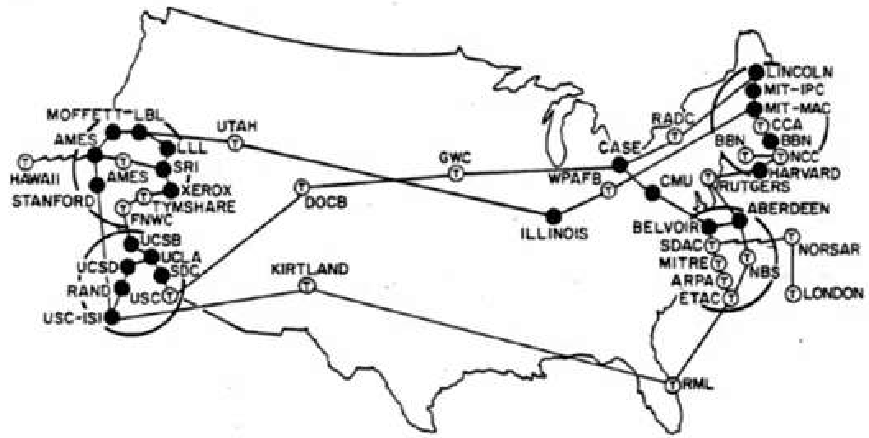
A bit of history

- Early work on the computer networks that would evolve into today's Internet began in the 1960s.
  - The US Department of Defense's **ARPA** (Advanced Research Projects Agency) funded **ARPANET**, which came online in 1969 to connect mainframes at universities and labs.
  - Originally used a protocol called NCP (Network Control Program).

- Vint Cerf and Bob Kahn (working at DARPA) began developing **TCP** (Transmission Control Program) in 1974.
  - Later divided into two protocols still used today:
    - **Transmission Control Protocol (TCP)**
    - **Internet Protocol (IP)**
- These two protocols form the foundation of the protocol suite known as TCP/IP today.
  - ARPANET fully switched to TCP/IP on January 1, 1983.



"Arpanet in the 1970s" – Semaforo GMS, CC BY-SA 4.0, via Wikimedia Commons.

- TCP/IP became dominant over vendor-proprietary solutions at the time because it was published as a set of open standards that any vendor could implement, and it could run over many different types of networks.

Who defines the standards?

- Most networking standards are developed by independent standards organizations, not by a single vendor, with participation from engineers at many companies.

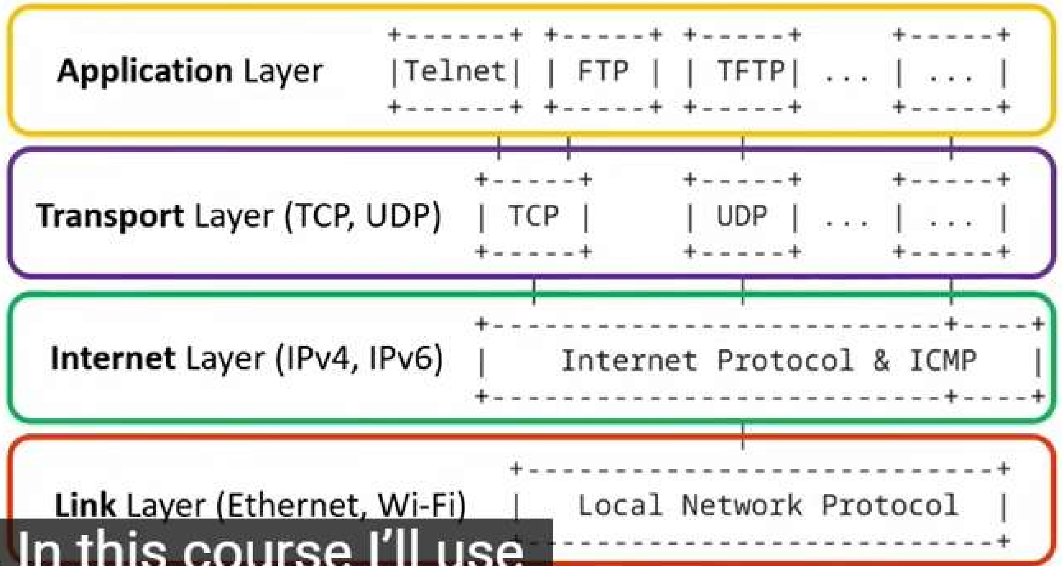
- **IEEE (Institute of Electrical and Electronics Engineers)**
  - Develops many of the technologies used on local area networks:
    - **Ethernet (802.3)**
    - **Wi-Fi (802.11)**

- **IETF (Internet Engineering Task Force)**
  - Open community that defines protocols used on the Internet:
    - TCP, IP, UDP, HTTP, DNS, etc.
  - Publishes standards in documents called **RFCs** (Requests for Comments).

Layered models

- Networks do a lot of different jobs to move data from one computer to another.
  - Physical transmission of signals, local delivery on a LAN, routing traffic between networks, end-to-end conversations, applications, etc.
- A model lets us group related jobs into layers.
  - Each layer has a specific role.
  - Each layer uses the services of the layer below and provides services to the layer above.
- Protocols live (mostly) at one layer.
  - Examples later: IP, TCP, HTTP, etc.
  - Together they form a **stack** of protocols that work as a team (the **network stack**).

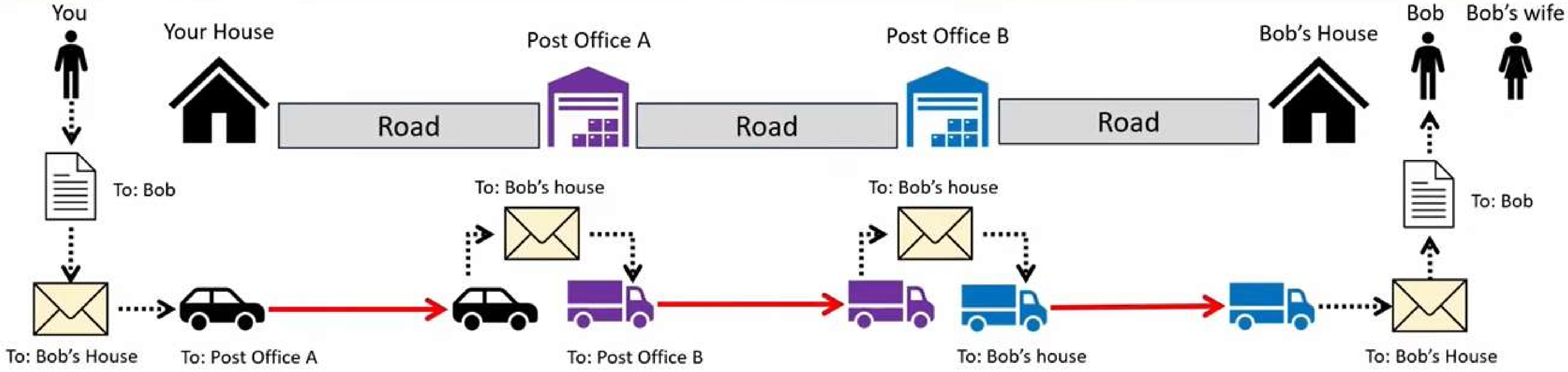
- The model is a description, not a law.
  - Different textbooks/courses use slightly different models (4-layer, 5-layer, etc.).



and some with more. In this course I'll use a five-layer model that builds on this one.

From RFC 791, "Internet Protocol" (1981)

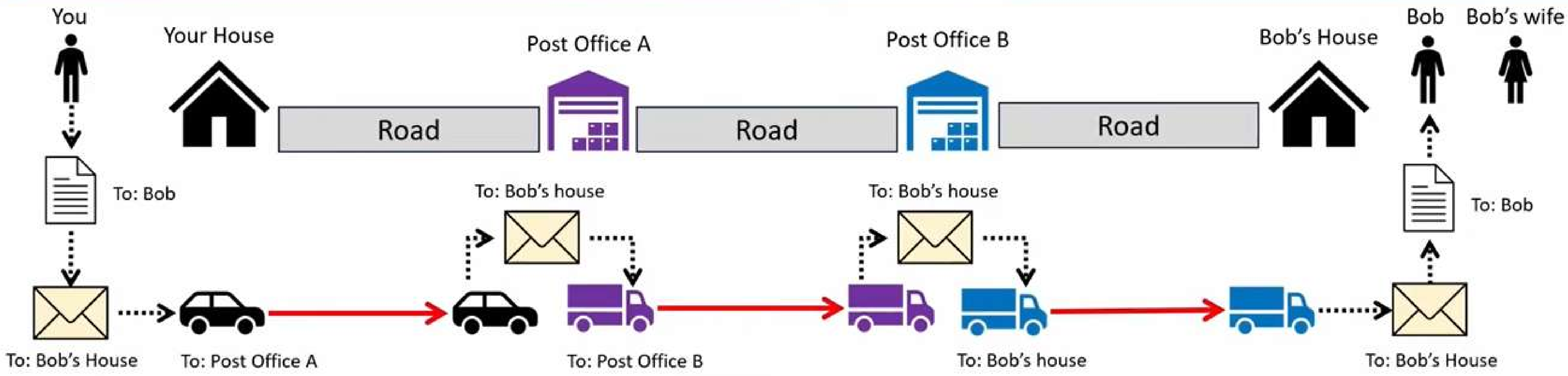
Sending a letter



- You write a letter addressed to your friend, **Bob**.
- You put the letter in an envelope addressed to **Bob's house**.
- You deliver the envelope in your car to **post office A**.
  - Post office A moves the envelope to a truck and delivers it to **post office B**.
  - Post office B moves the envelope to a new truck and delivers it to **Bob's house**.
- The letter, addressed to **Bob**, is read by Bob.

We can turn those roles into a layered model,
and then compare that model to how TCP/IP works.

Building a model



Content layer: the text of the letter, what you actually want to say.

Recipient layer: "To: Bob" vs "To: Bob's wife": the intended recipient inside the house.

Address layer: the intended destination address for the house where the recipient lives.

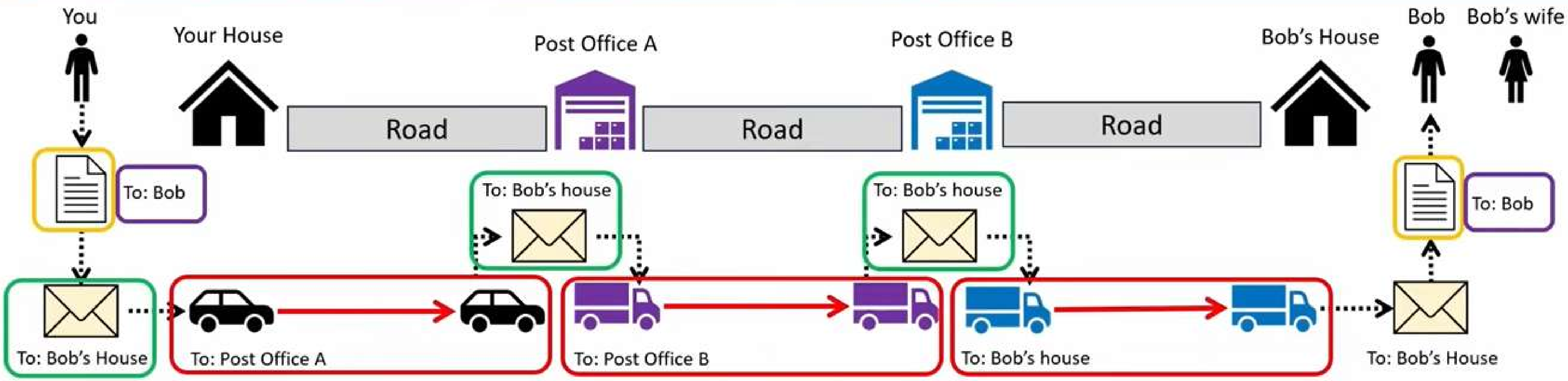
Local Delivery layer: delivery to the next stop on the path using cars/trucks: post office A, then post office B, then Bob's house.

Infrastructure layer: the roads that carry the cars and trucks.

Related layers:
- Travel by ground?
  → use a car or truck
- Travel by air?
  → use an airplane
- Travel by water?
  → use a ship

for example. You can either think of these
bottom two layers as one combined delivery layer,

Separation of layers



Content layer

Recipient layer

Address layer

Local Delivery layer

Infrastructure layer

- Each layer has its own job.
  - Layers work together to deliver the message, but each one focuses on its own task.
- What happens inside one layer doesn't change the job of the other layers.
  - Changing the content of the letter doesn't change the delivery steps.
  - Changing the delivery path doesn't affect the letter itself.

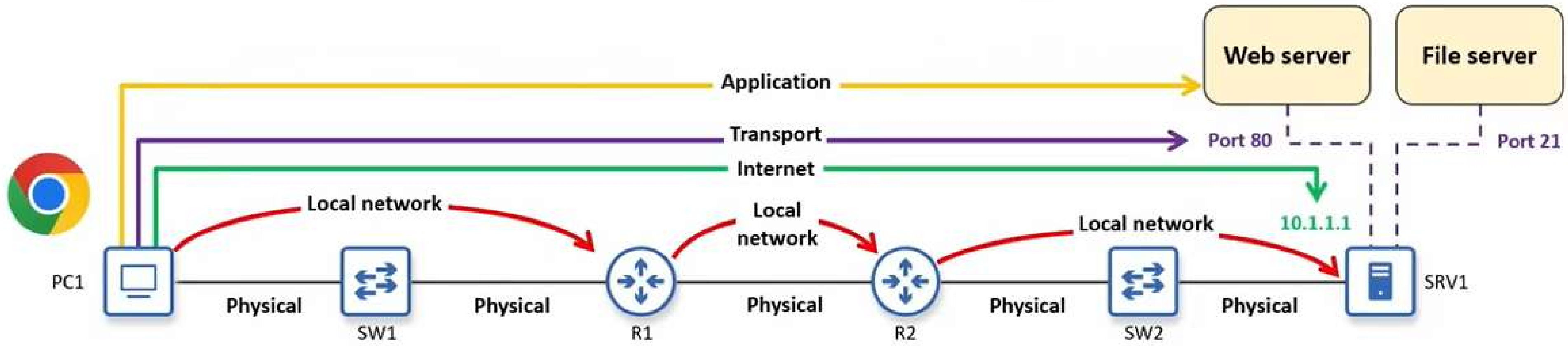
letters with data and roads with networks, and
build the 5-layer TCP/IP model we'll actually use.

The TCP/IP model

Content layer: the text of the letter, what you actually want to say.	Application layer
Recipient layer: "To: Bob" vs "To: Bob's wife": the intended recipient inside the house.	Transport layer
Address layer: the intended destination address for the house where the recipient lives.	Internet layer
Local Delivery layer: delivery to the next stop on the path using cars/trucks: post office A, then post office B, then Bob's house.	Local Network layer
Infrastructure layer: the roads that carry the cars and trucks.	Physical layer

so next we'll see how they actually work in
a real network between a client and a server.

The TCP/IP model



Application layer
- Protocols for communication between application processes; create and interpret the data.

Transport layer
- Provides end-to-end communication between application processes using **port numbers**.

Internet layer
- Provides end-to-end communication between hosts across networks using **IP addresses** and routers.

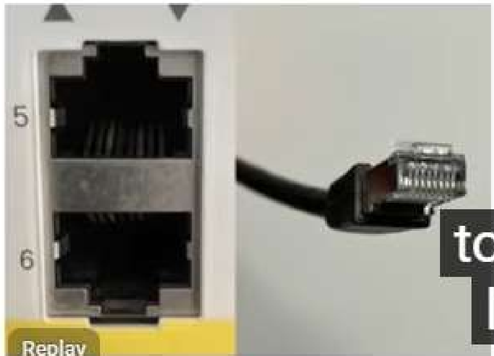
Local Network layer
- Provides hop-to-hop delivery within a local network using **MAC addresses** and switches.

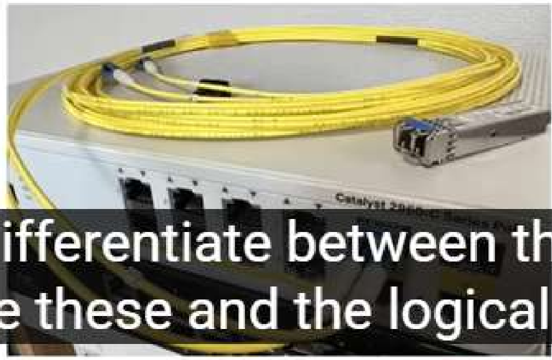
Physical layer
- optical signals over fiber-optic cables, and ium.

radio signals over wireless Wi-Fi connections.

Layer 1: The Physical layer

- The **Physical Layer** (Layer 1) sends and receives bits as electrical, optical, or radio signals over the medium.
- Defines things like cables, connectors, signal levels, and link speeds.
- Examples: copper UTP cables, fiber-optic cables, Wi-Fi radios and antennas, network interface cards (NICs).
- The physical aspects of transmitting data are very complex.
  - Network engineers typically don't have to know the low-level details.





Application layer

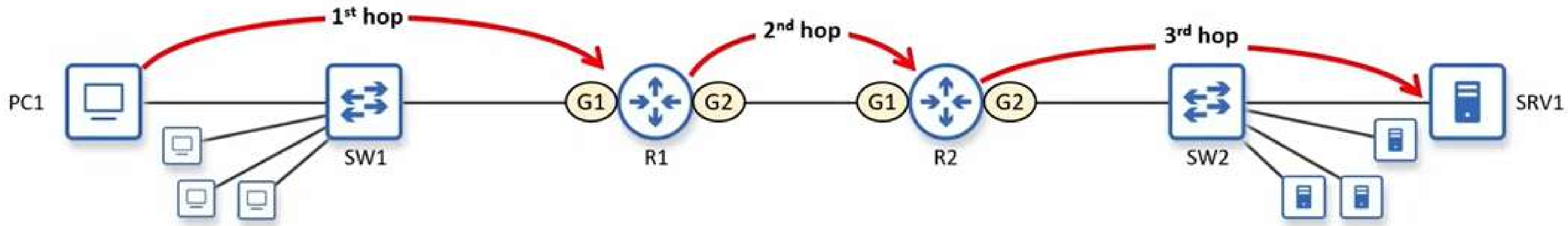
Transport layer

Internet layer

Local Network layer

Physical layer

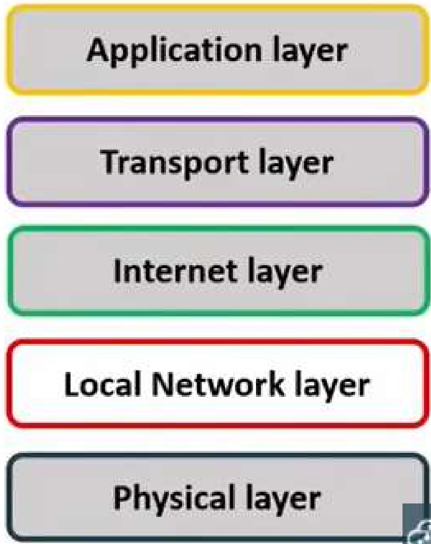
Layer 2: The Local Network layer



- The **Local Network layer** (Layer 2) provides **hop-to-hop** delivery of messages on a local network.
  - A **hop** is one step along the path between two devices:
    - from one router or host, to the next router or host in the path.
  - Switches don't count: a switch just extends the local network, allowing multiple devices to connect.

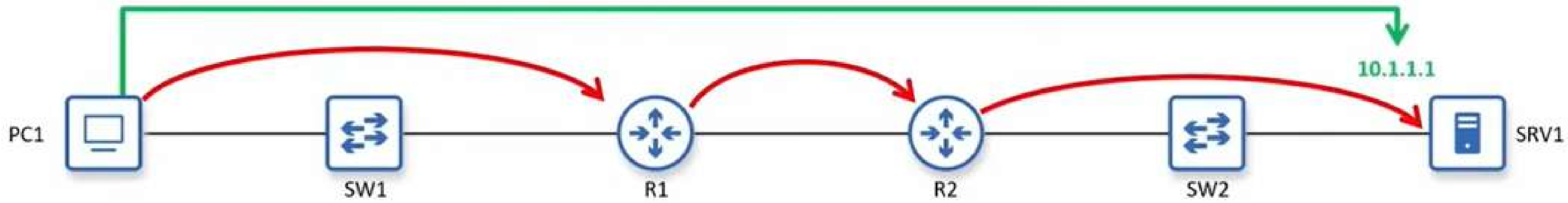
- Uses **MAC (Media Access Control) addresses** to identify interfaces.
  - **PC1** sends the message to the MAC address of **R1's G1 interface (NIC)**.
  - **R1** sends the message to the MAC address of **R2's G1 interface (NIC)**.
  - **R2** sends the message to the MAC address of **SRV1's interface (NIC)**.

- Protocols at this layer include:
  - Ethernet (IEEE 802.3)
  - Wi-Fi (IEEE 802.11)



exist, of course, but these are by far the
most commonly used Layer-2 protocols today.

Layer 3: The Internet layer



- The **Internet layer** (Layer 3) provides end-to-end delivery between hosts across multiple networks.
  - Internet = internetwork (between networks)
- Uses **IP addresses** to identify hosts in the network.
  - When PC1 sends a message to SRV1, it addresses the message to SRV1's IP address.
- **Routers** operate mainly at this layer, using the message's destination IP address to forward the message toward its final destination host.
- Protocols at this layer include:
  - IP (IPv4, IPv6)
  - ICMP (Internet Control Message Protocol)

Application layer

Transport layer

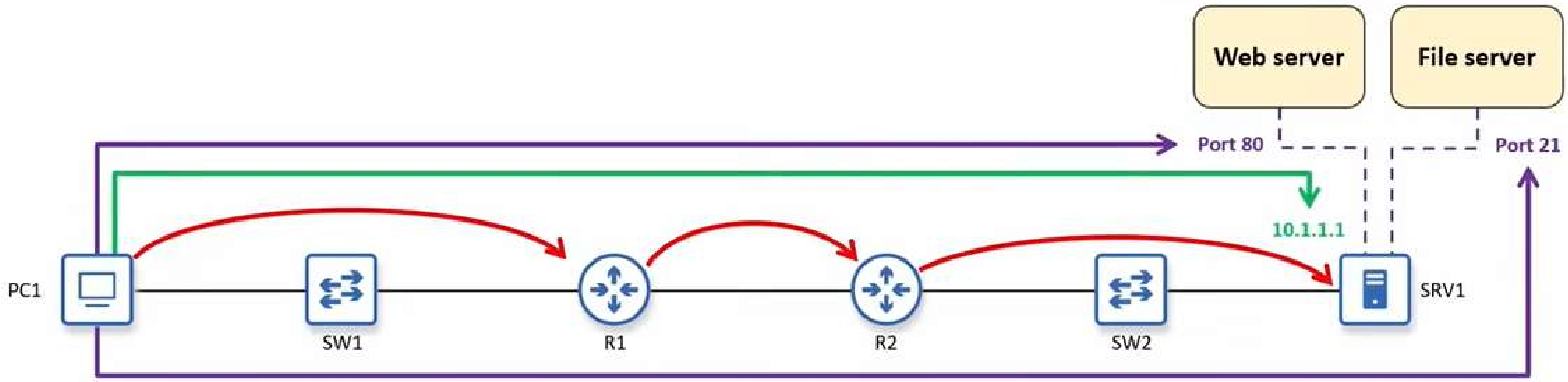
Internet layer

Local Network layer

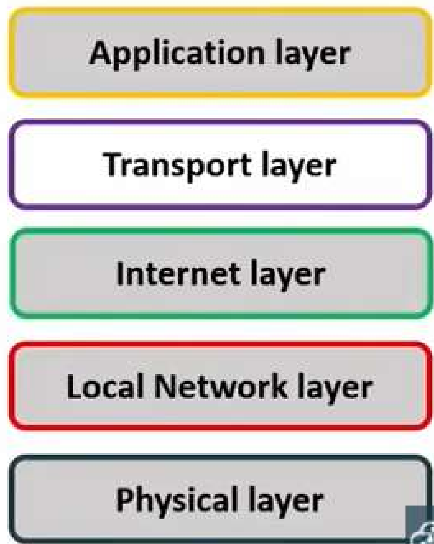
Physical layer

both version 4 and version 6, and ICMP,
the Internet Control Message Protocol.

Layer 4: The Transport layer

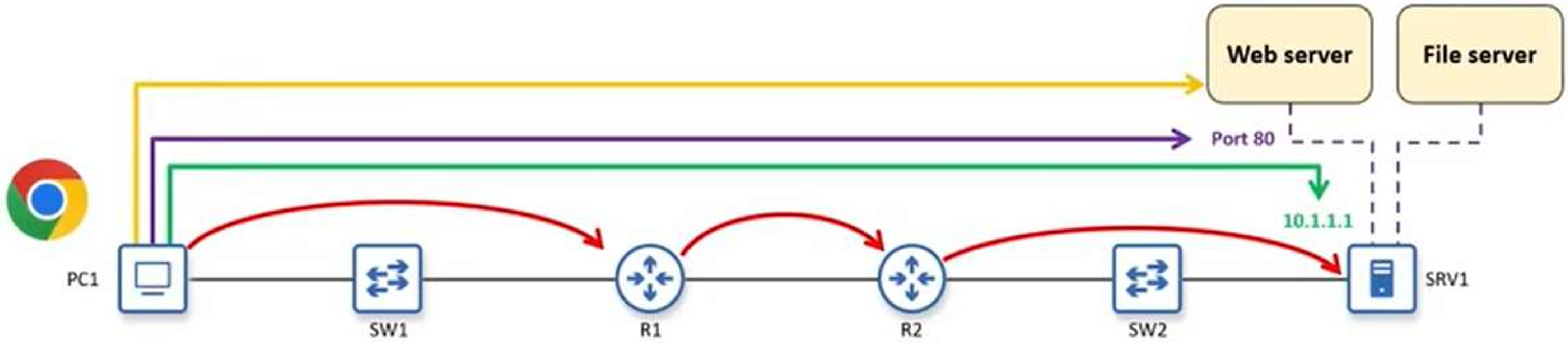


- The **Transport layer** (Layer 4) provides end-to-end communication between application processes.
  - Also called "process-to-process" or "service-to-service".
- Uses **port numbers** to identify the processes on each host.
  - When the web client on PC1 wants to send a request to the web server running on SRV1, it addresses the message to port 80.
- Runs mainly on the communicating hosts (PC1 and SRV1); routers normally operate based on IP (Layer 3), not on Transport-layer information.
- Protocols at this layer include:
  - UDP (User Datagram Protocol): simple and efficient
  - TCP (Transmission Control Protocol): more robust features beyond basic message addressing



Control Protocol. They both offer some different features, and we'll cover them in this course.

Layer 5: The Application layer

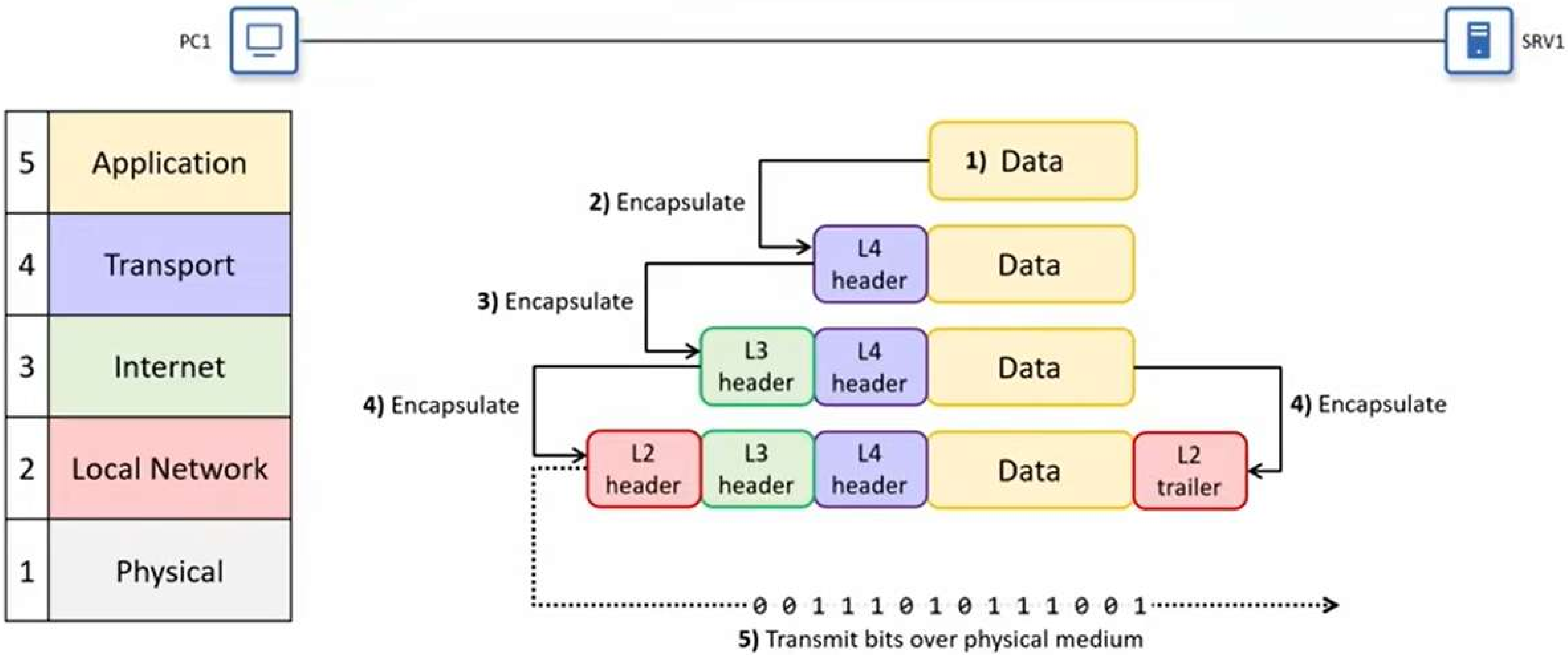


- The **Application layer** (Layer 5) is where network communications meet applications.
  - Usually called **Layer 7** (more info on that later...)
- Defines how application processes format, send, and interpret data.
- Protocols at this layer define message formats and rules for specific tasks, such as:
  - Browsing web pages (HTTP/HTTPS)
  - Transferring files (FTP, TFTP)
  - Sending/receiving email (SMTP, POP3, IMAP)
- Network infrastructure devices (routers, switches) don't care about Application-layer details.
  - They just move messages across the network.
  - Only the communicating hosts interpret the data.

Application layer
Transport layer
Internet layer
Local Network layer
Physical layer

how does a single message actually include all of
this information at once? Let's see how the layers

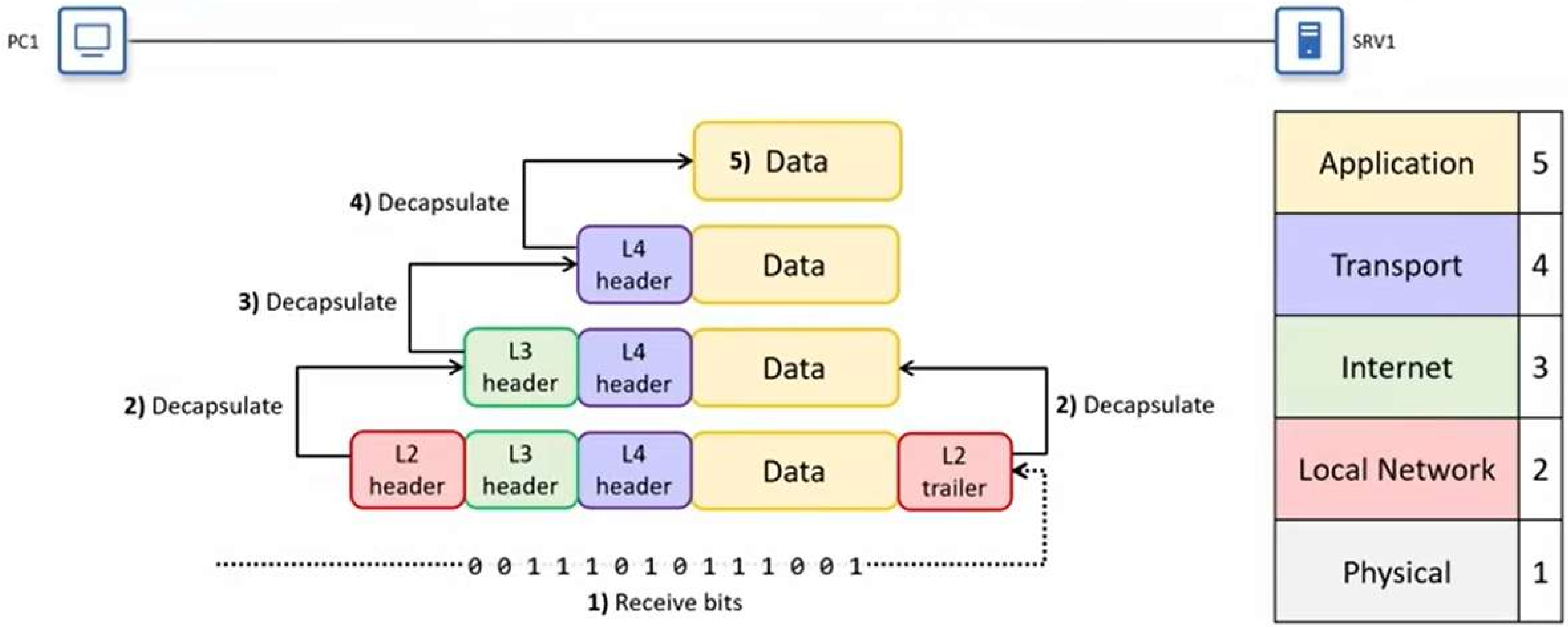
Encapsulation & decapsulation



- The **Application layer** prepares the data to be sent over the network.
- As the message moves down the stack, each layer **encapsulates** the data with a **header** including the information needed for that layer.
  - Source and destination addresses (port numbers, IP addresses, MAC addresses), etc.
  - Layer 2 also adds a **trailer** that the receiving device uses to check for transmission errors.
- The **Physical layer** transmit
  - The **L2 header** is transmitt

right here. So the Layer 2 header is transmitted
first, and Layer 2 trailer is transmitted last.

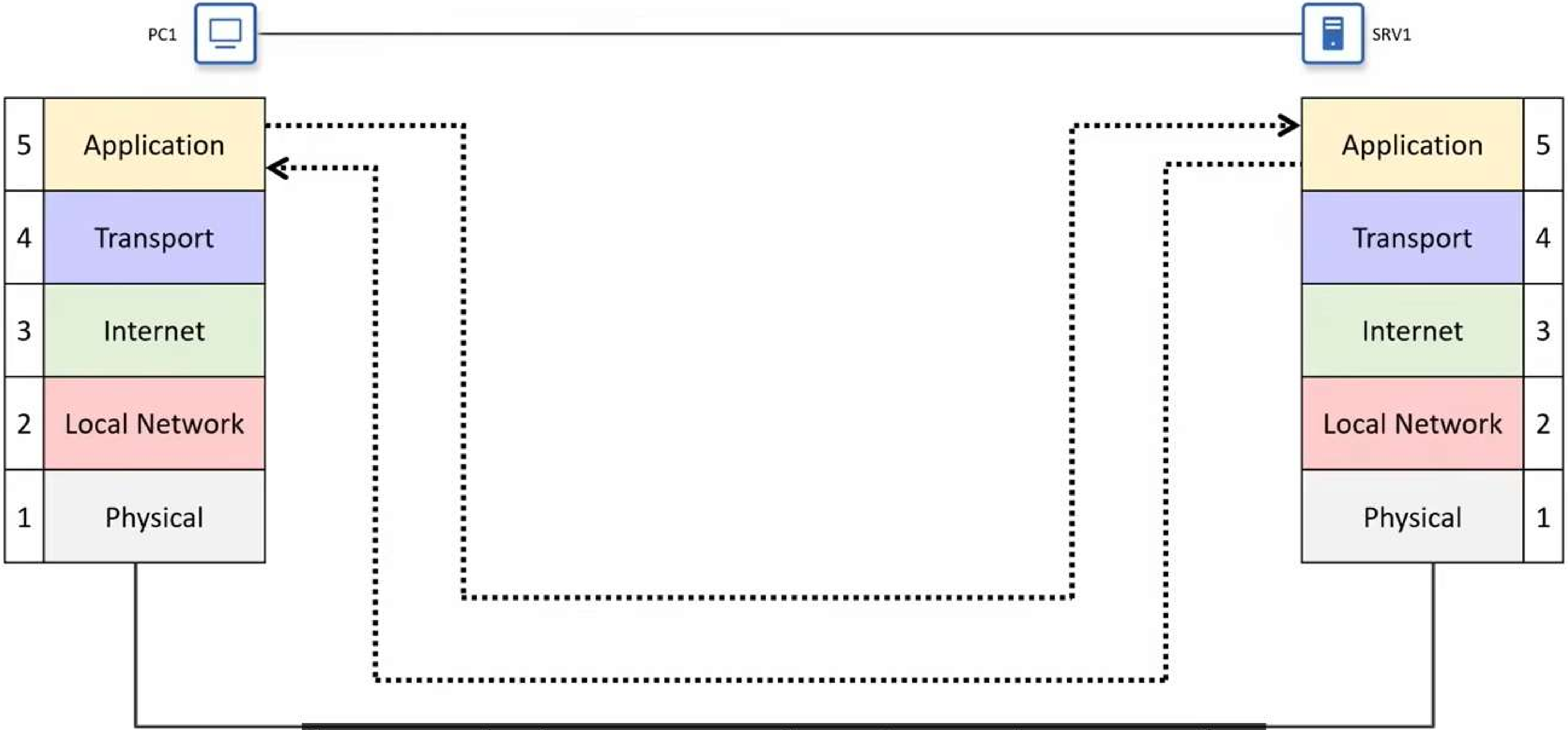
Encapsulation & decapsulation



- The receiving device receives the message as a stream of bits at Layer 1.
- The device examines the information in the Layer 2 header and trailer, and then removes them (**decapsulation**).
  - The decapsulation process continues up the stack: Layer 3 removes the L3 header, then Layer 4 removes the L4 header, and then the data is delivered to the Application layer.
- The application processes the data and, if needed, generates a response that goes back down the stack.

PC1. So, that's the decapsulation process, which is basically the encapsulation process in reverse.

Encapsulation & decapsulation



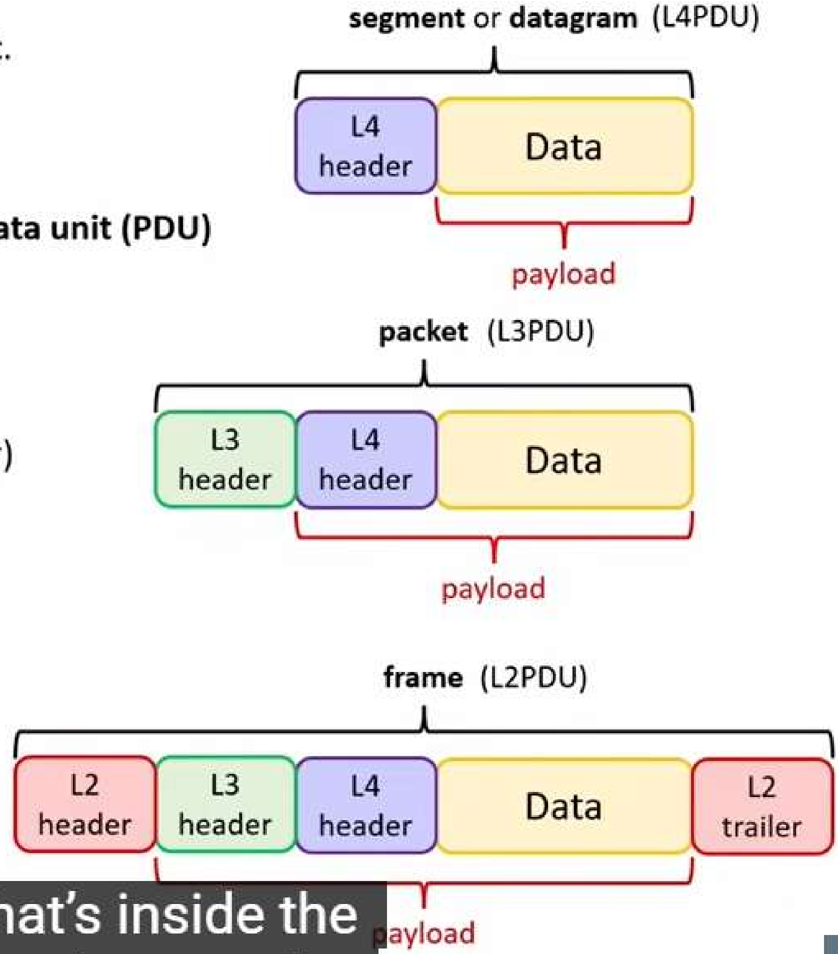
host sends the message down its stack, across the network, and up the other host's stack. We'll look

Protocol data units

- At each stage in the encapsulation/decapsulation process, there is a name given to the message:
  - The combination of data and a L4 header is called a **segment** (TCP) or **datagram** (UDP).
    - TCP creates segments, UDP creates datagrams.
  - The combination of a **segment/datagram** and a L3 header is called a **packet**.
  - The combination of a packet and a L2 header/trailer is called a **frame**.
    - This is what is actually sent over the wire.

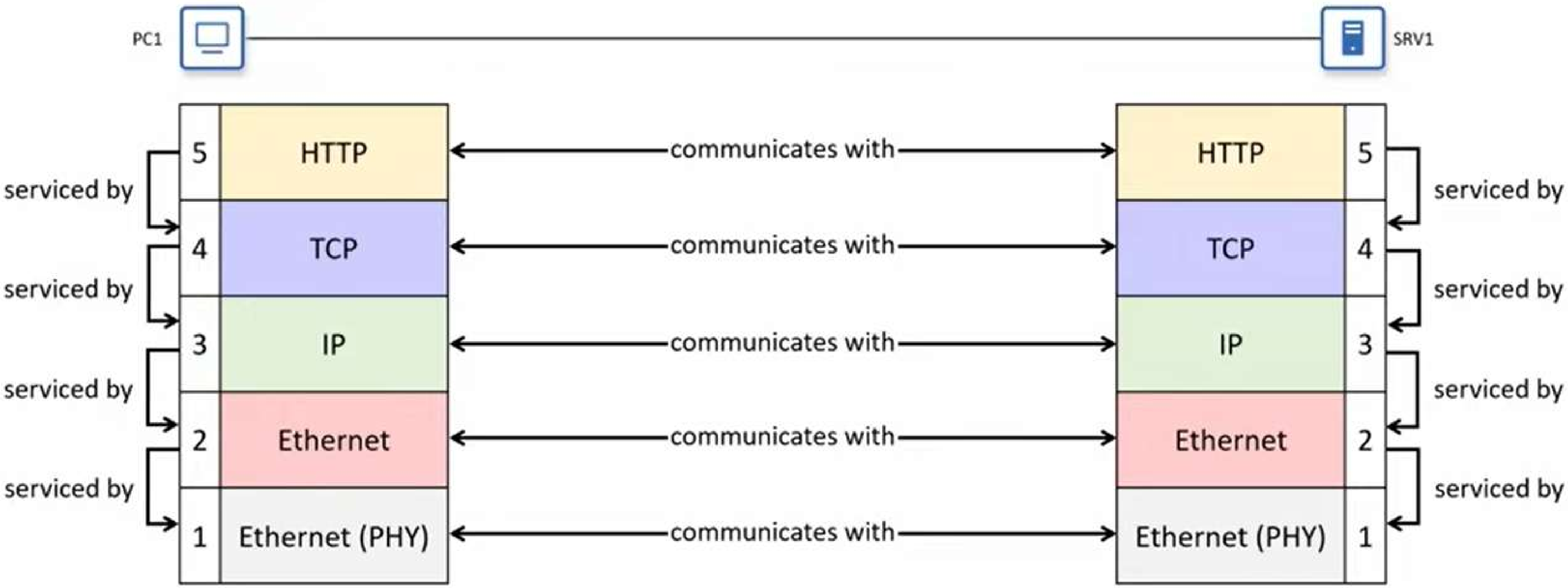
- We can use alternative names to describe the message at each stage: **protocol data unit (PDU)**
  - A segment or datagram is a **Layer 4 PDU (L4PDU)**.
  - A packet is a **Layer 3 PDU (L3PDU)**.
  - A frame is a **Layer 2 PDU (L2PDU)**.

- The contents of each PDU (everything encapsulated by that layer's header/trailer) are called the **payload**.
  - A **segment** or **datagram's** payload is the **application data**.
  - A **packet**'s payload is a **segment** or **datagram**.
  - A **frame**'s payload is a **packet**.



remember that the payload is what's inside the PDU, not including that layer's header or trailer.

Same-layer interaction



- Each layer communicates with the same layer on other devices (**same-layer interaction**).
  - The Application layer on one host sends data to the Application layer on the other host.
  - A segment/datagram is addressed to the **Layer 4 port number** of the correct application on the destination host.
  - A packet is addressed to the **Layer 3 IP address** of the destination host.
  - A frame is addressed to the
  - Signals sent out of a physical port are received by a **physical port** on the connected device.

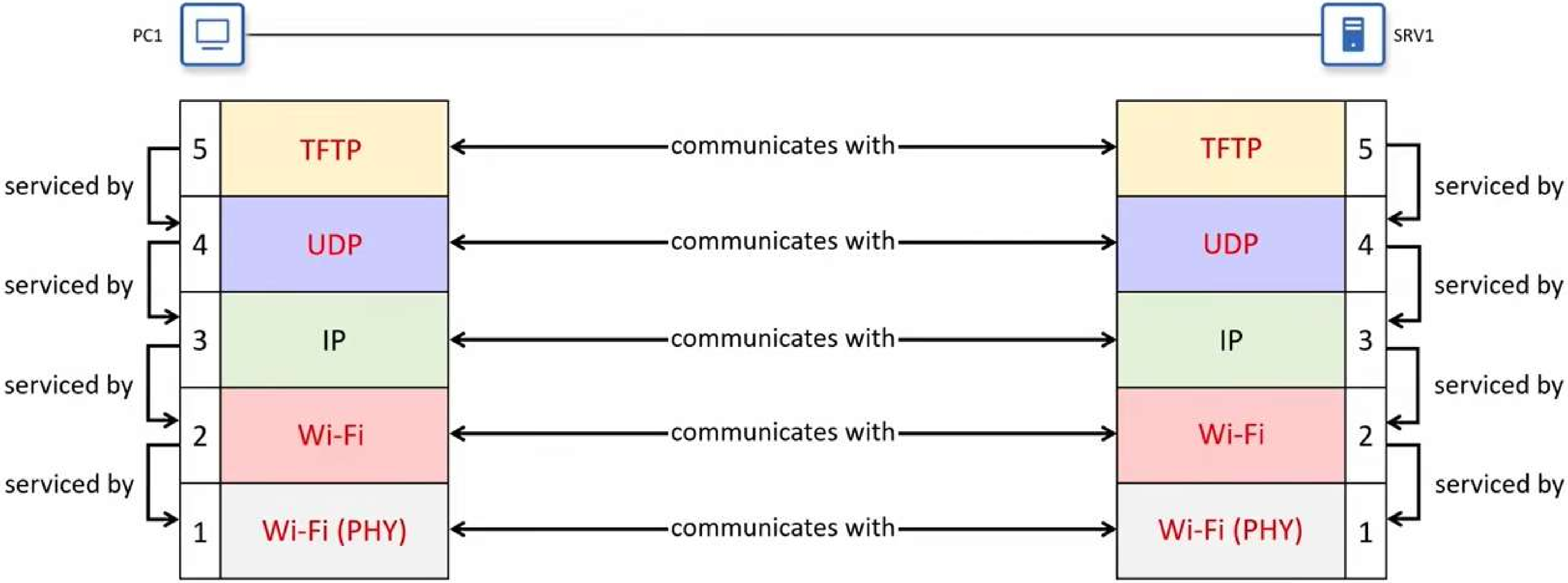
physical port on the connected device.
This layered cooperation - within each

Adjacent-layer interaction

- Each layer provides a service to the layer above it, and is serviced by the layer below it (**adjacent-layer interaction**).
  - **Layer 4** provides a service to **Layer 5** by delivering data to the correct application using port numbers.
  - **Layer 3** provides a service to **Layer 4** by delivering segments/datagrams to the correct destination host using IP addresses.
  - **Layer 2** provides a service to **Layer 3** by delivering packets to the next hop using MAC addresses.
  - **Layer 1** provides a se … nals.

or radio signals over the physical medium. Each
layer relies on the layer below it to do its job.

Separation of layers



- Each layer has its own job and provides a specific service to the layers above.
  - The layers are modular.
- As long as each layer keeps its "contract" with the other layers, we can improve or replace protocols at different layers without redesigning everything.
- That flexibility is one of the main ben

layered model. Now that we've built this 5-layer TCP/IP model, let's see

The OSI model

- TCP/IP development started in the 1970s (ARPANET work, early TCP/IP specs).
- In the late 1970s and 1980s, the **International Organization for Standardization (ISO)** designed a 7-layer **Open Systems Interconnection (OSI) model** and a matching protocol suite.
  - The goal was to create international, vendor-neutral networking standards that could unify existing proprietary stacks and potentially replace TCP/IP.
- Governments, including the US, promoted OSI as the preferred/recommended stack for new deployments.
- OSI protocols ended up being late and complex, and never gained the same deployment as TCP/IP.
  - TCP/IP "won" in the real world, although some OSI technologies are still used.
- Today, almost all real networks use TCP/IP, but the 7-layer OSI model survives as a reference/teaching model and a common way to talk about "layers".
- Most networking resources use a 5-layer model like the one covered in this video, but with names from the OSI model.
  - That is why the TCP/IP Application layer is often called **Layer 7**.

Adapted 5-layer model

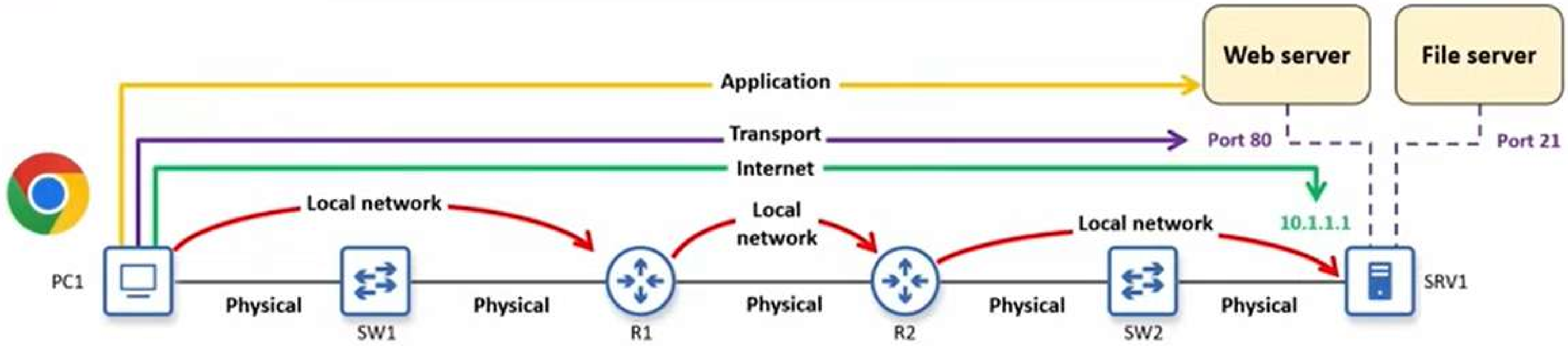
7	Application
4	Transport
3	Network
2	Data Link
1	Physical

OSI model

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

Layer 3 is called the Network layer, and
Layer 2 is called the Data Link layer.

Review (TCP/IP model)



5	Application
4	Transport
3	Internet
2	Local Network
1	Ethernet (PHY)

Common 5-layer model

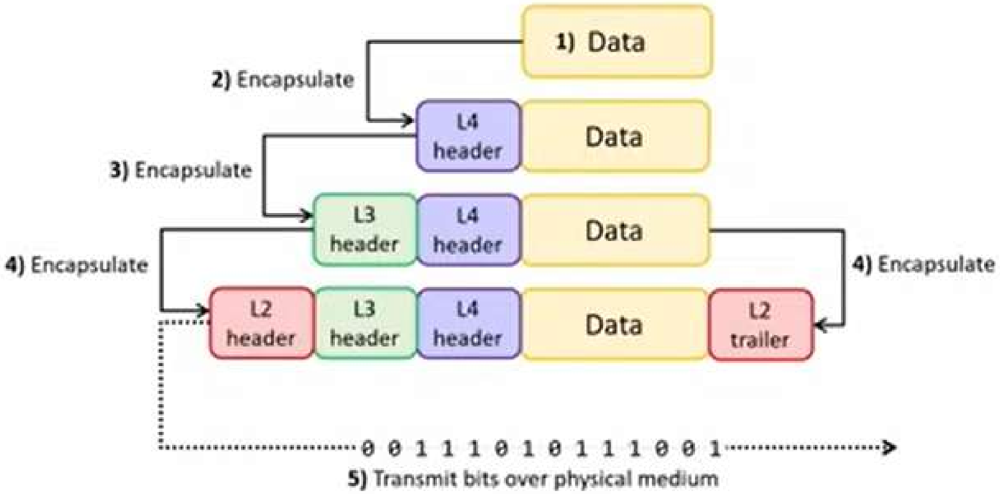
7	Application
4	Transport
3	Network
2	Data Link
1	Physical

layers. I also recommend knowing these more
commonly used names borrowed from the OSI model.

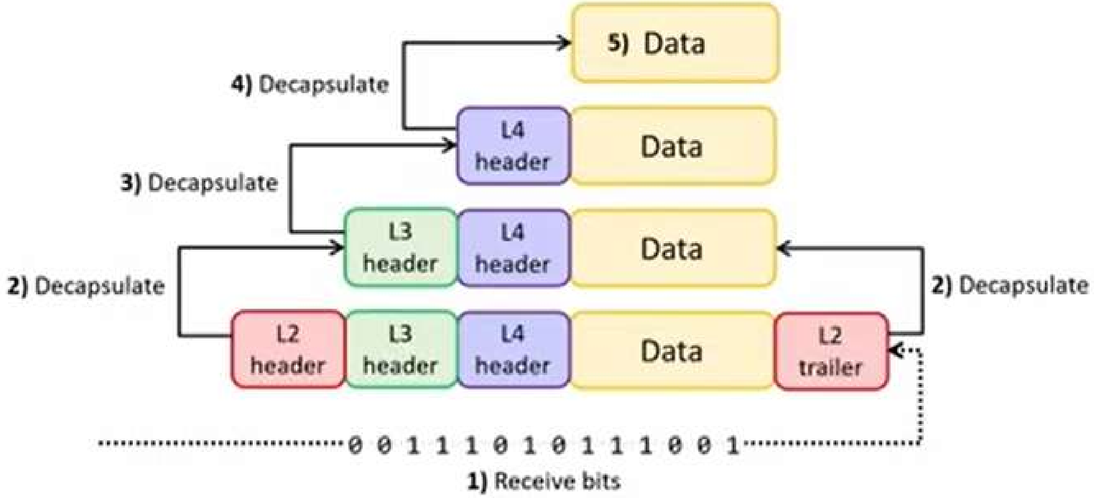
How the TCP/IP Model Actually Works | CCNA Day 3

Review (Encapsulation/Decapsulation)

Encapsulation



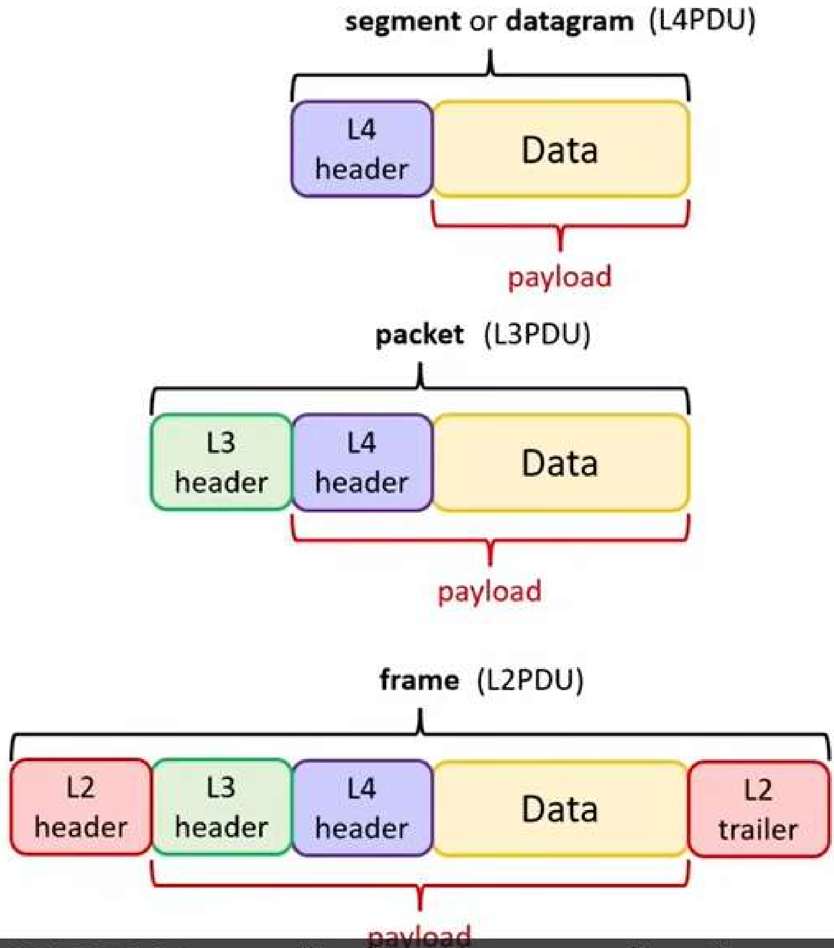
Decapsulation



the receiving host removes the headers and trailer
layer by layer until it gets to the data inside.

40:14 / 42:41 Review >

Review (PDUs)



Layer 2 PDU is called a frame; the frame is what is actually transmitted over the physical medium.