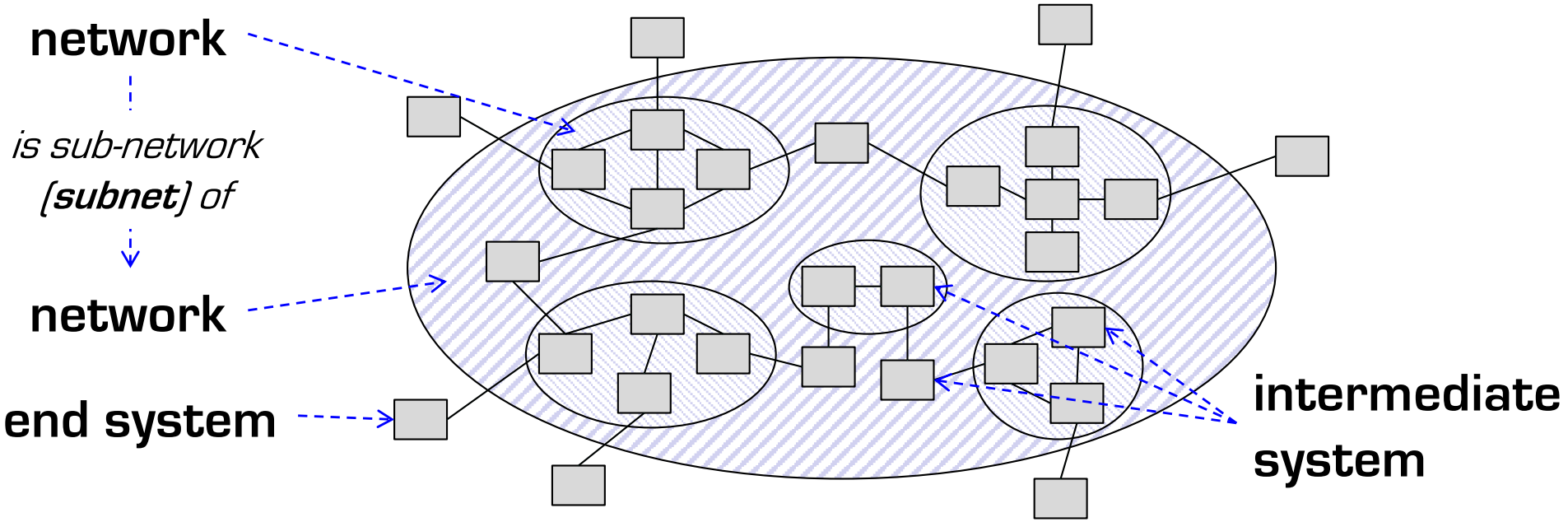
IN2140:
Introduction to Operating Systems and Data Communication



Network structures

Structures

Network Components



ES - End system

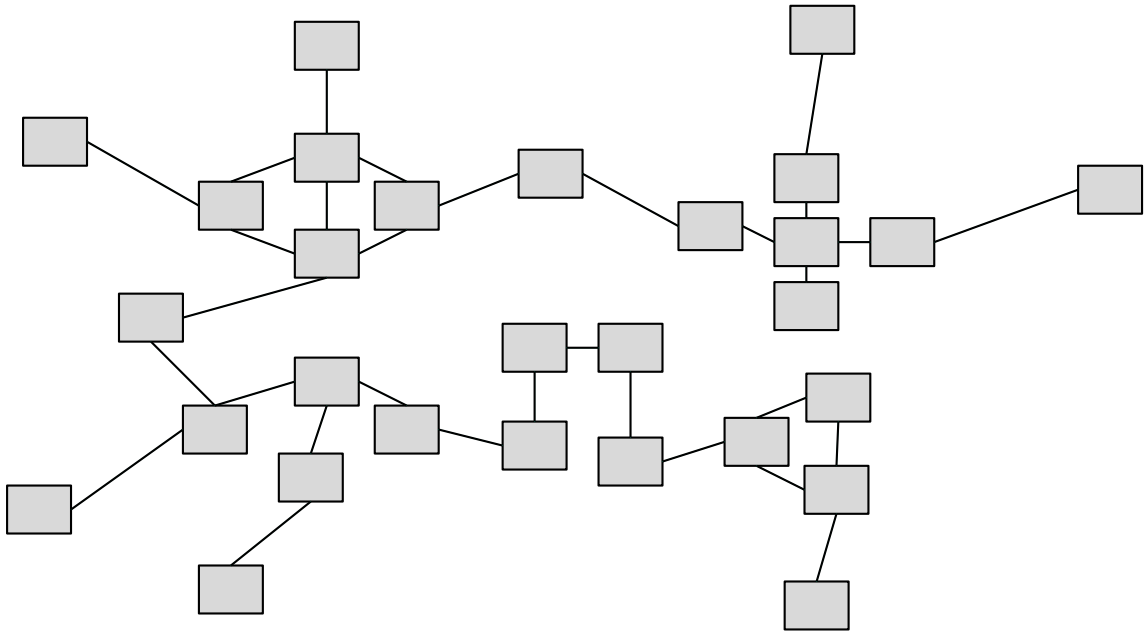
- end systems are “at the edge” of a network
- examples: computer, mobile phone, tablet, smart watch, printer, TV, smoke detector, weather station, lamp, door opener, fridge, traffic light ...

IS - Intermediate system

- examples:
  - router, switch
  - base station, modem
  - gateway: web proxy, firewall, NAT gateway
  - repeater, bridge

node

Network Structures



without the lines showing network boundaries,
this looks like a typical **graph**

there are **nodes**, and **edges** connecting pairs of nodes to each other

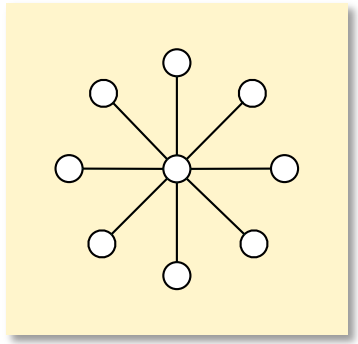
a specific arrangement of nodes and edges is called a **topology**

the terminology implies that edges are network connections

Network Structures

Point-to-point channels

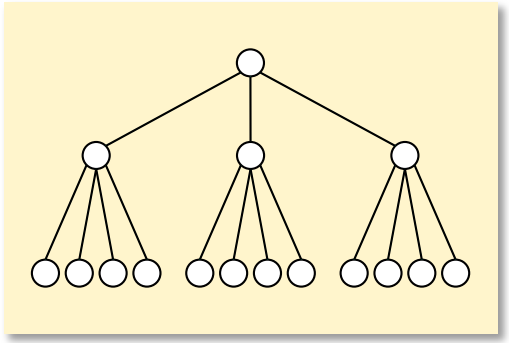
- Docsis (cable modems)
  - star topology

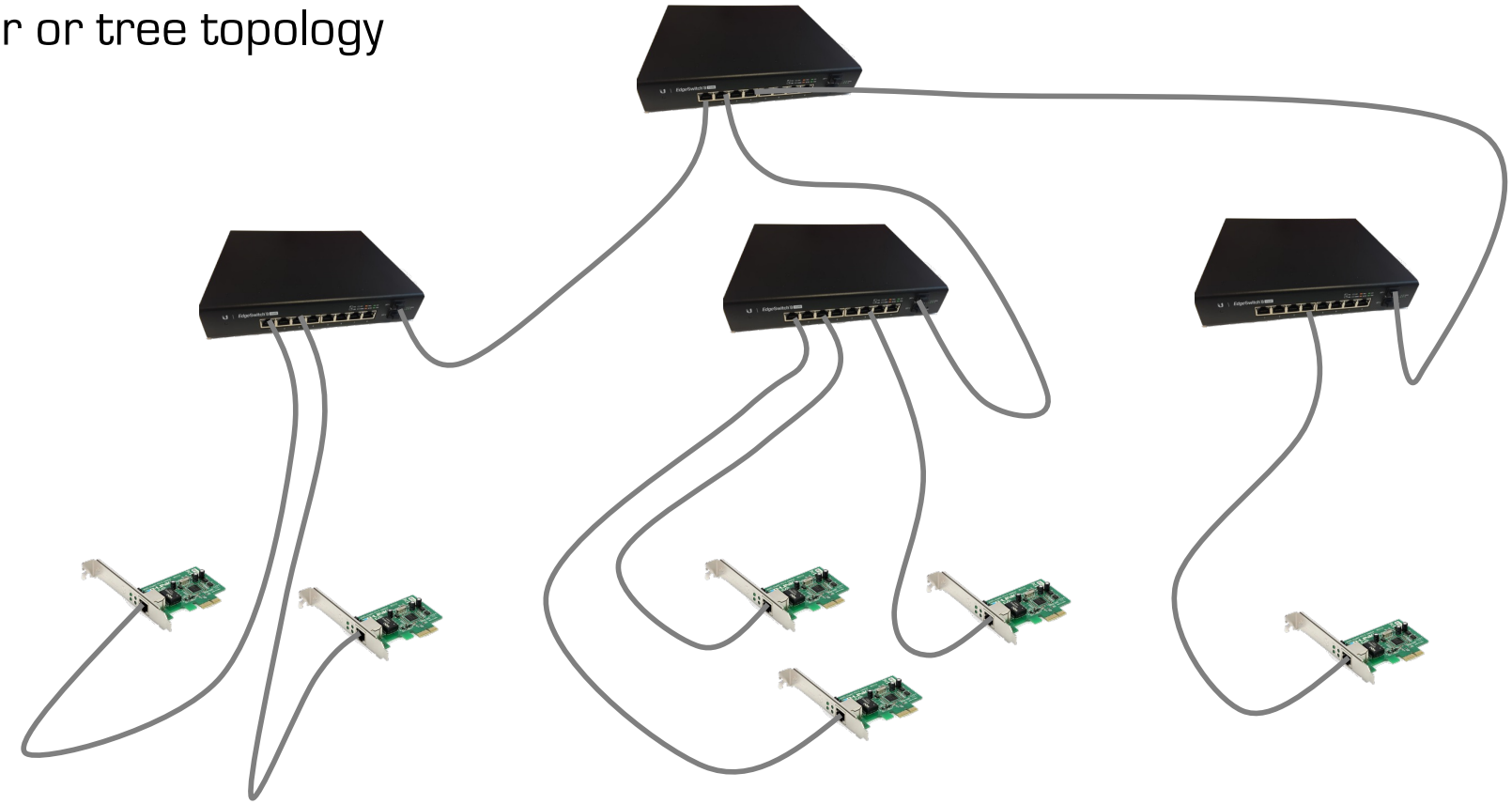


Network Structures

Point-to-point channels

- Docsis (cable modems)
  - star topology
- Gigabit Ethernet ("1GB Ethernet")
  - star or tree topology

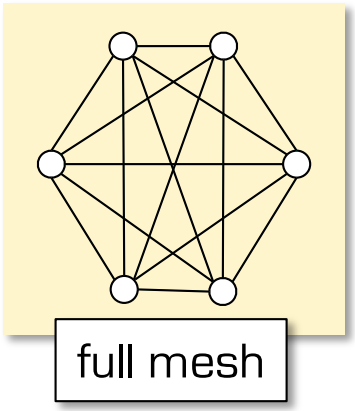


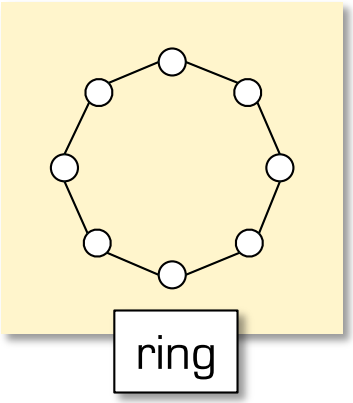


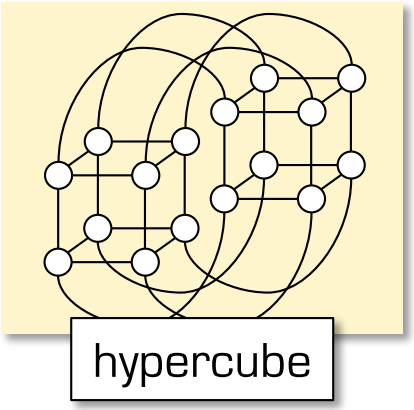
Network Structures

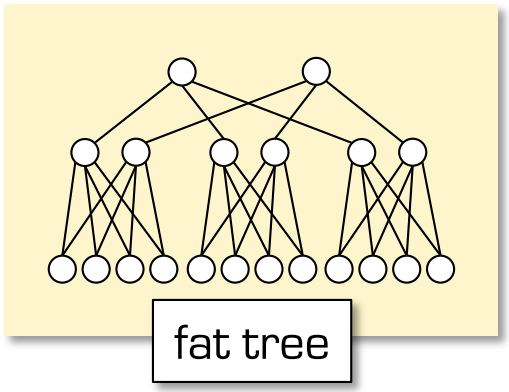
Point-to-point channels

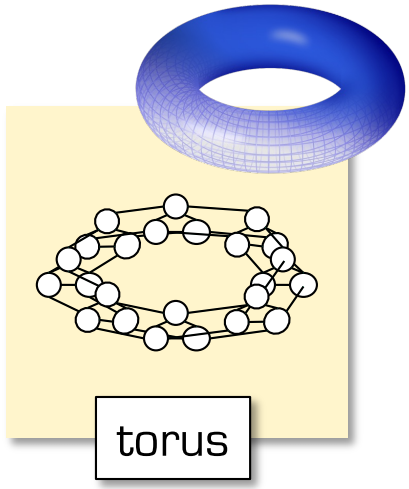
- Docsis (cable modems)
  - star topology
- Gigabit Ethernet ("1GB Ethernet")
  - star or tree topology
- IEEE 802.5 "TokenRing" (outdated)
  - ring topology
- Some supercomputers use
  - full mesh topology
  - hypercube topology
  - torus topology
  - fat tree topology



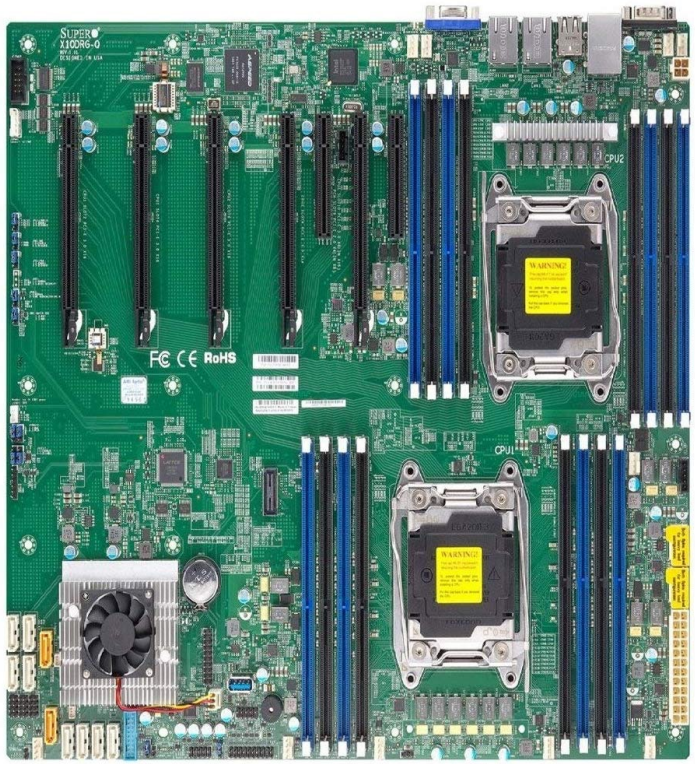








Questions

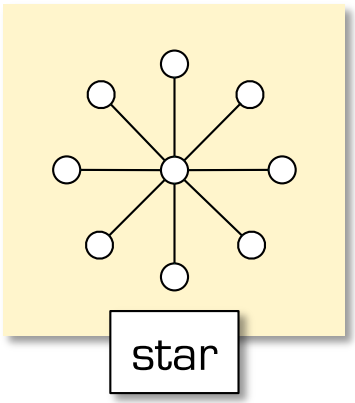


The majority of computers today use PCI Express (PCIe) for connectivity between devices

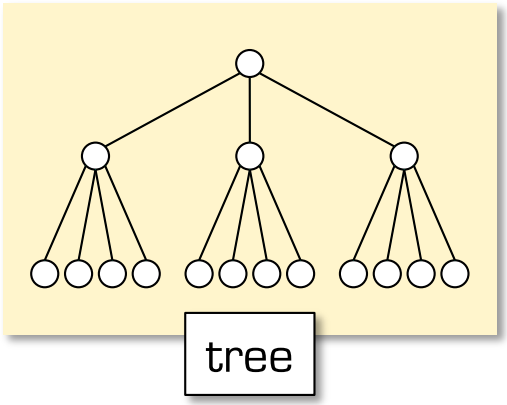
Q: Which topology do most desktops use between devices today?

Questions

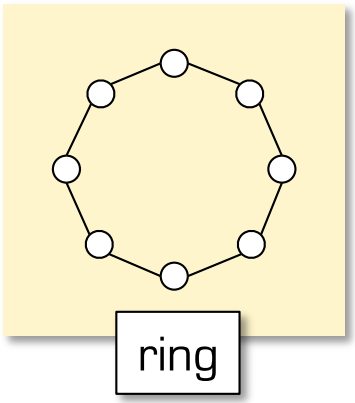
Q: Which topology do most desktops use between devices today?



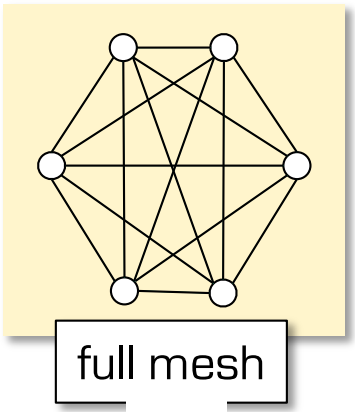
A



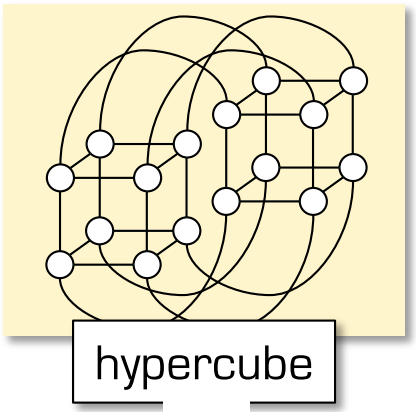
B



C

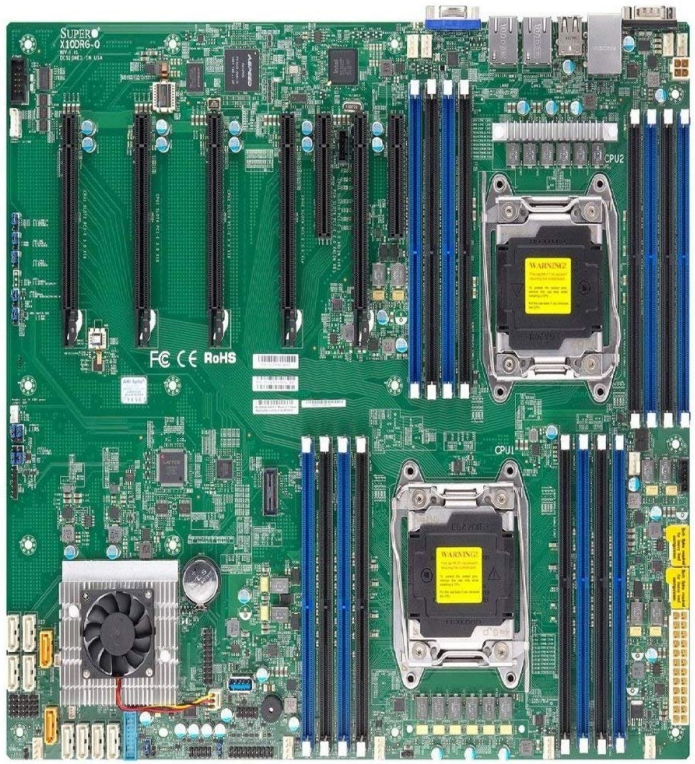


D



E

Questions



The connectivity between CPUs and RAM banks is solved by a network.

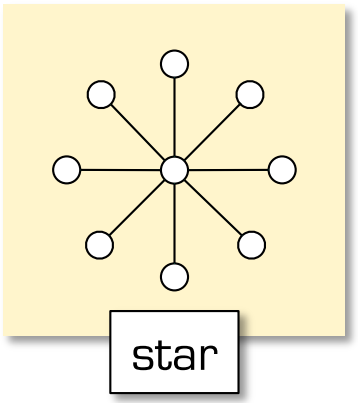
Names of recent technologies are

- *Hypertransport*
- *QuickPath Interconnect*
- *Ultra Path Interconnect*

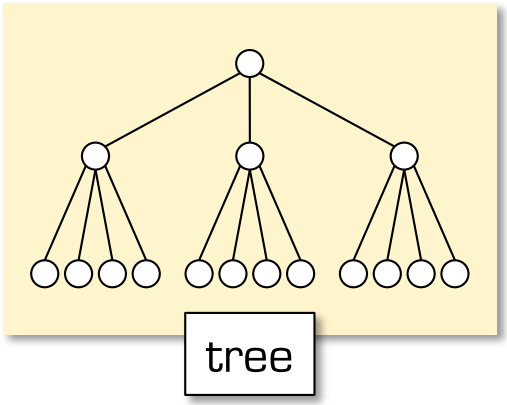
Q: Which topology do computers with several CPUs use for connecting to RAM today?

Questions

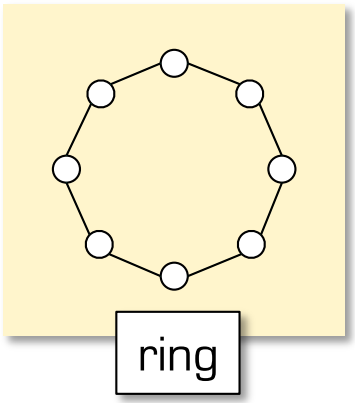
Q: Which topology do computers with several CPUs use for connecting to RAM today?



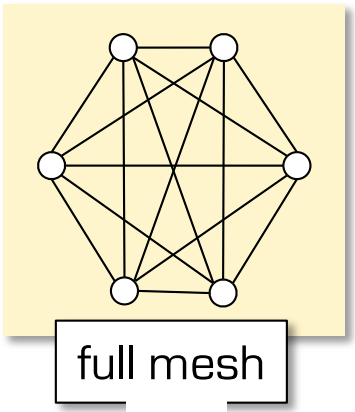
A



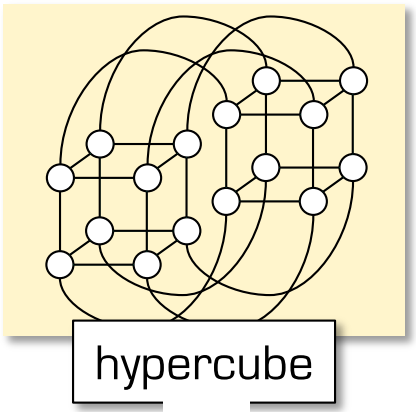
B



C



D



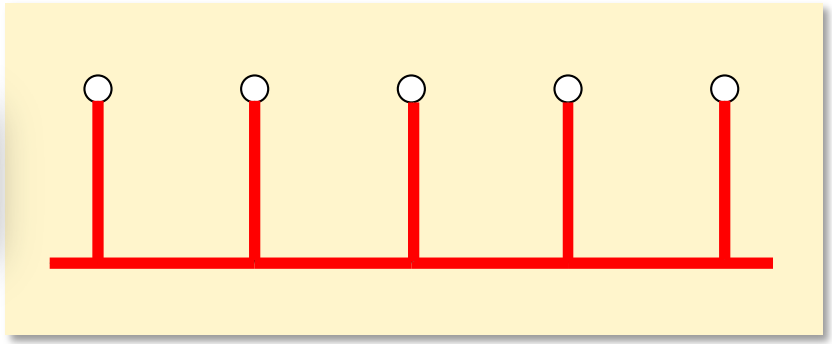
E

Network Structures

Broadcasting channels

- Cable
  - old-fashioned Ethernet
- Radio
  - Aloha (first wireless data transmission)
  - WiFi (IEEE 802.11)
  - mobile: 3G, 4G, 5G
  - satellites
- Properties
  - when one node sends, potentially many nodes can hear it





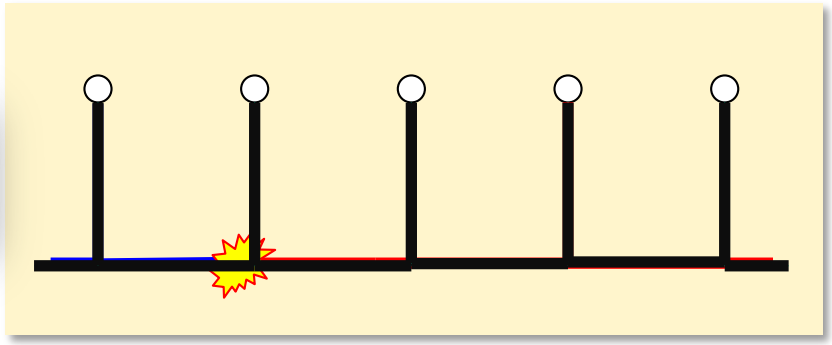


Network Structures

Broadcasting channels

- Cable
  - old-fashioned Ethernet





- Radio
  - Aloha (first wireless data transmission)
  - WiFi (IEEE 802.11)
  - mobile: 3G, 4G, 5G
  - satellites



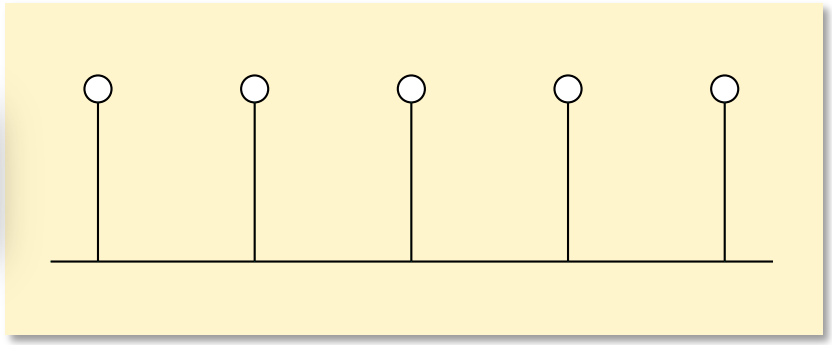
- Properties
  - when one node sends, potentially many nodes can hear it
  - when two nodes send, both messages are potentially ruined
  - error detection is important
  - coordination is desirable

Network Structures

Broadcasting channels

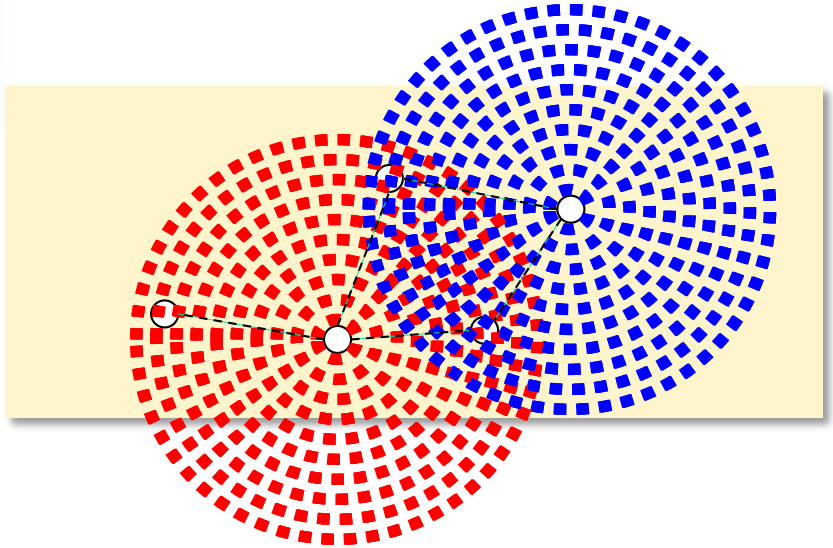
- Cable
  - old-fashioned Ethernet



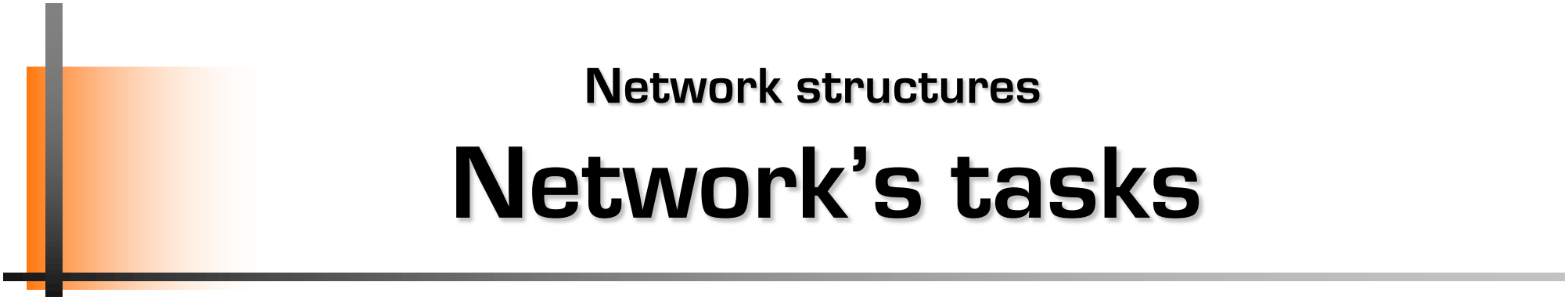


- Radio
  - Aloha (first wireless data transmission)
  - WiFi (IEEE 802.11)
  - mobile: 3G, 4G, 5G
  - satellites





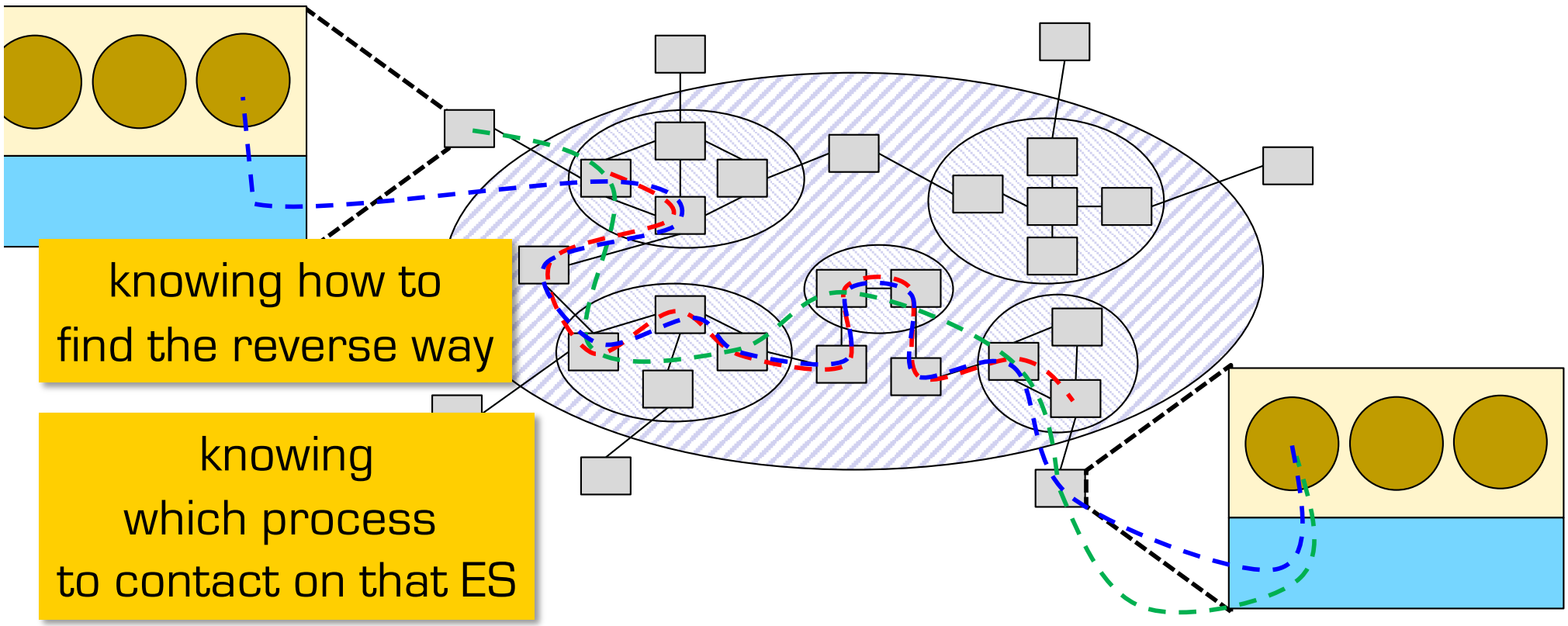
- Properties
  - when one node sends, potentially many nodes can hear it
  - when two nodes send, both messages are potentially ruined
  - error detection is important
  - coordination is desirable



Network structures

Network's tasks

A lot of tasks



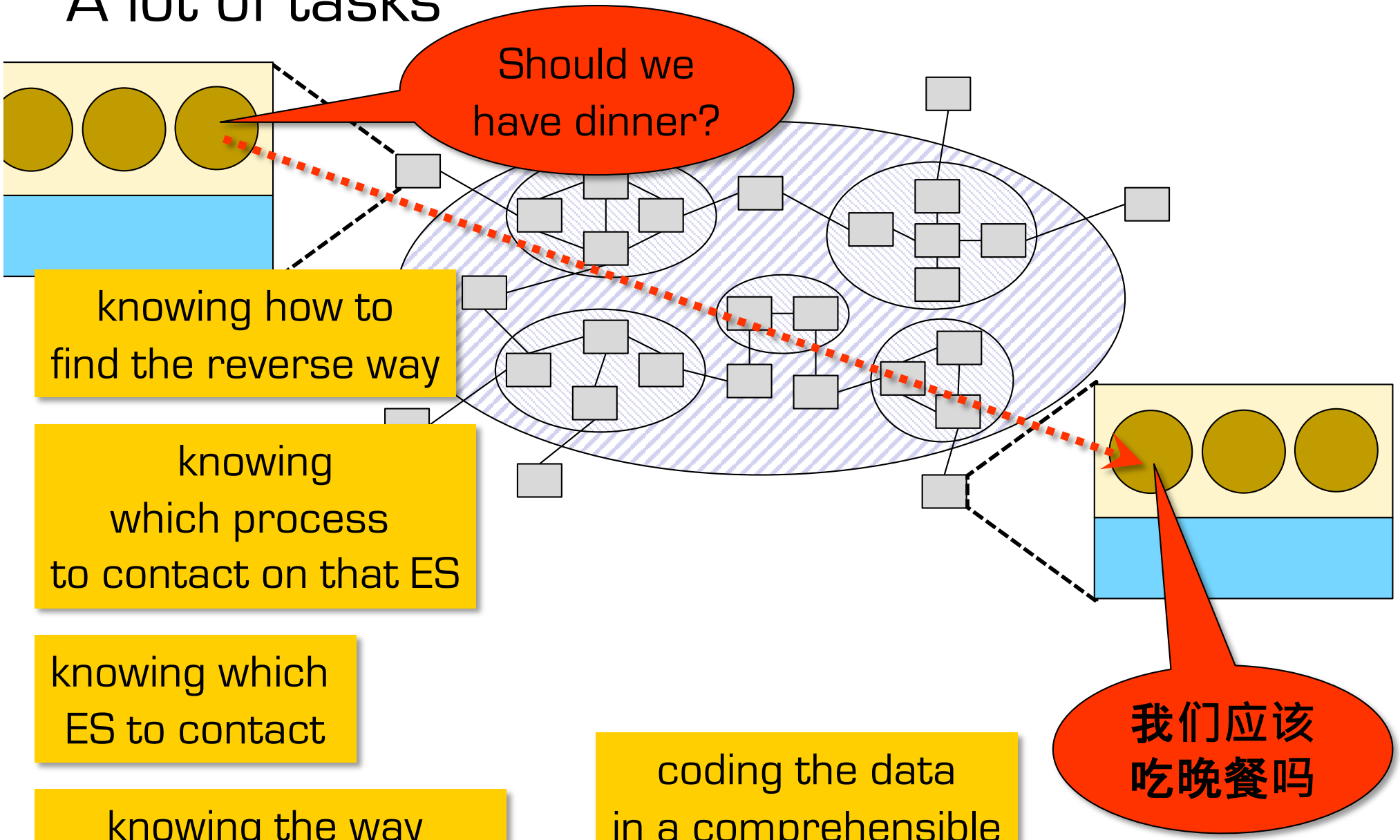
knowing how to find the reverse way

knowing which process to contact on that ES

knowing which ES to contact

knowing the way from one IS to another

A lot of tasks



knowing how to find the reverse way

knowing which process to contact on that ES

knowing which ES to contact

knowing the way from one IS to another

coding the data in a comprehensible manner

A lot of tasks

dealing with networking problems

knowing how to find the reverse way

knowing which process to contact on that ES

knowing which ES to contact

knowing the way from one IS to another

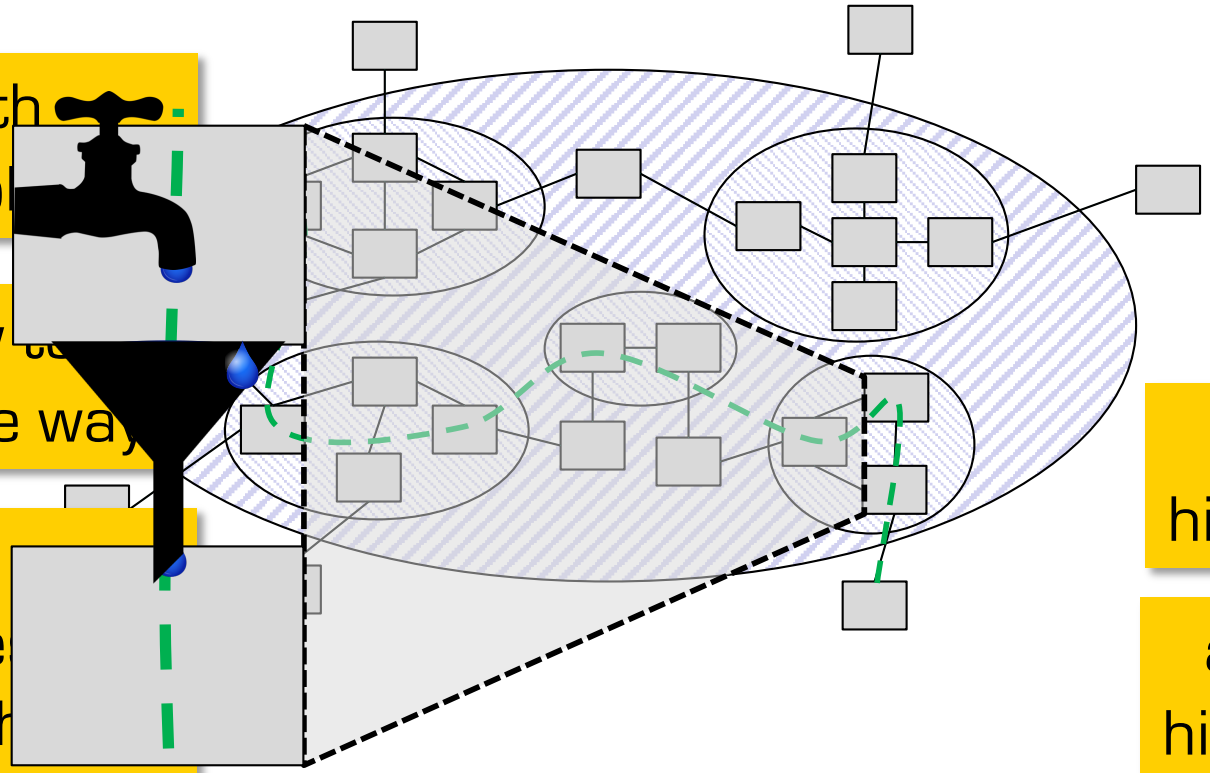


data in a comprehensible manner

maintaining privacy

maintaining security

A lot of tasks



dealing with networking pro

knowing how t find the reverse wa

knowing which proce to contact on th

knowing which ES to contact

knowing the way from one IS to another

coding the data in a comprehensible manner

support high traffic

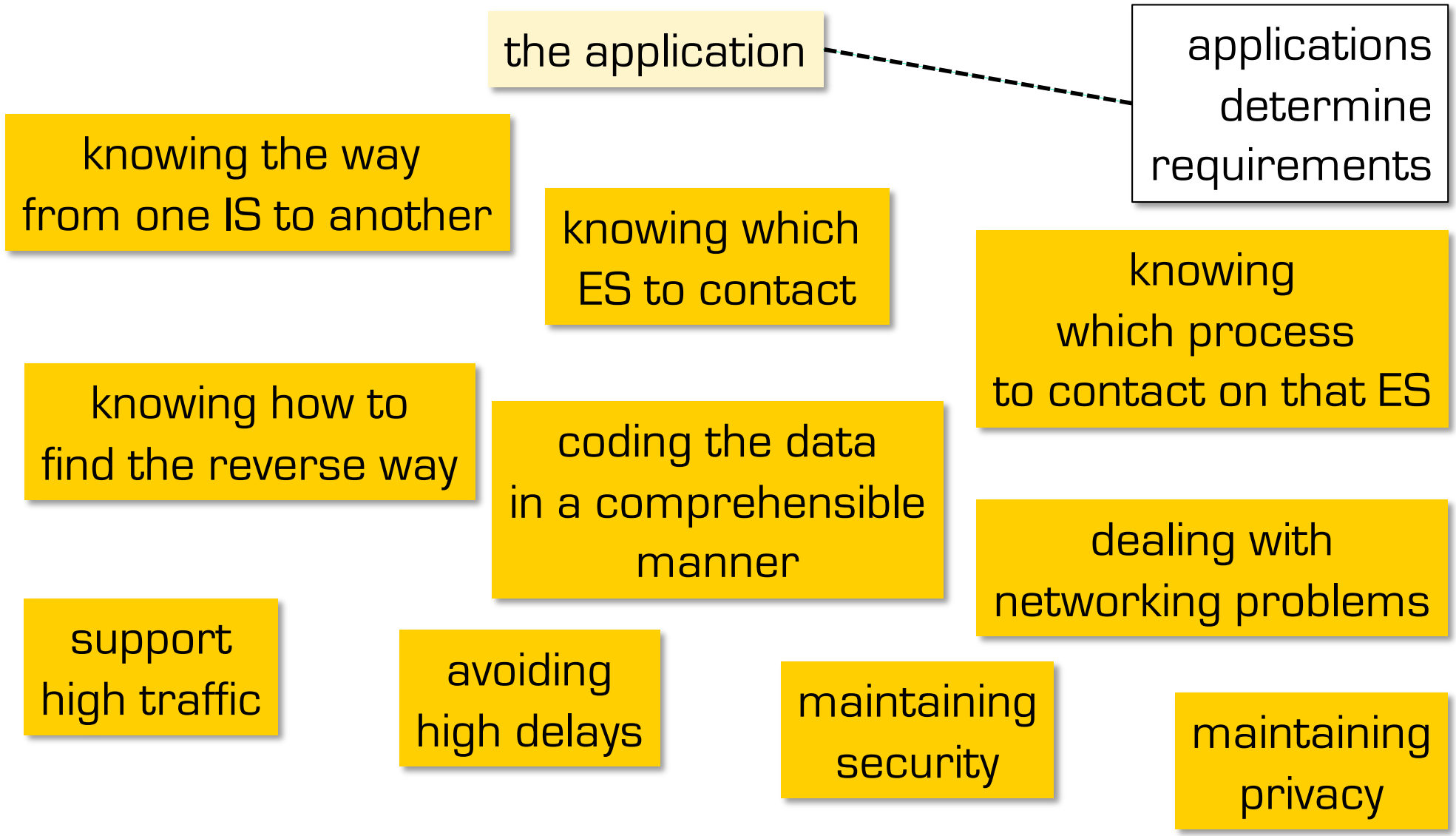
avoiding high delays

maintaining privacy

maintaining security

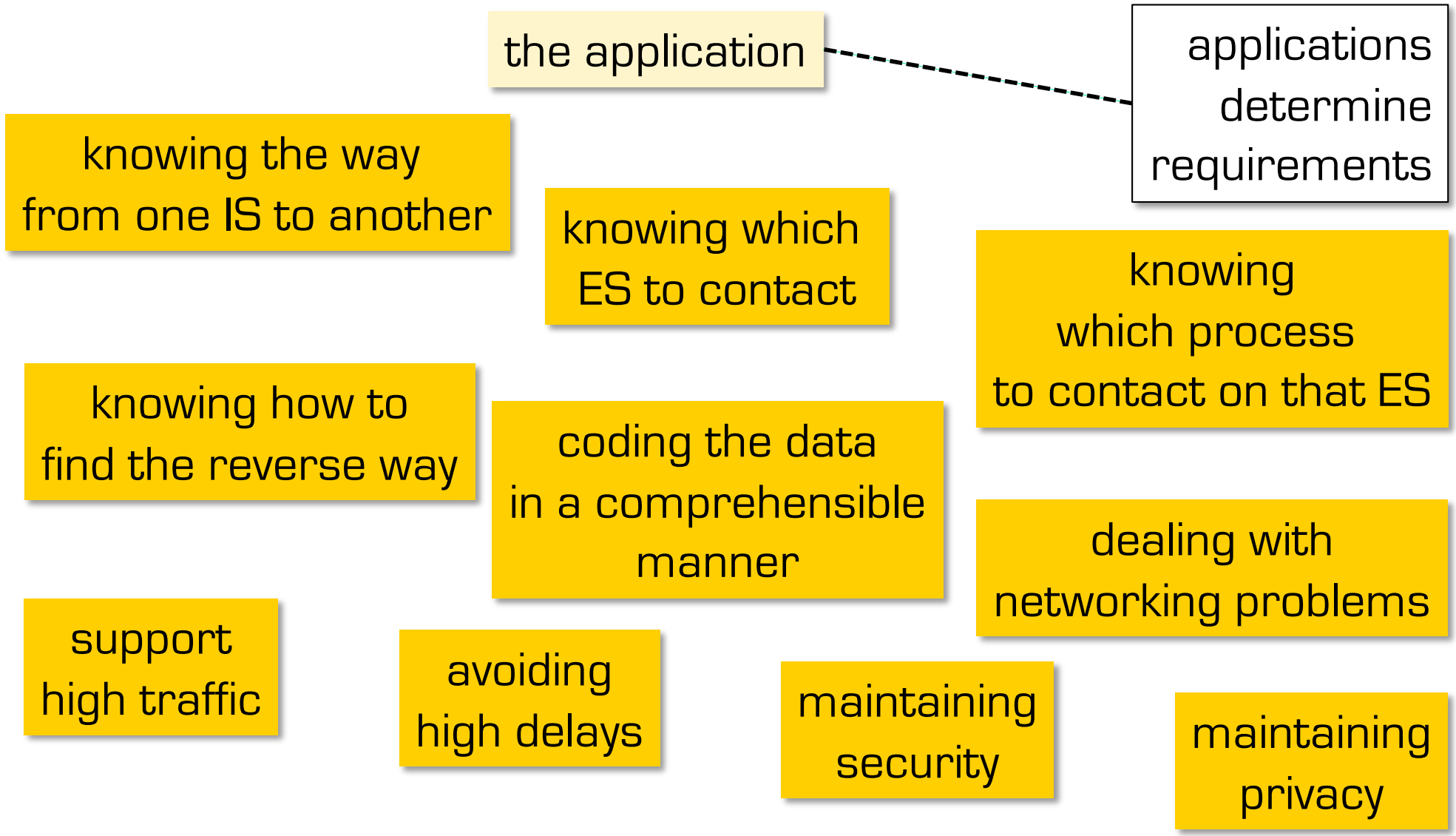
A lot of tasks

There are a lot of aspects to worry about



A lot of tasks

There are a lot of aspects to worry about



the application

applications determine requirements

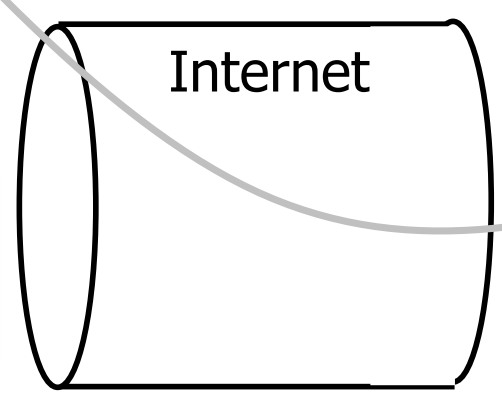
dealing with networking problems

support high traffic

avoiding high delays

The application and Web browsing

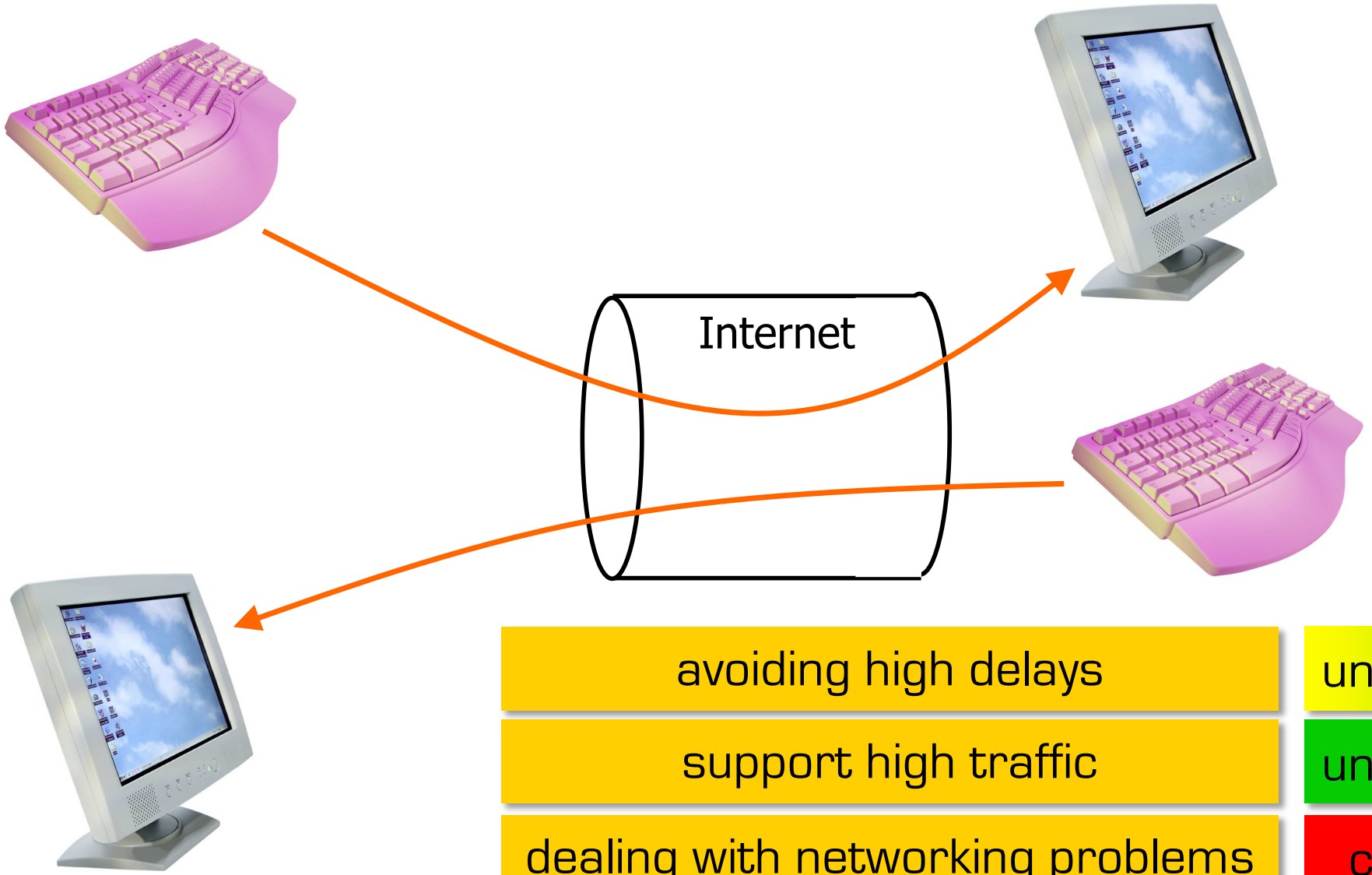






avoiding high delays	uncritical
support high traffic	uncritical
dealing with networking problems	critical

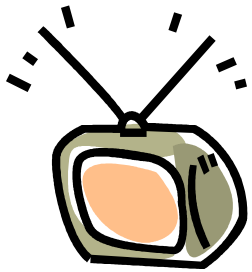
Textual commands and textual chat

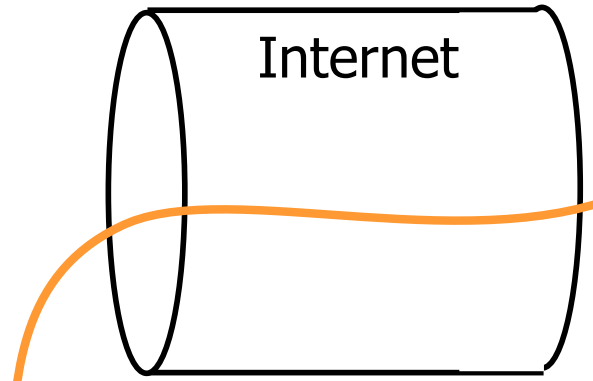


avoiding high delays	uncritical
support high traffic	uncritical
dealing with networking problems	critical

Live and on-Demand Streaming

avoiding high delays	some ok
support high traffic	critical
dealing with networking problems	some ok

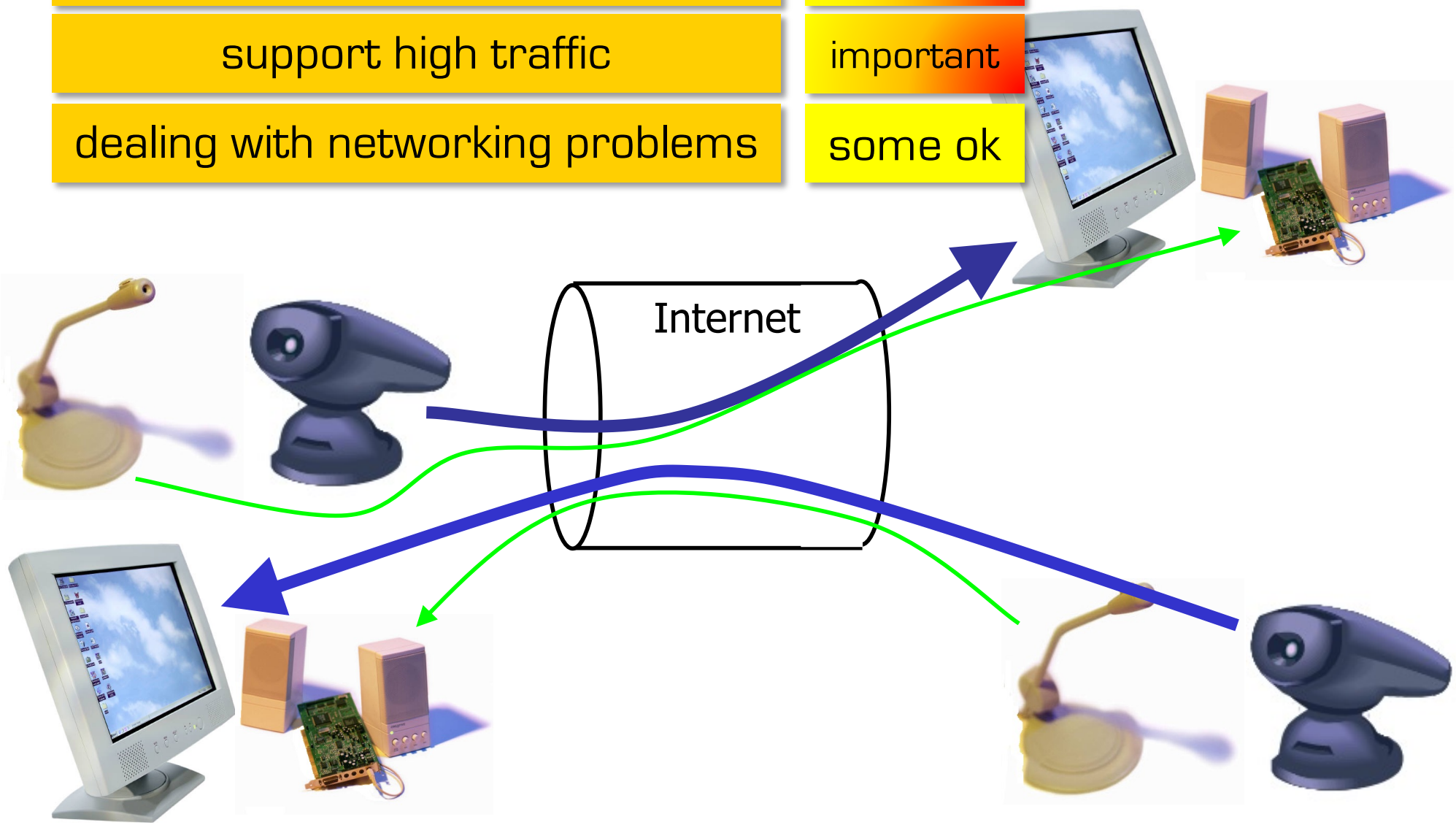




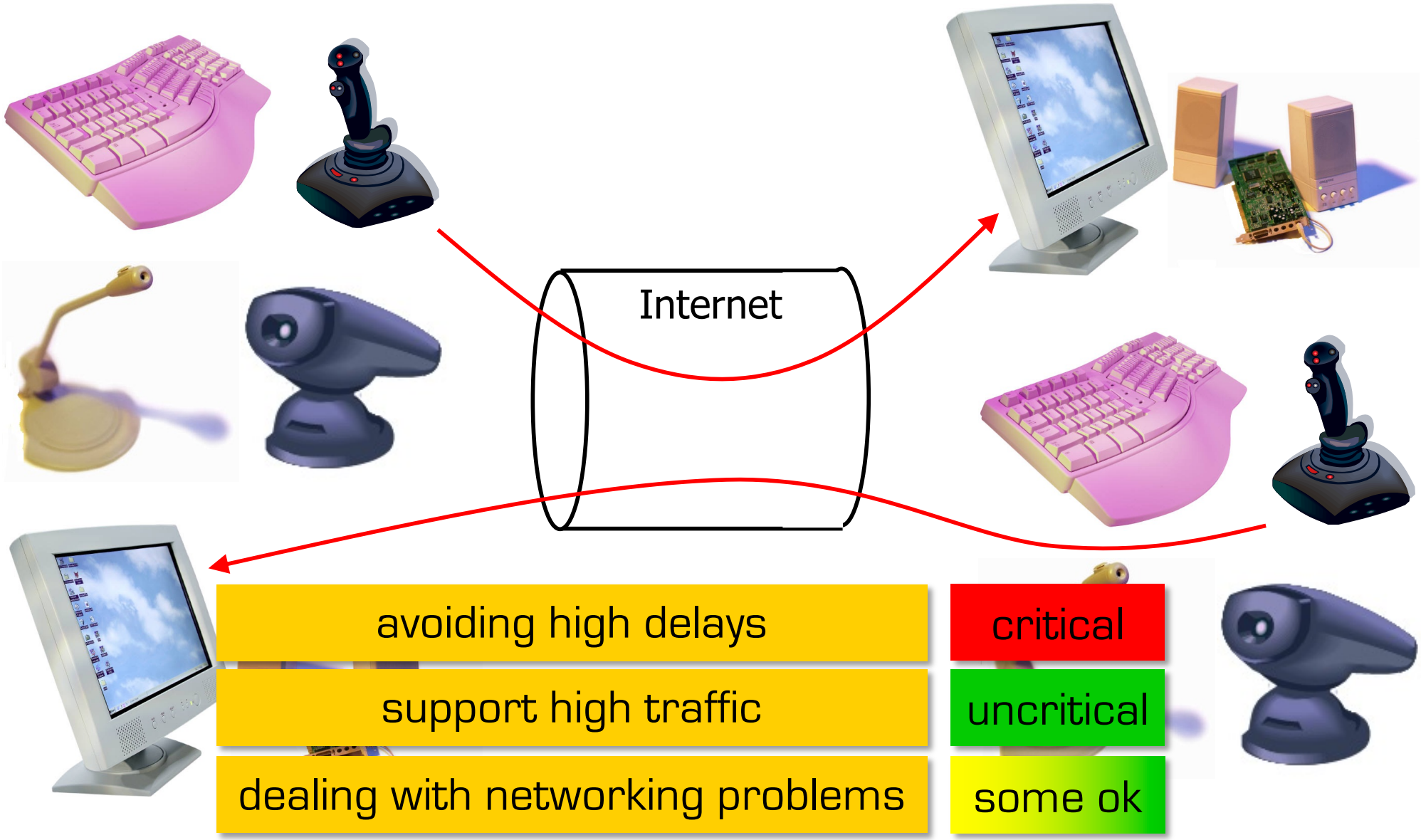


AV chat and AV conferencing

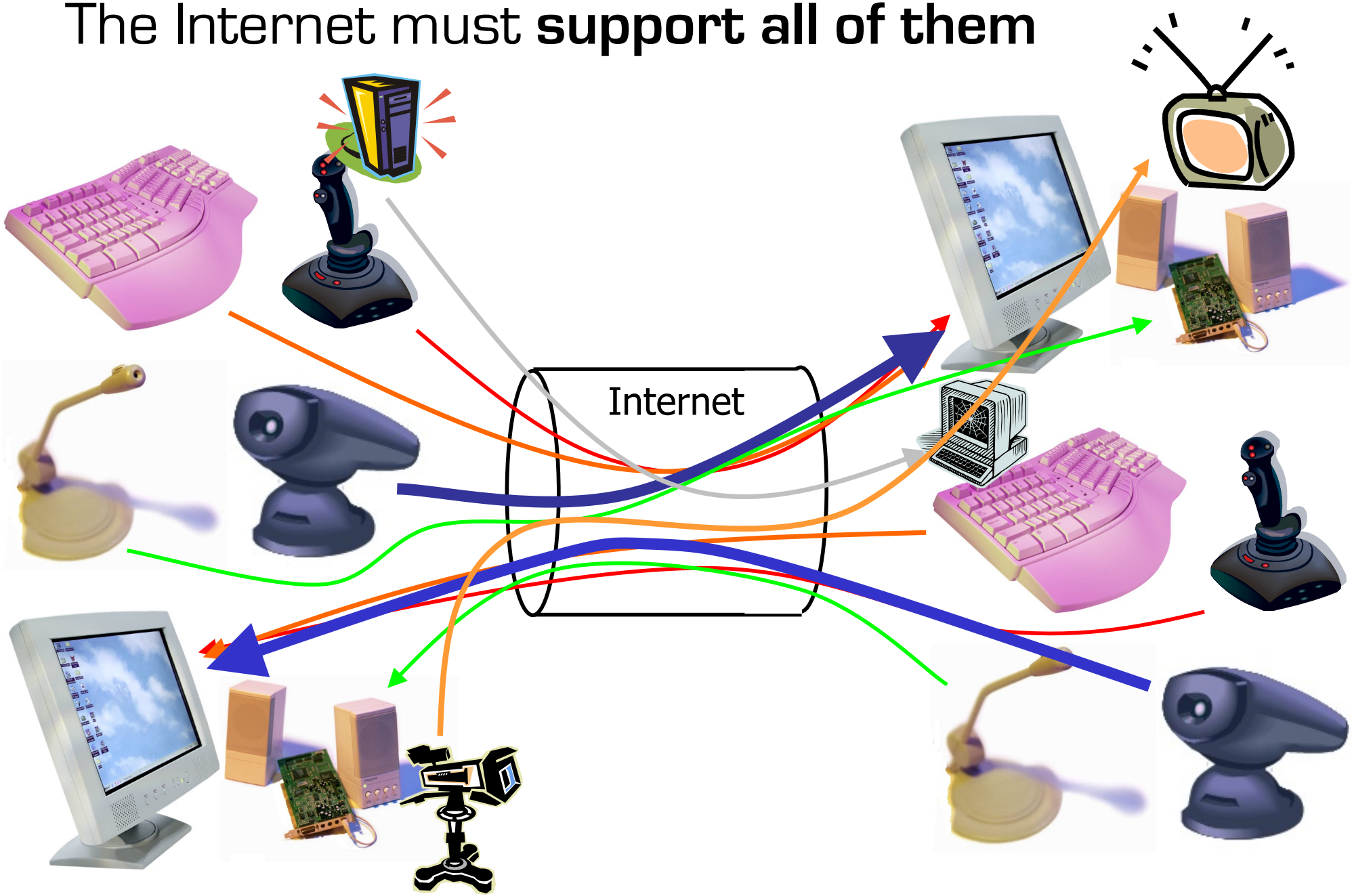
avoiding high delays	some ok
support high traffic	important
dealing with networking problems	some ok



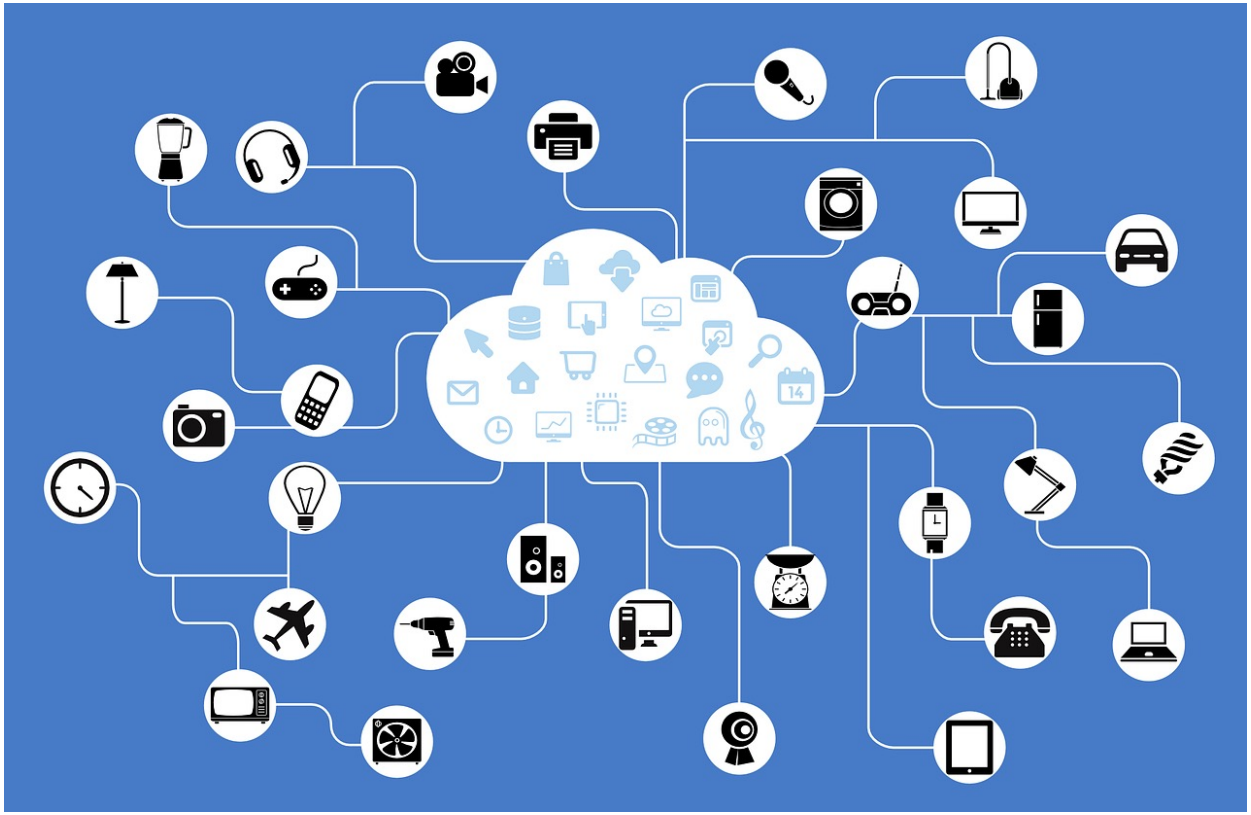
Haptic Interaction



The Internet must **support all of them**



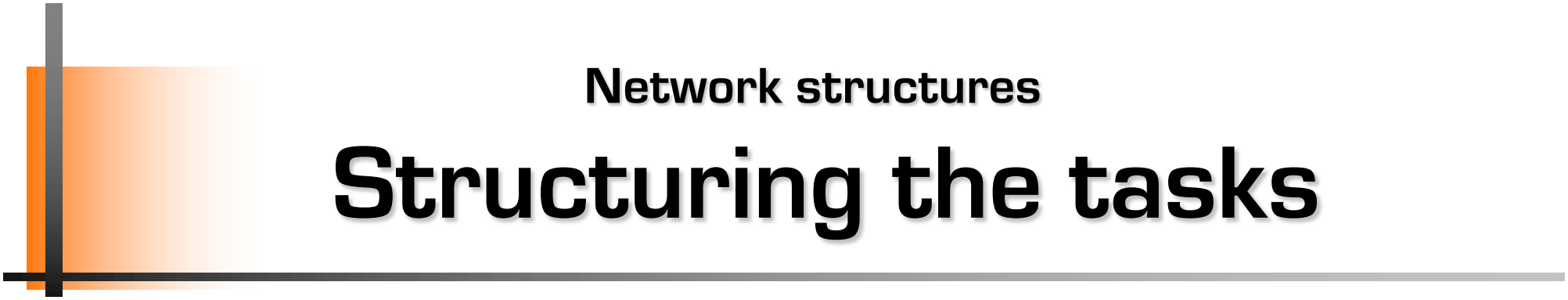
Questions



The previous slides showed some examples of applications that must be supported by the Internet. But some really important ones are missing.

Q: Can you give an example of an important kind of application that was missing?

IN2140:
Introduction to Operating Systems and Data Communication

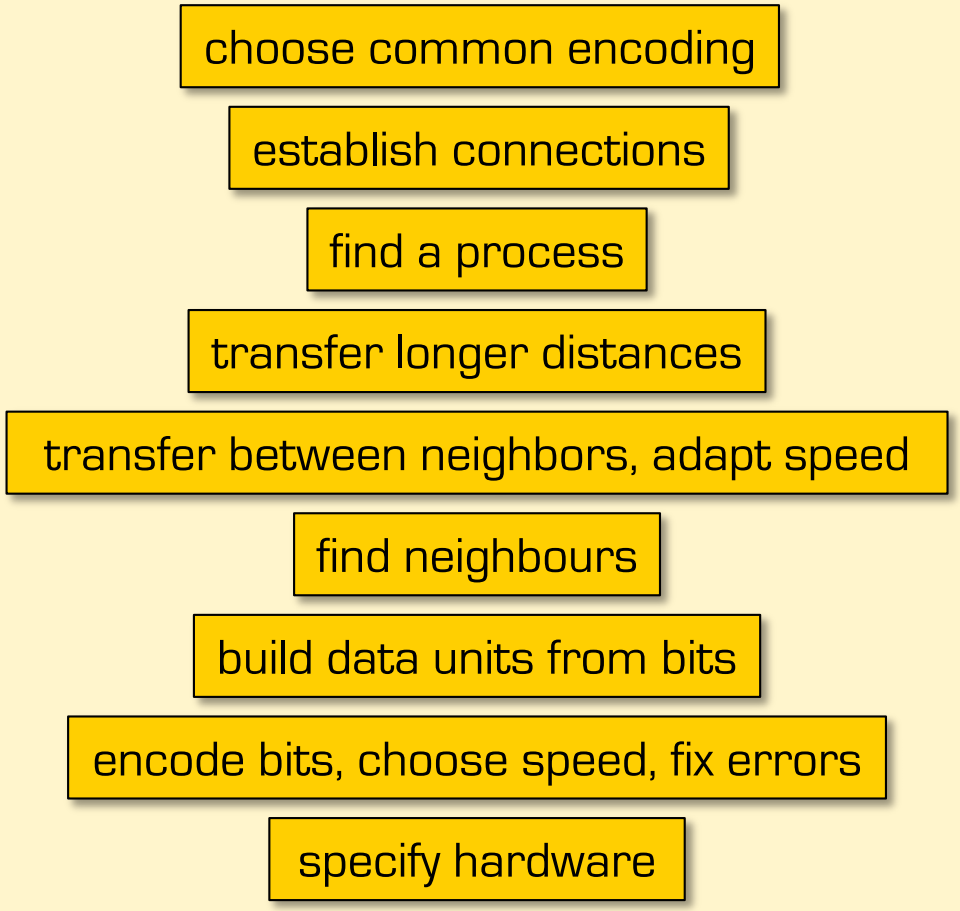


Network structures

Structuring the tasks

Approaches to structure the tasks

layered approach: arrange tasks in layers



advantages:

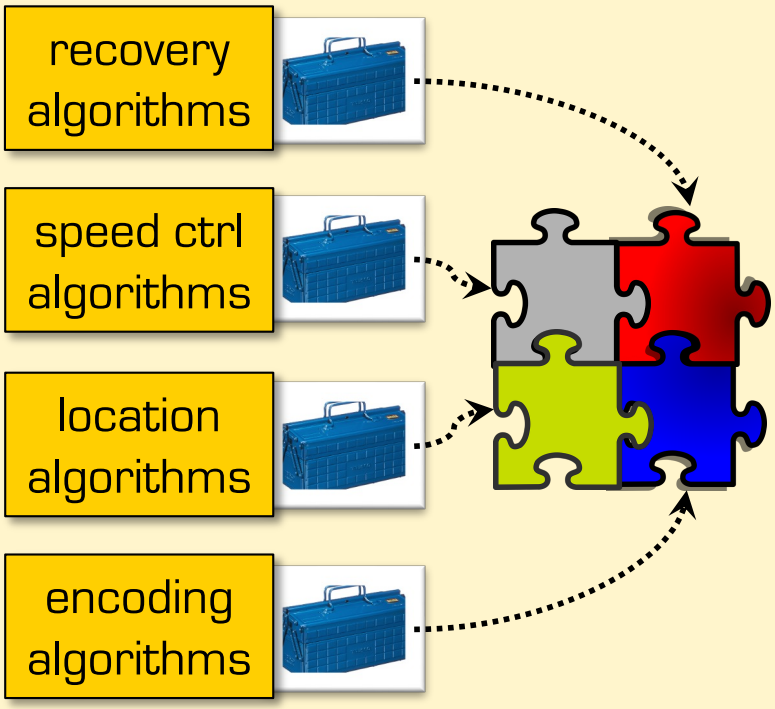
- clear interfaces
- clear assignment of responsibilities
- develop layers independently

disadvantages:

- not perfectly suited for all jobs
- similar problems are solved several times

Approaches to structure the tasks

component approach: interacting components



advantages:

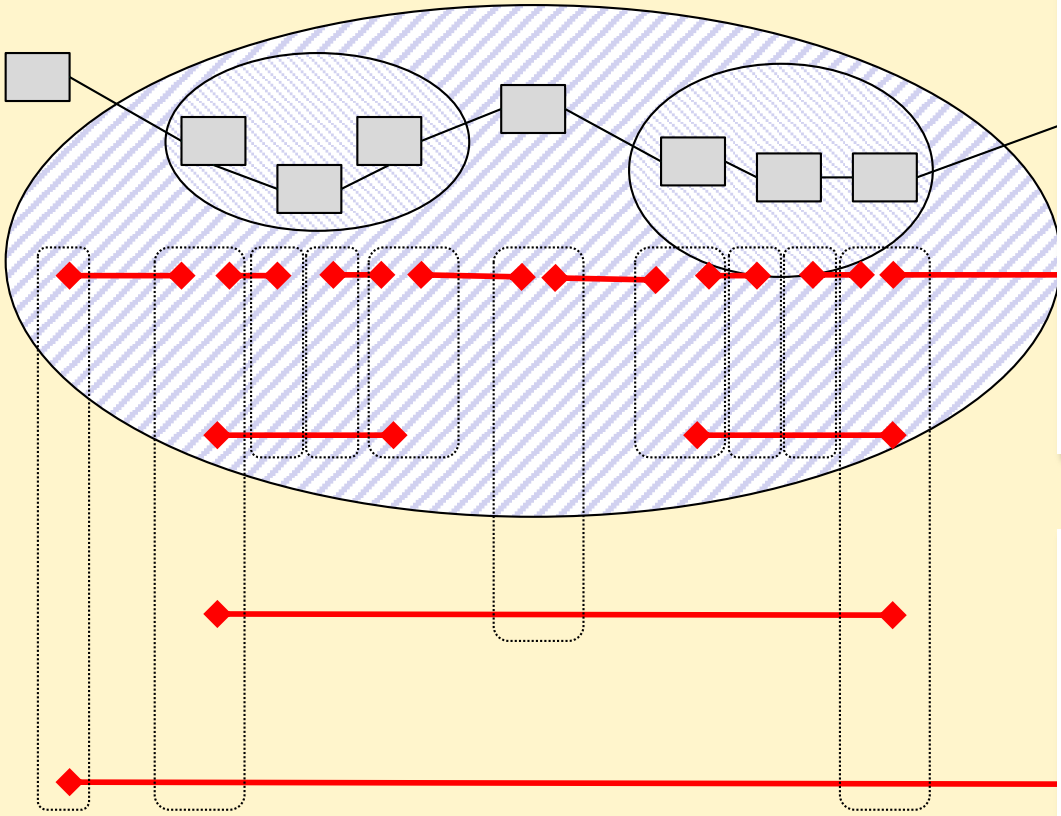
- possible to avoid duplicated functions
- possible to choose perfect network behaviour for every application

disadvantages:

- must negotiate choice of every piece with all nodes
- toolbox must be complete on all nodes
- needs flexible interfaces

Approaches to structure the tasks

recursive approach: handle challenges locally



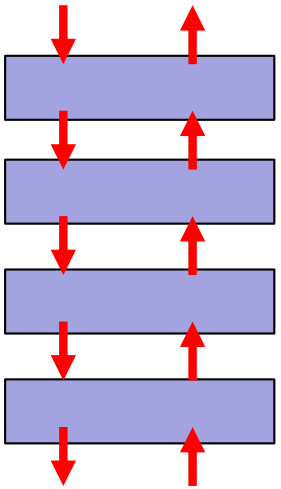
advantages:

- reuse the concept of inter-process communication on all levels
- concepts are repeated at every level
- all challenges can be solved as local as possible

disadvantages:

- more negotiations and setup than layered
- unclear how to best share resources

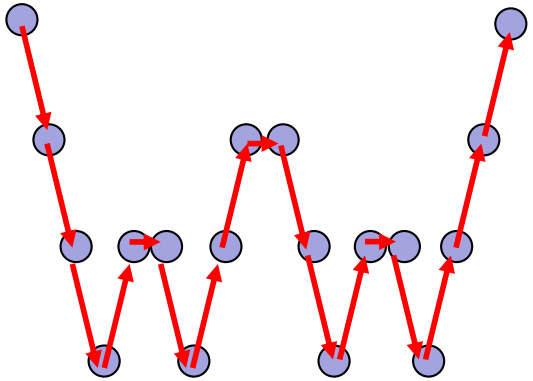
Questions



Layers



Components

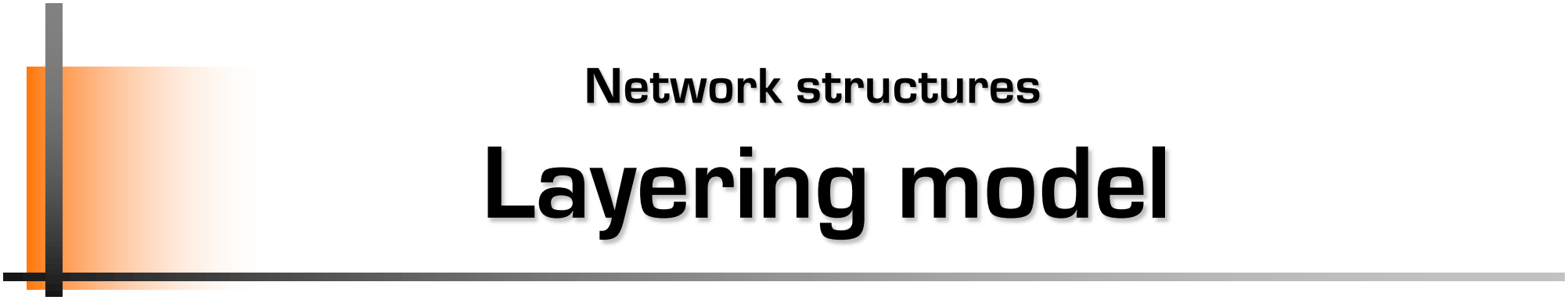


Recursion

These three ways of structuring the tasks of networks are frequently discussed, as well as used. The reality is a mixture, but one of them dominates how we think about task structuring.

Q: Which of these three approaches is the dominant one?

IN2140:
Introduction to Operating Systems and Data Communication



Network structures

Layering model

Approaches to structure the tasks

Right now, the layered models dominate

ISO OSI (Open Systems Interconnection) Reference Model

and

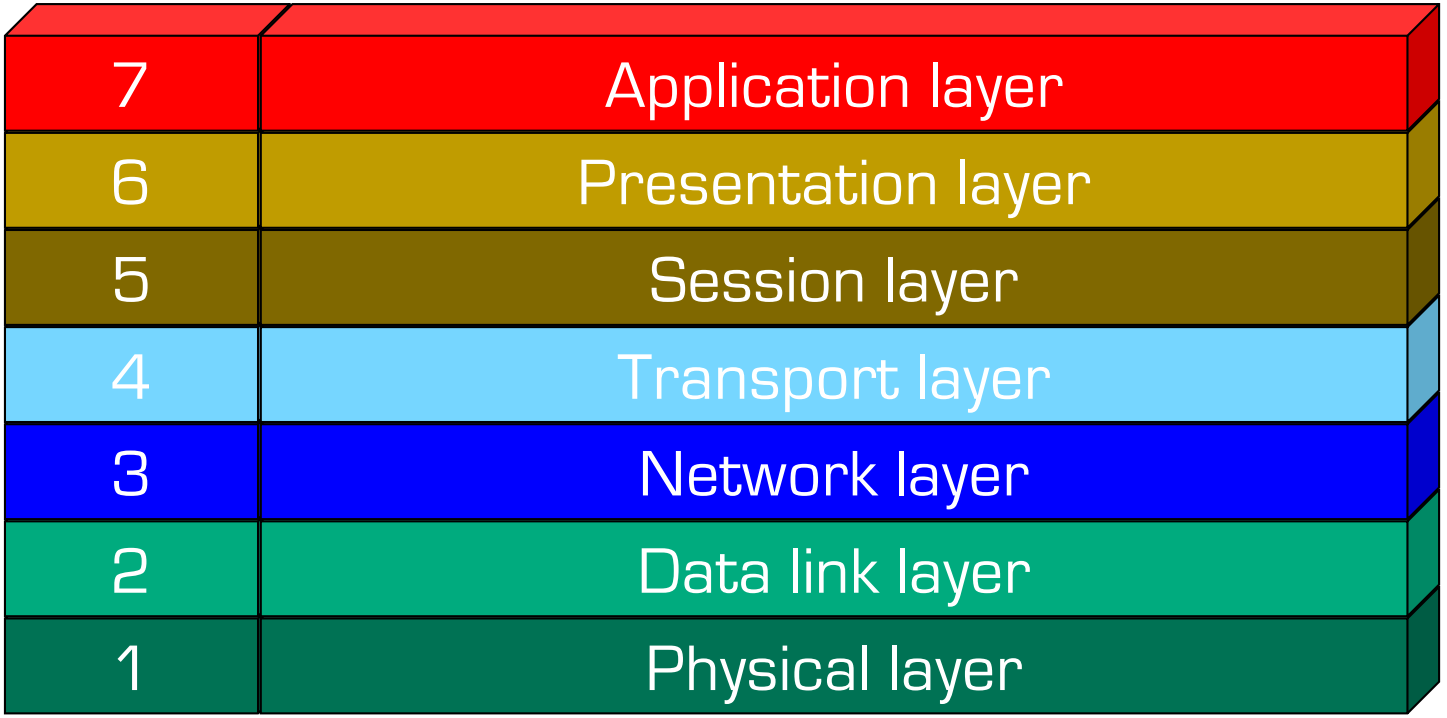
TCP/IP Reference Model Internet Architecture

- layers are easy to understand
- interfaces are clearly defined
- not every node must implement every layer
- ...

Reference Model for Open Systems Interconnection

ISO OSI (Open Systems Interconnection) Reference Model

- model for layered communication systems
- defines fundamental concepts and terminology
- defines 7 layers and their functionalities

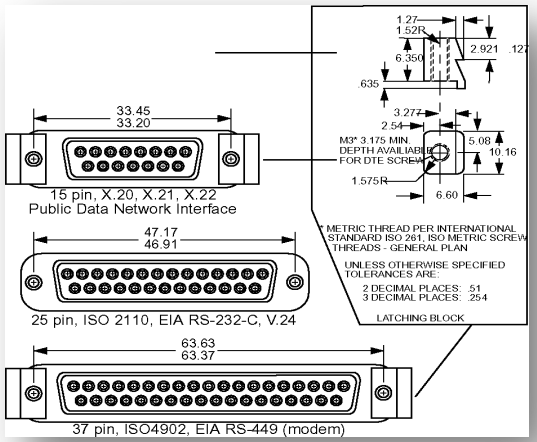


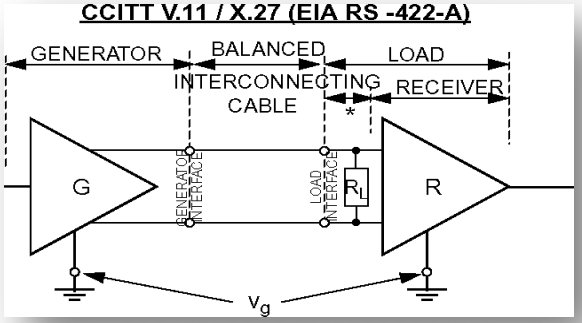
Layer functions: physical layer

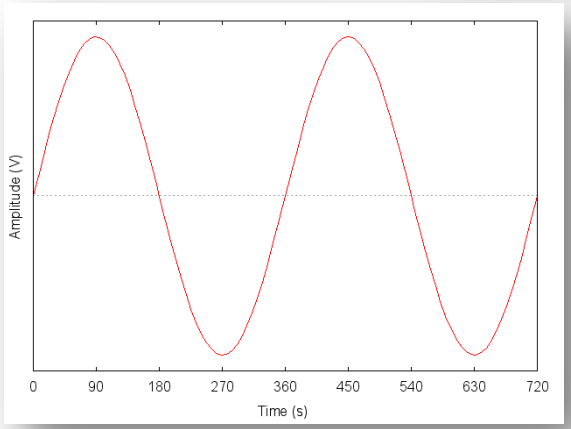
Responsibility: insecure bitstream between adjacent systems

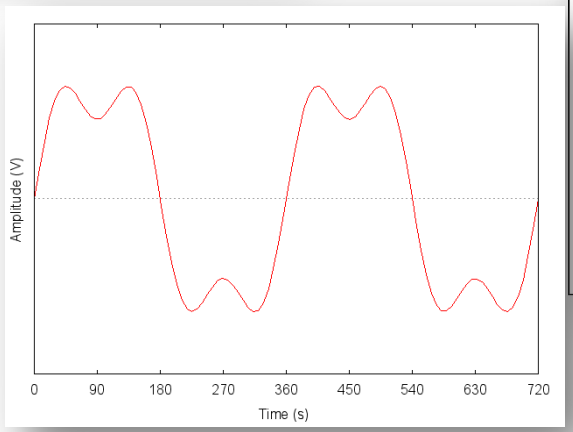
- mechanics
- electronics
- procedural

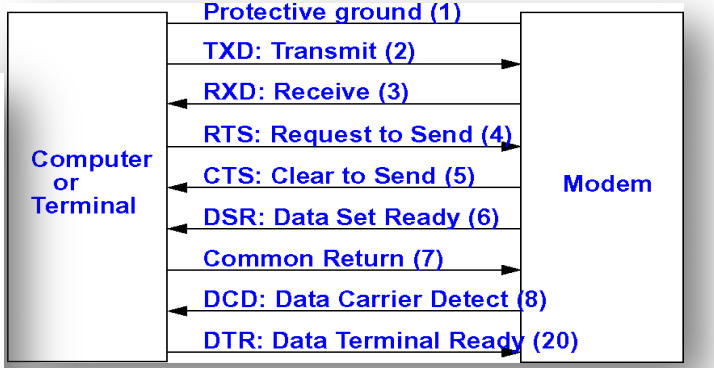
encoding and decoding of bits









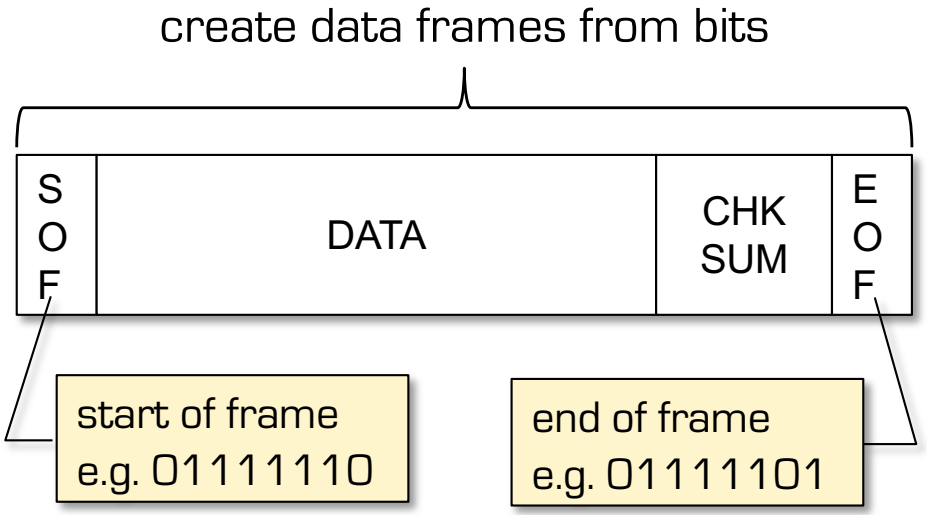


1 Physical layer

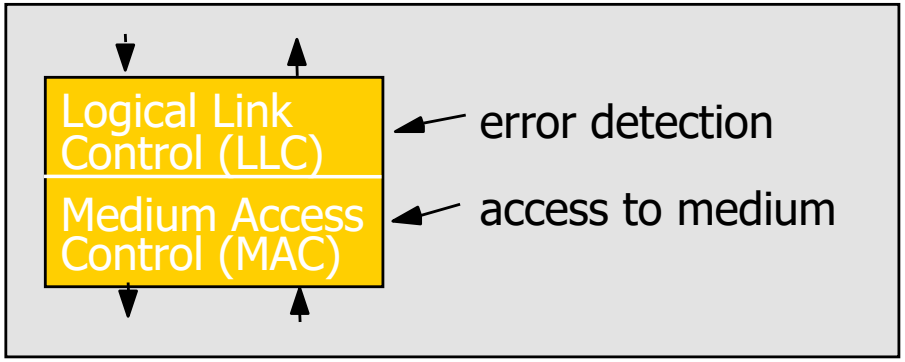
Layer functions: data link layer

Responsibility: error-recovering frame stream, adjacent systems

Reliable data transfer between adjacent stations with frames



handle speed differences between nodes
handle several L3 protocols
translate between different MAC layers

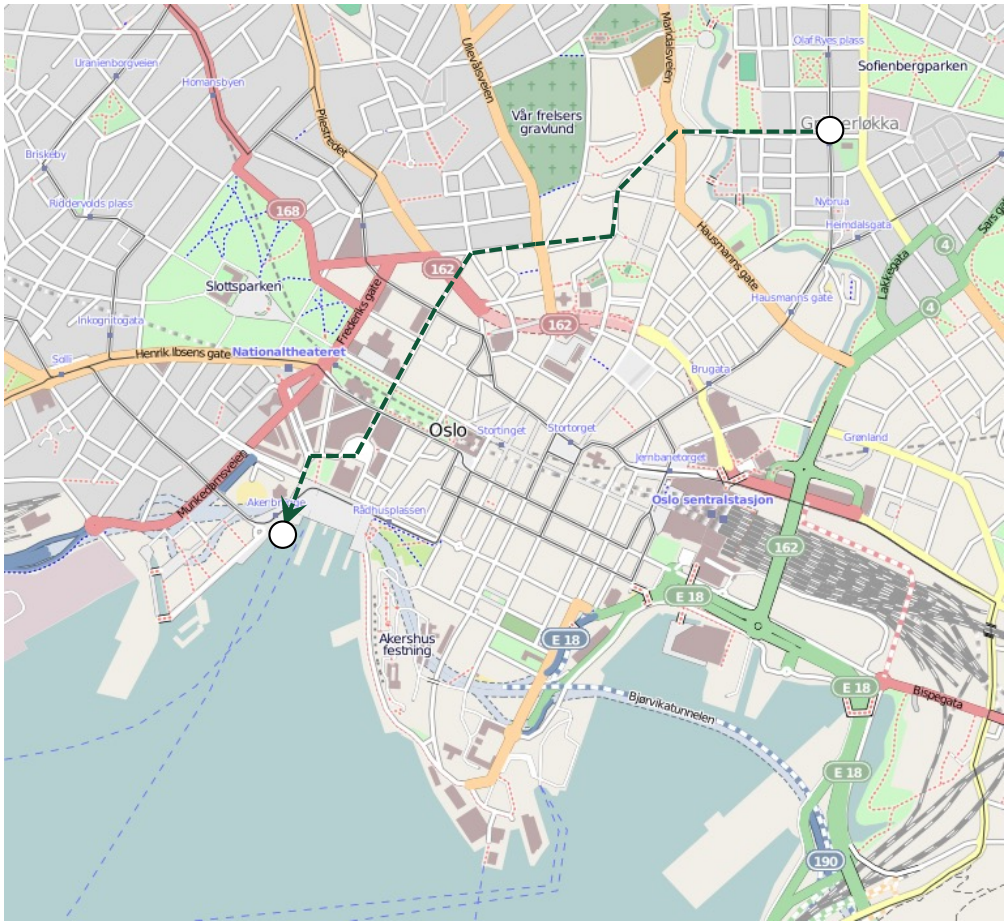


where to send? → MAC address
when to send? → wait for silence, wait for token
correct frame? → check for bit errors and repair



Layer functions: network layer

Responsibility: packet stream between end systems



end-to-end transport of packets

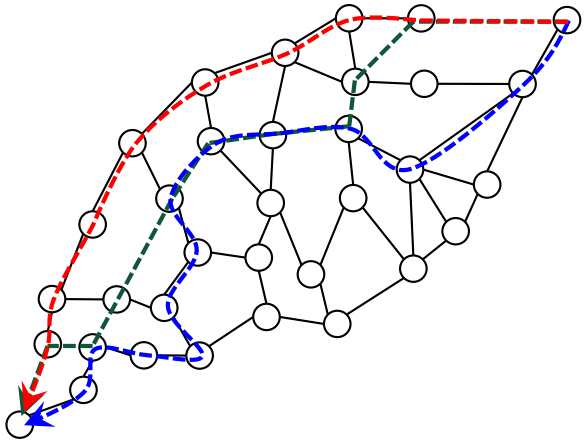
ability to address source and target nodes

routing from source node to target node

3 Network layer

Layer functions: network layer

Responsibility: packet stream between end systems

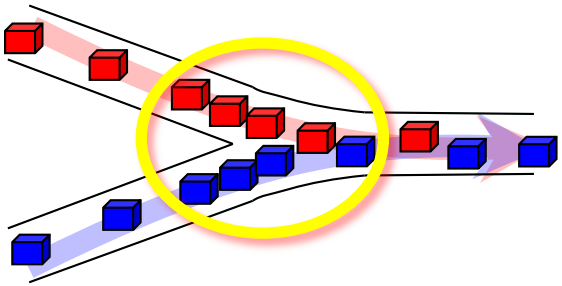


end-to-end transport of packets

ability to address source and target nodes

routing from source node to target node

- find routes
- choose between alternative routes
- determine best packet size for a route
- translate addresses
- prevent or handle congestion
- multiplexing of L4 packets



3 Network layer

Layer functions: transport layer

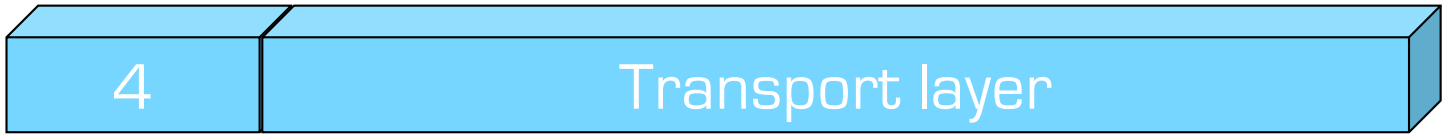
Responsibility: end-to-end message stream between processes

wide range of services offered to L5:	
connection establishment	no connection
hide L3 packet size limitations	limit L4 packet size to L3 packet's
error correction	no error correction
ensure order of received packets	deliver packets as the arrive
adapt sending speed to receiver	send as fast as possible

end-to-end transport of packets

ability to address source and target processes

- establish process-to-process relation
- multiplex traffic
- end-to-end flow control (handles speed differences)
- end-to-end error correction



Layer functions: session layer

Responsibility: structured dialogue

support a "session" over a longer period

session management
- establishing identities
- assigning writes
- tracking identities (cookies)

checkpointing
- make program snapshots to disk
- restart after crash

synchronization
- lip synchronization of speech and video in tele-conferencing
- show live football concurrently on all devices

token management
- passing permission to speak in a (large) tele-conference
- write-locking of networked files
- transaction management in databases

Google OT (operation transformation) allows Google Docs to work
- user identify when multiple devices are used
- several inputs on the same document at the same time
- conflict resolution when writing to the same location



Layer function: presentation layer

Responsibility: exchange of data (semantics!)

encoding of int
- big endian
- little endian
- XML (as string)
- ASN.1 (shortest possible big endian bit sequence)
- XDR (4-byte big endian)

encoding of structs ("serialization")
- ASN.1
- XDR
- XML
- JSON
- Java serialization
- Google protocol buffers

file name **representation**
- `/mnt/user/n.txt`
- `m:\user\n.txt`

image **formats**
- JPG, PNG

compression
- zip, gzip, bzip2

encryption methods
- PGP, S/MIME

semantics
- NOK, EUR, USD

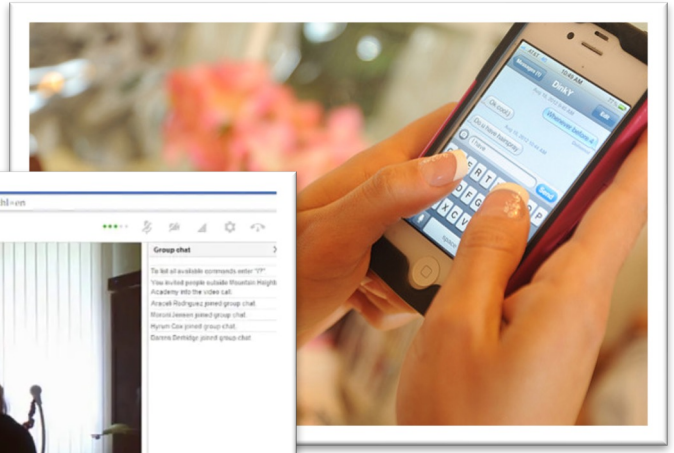
encoding of strings
- ASCII
- UTF-8
- Unicode
- EBCDIC

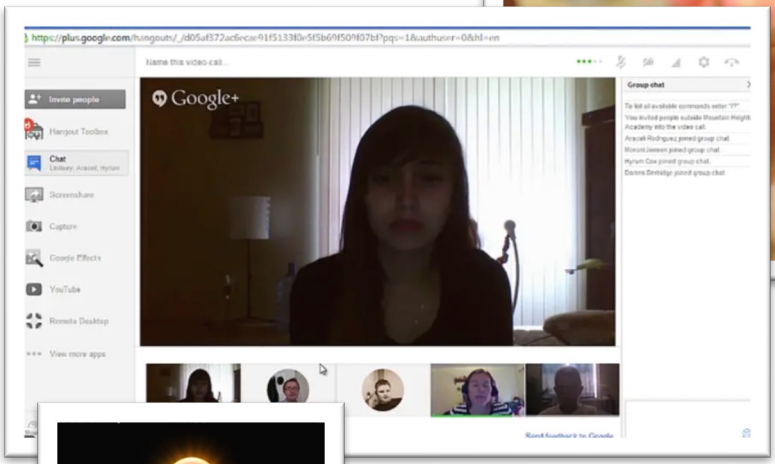
encoding of date
- seconds since 1.1.1970
- nanoseconds since 1.1.1601
- string "12 March 2019 13:32:54 UTC"

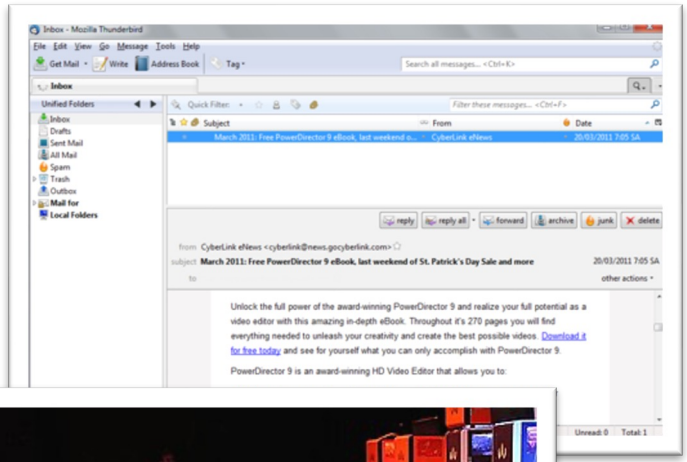


Layer functions: application layer

Responsibility: cooperating entities









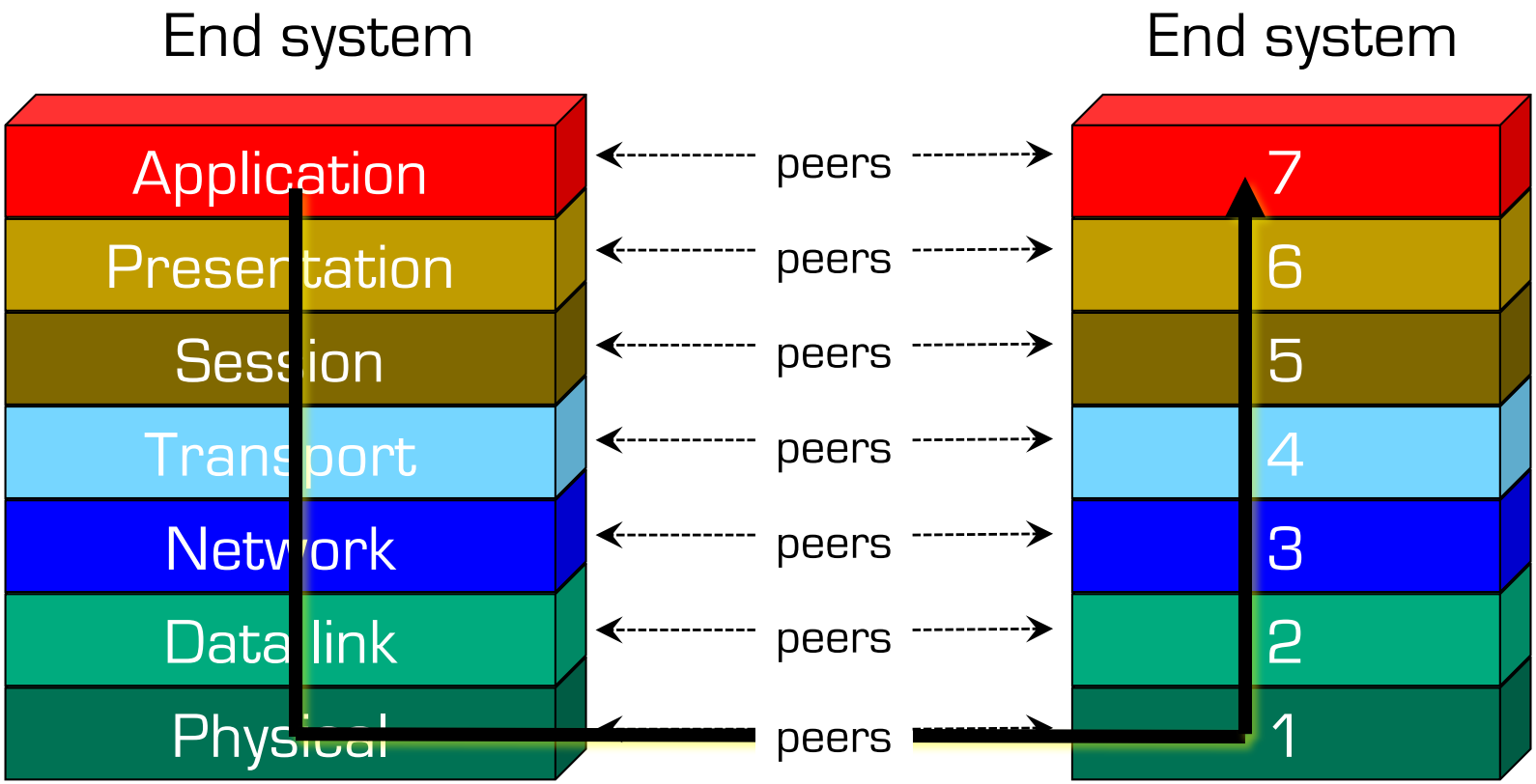




7 Application layer

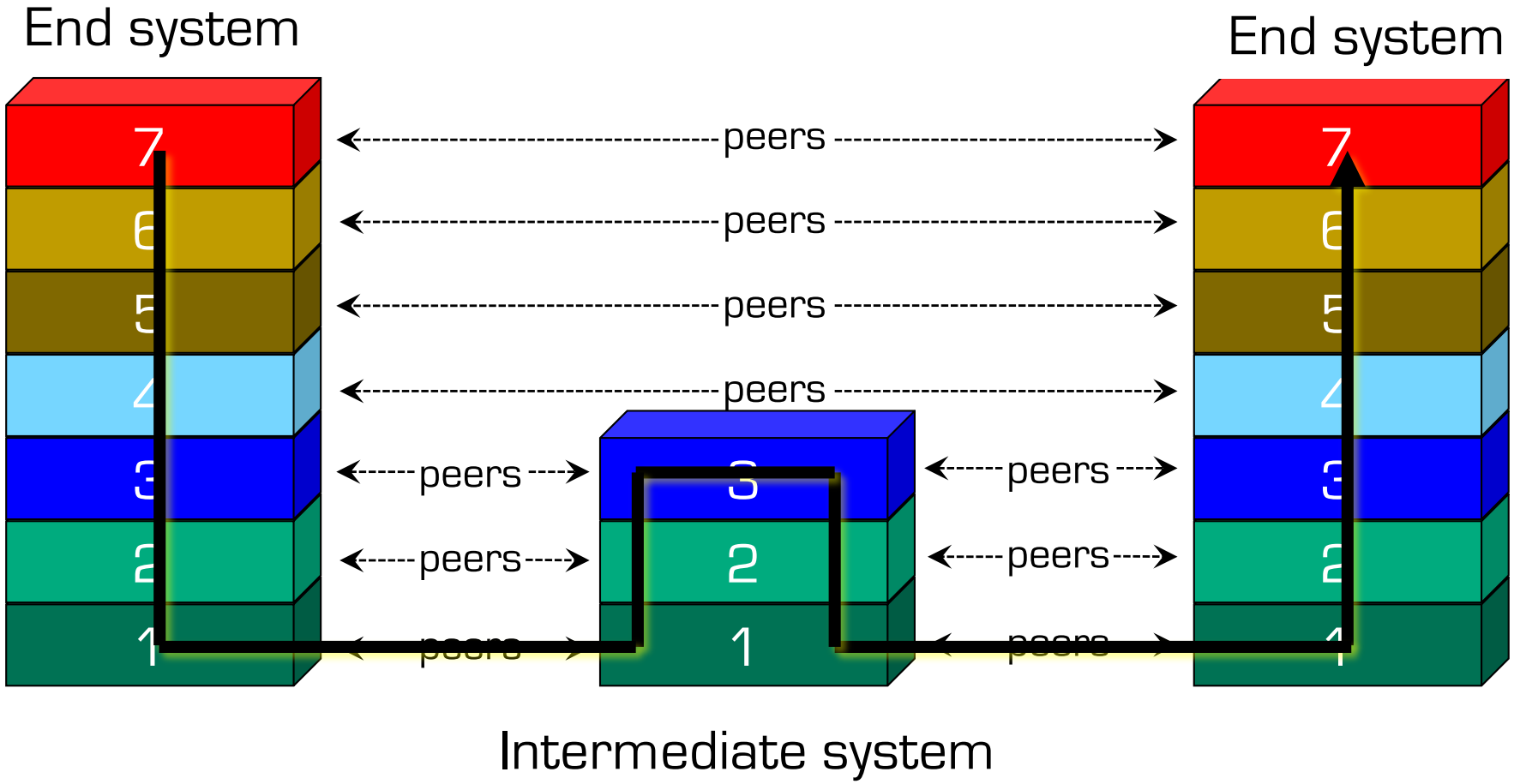
Architecture

Data flow between two adjacent systems

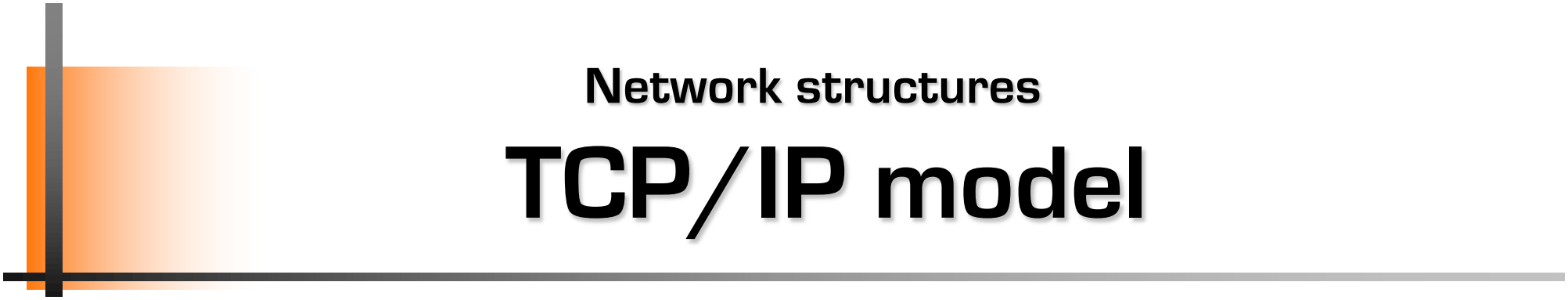


Architecture

Data flow between two non-adjacent systems



IN2140:
Introduction to Operating Systems and Data Communication

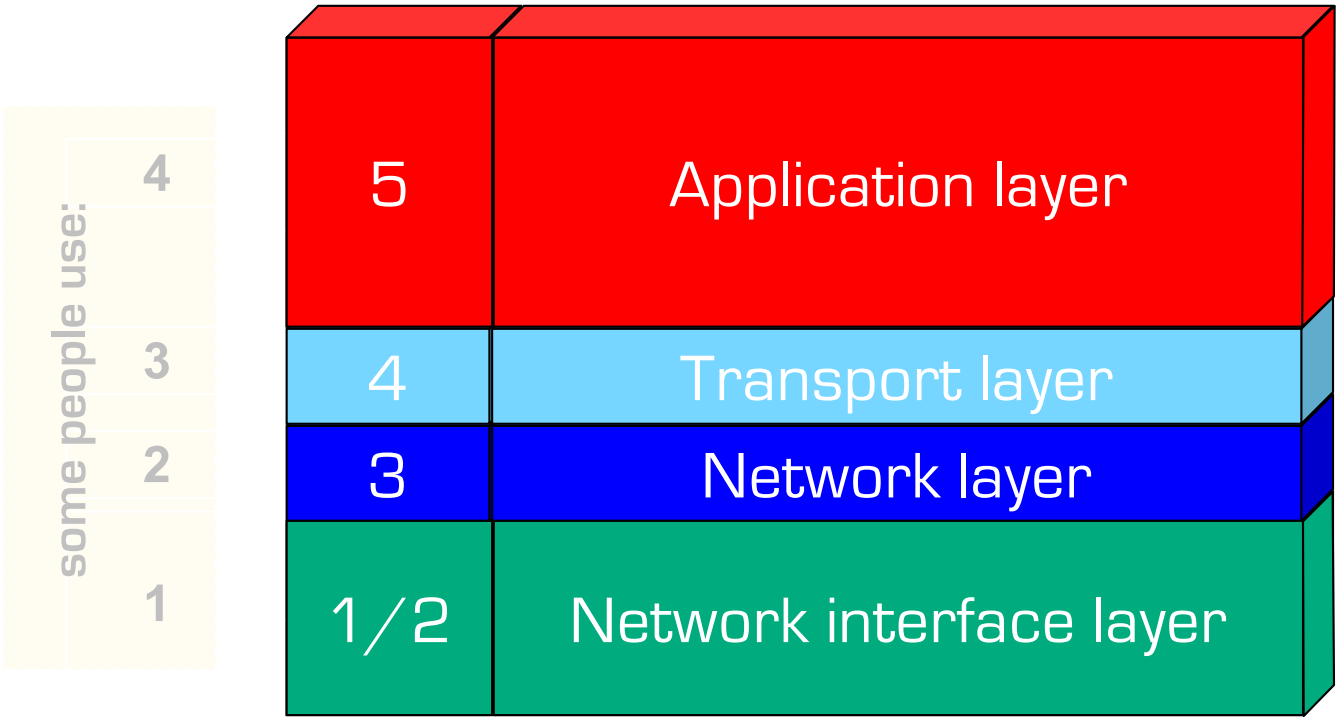


Network structures

TCP/IP model

Five Layer Reference, Internet Reference Model and a Comparison

OSI Reference Model



TCP/IP Reference Model Internet Architecture

- ISO-OSI presentation, session and application layer merged
- ISO-OSI data link layer and physical layer merged to form Network Interface

TCP/IP Layering considerations

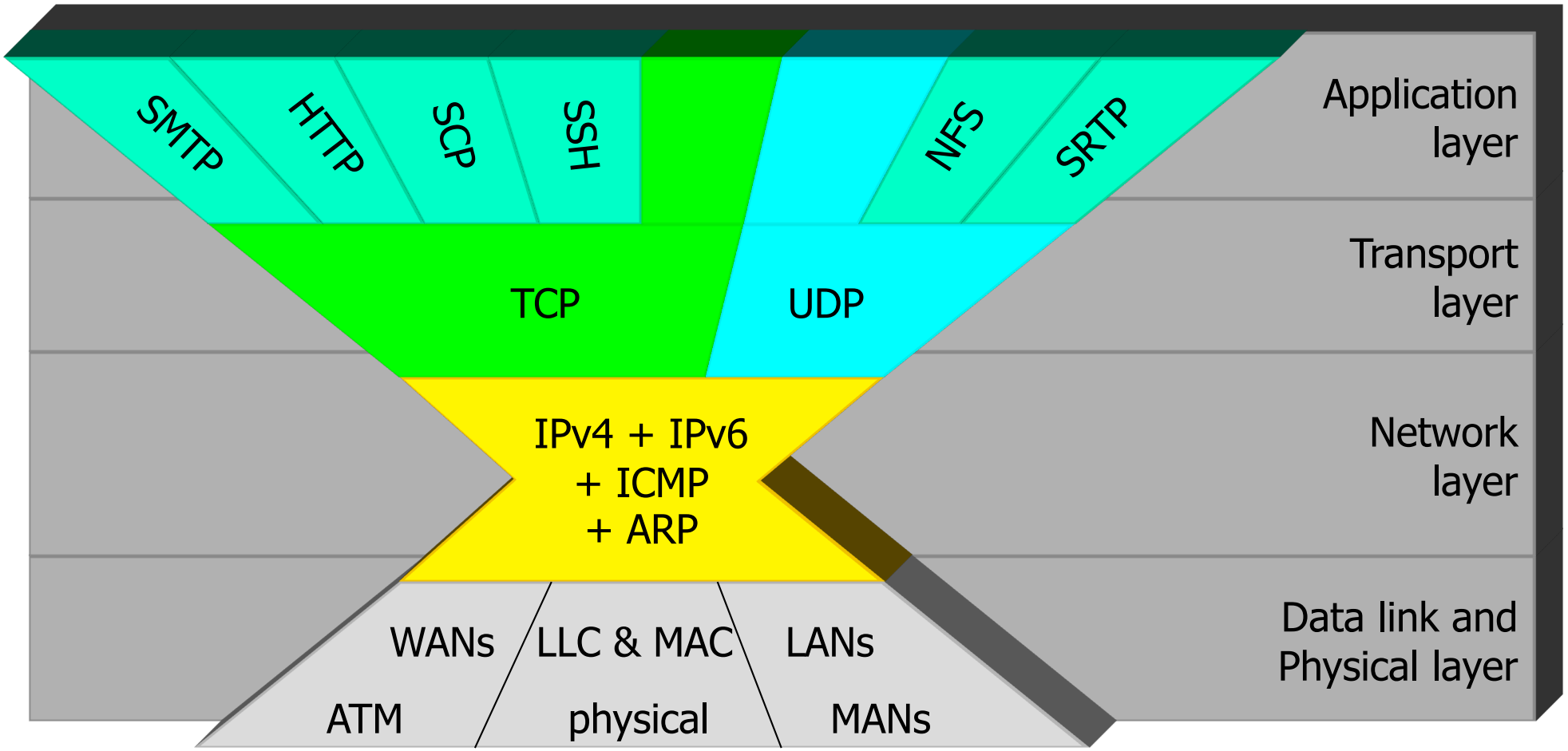
Why no clear separation of upper layers?

- layers 1-4 are essential for co-existence on the Internet
  - e.g. different congestion control mechanisms on different hosts can lead to strong congestion
- session and presentation layer functions provide mostly application support

Layers 3 and 4 are not clearly separated

- transport protocol (TCP, UDP, others) and network protocol IP
- sometimes hard to draw a clear line where TCP ends and IP begins
- example:
  - Explicit Congestion Notification (ECN) capability is indicated on layer 3 and congestion is indicated on layer 3
  - sender is told about receiver's reception of congestion signal on layer 4

Internet Protocol Stack



Nickname: “Hourglass Model”