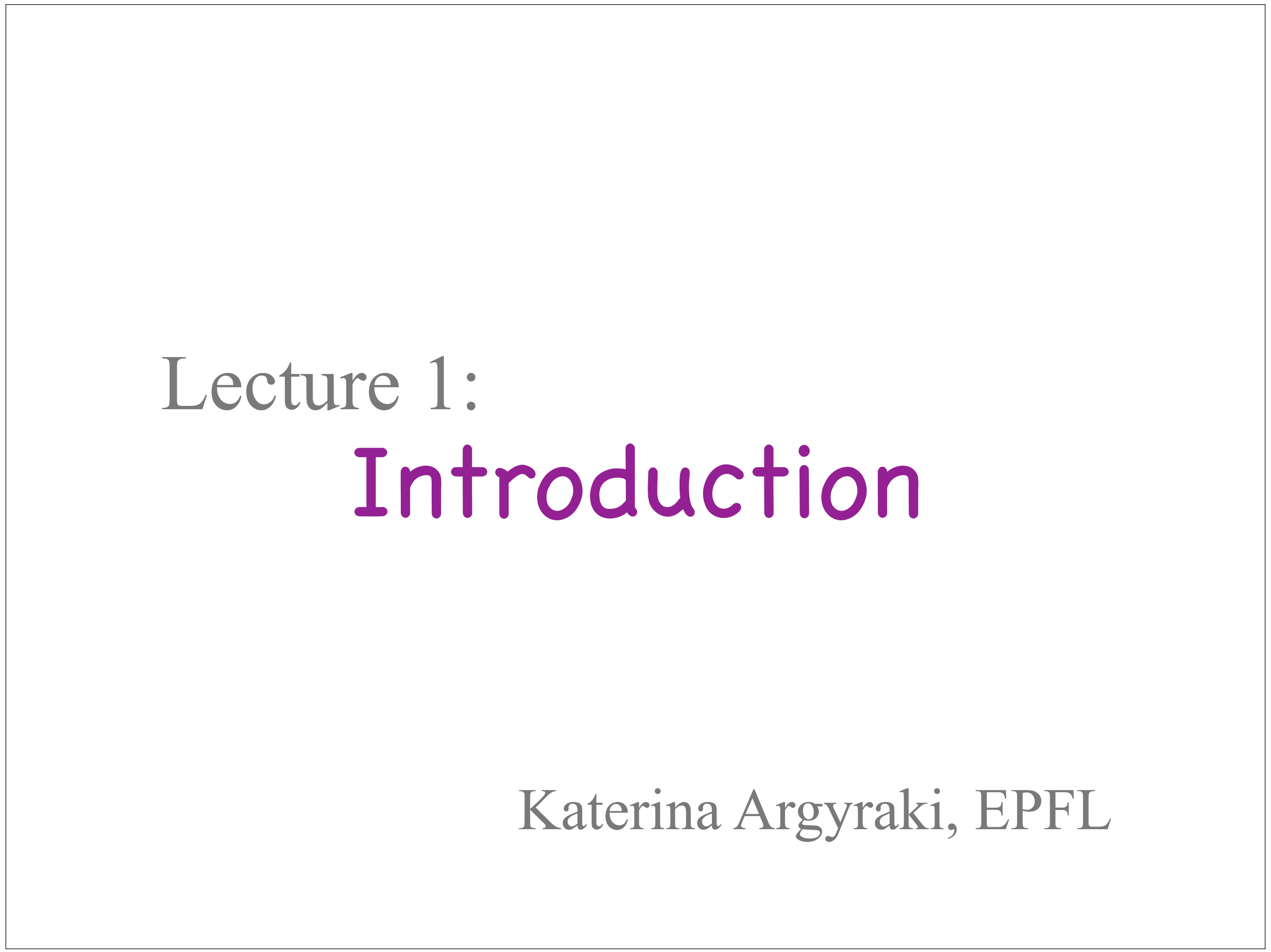


Lecture 1: Introduction

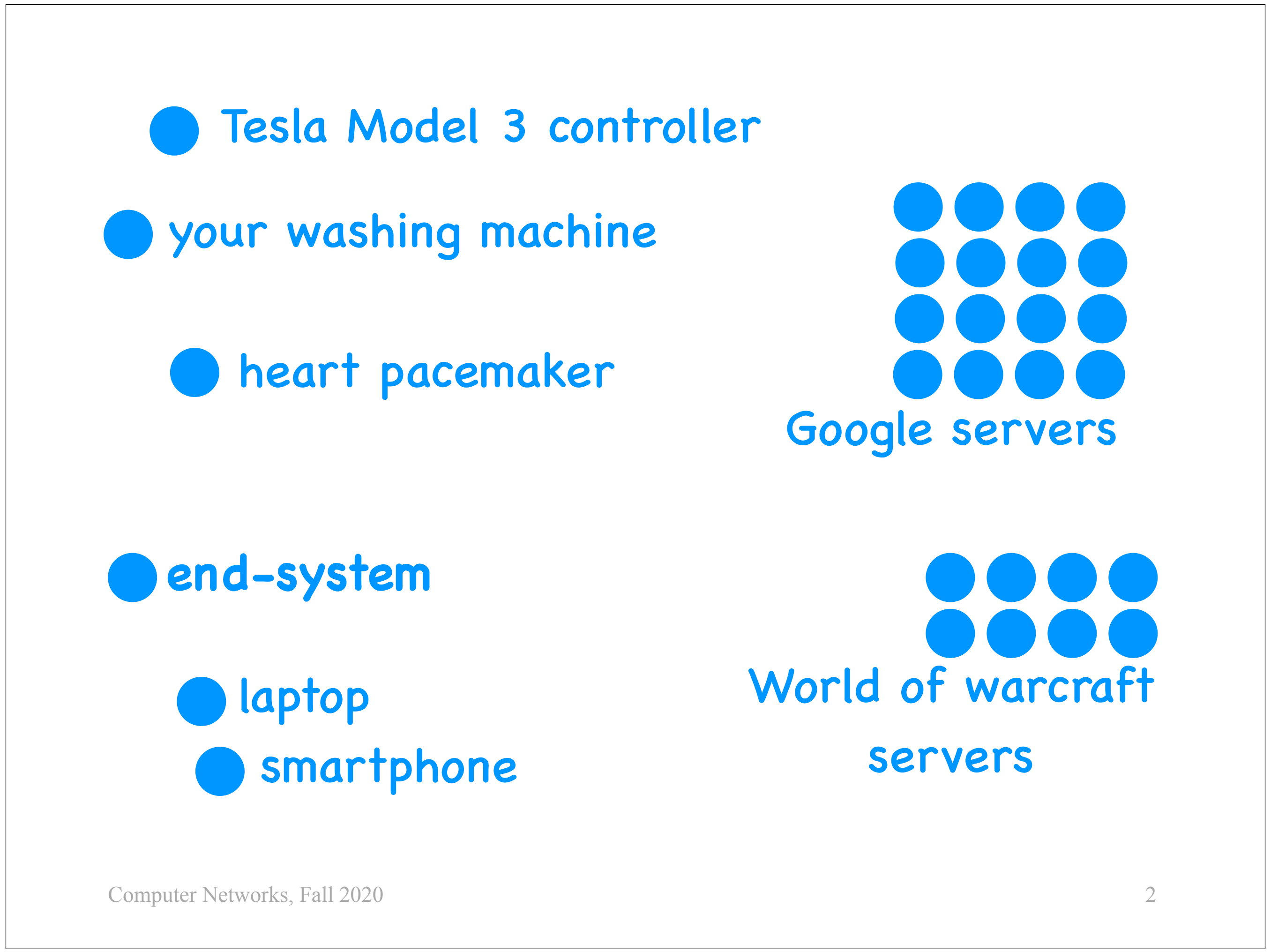
Katerina Argyraki, EPFL

Welcome to Computer Networks!

My name is Katerina, and I will be your main instructor.

This is a basic, introductory course. So, if you have never studied network protocols before, you are in the right place.

I will dive right into the topic of the first lecture, which is an introduction, and I will talk about course logistics at the end.



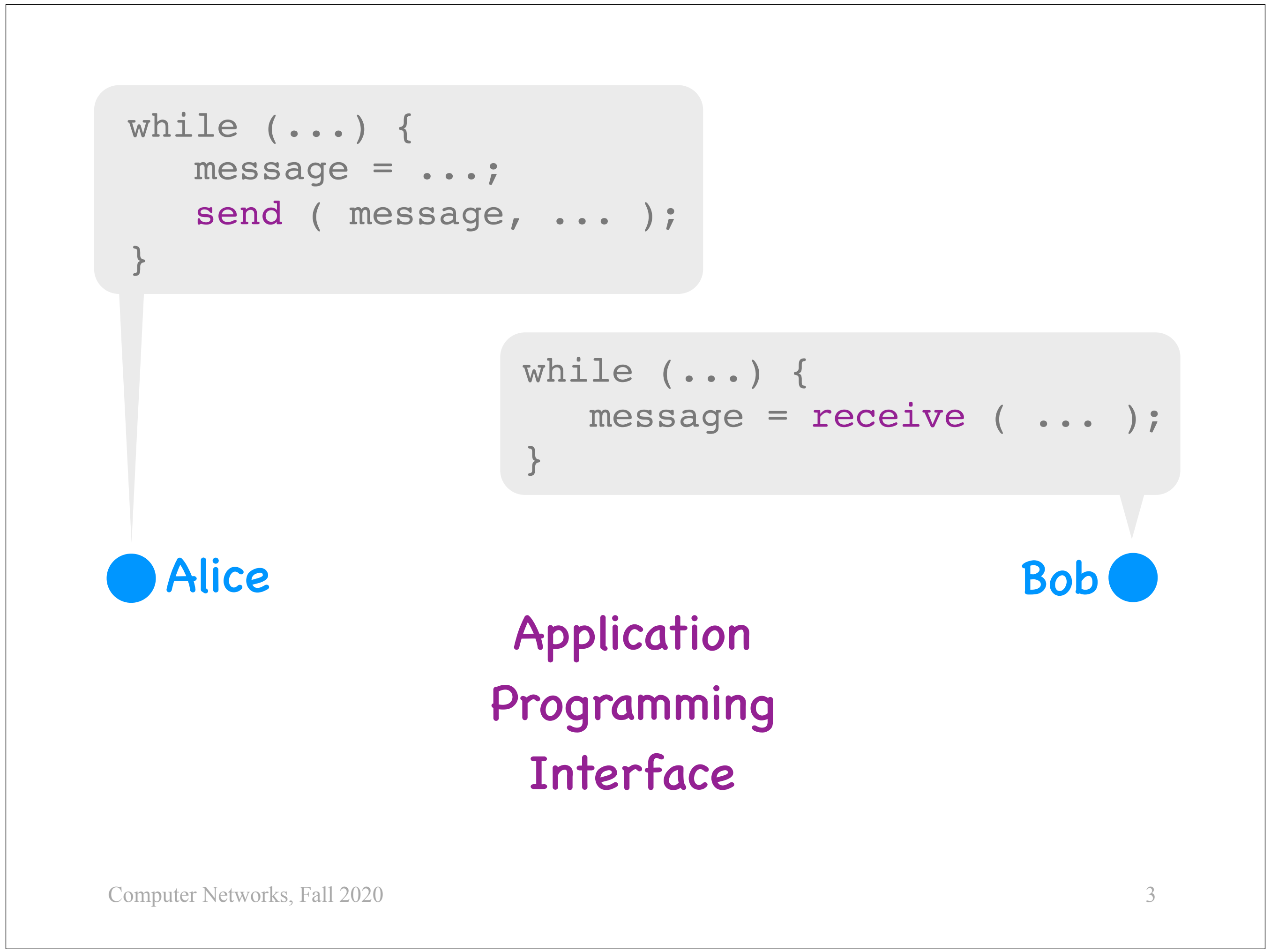
One component of the Internet that we will talk about a lot is the set of "end-systems" or "end-hosts".

These are all the devices that use the Internet for their communications:
- Your laptop and your smart-phone.
- All the servers that you access when you play a game, or when you use Google services.
- All the servers that your car accesses, e.g., to download navigation maps.
- There are web cams, washing machines, fire sensors that communicate over the Internet.
- There even exist heart pacemakers that use the Internet to communicate statistics about a patient's heart to doctors and nurses.

All these things are end-systems, and I will be representing them abstractly as blue circles.

The goal of the Internet is to enable all these end-systems to communicate with each other.
And they do that by running what we call "distributed applications".
You know what an application is: code running on a computer.
A *distributed application* is multiple pieces of code running on multiple computers, multiple end-systems.

So, what does a distributed application look like?
Let's look at a



```
while (...) {
   message = ...;
   send ( message, ... );
}
```

```
while (...) {
   message = receive ( ... );
}
```

simplified example.

We have two end-systems, called Alice and Bob,
and Alice wants to periodically send messages to Bob.

To do this, Alice runs a piece of code that keeps constructing and sending messages,
while Bob runs a piece of code that keeps receiving messages.
These two pieces of code together form a distributed application.

You already know how to develop a local application, which runs on a single computer without interacting with the rest of the world.

Developing a distributed application is much harder, because you need to account for all the possible combinations of Alice's and Bob's behavior.
For instance, what should Bob do if Alice stops sending messages, or if she sends messages too fast, or if she sends the wrong message?
This is the bad news.

The good news is that when you sit down to develop a distributed application, you will not write from scratch code that sends and receives messages.
You will use existing functions, like "send" and "receive", which are provided to you by your operating system.
These functions form an "Application Programming Interface" (API) for using the Internet.

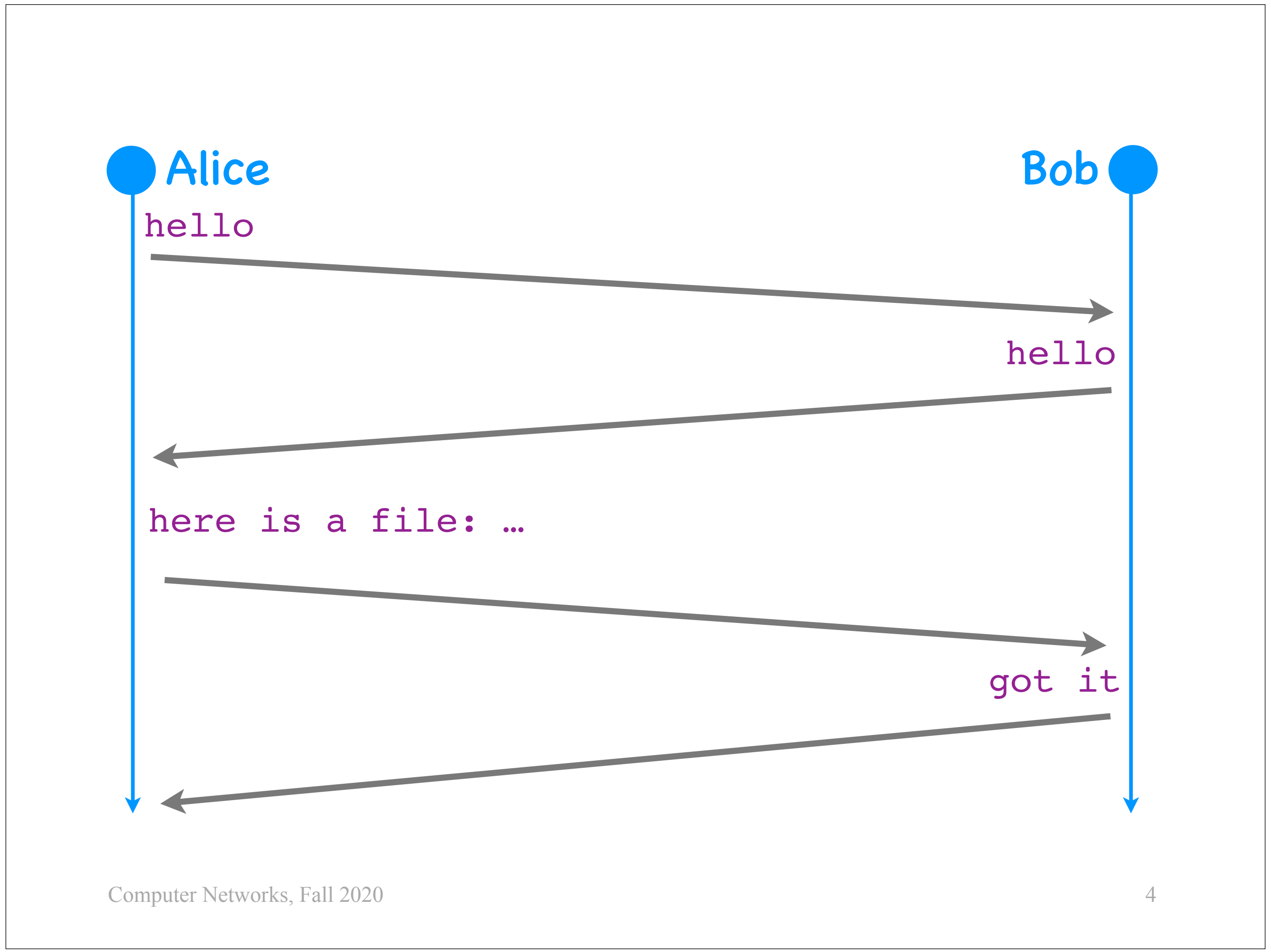
What does this mean?

When you want to send a letter through the postal system, you have to follow certain rules.
- You need to put your letter in an envelope and write a correct address on a particular part of the envelope.

- You need to drop your letter in a mailbox.
These rules are the “interface” between you and the postal system -- your only way of using the postal system successfully.

Similarly, when a distributed application wants to send a message through the Internet, it has to use certain functions in a certain way.
So, these functions are the “interface” between the application and the Internet, this is why they are called an “Application Programming Interface”.

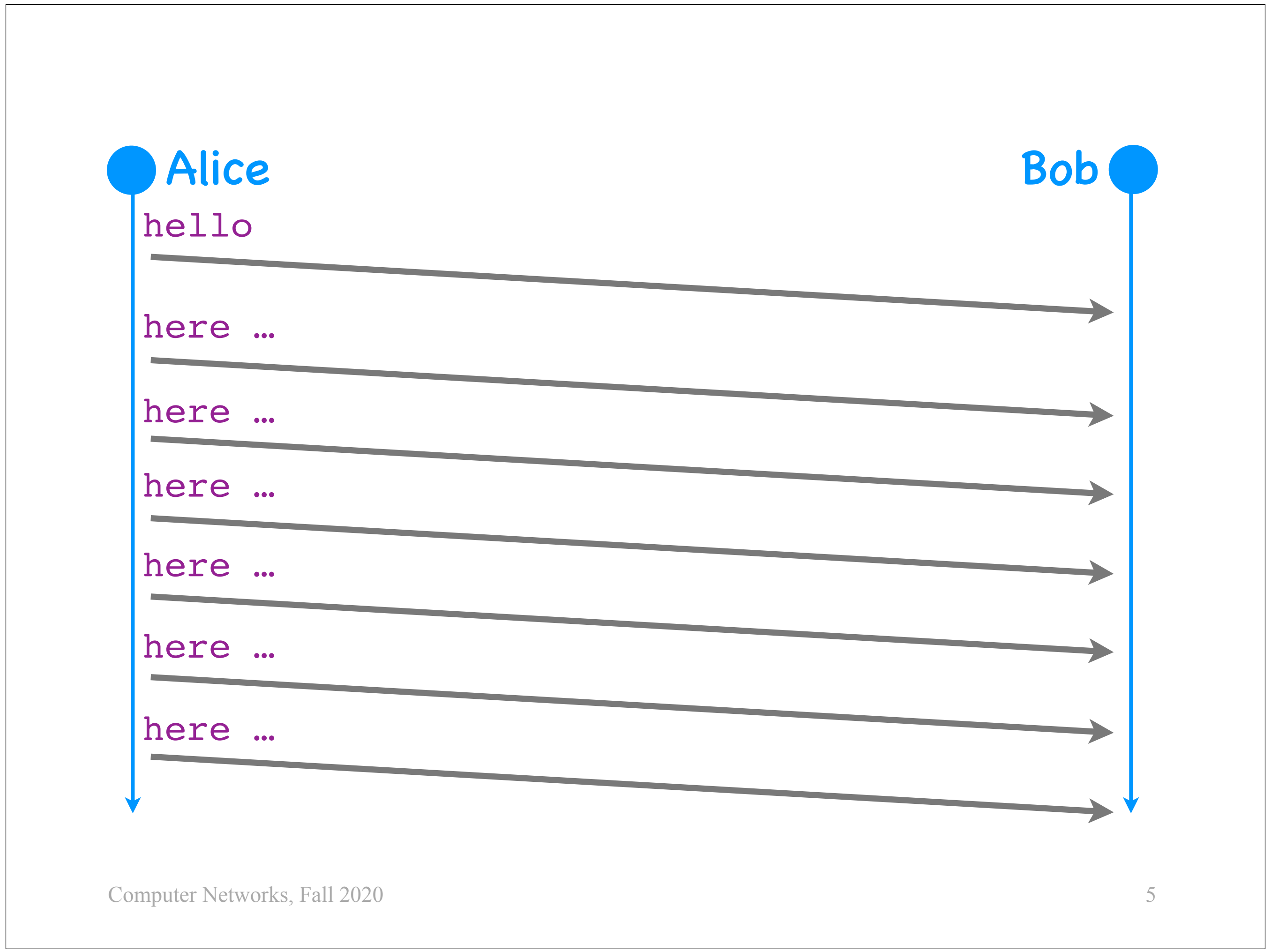
Developing distributed applications that use this API correctly is one of our goals in this course.



When you develop a distributed application, before you start writing code, you will need to design a “communication protocol”.
That is exactly what it sounds like: a specification of all the possible sequences of messages that the communicating entities can exchange.

Here is a simple example of a protocol between two end-systems:
- The communication must start with one party saying hello.
- Then, the other party must respond.
- After that, the first party must provide a file.
- Then the other party must respond by saying that it received the file.

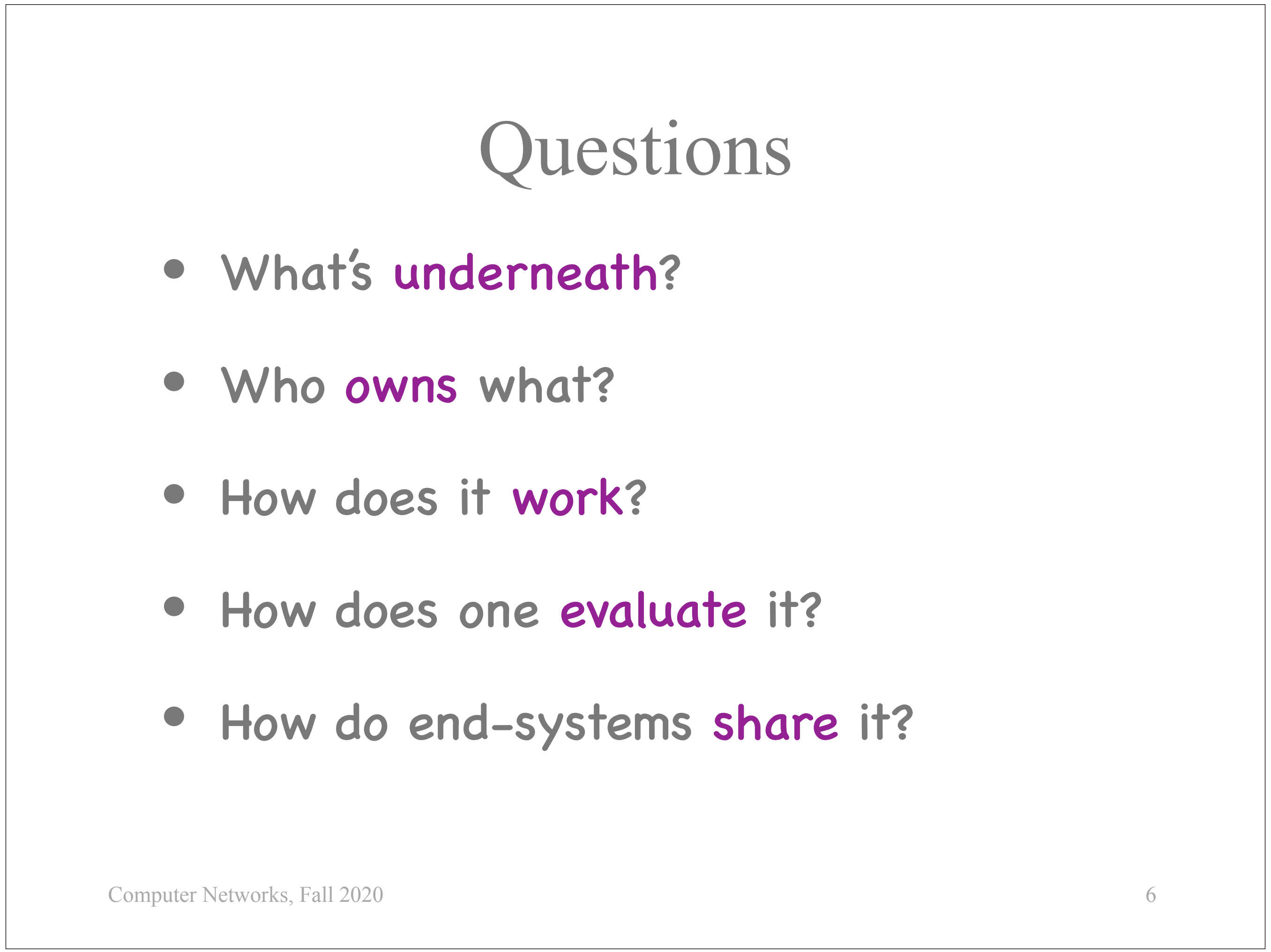
If any one of the parties does something outside this specification, then it violates the protocol.
For example,



Suppose Alice says “hello,” but then, instead of waiting for Bob’s response, she starts sending one file after the other.
Then she violates the protocol, and Bob will stop talking to her.

Computer communication protocols are very much like human communication protocols.
If I say “good morning, how are you”, I should wait to hear your response.
If I start talking immediately, you will conclude that I am impolite and stop talking to me.

Understanding the design of Internet protocols -- why they were designed this way, what is good and what is bad about them -- is our main goal in this course.

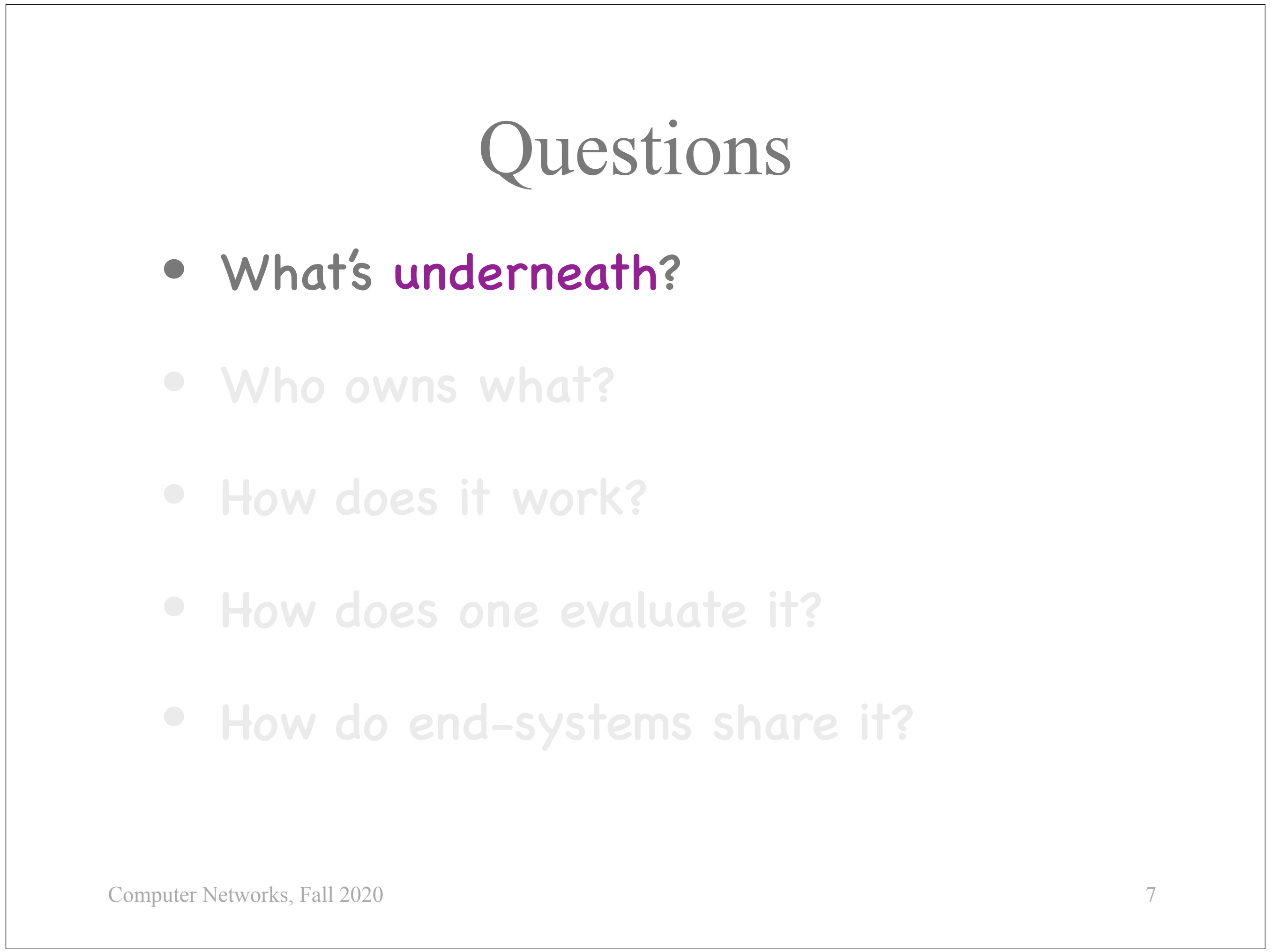


Questions

- What's **underneath**?
- Who **owns** what?
- How does it **work**?
- How does one **evaluate** it?
- How do end-systems **share** it?

Here is a set of questions that we will explore about the Internet:

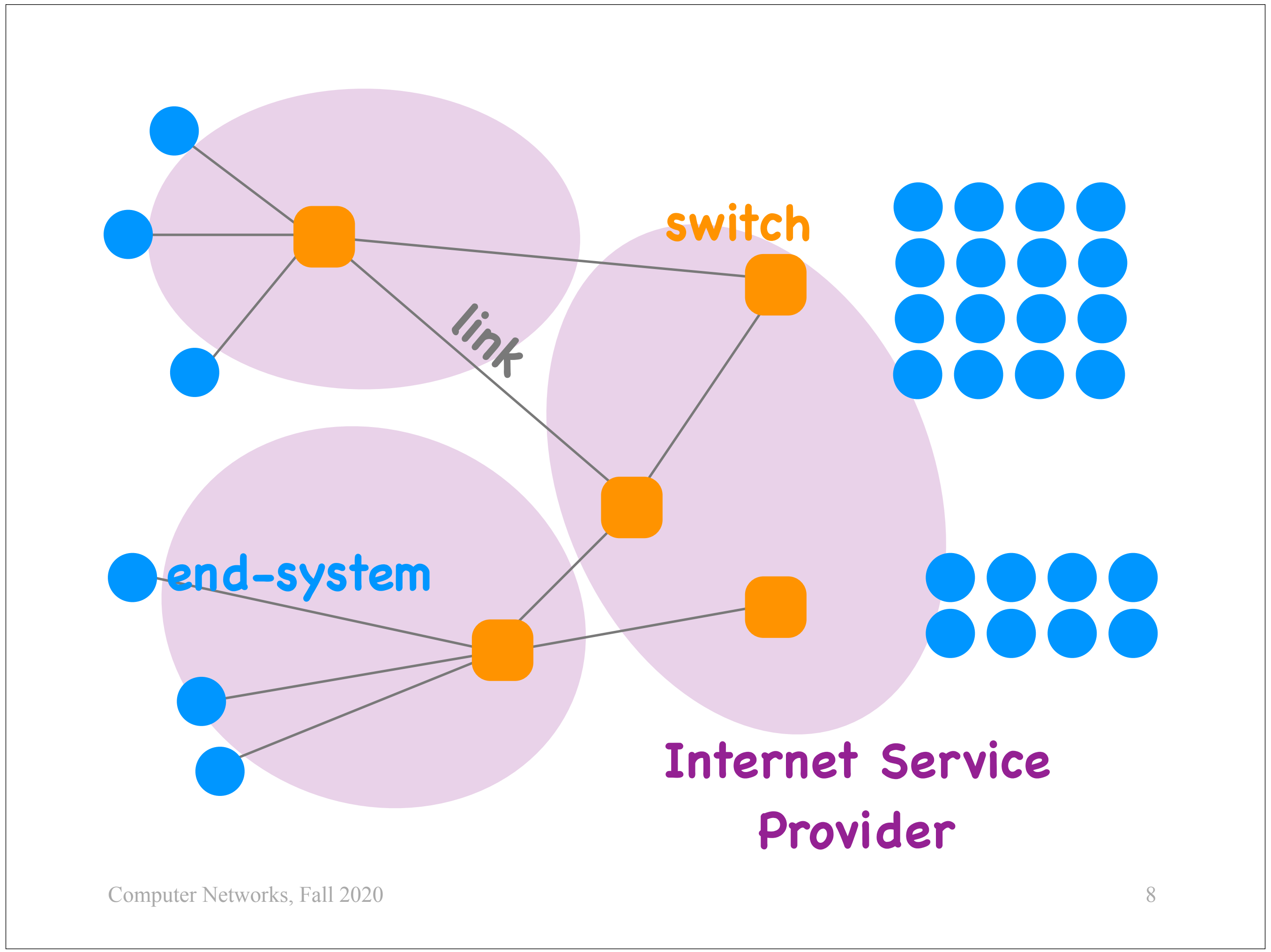
- What's underneath? What infrastructure is used when we communicate over the Internet?
- Who owns and manages all that infrastructure?
- How does it work? What happens exactly when distributed applications exchange messages over the Internet?
- How do we evaluate how well an Internet connection works?
- How do billions of end-systems successfully share the Internet infrastructure?



Questions

- What's underneath?
- Who owns what?
- How does it work?
- How does one evaluate it?
- How do end-systems share it?

Let’s start with the first question: what infrastructure is used when we communicate over the Internet?



First, there are **packet switches*, which are devices that enable the end-systems to communicate with each other.
Your DSL modem or cable modem, or the wireless router that you have at home are examples of packet switches.

Second, there are **links**, which connect the packet switches to the end-systems and among themselves.

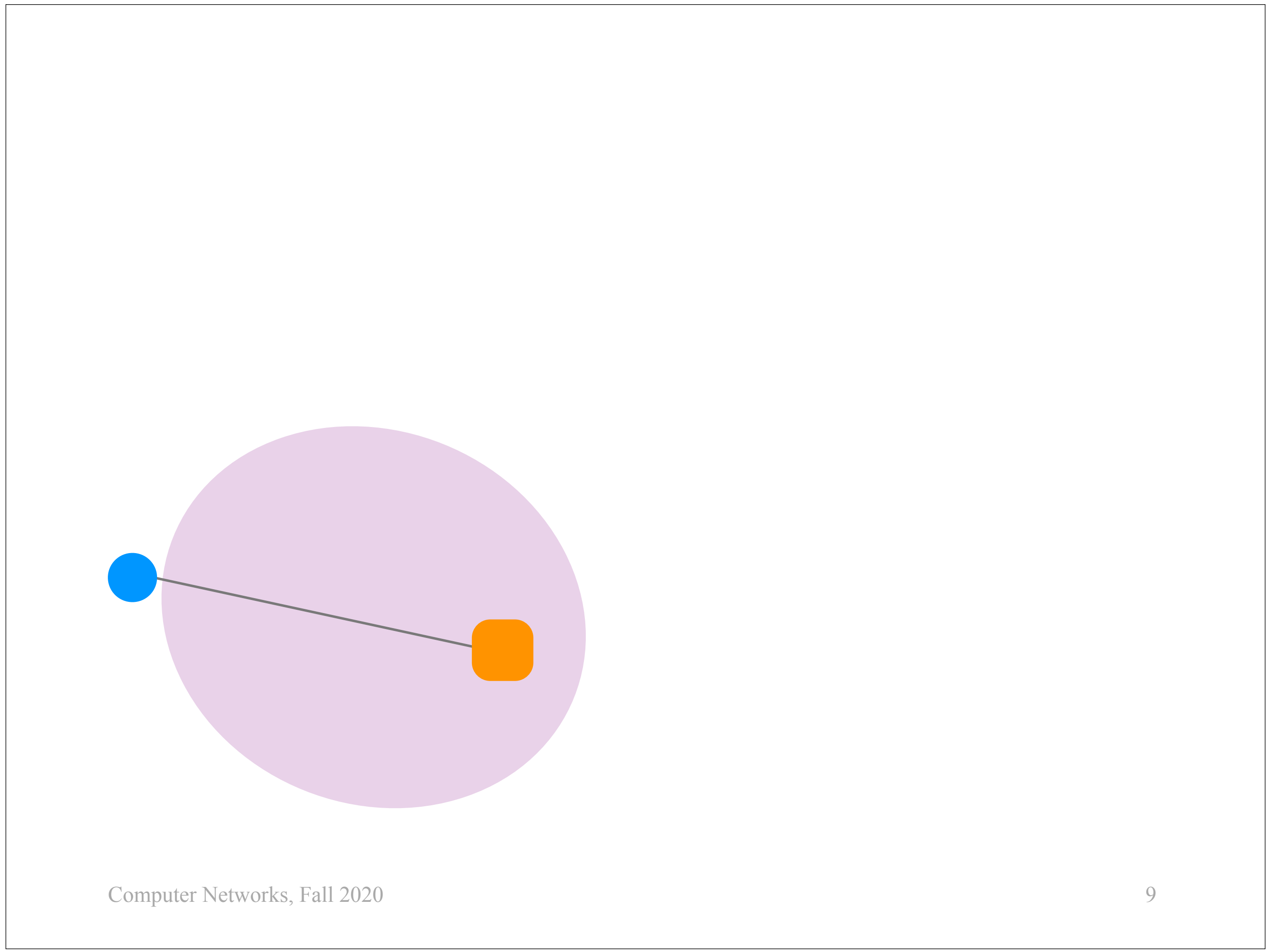
Who owns and manages all these packet switches and links?
Special companies that are called **Internet Service Providers** or **ISPs**.

All the ISPs in the world are interconnected in a sort of hierarchy -- lots of small ISPs connect to bigger ISPs, which connect to even bigger ISPs. We will examine that hierarchy later today.

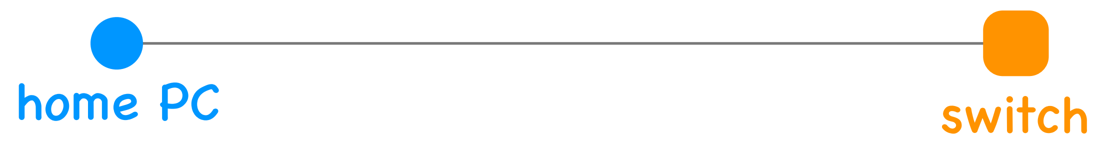
You should remember that each ISP is its own separate administrative and economic entity.
This administrative and economic separation may not be a technical matter, but it has technical repercussions.
For instance, when an ISP faces technical problems (e.g., a network link has failed), it may not have an incentive to reveal that to another ISP.
As a result, ISPs rarely collaborate to diagnose network problems.

For example, when packets from this source to this destination are lost, the two involved ISPs do not collaborate to diagnose the problem.
This partly explains why sometimes you have bad Internet connectivity, you report it to your ISP, yet the problem persists for many hours.

Now, let's see what these links are made of.
I am going to keep only

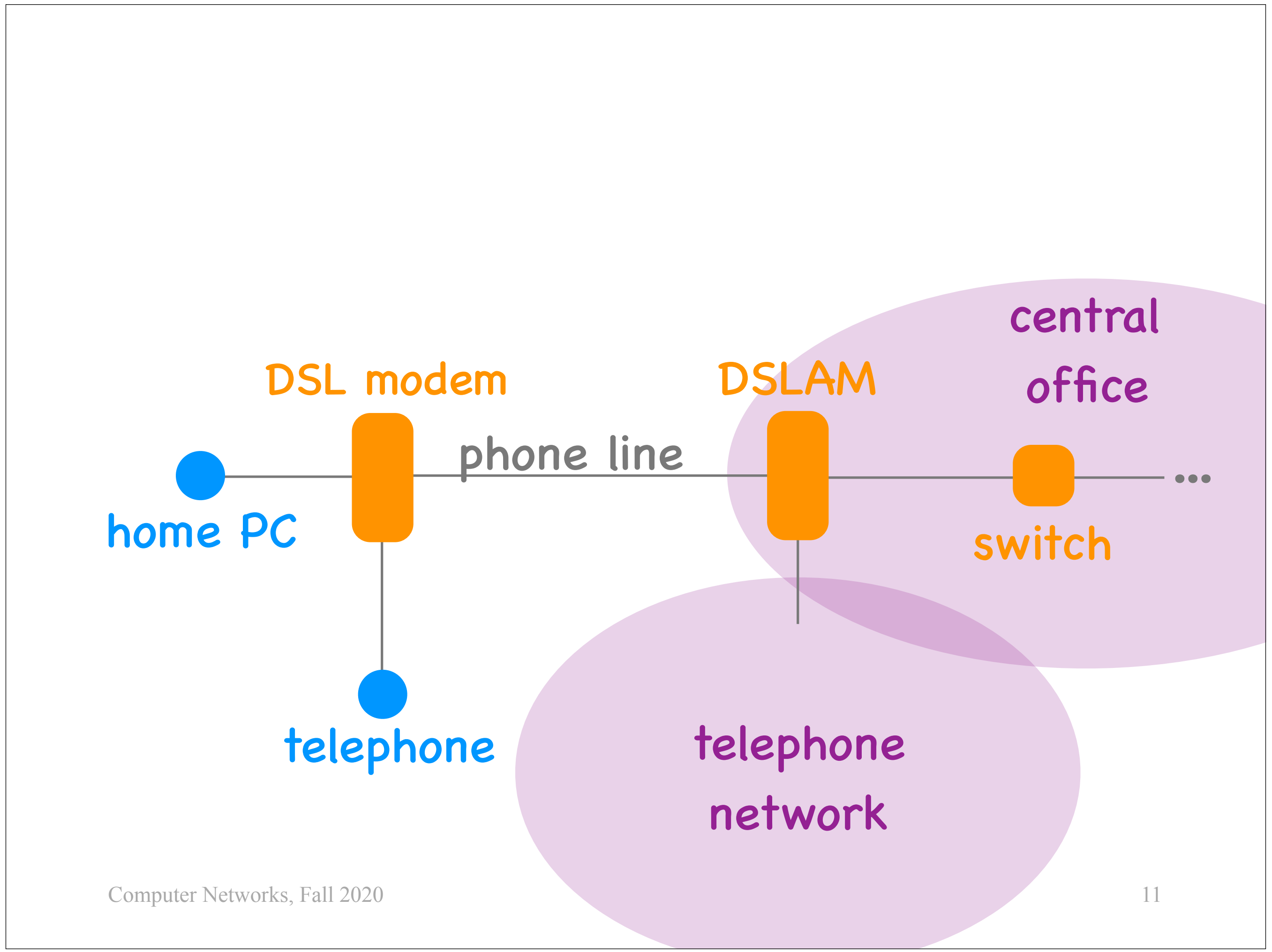


one link between an end-system and a switch,
and I am going to



zoom in on this link.

Suppose this end-system is a PC you have at home.
How is it connected to the Internet?



One option is that the home PC is connected to a DSL modem,
which is connected to the household phone line,
which goes to a place called a “central office”, which is located within a few km from the household.

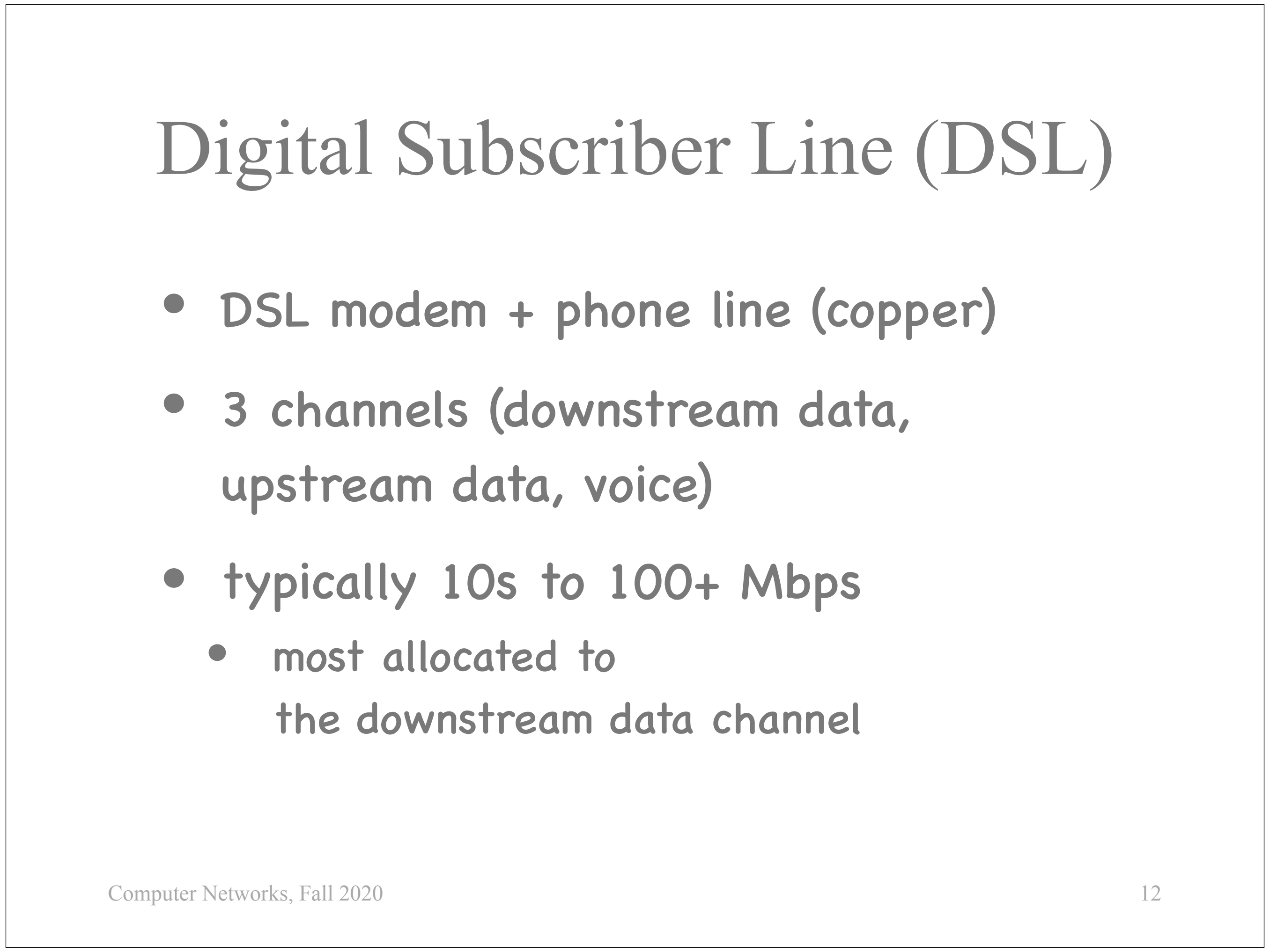
Inside the central office, the phone line is terminated at a device called a DSLAM,
which is connected to a switch,
which is connected to the rest of the Internet.

DSLAM stands for DSL Access Multiplexer.

If the household has a traditional telephone device, that is also connected to the DSL modem, which multiplexes the data from the PC and the voice signal from the telephone onto the phone line.

The DSLAM then separates the two and sends the data from the PC toward the Internet and the voice signal from the telephone toward the traditional telephone network.

So, this is what happens when you get a DSL Internet connection.
You buy a DSL modem and you pay your phone company (which now also becomes your Internet service provider) to carry data from your DSL modem to the Internet and vice versa.



Digital Subscriber Line (DSL)

- DSL modem + phone line (copper)
- 3 channels (downstream data, upstream data, voice)
- typically 10s to 100+ Mbps
  - most allocated to the downstream data channel

DSL stands for Digital Subscriber Line.

It is a technology used by phone companies to connect households to the Internet.

It relies on a DSL modem and a phone line, which is made of copper.

In particular, the DSL modem creates three separate channels on the phone line:
- one for downstream data (from the Internet to the home PC),
- one for upstream data (from the home PC to the Internet), and
- one for the voice signal from and to the telephone.

The supported data rate varies typically from a few tens to about 100 Mbps,
and most of it is typically allocated to the downstream data channel.

Why phone lines?

Why use phone lines to connect households to the Internet?

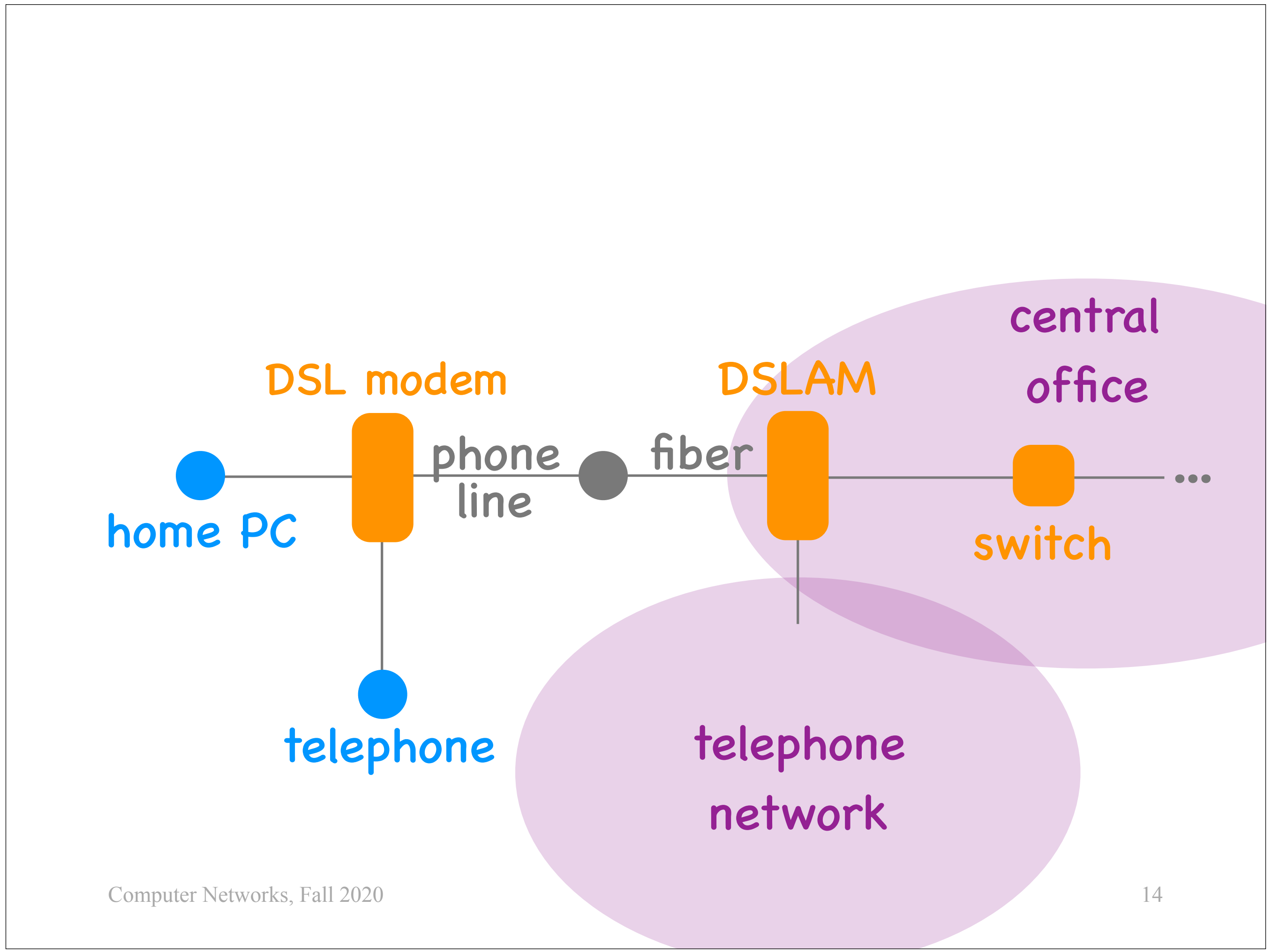
In most developed countries, phone lines are already decades old, so, clearly they are not the state of the art as far as communication media go.

So, why do we use them to connect to the Internet?

We use phone lines, because they are already there.

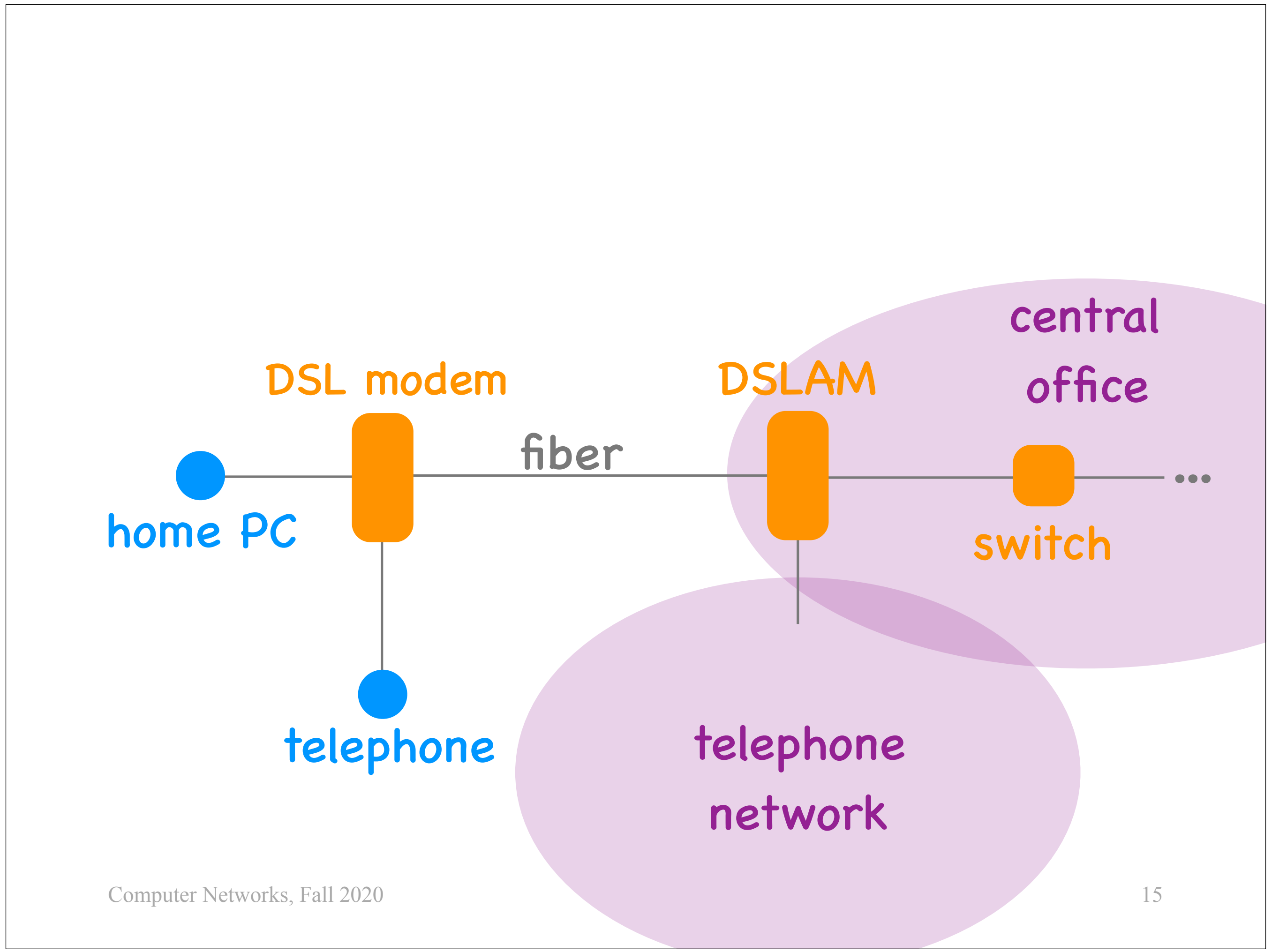
When people started to be interested in Internet connectivity, phone companies (and in particular national phone companies) had a big advantage: they already owned a physical connection to every household in the country. So, providing Internet connectivity over these existing physical connections was an obvious business opportunity.

Laying down new and better communication media, like fiber, is expensive, and it does not always make sense from an economic point of view.



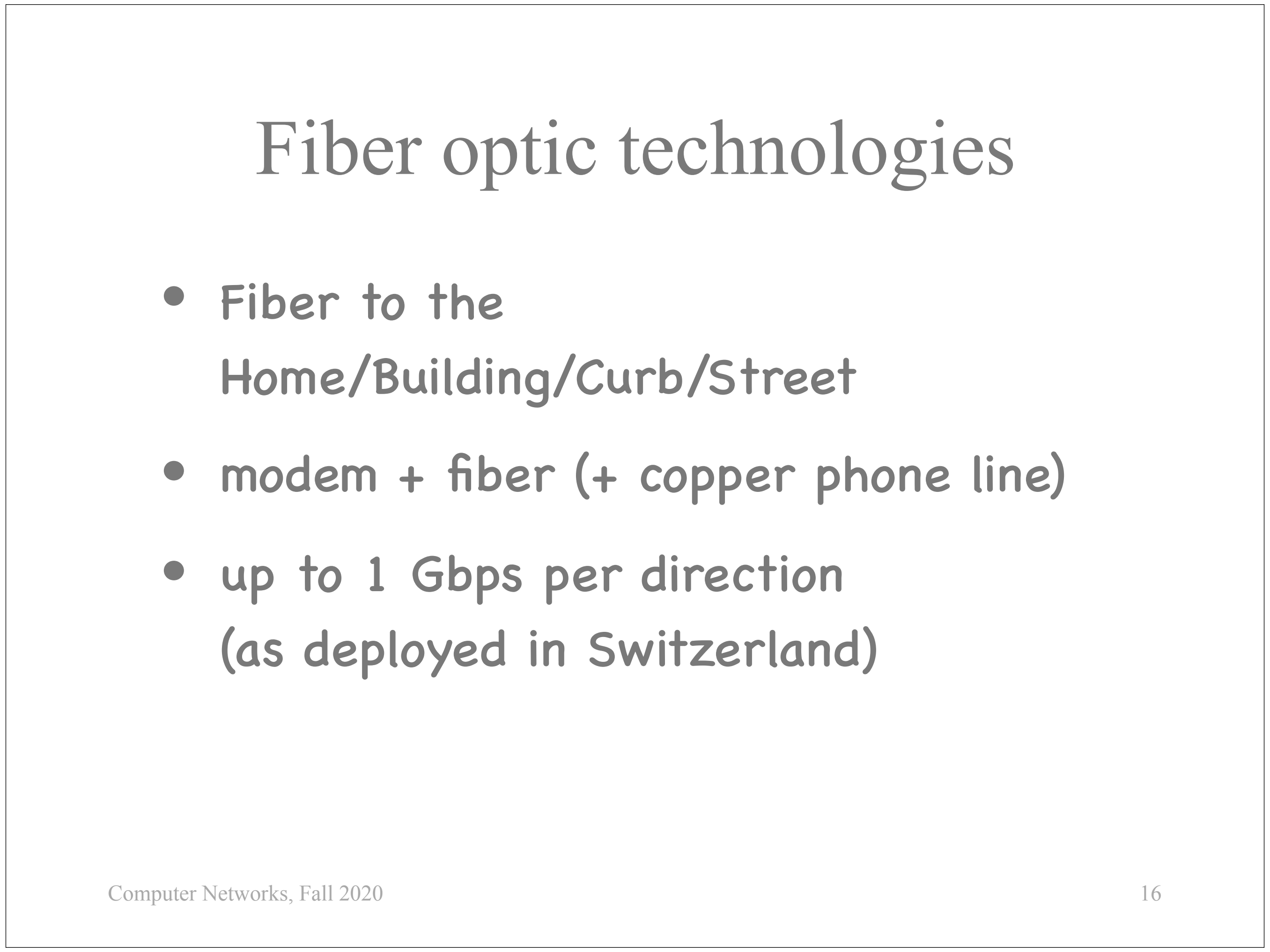
That said, in Switzerland, traditional copper phone lines are actually being replaced by fiber.

Depending on the area, the fiber may reach the street, the curb, the building, or…



go all the way to the home (in which case, it replaces the traditional copper phone line completely).

So, DSL is being replaced by

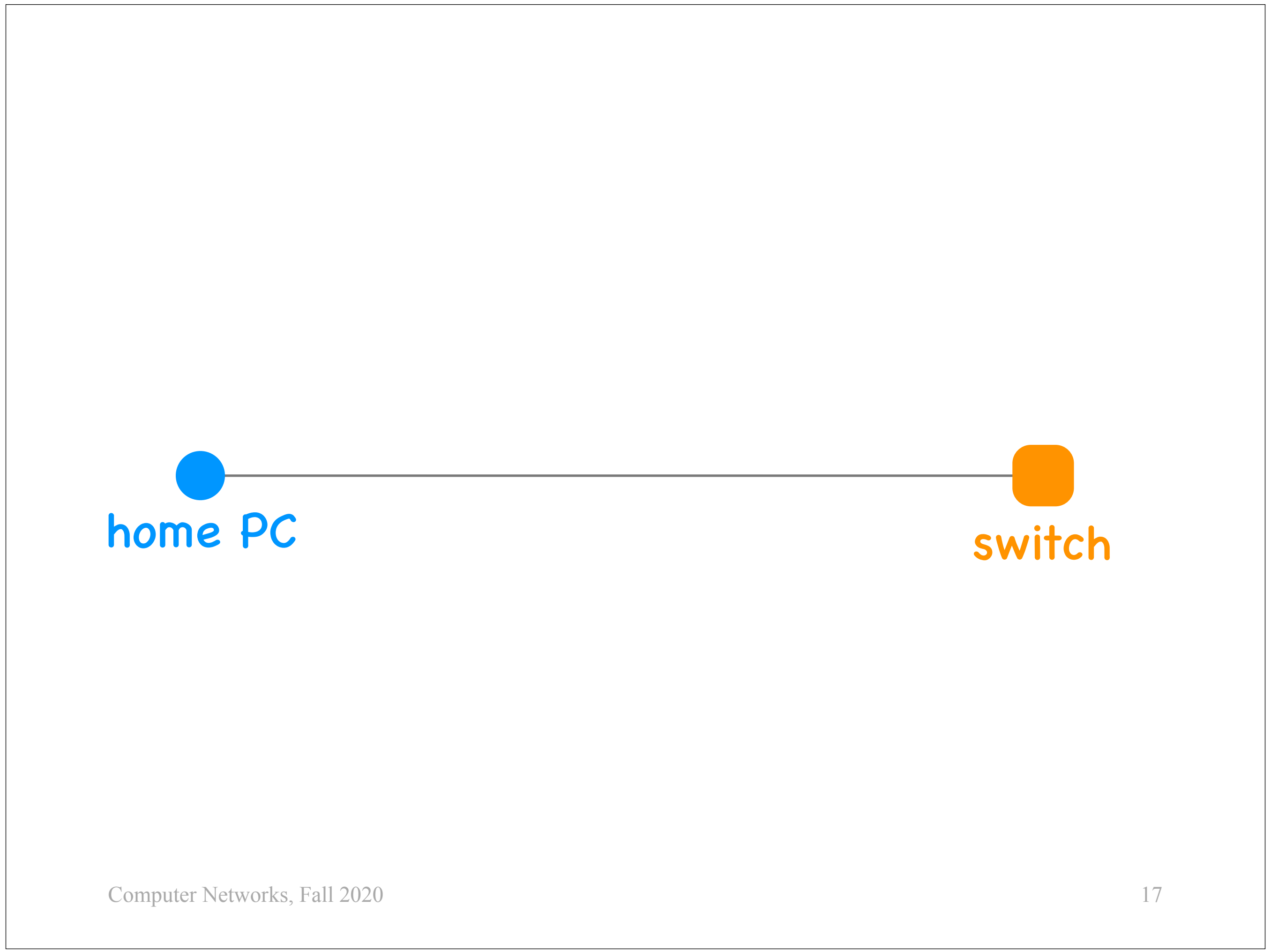

Fiber optic technologies

- Fiber to the Home/Building/Curb/Street
- modem + fiber (+ copper phone line)
- up to 1 Gbps per direction (as deployed in Switzerland)

Fiber optic technologies,
which are conceptually similar,
but replace part or all of the traditional copper phone line with fiber,
hence achieve higher data rates.

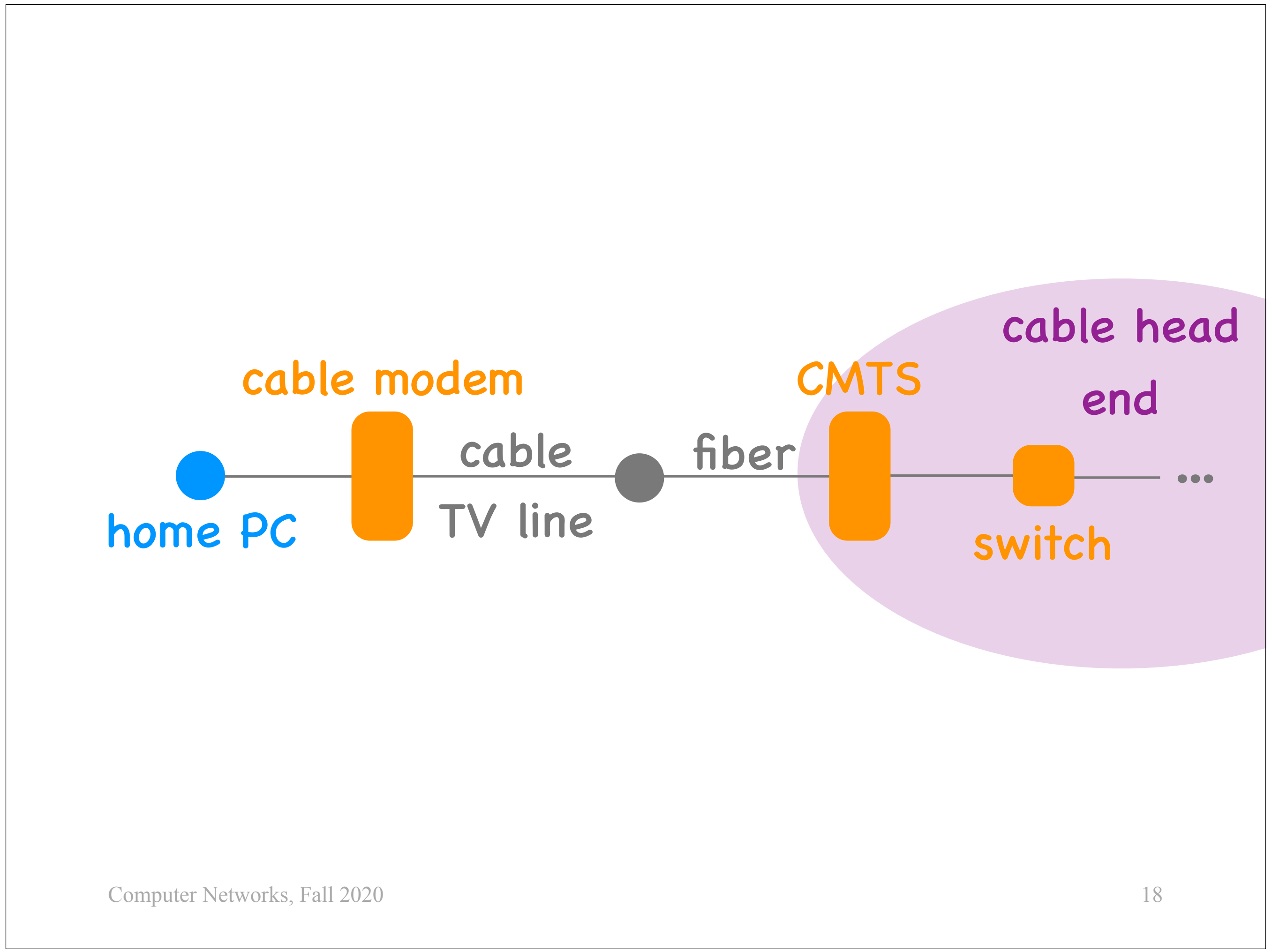
—

What other companies own physical connections to many households?



Let’s consider again a home PC and the first switch that connects it to the Internet.

We said that this link could be provided by a phone company.
Another option is that it’s provided by a cable company.



In this case, the home PC is connected to a cable modem,
which is connected to the household cable TV line,
which goes to a “fiber node” and then to a place called “cable head end”, which is located within a few km from the household.

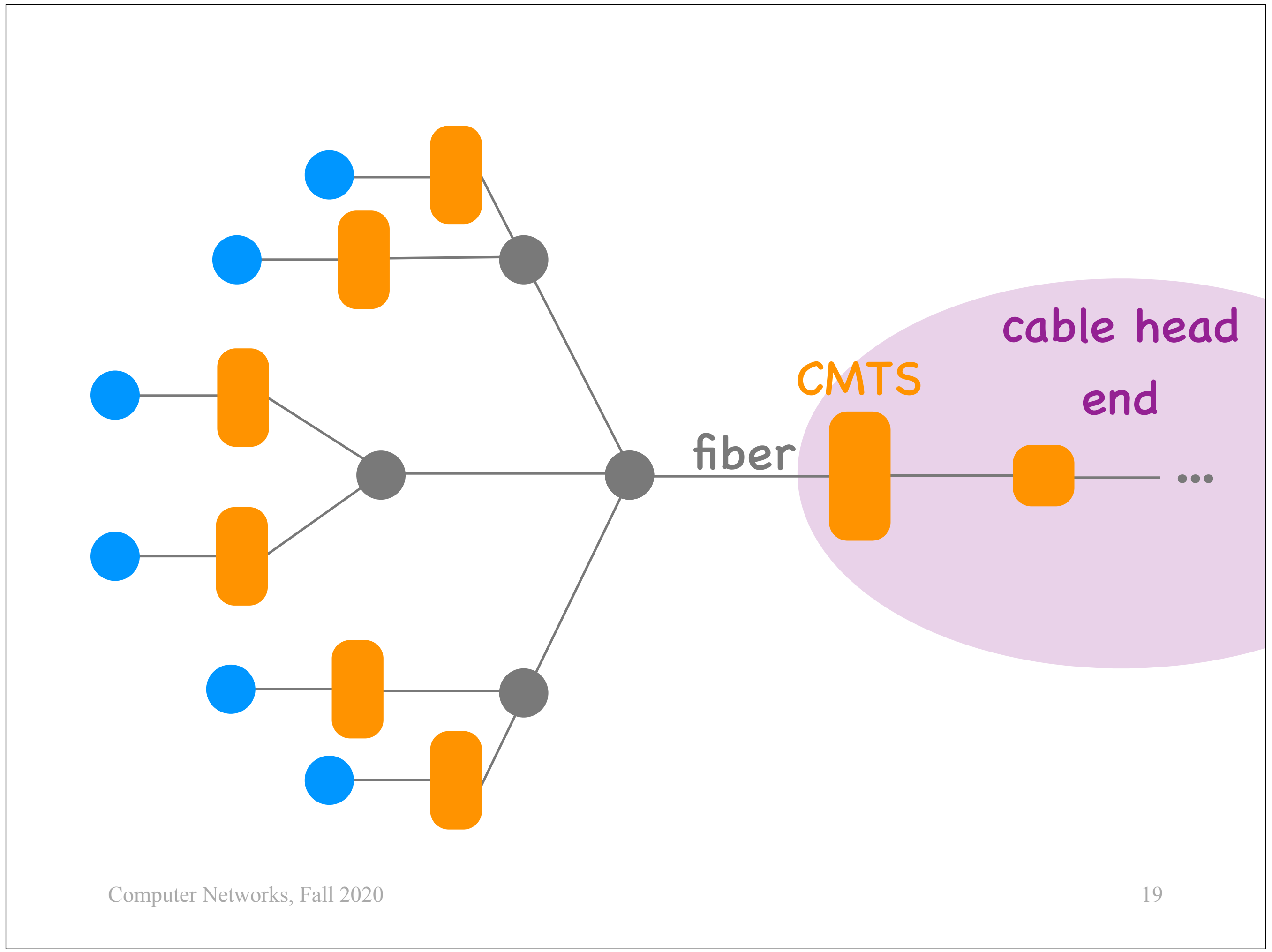
Inside the cable head end, the fiber is terminated at a device called a CMTS,
which is connected to a packet switch,
which is connected to the rest of the Internet.

CMTS stands for Cable Modem Termination System.

So, this is what happens when you get a cable Internet connection.
You buy a cable modem and you pay your cable company (which now also becomes your Internet service provider) to carry data from your cable modem to the Internet and vice versa.

I hope this does remind you of something. Conceptually, it is very similar to DSL. Instead of a DSL modem we have a cable modem. Instead of a phone line we have a cable TV line. And so on.

But there **is** one difference between DSL and cable.



Unlike DSL is that cable is a “shared medium” and a “broadcast medium”.

What do these terms mean?

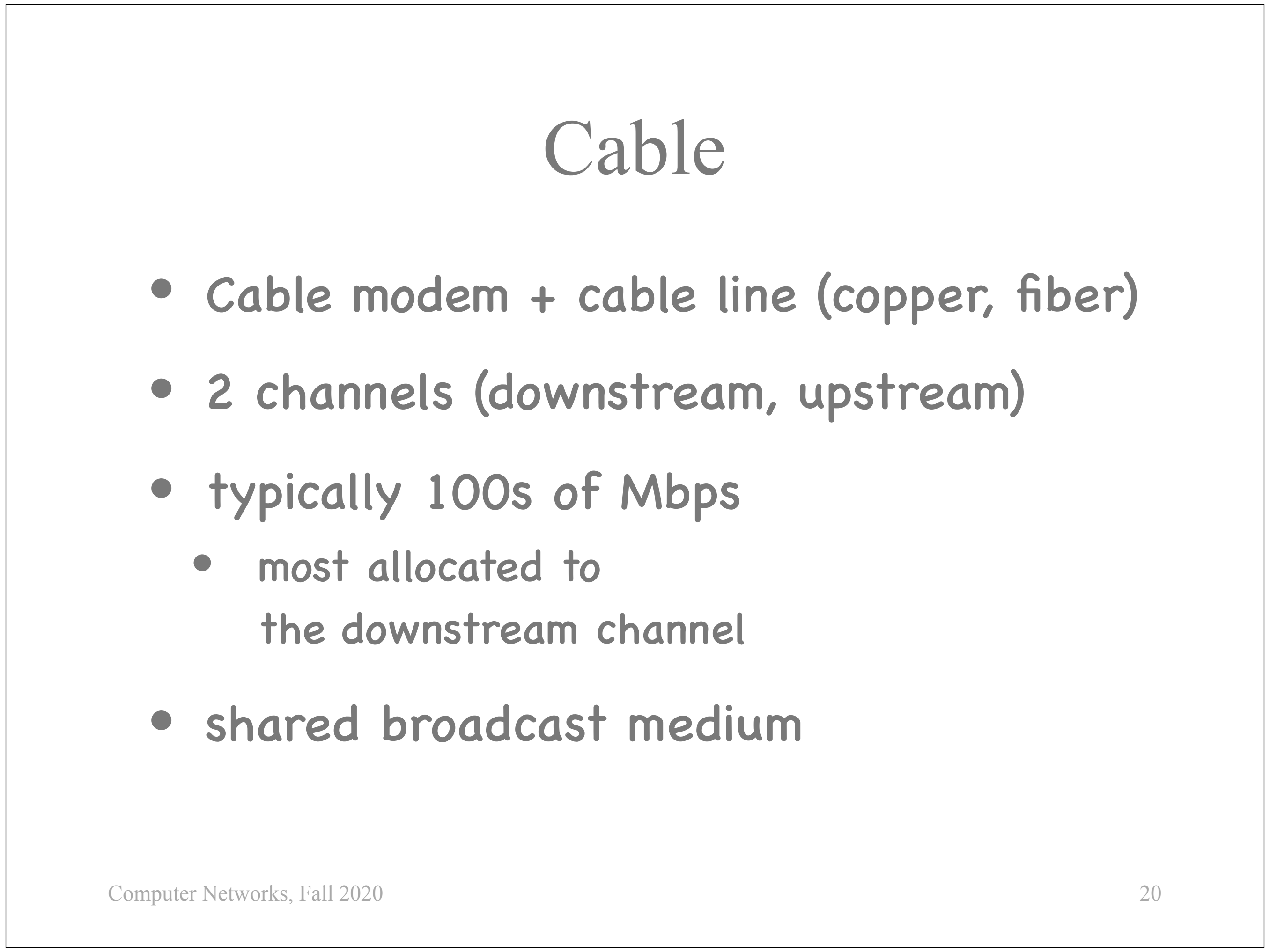
Cable is a “shared medium” in the sense that:
- Many end-systems may share the same link.
- All the end-systems that are connected to the same tree have to share the last hop to the cable head end.

In contrast, with DSL, each household has its own separate phone line to the central office.

Moreover, cable is a “broadcast medium” in the sense that:
When the cable head end transmits a piece of data, that goes to every single household that is connected to that cable head end.

In contrast, with DSL, when a household PC downloads something from the Internet, that goes only to that household.

—>Why was cable designed to be a shared broadcast medium?
Because it was originally designed to carry a TV signal.
The traditional TV model is that somebody broadcasts a given piece of content, whoever is interested turns on the TV set and watches.



Cable

- Cable modem + cable line (copper, fiber)
- 2 channels (downstream, upstream)
- typically 100s of Mbps
  - most allocated to the downstream channel
- shared broadcast medium

Cable is a technology used by cable companies to connect households to the Internet.

It relies on a cable modem and a cable TV line, which is made of copper and fiber.

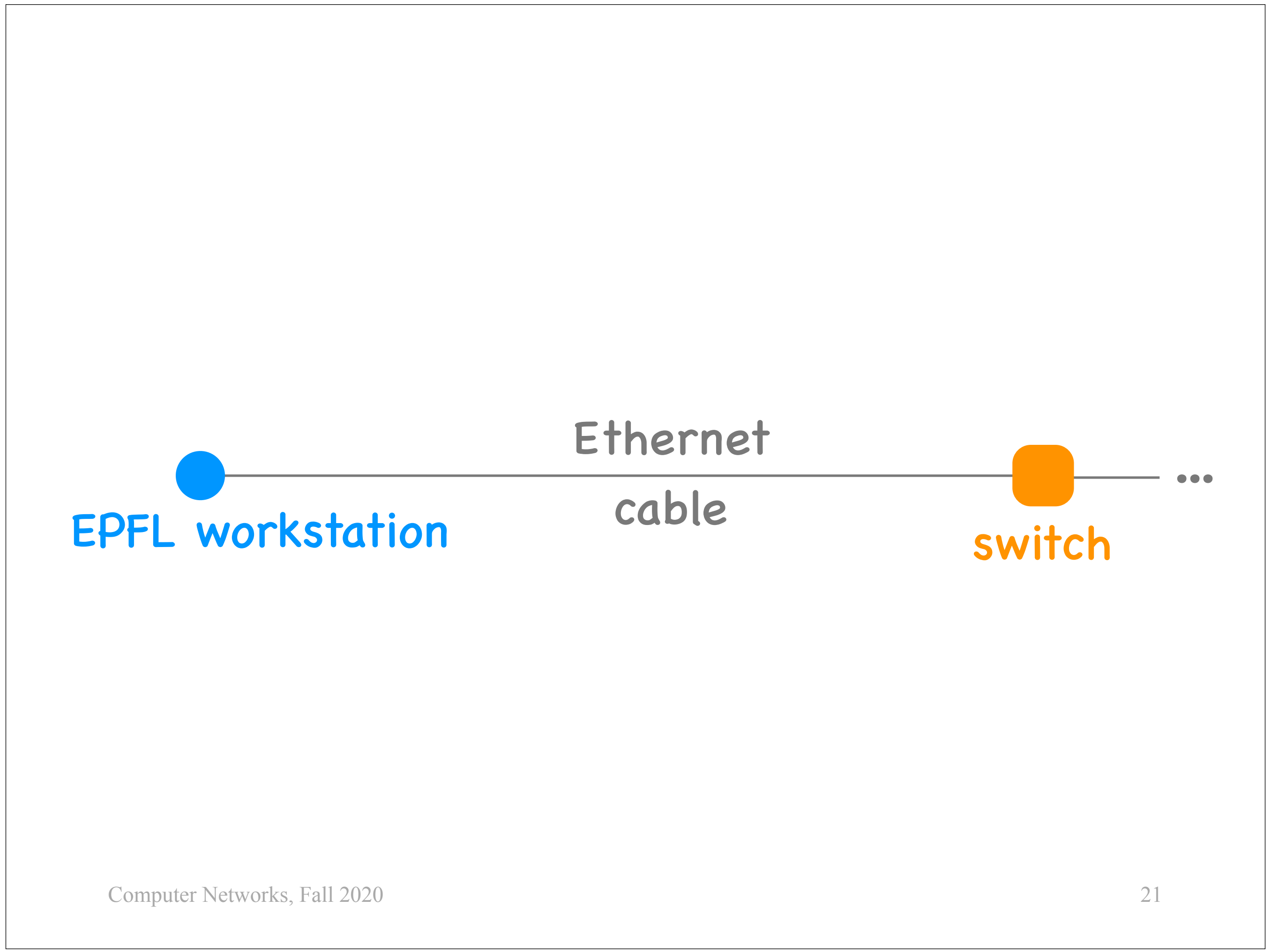
There are two channels:

- one for downstream data (from the Internet to the home PC), and
- one for upstream data (from the home PC to the Internet)

The supported data rate is typically 100s of Mbps,
and most of it is allocated to the downstream channel.

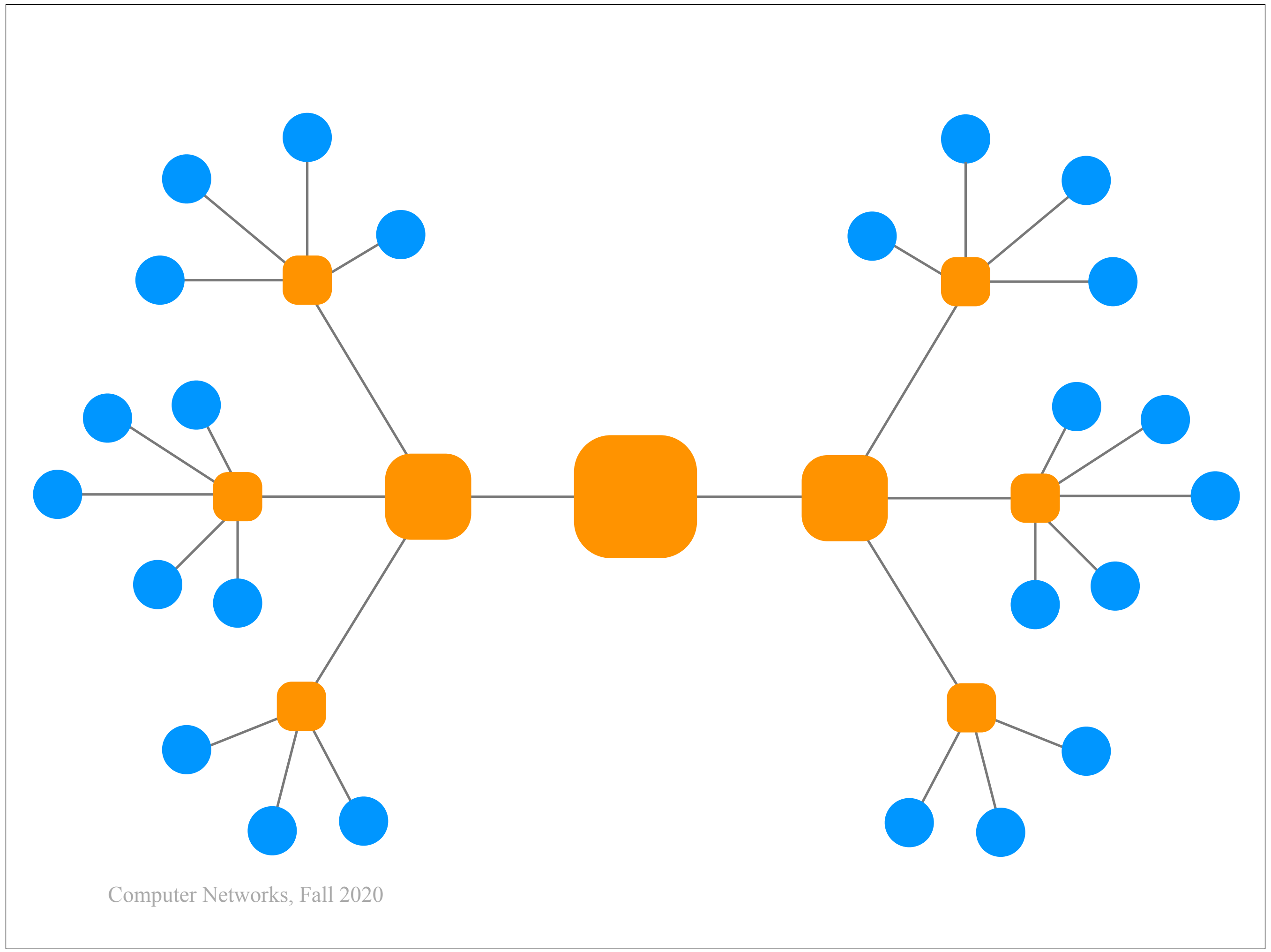
And, unlike DSL, cable is a shared broadcast medium,
which means that end-systems can affect one another.

—>Why is cable performance bad on Saturday night?

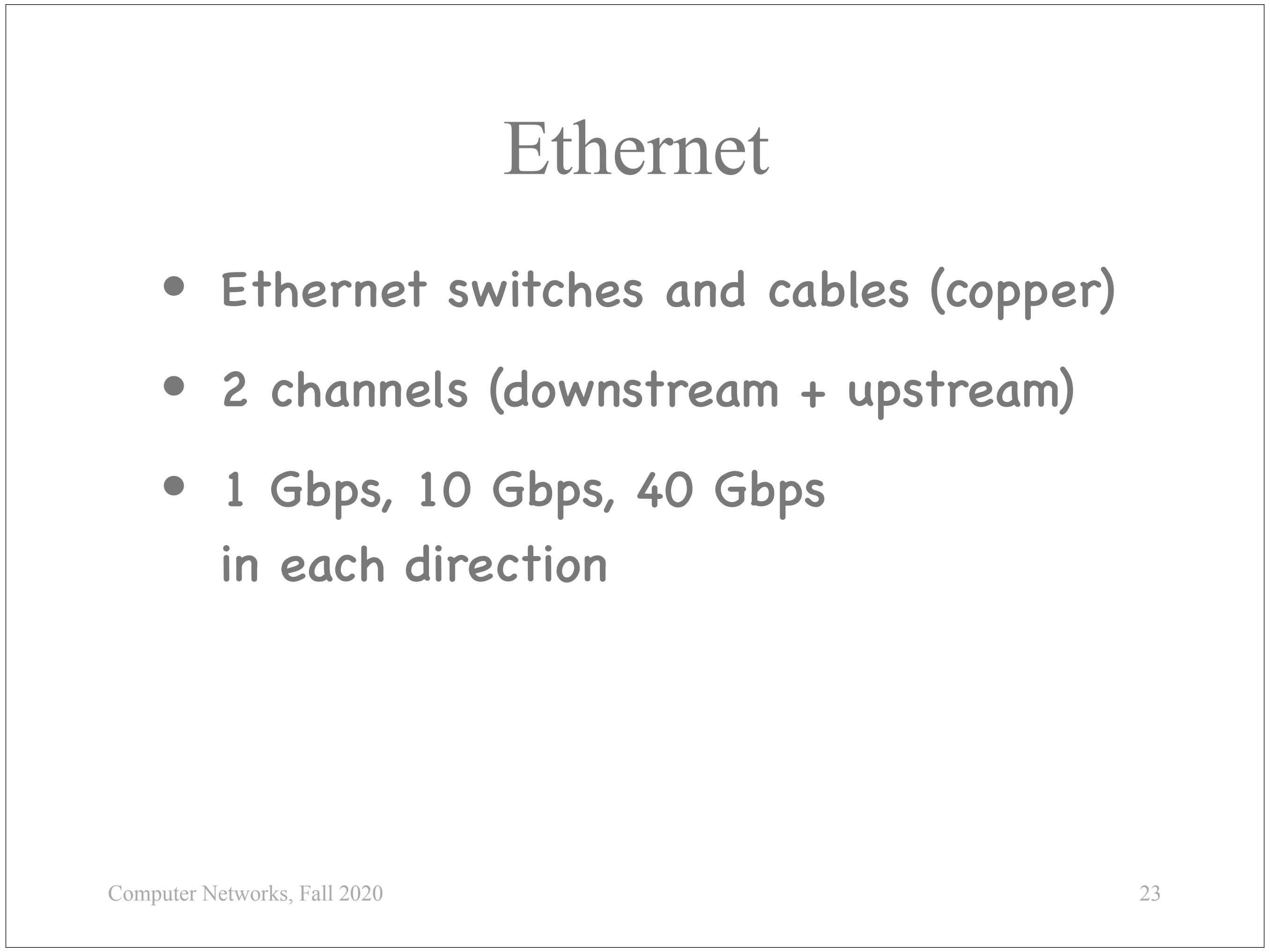


There exist many end-systems that are connected to the Internet neither through DSL nor cable. For example, the workstations that you will use to do the lab exercises.

Each of these workstations is directly connected to a switch through an Ethernet cable.



More precisely, in many universities, companies, and data-centers, the networks is organised in a **hierarchy**:
there are many workstations connected to small switches,
then many small switches connected to medium switches,
then many medium switches connected to big switches.



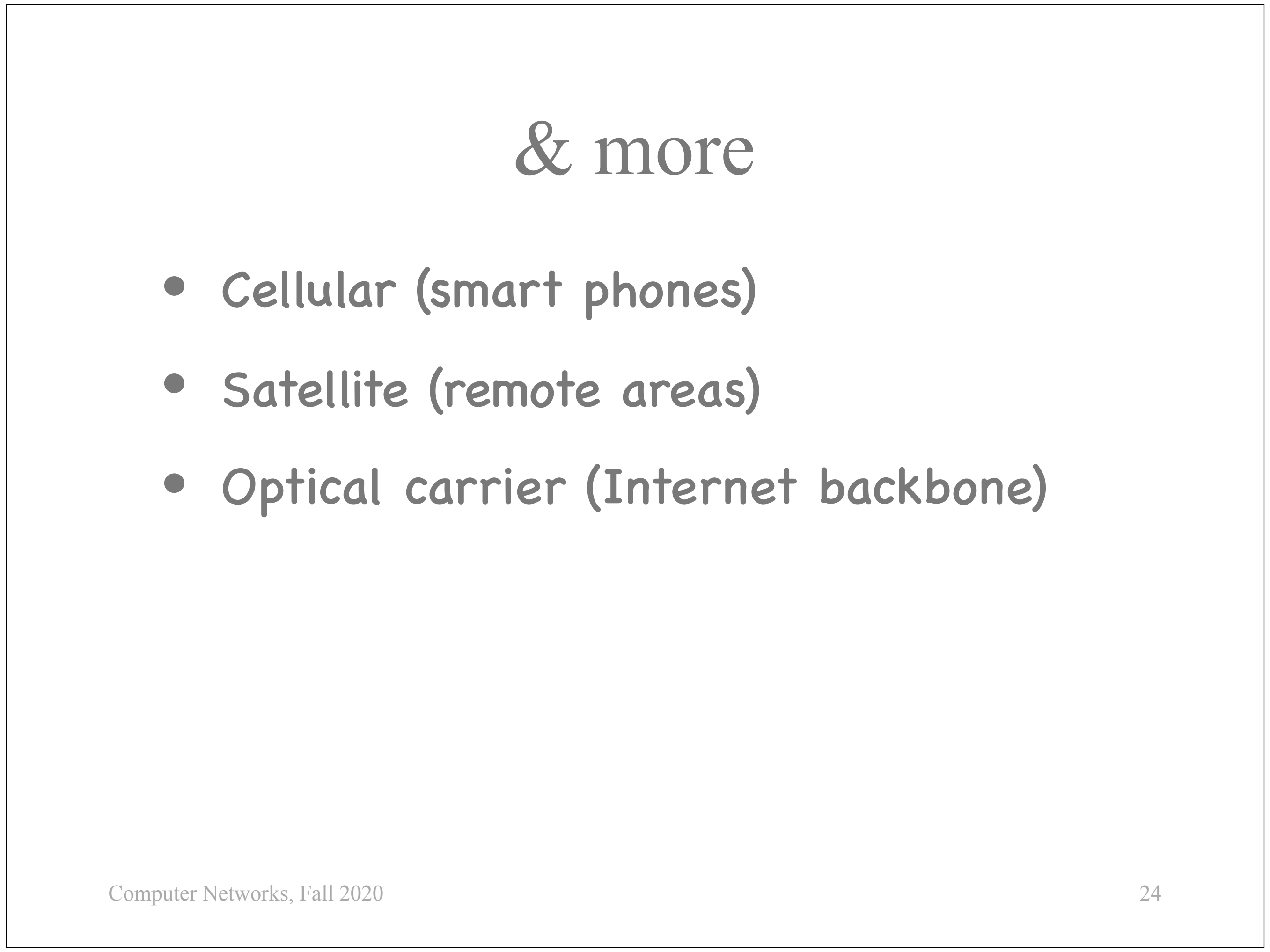
Ethernet

- Ethernet switches and cables (copper)
- 2 channels (downstream + upstream)
- 1 Gbps, 10 Gbps, 40 Gbps
  in each direction

Ethernet is a technology used to connect end-systems and switches to one another.

It is often used in universities, companies, and data-centers,
to connect workstations and servers between them and to the Internet.

It relies on Ethernet switches and cables, which are made of copper.
Each cable has 2 channels, and each channel supports from 1 Gbps to 40 Gbps, with 100Gbps coming up in the next few years.



& more

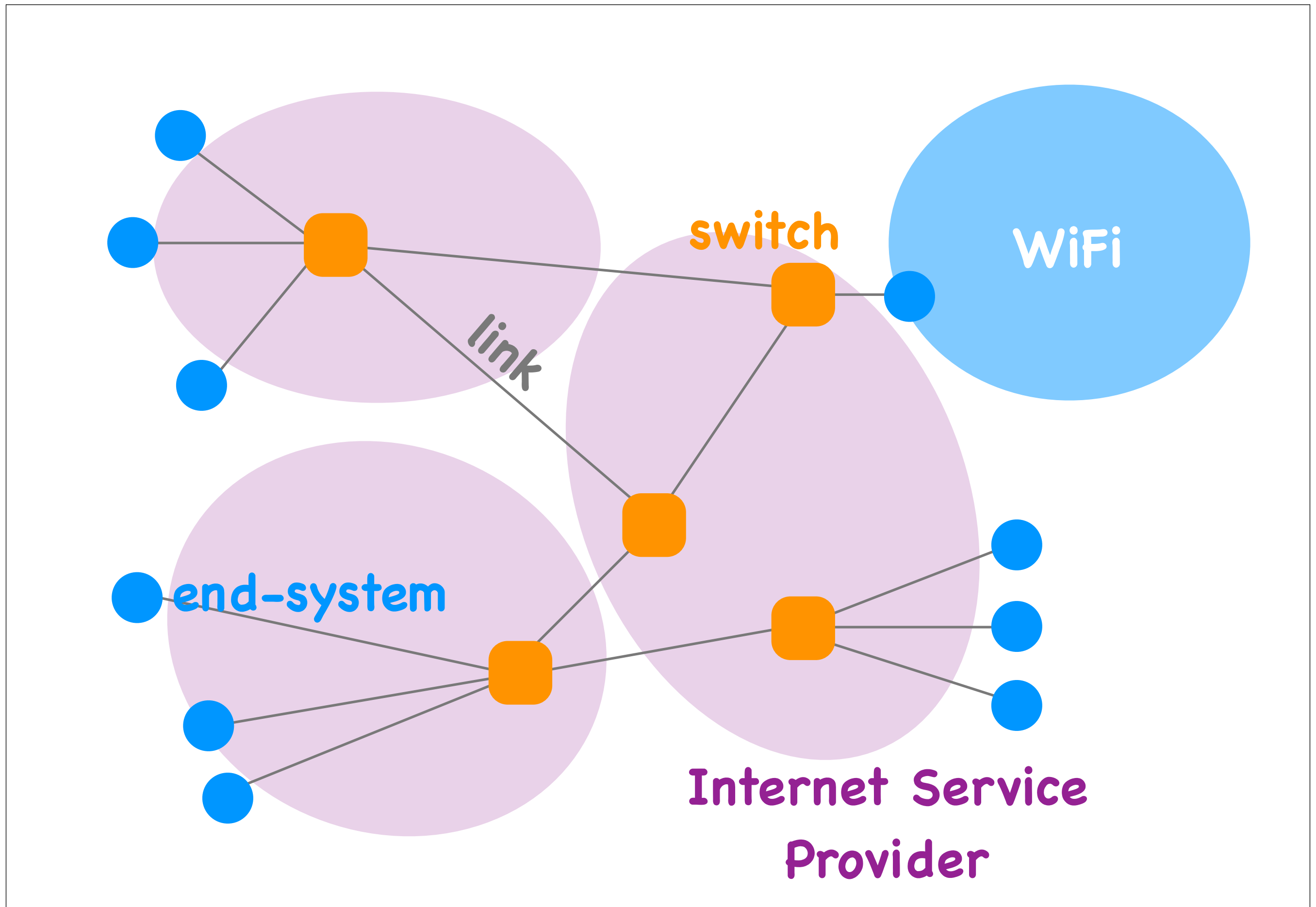
- Cellular (smart phones)
- Satellite (remote areas)
- Optical carrier (Internet backbone)

There are several other kinds of technologies used to interconnect end-systems and packet switches.
Basically, any technology that we were already using for some other kind of communication, we are reusing for Internet connectivity.

—> What technology other than traditional phone and cable were we already using for communication?

—> Let’s say you are a scientist and you are on a mission to the North pole. No phone, no cable, no cellphone coverage. What technology would you use?

And there is something called Optical carrier, which relies on fiber, and it typically interconnects packet switches inside and between Internet Service Providers.



And we are back to the big picture.

There is one kind of technology that you use all the time to connect to the Internet,
and I did not mention it.

WiFi. The wireless technology that you use to connect your laptop to the EPFL network, or to a Starbucks hotspot.

Where does WiFi fit in this picture?

A WiFi network is typically attached to a device that is itself connected to the rest of the Internet using one of the technologies we already discussed.

How do you setup WiFi access at home?

First, you use DSL or Cable to connect to the rest of the Internet.

Then you buy a “wireless router”, you connect it to your DSL or cable modem, you configure it, and that’s it.

So, your wireless router acts as an intermediary that relays the data between your laptop and the Internet.

What physical infrastructure is already available?

A few decades from now, when you are successful computer and communications engineers, many of these technologies will be obsolete.

The performance numbers will certainly have changed.

But what you should keep is that, when you are asked to deploy a new network, or expand the range of a network, you should always ask “what physical infrastructure is already in place”?

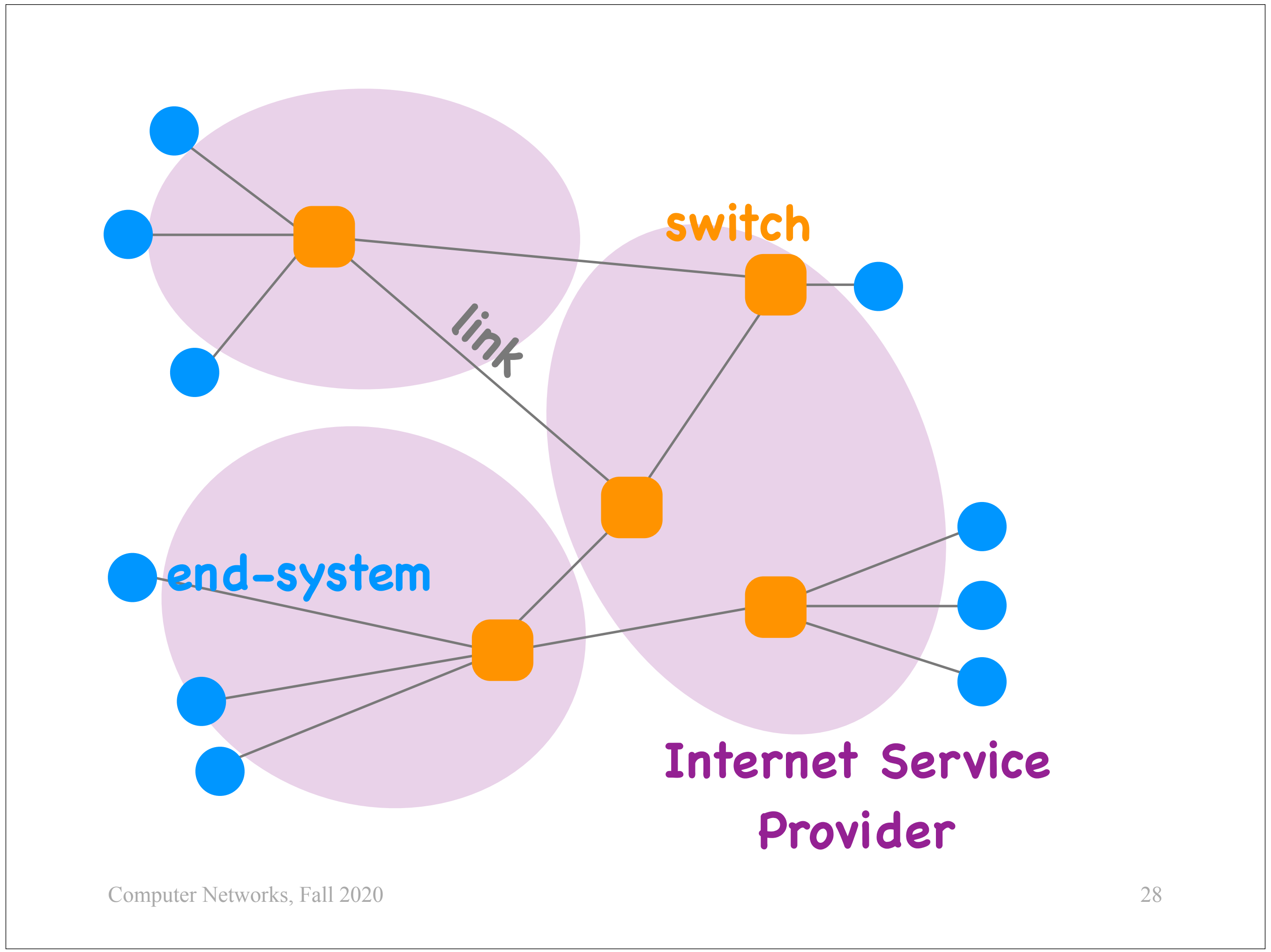
What media are people already familiar with and used to paying for?

The answers will help you decide how to build your network.

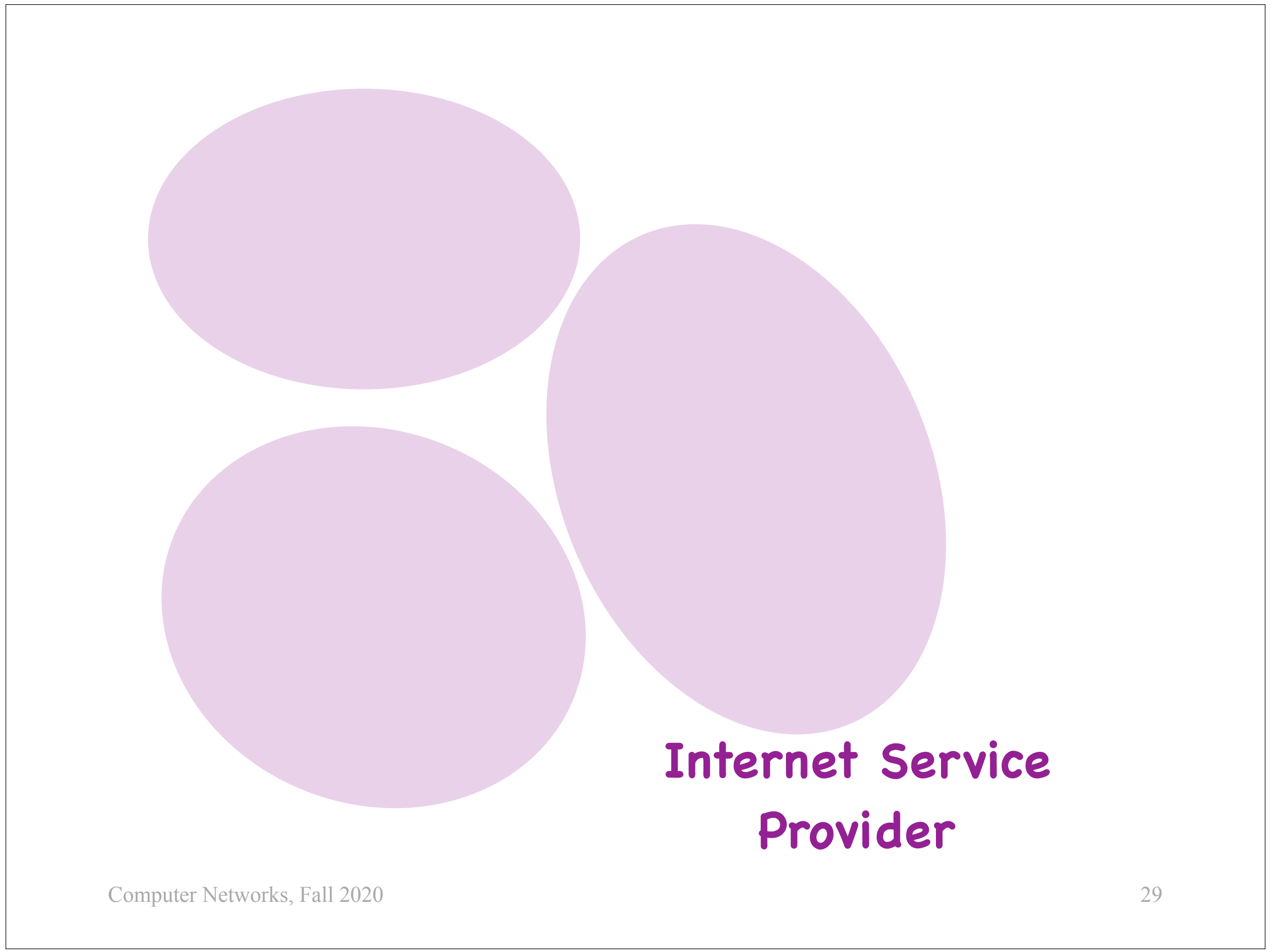
Questions

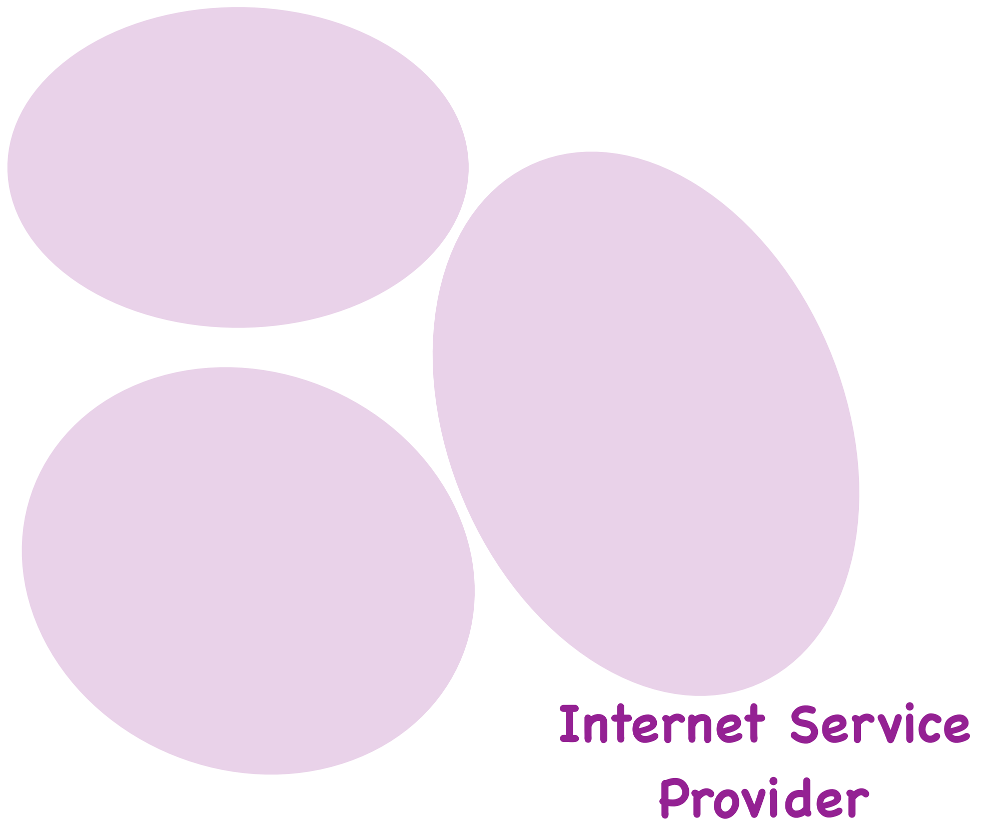
- What's underneath?
- Who owns what?
- How does it work?
- How do we evaluate it?
- How do we share it?

Who owns and manages the Internet infrastructure?



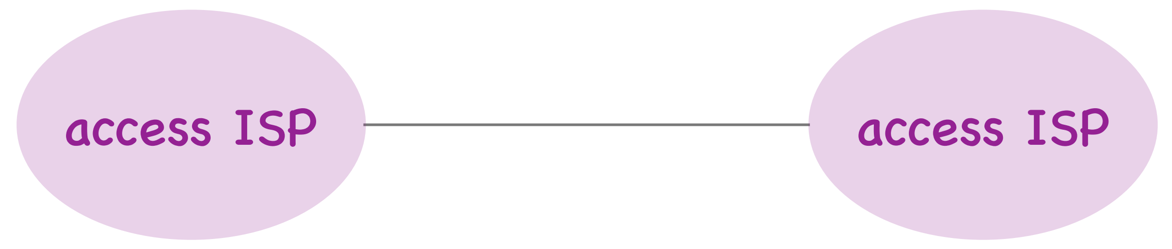
We said that a big chunk of the switches and links on the Internet are owned and managed by Internet service providers (ISPs).





Let’s forget, for a moment, about end-systems, switches, and links, and focus only on ISPs.

We said earlier that ISPs are organised in a hierarchy.
What does that mean?



Consider two **access ISPs**.

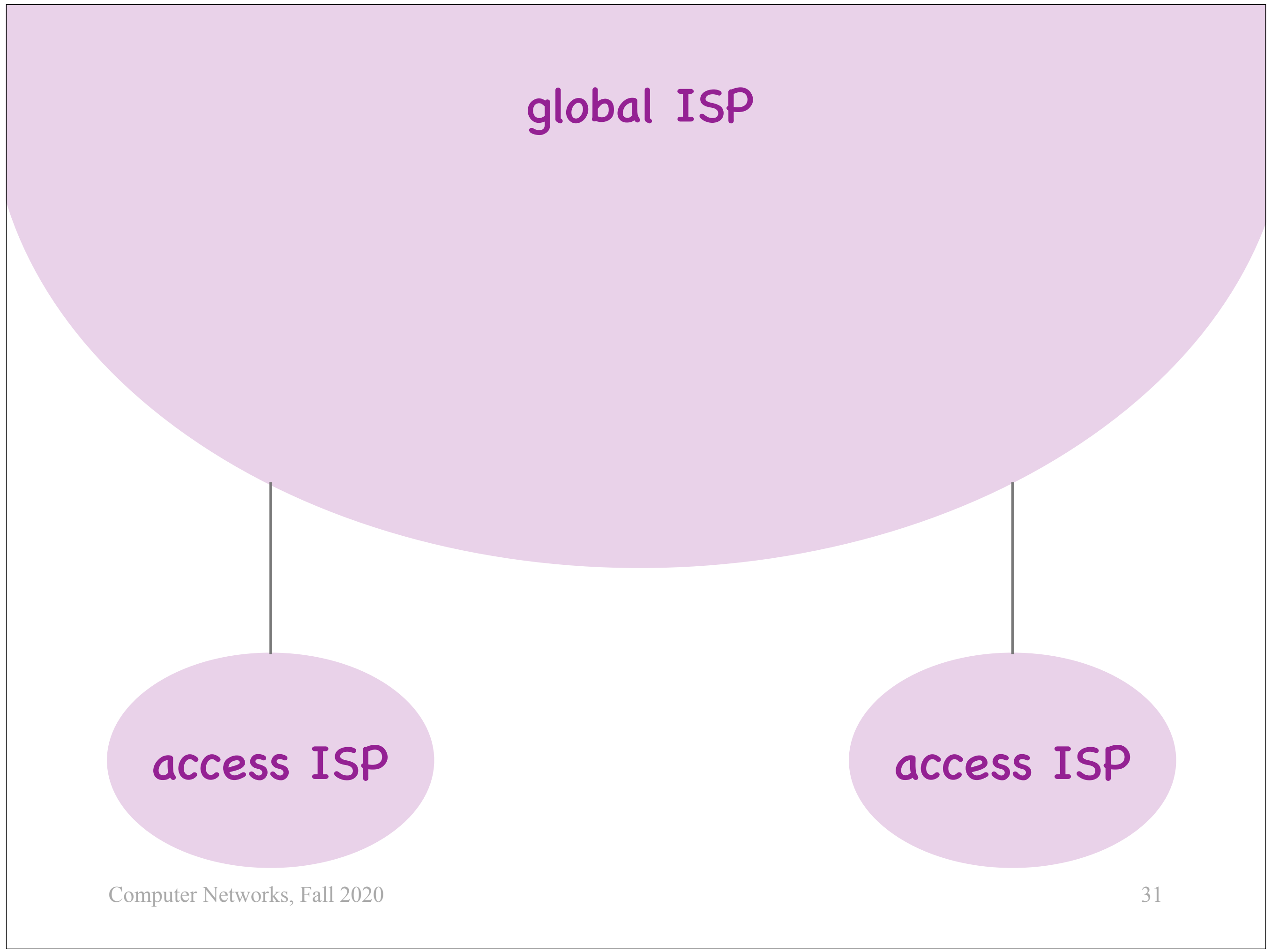
These are ISPs, like Swisscom and Cablecom, that directly interface with end-systems and provide Internet connectivity, e.g., through DSL and cable as we saw earlier.

Every access ISP in the world needs to be connected to every other access ISP in the world
(such that all Internet end-systems can communicate with each other).

One approach would be to connect the two access ISPs directly, i.e., draw a physical link between them.
Is this a good idea?

How many phone companies and cable companies and universities and enterprises are there in the world?
Could we possibly draw a physical link between every single pair of them?

No.

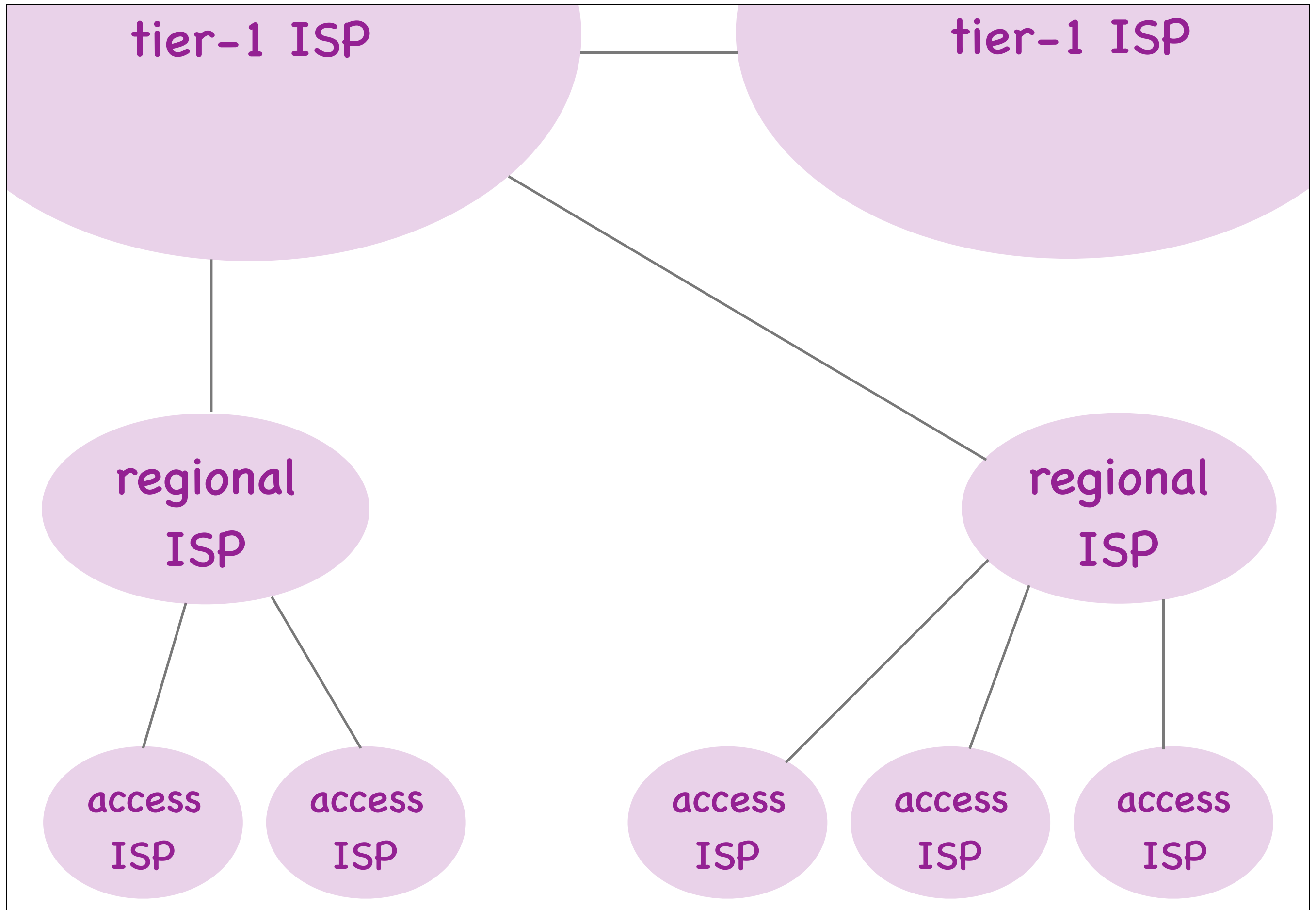


A more reasonable approach would be to create a hierarchy of ISPs.

E.g., we could have one global ISP, and each access ISP would be directly connected only to this one global ISP.
Then the job of the global ISP would be to provide connectivity between all the pairs of access ISPs in the world.

There do exist very large ISPs today that cover multiple continents, but I am not aware of any ISP that has truly global presence.

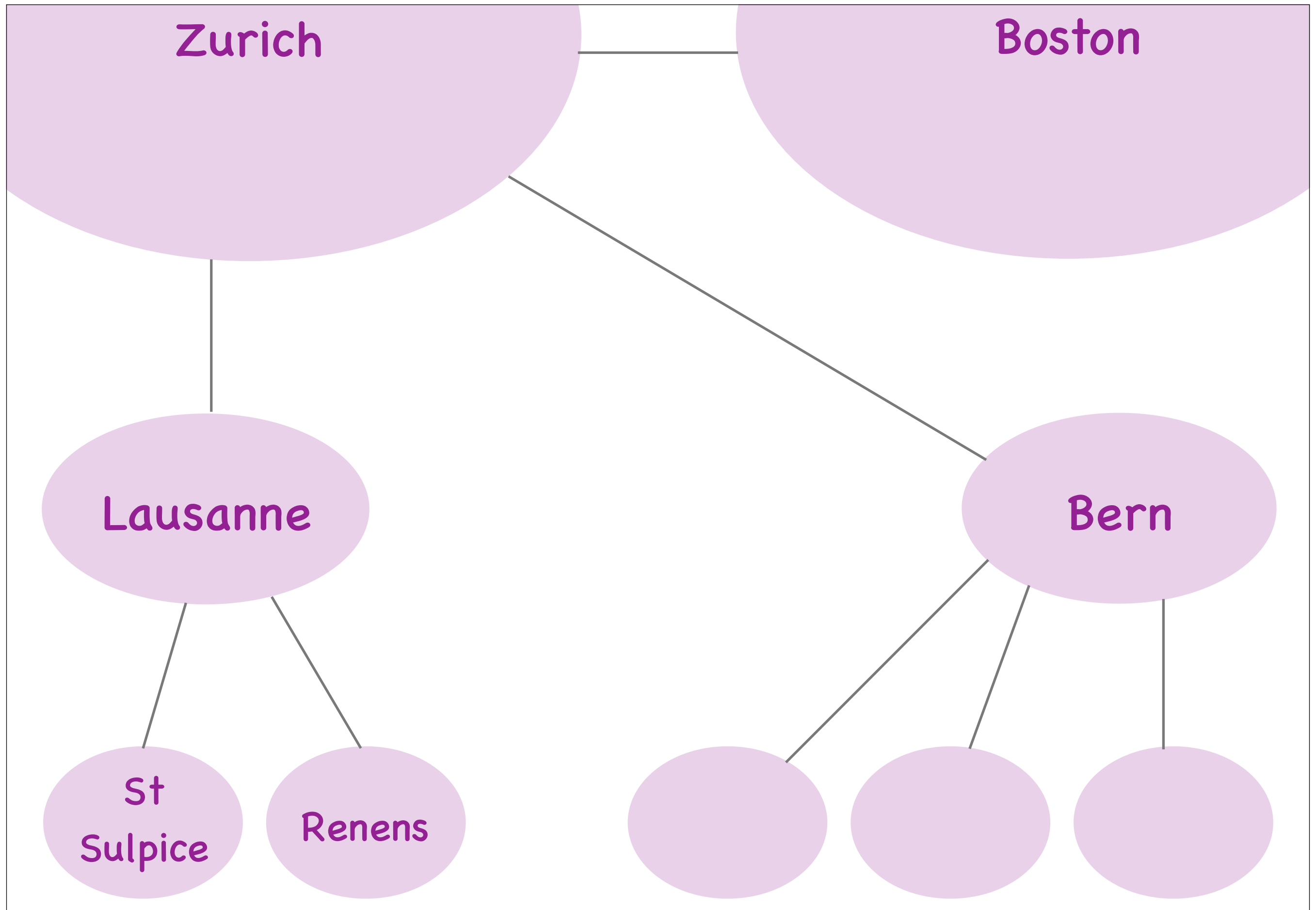
Instead, what we have is a



three-level hierarchy:

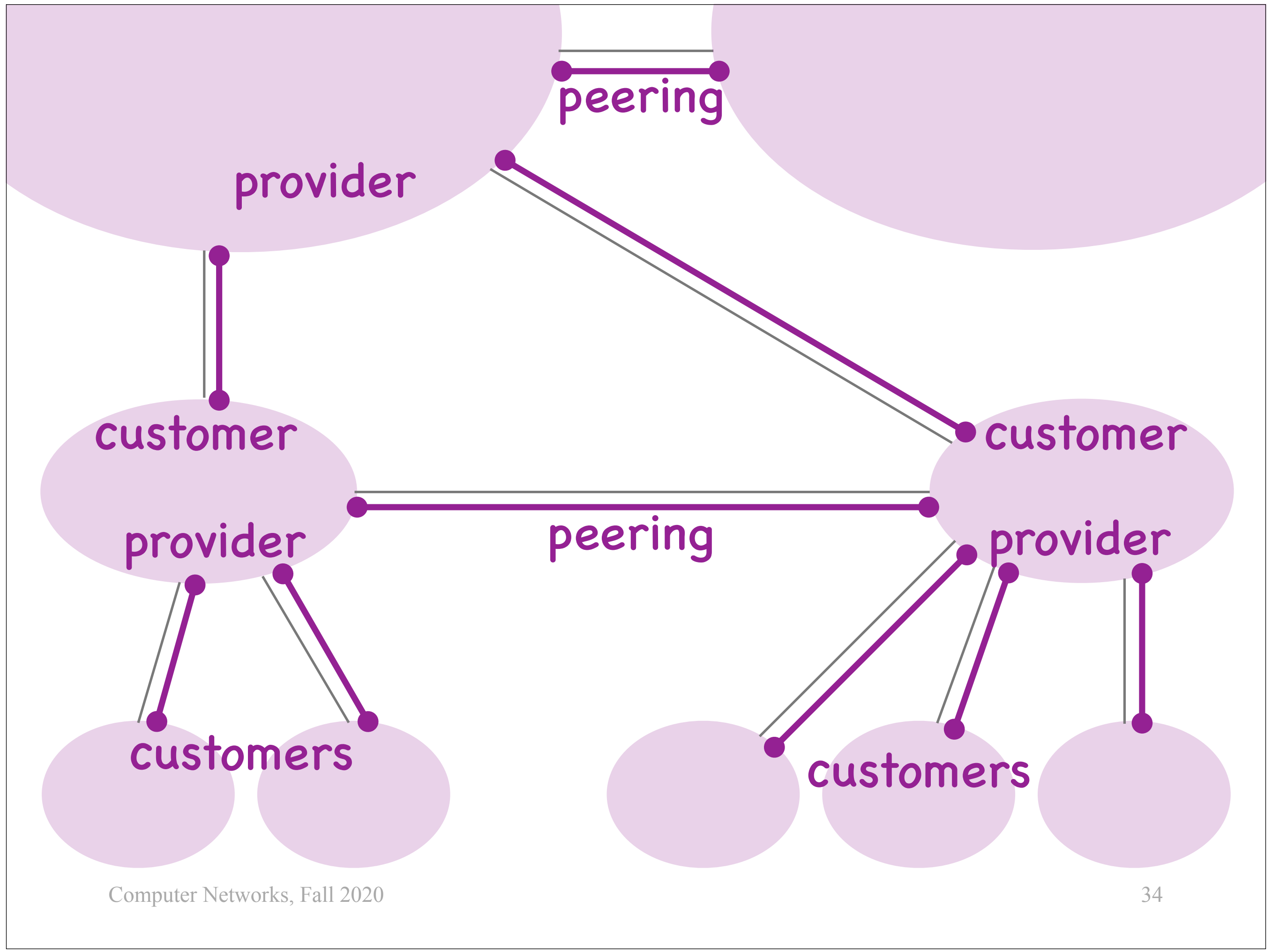
- At the bottom we have lots and lots of access ISPs,
- which connect to fewer regional ISPs,
- which connect to even fewer tier-1 ISPs.

The tier-1 ISPs are directly connected to each other.



You can think of this like the international network of transportation:

- The access ISPs are like the small towns and villages that are connected to the cities by bus or metro.
- The regional ISPs are like cities that are connected to bigger cities by trains or regional flights.
- The Tier-1 ISPs are like the bigger cities that are connected by international flights.



This is not only a physical,
but also an economic hierarchy:

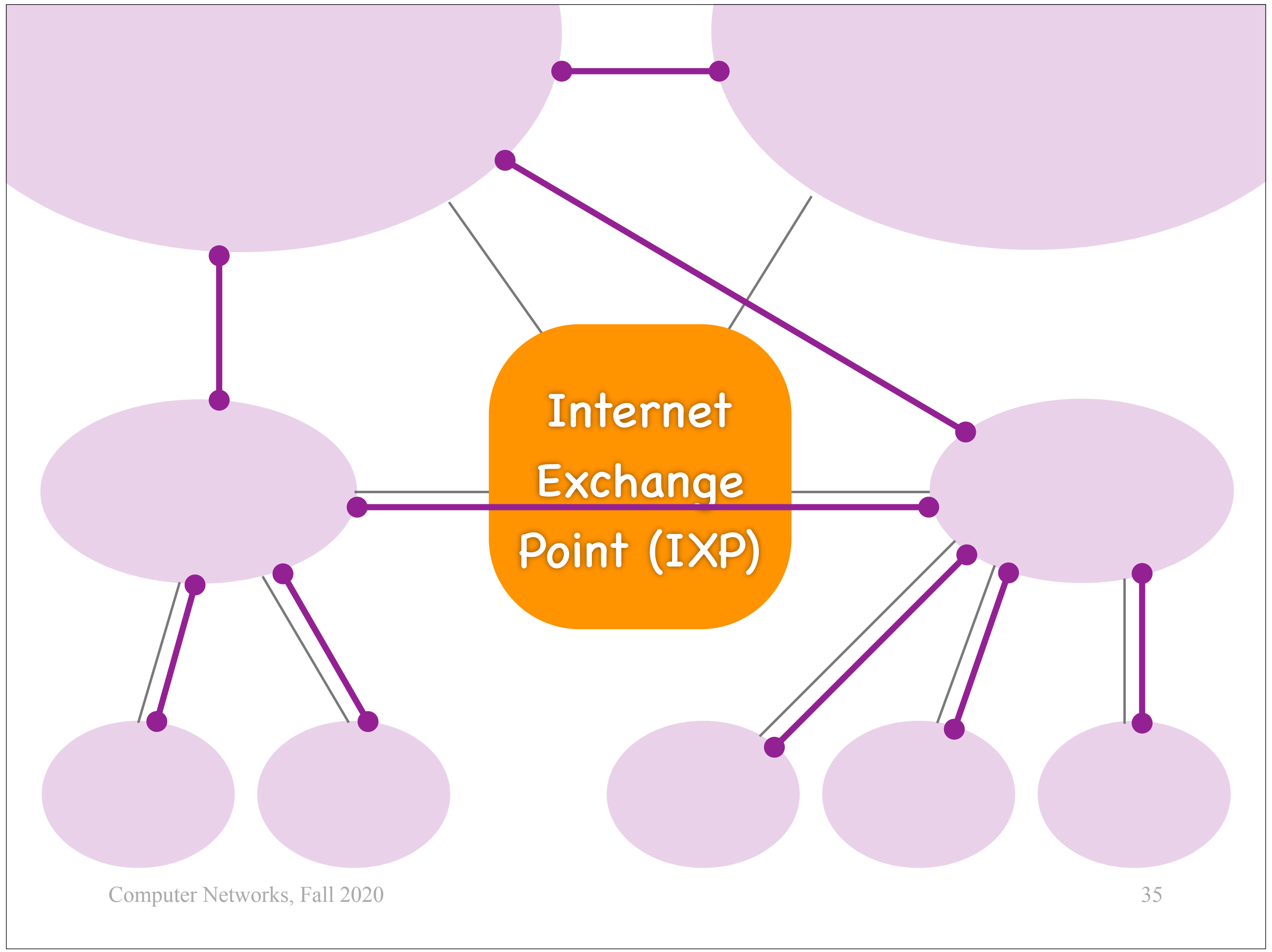
- The access ISPs pay the regional ISPs for connectivity, so access and regional ISPs have a customer-provider relationship.
  - In this picture, I am using gray lines to represent physical connections and purple lines to represent business connections.
- The regional ISPs pay the tier-1 ISPs for connectivity, so regional and tier-1 ISPs also have a customer-provider relationship.
- The Tier-1 ISPs are nobody's customers, but of course they need to exchange traffic with each other,
  and they do that under special **peering agreements**.
  - A peering agreement can be as simple as nobody pays anyone and we agree to exchange traffic such that our customers can exchange traffic…
  - Or much more sophisticated, for instance, specifying how much one peer will pay the other as a function of the imbalance of the traffic rate flowing in each direction.

But, the picture is not that simple.
Some regional ISPs may be directly connected to each other.
For instance, consider two regional ISPs that are both located in Switzerland.
Suppose their customers exchange a lot of traffic.
The two regional ISPs may decide that, instead of paying their tier-1 ISPs to exchange all that traffic,
they are better off connecting directly to each other and exchanging that traffic for free.
In this case, we call these two regional ISPs peers.

So, what we have today is a three-tier hierarchy, where ISPs at lower levels are customers of ISPs at higher levels,
but we also have peering relationships at all levels of the hierarchy.

And it gets more complicated.

Many ISPs that have a direct business relationship are actually

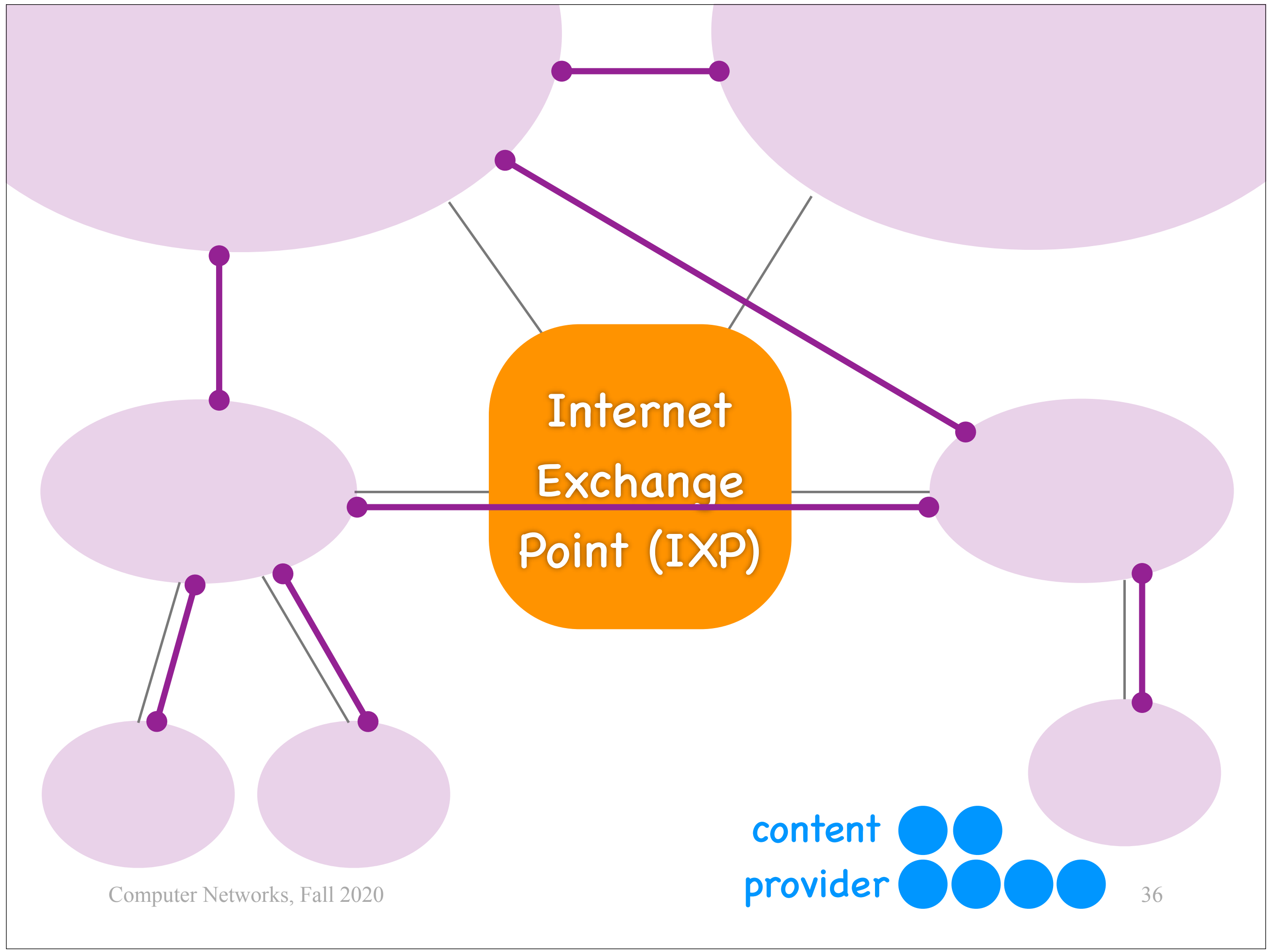


not directly connected to each other physically.

Instead, they are directly connected physically to something that is called an “Internet eXchange Point” (IXP), which is essentially a giant switch that provides physical connections.

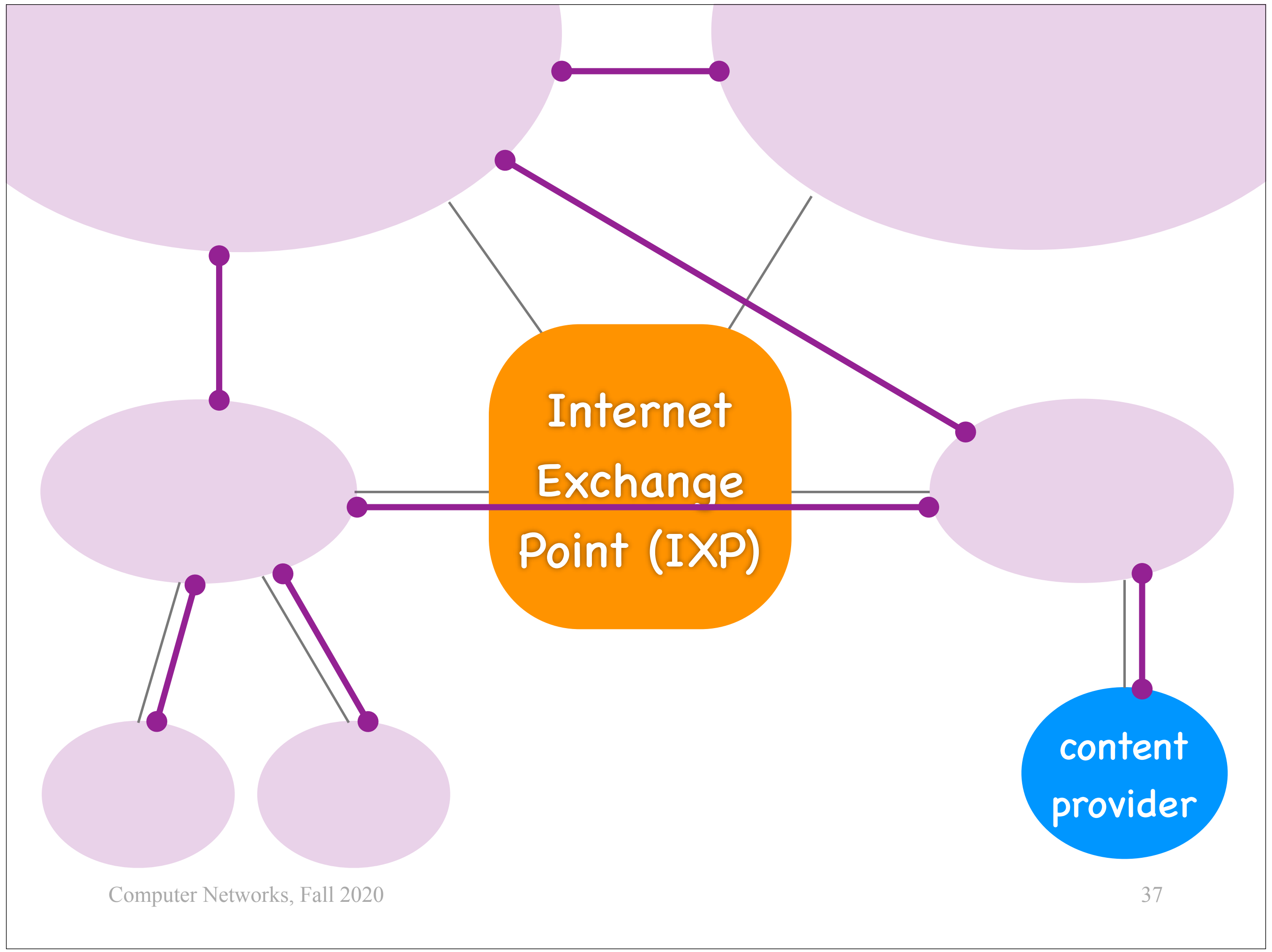
So, two ISPs may have a provider/customer or peering business relationship without being directly connected to each other physically.

Why does this happen?
Because it is simpler logistically for an ISP to draw a single cable to an IXP, than to draw one cable to each ISP it has a business relationship with.



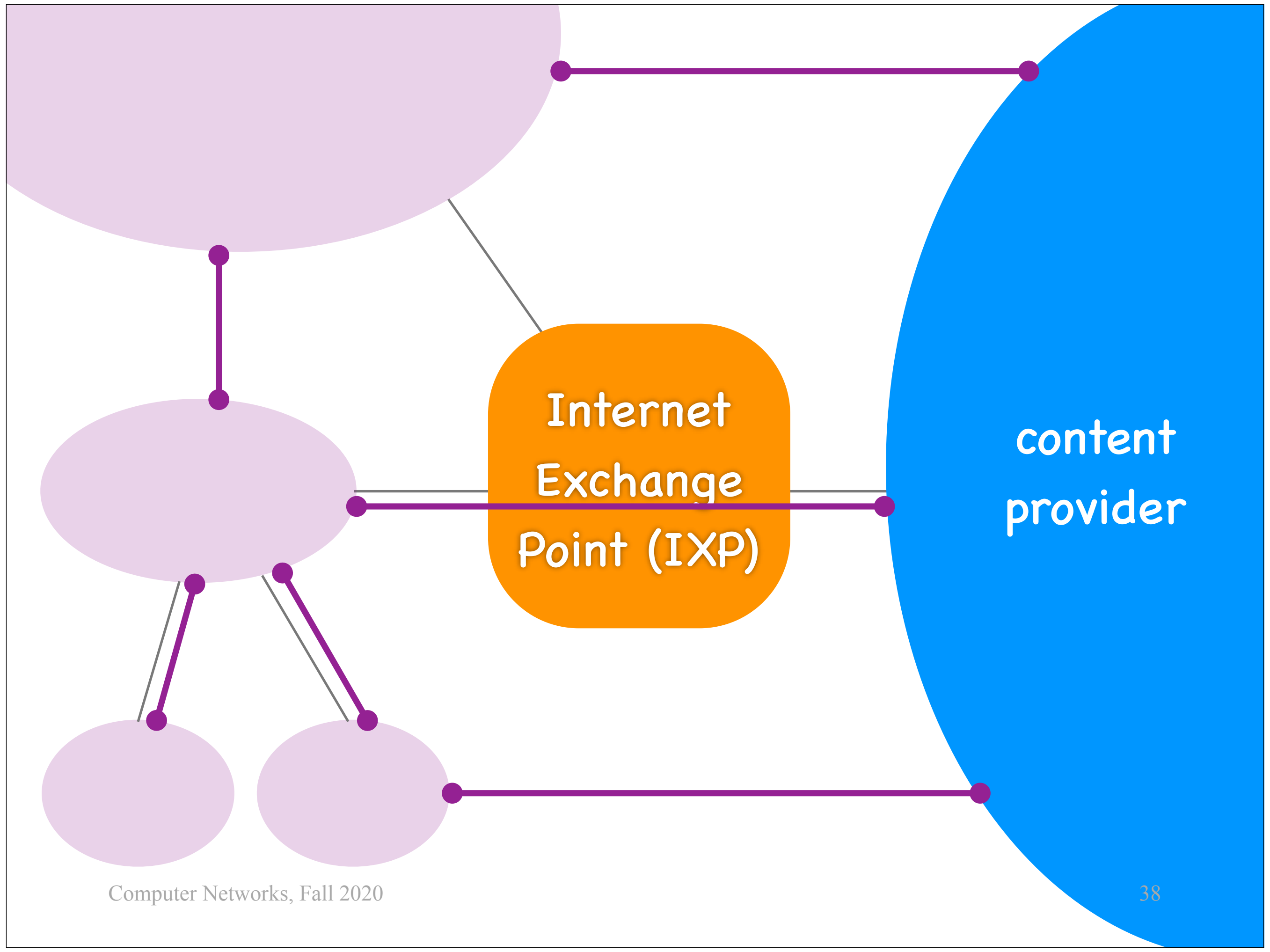
And then, there is Google.
You probably think of Google as a search engine, not an ISP.
And indeed, once upon a time, Google was a set of servers that provided the search-engine service,
which were connected to an access ISP, like all the other end-systems.

However, as the company grew and offered more services,
it became increasingly important to serve its content quickly to its users,
so it connected



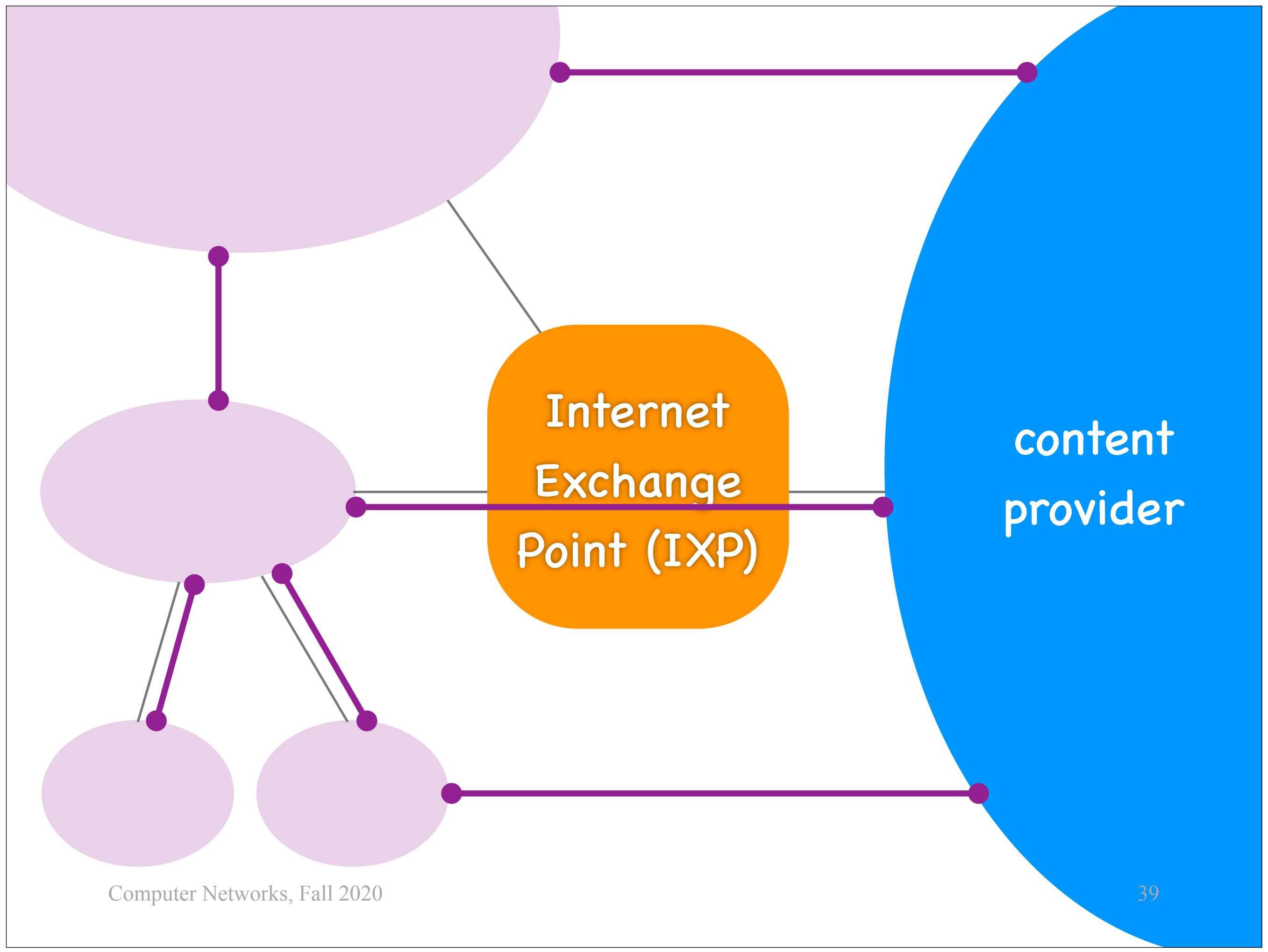
itself directly to large ISPs,
in a way, becoming itself the access ISP for its servers.

And then it grew even more, until…



it essentially built its own, almost global network,
which connects directly to other ISPs, Internet eXchange Points, even access ISPs.

And other content providers are adopting this model as well.



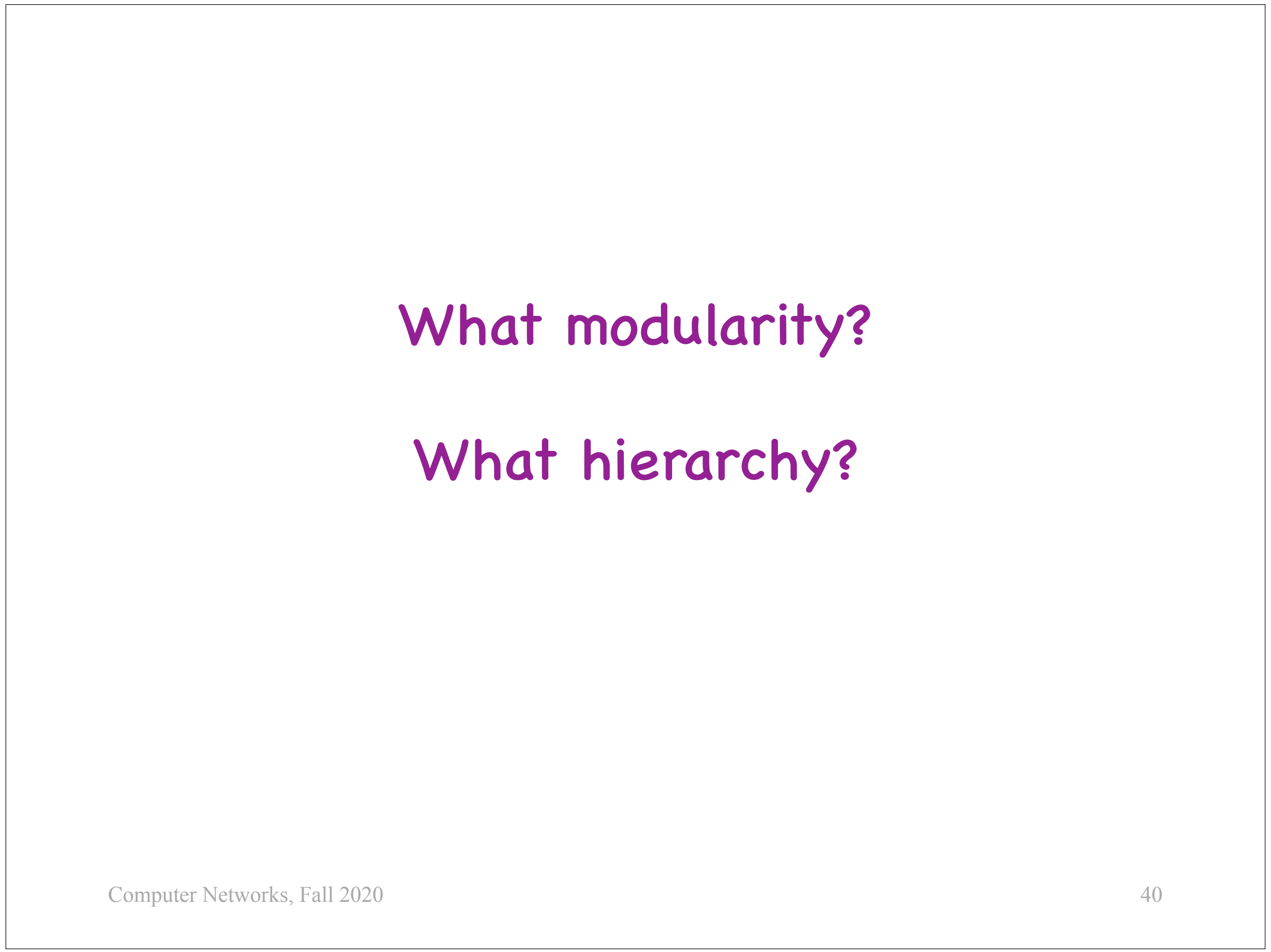
So, today we have a three-level hierarchy of ISPs:

- At the bottom, we have access ISPs (phone companies, cable companies, universities and so on), which provide connectivity to the end-systems (our laptops and smart phones).
- On top we have fewer and bigger regional ISPs.
- And on top we have even fewer and bigger tier-1 ISPs, which are connected to each other.

This hierarchy is not clean, i.e., sometimes regional ISPs connect directly to each other and bypass their tier-1 ISPs for economic reasons.

And sometimes ISPs connect to each other not directly, but through and Internet eXchange Point (IXP).

And then we have a few big content providers that have built their own networks and are peering with ISPs at all levels of the hierarchy.



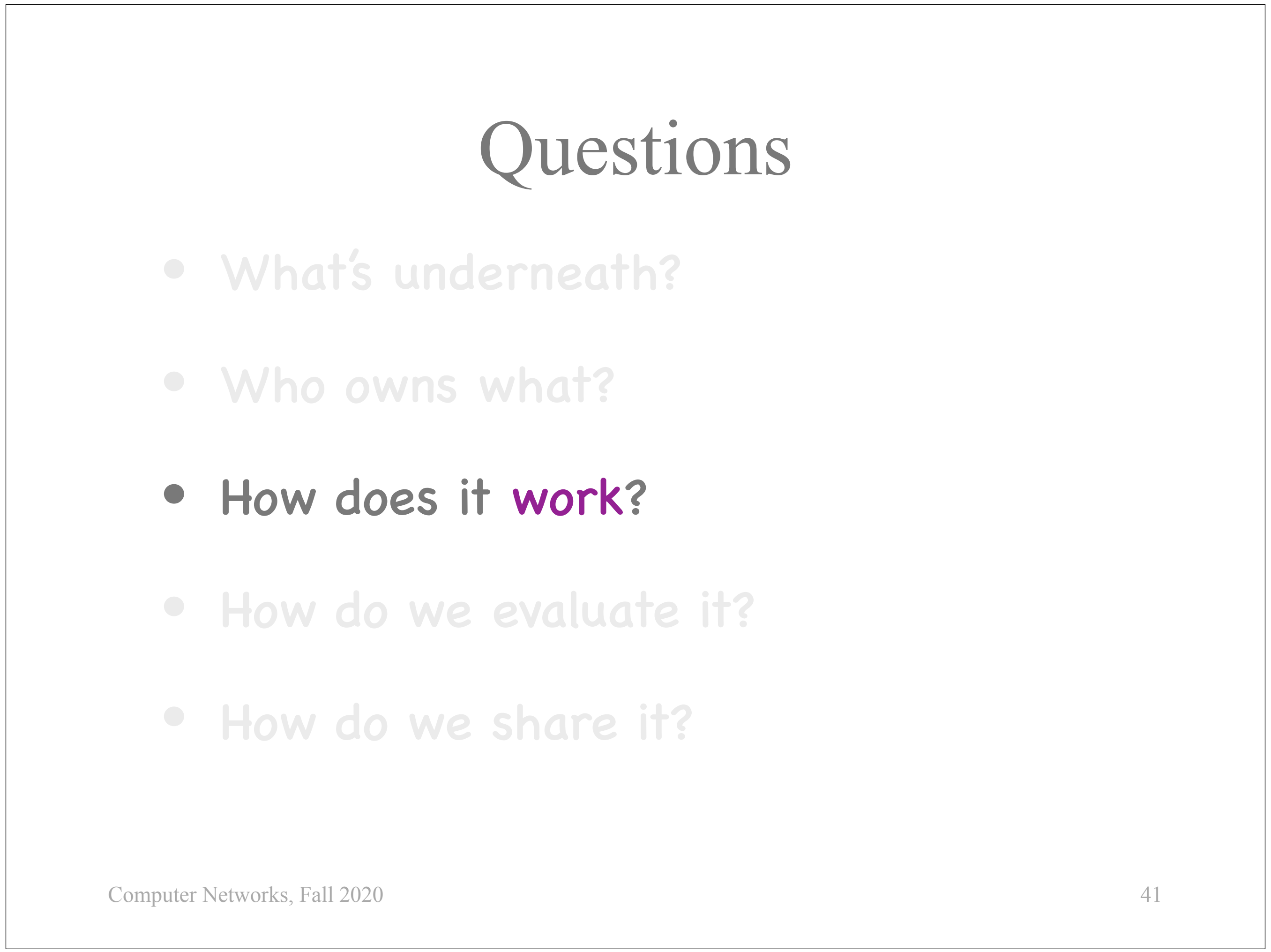
What modularity?

What hierarchy?

Here are two more classic questions in communication systems.

When you design a large communication system, you will need to cut it into modules (into pieces)
and you will typically need to organize these modules into a hierarchy.

So, you will need to decide how big each module,
how complex the hierarchy,
so that your system makes sense, both from a technical and an economic point of view.

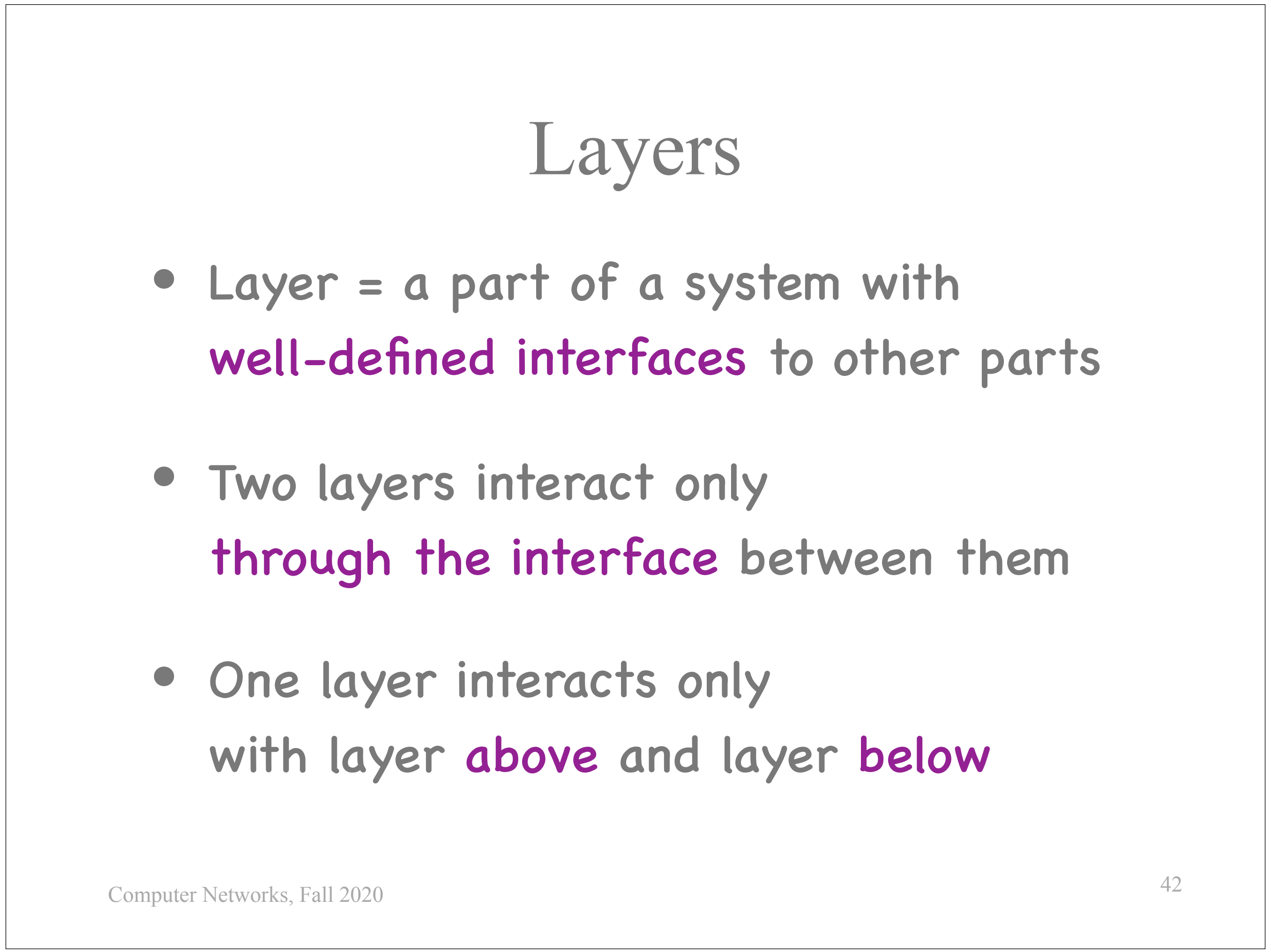


Questions

- What’s underneath?
- Who owns what?
- How does it work?
- How do we evaluate it?
- How do we share it?

How does the Internet work? What happens under the covers when distributed applications exchange messages?

The Internet architecture is based on…



Layers

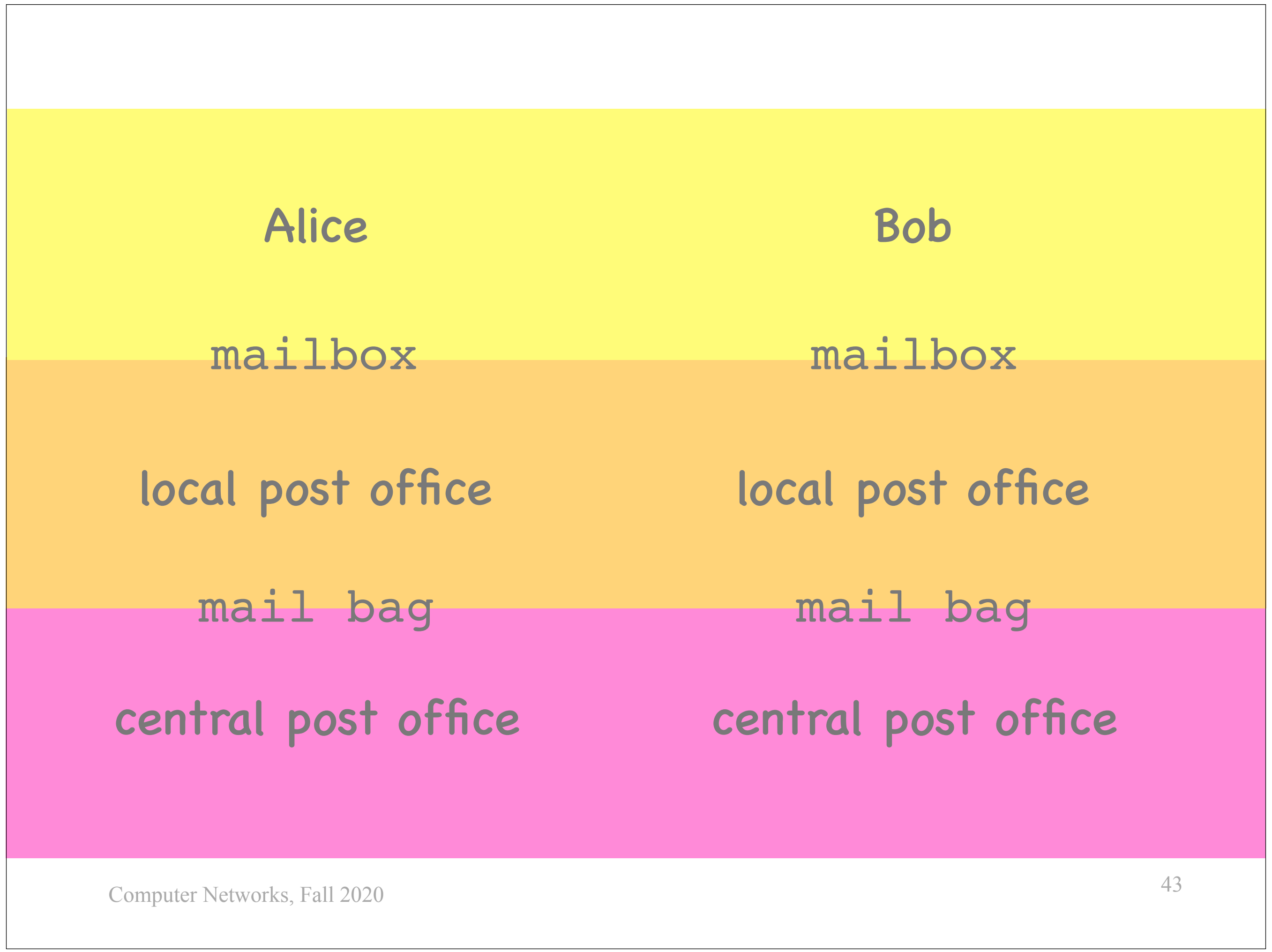
- Layer = a part of a system with **well-defined interfaces** to other parts
- Two layers interact only **through the interface** between them
- One layer interacts only with layer **above** and layer **below**

layers.

…

What does all this mean?

Let’s forget about computer networks for a moment.



Suppose Alice and Bob are human friends, and Alice wants to send a letter to Bob.

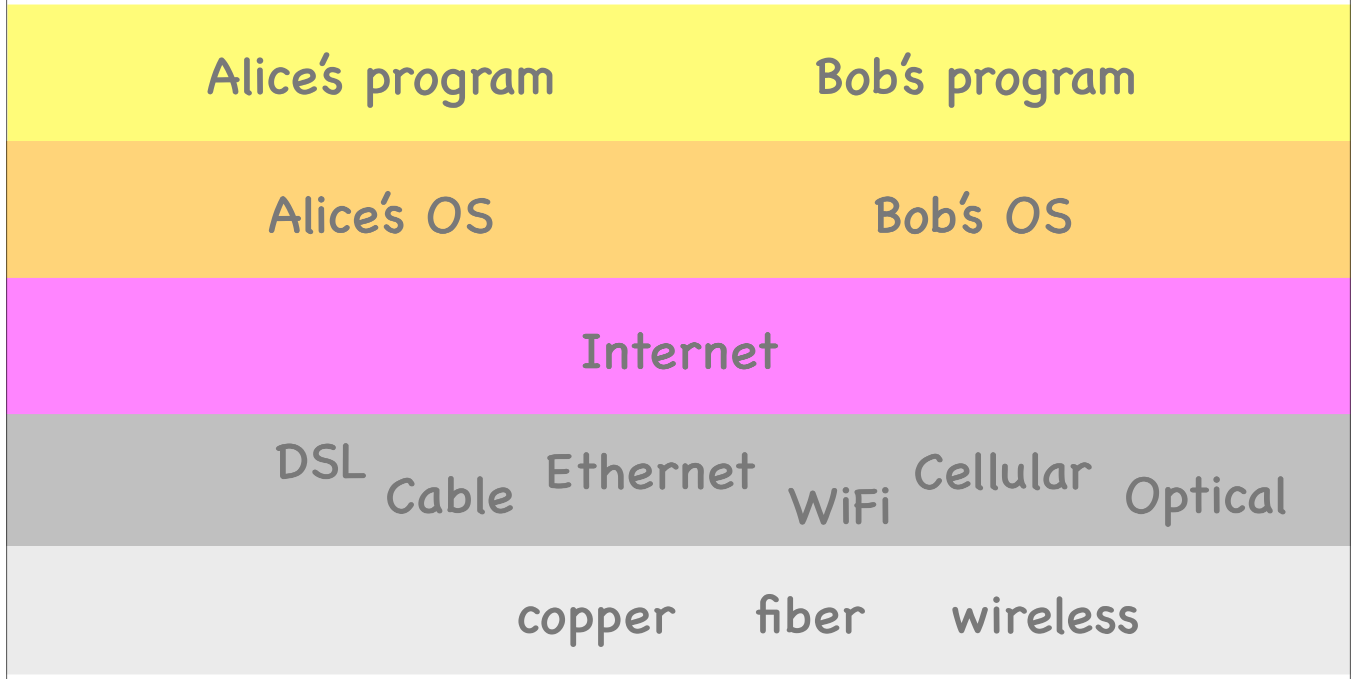
To do that, she puts the letter in a mailbox.
From there, the letter is taken to a local post office, where letters are sorted out and put in mail bags.
From there the letter is taken to a central post office,
then transferred potentially to another central post office, closer to Bob,
then to another local post office,
and then to Bob's mailbox.

One might argue that the postal system operates in layers.

At the top layer we have the users of the postal system,
at the middle layer we have the local post offices,
and at the bottom layer we have the central post offices.

What does it mean that Alice's local post office and Bob's local post office are "in the same layer"?
It means that they perform the same function, albeit to different users.

Moreover, there exists a well-defined interface between each pair of layers.
Between the users and the local post offices, there are the mailboxes.
Alice can only pass a letter to her local post office through the mailbox; she cannot throw it through the window of the local post office.
Moreover, Alice must use the mailbox correctly, otherwise her letter will be discarded: she must put it in an envelope, write the destination address at the proper place, and place a stamp at the proper place. That is the correct way to use this interface.

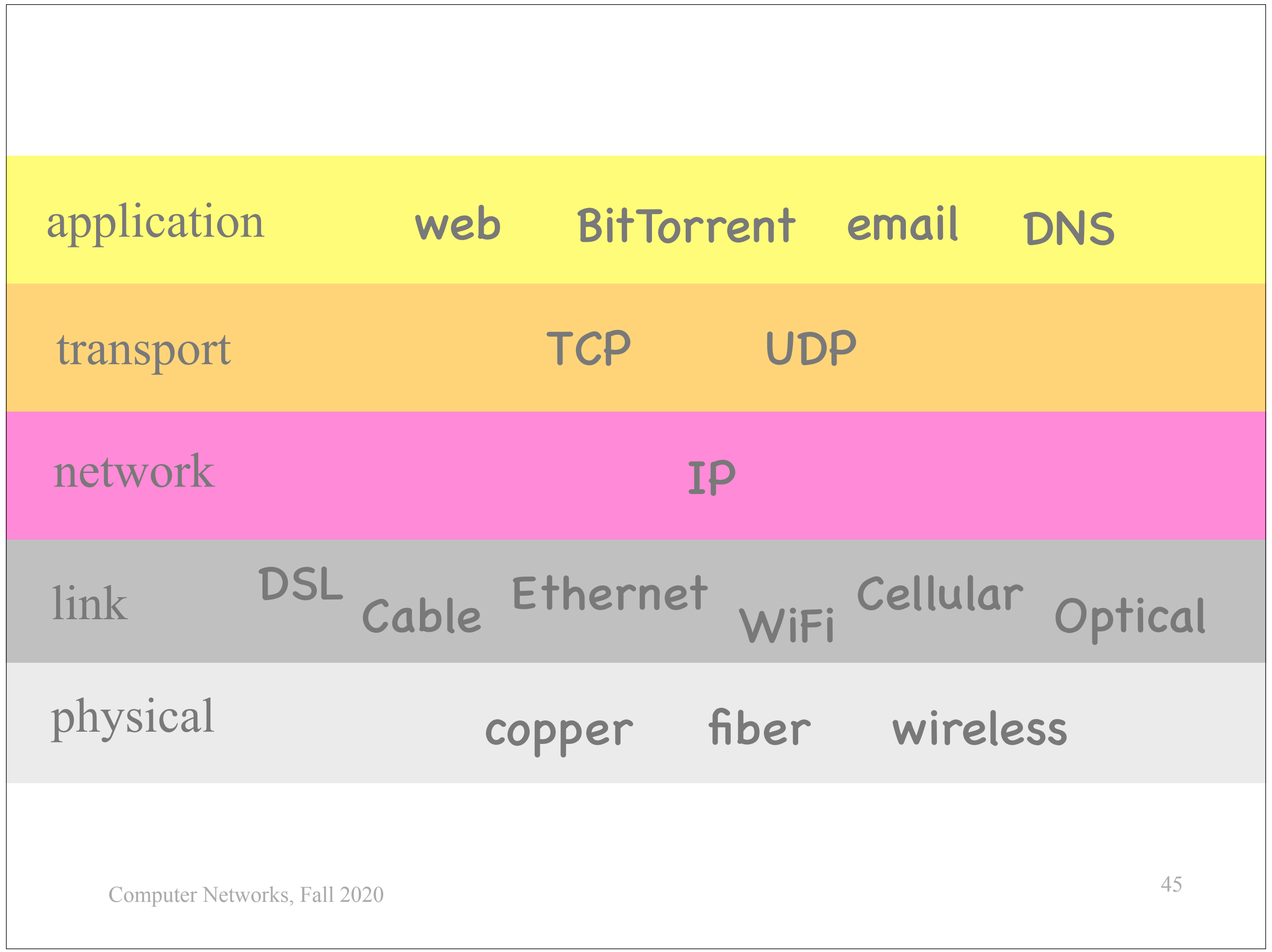


Now let's get back to computer networks.

Suppose Alice uses a program, running on her computer, to send a message to Bob.
When Alice presses "send", her program passes the message to the OS running on Alice's computer,
which, in turn, sends it out on the Internet.
Based on what we saw earlier, on the Internet, the message may traverse multiple "local networks,"
each of them using a potentially different underlying technology, like DSL, cable, etc.
Each of these technologies transforms the message into a sequence of electromagnetic waves and transmits it over coper or fiber or the air.

Eventually the message makes it to Bob's computer, where it is processed by the OS running there,
which passes it to Bob's program.

These 5 stages that process Alice's message **roughly** correspond to the 5 layers of the Internet architecture.



They have names: …

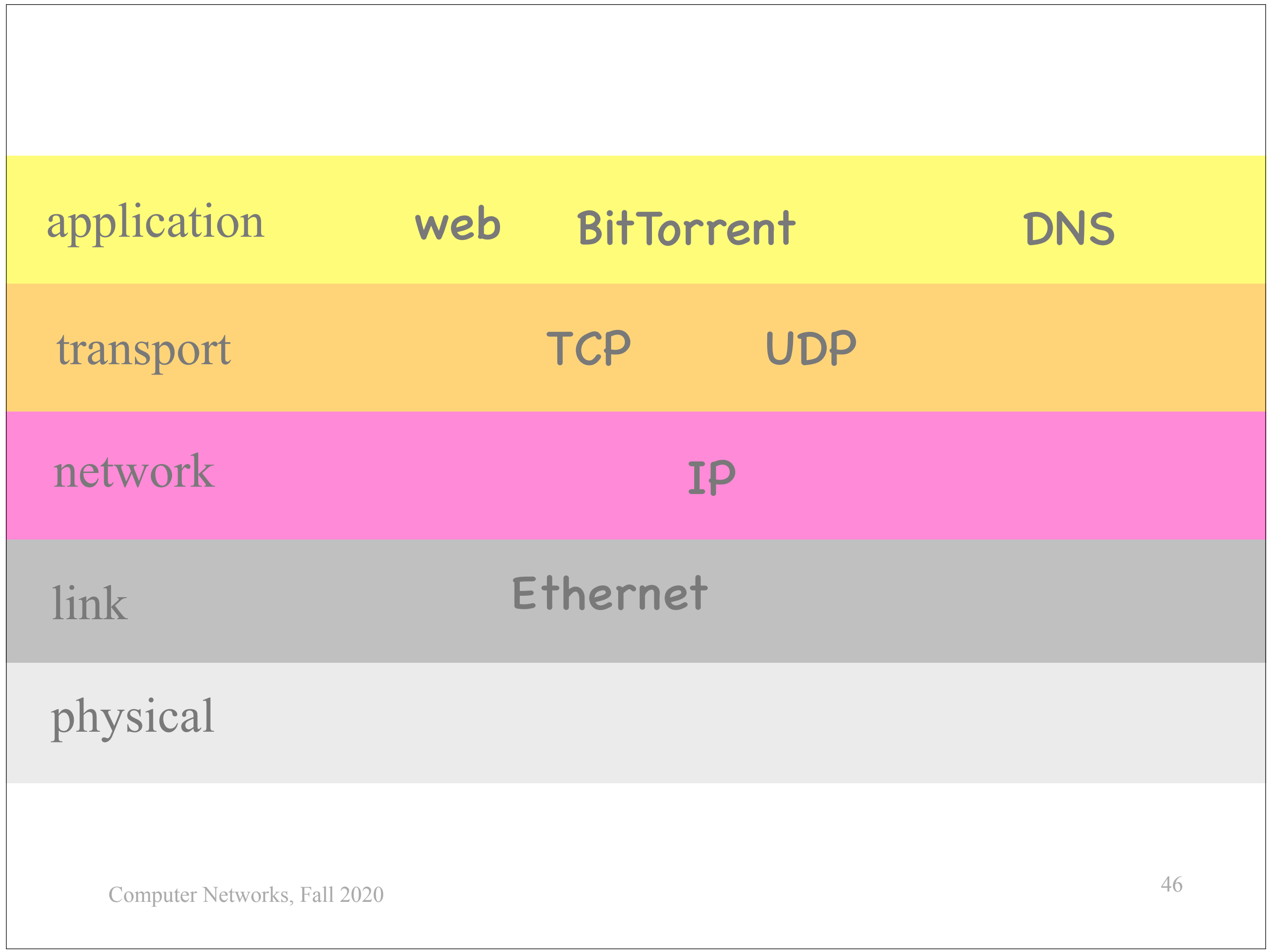
At the link layer, we have the technologies mentioned earlier,
while at the physical layer, we have the physical media that these technologies use.

At the application layer, we have, of course, all the applications that use the Internet for their communications.

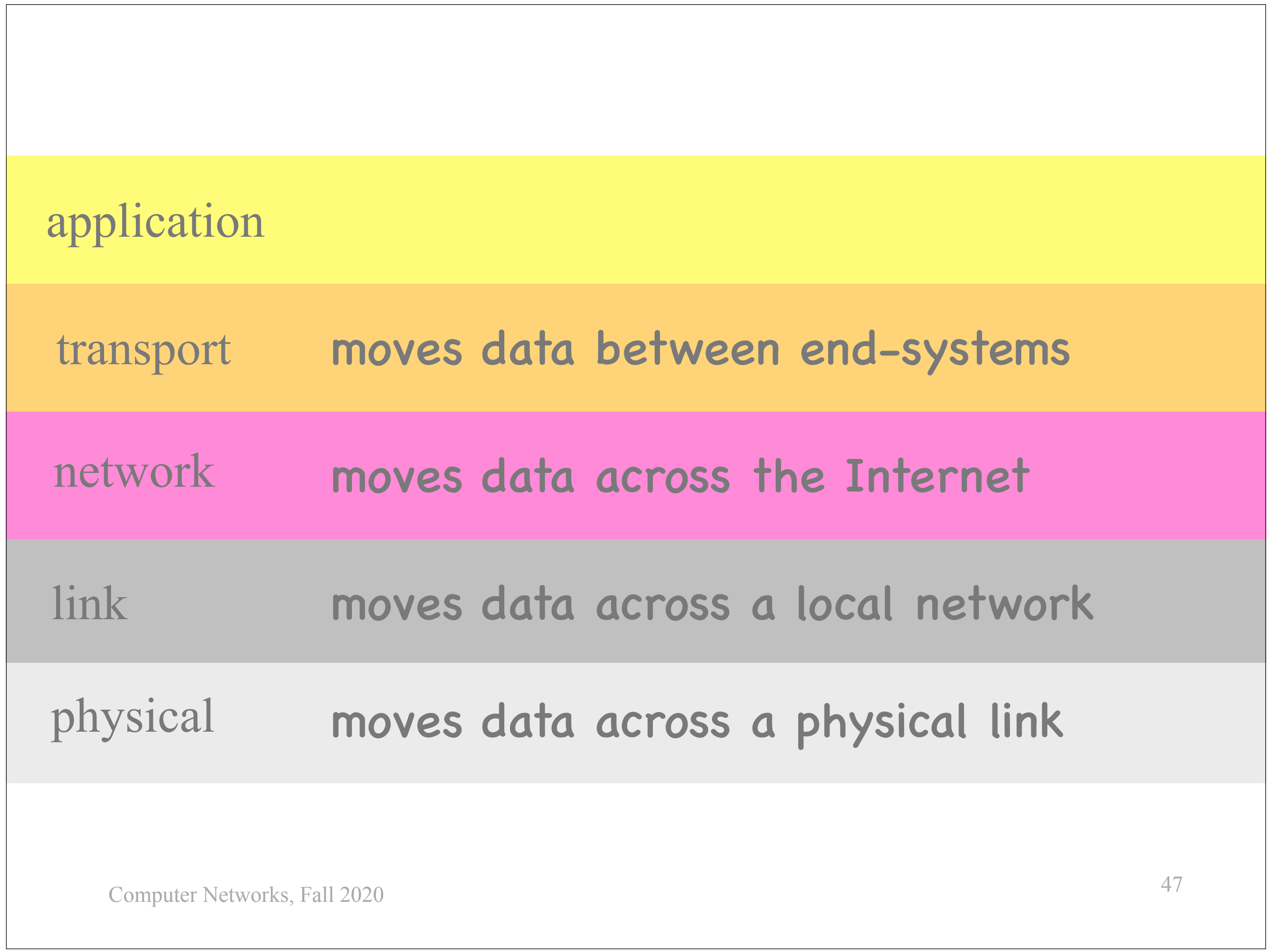
At the transport layer, we have two different technologies: TCP and UDP.

And at the network layer, we have a single technology: IP, which stands for “Internet Protocol.”

In this semester, we will focus on a

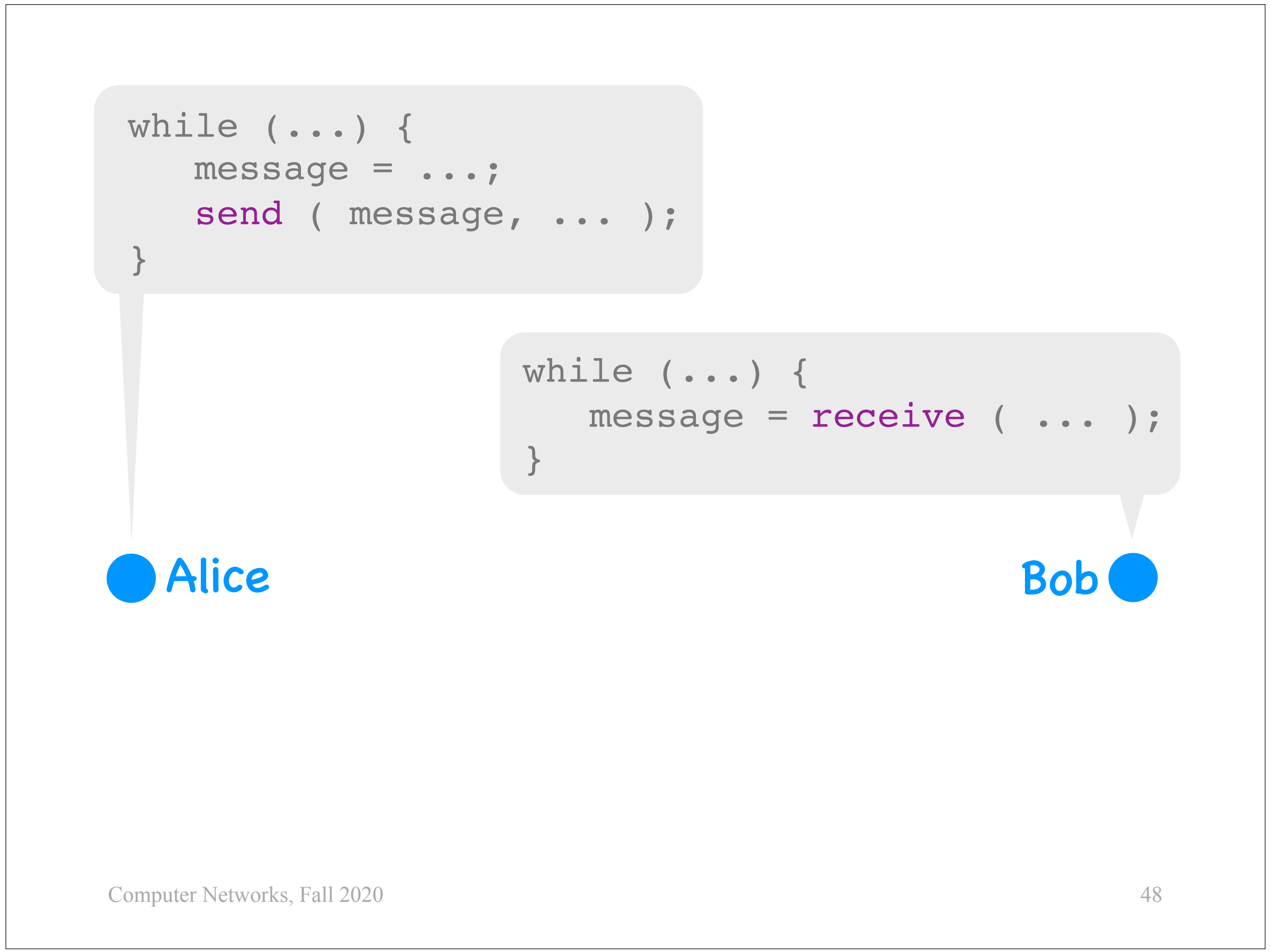


subset of these technologies and study how they work in detail.



application	
transport	moves data between end-systems
network	moves data across the Internet
link	moves data across a local network
physical	moves data across a physical link

Other than the application layer, it’s hard to describe what the other layers exactly do without looking at the details, but here’s a quick summary: …



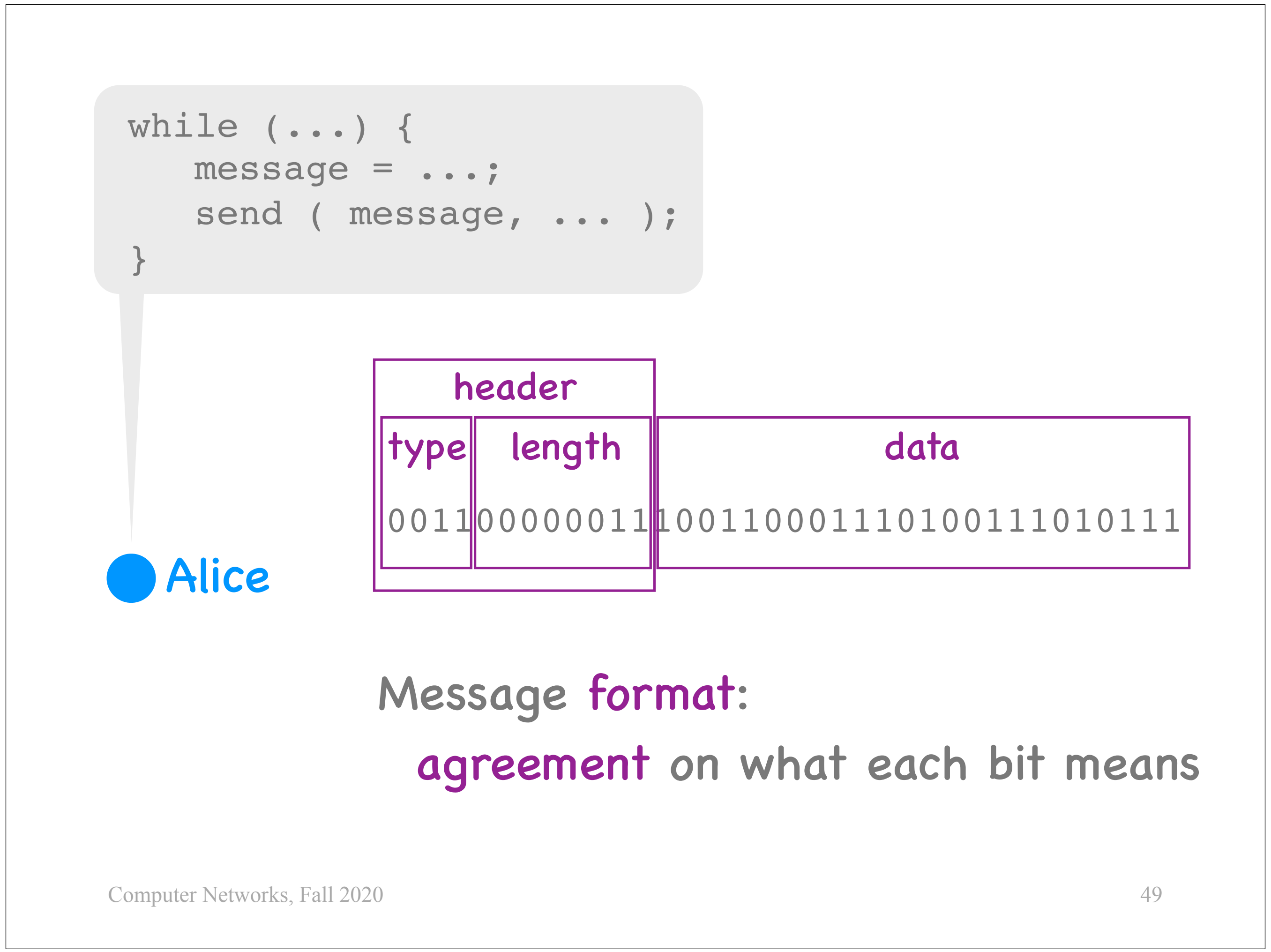
```
while (...) {
   message = ...;
   send ( message, ... );
}
```

```
while (...) {
   message = receive ( ... );
}
```

Now we will dive in to make the concept of layers a bit more concrete.

We talked earlier about a distributed application where Alice and Bob exchange messages, which are part of a communication protocol between Alice and Bob.

Let’s



```
while (...) {
   message = ...;
   send ( message, ... );
}
```

Message format:
agreement on what each bit means

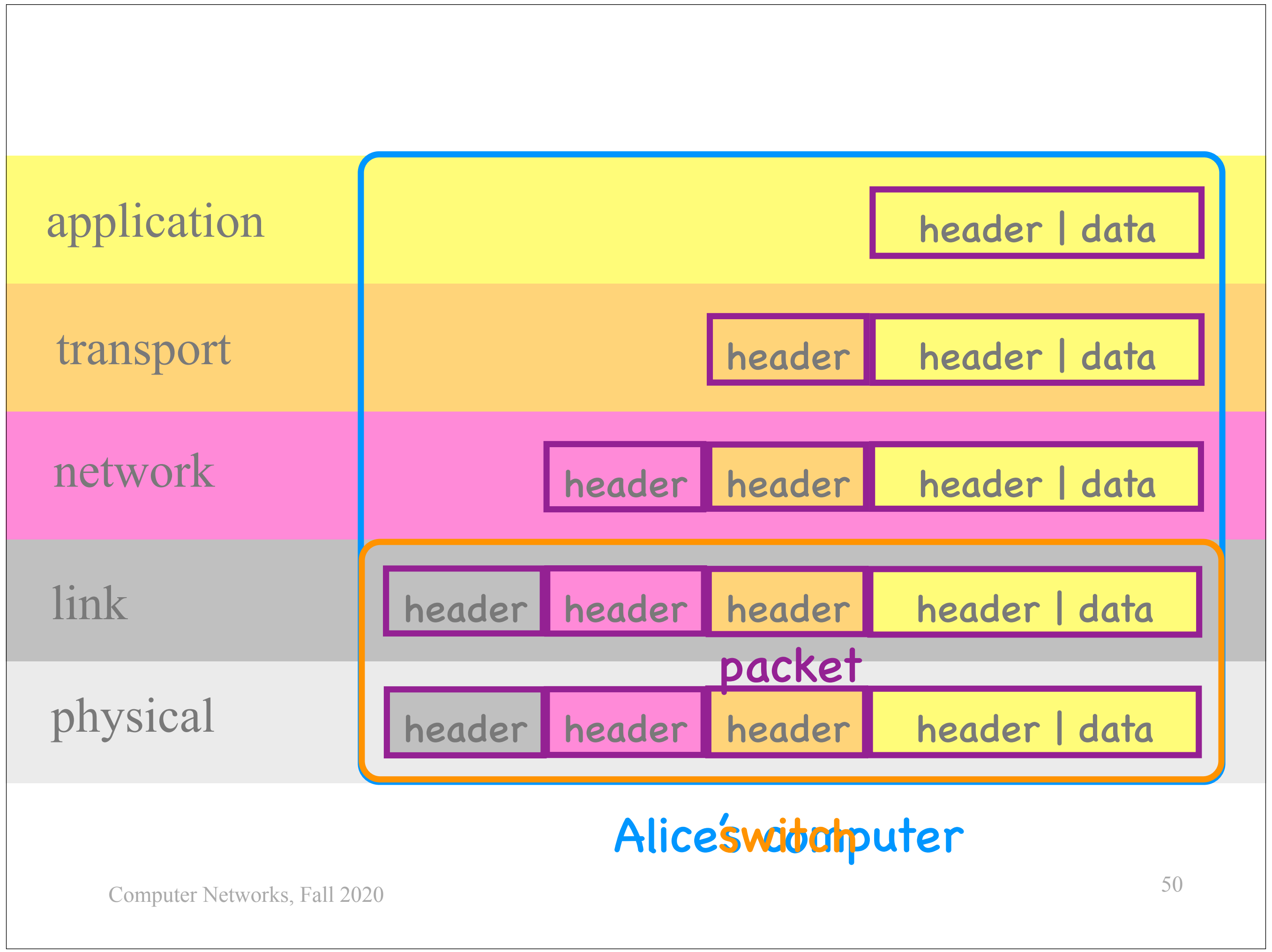
focus on Alice for a moment.

Suppose Alice sends this message to Bob. A message is, essentially, a sequence of bits. Alice and Bob must have an understanding of the semantics of these bits, what each bit means.

For example, it could be that the first 4 bits of each message indicate the type of the message (e.g., is it a hello message or a file)? If the type indicates that this message carries a file, then the next 8 bits could indicate the length of the file, while the remaining bits could be the content of the file, the actual data.

So, a message typically consists of two parts:
- meta-data fields like type and length, which we call a “header”
- and the actual data.

We refer to this structure of a message as its “format.” So, in our example, the format of a file message sent by Alice to Bob consists of a 4-bit type field, an 8-bit length field, and a data field. The type and length fields together constitute the message’s header.



I am showing you here the 5 layers of the Internet architecture,
and Alice's computer, which has functionality that belongs to all these layers.

When Alice runs an application that creates a message to send to Bob,
that happens at the application layer.

When Alice presses "send," her program passes the message down to the transport layer, which is part of the OS running on Alice's computer.

The transport layer prepends a new header to the message — a "transport-layer header" —
and passes the message down to the network layer,
which is also part of the OS running on Alice's computer.

The network layer prepends a new header to the message — a "network-layer header" —
and passes the message down to the link layer,
which is part of the hardware of Alice's computer.

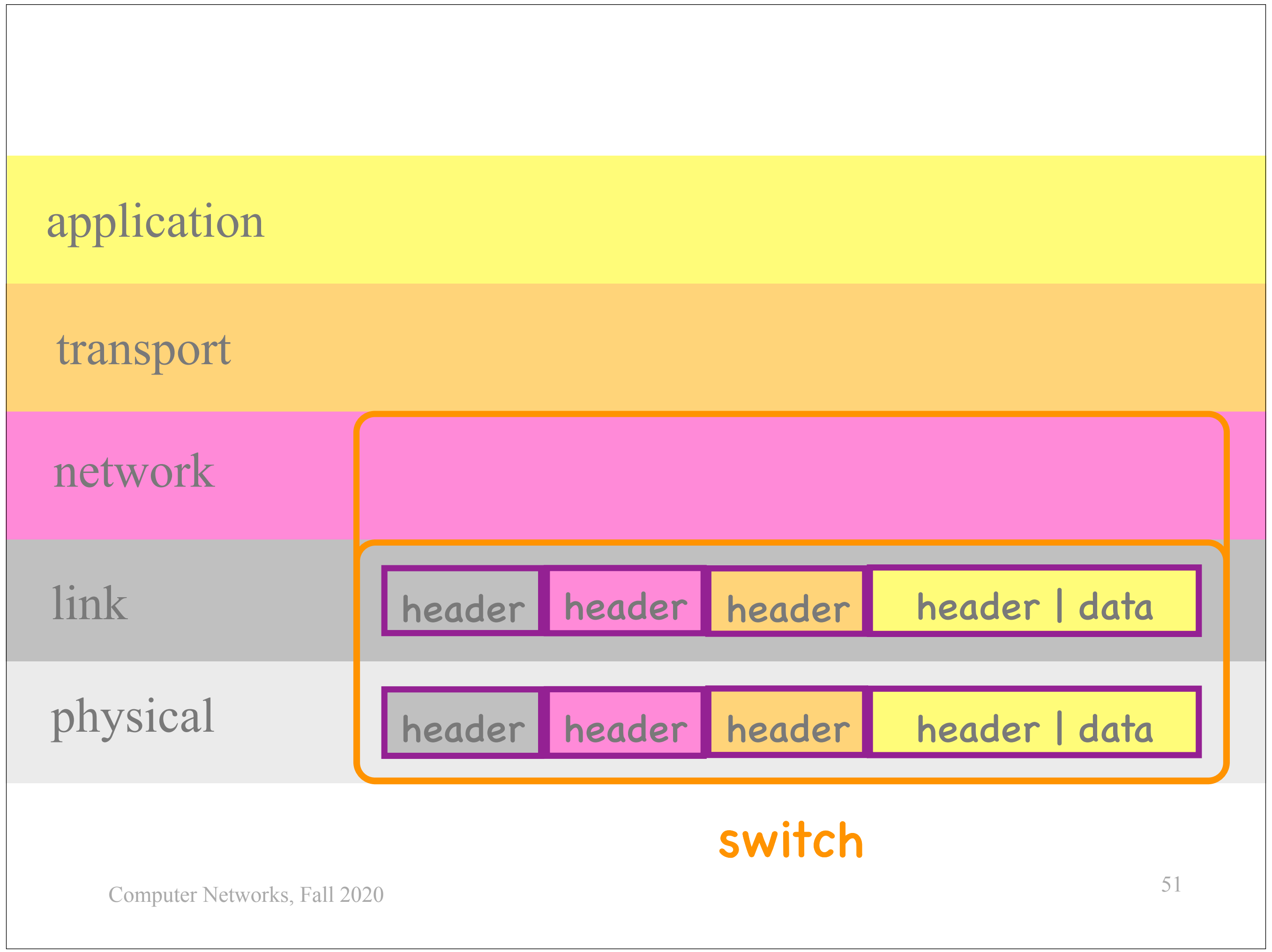
The link layer prepends a new header to the message — a "link-layer header —
and passes the message down to the physical layer.

We call this message with all the headers prepended by all the layers a "packet." Each layer only understands its own header. For example, the link layer can make no sense of the transport-layer header, or the application-layer header and data.

At the physical layer, the packet takes the form of a sequence of electromagnetic waves and travels from Alice's computer to the first packet switch.

There are two kinds of switches on the Internet: link-layer switches and network-layer switches (also called “routers”).

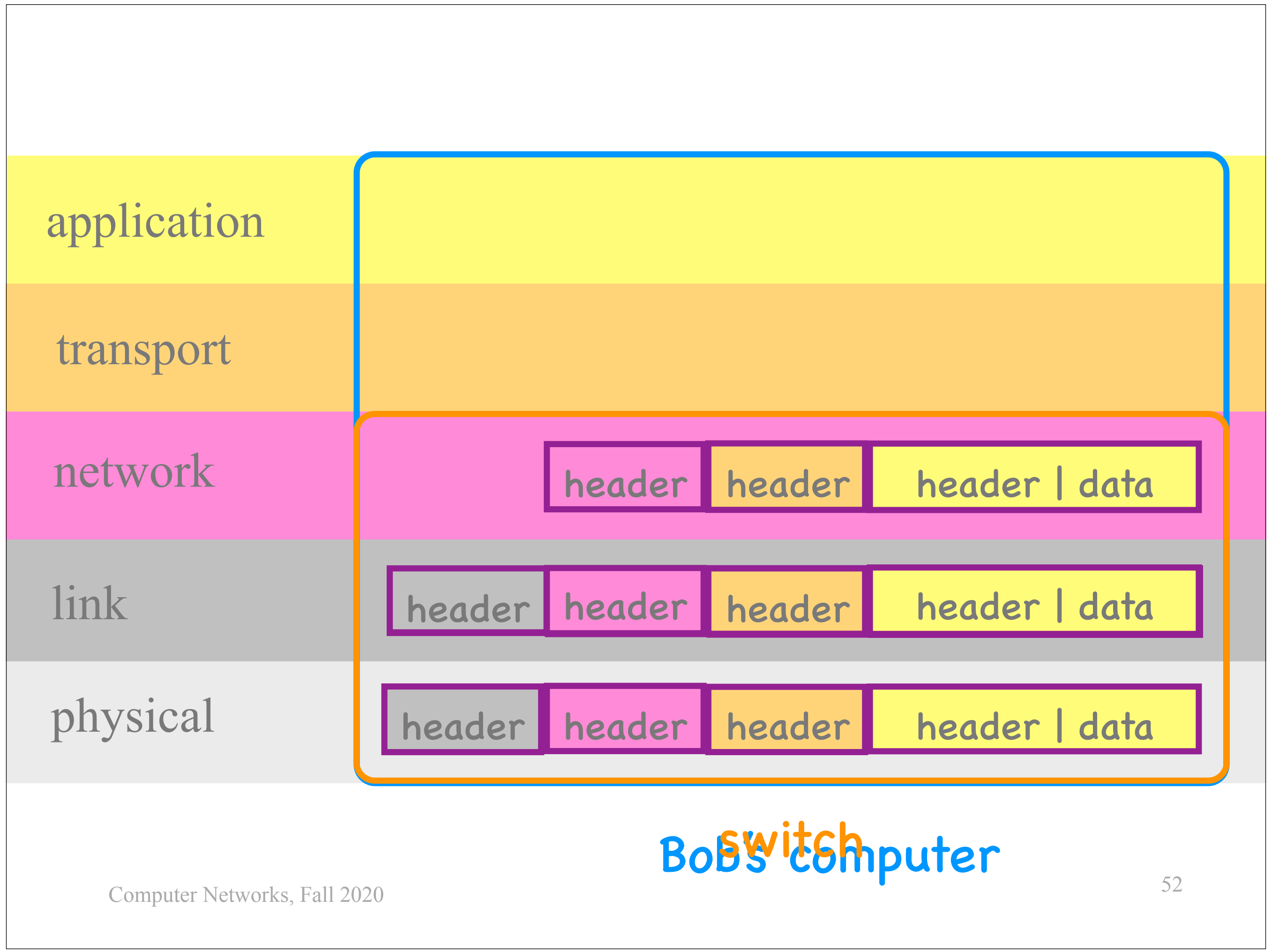
Let’s say that this switch is a link-layer switch.
When the packet arrives, the physical layer passes it



up to the link layer of the switch,
which processes the link-layer header,
then passes the packet back down to the physical layer,
which sends it on to the next switch.

Now let's say this switch is a network-layer switch.
Here's what happens:

When the packet arrives, the physical layer passes it



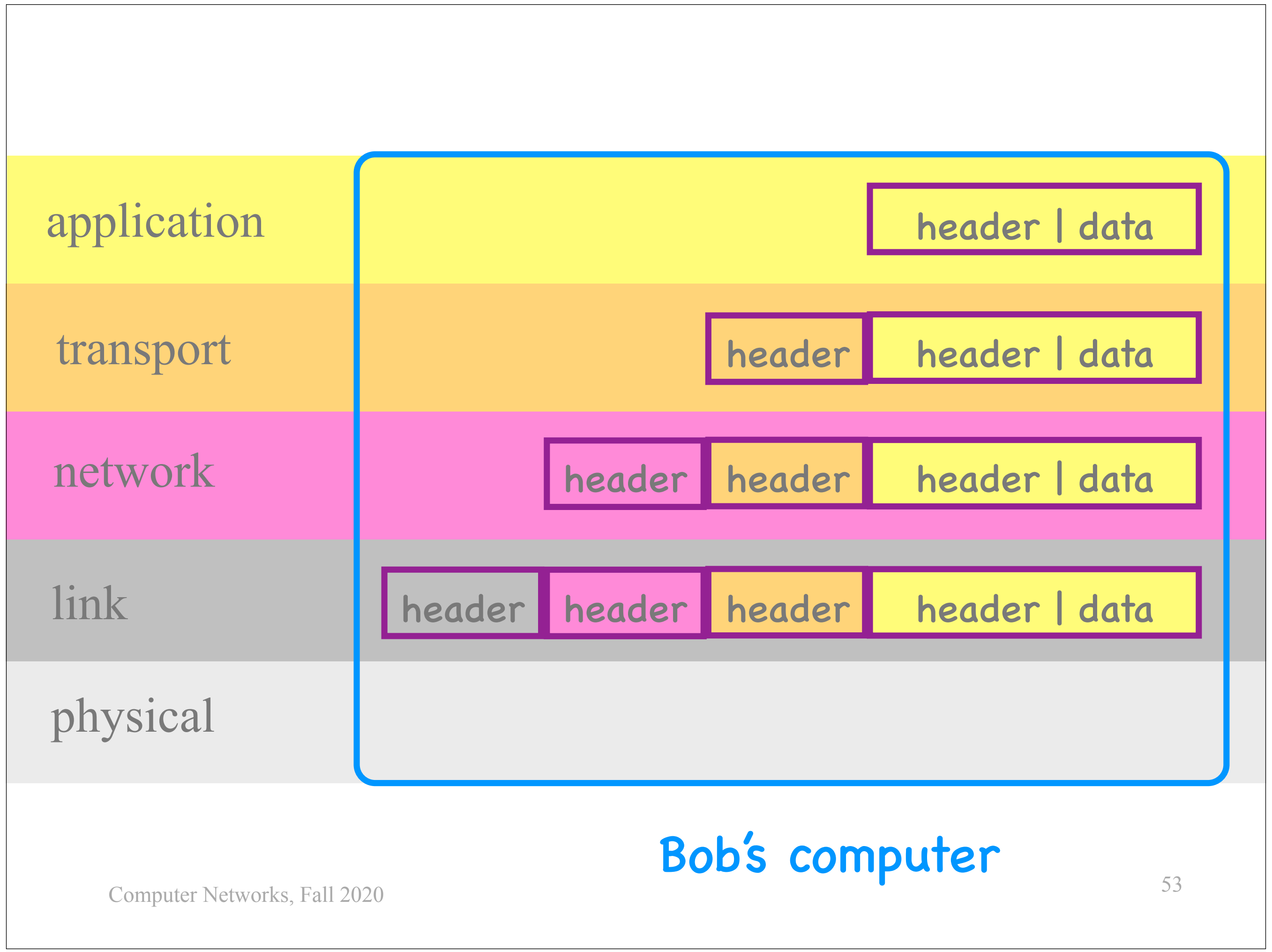
up to the link layer of the switch,
which processes the link-layer header,
removes the link-layer header.
and passes the packet up to the network layer of the switch.

The network layer processes the network-layer header,
then passes the packet back down to the link layer.

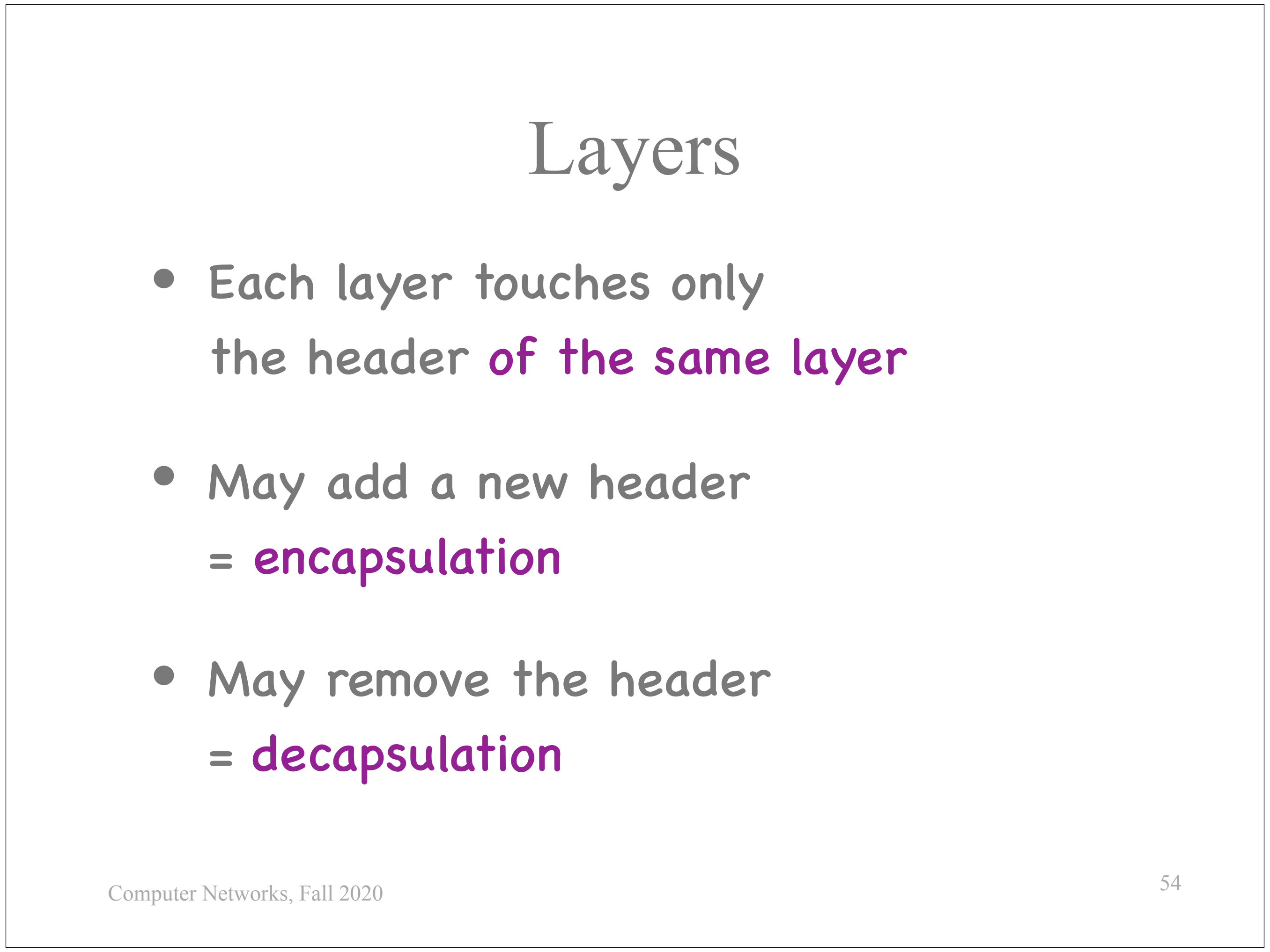
The link layer then prepends a new link-layer header,
then passes the packet back down to the physical layer,
which sends it on.

This may be repeated several times,
until the packet reaches Bob's computer.

There, the physical layer passes the packet



up to the link layer,
which removes the link-layer header and passes the packet up to the network layer,
which removes the network-layer header and passes the packet up to the transport layer,
which removes the transport-layer header and passes the message up to Bob's program,
and then Bob can finally read it.



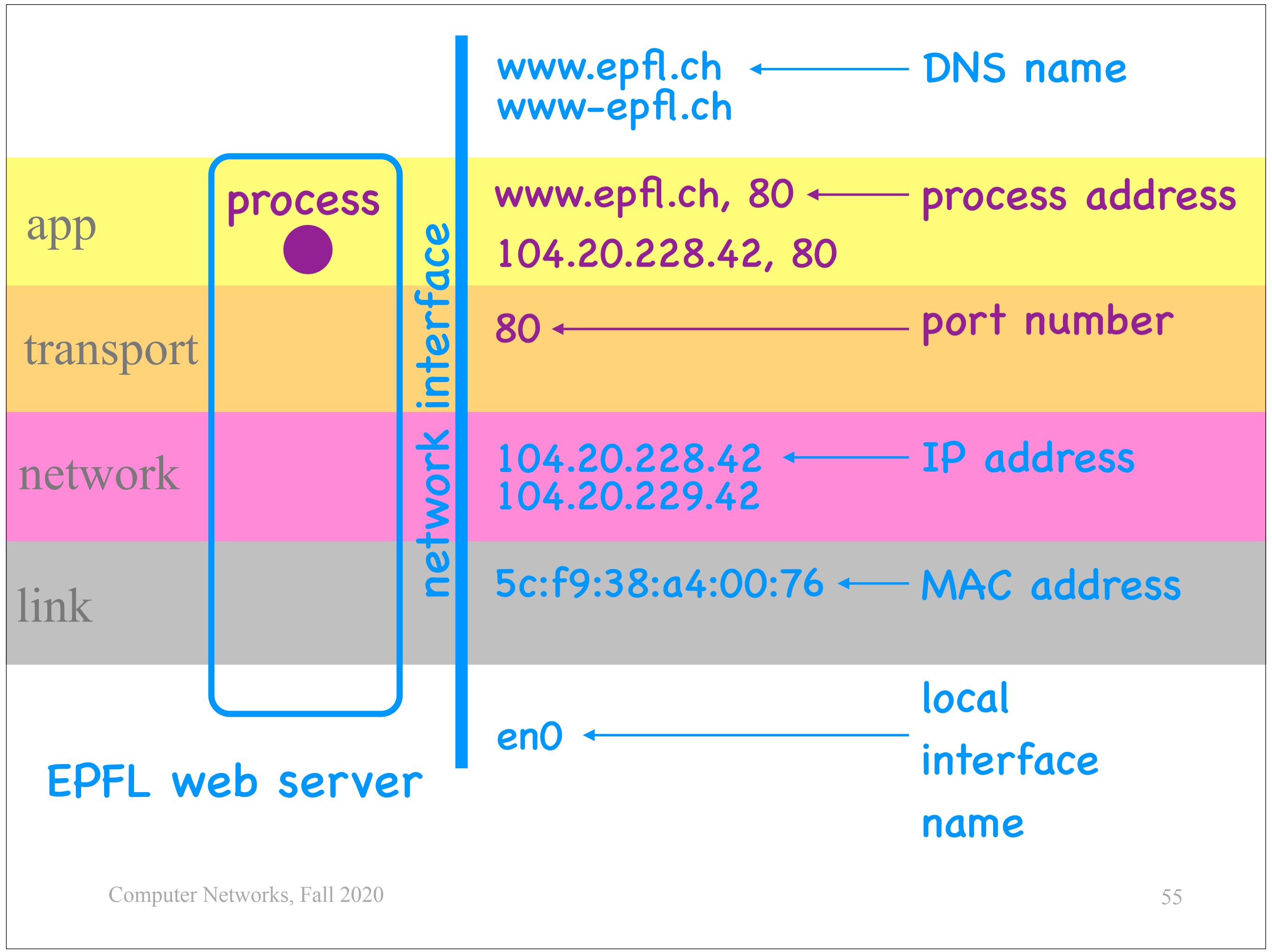
Layers

- Each layer touches only the header of the same layer
- May add a new header = encapsulation
- May remove the header = decapsulation

So:

The principle is that each layer touches only the header of that same layer:
- It may add a new header (and this is a process called *encapsulation*).
- It may remove the current header (and this is a process called *decapsulation*).
- Or it may simply read the header that was added at the same layer running at a different computer or switch.

One of the hardest thing about layers is that each layer has its own vocabulary,
and that can introduce tons of confusion.



For instance, suppose this is an EPFL web server.
So, it is one of the computers that may respond if you type in your browser www.epfl.ch.

Every computer has a "network interface."
This is not like the interface between two layers, it is rather the interface between the computer and the rest of the world.

Whenever an entity wants to communicate with this EPFL web server,
it must *name* this network interface.

Different entities use different names to name this network interface:
- A human uses a "DNS name," which looks like this.
  In fact, the same network interface can have multiple DNS names or "aliases."
- The network layer of another computer or a packet switch uses an "IP address," which looks like this.
  In fact, the same network interface can have multiple IP addresses.
- The link layer of another computer or a packet switch uses a "MAC address," which looks like this.
- The OS running on the EPFL web server itself uses a local network-interface name, which looks like this.

All these names refer to the same thing: the network interface of this EPFL web server.

Now consider the web process running at the application layer of this EPFL web server.
So, this is the program that processes web requests and constructs the corresponding responses.

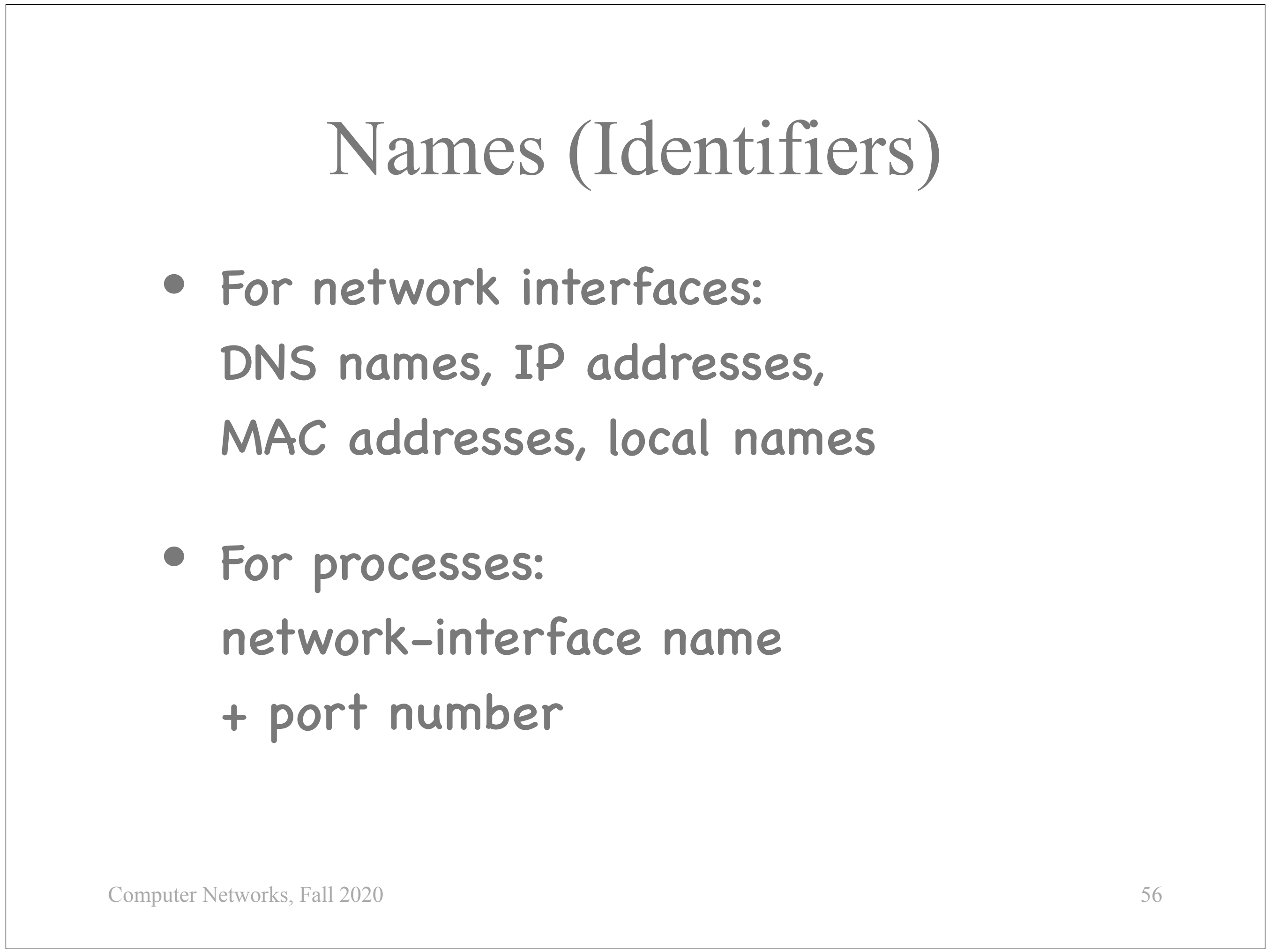
The transport layer of the EPFL web server itself uses a local port number to name this process, which looks like this (it's just an integer).

The application layer of another computer that wants to communicate with this web process,
uses a “process name” that looks like this:

- The first part is a DNS name that names the target network interface.
- The second part is a port number that names the specific process running in the application layer of the computer that owns the target network interface.

Alternatively, a process name may look like this:

- The first part is an IP address naming the target network interface.
- The second part is again a port number naming the target process.



Names (Identifiers)

- For network interfaces:
  DNS names, IP addresses,
  MAC addresses, local names
- For processes:
  network-interface name
  + port number

So: Lots of names, or identifiers.

Network interfaces can be named with DNS names, IP addresses, MAC addresses, or local OS names.

Processes are names as tuples: first a name that specifies the target network interface, then a port number that specifies the target process.

In the first lab session, which is on Wednesday, you will play around with all these names.

Translate DNS names to IP addresses:

```
> host www.epfl.ch

  www.epfl.ch is an alias for
  www.epfl.ch.cdn.cloudflare.net

  www.epfl.ch.cdn.cloudflare.net
  has address 104.20.229.42

  www.epfl.ch.cdn.cloudflare.net
  has address 104.20.228.42
```

DNS names

IP addresses

Find out the MAC and IP address of the network interface(s) on your computer:

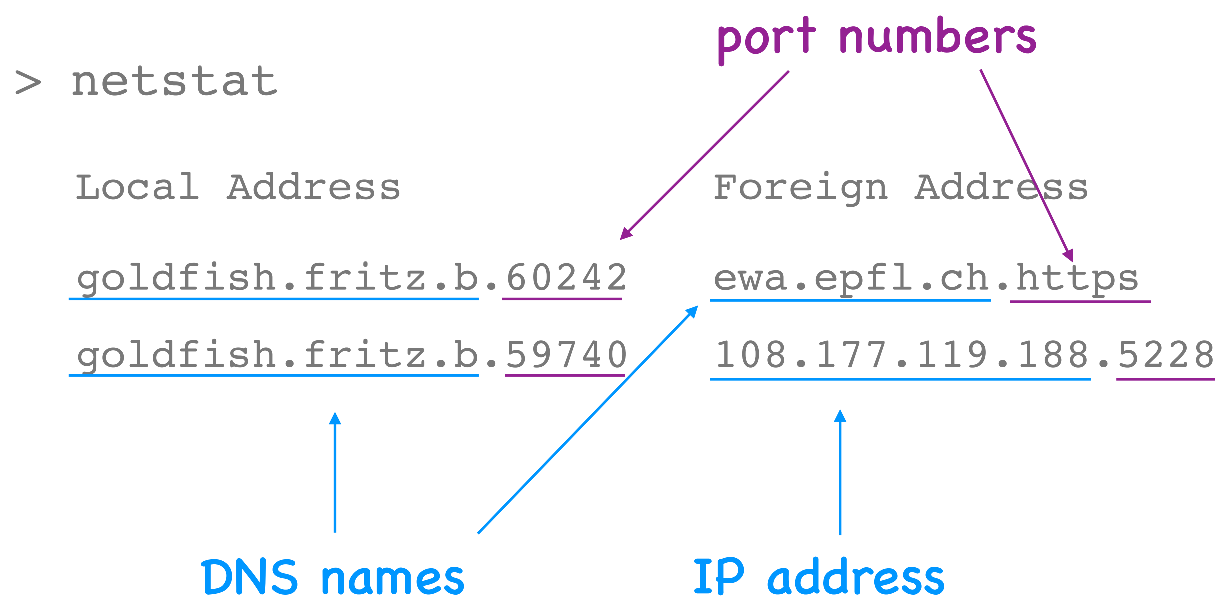
```
> ifconfig
```

MAC address

```
  en0:
        ether 5c:f9:38:a4:00:76
        inet 128.168.178.27
```

IP address

Check whom you are communicating with:



```
> netstat

    Local Address                 Foreign Address

    goldfish.fritz.b.60242       ewa.epfl.ch.https
    goldfish.fritz.b.59740       108.177.119.188.5228
```

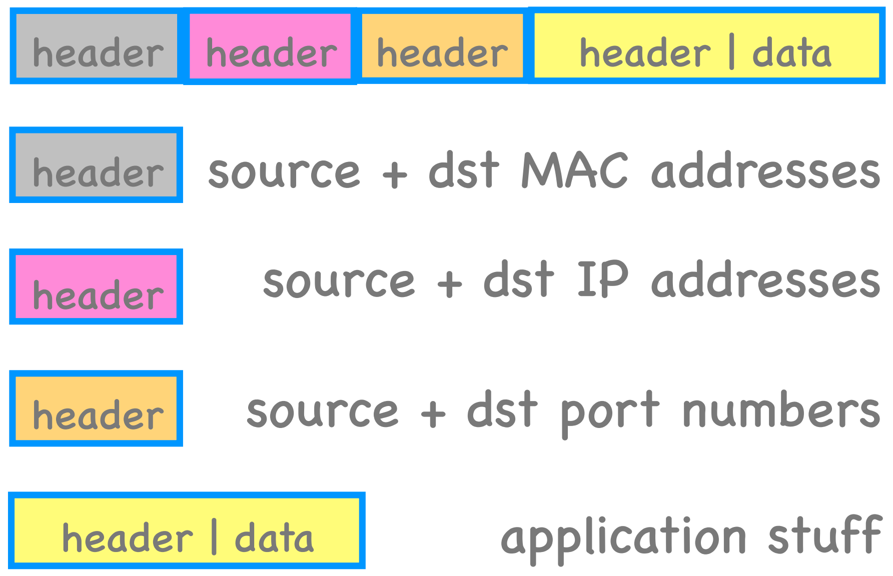
Discover packet switches between your computer and a remote one:

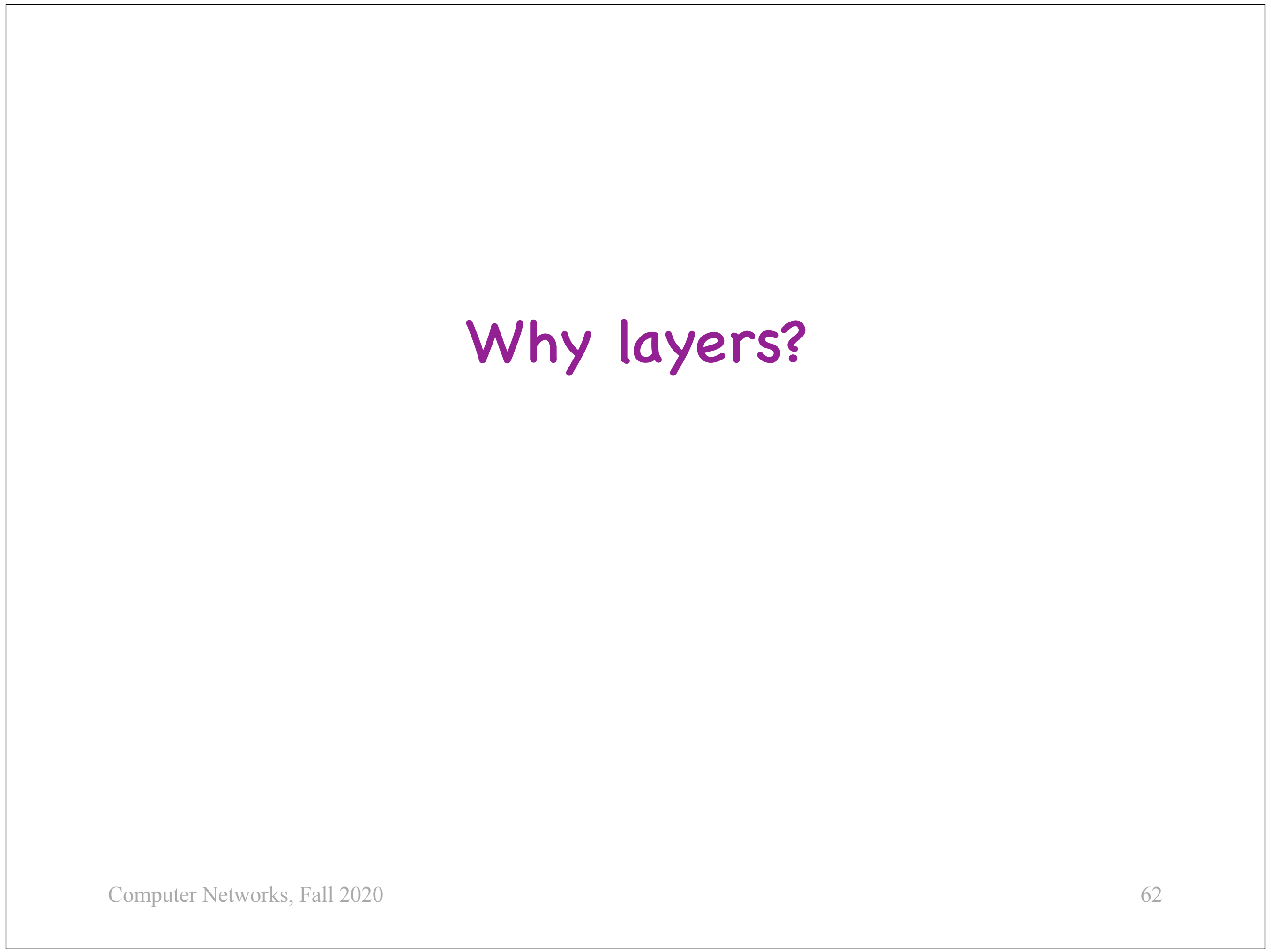
```
> traceroute www.epfl.ch

  1 fritz.box (192.168.178.1) …
  2 10.136.43.239 (10.136.43.239) …
  3 as13335.swissix.ch (91.206.52.192) …
```

DNS names

IP addresses

Look inside network packets:

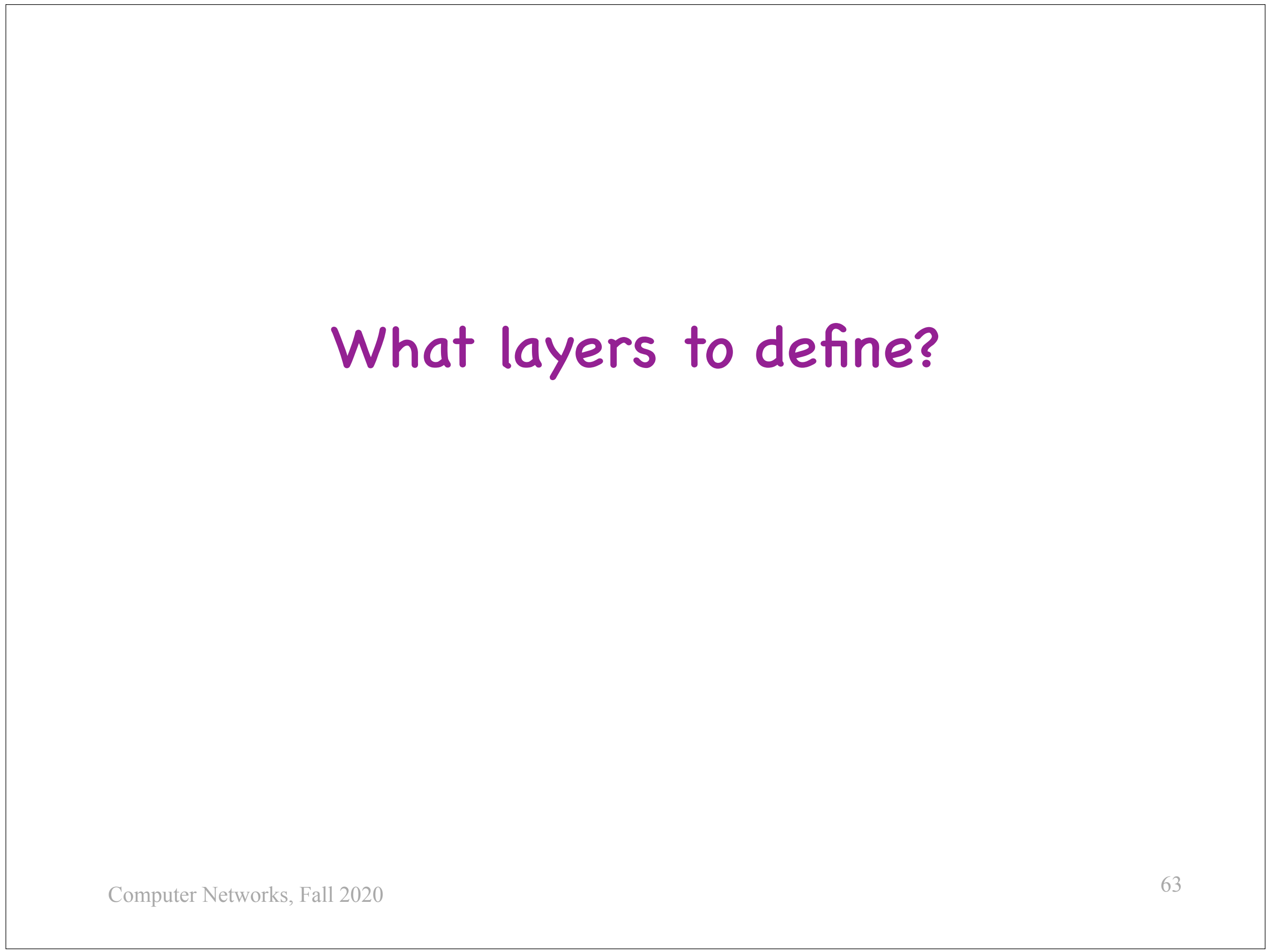




Why layers?

Why does the Internet architecture use layers?
Does layering improve performance?

I will not give the answer here, I would like you to think about it, and we will discuss it on Friday.



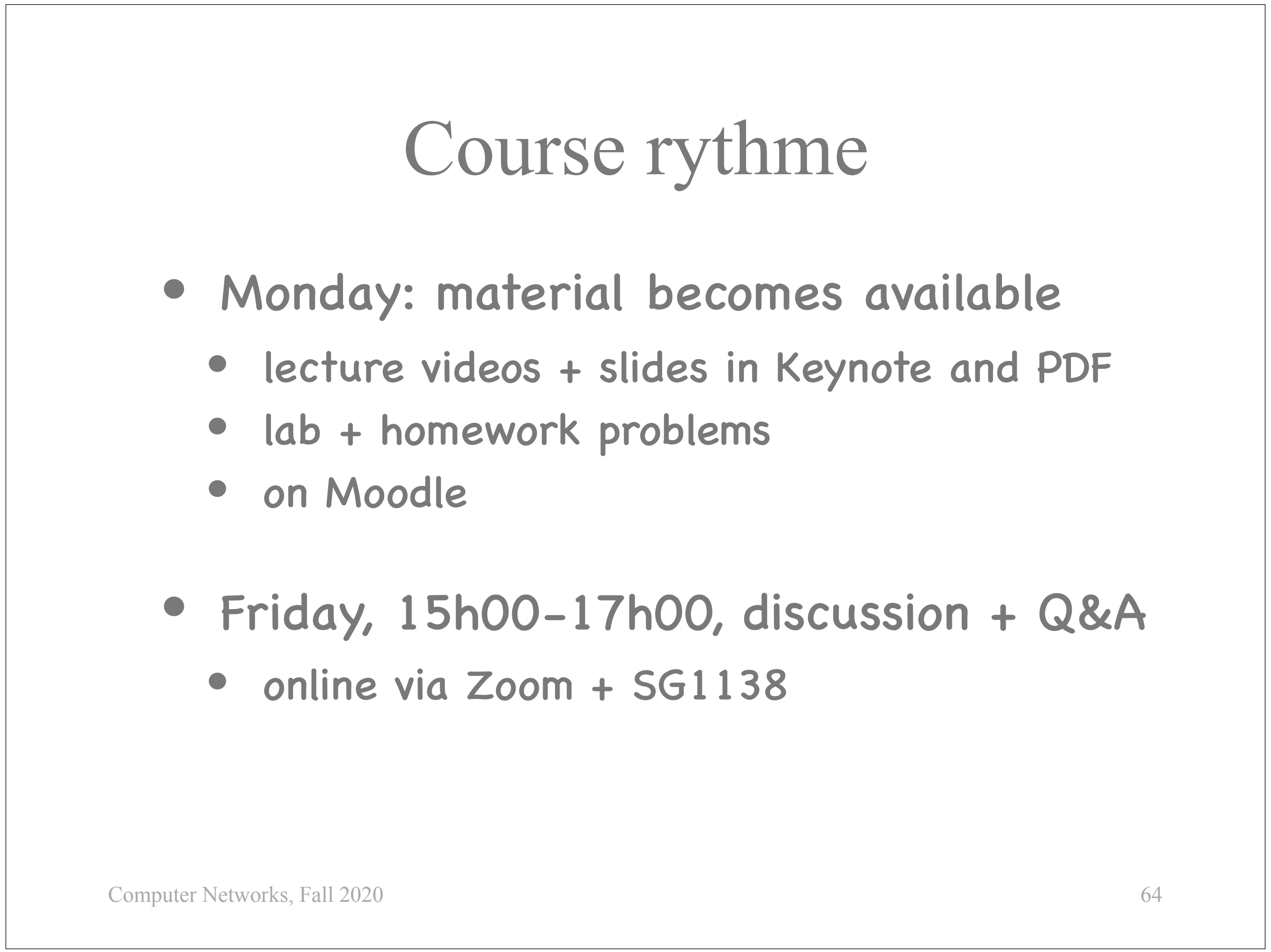
What layers to define?

One of the most fundamental questions to ask when designing a system is what layers to define.

Btw, this is not only a networking issue.
To those of you who like operating systems, separating the kernel from the user processes is a question of layering,
separating the libraries from the applications is a question of layering.

—>What is the interface between the kernel and user processes?
System calls.

This brings me to the end of the first lecture. The next and last video for this week will walk you through the course organization.



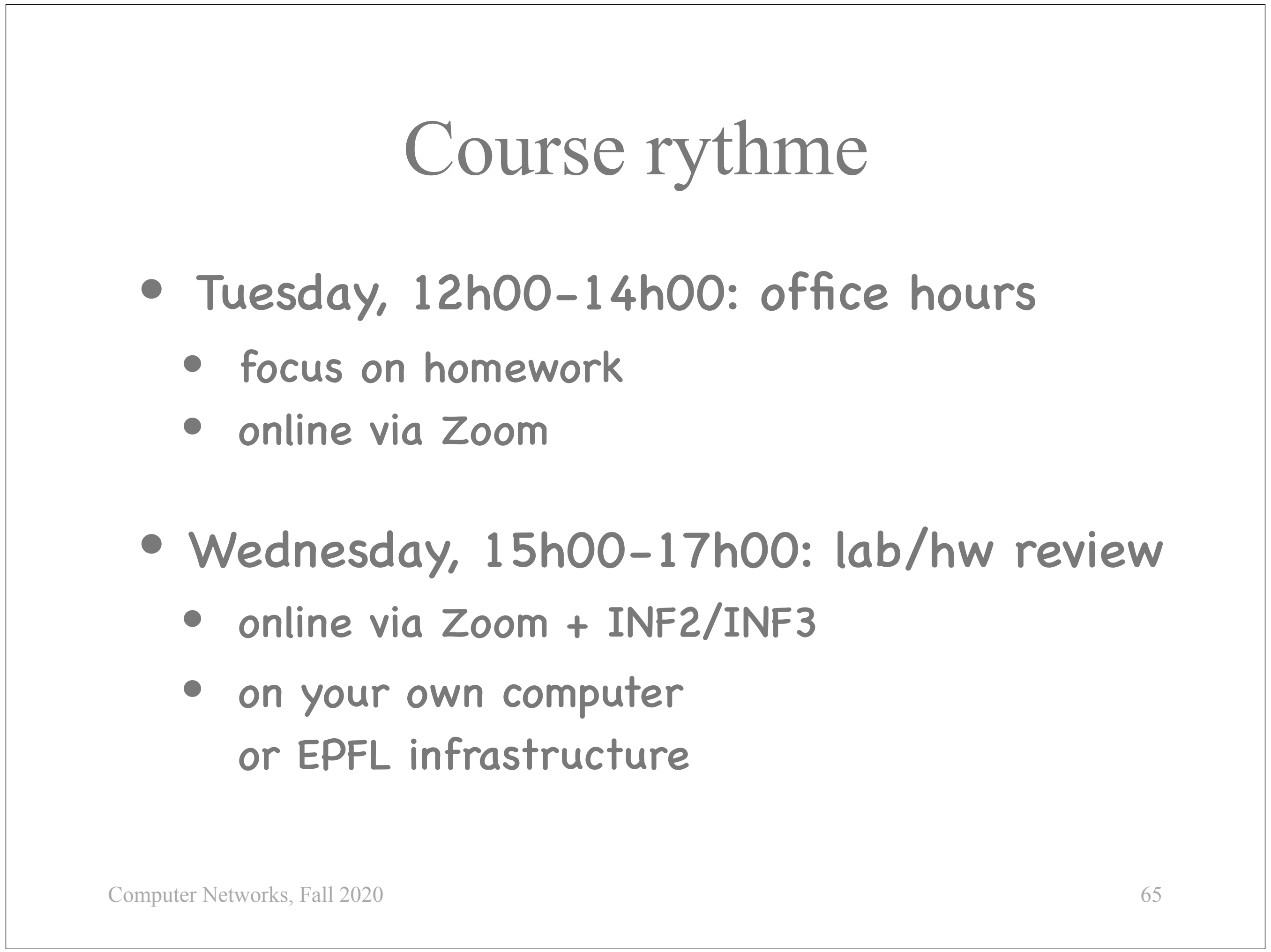
Course rythme

- Monday: material becomes available
  - lecture videos + slides in Keynote and PDF
  - lab + homework problems
  - on Moodle
- Friday, 15h00-17h00, discussion + Q&A
  - online via Zoom + SG1138

Here is how the course will work:

Every Monday, material for a new module will become available on Moodle: …

The following Friday, 15h00 to 17h00, I will be available to discuss the lecture videos and answer any questions you may have. You can participate either online via Zoom, or by coming in person to SG1138.



Course rythme

- Tuesday, 12h00-14h00: office hours
  - focus on homework
  - online via Zoom
- Wednesday, 15h00-17h00: lab/hw review
  - online via Zoom + INF2/INF3
  - on your own computer
    or EPFL infrastructure

The next Tuesday, there will be office hours to discuss any extra questions you may have, especially on the homework. This will take place online via Zoom.

The next Wednesday, 15h00 to 17h00, there will be either a lab or homework review session. You can solve the labs during the corresponding lab session, while having real-time support by the teaching team, or, even better, you can solve the labs before the corresponding lab session, and use the session to ask questions and clarifications. You can participate either online via Zoom or by coming in person to INF2/INF3. Either way, you will have the option to do the labs on EPFL infrastructure or on your own computer.

Quizzes

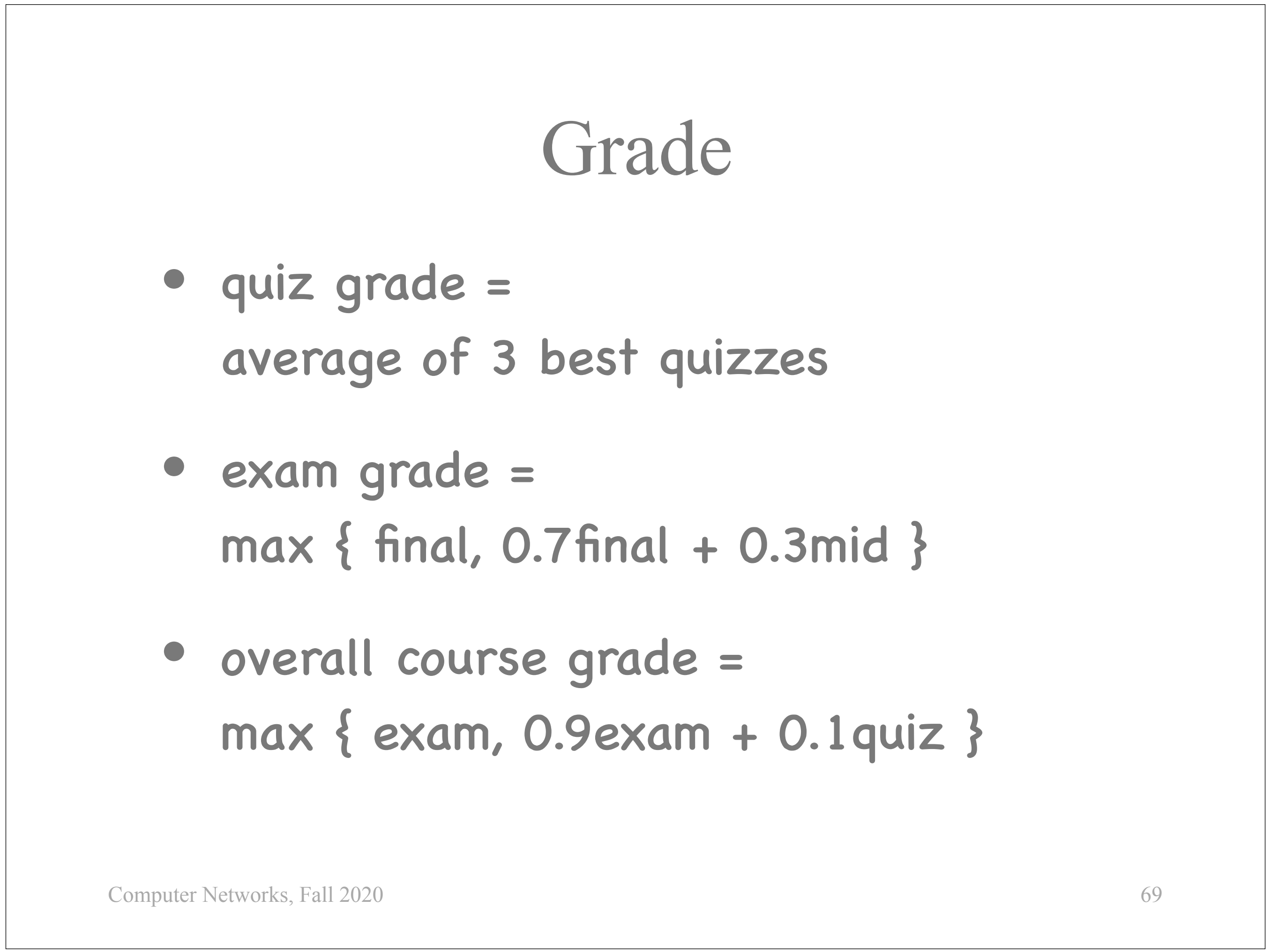
- Voluntary, count only if you do well
- 10min, every other week
- Cover the 2 previous lectures
- Online via Moodle

Midterm

- Voluntary, counts only if you do well
- Exact format TBD
- Covers all lectures, homework, and labs prior to midterm date
- Online via Moodle

Final

- Mandatory, counts always
- 2 hours, closed-book, date + room TBD
- Covers all lectures, homework, and labs
- During winter exam session



Grade

- quiz grade =
  average of 3 best quizzes
- exam grade =
  max { final, 0.7final + 0.3mid }
- overall course grade =
  max { exam, 0.9exam + 0.1quiz }

The rationale is the following:

The midterm covers less material, so it can be easier to get a good grade in it.

However, not all students have gotten the hang of the course by midterm time. Some students take longer to adjust, and they may do much better in the final than in the midterm.

So, the midterm counts only if it helps, same with the quizzes.

Information

- Moodle page
  - lecture videos & slides
  - lab & homework problems, solutions
  - announcements on the News forum
  - Zoom links + information about where to login to do the labs remotely
- https://compnet.epfl.ch
  - names of the teaching team + the information I am giving you now

Communication

- Moodle Q&A forum
  - fastest response + knowledge sharing
- com208-staff@groupes.epfl.ch
  - for questions you don't want to share

Welcome aboard!