IN2140:
Introduction to Operating Systems and Data Communication

Data Communication:

Network structures

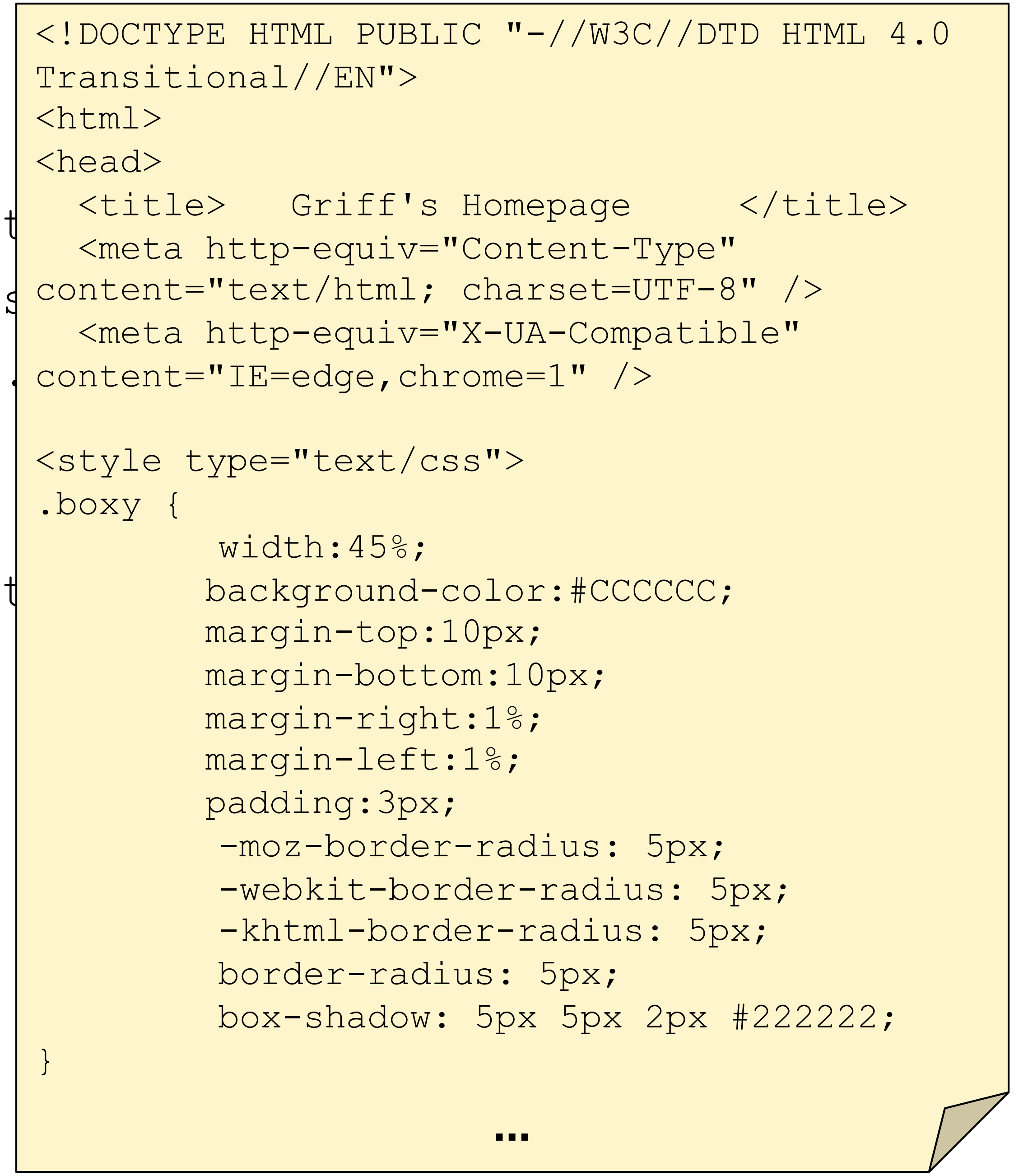
Monday, March 8, 2021

Programming the network

```
import urllib.request

contents = urllib.request.urlopen(
    urllib.request.Request(
        "http://heim.ifi.uio.no/griff/index.html")
    ).read()

print(contents.decode("utf-8”))
```

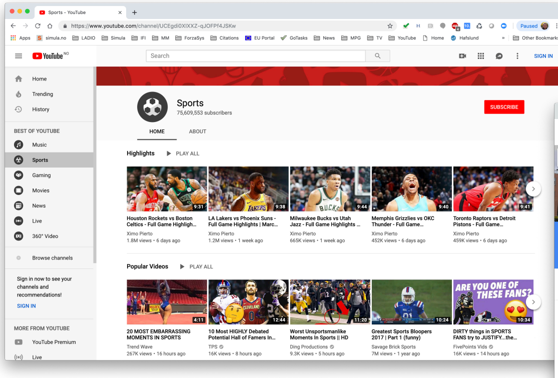
Programming the network

```
import urllib.request

contents = urllib.request
    urllib.request.Reques
        "http://heim.ifi.
    ).read()

print(contents.decode("ut
```

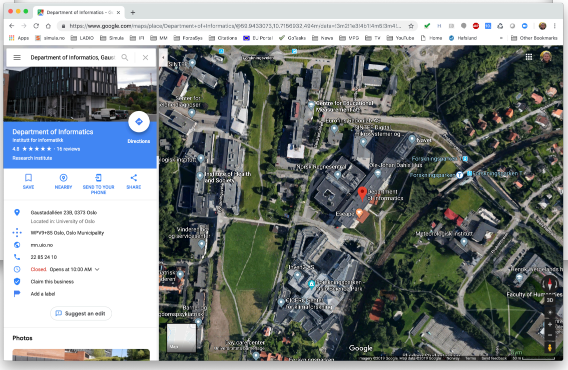


```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0
Transitional//EN">
<html>
<head>
  <title>   Griff's Homepage     </title>
  <meta http-equiv="Content-Type"
content="text/html; charset=UTF-8" />
  <meta http-equiv="X-UA-Compatible"
content="IE=edge,chrome=1" />

<style type="text/css">
.boxy {
         width:45%;
        background-color:#CCCCCC;
        margin-top:10px;
        margin-bottom:10px;
        margin-right:1%;
        margin-left:1%;
        padding:3px;
         -moz-border-radius: 5px;
         -webkit-border-radius: 5px;
         -khtml-border-radius: 5px;
         border-radius: 5px;
         box-shadow: 5px 5px 2px #222222;
}

                      ...
```

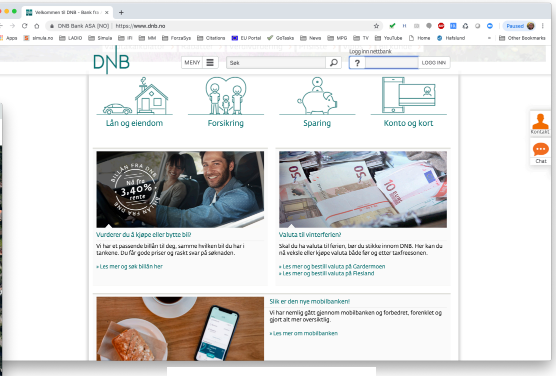
Web-based software



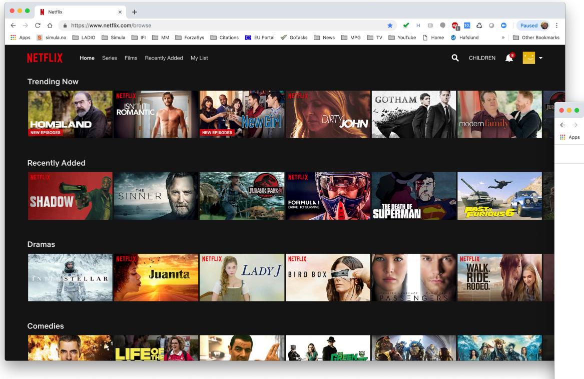
Video



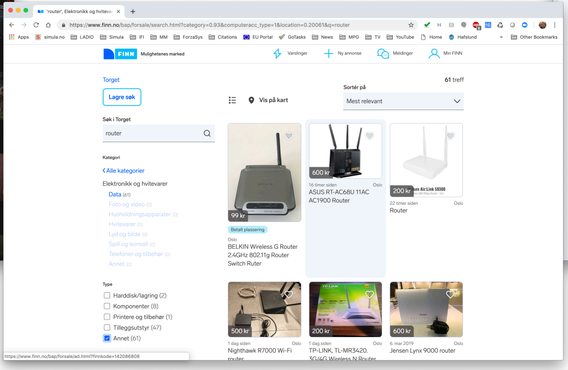
Navigation



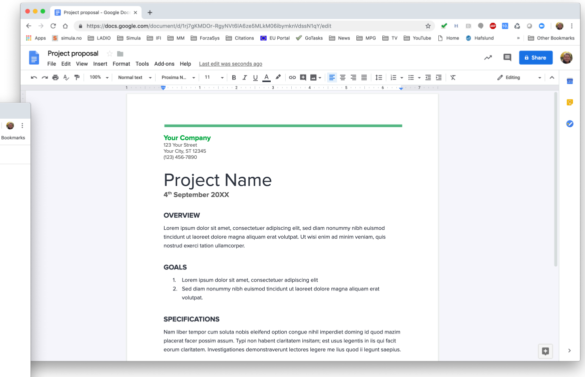
Banking



TV



Shopping



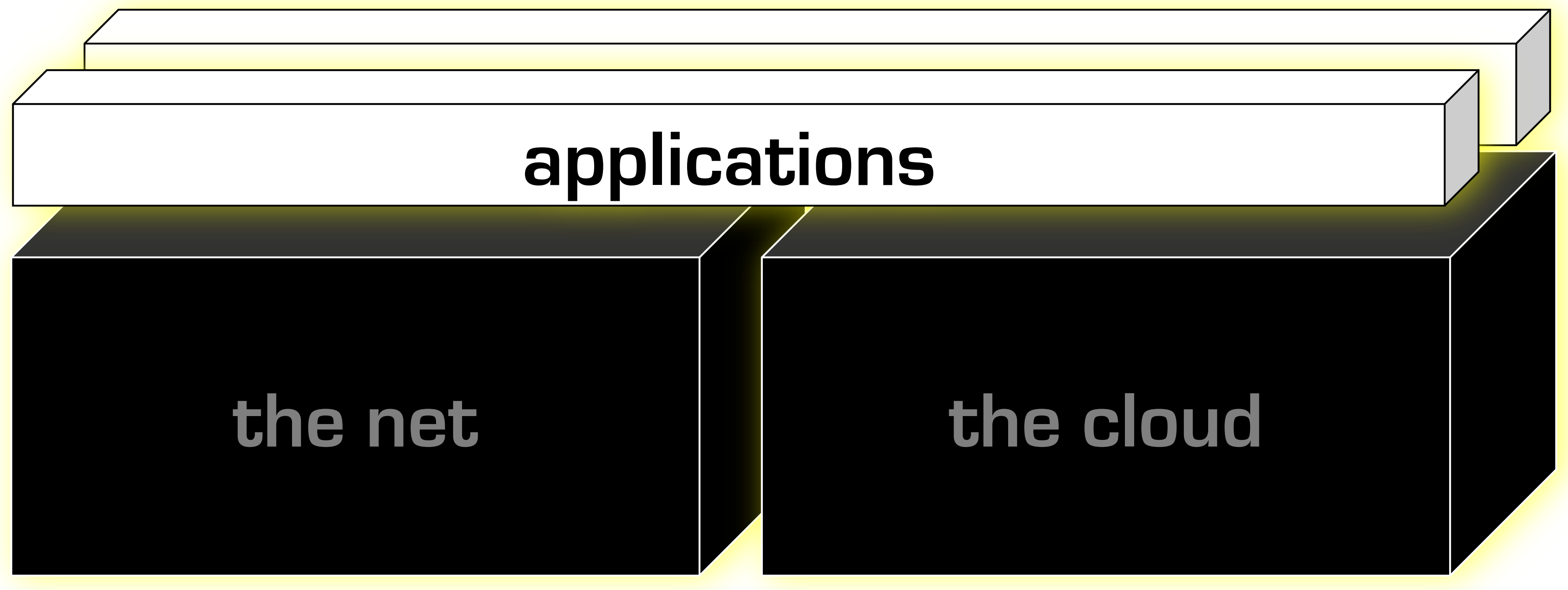
Office tools

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?

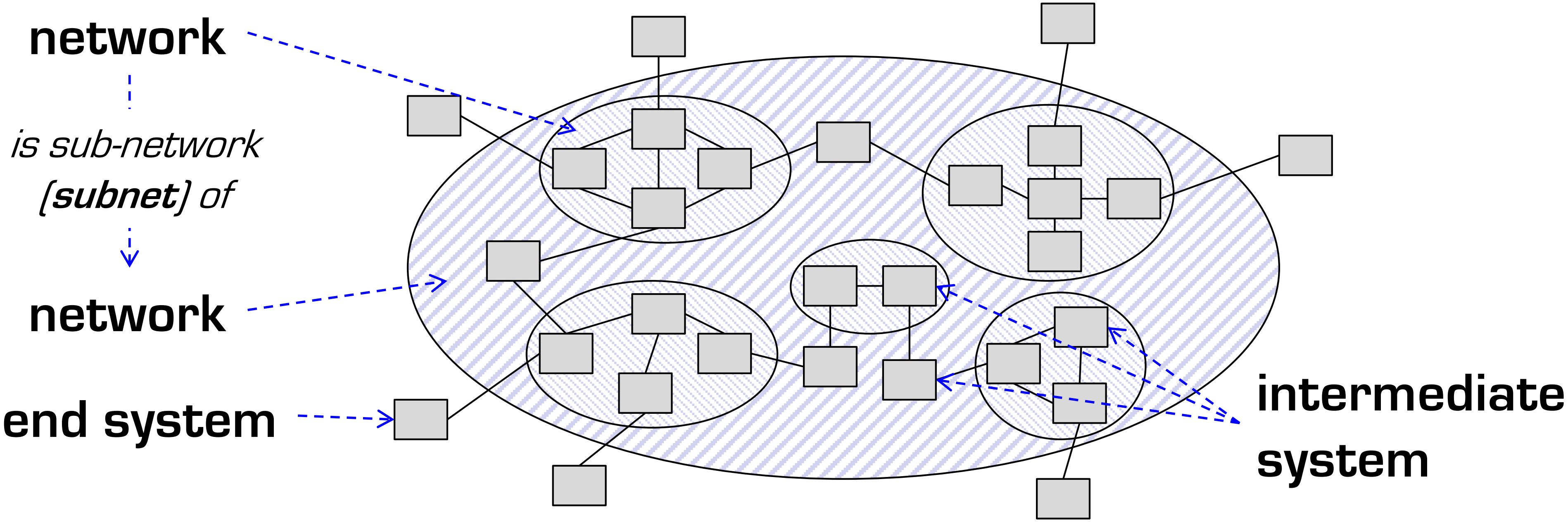
IN2140:
Introduction to Operating Systems and Data Communication

Network structures

Structures

Monday, March 8, 2021

Network Components



network

is sub-network ***(subnet)*** *of*

network

end system

intermediate system

ES - End system

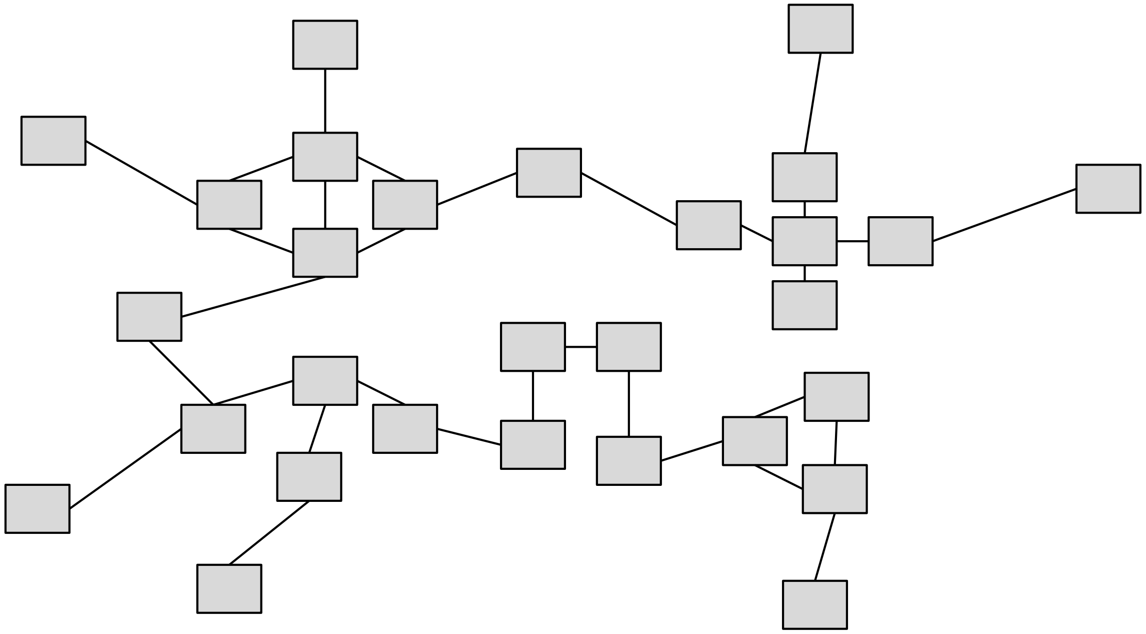
- end systems are "at the edge" of a network
- examples: computer, mobile phone, tablet, smart watch, printer, TV, smoke detector, weather station, lamp, door opener, fridge, traffic light …

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

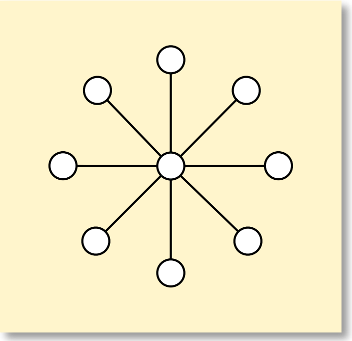
a specific arrangement of nodes and edges is called a **topology**

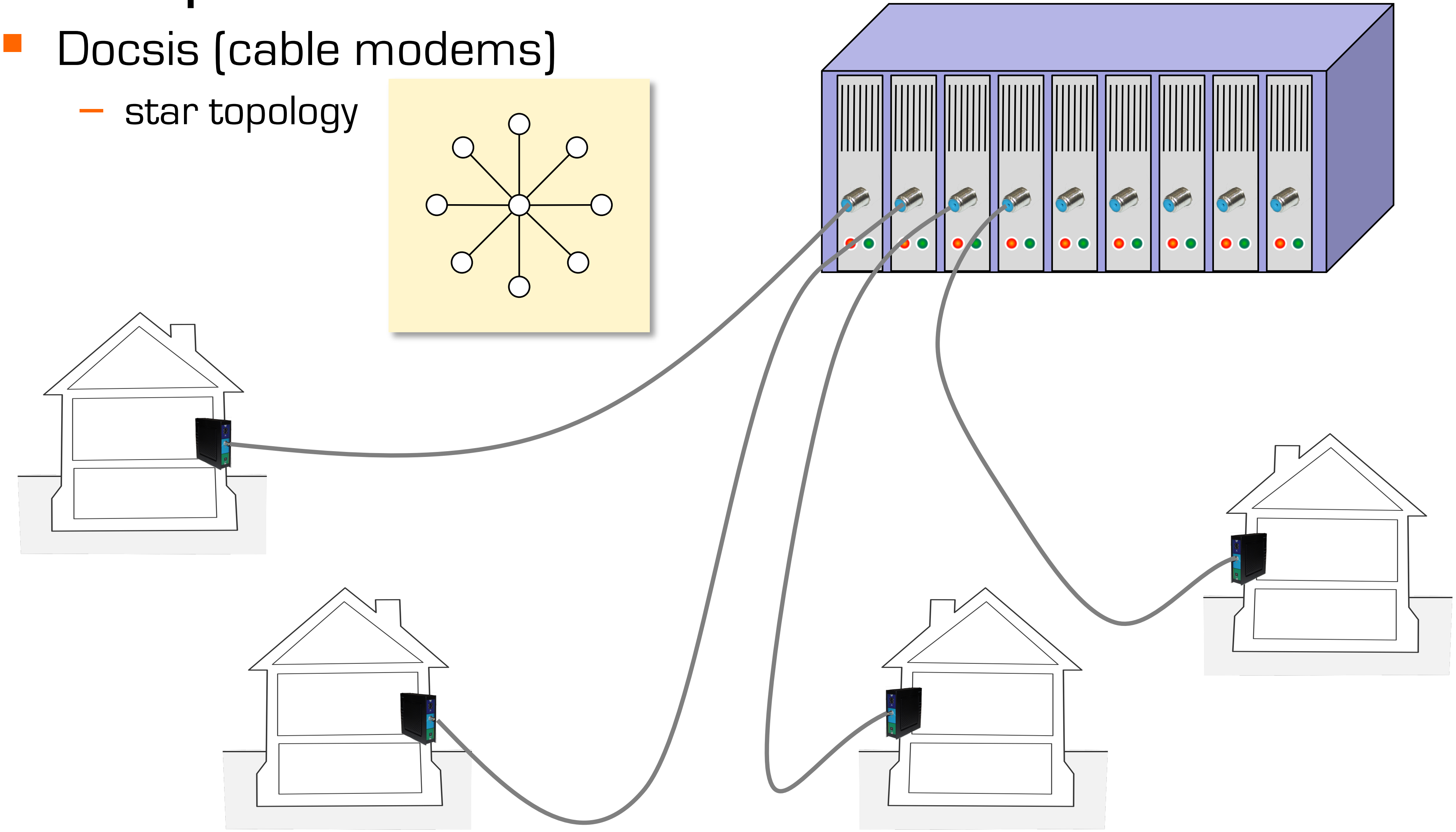
the terminology implies that edges are network connections

Network Structures

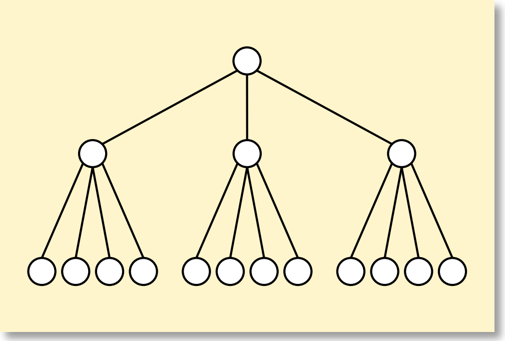
Point-to-point channels

- Docsis (cable modems)
  - star topology



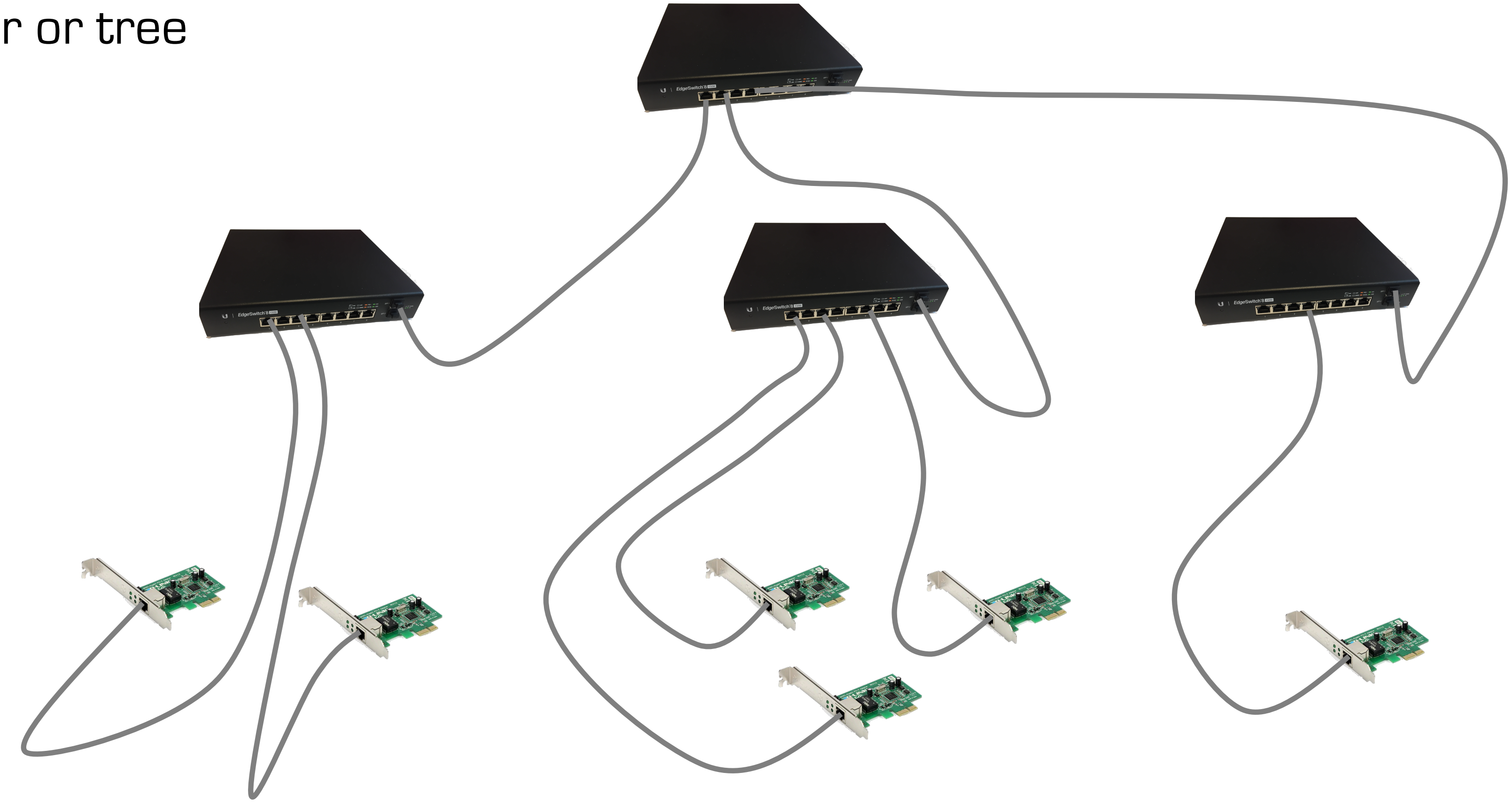


Network Structures



Point-to-point channels

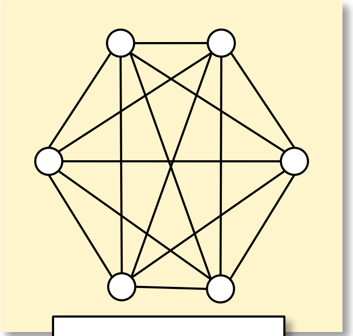
- Docsis (cable modems)
  - star topology
- Gigabit Ethernet (“1GB Ethernet”)
  - star or tree



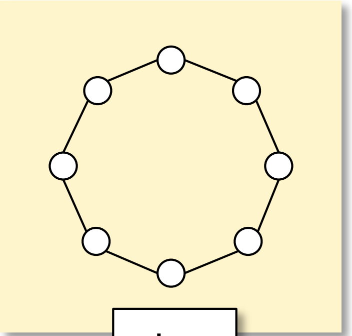
Network Structures

Point-to-point channels

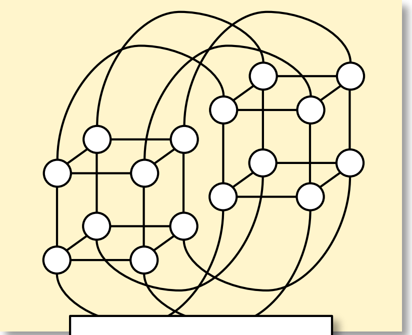
- Docsis (cable modems)
  - star topology
- Gigabit Ethernet ("1GB Ethernet")
  - star or tree
- IEEE 802.5 "TokenRing" (outdated)
  - ring
- Some supercomputers use
  - full mesh
  - hypercube
  - torus
  - fat tree



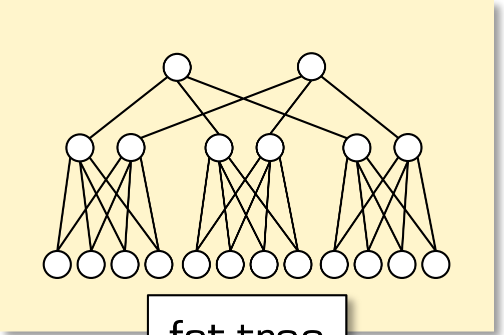
full mesh



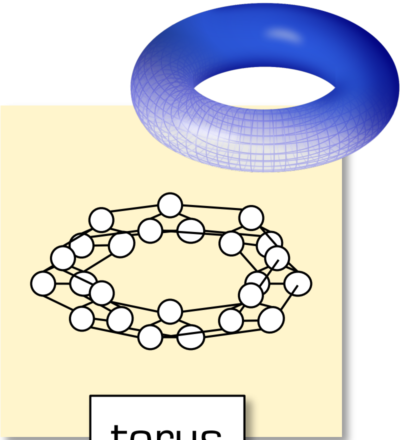
ring



hypercube



fat tree



torus

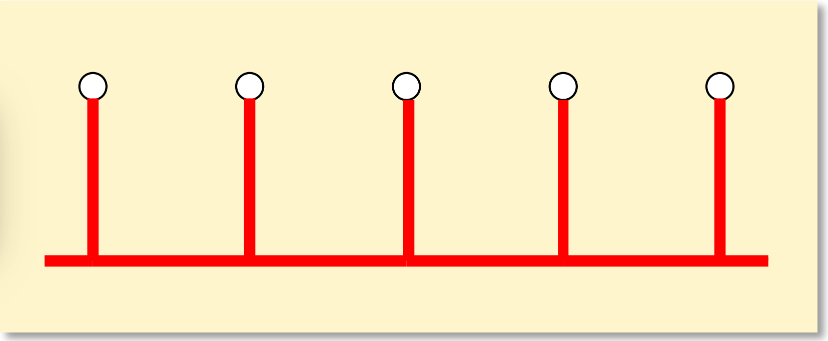
Network Structures

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



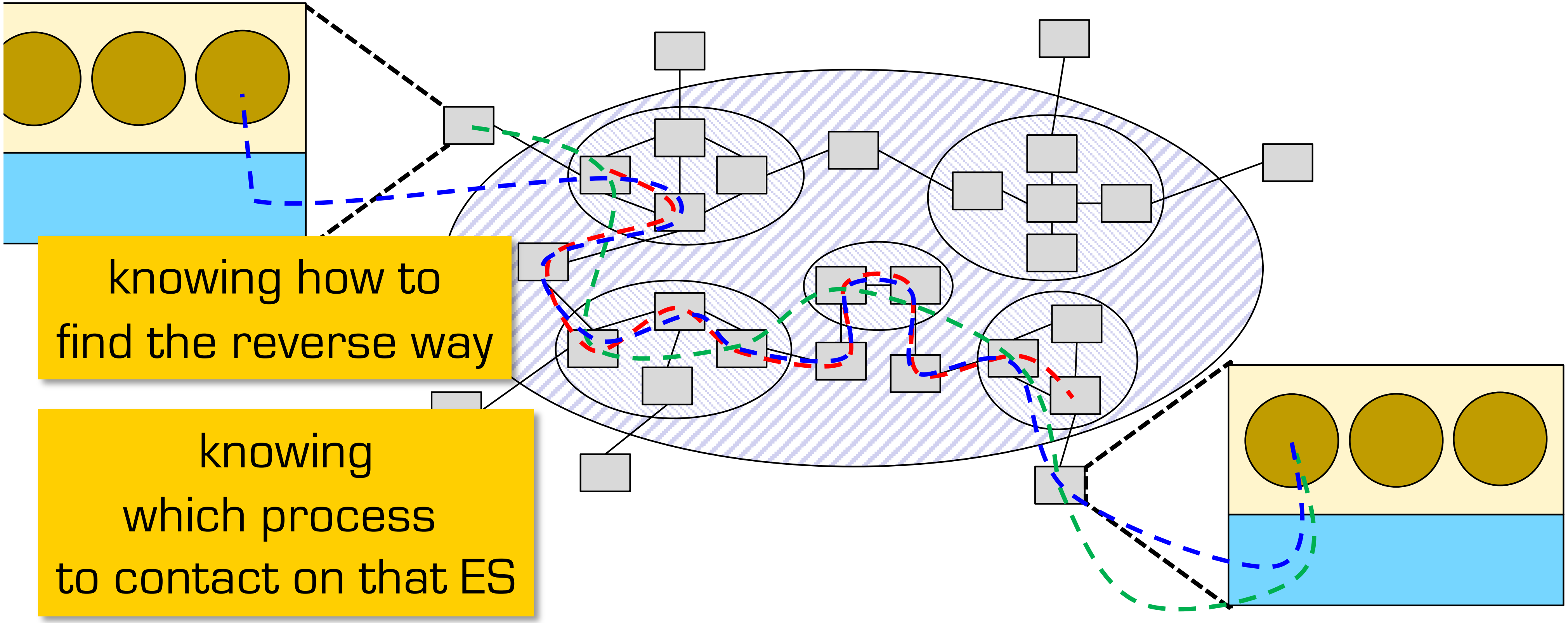
IN2140:
Introduction to Operating Systems and Data Communication

Network structures

Network’s tasks

Monday, March 8, 2021

A lot of tasks



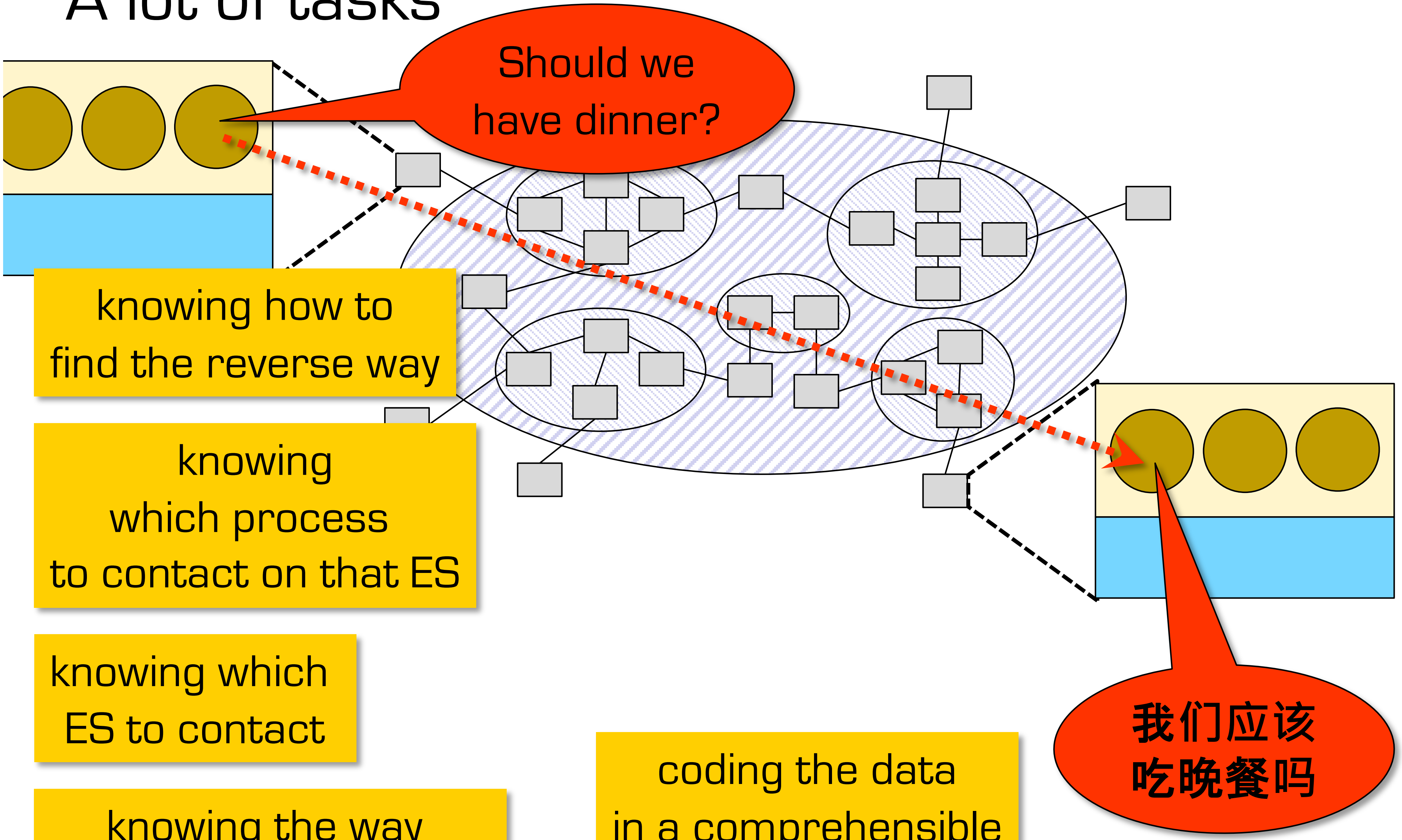
knowing how to find the reverse way

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

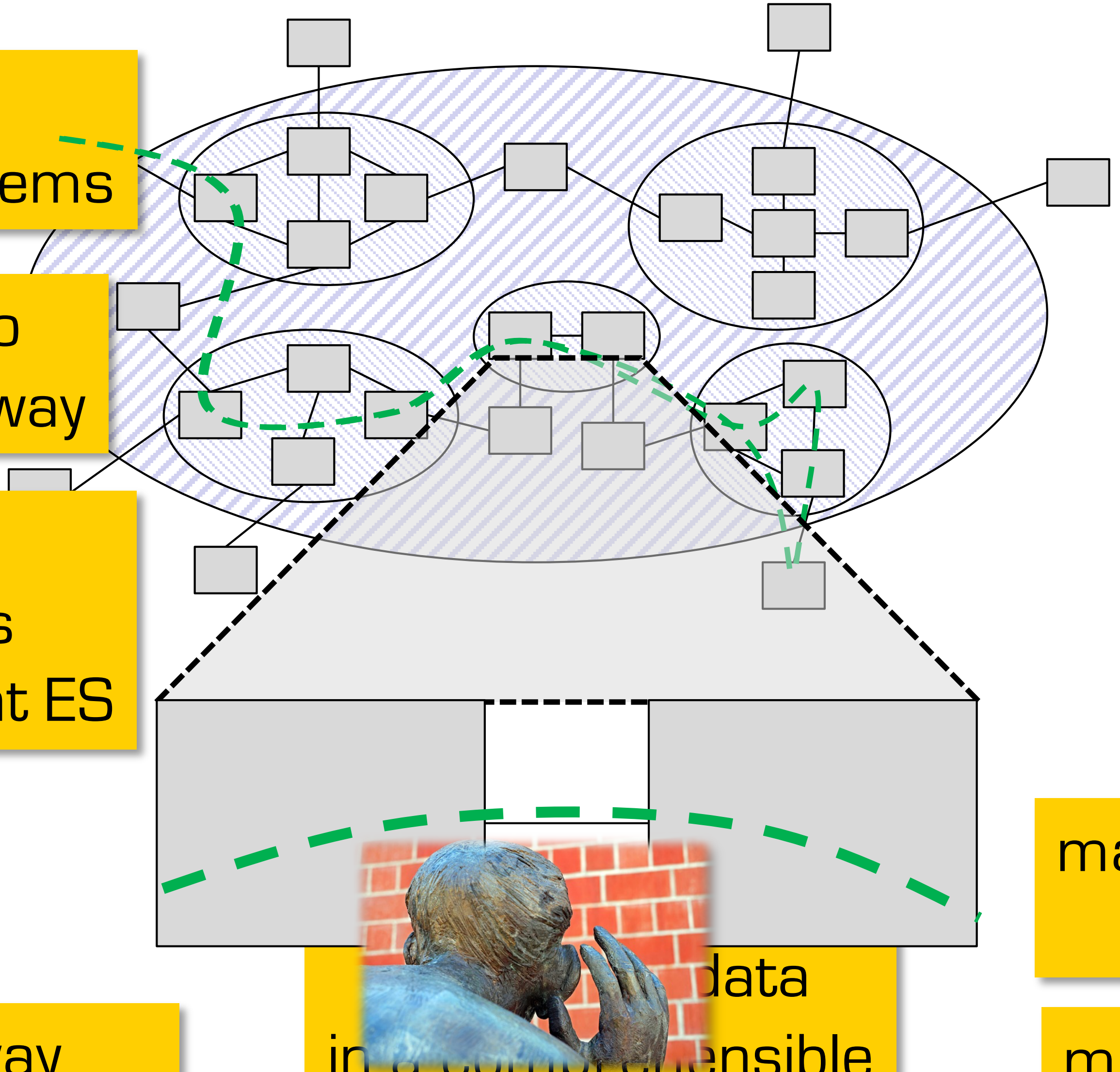
knowing which process to contact on that ES

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

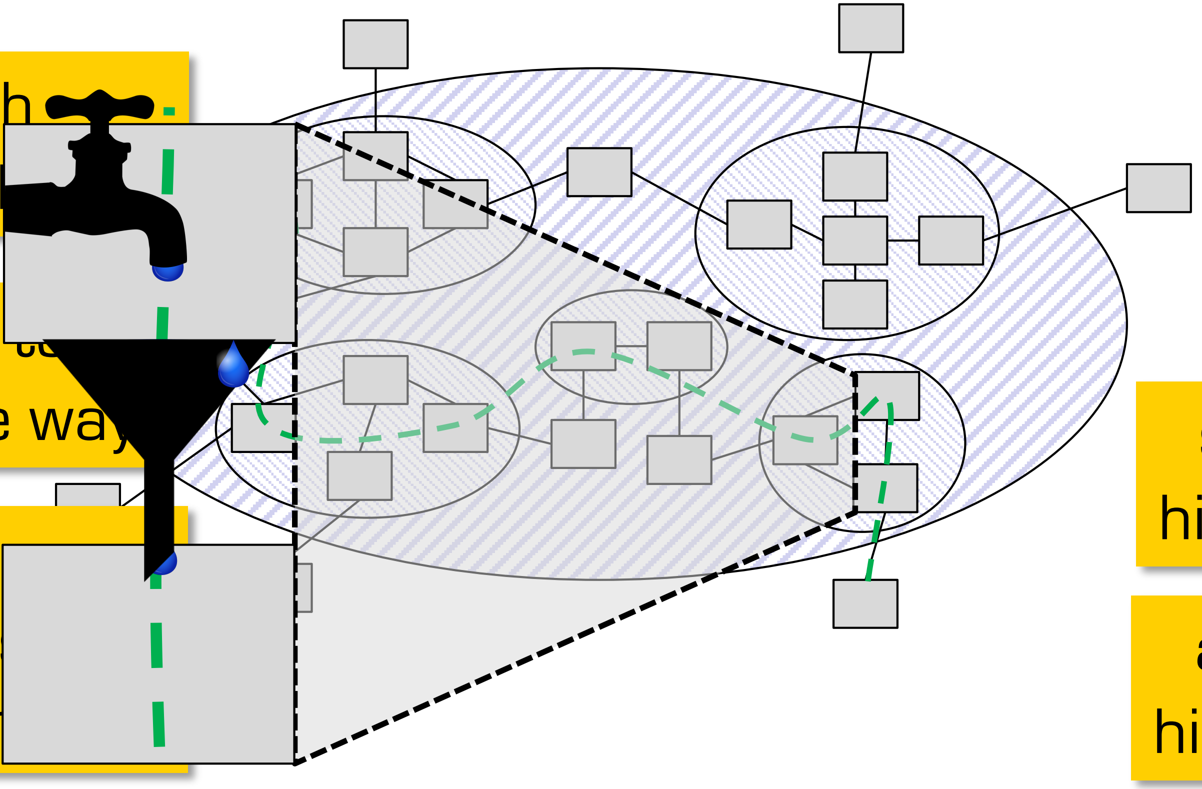
maintaining security

A lot of tasks

dealing with
networking pro

knowing how
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

the application

applications determine requirements

knowing the way from one IS to another

knowing which ES to contact

knowing which process to contact on that ES

knowing how to find the reverse way

coding the data in a comprehensible manner

dealing with networking problems

support high traffic

avoiding high delays

maintaining security

maintaining privacy

A lot of tasks

There are a lot of aspects to worry about

the application

applications determine requirements

knowing the way from one IS to another

knowing which ES to contact

knowing which process to contact on that ES

knowing how to find the reverse way

coding the data in a comprehensible manner

dealing with networking problems

support high traffic

avoiding high delays

maintaining security

maintaining privacy

the application

applications
determine
requirements

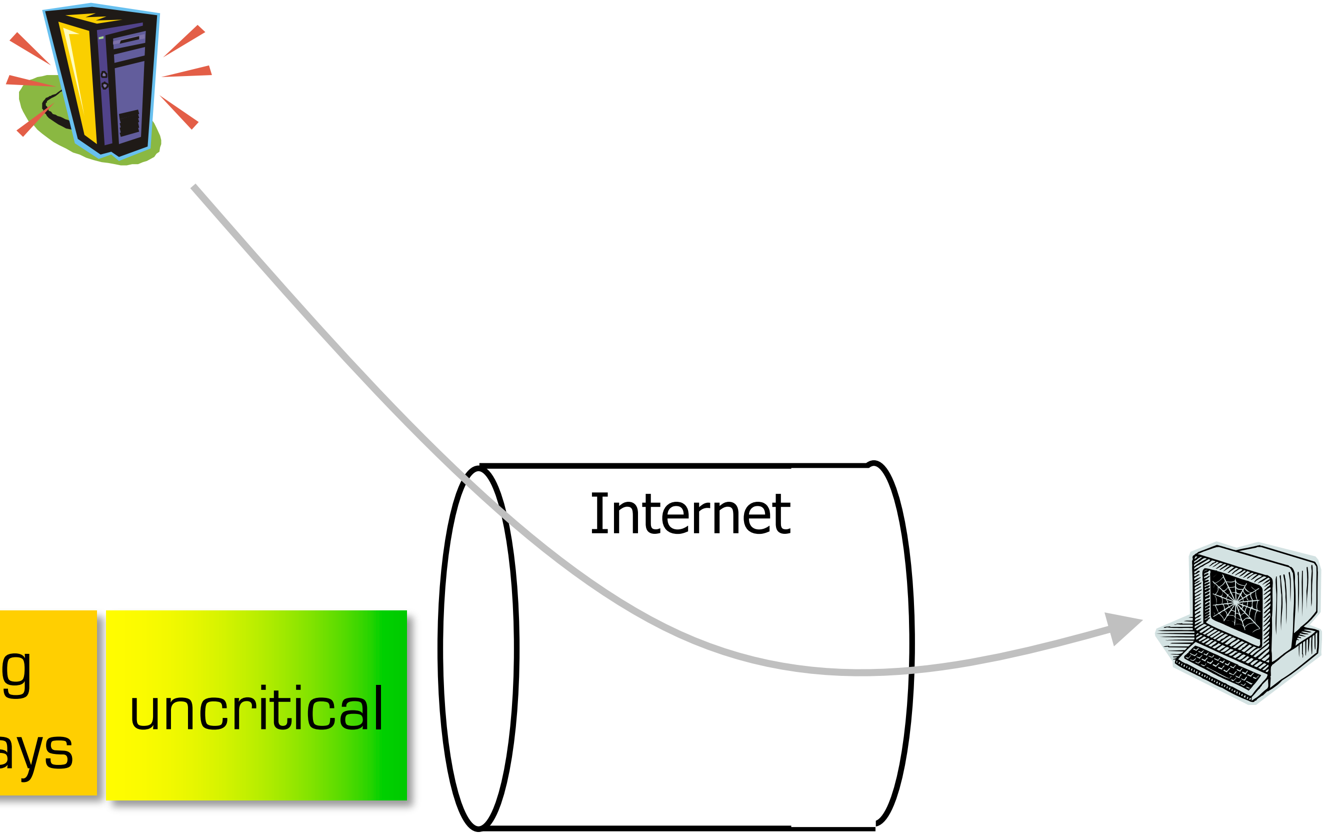
dealing with
networking problems

support
high traffic

avoiding
high delays

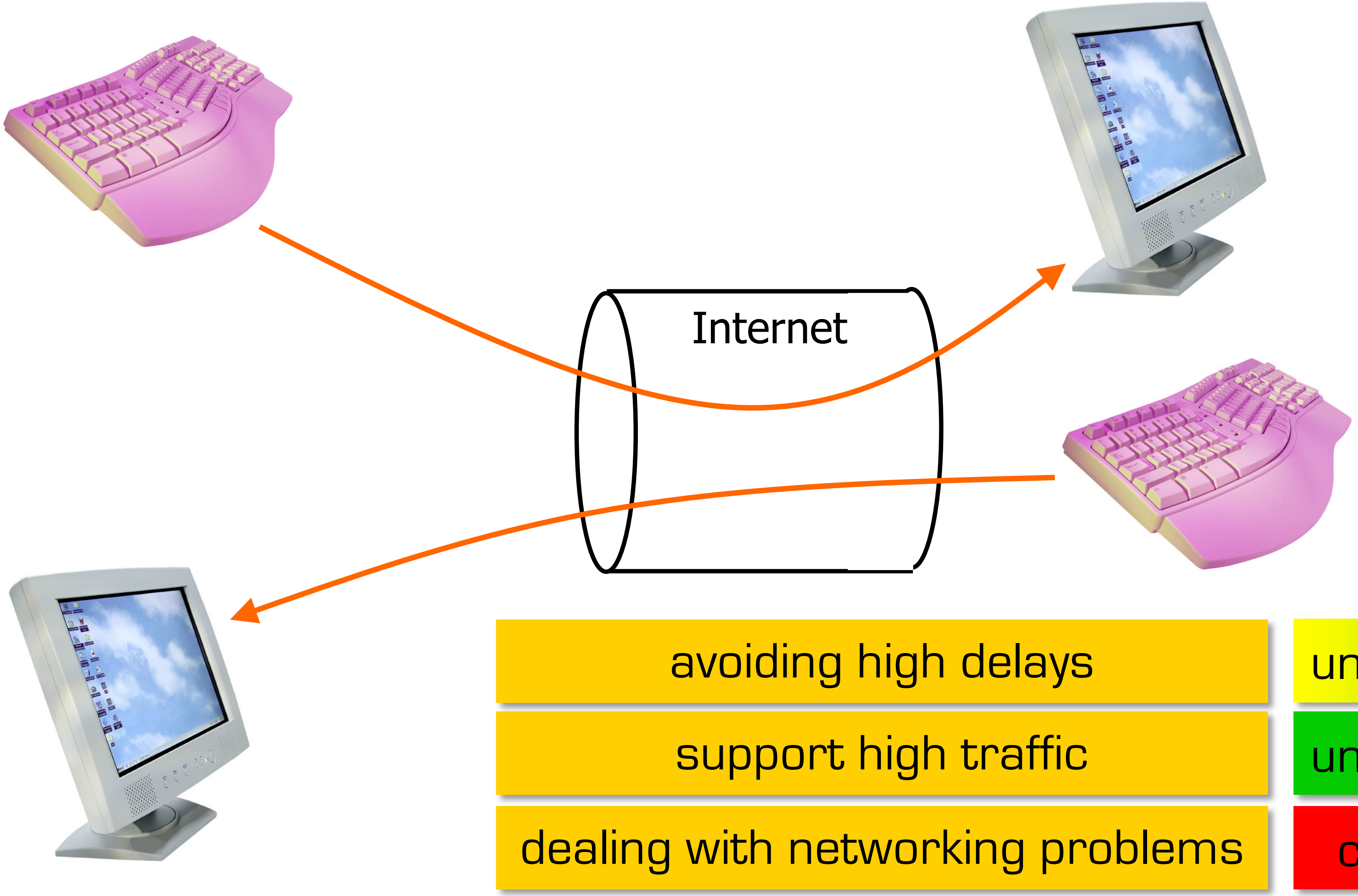
The application

File download and Web browsing



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

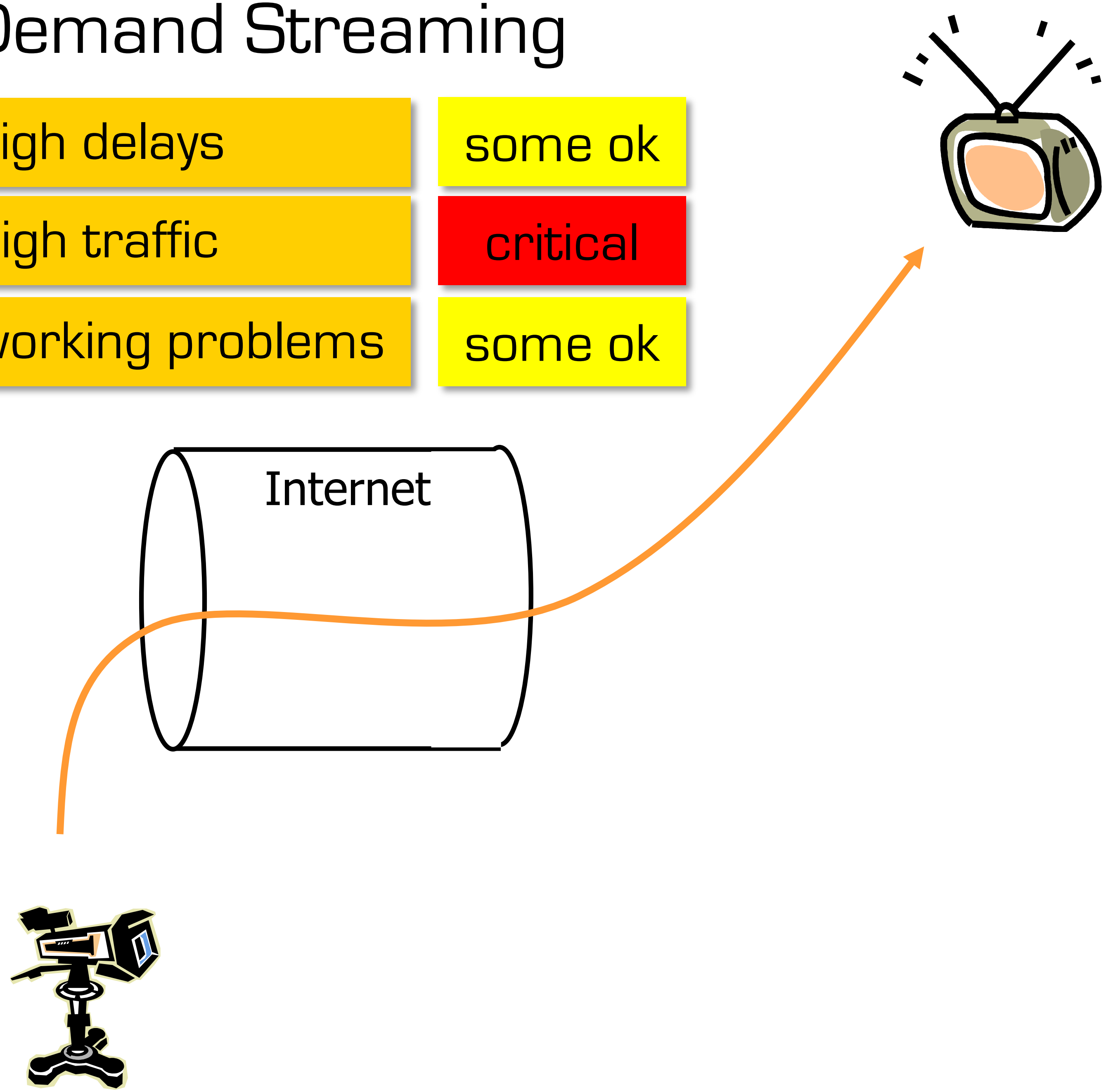
Textual commands and textual chat



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

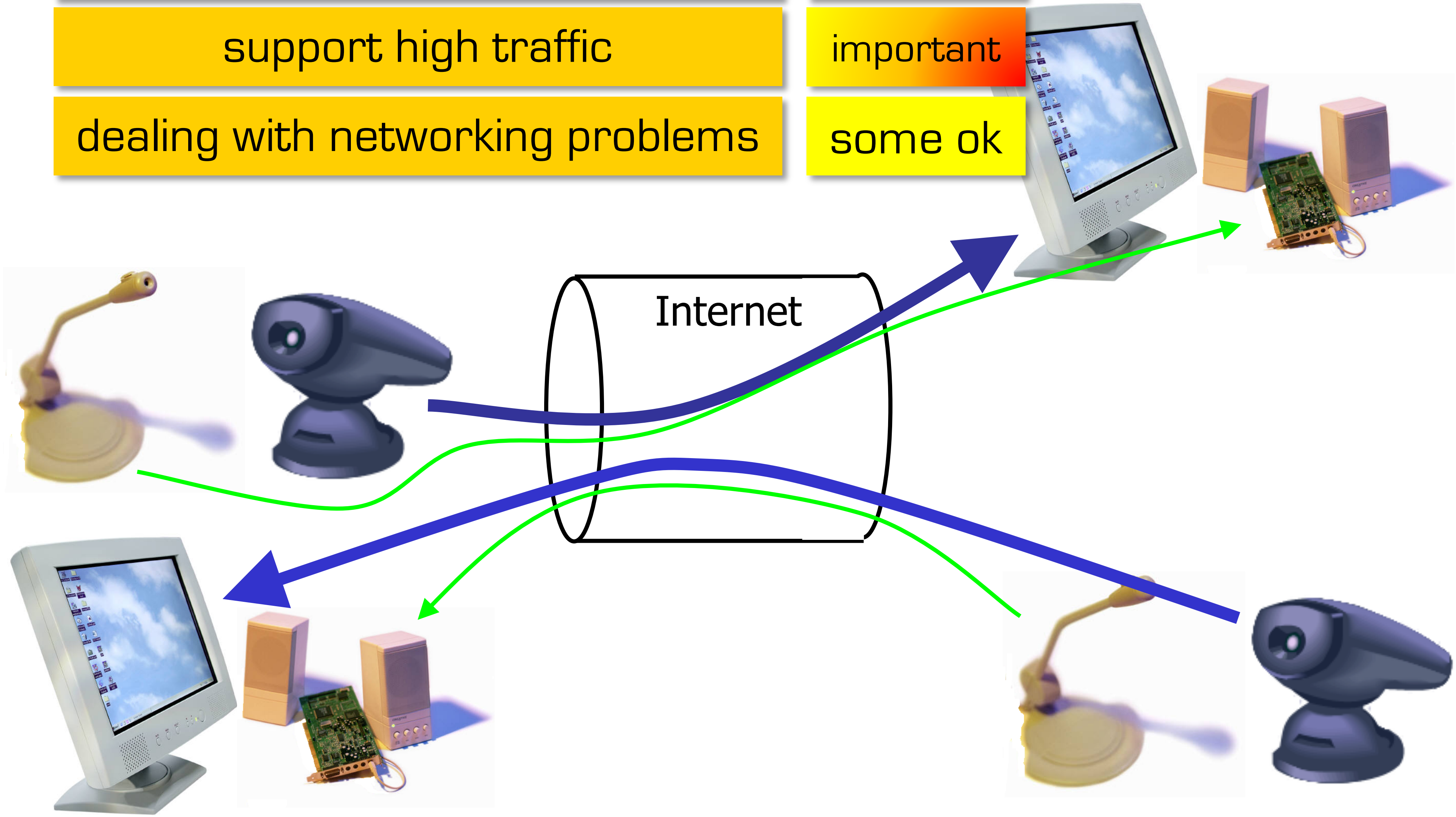
Live and on-Demand Streaming

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

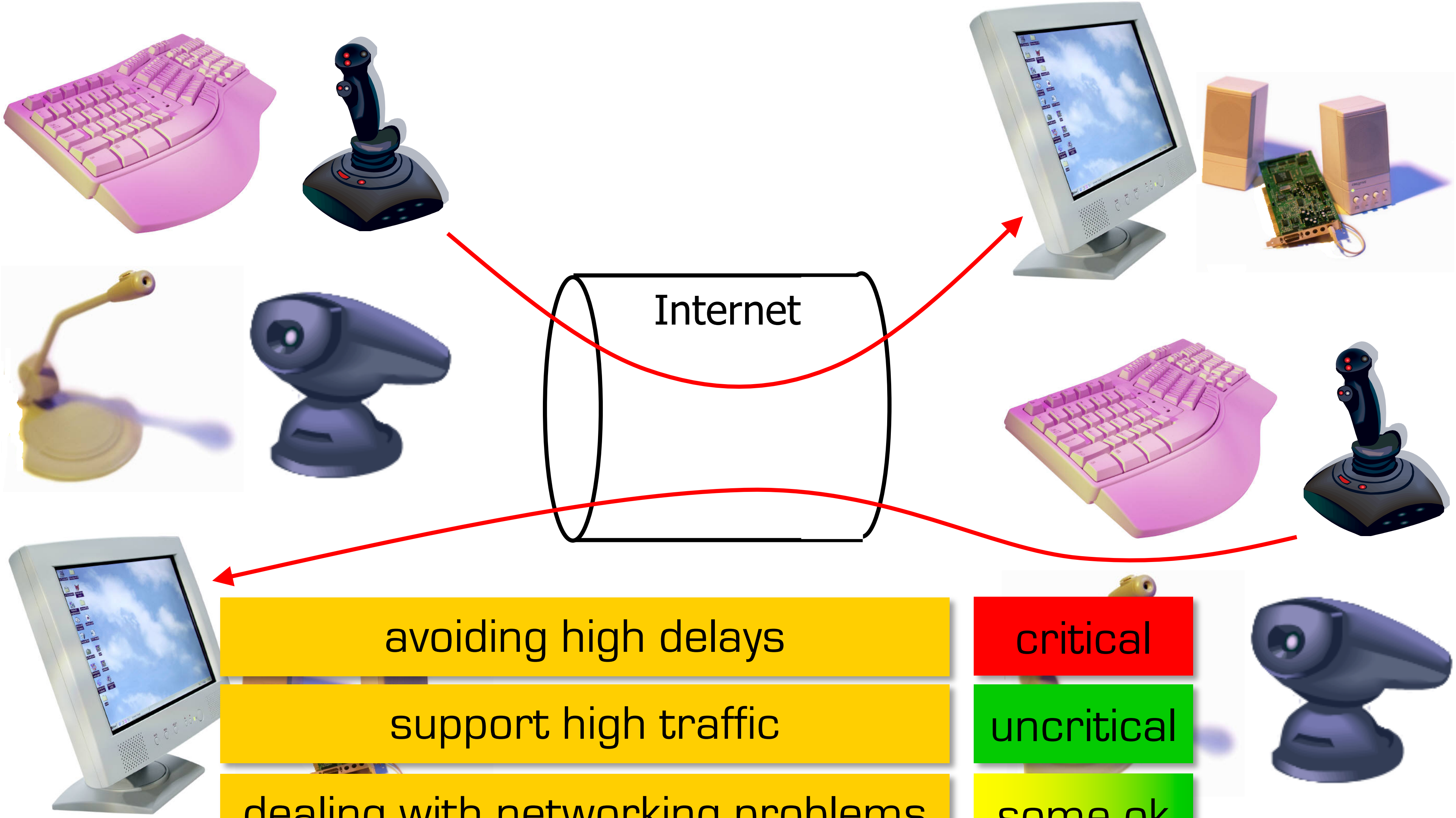


AV chat and AV conferencing

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

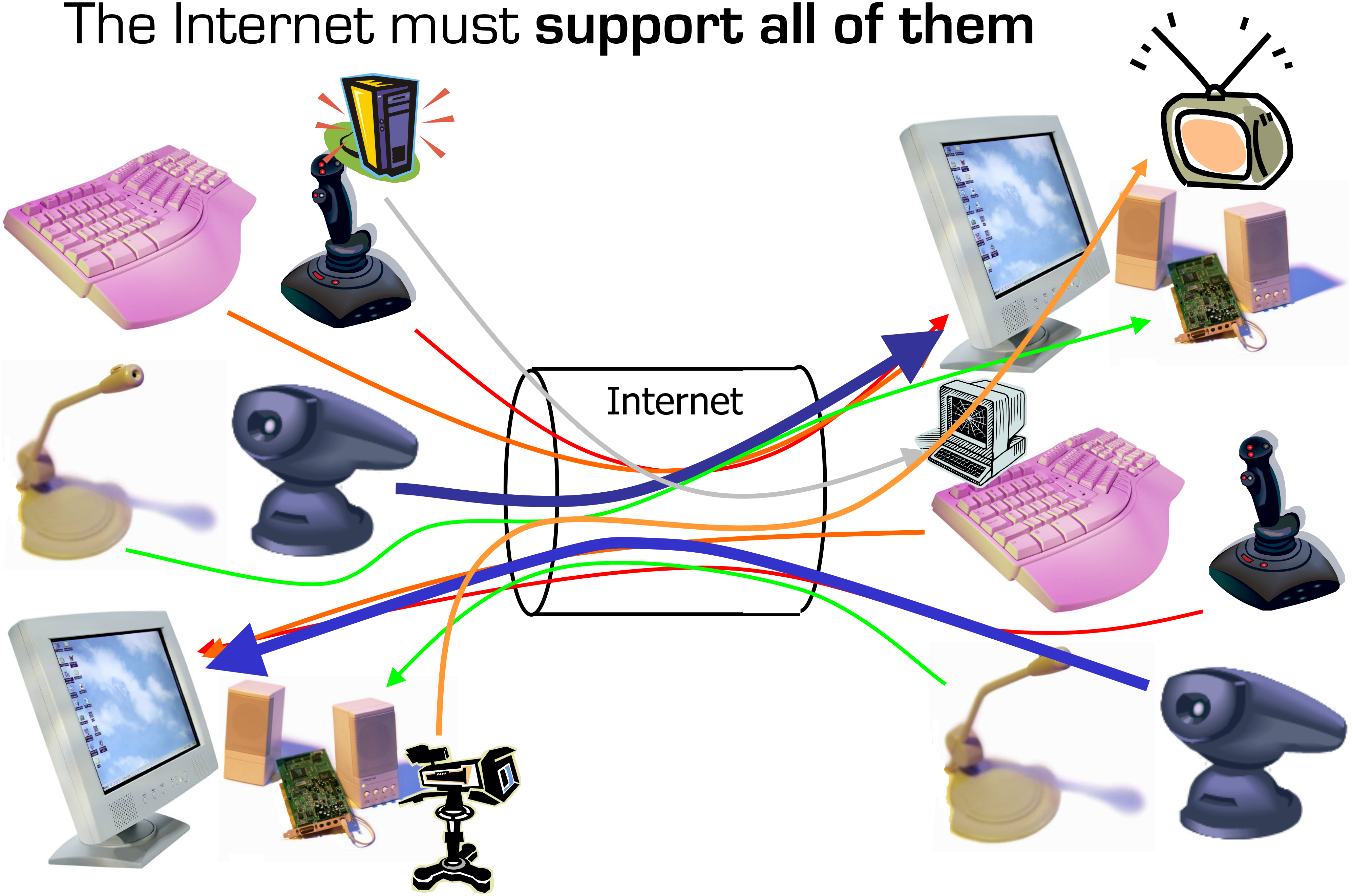


Haptic Interaction



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

The Internet must **support all of them**

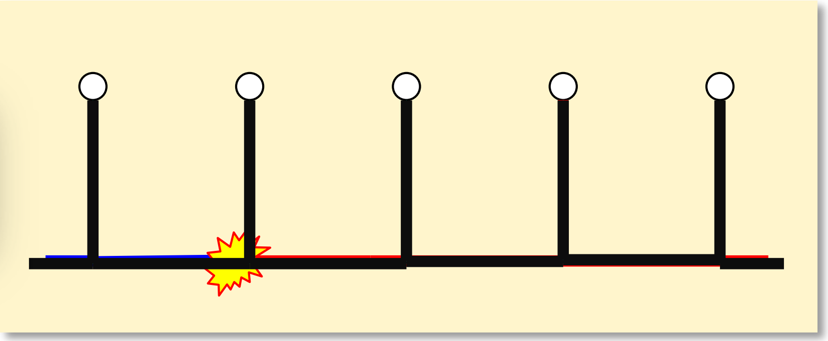


Network Structures

Broadcasting channels

- Cable
  - old-fashioned Ethernet

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



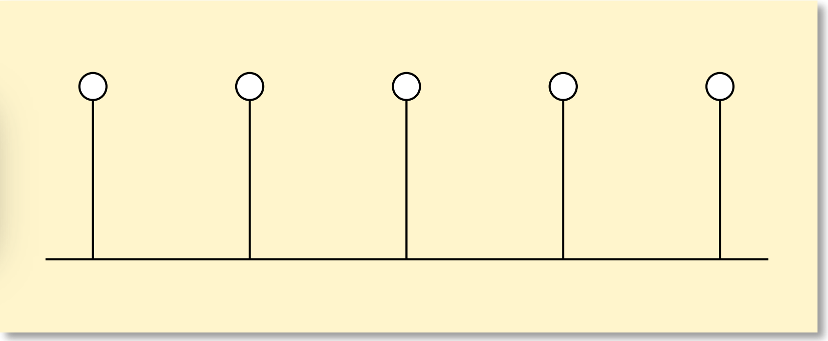
- Properties
  - when one node sends, potentially many nodes can hear it
  - when two nodes send, both messages are potentially ruined
  - error detection is important
  - coordination is desirable

Network Structures

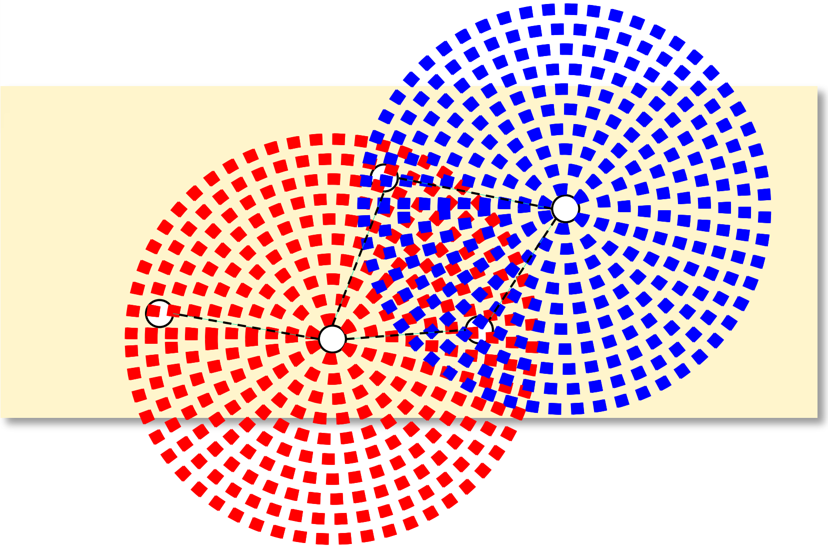
Broadcasting channels

- Cable
  - old-fashioned Ethernet

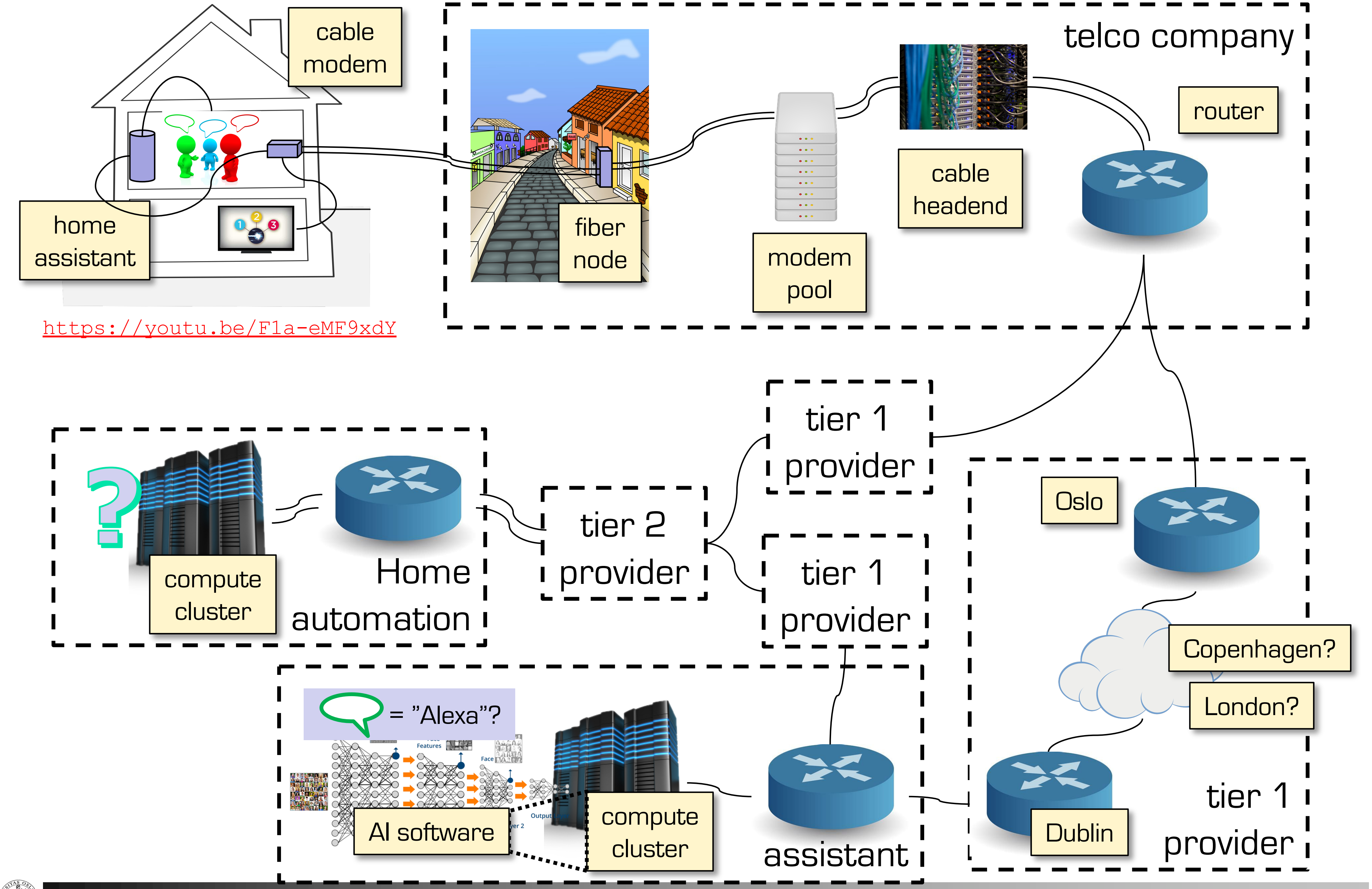
- Radio
  - Aloha (first wireless data transmission)
  - WiFi (IEEE 802.11)
  - mobile: 3G, 4G, 5G
  - satellites



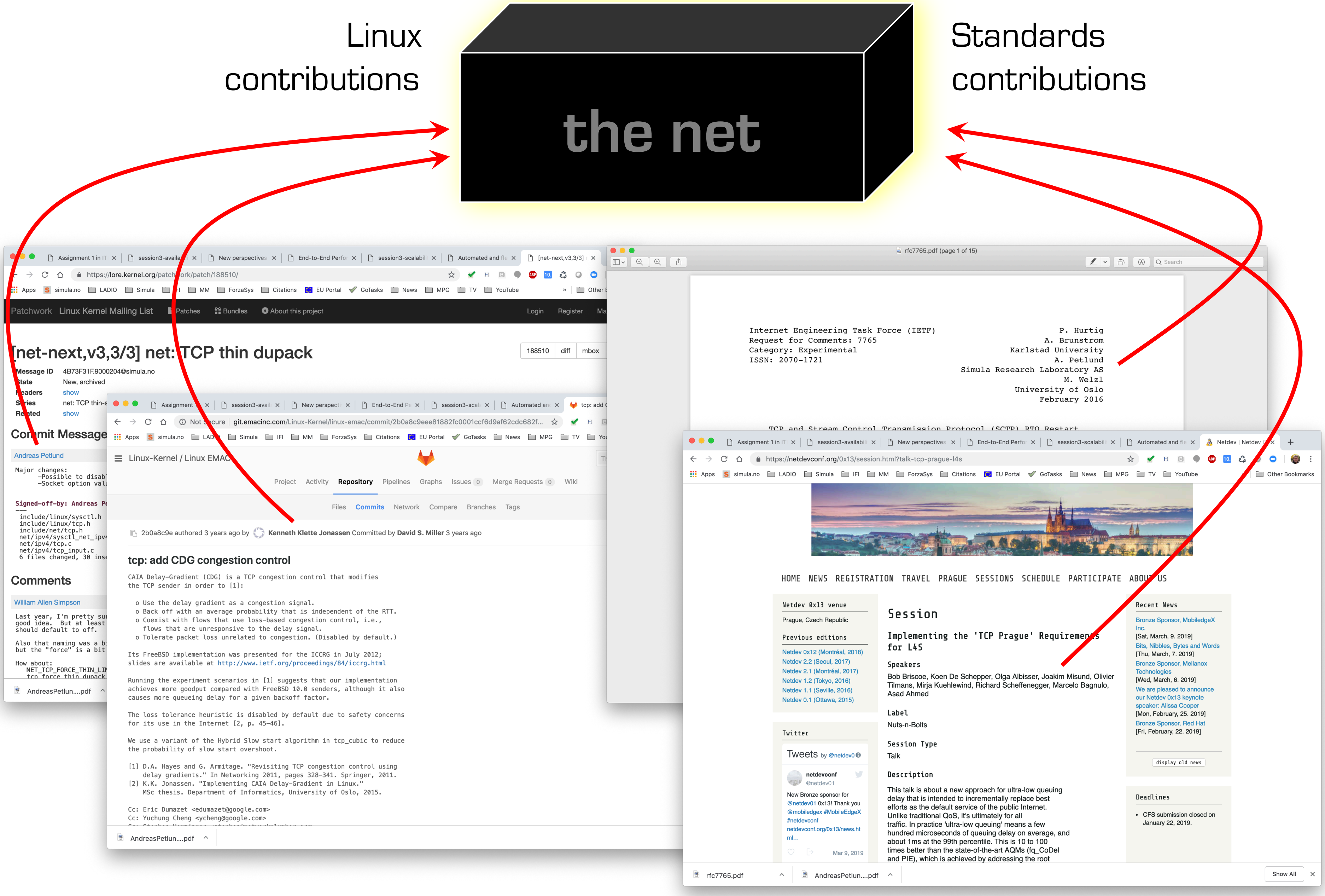
- Properties
  - when one node sends, potentially many nodes can hear it
  - when two nodes send, both messages are potentially ruined
  - error detection is important
  - coordination is desirable



A possible path communication path



But does it matter if *we* understand it?



[net-next,v3,3/3] net: TCP thin dupack

Message ID 4B73F31F.9000204@simula.no
State New, archived
Headers show
Series net: TCP thin-s
Related show

Commit Message

Andreas Petlund

```
Major changes:
      -Possible to disabl
      -Socket option valu

Signed-off-by: Andreas Pe
---
 include/linux/sysctl.h
 include/linux/tcp.h
 include/net/tcp.h
 net/ipv4/sysctl_net_ipv4
 net/ipv4/tcp.c
 net/ipv4/tcp_input.c
 6 files changed, 30 inse
```

Comments

William Allen Simpson

```
Last year, I'm pretty su
good idea.  But at least
should default to off.

Also that naming was a bi
but the "force" is a bit

How about:
   NET_TCP_FORCE_THIN_LIN
   tcp_force_thin_dupack
```

Not Secure | git.emacinc.com/Linux-Kernel/linux-emac/commit/2b0a8c9eee81882fc0001ccf6d9af62cdc682f...

2b0a8c9e authored 3 years ago by Kenneth Klette Jonassen Committed by David S. Miller 3 years ago

tcp: add CDG congestion control

```
CAIA Delay-Gradient (CDG) is a TCP congestion control that modifies
the TCP sender in order to [1]:

  o Use the delay gradient as a congestion signal.
  o Back off with an average probability that is independent of the RTT.
  o Coexist with flows that use loss-based congestion control, i.e.,
    flows that are unresponsive to the delay signal.
  o Tolerate packet loss unrelated to congestion. (Disabled by default.)

Its FreeBSD implementation was presented for the ICCRG in July 2012;
slides are available at http://www.ietf.org/proceedings/84/iccrg.html

Running the experiment scenarios in [1] suggests that our implementation
achieves more goodput compared with FreeBSD 10.0 senders, although it also
causes more queueing delay for a given backoff factor.

The loss tolerance heuristic is disabled by default due to safety concerns
for its use in the Internet [2, p. 45-46].

We use a variant of the Hybrid Slow start algorithm in tcp_cubic to reduce
the probability of slow start overshoot.

[1] D.A. Hayes and G. Armitage. "Revisiting TCP congestion control using
    delay gradients." In Networking 2011, pages 328-341. Springer, 2011.
[2] K.K. Jonassen. "Implementing CAIA Delay-Gradient in Linux."
    MSc thesis. Department of Informatics, University of Oslo, 2015.

Cc: Eric Dumazet <edumazet@google.com>
Cc: Yuchung Cheng <ycheng@google.com>
```

rfc7765.pdf (page 1 of 15)

```
Internet Engineering Task Force (IETF)                        P. Hurtig
Request for Comments: 7765                                A. Brunstrom
Category: Experimental                               Karlstad University
ISSN: 2070-1721                                              A. Petlund
                                            Simula Research Laboratory AS
                                                                 M. Welzl
                                                      University of Oslo
                                                           February 2016


        TCP and Stream Control Transmission Protocol (SCTP) RTO Restart
```

https://netdevconf.org/0x13/session.html?talk-tcp-prague-l4s

HOME NEWS REGISTRATION TRAVEL PRAGUE SESSIONS SCHEDULE PARTICIPATE ABOUT US

Netdev 0x13 venue

Prague, Czech Republic

Previous editions

- Netdev 0x12 (Montréal, 2018)
- Netdev 2.2 (Seoul, 2017)
- Netdev 2.1 (Montréal, 2017)
- Netdev 1.2 (Tokyo, 2016)
- Netdev 1.1 (Seville, 2016)
- Netdev 0.1 (Ottawa, 2015)

Twitter

Tweets by @netdev0

netdevconf @netdev01

New Bronze sponsor for @netdev01 0x13! Thank you @mobiledgex #MobileEdgeX #netdevconf netdevconf.org/0x13/news.html...

Mar 9, 2019

Session

Implementing the 'TCP Prague' Requirements for L4S

Speakers

Bob Briscoe, Koen De Schepper, Olga Albisser, Joakim Misund, Olivier Tilmans, Mirja Kuehlewind, Richard Scheffenegger, Marcelo Bagnulo, Asad Ahmed

Label

Nuts-n-Bolts

Session Type

Talk

Description

This talk is about a new approach for ultra-low queuing delay that is intended to incrementally replace best efforts as the default service of the public Internet. Unlike traditional QoS, it's ultimately for all traffic. In practice 'ultra-low queuing' means a few hundred microseconds of queuing delay on average, and about 1ms at the 99th percentile. This is 10 to 100 times better than the state-of-the-art AQMs (fq_CoDel and PIE), which is achieved by addressing the root

Recent News

- Bronze Sponsor, MobiledgeX Inc. [Sat, March, 9. 2019]
- Bits, Nibbles, Bytes and Words [Thu, March, 7. 2019]
- Bronze Sponsor, Mellanox Technologies [Wed, March, 6. 2019]
- We are pleased to announce our Netdev 0x13 keynote speaker: Alissa Cooper [Mon, February, 25. 2019]
- Bronze Sponsor, Red Hat [Fri, February, 22. 2019]

display old news

Deadlines

- CFS submission closed on January 22, 2019.

IN2140:
Introduction to Operating Systems and Data Communication

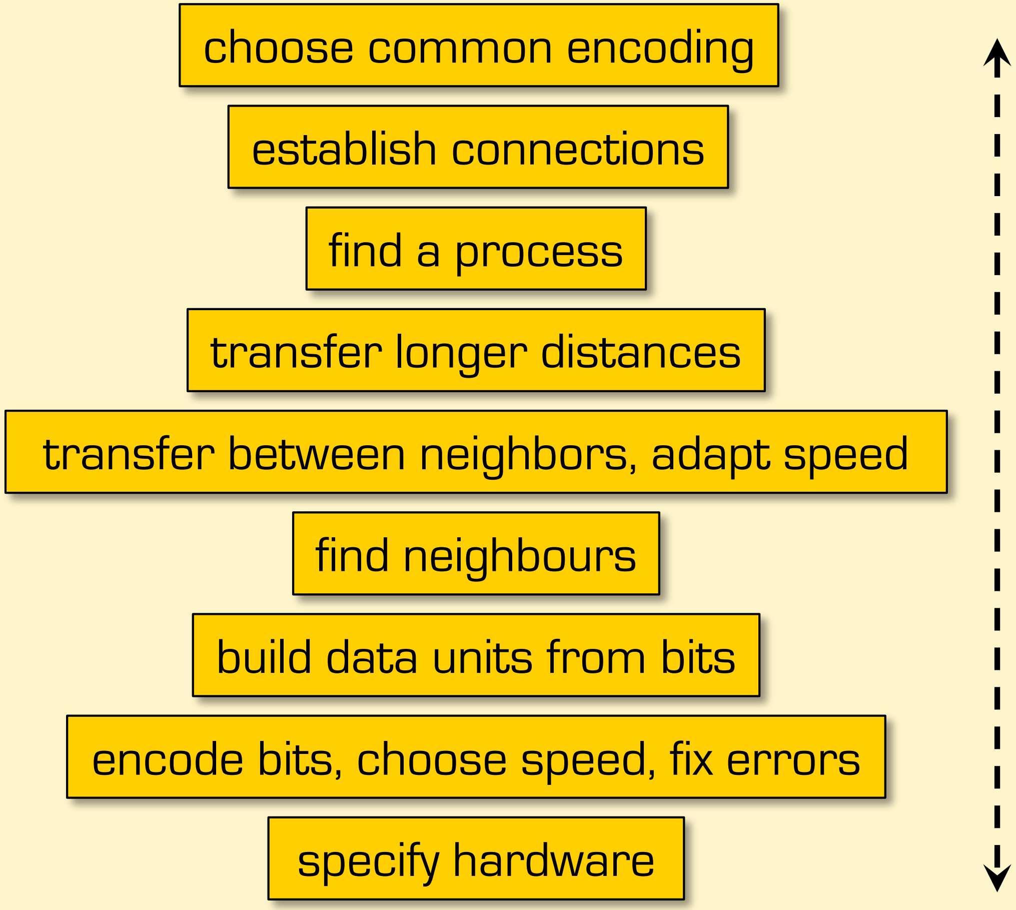
Network structures

Structuring the tasks

Monday, March 8, 2021

Approaches to structure the tasks

layered approach: arrange tasks in layers


choose common encoding

establish connections

find a process

transfer longer distances

transfer between neighbors, adapt speed

find neighbours

build data units from bits

encode bits, choose speed, fix errors

specify hardware

advantages:

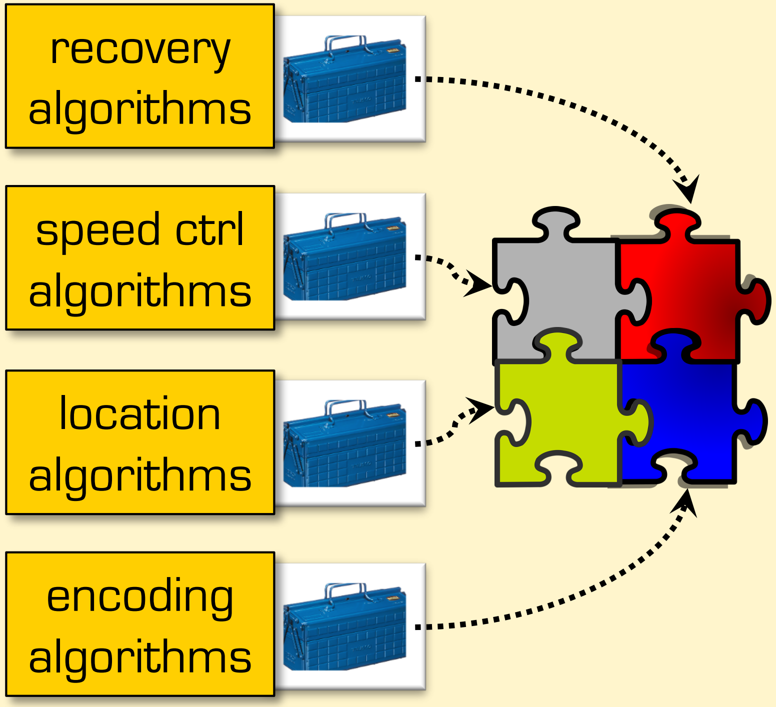
- clear interfaces
- clear assignment of responsibilities
- develop layers independently

disadvantages:

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

Approaches to structure the tasks

component approach: interacting components



recovery algorithms

speed ctrl algorithms

location algorithms

encoding algorithms

advantages:

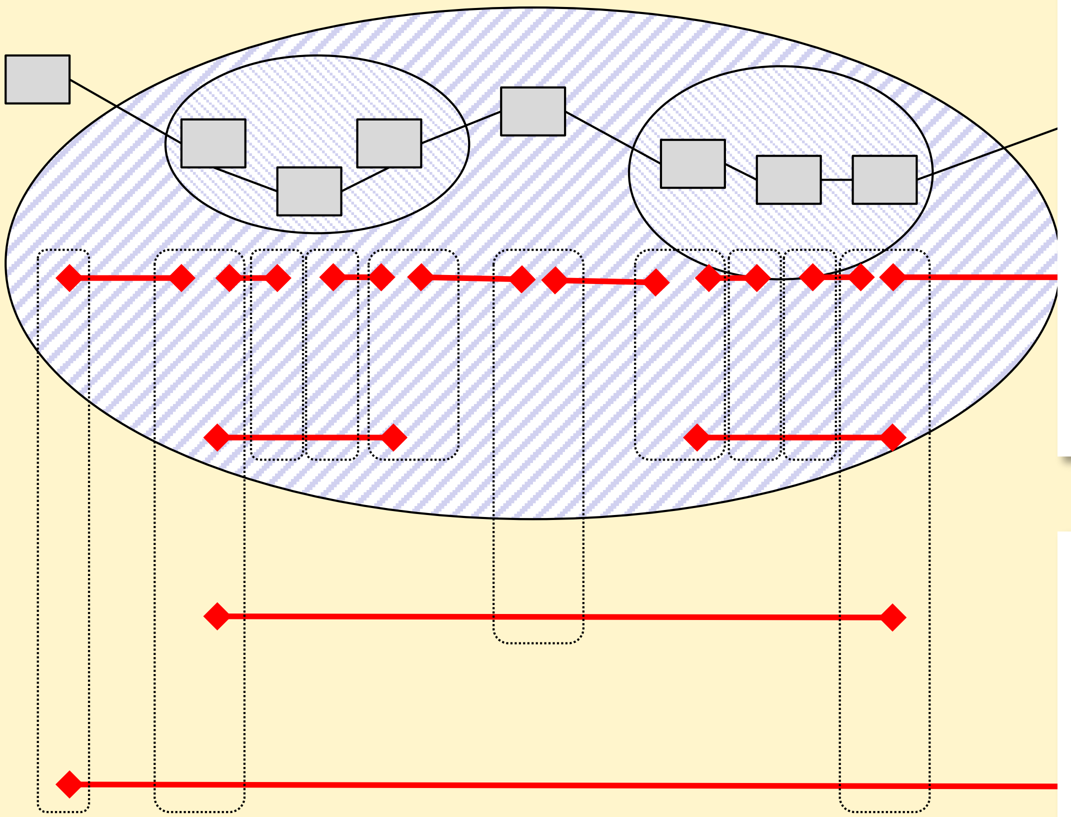
- possible to avoid duplicated functions
- possible to choose perfect network behaviour for every application

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

IN2140:
Introduction to Operating Systems and Data Communication

Network structures

Layering model

Monday, March 8, 2021

Approaches to structure the tasks

Right now, the layered models dominate

ISO OSI (Open Systems Interconnection) Reference Model

and

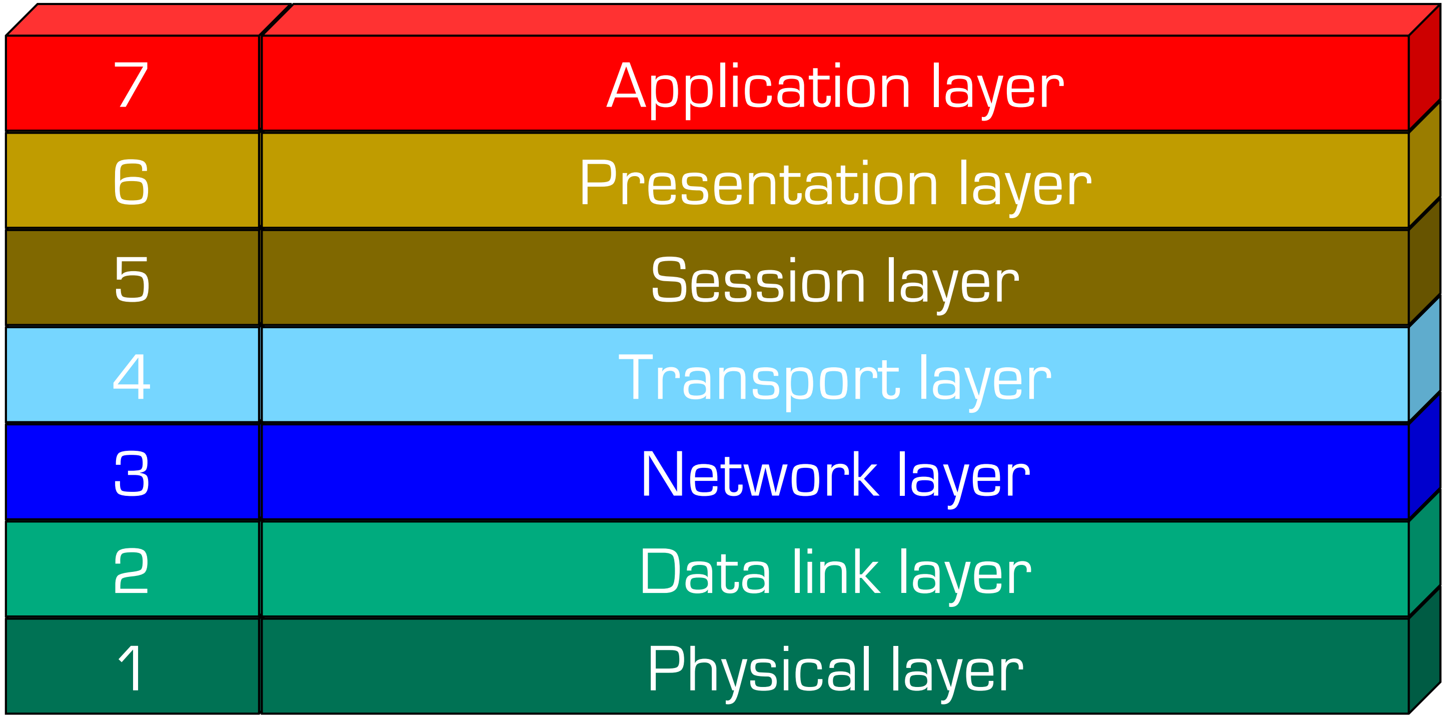
TCP/IP Reference Model Internet Architecture

- 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



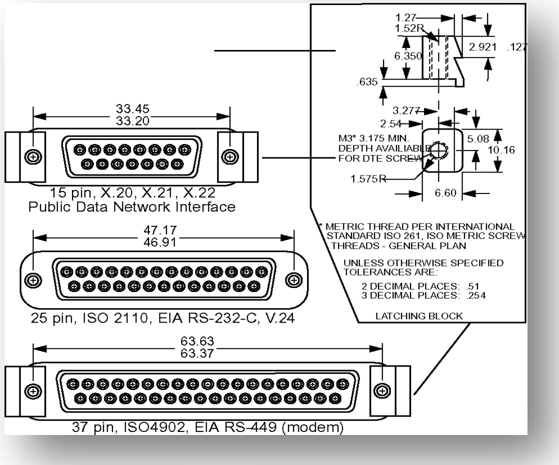
7	Application layer
6	Presentation layer
5	Session layer
4	Transport layer
3	Network layer
2	Data link layer
1	Physical layer

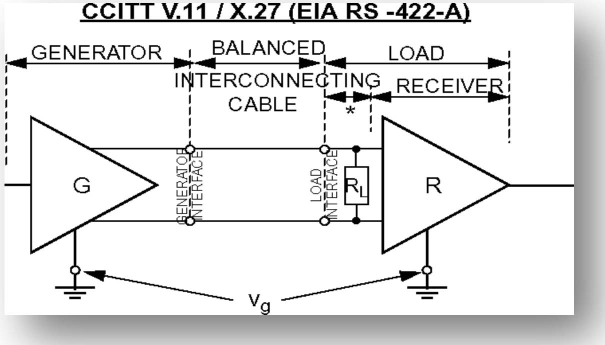
Layer functions: physical layer

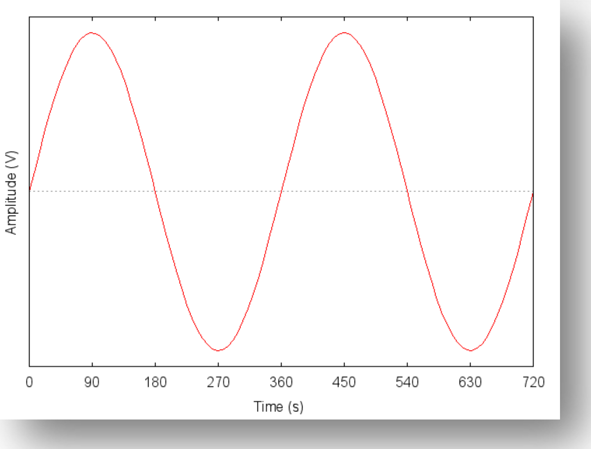
Responsibility: insecure bitstream between adjacent systems

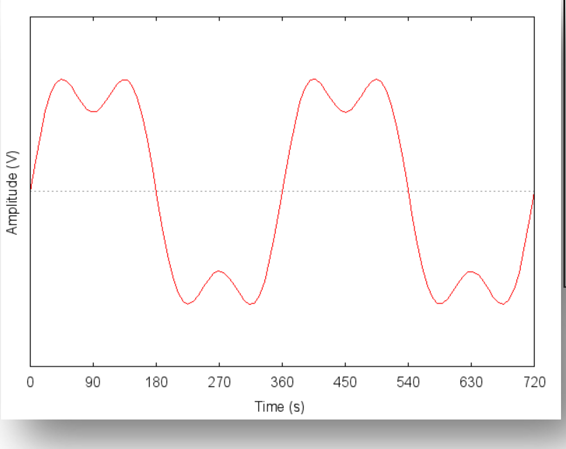
- mechanics
- electronics
- procedural

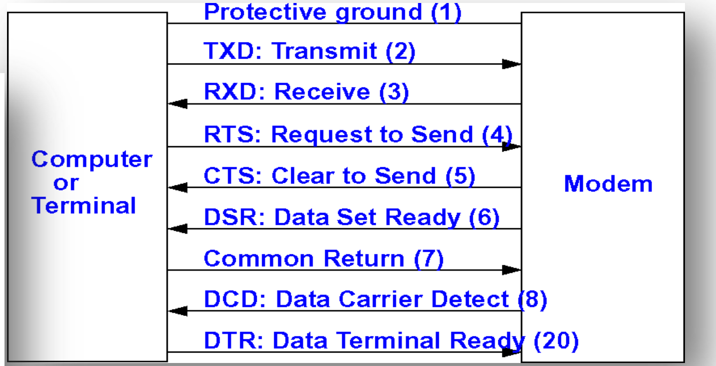
encoding and decoding of bits


33.45
33.20
15 pin, X.20, X.21, X.22
Public Data Network Interface
47.17
46.91
25 pin, ISO 2110, EIA RS-232-C, V.24
63.63
63.37
37 pin, ISO4902, EIA RS-449 (modem)
1.27
1.52R
6.350
2.921
.127
.635
3.277
2.54
M3* 3.175 MIN.
DEPTH AVAILIABLE
FOR DTE SCREW
5.08
10.16
1.575R
6.60
* METRIC THREAD PER INTERNATIONAL STANDARD ISO 261, ISO METRIC SCREW THREADS - GENERAL PLAN
UNLESS OTHERWISE SPECIFIED TOLERANCES ARE:
2 DECIMAL PLACES: .51
3 DECIMAL PLACES: .254
LATCHING BLOCK









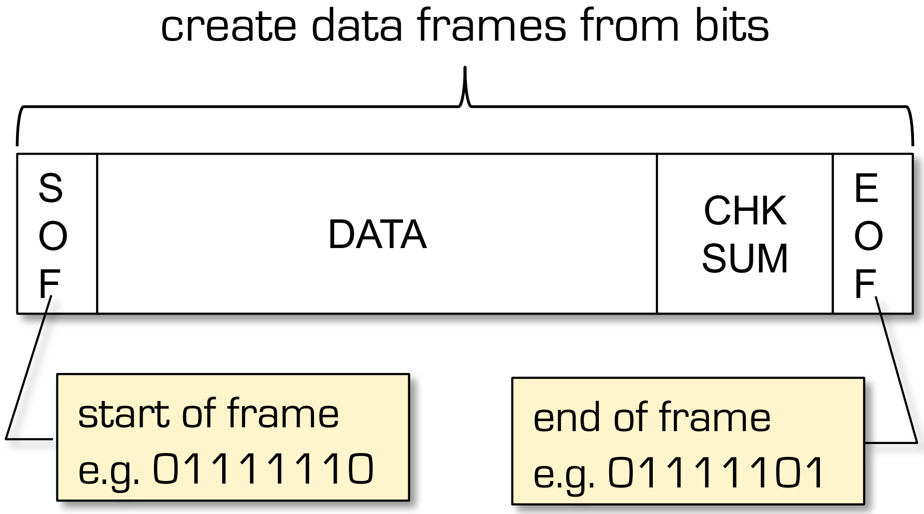
Protective ground (1)
TXD: Transmit (2)
RXD: Receive (3)
RTS: Request to Send (4)
CTS: Clear to Send (5)
DSR: Data Set Ready (6)
Common Return (7)
DCD: Data Carrier Detect (8)
DTR: Data Terminal Ready (20)

1 Physical layer

Layer functions: data link layer

Responsibility: error-recovering frame stream, adjacent systems

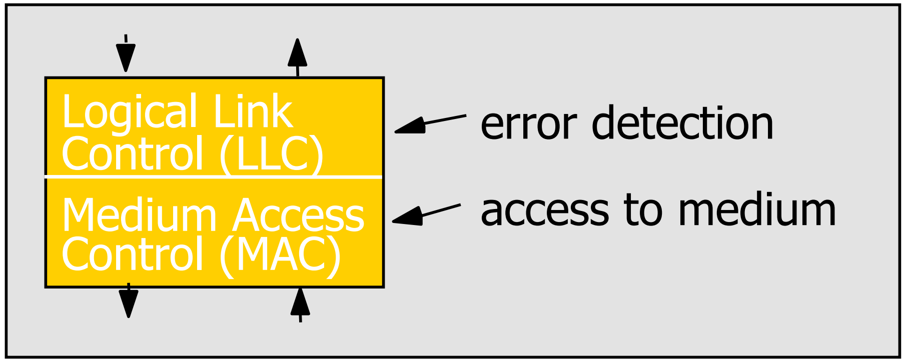
Reliable data transfer between adjacent stations with frames


create data frames from bits

start of frame
e.g. 01111110

end of frame
e.g. 01111101

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

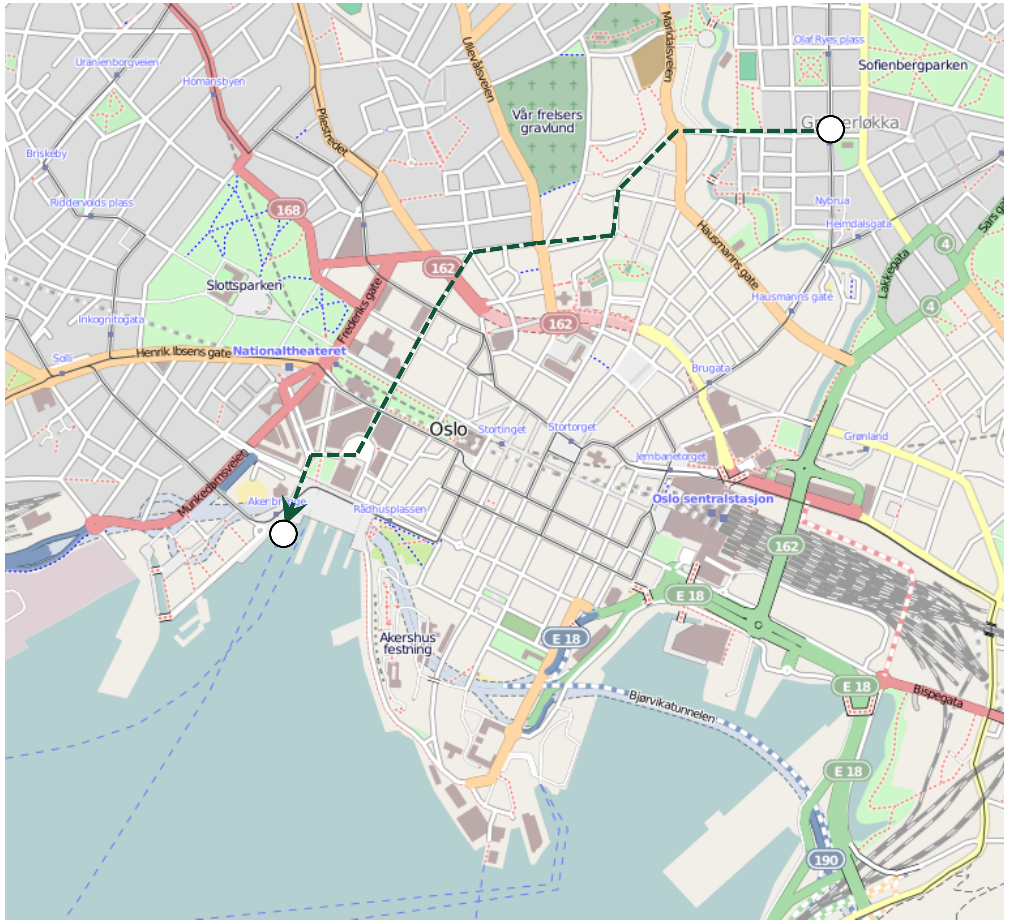


where to send? → MAC address
when to send? → wait for silence, wait for token
correct frame? → check for bit errors and repair

2 Data link layer

Layer functions: network layer

Responsibility: packet stream between end systems



end-to-end transport of packets

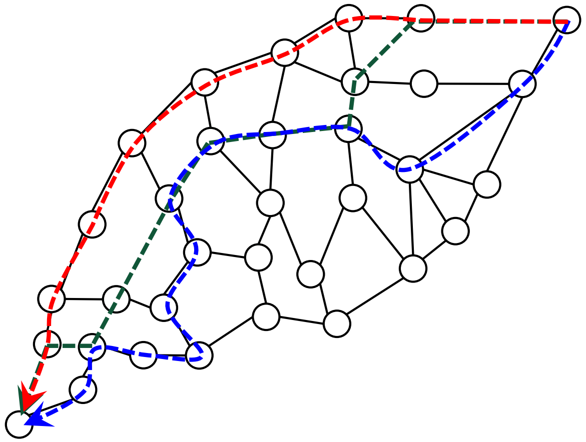
ability to address source and target nodes

routing from source node to target node

3 Network layer

Layer functions: network layer

Responsibility: packet stream between end systems

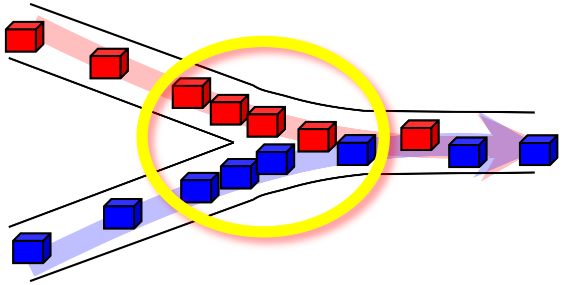


end-to-end transport of packets

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

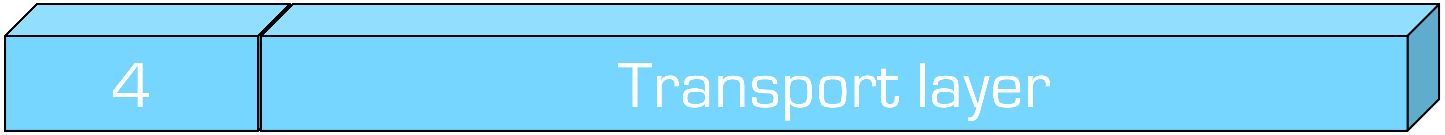
wide range of services offered to L5:

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 strings
- ASCII
- UTF-8
- Unicode
- EBCDIC

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

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

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

image **formats**
- JPG, PNG

compression
- zip, gzip, bzip2

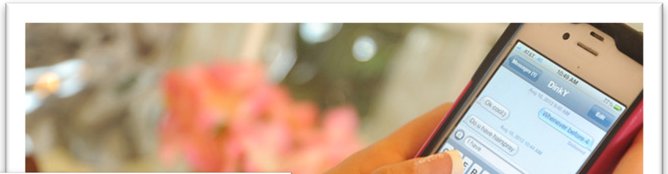
encryption methods
- PGP, S/MIME

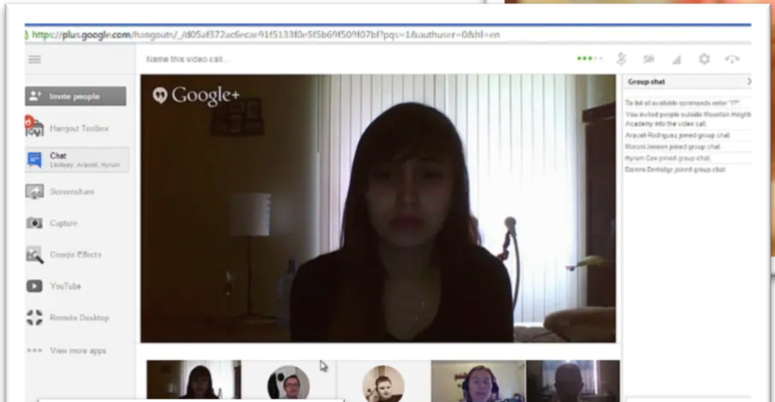
semantics
- NOK, EUR, USD

6 Presentation layer

Layer functions: application layer

Responsibility: cooperating entities





To list all available commands enter "/?"

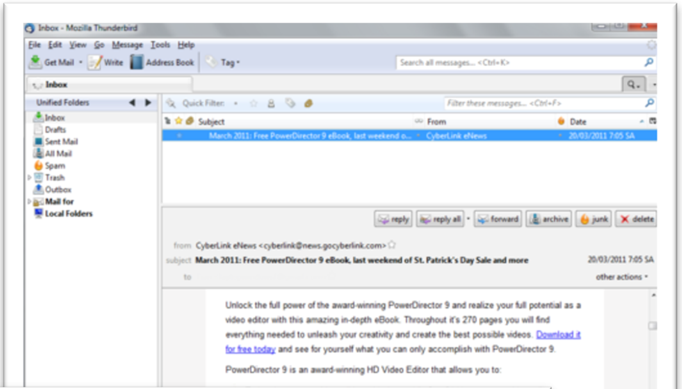
You invited people outside Mountain Heights Academy into the video call.

Araceli Rodriguez joined group chat.

Moroni Jensen joined group chat.

Hyrum Cox joined group chat.

Darren Dethidge joined group chat.



Subject	From	Date
March 2011: Free PowerDirector 9 eBook, last weekend o...	CyberLink eNews	20/03/2011 7:05 SA

from CyberLink eNews <cyberlink@news.gocyberlink.com>

subject **March 2011: Free PowerDirector 9 eBook, last weekend of St. Patrick's Day Sale and more** 20/03/2011 7:05 SA

to

Unlock the full power of the award-winning PowerDirector 9 and realize your full potential as a video editor with this amazing in-depth eBook. Throughout it's 270 pages you will find everything needed to unleash your creativity and create the best possible videos. Download it for free today and see for yourself what you can only accomplish with PowerDirector 9.

PowerDirector 9 is an award-winning HD Video Editor that allows you to:

Manchester 17°
H: 19° L: 9°

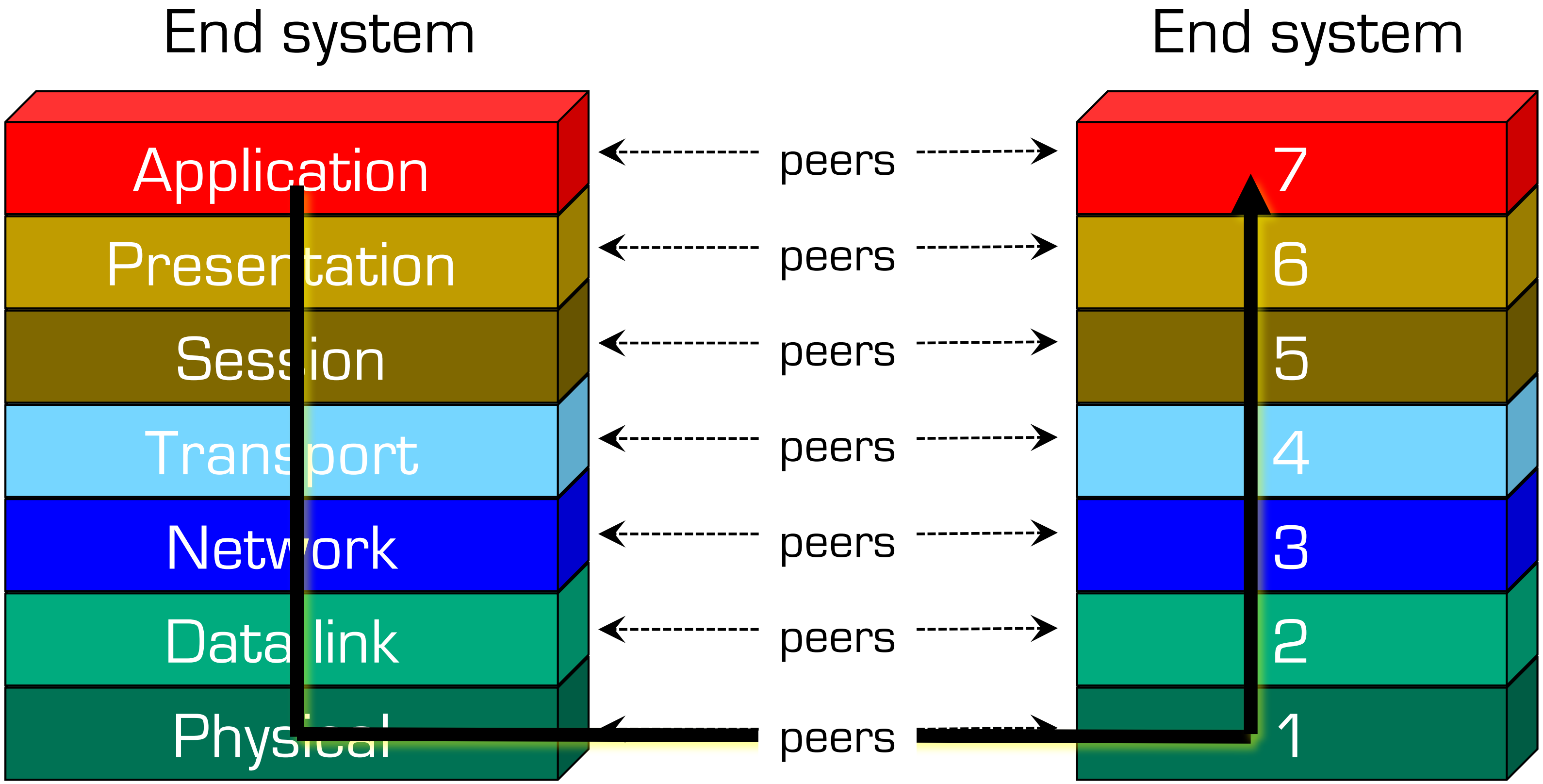
WEDNESDAY	19°	9°
THURSDAY	19°	8°
FRIDAY	20°	9°
SATURDAY	17°	10°
SUNDAY	17°	9°
MONDAY	17°	9°

Updated 09/09/2009 17:57

7
Application layer

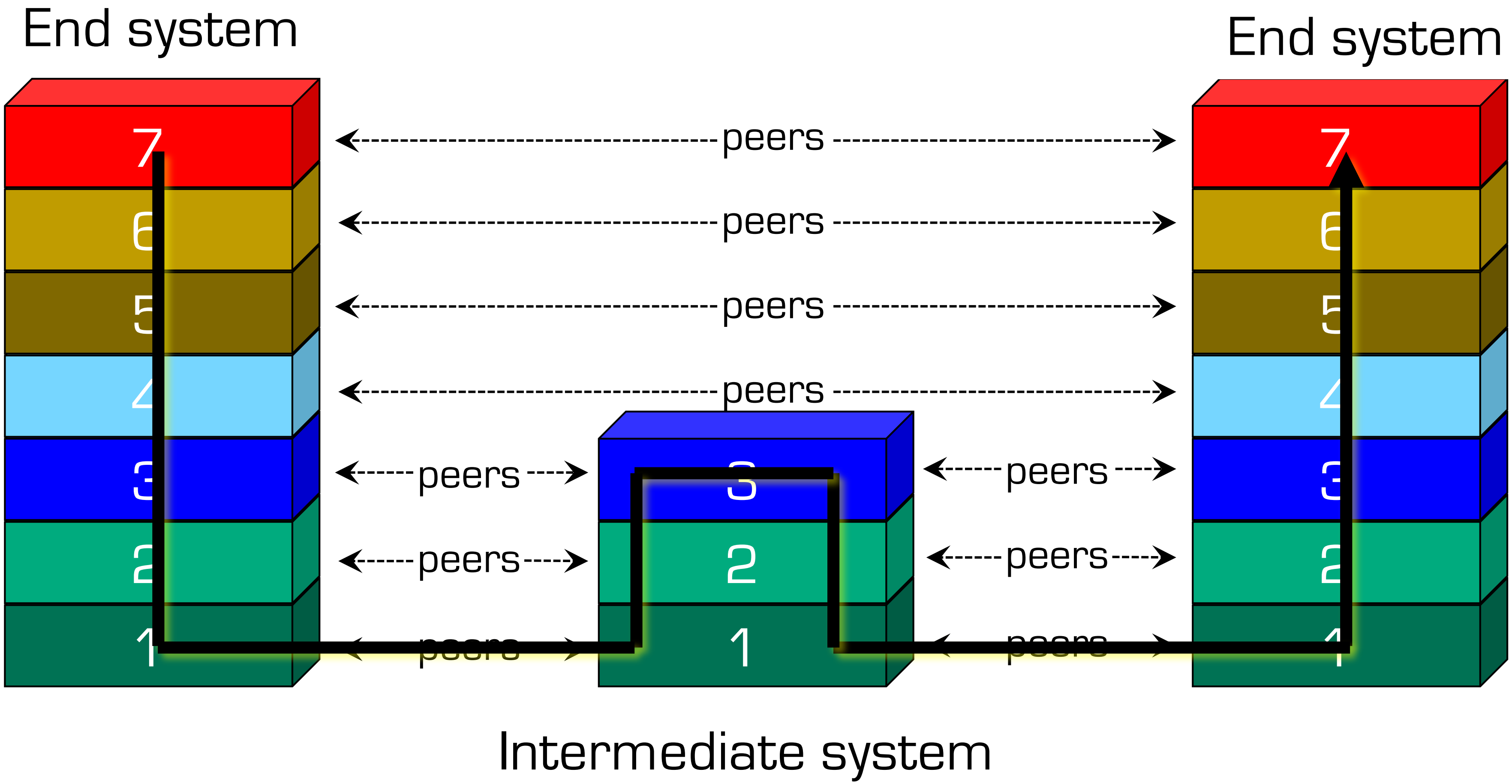
Architecture

Data flow between two adjacent systems



Architecture

Data flow between two non-adjacent systems



IN2140:
Introduction to Operating Systems and Data Communication

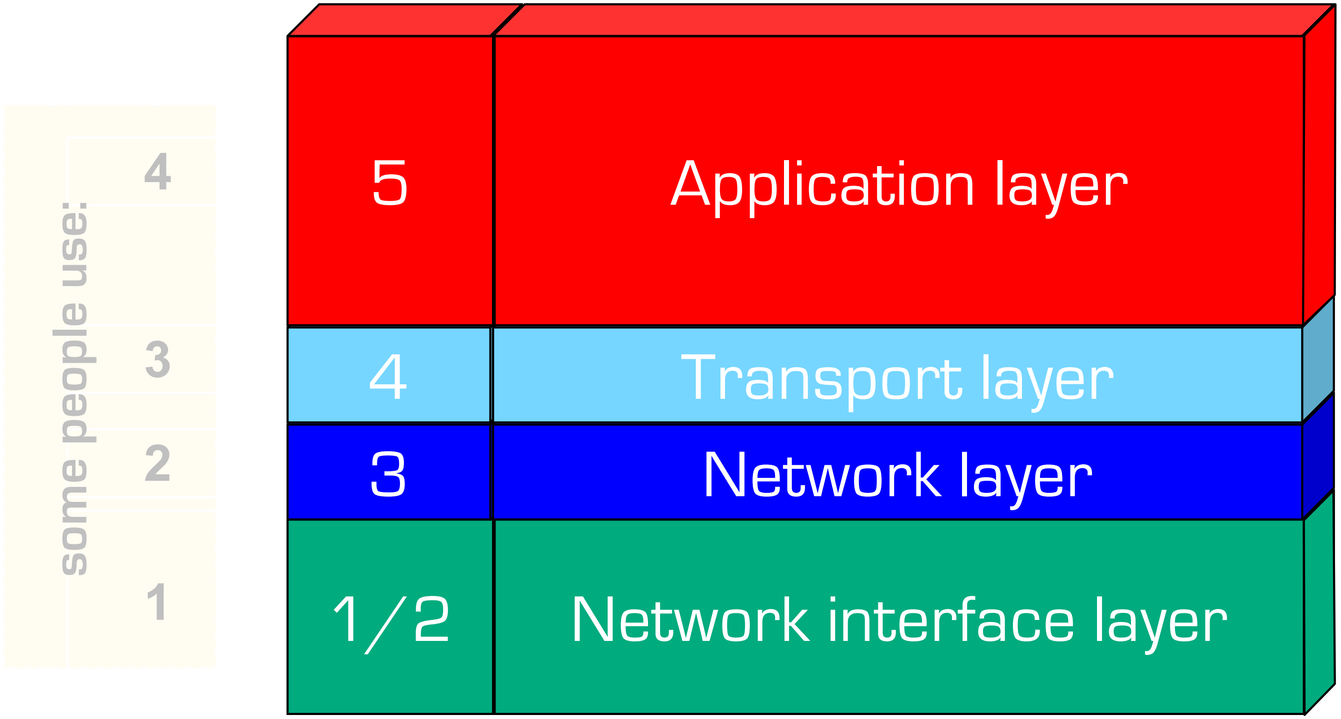
Network structures

TCP/IP model

Monday, March 8, 2021

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

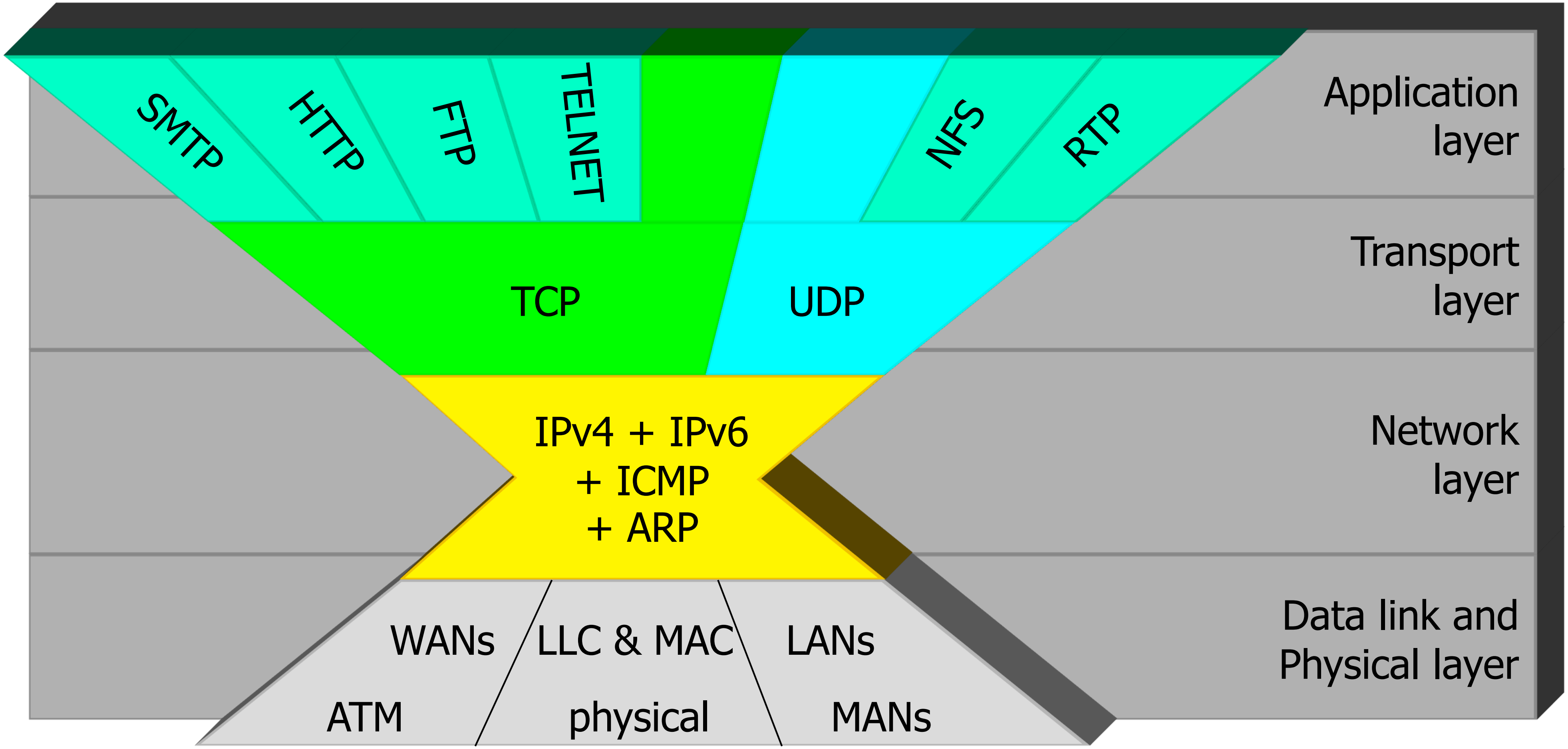
Why no clear separation of upper layers?

- 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”