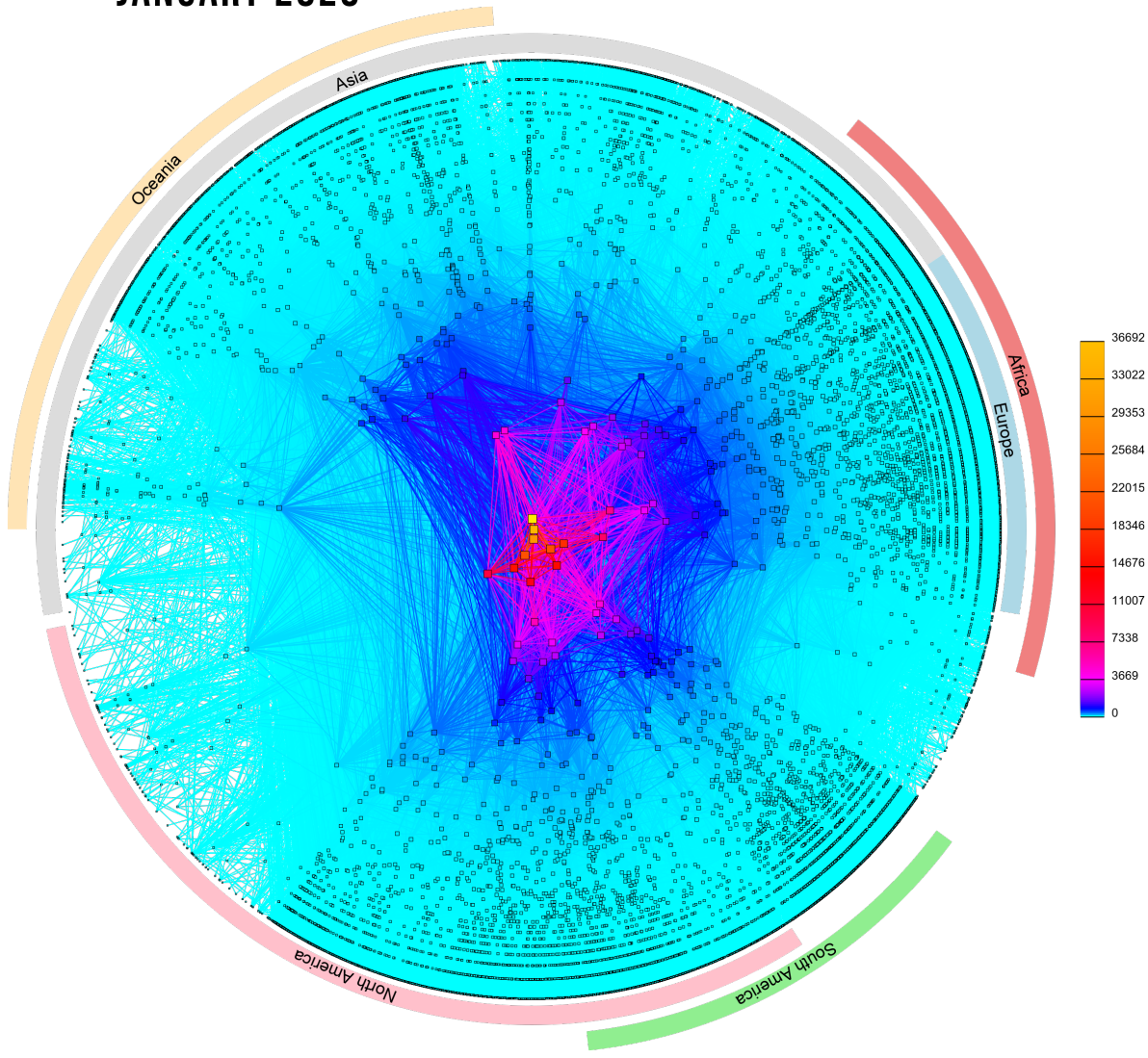
UE PRIP
Principes des réseaux informatiques par la pratique
Introduction

Isabel Amigo

2022

CAIDA'S IPV4 AS CORE GRAPH
JANUARY 2020



source https://www.caida.org/projects/as-core/2020/

Today's objective

- get "feel" and some terminology
- more depth and detail later in course

Approach:

1. Introductory course using Internet as example
2. Individual quizz
3. Labs/TPs

Today's roadmap

1. what's the Internet?
2. what's a protocol?
3. network edge: hosts, access net, physical media
4. network core: packet/circuit switching
5. protocol layers, service models, encapsulation
6. Internet's structure

1. What's the Internet?

What's the Internet?

A. "Construction" point of view (describing elements)
B. Service point of view

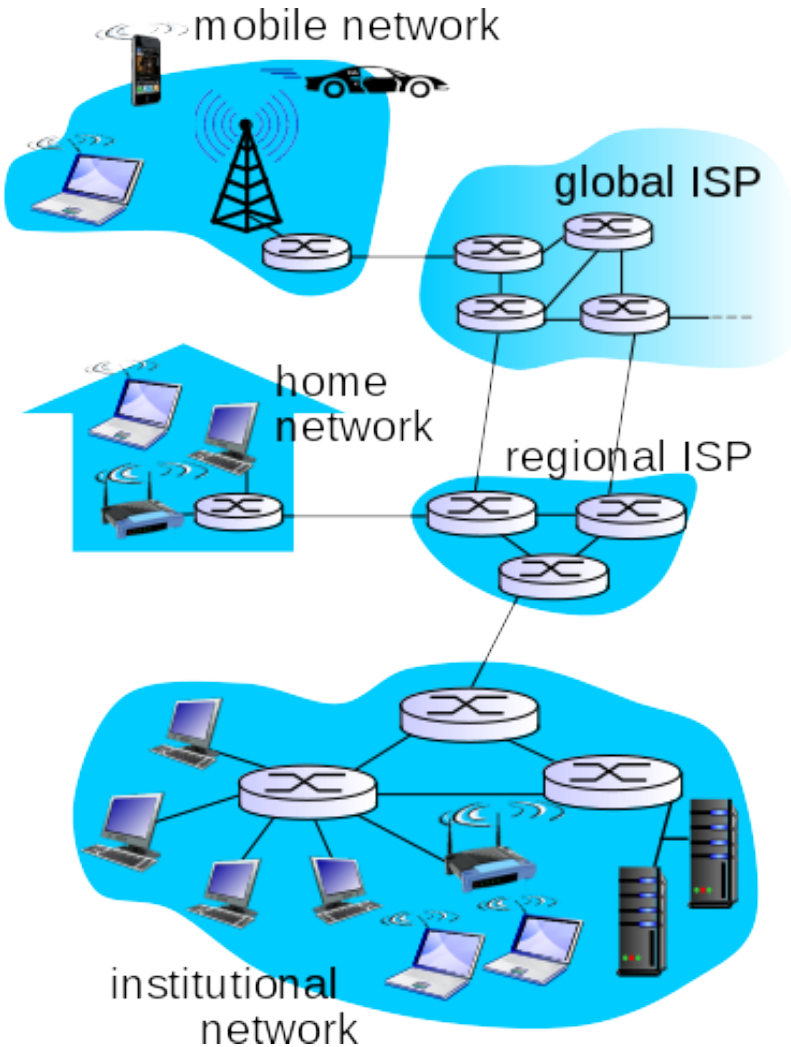
What's the Internet?

A. "Construction" point of view (describing elements)

- millions of connected computing devices: hosts running network apps
- communication links: fiber, copper, radio, satellite transmission rate: *bandwidth*
- routers: forward packets (chunks of data)

What's the Internet?

A. "Construction" point of view (describing elements)



What's the Internet?

A. A "construction" point of view (describing elements)

Some characteristics:

- Internet: "network of networks" loosely hierarchical
- protocols e.g., TCP, IP, HTTP, Skype, Ethernet
- Internet standards
  - RFC: Request for comments
  - IETF: Internet Engineering Task Force
  - IEEE

Some questions:

Q public or private?

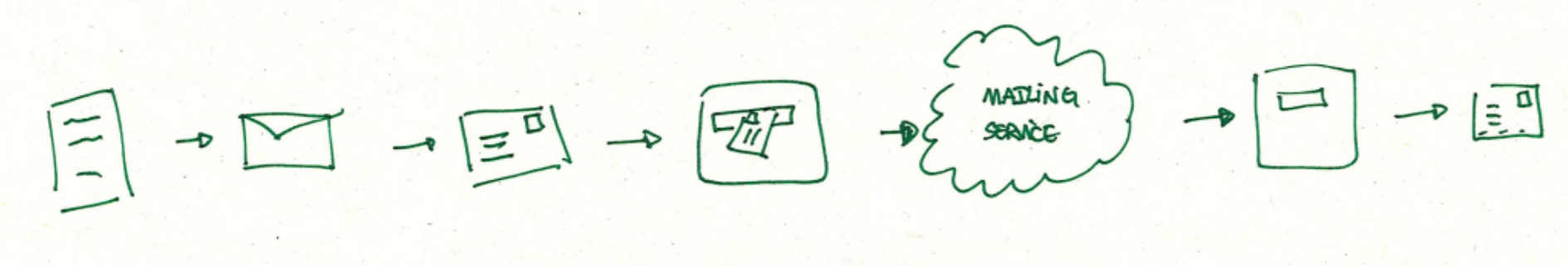
Q who governs Internet?

What's the Internet?

B. A services view

- A "thing" that allows as to use distributed applications
  - **Q** where do these applications run?
  - **Q** what are these applications?
- To use this "thing", we have to respect some "rules "
  - analogy: mail (physical)
  - APIs and Protocols (next slides)

thing = communication infrastructure
"rules" = APIs and protocols



2. What's a protocol?

What's a protocol?

Human protocols:

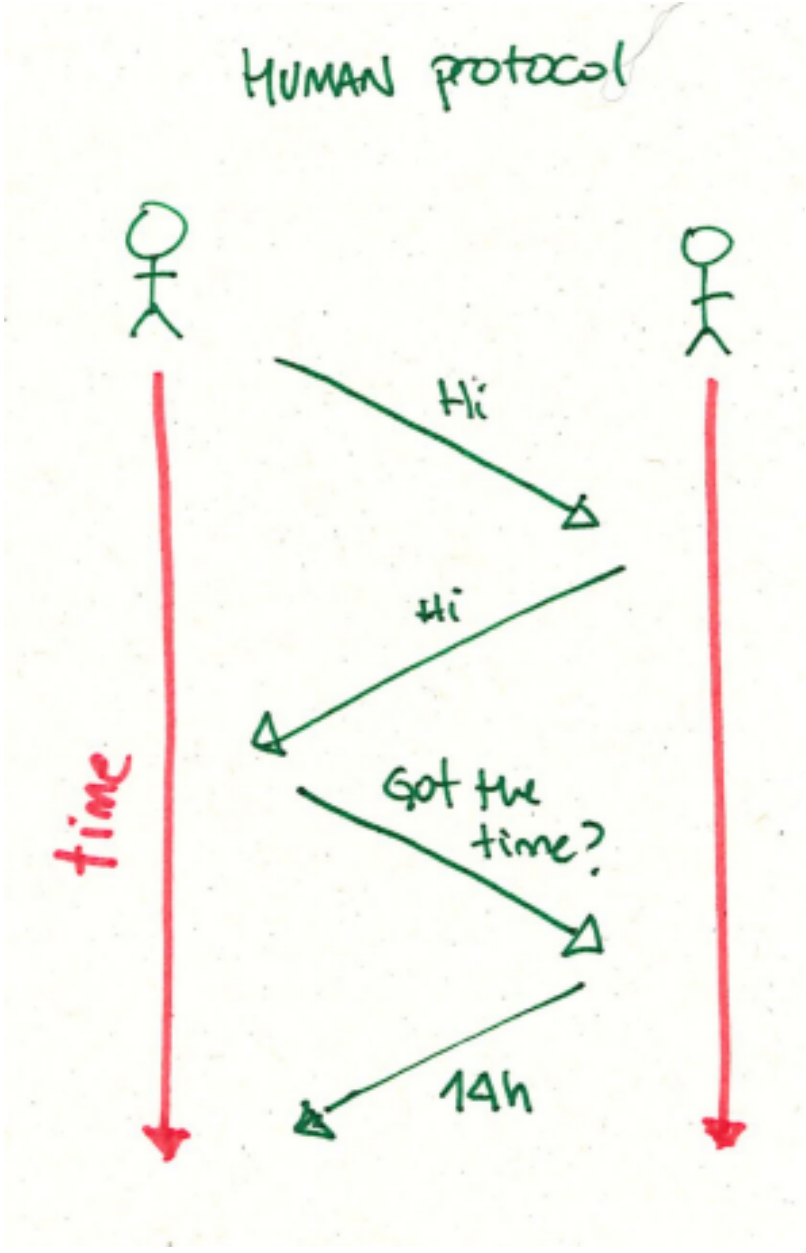
- introductions "what's the time?" "I have a question"
- specific msgs sent
- specific actions taken when msgs received, or other events

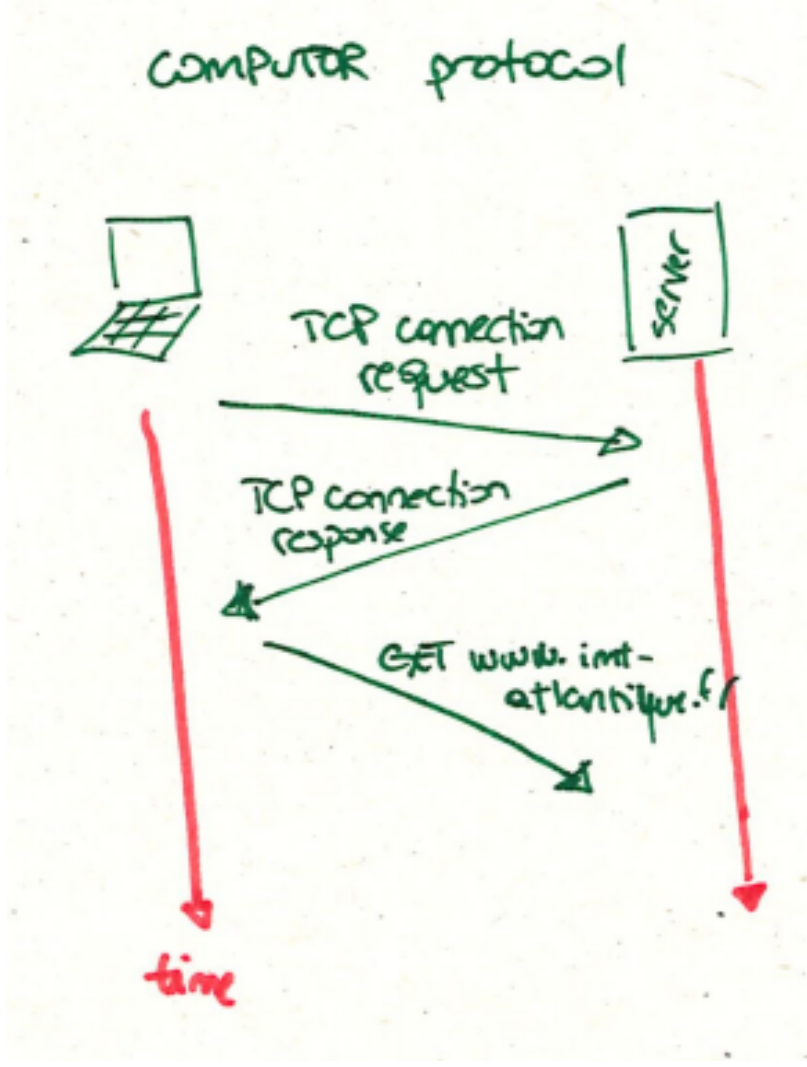
Network protocols:

- analogue to human protocols but
- machines rather than humans
- all communication activity in Internet governed by protocols

What's a protocol?

Examples



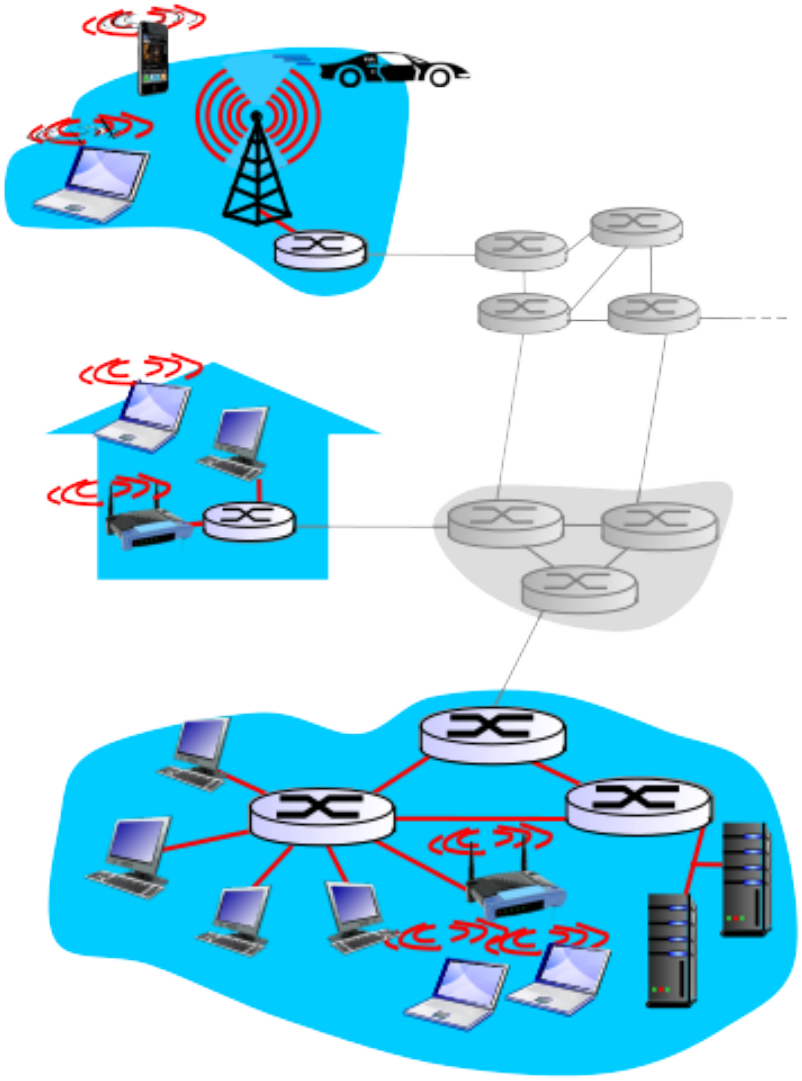


What's a protocol?

Protocols define format, order of msgs sent and received among network entities, and actions taken on msg transmission and reception.

3. Network edge

Network Edge

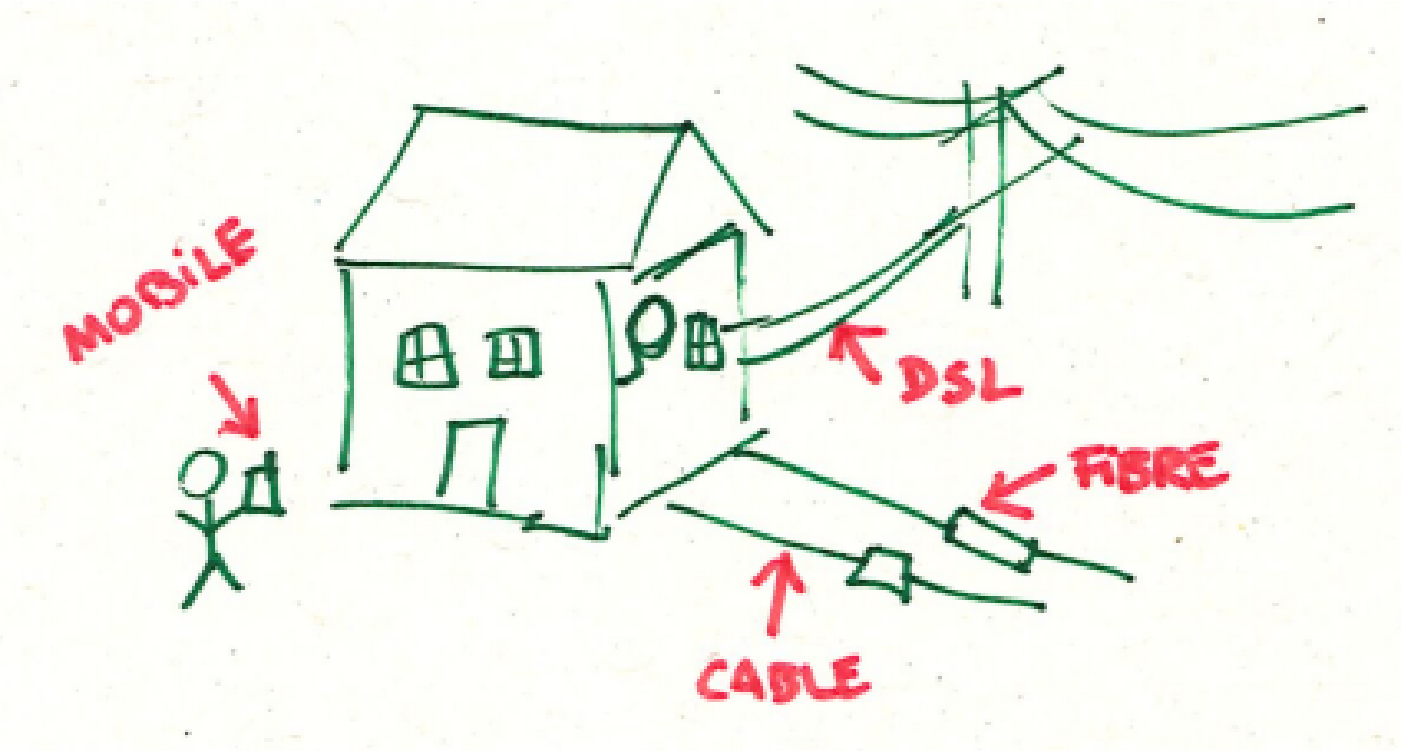


Network Edge

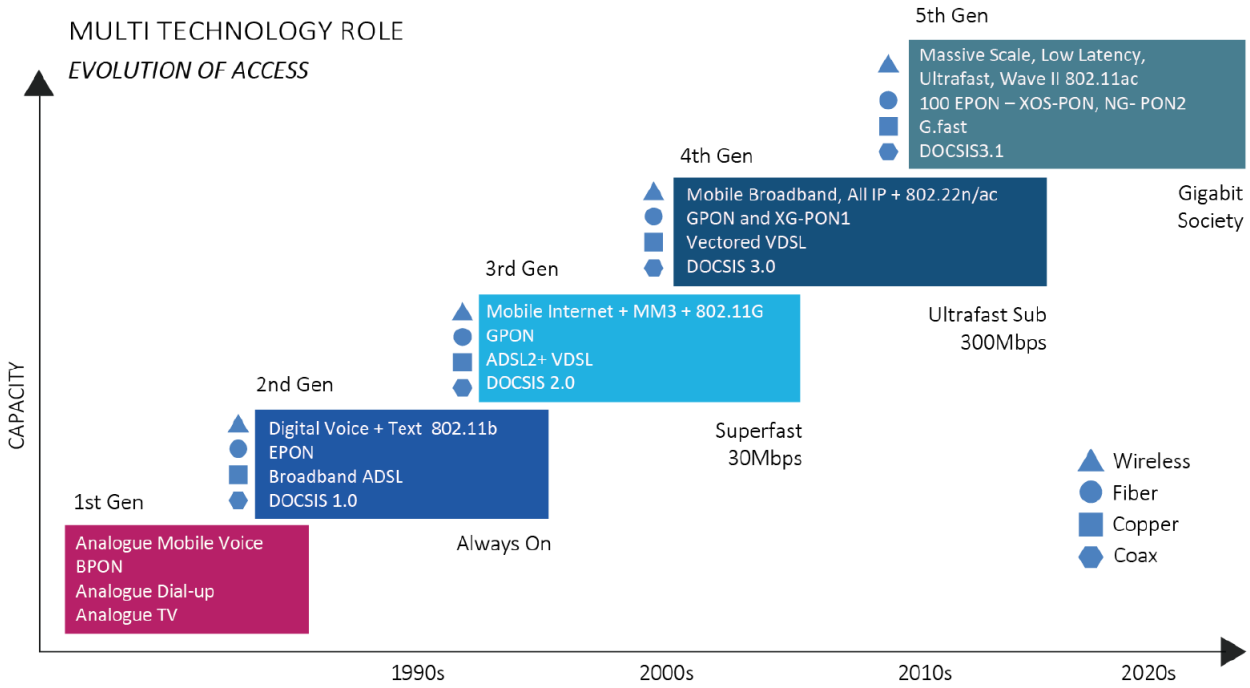
What is in the networks' edge?

- end systems
  - run applications (e.g. email, web,..)
- access networks
  - residential
  - institutional
  - mobile
- access technologies
  - physical media: wireless, wired, fiber
  - evolutions e.g. 1G, 2G .. VDLS, VDSL2, IEEE 802.11 a, b ... ax

Broadband access technologies



Broadband access technologies

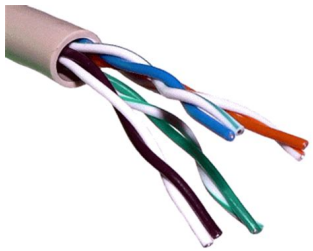


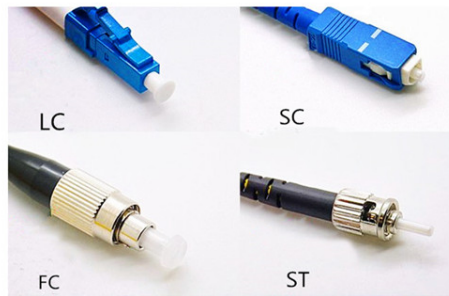
Source: ADTRAN relayed by `http://reports.broadbandcommission.org/`

Keep in mind! Bandwidth (bps)? dedicated or shared?

Physical media

Copper, Coax, Fiber, Air





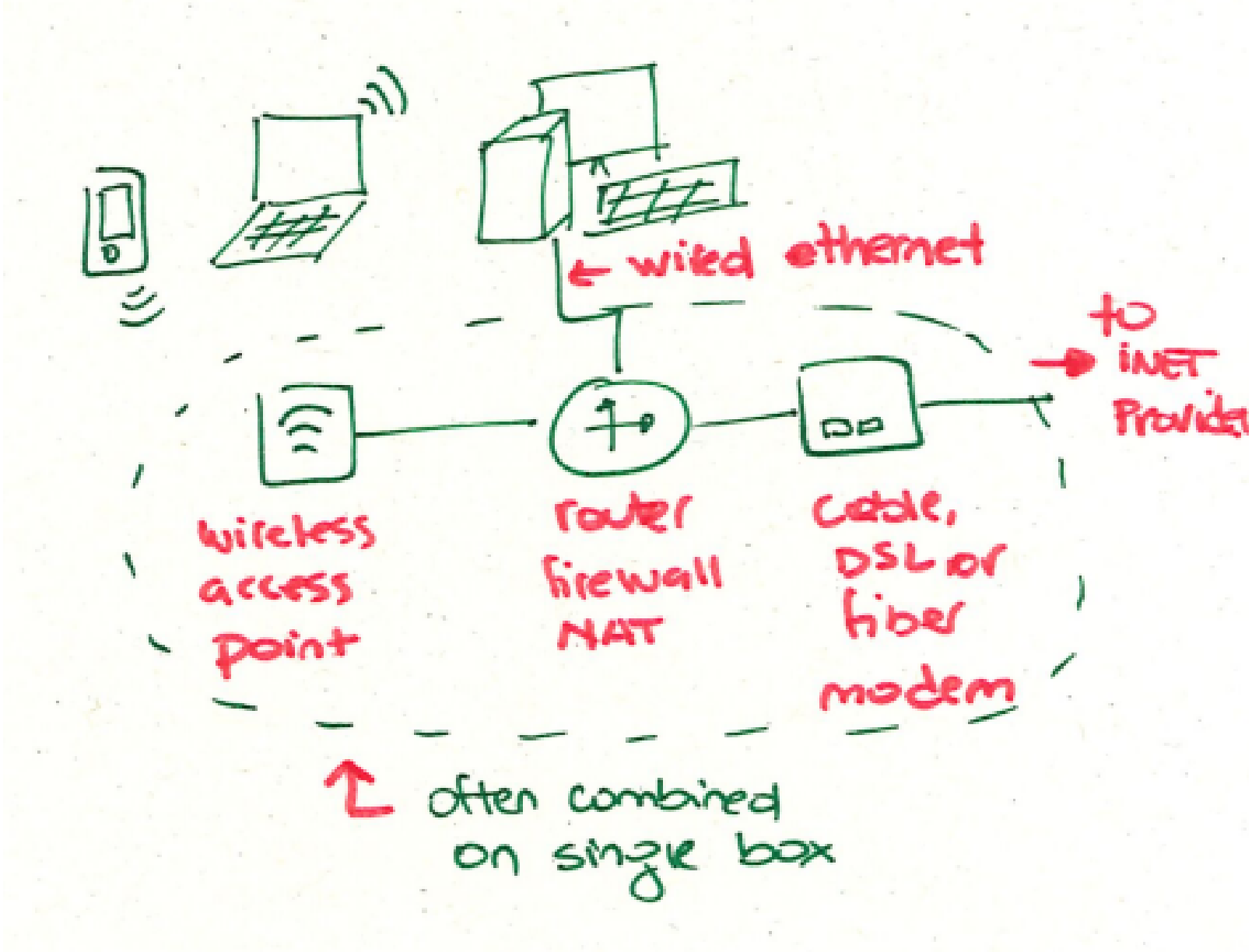
Twisted Pair (copper), RJ45 connector for twisted pair, RJ45 and Coaxial connectors, Ethernet NIC (RJ45), Coaxial NIC, Fiber connectors

Access networks

Mobile access

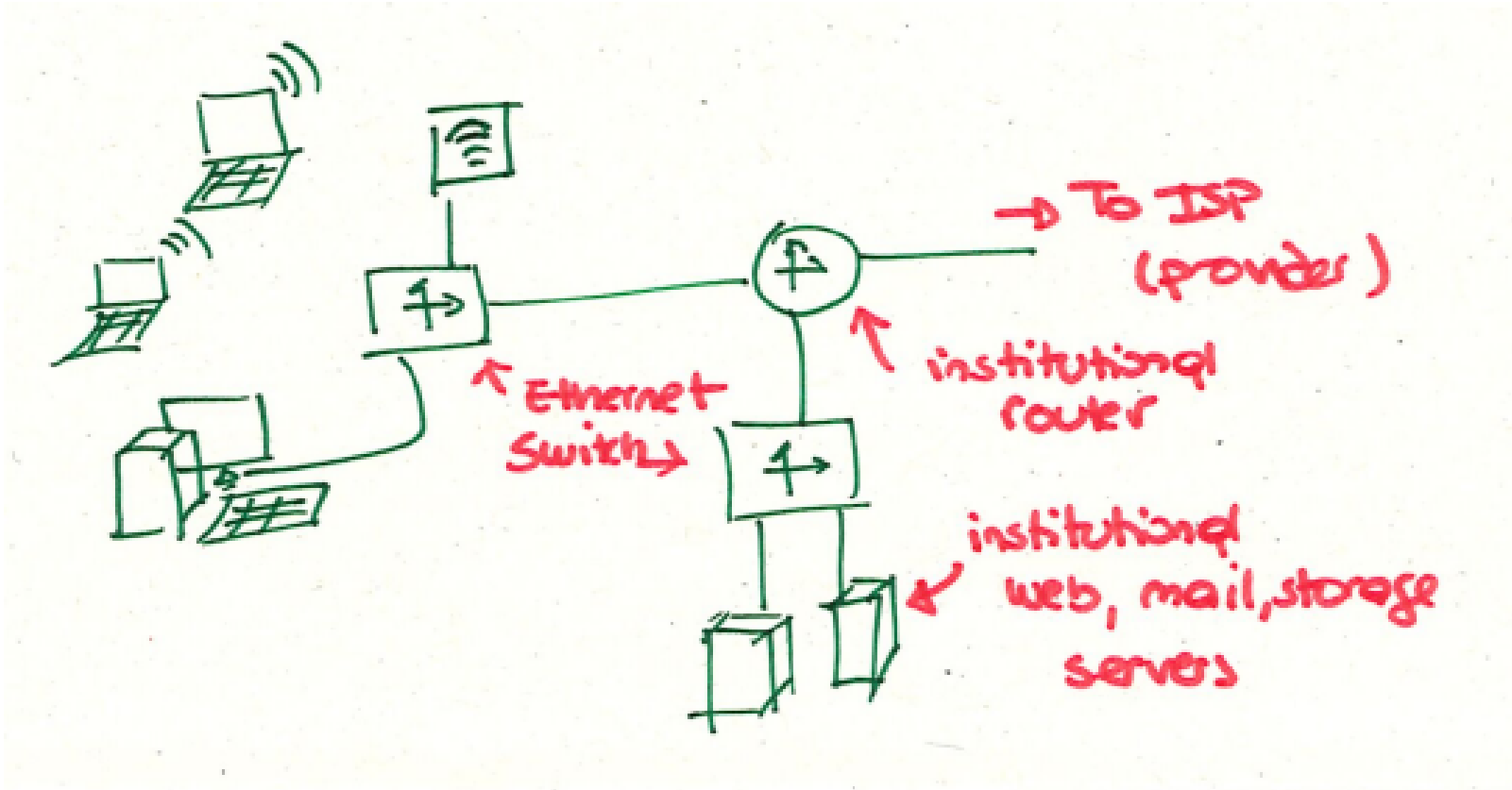
Access networks

Home/ Residential network



Access networks

Institutional network



End hosts

End host:

- takes applications' message and breaks it into chunks of data (packets) of say length L bits
- sends the packet through a network interface at transmission rate R bps
  - transmission rate also known as bandwidth also known as capacity

Transmission delay (D) is thus given by L and R.

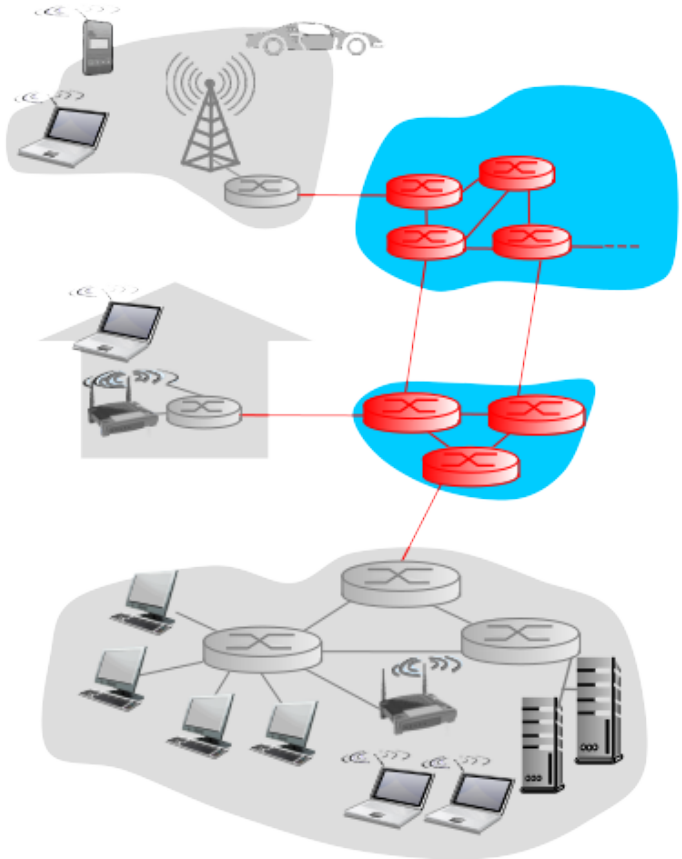
$$D = \frac{L \text{ (bits)}}{R \text{ (bits/second)}}$$

! There are other sources of end-to-end delay: propagation, queuing, processing, we'll see it later!

4. Network Core

Network Core

Packet switching core



- we usually talk of aggregation network and core network, but today we are simplifying
- mesh of interconnected *routers*

4. Network Core

Routers forward packets from *one router to the next*, across links based on packet destination, each packet transmitted at full link capacity

How is the *next router* determined?

Some key vocabulary: routing and forwarding

Routing function: determines source-destination route taken by packets on a *per-hop* basis

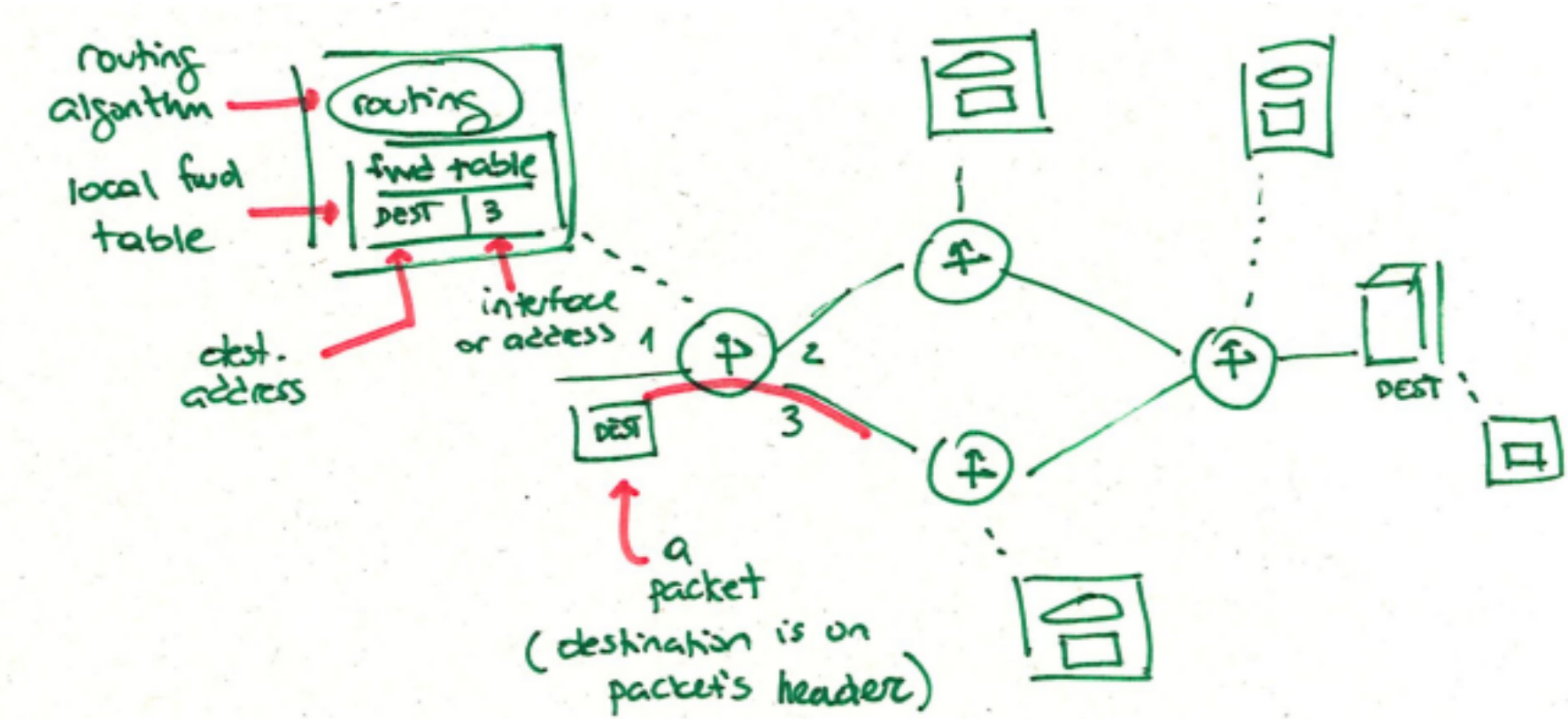
- The routing function, thanks to a routing algorithm, determines the routing table.

Forwarding function: moves packets from router's input interface to appropriate router outputs interface

- The forwarding function, uses such table to decide through which output interface send the packet

! Both functions performed at each router.

Routing and Forwarding functions



Store-and-forward packet-switching

Store and forward: entire packet must arrive at router before it can be transmitted on next link

- ! introduces a delay of $D = L/R$ seconds at each *hop* (where as before L is the length of the packets in bits and R the interface's capacity in bps)
- ! This is how Internet works nowadays

Alternative Core: Virtual circuit

End-end resources allocated to, reserved for, "call" between source, dest.

- dedicated resources: no sharing
  - circuit-like (guaranteed) performance
  - resource piece remains idle if not used by owning call (no sharing)
- dividing link bandwidth into "pieces"
  - frequency division
  - time division

Example: telephony network

Circuit-switched vs Packet switching networks

Analogy: restaurant with and without reservations.

Restaurant with reservations:
☺You arrive and are served immediately
☹You have to call and reserve before going
☹If you don't show, the restaurant wastes a table

Restaurant without reservations:
☹You might have to wait on arrival
☺You don't have to call in advance to reserve
☺Restaurant doesn't take the risk of wasting a table

Statistical multiplexing

Packet switching allows more users to use network!

Example

- 1 Mbps link
- each user is active (i.e. sends data) 10% of the time
- each user when active sends data at rate 100kbps

Q How many users can be supported?

- Circuit-switching: 10 users
- Packet-switching: With 35 users probability of more than 10 users transmitting at the same time is ~ 0.0004

Q How did we get value 0.0004?
Q What happens if there are more than 35 users?

Is packet-switching *better* than circuit-switching?

☺Packet-switching is great for bursty data

- resource sharing
- simpler, no call setup

☹No performance guarantees, excessive congestion possible: packet delay and loss

- protocols needed for reliable data transfer and congestion control

Q Examples of applications generating bursty and non-bursty data?

Q How to provide bandwidth guarantees, needed for some applications, on packet-switching networks?

- several "patches" exist, but till a research problem!

A word on end-to-end packet delay

Packet-switched networks experience different source of delays:

- **Transmission** at each node
- **Propagation** through physical media
- **Queuing** at nodes
- **Processing** at nodes

5. Layers model

Protocol Layers

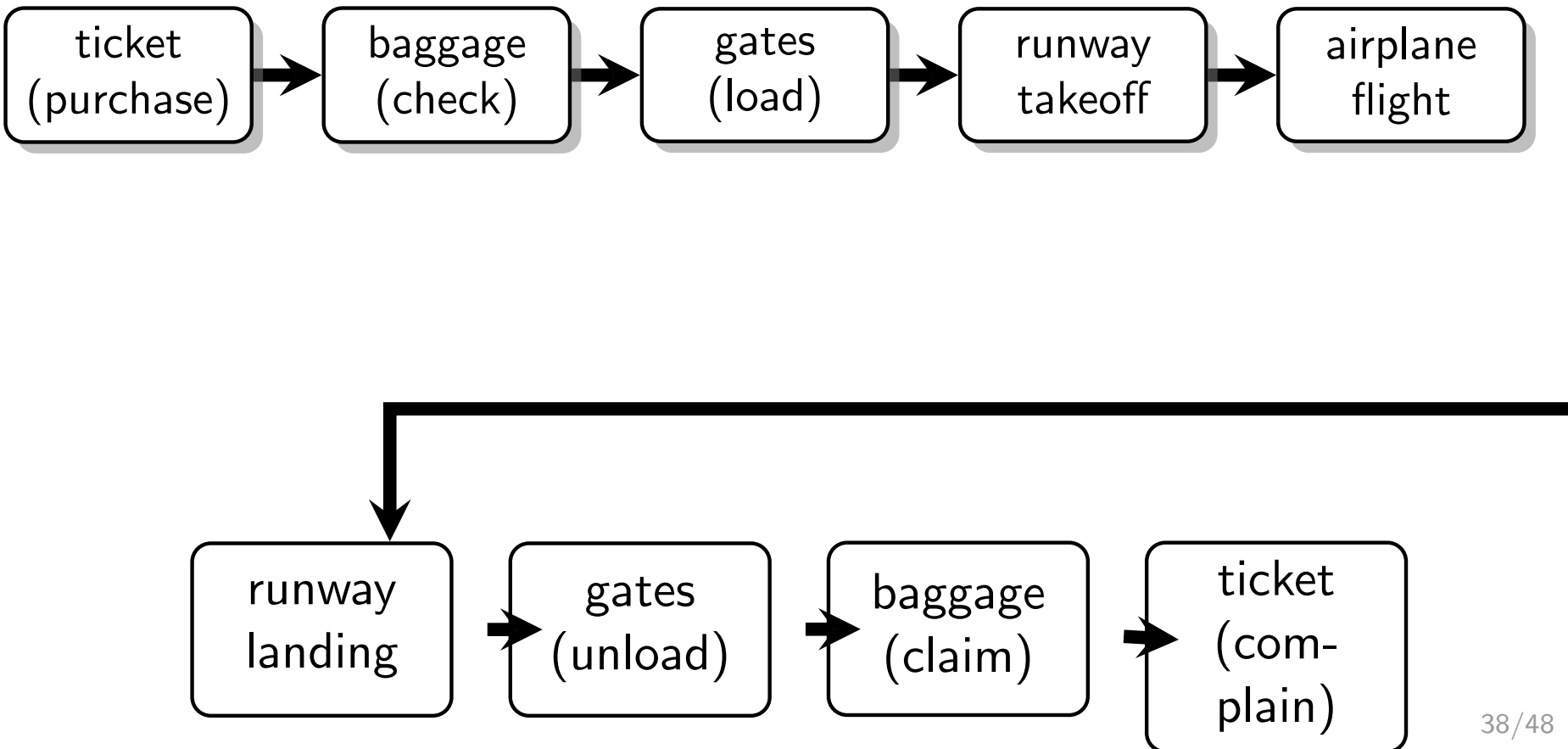
Networks are complex!

Q How can we organize the discussion to make it simpler?

Q How can we organize such systems to make them simpler, easier to develop, as little as possible prune to errors, upgradeable, scalable, ...?

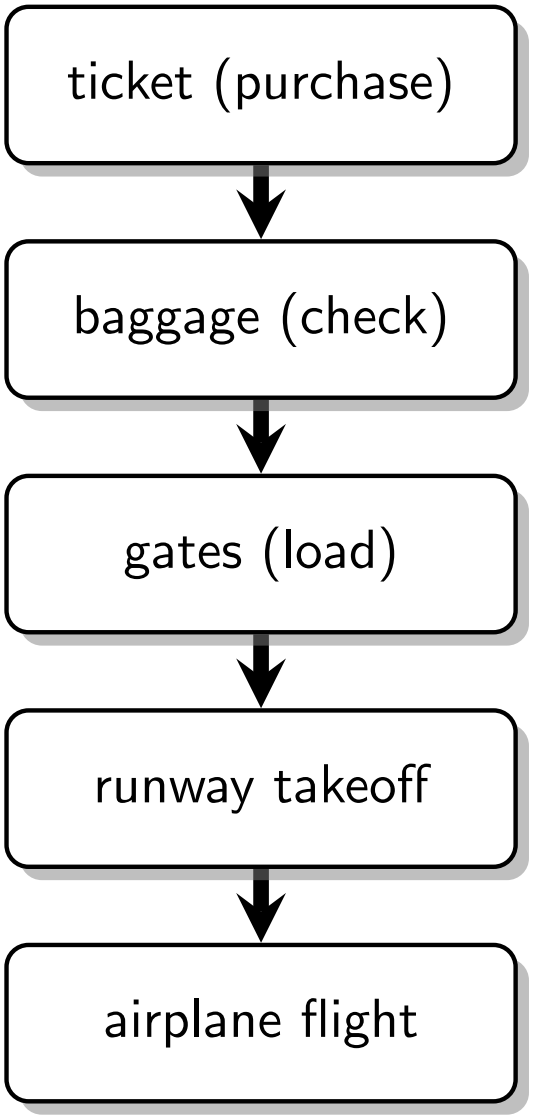
Analogy: organization of a travel

Different steps



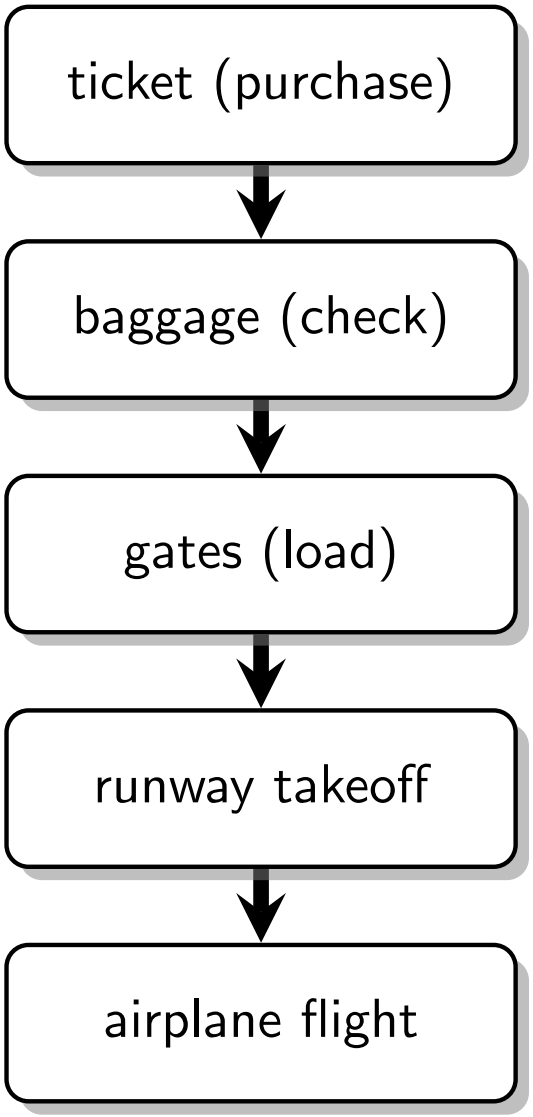
Analogy: organization of a travel

Let's arrange steps differently



Analogy: organization of a travel

Process on different entities but of same 'layer' can 'understand' each other



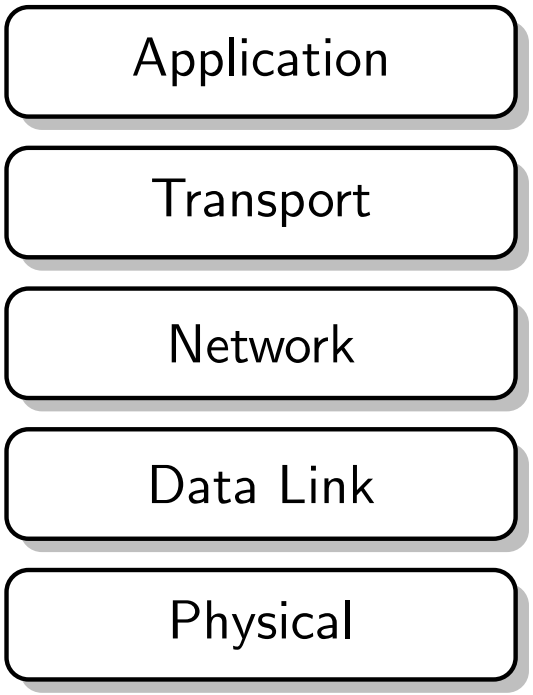
Analogy: organization of a travel

We have constructed a layered service architecture

Each layer provides a service

- via its own internal-layer actions and functions
- relying on services provided by layer below
- each layer ‘talks’ to its homologue

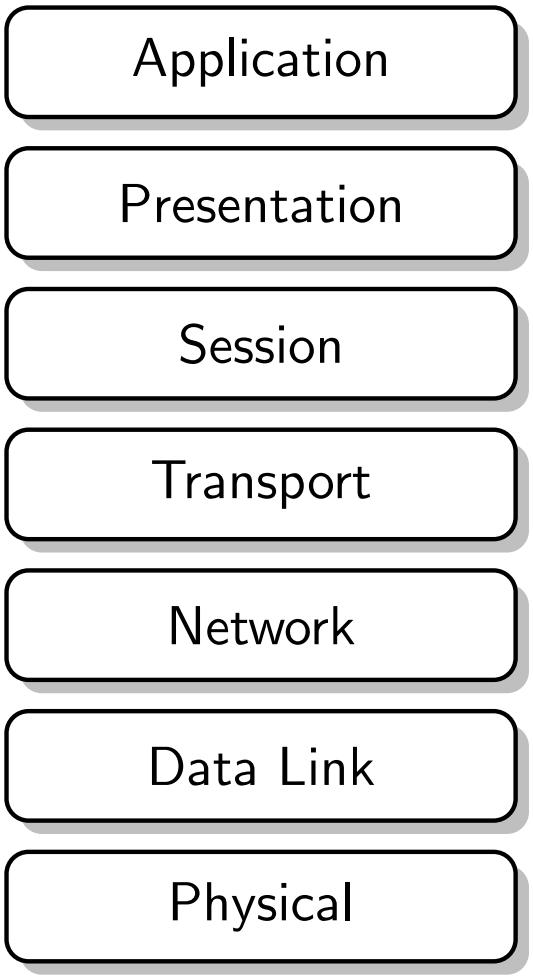
Layered reference model in the Internet



Layer 5 supporting network applications e.g. FTP, SMTP, HTTP

Layer 4 process-process data transfer e.g.TCP, UDP

ISO/OSI reference model



! Internet stack "missing" two layers! $\Rightarrow$ these services, if needed, must be implemented in application

Q Are they needed?

allow applications to interpret meaning of data, e.g.

Layers as a means to deal with complex systems

Some motivations for layering:

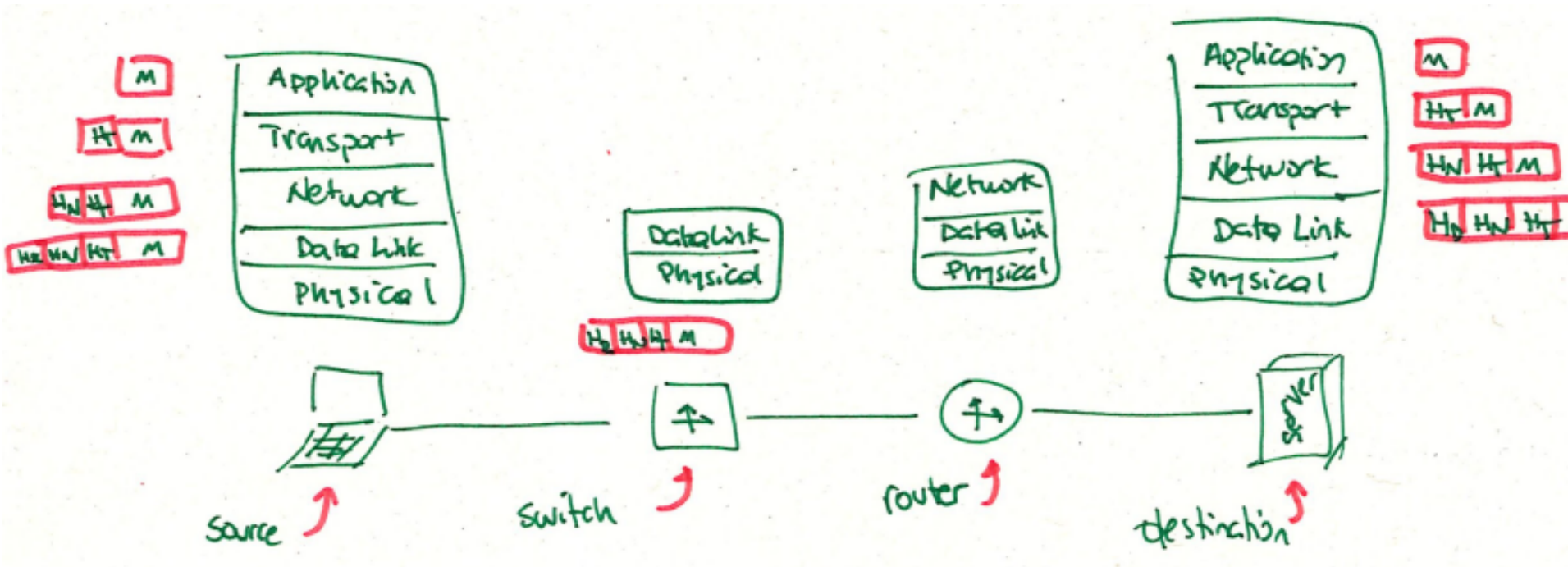
- ease implementation: focus on one function, rely on others for solving other 'problems'
- layered reference model for discussion
- modularization eases maintenance, updating of system
- change of implementation of layer's service transparent to rest of system e.g., change in gate procedure doesn't affect rest of system

Q Can you think of any cons of layering?

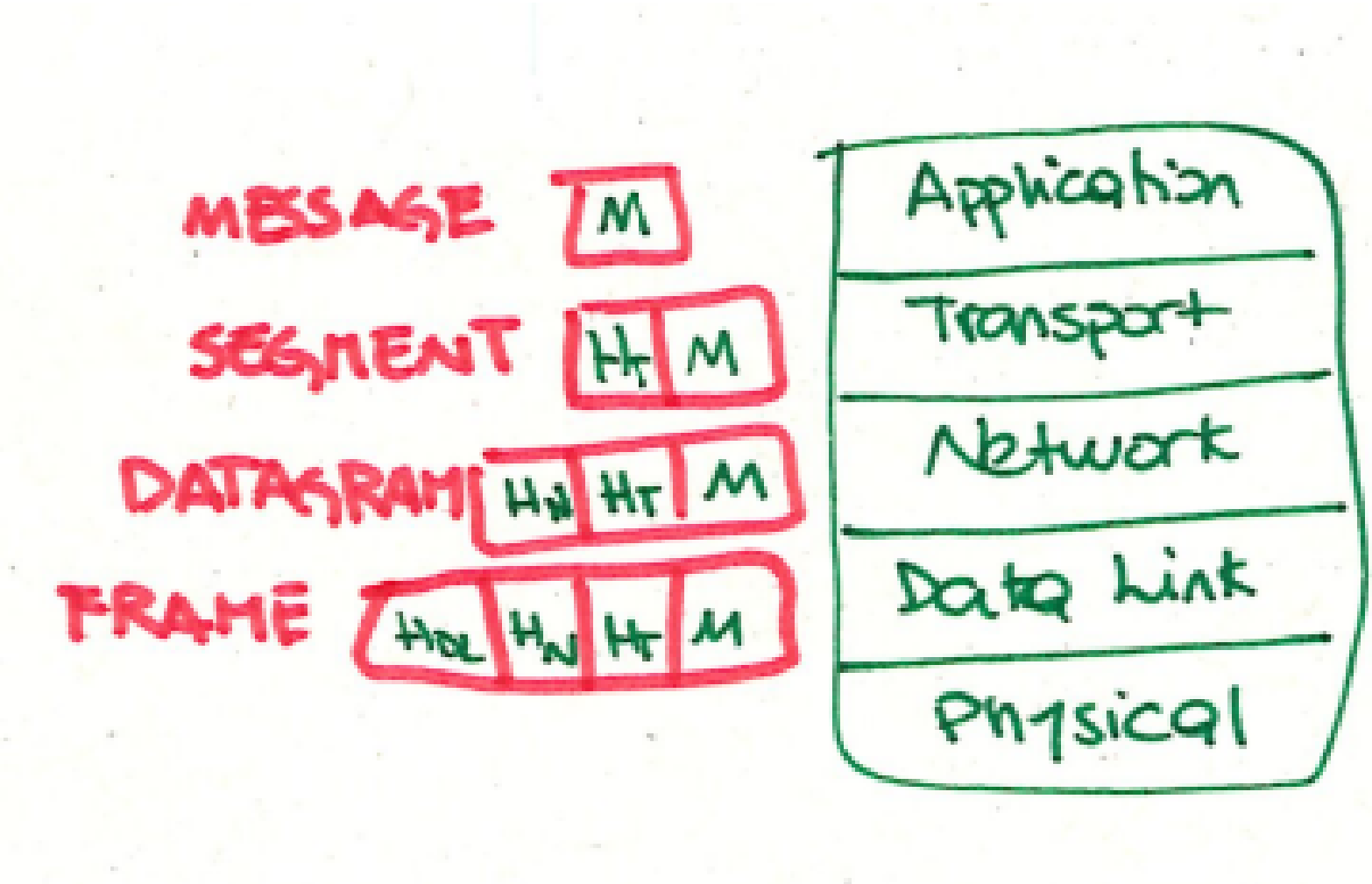
Encapsulation

How does each process "talk" to its homologue on another equipment?

Messages are *encapsulated* i.e. extra information (known as *headers*) are added by each layer in the source, and interpreted by corresponding layers in middle equipments (e.g. switches,routers) and by the destination.



Some vocabulary



Q Analogy between headers, encapsulation and our travel planning example?

Summary

Make sure you have some initial idea of the following concepts!
(we'll come back to them during next classes)

- Access network, Access technologies, Core network
- Packet
- Protocol
- Layers
- Delay

Acknowledgements

The contents of these slides are partially taken from Computer Networking a Top Down approach, J. KUROSE and K. ROSS and K. KUROSE's networking course
`http://www-net.cs.umass.edu/cs453_fall_2013/`