2. Computer Networks

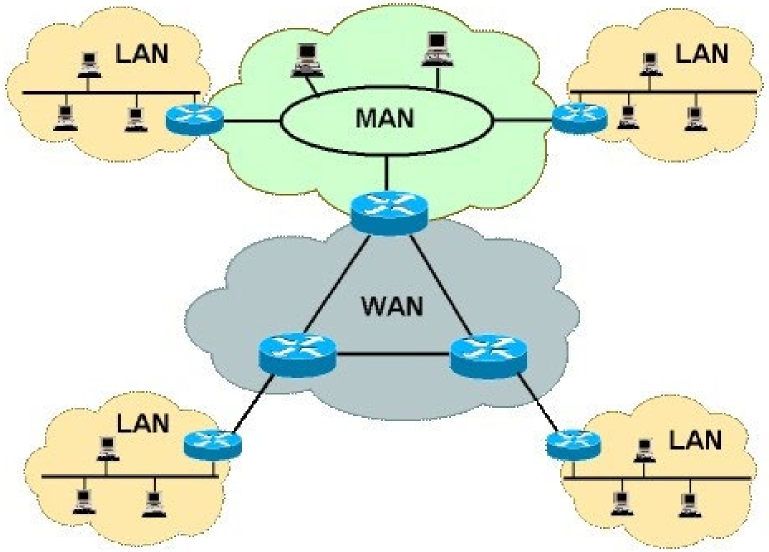
Protocols

Summary

- Internetwork
- Protocols
- Central Role of Software
- Network Communication
- Connection-oriented Services
- Connectionless Services

Internetwork

Internetwork or **Internet**: a collection of different networks connected via a gateway. Each network uses **different hardware** and, more importantly, **different protocols** to manage internal communication.



Protocols

Communication between various entities within networks is managed by **protocols**.

A **network protocol** is a formal, predefined **set of rules** that govern the interactions between two or more connected electronic devices to facilitate communication. These rules primarily address the software components of the devices.

The earliest computer networks were primarily focused on hardware, considering software merely as an adjunct. This approach is outdated. In modern contexts, **network software** is intricately structured and plays a **central role** in network functionality.

Central Role of Software/1

Feature Complexity:

Modern networks must support a wide range of services and applications, from simple data transmission to complex security operations, traffic management, performance optimization, and cloud integration. These functions require sophisticated software capable of dynamically managing various needs and configurations.

Adaptability and Scalability:

Today's networks need to be highly adaptable and scalable to swiftly respond to changing business or consumer demands. Network software enables this flexibility, allowing updates and modifications to be implemented much more quickly and at lower costs compared to hardware changes.

Security:

The security of modern networks heavily relies on advanced software to implement firewalls, intrusion detection systems, encryption, and other essential security measures to protect data and keep the network secure against increasingly sophisticated threats.

Central Role of Software/2

Network software is crucial not only for managing functionality but also for **ensuring compatibility** and **communication** among **diverse hardware** components.

By acting as an **intermediary**, it enables **different devices**, potentially from various manufacturers, to **work together** seamlessly.

This **interoperability** is key to building flexible and efficient network systems that can adapt to **new technologies** and **requirements**.

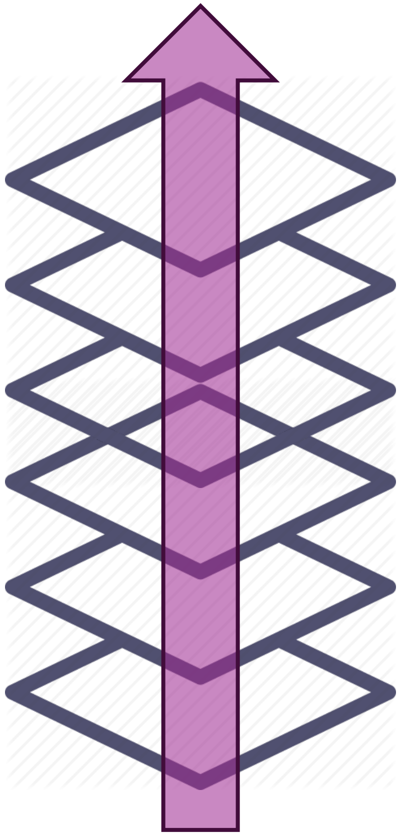
Stack of Layers

To simplify design, networks are structured into **layers** that are built upon one another.

Differences between networks can include:

- The number of layers
- The names of the layers
- The content of each layer
- The functionality of each layer

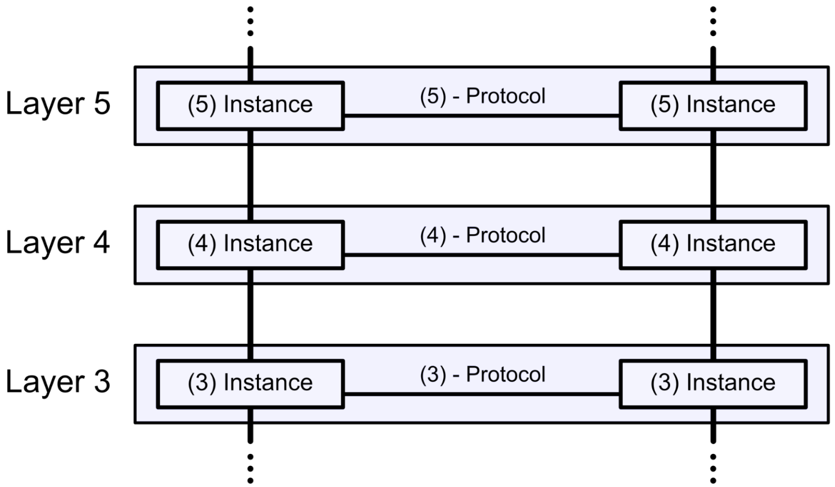
Each layer serves to provide **specific services** to the layers above it, while **abstracting the details** of how those services are implemented.



Layer Protocol

When the same network layer, referred to as **layer n**, communicates across different machines, the rules and conventions guiding this communication are known collectively as the **layer n protocol**.

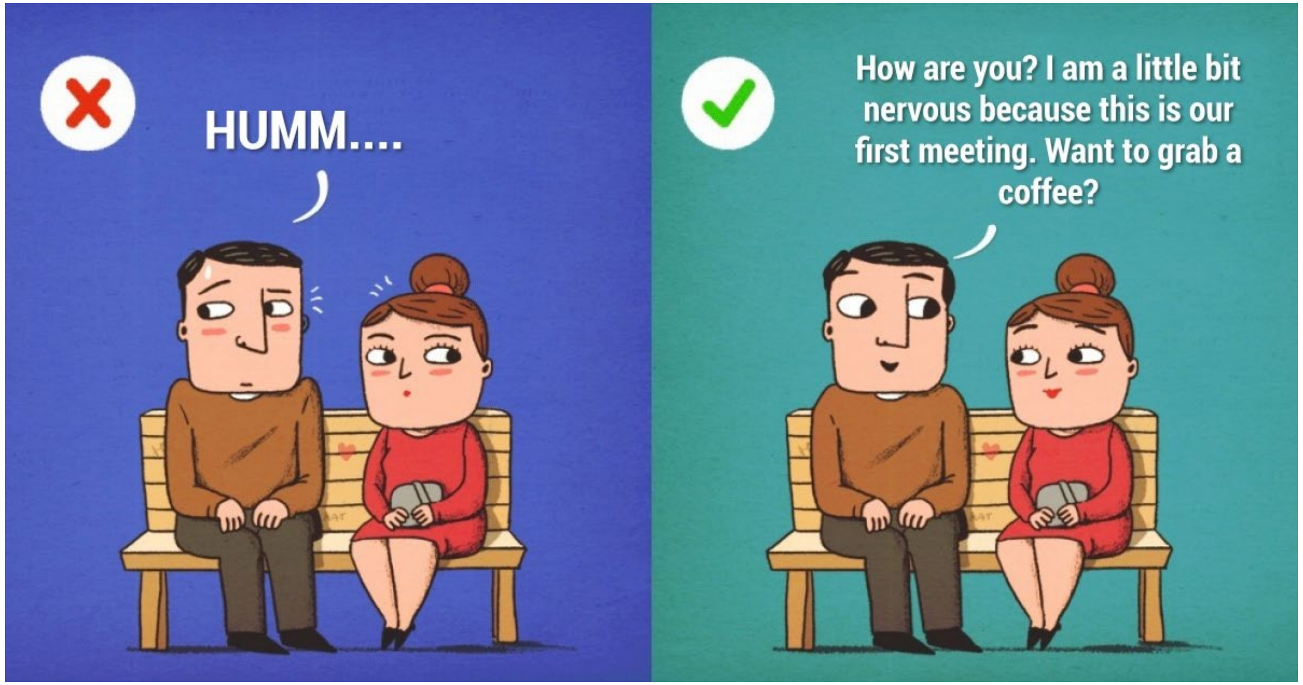
A **protocol** is essentially an agreement that dictates the **procedures for communication** between the parties involved.



Communication between the n-th layer of one machine and the n-th layer of another machine is enabled by the services offered by the underlying layer (n-1).

Protocols - Analogy

- Disregarding the protocol could complicate communication, potentially to the point of impossibility.


HUMM....
How are you? I am a little bit nervous because this is our first meeting. Want to grab a coffee?

Network Communication - Analogy/1

Start and End Points:

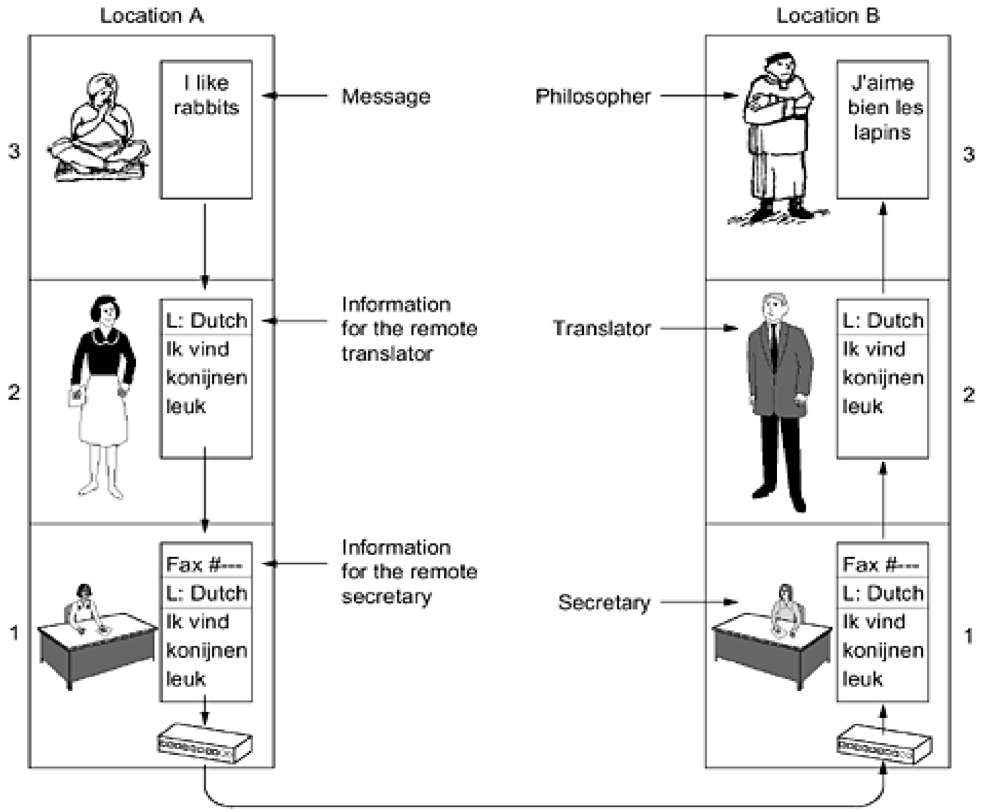
The process begins at Location A where a person (the sender) expresses a message, "I like rabbits." It ends at Location B with a philosopher (the receiver) receiving the message translated into French, "J'aime bien les lapins." In networking, these points represent the source and destination nodes.

Transmission and Translation Layers:

The message undergoes several steps before reaching the philosopher. Initially, it is translated into Dutch and passed through various intermediaries (secretary and translator) who handle and forward the information, akin to data passing through multiple network devices like routers and switches.

Data Packet Analogy:

The original message "I like rabbits" can be seen as data packets. As it moves from the source, it is processed and repackaged at various stages—translation into another language can represent encoding or encryption in network terms.



Network Communication - Analogy/2

Routing:

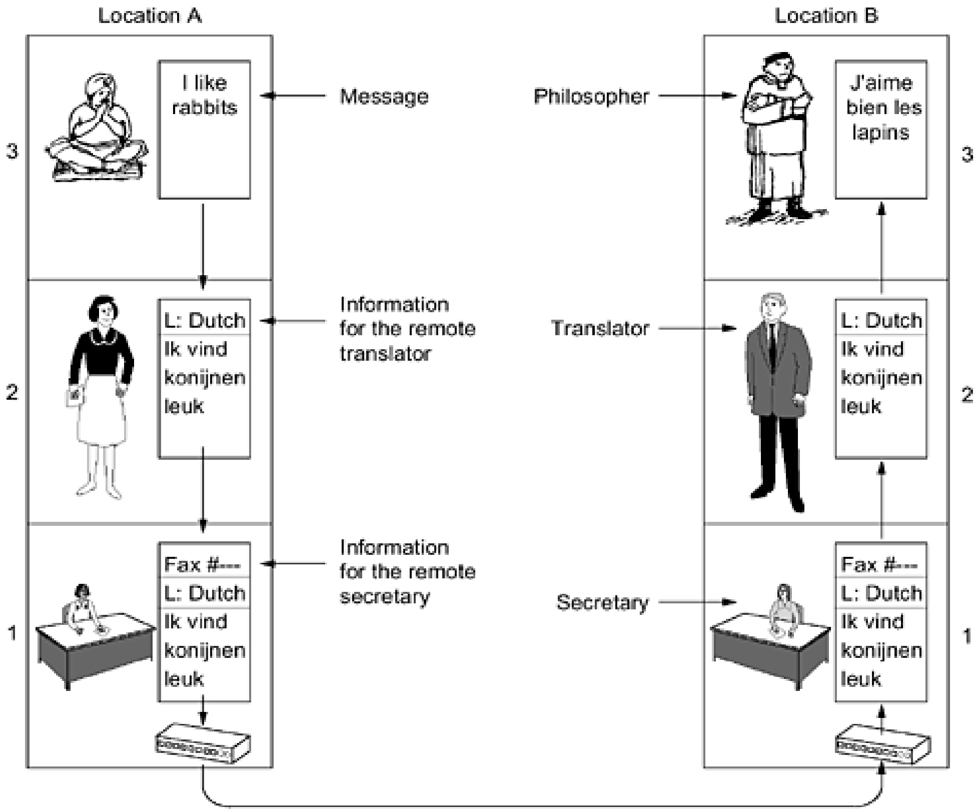
The use of faxes and different personnel for processing the information is similar to how data packets are routed through different network paths. Each node (person or device in the diagram) decides how and where to send the data next based on network protocols.

Protocol Layers:

Each step in the diagram where the message is handled and processed represents different layers in network protocols (like the OSI model). For example, the secretary could be seen as operating in the presentation layer, translating data formats, while the translator operates in the application layer, converting the message contents into different languages.

End-to-End Delivery:

Just as the message starts as a simple expression and ends up being understood by someone who speaks a different language, in networking, data starts from a source and is delivered to a destination across various network segments and possibly transformed or secured in the process, ensuring it is usable upon receipt.


Location A
Location B
I like rabbits
Message
Philosopher
J'aime bien les lapins
3
3
L: Dutch Ik vind konijnen leuk
Information for the remote translator
Translator
L: Dutch Ik vind konijnen leuk
2
2
Fax #---
L: Dutch Ik vind konijnen leuk
Information for the remote secretary
Secretary
Fax #---
L: Dutch Ik vind konijnen leuk
1
1

Network Communication

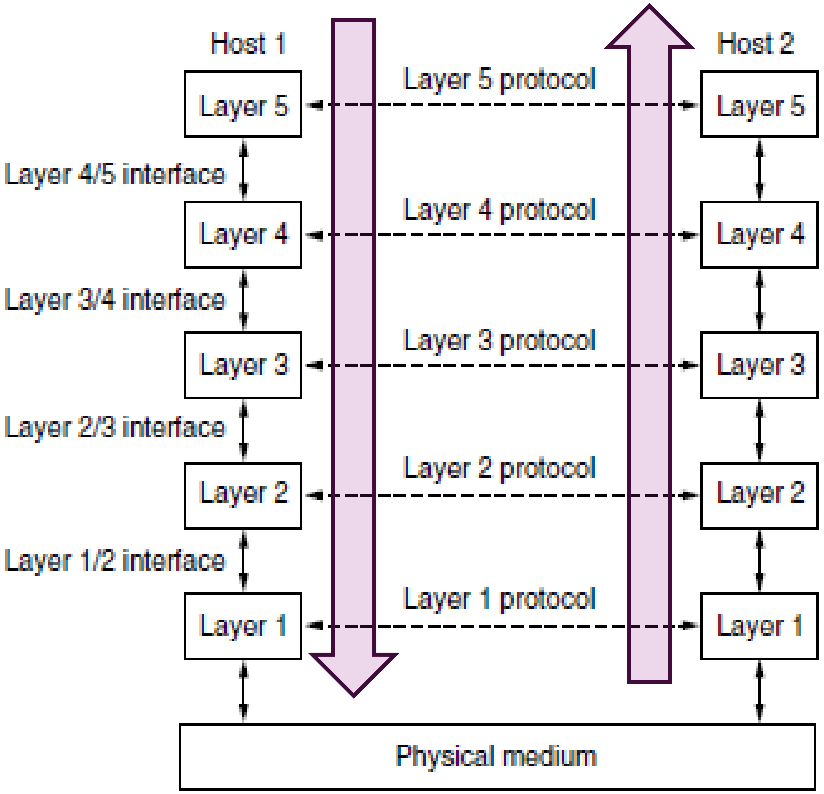
Data is not directly transferred from layer *n* on one machine to layer *n* on another machine.

- **Source (top-down):** each layer passes data and control information down to the immediately lower layer, continuing until it reaches the lowest layer.

- **Destination (bottom-up):** each layer passes data and control information up to the immediately upper layer, continuing until it reaches the highest layer.

Interface: Specifies the operations and services that the lower layer provides to the upper one.

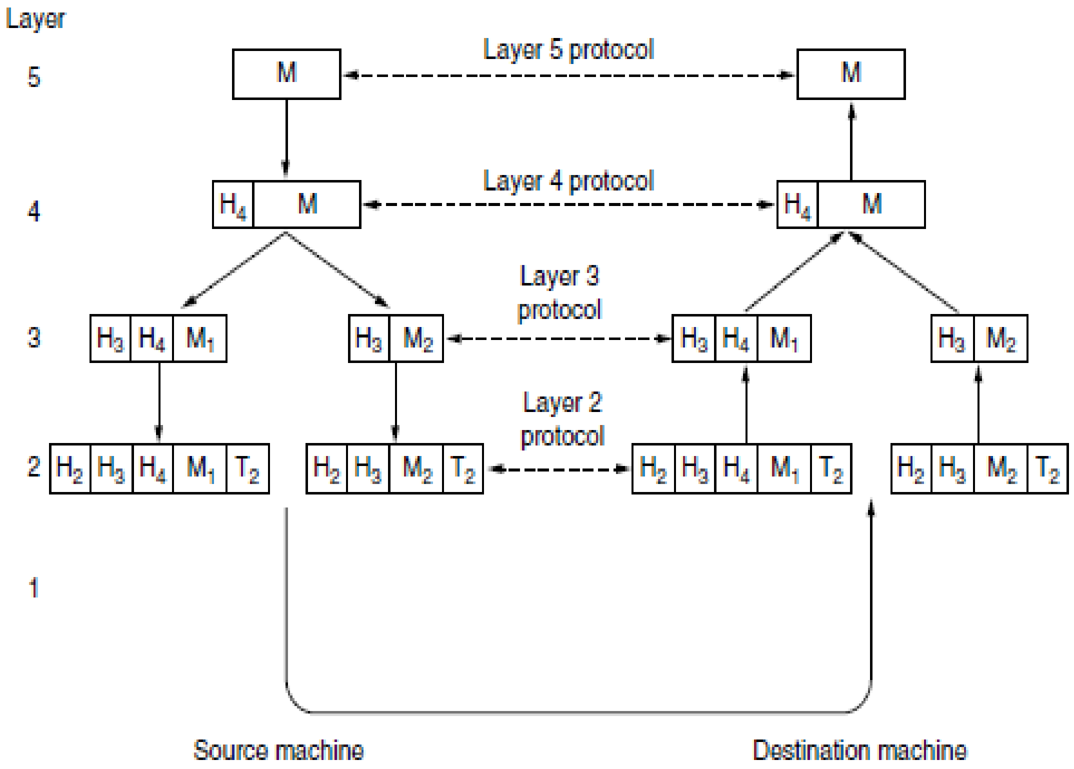
Below level 1, the **physical medium** facilitates data transfer from host 1 to host 2.



Network Communication – Technical Example/1

The application program needs to send the message M to its peer entity (i.e., the same application program on another machine).

1. **Layer 5** delivers message M to **layer 4.**
2. **Layer 4** adds a header $\boldsymbol{H_4}$ to the message. The headers contain control information such as sequence number, data offset, acknowledgment number, size, etc. The **layer 4** delivers message M to **layer 3**.
3. **Layer 3** divides the data into smaller units (packets) and adds a header $\boldsymbol{H_3}$ to each packet. The headers contain source and destination IP addresses, total length, a unique identifier for the packet, header length, TTL etc. The **layer 3** delivers packets to **layer 2**.

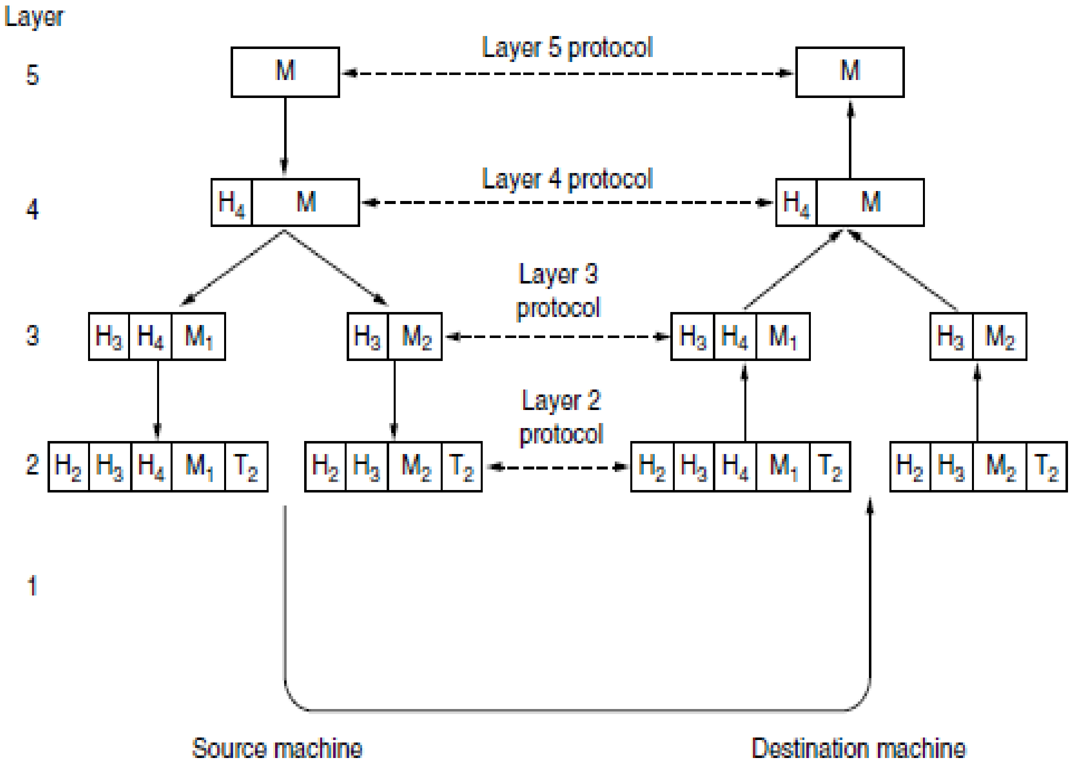


Network Communication – Technical Example/2

4. **Layer 2** adds a header $\boldsymbol{H_2}$ and a trailer $\boldsymbol{T_2}$ to each received packet referred as to frame. The headers contain source and destination MAC addresses, payload length, etc. The trailer is used for transmission error-checking. **Layer 2** delivers packets to **layer 1**.

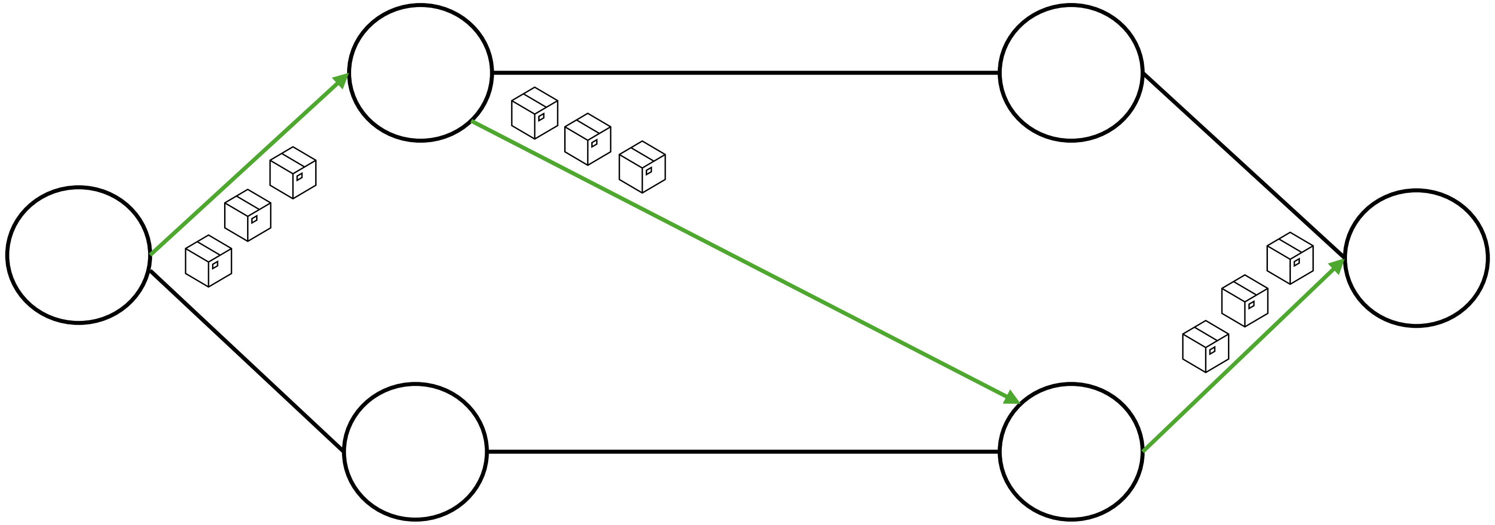
5. **Layer 1**, the **physical layer**, transfers data to the destination machine.

6. **At the destination machine,** the message moves upward through each layer, with headers and trailers being removed at each step. The message is **fully reconstructed** and becomes readable once it reaches **layer 5**.



Connection-oriented Services/1

Connection-oriented services are those in which a **dedicated connection** is established between the two communication nodes before data transfer begins. This type of service ensures that data is **transmitted reliably** and in the **correct order**.



Connection-oriented Services/2

Key characteristics of connection-oriented services:

- **Reliability:**

Data transmission is guaranteed, and transmission errors are detected and corrected. Used in applications where reliability is more important than speed, such as email and online banking.

- **Ordering:**

Data arrives in the order it was sent.

- **Flow Control:**

Regulates the amount of data sent to prevent overwhelming the receiver.

- **Establishing and Closing the Connection:**

Requires an initial handshake process to establish the connection and a closing process to terminate it.

Connection-oriented Services/3

Handshaking (Opening the Connection)

1. **Connection request:**

Device A sends a request message to Device B to initiate the connection.

2. **Response:**

Device B responds to Device A, confirming that it has received the request and is ready to communicate.

3. **Confirmation:**

Device A sends a confirmation that it has received Device B's response.

Closing the Connection

1. **Termination request:**

When the communication is complete, Device A sends a termination request to Device B.

2. **Acknowledgment:**

Device B acknowledges the termination request, indicating it is ready to close the connection.

3. **Final confirmation:**

Device A sends a final confirmation, completing the termination process.

Connection-oriented Services/4

The **ACK** (**Acknowledgment**) service is a mechanism used to ensure that the data sent is correctly received by the recipient.

1. **Packet sending:**

Device A sends a data packet to Device B.

2. **Reception and acknowledgment:**

When Device B receives the packet, it sends an acknowledgment message (ACK) to Device A to indicate that the packet has been received correctly.

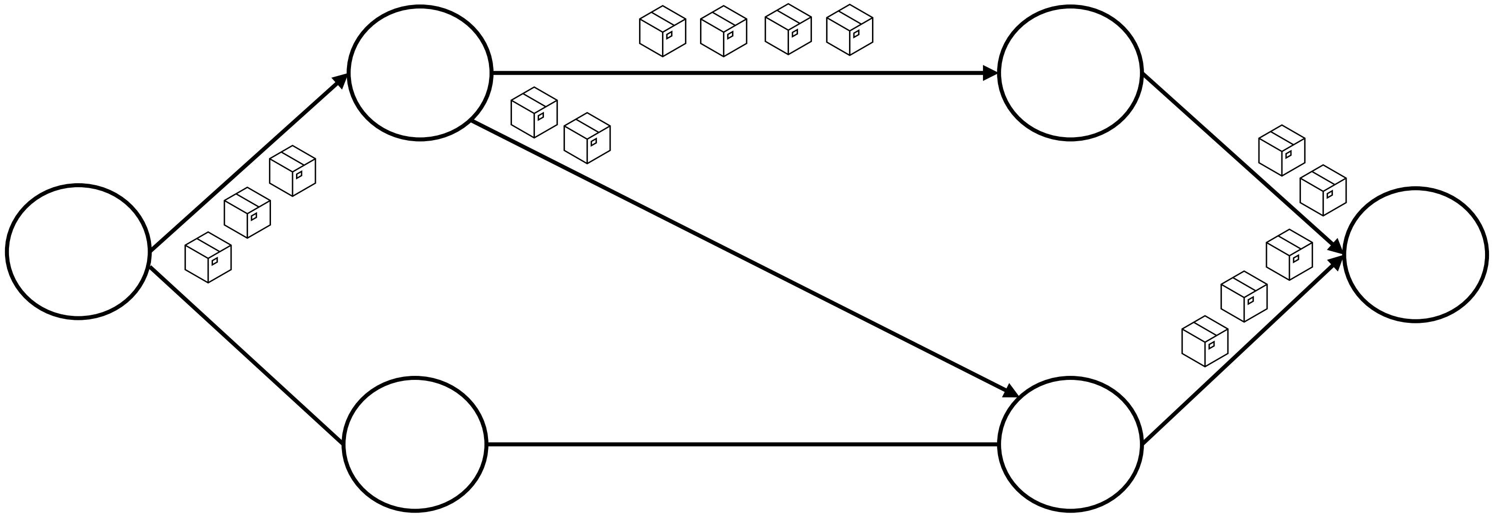
- **Retransmission in case of error:** If Device A does not receive the ACK within a certain period of time, it retransmits the packet, assuming it was lost or damaged during transmission.

This process continues **for each data packet** sent, ensuring that all packets are **received correctly** and in the **right order**.

Connectionless Services/1

Connectionless services do **not require** the establishment of a **dedicated connection** before sending data. Instead, each data packet is **treated independently** and contains all the necessary information to reach its destination.

Two packets with the same source and destination machines can travel through different paths and arrive at different times.



Connectionless Services/2

Key characteristics of connectionless services:

- **Simplicity and Speed:**

Does not require a handshake process, thus having lower latency compared to connection-oriented services.

- **No Delivery Guarantee:**

Packets can be lost, duplicated, or arrive out of order.

- **Packet Independence:**

Each packet is treated separately and contains all the necessary information to reach its destination.

- **Usage in Specific Applications:**

Used in applications where speed is more important than reliability, such as streaming and online gaming.