

Basic Concepts In Computer Networking

Antonio Carzaniga

Faculty of Informatics
Università della Svizzera italiana

September 21, 2018

Goal of this Lecture

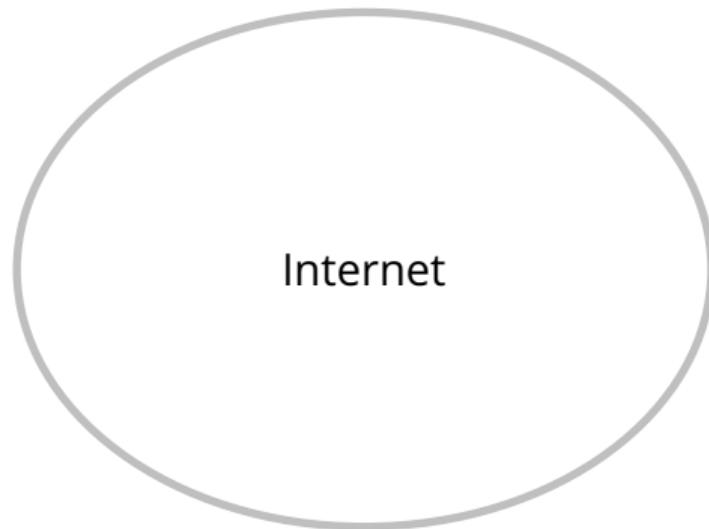
- Understand what *packet switching* is
- Understand what *circuit switching* is
- Understand their differences
- Understand what a *protocol* is

- What is the Internet?
- Types of network
- Types of service
- Protocols
- The Internet protocol stack

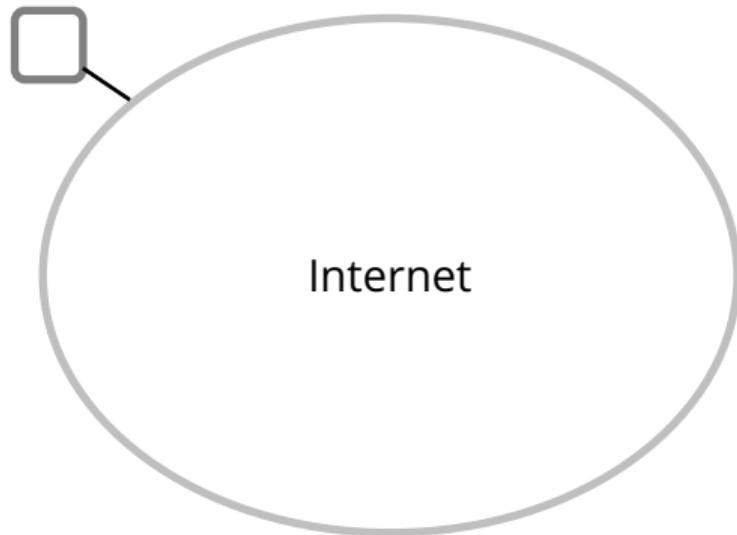




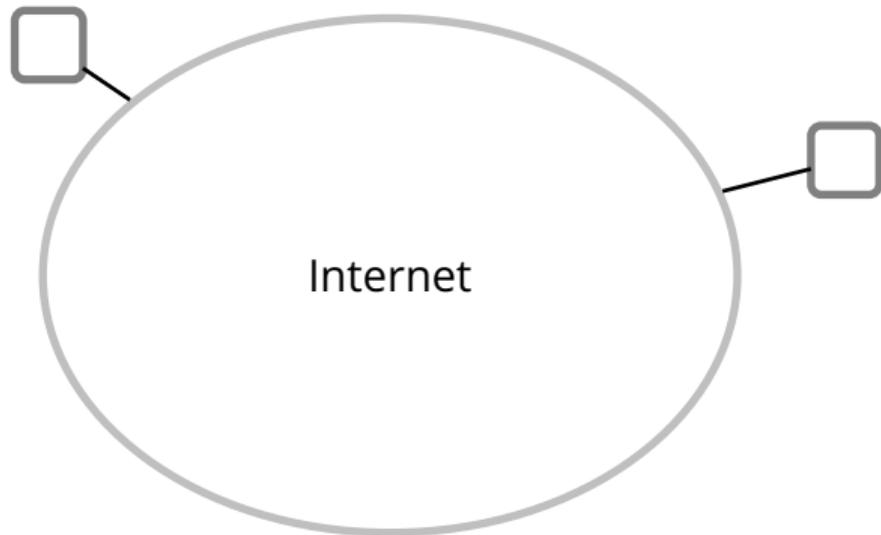
What is the Internet?



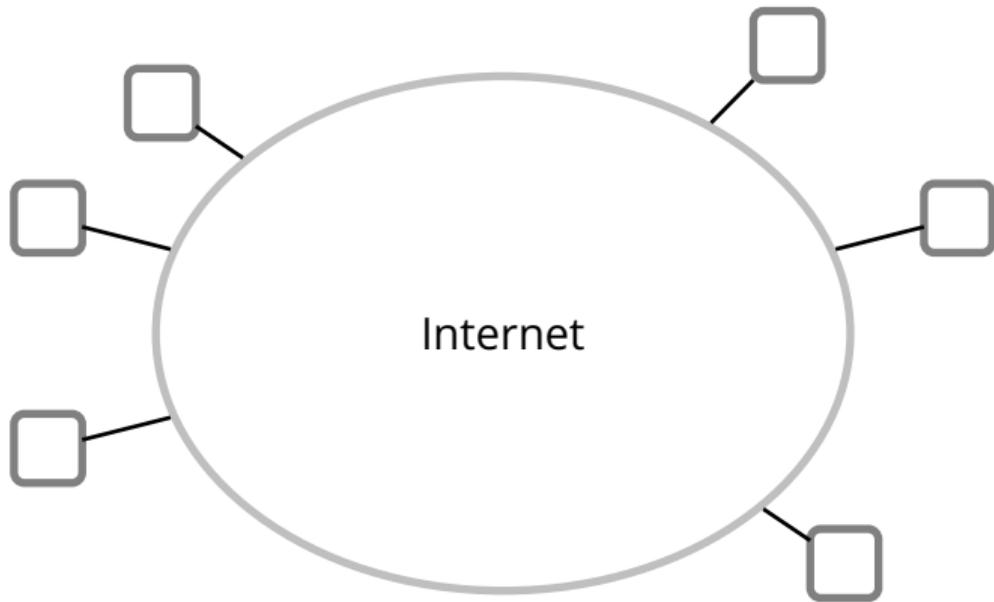
What is the Internet?



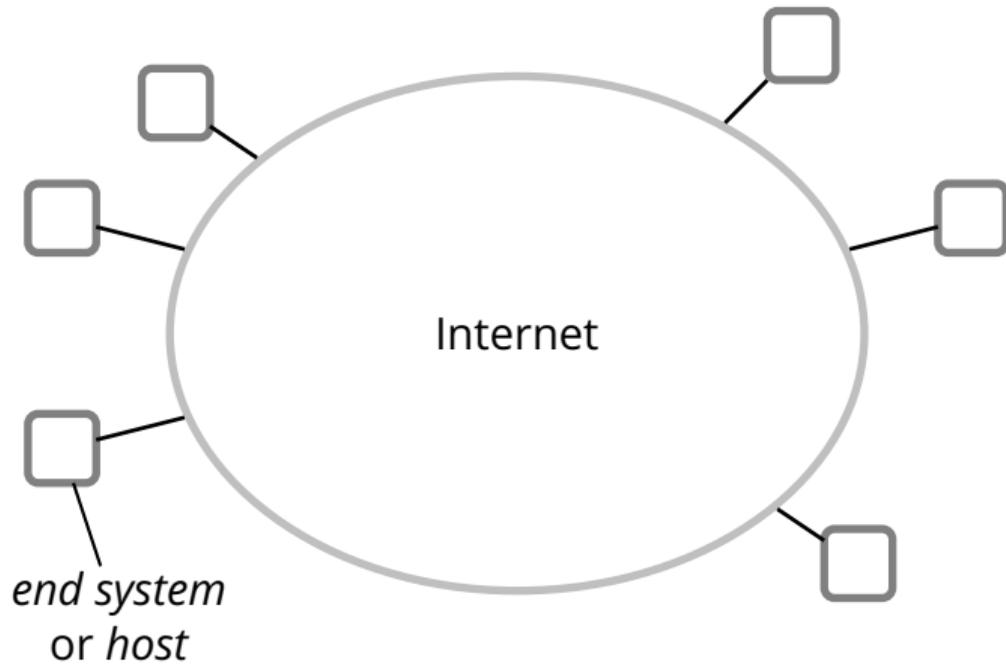
What is the Internet?



What is the Internet?



What is the Internet?



■ *End system* or *host* ()

- **End system** or **host** ()
 - ▶ a computer

■ **End system** or **host** ()

- ▶ a computer
- ▶ a phone (more or less “smart”)

■ *End system* or *host* ()

- ▶ a computer
- ▶ a phone (more or less “smart”)
- ▶ a server (well, that would also be a computer)

■ *End system* or *host* ()

- ▶ a computer
- ▶ a phone (more or less “smart”)
- ▶ a server (well, that would also be a computer)
- ▶ a camera (a.k.a., webcam)

■ *End system* or *host* ()

- ▶ a computer
- ▶ a phone (more or less “smart”)
- ▶ a server (well, that would also be a computer)
- ▶ a camera (a.k.a., webcam)
- ▶ a temperature sensor

■ *End system* or *host* ()

- ▶ a computer
- ▶ a phone (more or less “smart”)
- ▶ a server (well, that would also be a computer)
- ▶ a camera (a.k.a., webcam)
- ▶ a temperature sensor
- ▶ a PDA

■ *End system* or *host* (□)

- ▶ a computer
- ▶ a phone (more or less “smart”)
- ▶ a server (well, that would also be a computer)
- ▶ a camera (a.k.a., webcam)
- ▶ a temperature sensor
- ▶ a PDA
- ▶ ...
- ▶ a car
- ▶ a television set
- ▶ a picture frame
- ▶ a toaster
- ▶ ...

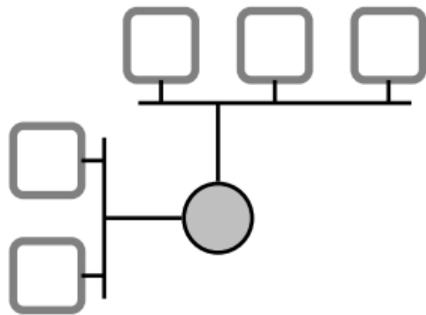
■ *End system* or *host* (□)

- ▶ a computer
- ▶ a phone (more or less “smart”)
- ▶ a server (well, that would also be a computer)
- ▶ a camera (a.k.a., webcam)
- ▶ a temperature sensor
- ▶ a PDA
- ▶ ...
- ▶ a car
- ▶ a television set
- ▶ a picture frame
- ▶ a toaster
- ▶ ...
- ▶ a toilet seat?
- ▶ a toothpick?
- ▶ ...

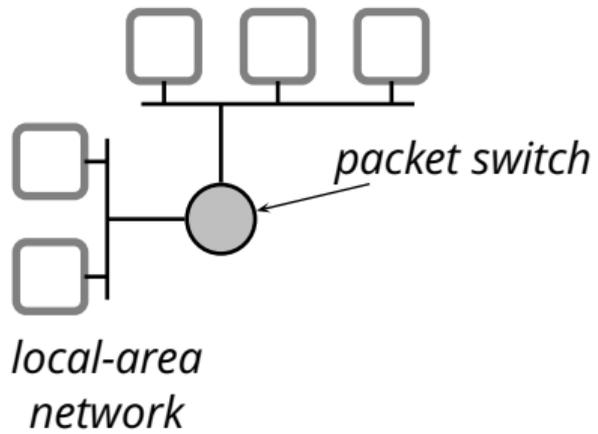
What is *Inside* the Internet?



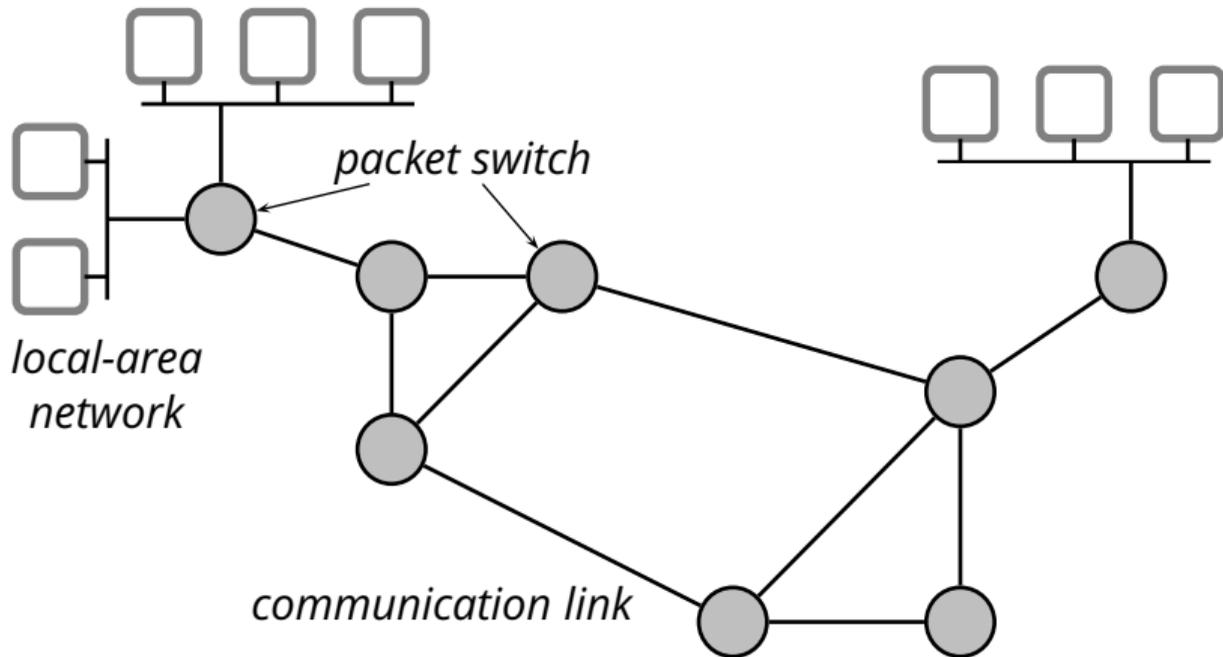
What is *Inside* the Internet?



What is *Inside* the Internet?



What is *Inside* the Internet?



- The Internet uses *packet switching*

- The Internet uses *packet switching*
- ***Packet switch:*** a *link-layer switch* or a ***router***

- The Internet uses ***packet switching***
- ***Packet switch:*** a *link-layer switch* or a ***router***
- ***Communication link:*** a connection between packet switches and/or end systems

- The Internet uses ***packet switching***
- ***Packet switch:*** a *link-layer switch* or a ***router***
- ***Communication link:*** a connection between packet switches and/or end systems
- ***Route:*** sequence of switches that a packet goes through (a.k.a. *path*)

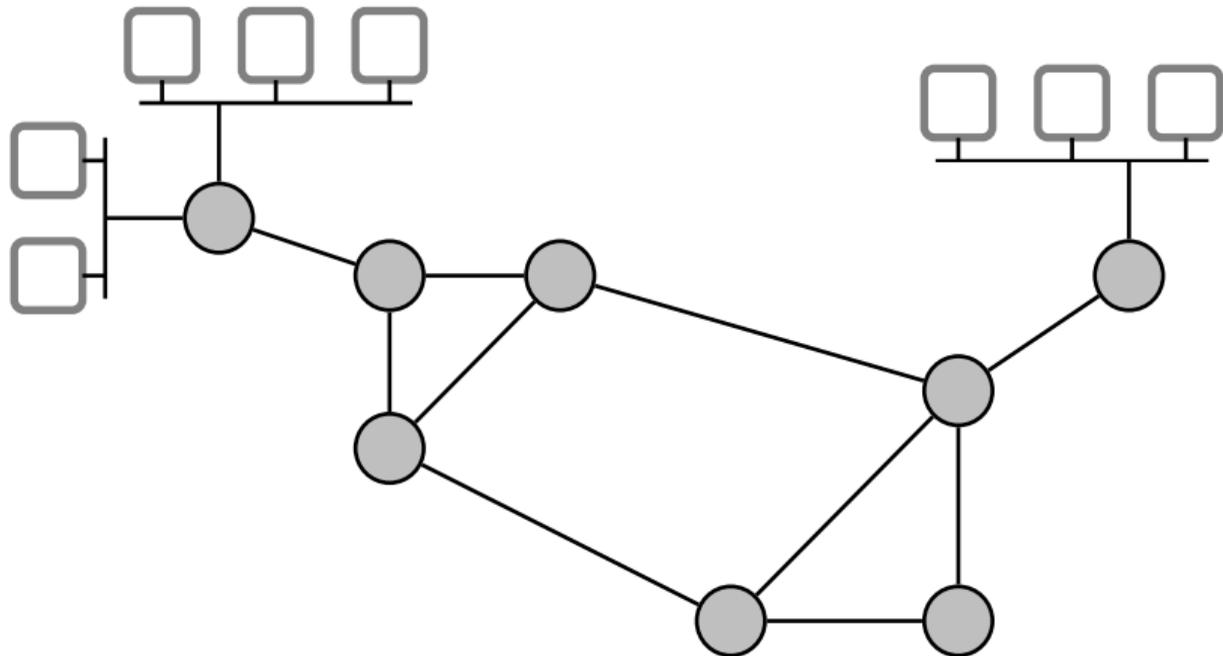
- The Internet uses ***packet switching***
- ***Packet switch***: a *link-layer switch* or a ***router***
- ***Communication link***: a connection between packet switches and/or end systems
- ***Route***: sequence of switches that a packet goes through (a.k.a. *path*)
- ***Protocol***: control the sending and receiving of information to and from end systems and packet switches

- Various types and forms of medium

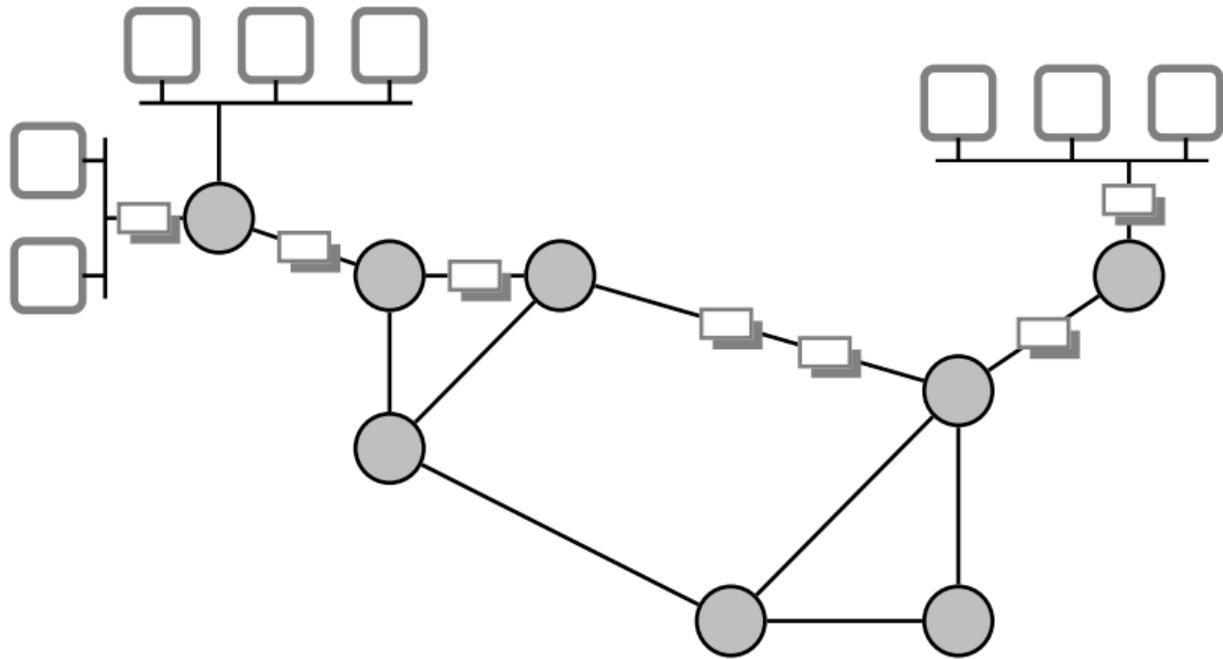
■ Various types and forms of medium

- ▶ Fiber-optic cable
- ▶ Twisted-pair copper wire
- ▶ Coaxial cable
- ▶ Wireless local-area links (e.g., 802.11, Bluetooth)
- ▶ Satellite channel
- ▶ ...

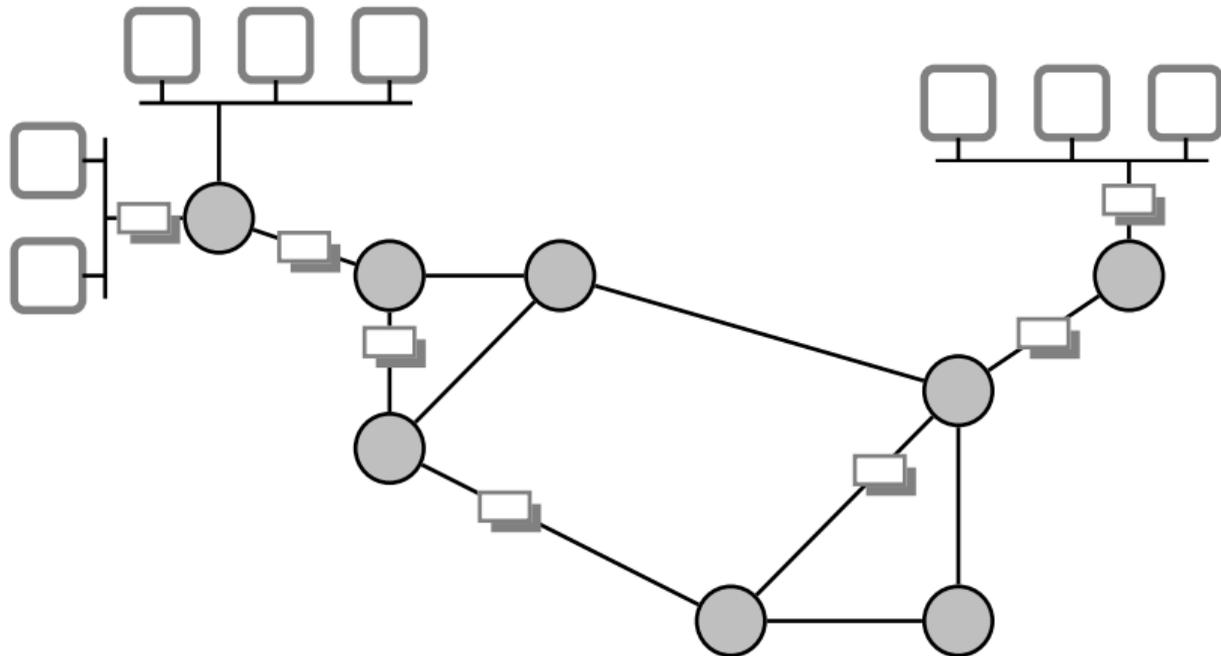
Packet Switching



Packet Switching



Packet Switching



- The Internet is a *packet-switched* network

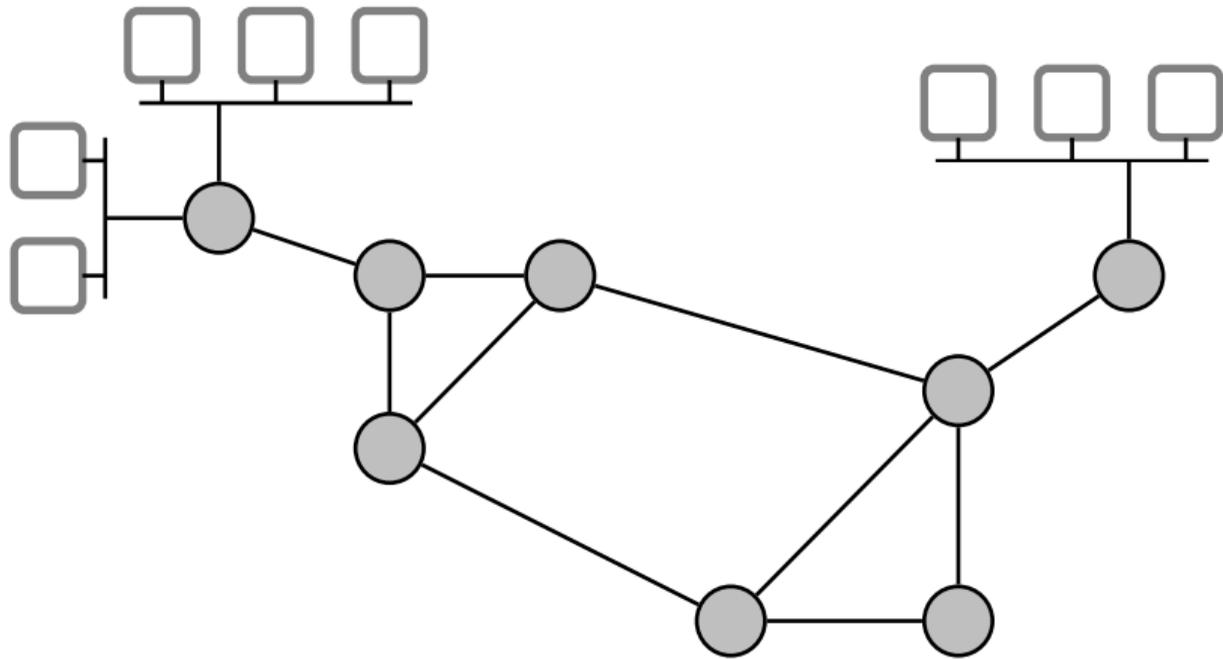
- The Internet is a *packet-switched* network
- Information is transmitted in *packets*

- The Internet is a *packet-switched* network
- Information is transmitted in *packets*
- Switches operate on individual packets

- The Internet is a *packet-switched* network
- Information is transmitted in *packets*
- Switches operate on individual packets
- A switch (router) receives packets and *forwards* them along to other switches or to end systems

- The Internet is a *packet-switched* network
- Information is transmitted in *packets*
- Switches operate on individual packets
- A switch (router) receives packets and *forwards* them along to other switches or to end systems
- Every forwarding decision is taken on the basis of the information contained in the packet

Circuit Switching



- The telephone network is a typical circuit-switched network
 - ▶ not any more, really, but still...

- The telephone network is a typical circuit-switched network
 - ▶ not any more, really, but still...
- Communication requires a ***connection setup*** phase in which the network reserves all the necessary resources for that connection (links, buffers, switches, etc.)

- The telephone network is a typical circuit-switched network
 - ▶ not any more, really, but still...
- Communication requires a **connection setup** phase in which the network reserves all the necessary resources for that connection (links, buffers, switches, etc.)
- After a successful setup, the communicating systems are connected by **a set of links dedicated to the connection** for the entire duration of their conversation

- The telephone network is a typical circuit-switched network
 - ▶ not any more, really, but still...
- Communication requires a **connection setup** phase in which the network reserves all the necessary resources for that connection (links, buffers, switches, etc.)
- After a successful setup, the communicating systems are connected by **a set of links dedicated to the connection** for the entire duration of their conversation
- When the conversation ends, the network tears down the connection, freeing the corresponding resources (links, buffers, etc.) for other connections

Circuit vs. Packet Switching

Circuit vs. Packet Switching



Circuit vs. Packet Switching

Circuit vs. Packet Switching

- Circuit switching requires an expensive setup phase
 - ▶ however, once the connection is established, little or no processing is required

Circuit vs. Packet Switching

- Circuit switching requires an expensive setup phase
 - ▶ however, once the connection is established, little or no processing is required
- Packet switching does not incur any setup cost
 - ▶ however, it always incurs a significant processing and space overhead, on a per-packet basis
 - ▶ *processing cost* for forwarding
 - ▶ *space overhead* because every packet must be self-contained

Circuit vs. Packet Switching (2)

Circuit vs. Packet Switching (2)

- Circuit switching admits a straightforward implementation of quality-of-service guarantees
 - ▶ network resources are reserved at connection setup time

Circuit vs. Packet Switching (2)

- Circuit switching admits a straightforward implementation of quality-of-service guarantees
 - ▶ network resources are reserved at connection setup time
- Guaranteeing any quality of service with packet switching is very difficult
 - ▶ no concept of a “connection”
 - ▶ and again, processing, space overhead, etc.

Circuit vs. Packet Switching (3)

- Circuit switching allows only a limited sharing of communication resources
 - ▶ once a connection is established, the resources are blocked even though there might be long silence periods
 - ▶ i.e., circuit switching is an inefficient way to use the network

Circuit vs. Packet Switching (3)

- Circuit switching allows only a limited sharing of communication resources
 - ▶ once a connection is established, the resources are blocked even though there might be long silence periods
 - ▶ i.e., circuit switching is an inefficient way to use the network
- Packet switching achieves a much better utilization of network resources
 - ▶ it is designed specifically to share links

- Idea: combine the advantages of circuit switching and packet switching

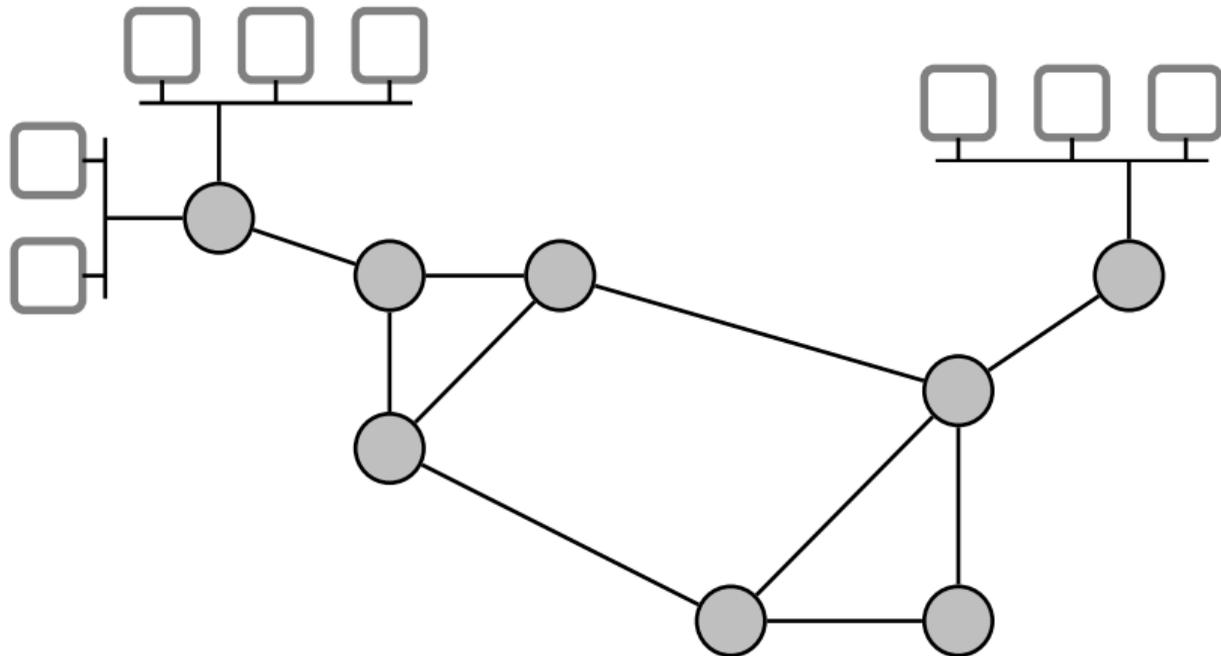
- Idea: combine the advantages of circuit switching and packet switching
- There is a connection setup phase

- Idea: combine the advantages of circuit switching and packet switching
- There is a connection setup phase
- The connection does not create a physical circuit, but rather a “virtual circuit”

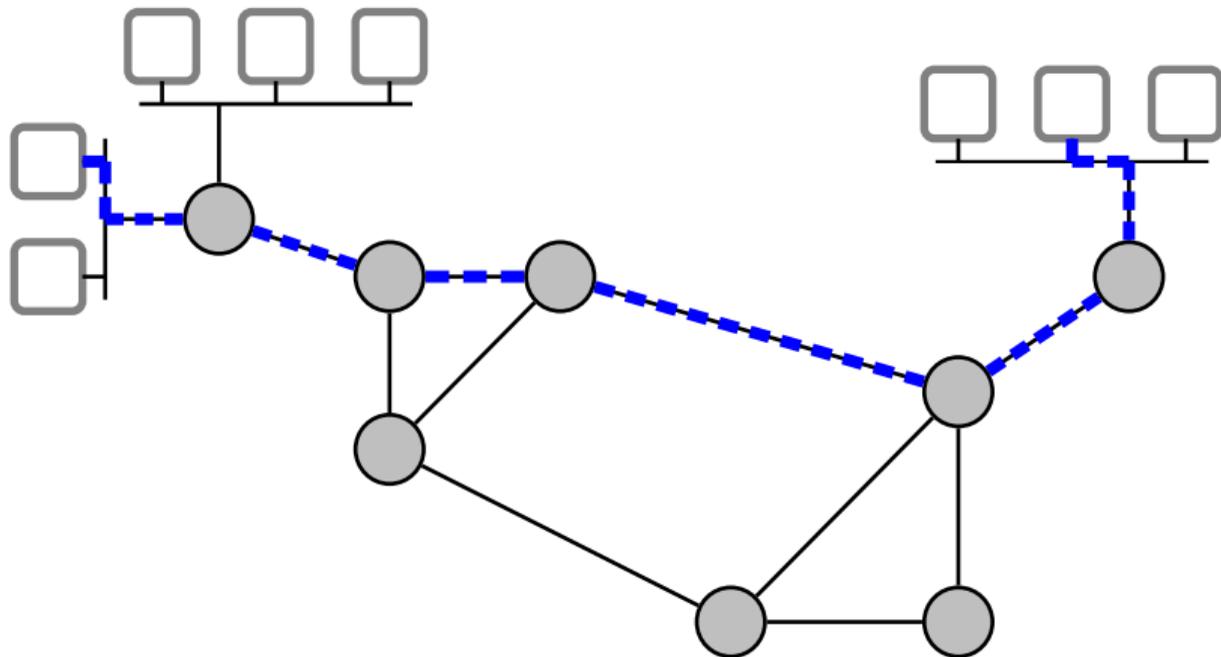
- Idea: combine the advantages of circuit switching and packet switching
- There is a connection setup phase
- The connection does not create a physical circuit, but rather a “virtual circuit”
- Information is sent in packets, so links can be shared more effectively

- Idea: combine the advantages of circuit switching and packet switching
- There is a connection setup phase
- The connection does not create a physical circuit, but rather a “virtual circuit”
- Information is sent in packets, so links can be shared more effectively
- Packets carry a *virtual circuit identifier* instead of the destination address

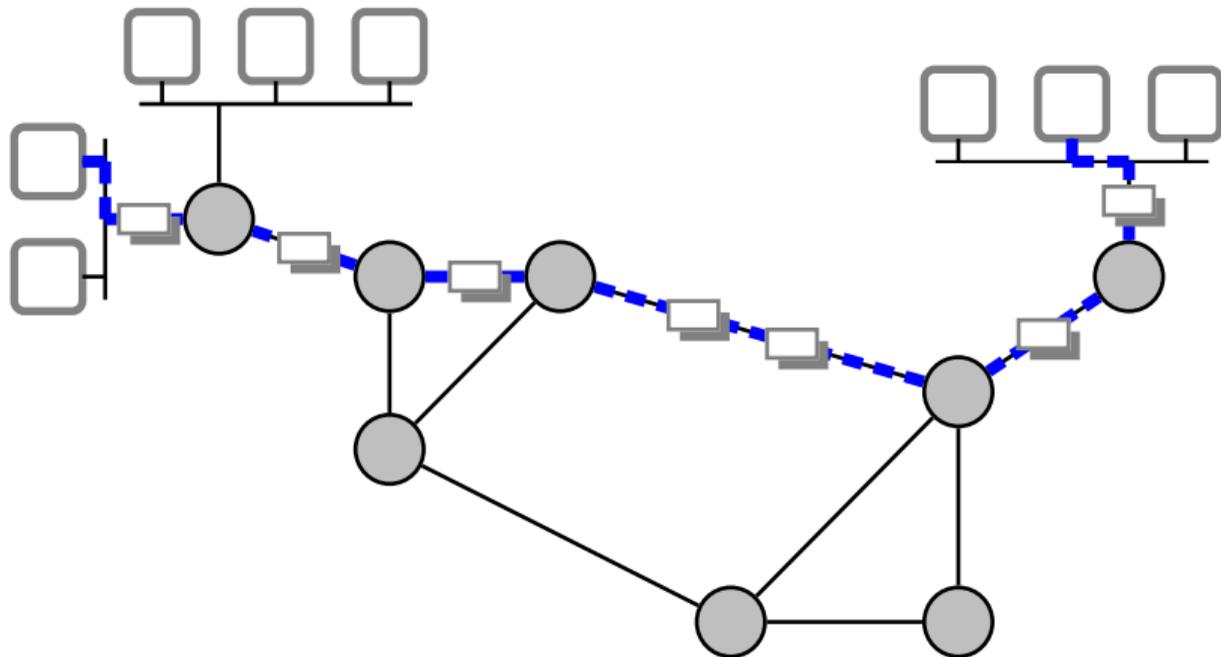
- Idea: combine the advantages of circuit switching and packet switching
- There is a connection setup phase
- The connection does not create a physical circuit, but rather a “virtual circuit”
- Information is sent in packets, so links can be shared more effectively
- Packets carry a *virtual circuit identifier* instead of the destination address
 - ▶ *Important observation:* at any given time there are much fewer *connections* than *destinations*
 - ▶ much faster per-packet processing (forwarding)
 - ▶ lower per-packet space overhead



Virtual Circuit



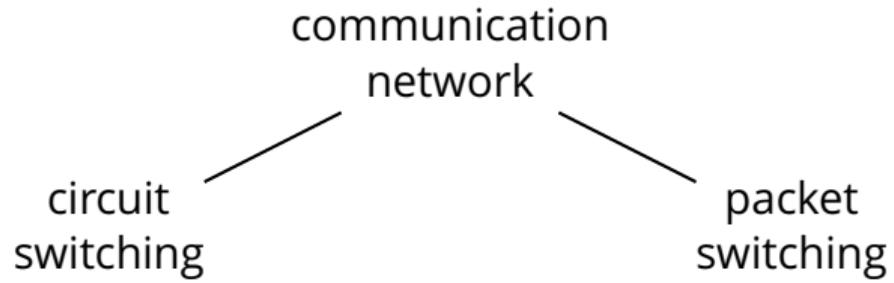
Virtual Circuit



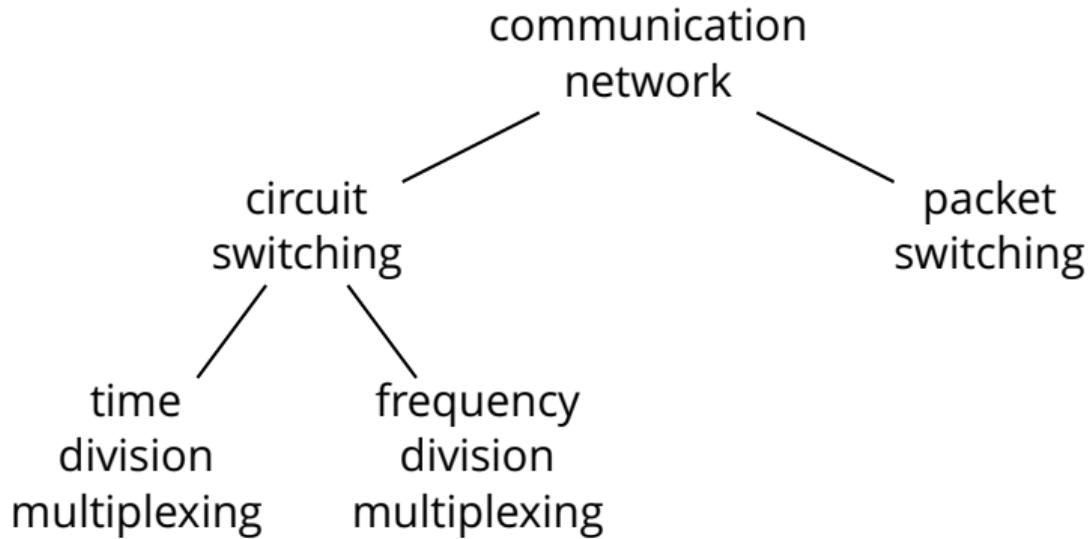
Taxonomy of Networks

communication
network

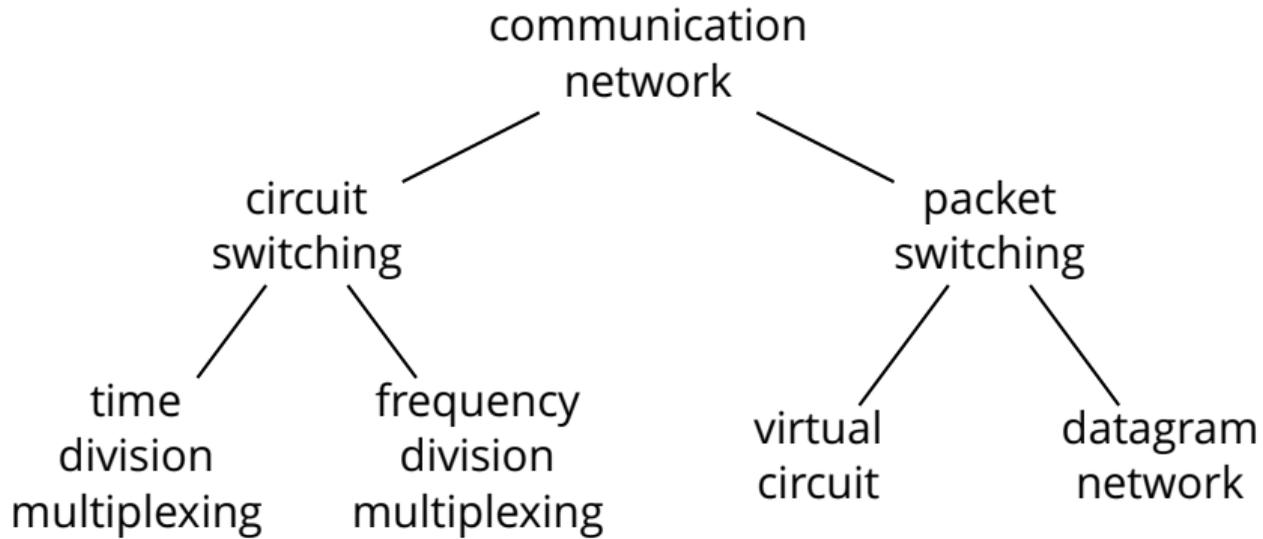
Taxonomy of Networks



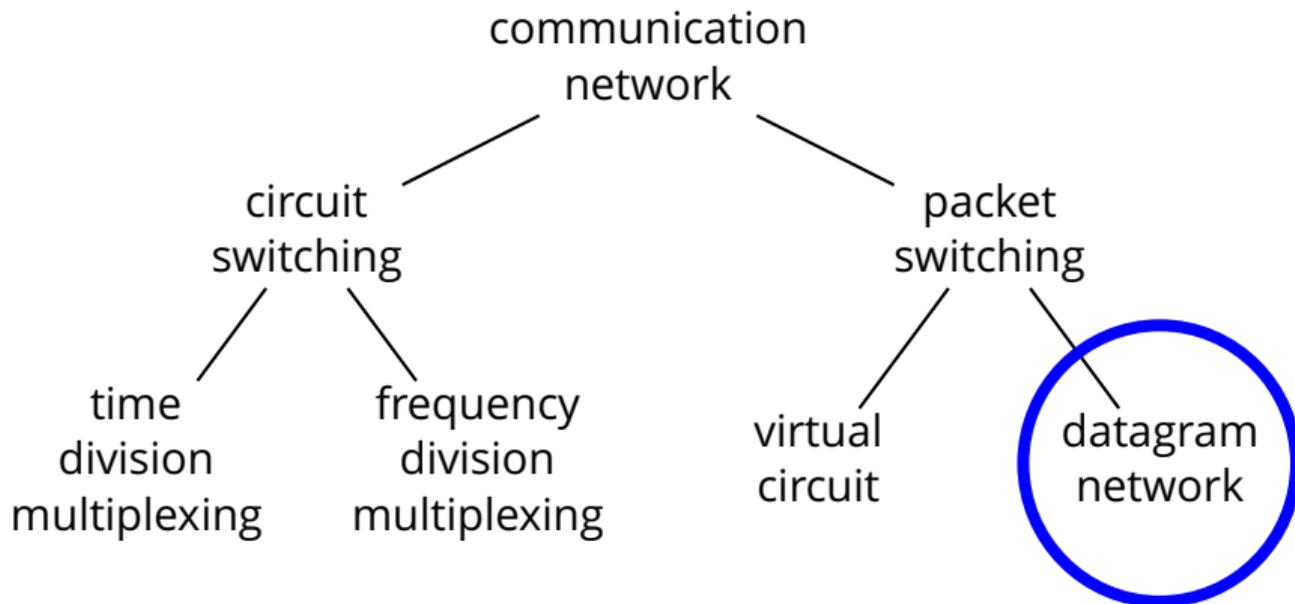
Taxonomy of Networks



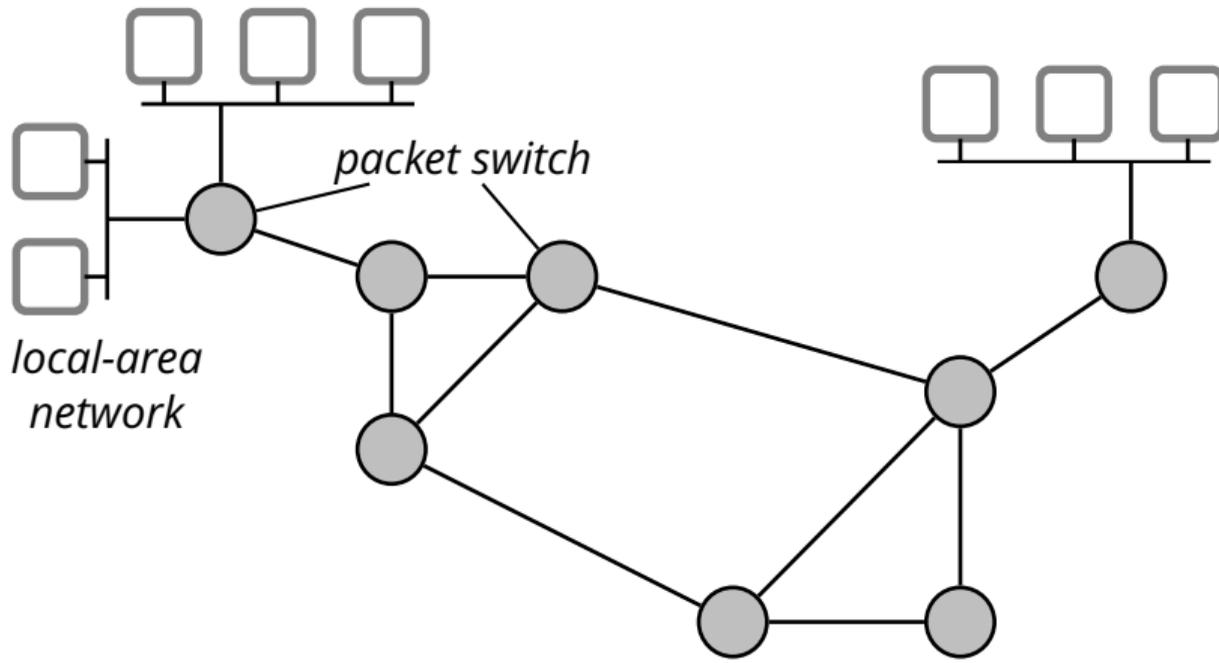
Taxonomy of Networks



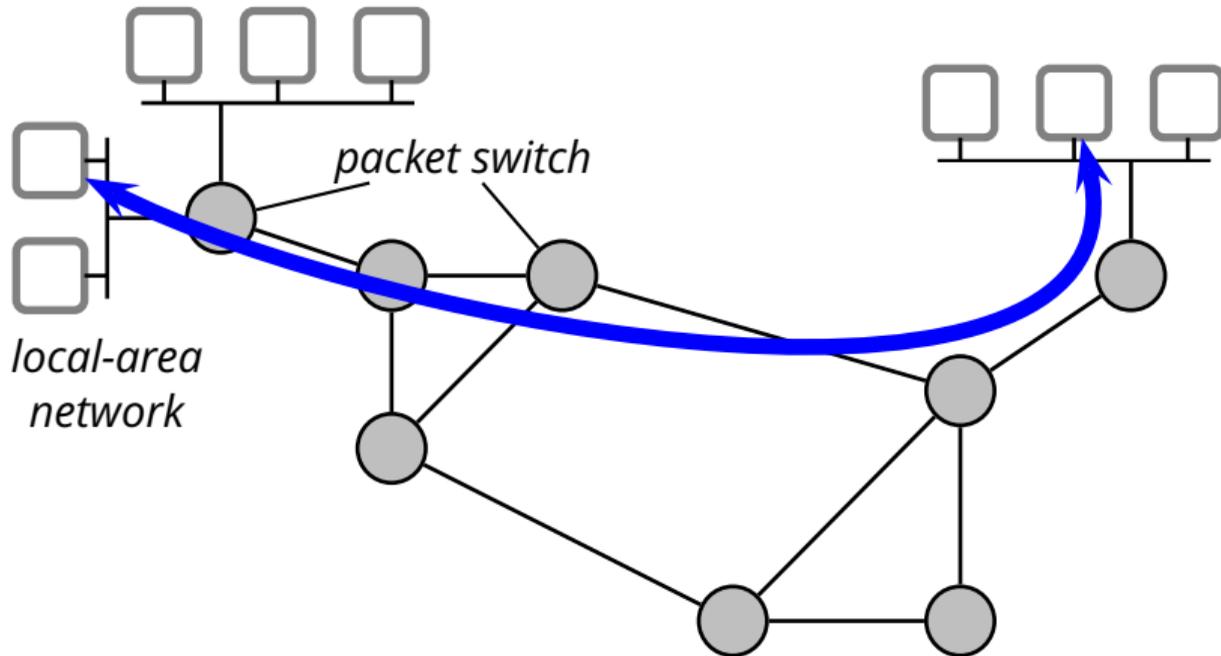
Taxonomy of Networks



Service Perspective



Service Perspective



- What kind of *service* does the Internet offer to end systems?

- Two end systems can communicate through the Internet, but exactly what kind of communication service is that of the Internet?

- Two end systems can communicate through the Internet, but exactly what kind of communication service is that of the Internet?
- **Connectionless, “best effort”**
 - ▶ the network accepts “datagrams” for delivery—this is conceptually similar to the postal service
 - ▶ “best effort” really means *unreliable* though not malicious

- Two end systems can communicate through the Internet, but exactly what kind of communication service is that of the Internet?
- **Connectionless, “best effort”**
 - ▶ the network accepts “datagrams” for delivery—this is conceptually similar to the postal service
 - ▶ “best effort” really means *unreliable* though not malicious
- **Connection-oriented, reliable**
 - ▶ virtual duplex communication channel ($A \leftrightarrow B$)—conceptually similar to a telephone service
 - ▶ information is transmitted “reliably” and in order

- How reliable is a “reliable” service?

- How reliable is a “reliable” service?
- The term “reliable” means that information will eventually reach its destination if a route is viable within a certain amount of time

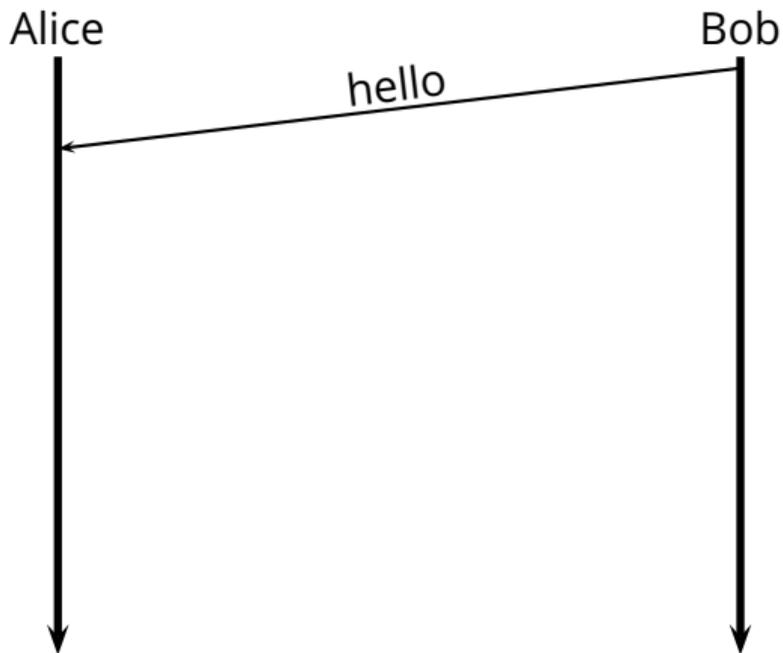
- How reliable is a “reliable” service?
- The term “reliable” means that information will eventually reach its destination if a route is viable within a certain amount of time
- The network makes absolutely no guarantees on *latency* (i.e., the time it takes to transmit some information from a source to a destination)

Communication Protocols

- End systems as well as packet switches run *protocols*. What is a protocol?

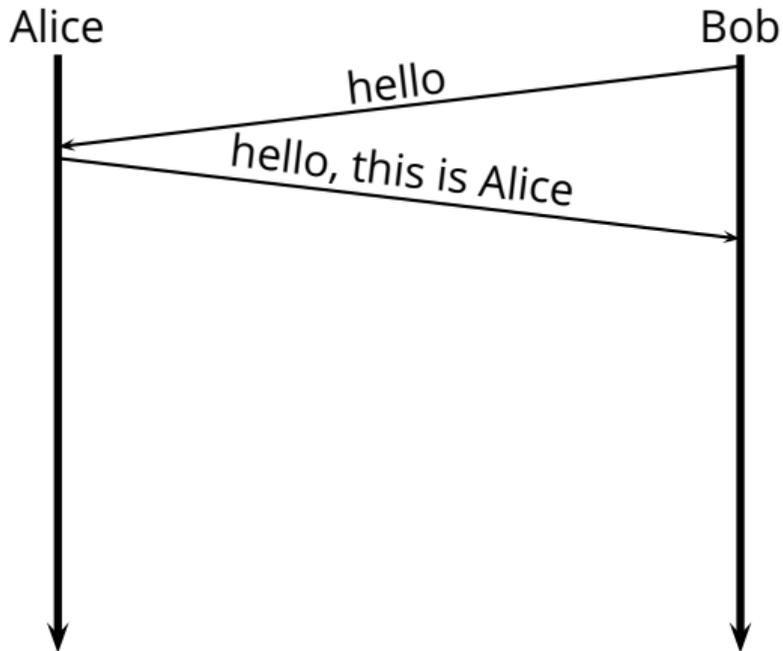
Communication Protocols

- End systems as well as packet switches run *protocols*. What is a protocol?
E.g., let's consider a phone call: Alice calls Bob



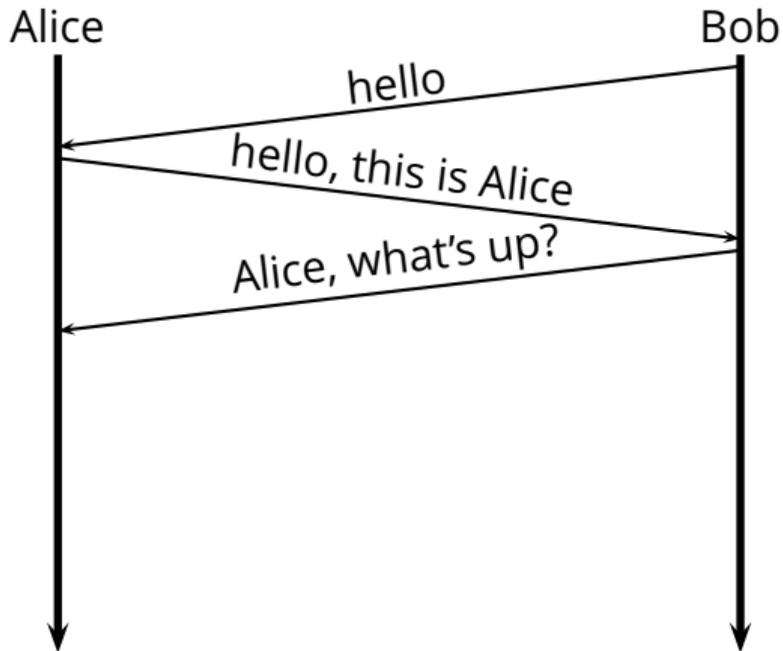
Communication Protocols

- End systems as well as packet switches run *protocols*. What is a protocol?
E.g., let's consider a phone call: Alice calls Bob



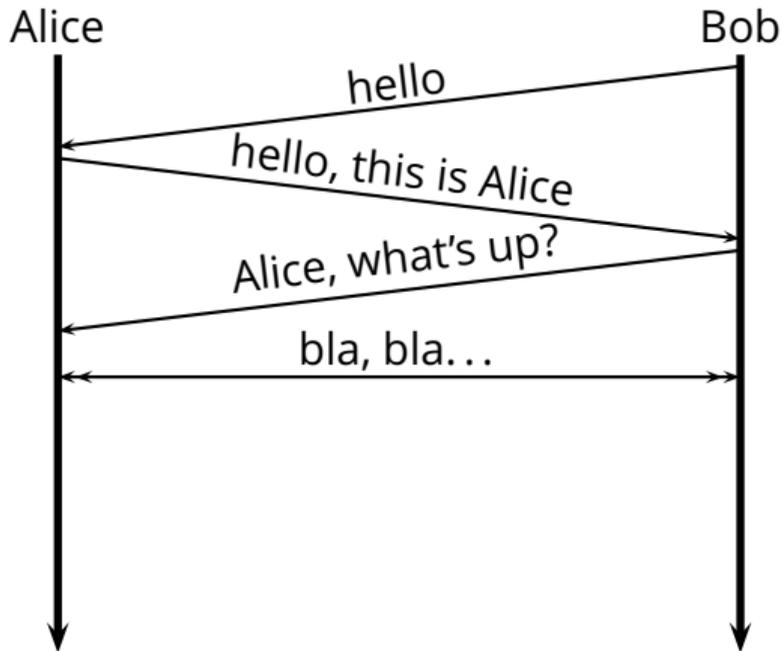
Communication Protocols

- End systems as well as packet switches run *protocols*. What is a protocol?
E.g., let's consider a phone call: Alice calls Bob



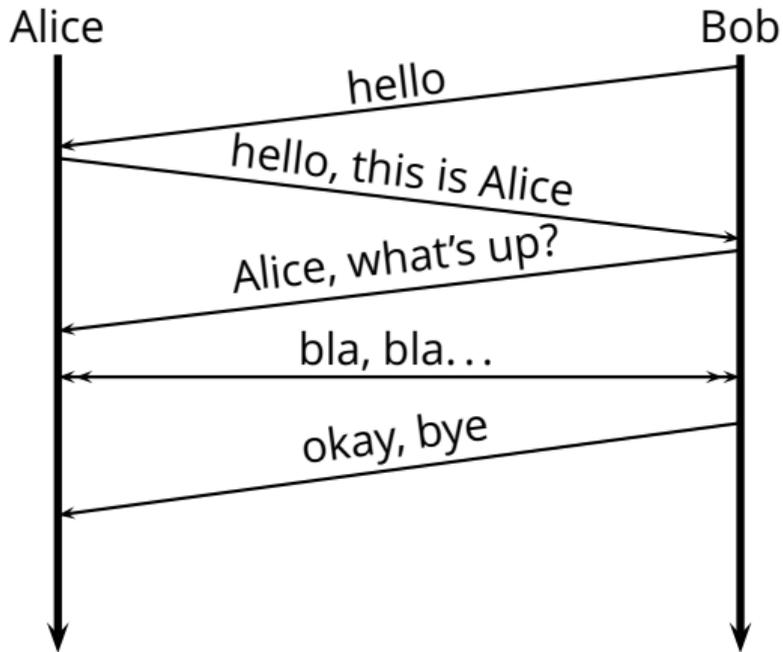
Communication Protocols

- End systems as well as packet switches run *protocols*. What is a protocol?
E.g., let's consider a phone call: Alice calls Bob

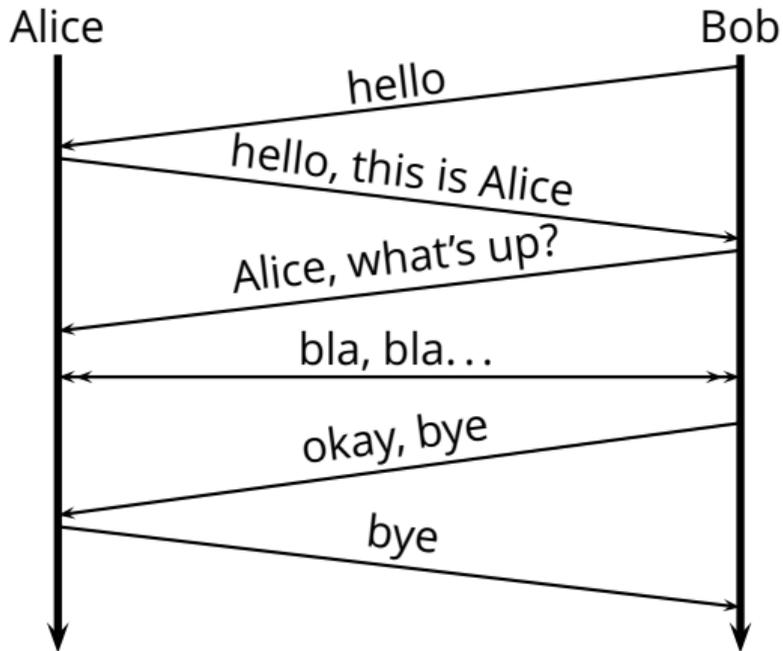


Communication Protocols

- End systems as well as packet switches run *protocols*. What is a protocol?
E.g., let's consider a phone call: Alice calls Bob



- End systems as well as packet switches run *protocols*. What is a protocol?
E.g., let's consider a phone call: Alice calls Bob



■ Phases of the protocol

- ▶ *handshake*: establishes the identities and/or the context
- ▶ *conversation*: free-form exchange
- ▶ *closing*: terminates the conversation

- Phases of the protocol
 - ▶ *handshake*: establishes the identities and/or the context
 - ▶ *conversation*: free-form exchange
 - ▶ *closing*: terminates the conversation
- This protocol assumes a connection-oriented medium
- The protocol involves two parties (Alice and Bob)
- ...

- Another example: air traffic control

- Another example: air traffic control
 - ▶ ...United 971, turn left heading 2-7-0

- Another example: air traffic control
 - ▶ ...United 971, turn left heading 2-7-0
 - ▶ left to 2-7-0, United 971

- Another example: air traffic control
 - ▶ ...United 971, turn left heading 2-7-0
 - ▶ left to 2-7-0, United 971
 - ▶ ...Alitalia 631, contact Malpensa approach at 119.20

- Another example: air traffic control
 - ▶ ...United 971, turn left heading 2-7-0
 - ▶ left to 2-7-0, United 971
 - ▶ ...Alitalia 631, contact Malpensa approach at 119.20
 - ▶ ...

- Another example: air traffic control
 - ▶ ...United 971, turn left heading 2-7-0
 - ▶ left to 2-7-0, United 971
 - ▶ ...Alitalia 631, contact Malpensa approach at 119.20
 - ▶ ...
 - ▶ ...Alitalia 631, contact Malpensa approach at 119.20

- Another example: air traffic control
 - ▶ ...United 971, turn left heading 2-7-0
 - ▶ left to 2-7-0, United 971
 - ▶ ...Alitalia 631, contact Malpensa approach at 119.20
 - ▶ ...
 - ▶ ...Alitalia 631, contact Malpensa approach at 119.20
 - ▶ 1-1-9 point 2-0, Alitalia 631, ciao

■ Another example: air traffic control

- ▶ ...United 971, turn left heading 2-7-0
- ▶ left to 2-7-0, United 971
- ▶ ...Alitalia 631, contact Malpensa approach at 119.20
- ▶ ...
- ▶ ...Alitalia 631, contact Malpensa approach at 119.20
- ▶ 1-1-9 point 2-0, Alitalia 631, ciao
- ▶ ...Center, request, Delta 800

■ Another example: air traffic control

- ▶ ...United 971, turn left heading 2-7-0
- ▶ left to 2-7-0, United 971
- ▶ ...Alitalia 631, contact Malpensa approach at 119.20
- ▶ ...
- ▶ ...Alitalia 631, contact Malpensa approach at 119.20
- ▶ 1-1-9 point 2-0, Alitalia 631, ciao
- ▶ ...Center, request, Delta 800
- ▶ ...United 971, climb and maintain flight level 3-7-0

■ Another example: air traffic control

- ▶ ... United 971, turn left heading 2-7-0
- ▶ left to 2-7-0, United 971
- ▶ ... Alitalia 631, contact Malpensa approach at 119.20
- ▶ ...
- ▶ ... Alitalia 631, contact Malpensa approach at 119.20
- ▶ 1-1-9 point 2-0, Alitalia 631, ciao
- ▶ ... Center, request, Delta 800
- ▶ ... United 971, climb and maintain flight level 3-7-0
- ▶ flight level 3-7-0, United 971

■ Another example: air traffic control

- ▶ ... United 971, turn left heading 2-7-0
- ▶ left to 2-7-0, United 971
- ▶ ... Alitalia 631, contact Malpensa approach at 119.20
- ▶ ...
- ▶ ... Alitalia 631, contact Malpensa approach at 119.20
- ▶ 1-1-9 point 2-0, Alitalia 631, ciao
- ▶ ... Center, request, Delta 800
- ▶ ... United 971, climb and maintain flight level 3-7-0
- ▶ flight level 3-7-0, United 971
- ▶ ... Delta 800, go ahead

■ Another example: air traffic control

- ▶ ... United 971, turn left heading 2-7-0
- ▶ left to 2-7-0, United 971
- ▶ ... Alitalia 631, contact Malpensa approach at 119.20
- ▶ ...
- ▶ ... Alitalia 631, contact Malpensa approach at 119.20
- ▶ 1-1-9 point 2-0, Alitalia 631, ciao
- ▶ ... Center, request, Delta 800
- ▶ ... United 971, climb and maintain flight level 3-7-0
- ▶ flight level 3-7-0, United 971
- ▶ ... Delta 800, go ahead
- ▶ requesting flight level 3-5-0, Delta 800

■ Another example: air traffic control

- ▶ ... United 971, turn left heading 2-7-0
- ▶ left to 2-7-0, United 971
- ▶ ... Alitalia 631, contact Malpensa approach at 119.20
- ▶ ...
- ▶ ... Alitalia 631, contact Malpensa approach at 119.20
- ▶ 1-1-9 point 2-0, Alitalia 631, ciao
- ▶ ... Center, request, Delta 800
- ▶ ... United 971, climb and maintain flight level 3-7-0
- ▶ flight level 3-7-0, United 971
- ▶ ... Delta 800, go ahead
- ▶ requesting flight level 3-5-0, Delta 800
- ▶ Delta 800, unable at the moment

Communication Protocols (2)

- A connectionless protocol

Communication Protocols (2)

- A connectionless protocol
- Multi-party communication

Communication Protocols (2)

- A connectionless protocol
- Multi-party communication
- Medium access control (MAC) protocol

Communication Protocols (2)

- A connectionless protocol
- Multi-party communication
- Medium access control (MAC) protocol
- Interleaved communication

Communication Protocols (2)

- A connectionless protocol
- Multi-party communication
- Medium access control (MAC) protocol
- Interleaved communication
- Acknowledgements

Communication Protocols (2)

- A connectionless protocol
- Multi-party communication
- Medium access control (MAC) protocol
- Interleaved communication
- Acknowledgements
- Timeout and retransmission

Communication Protocols (2)

- A connectionless protocol
- Multi-party communication
- Medium access control (MAC) protocol
- Interleaved communication
- Acknowledgements
- Timeout and retransmission
- “Master” role

Communication Protocols (3)

- Let's revisit the phone-call protocol

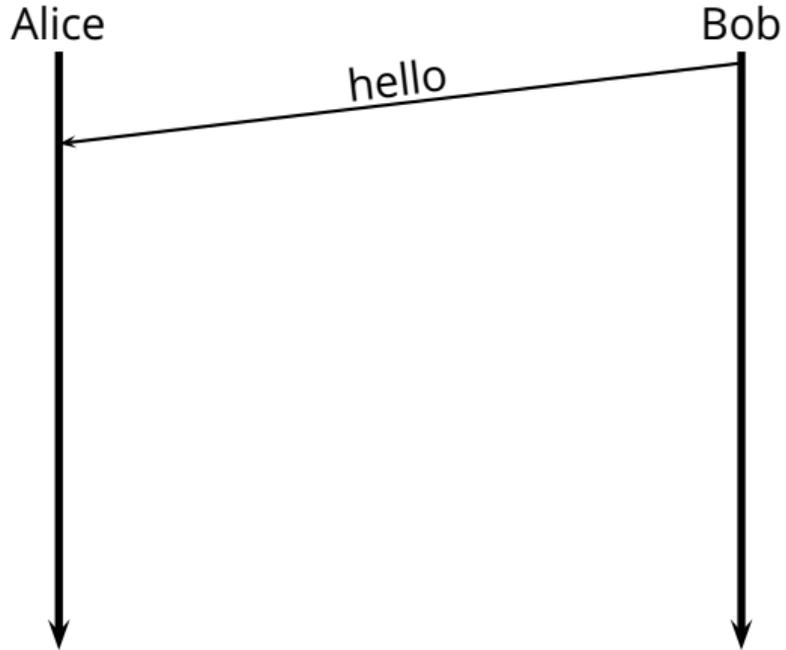
Alice



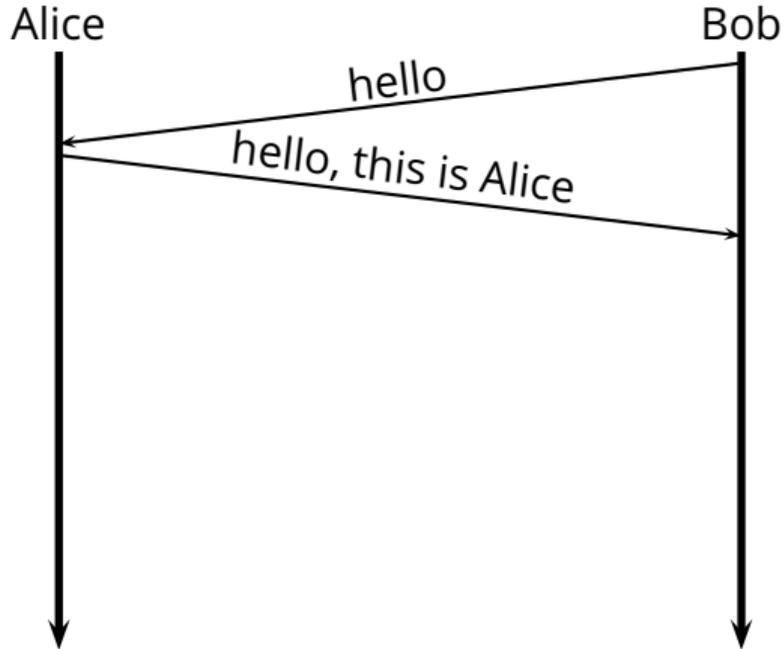
Bob



- Let's revisit the phone-call protocol

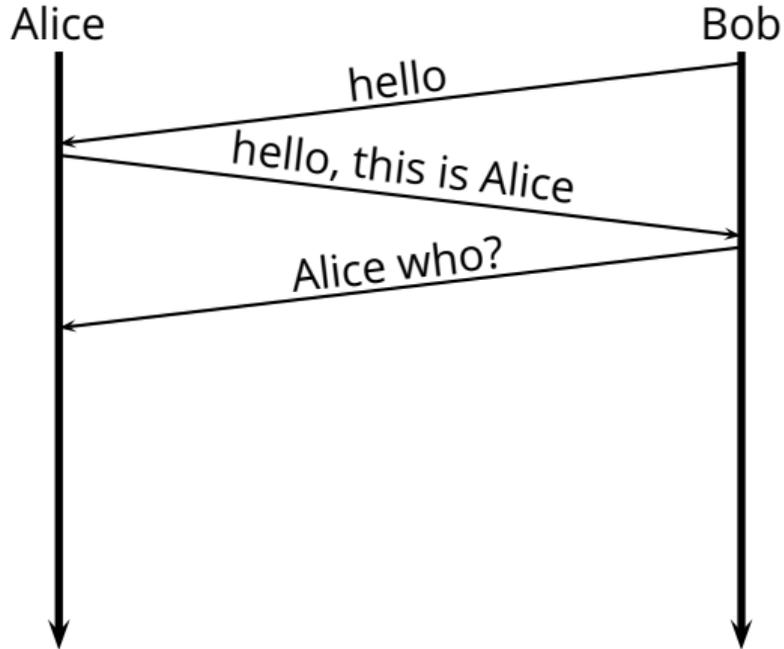


- Let's revisit the phone-call protocol

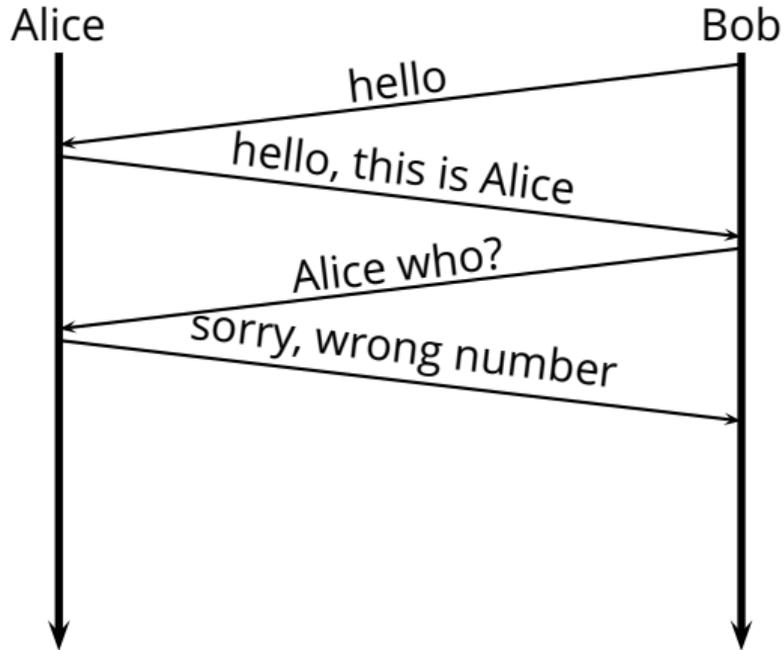


Communication Protocols (3)

- Let's revisit the phone-call protocol



- Let's revisit the phone-call protocol



Communication Protocols (4)

- Another run of the phone-call protocol

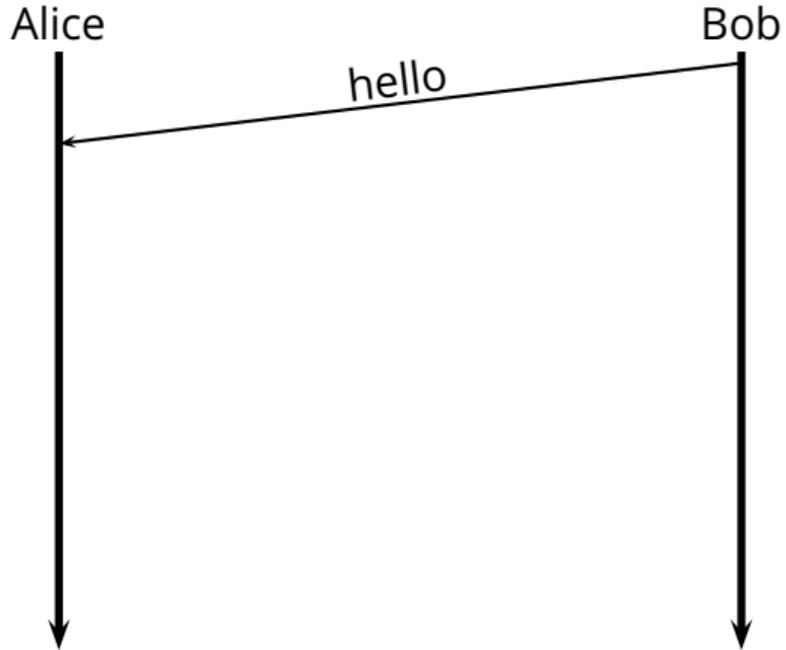
Alice



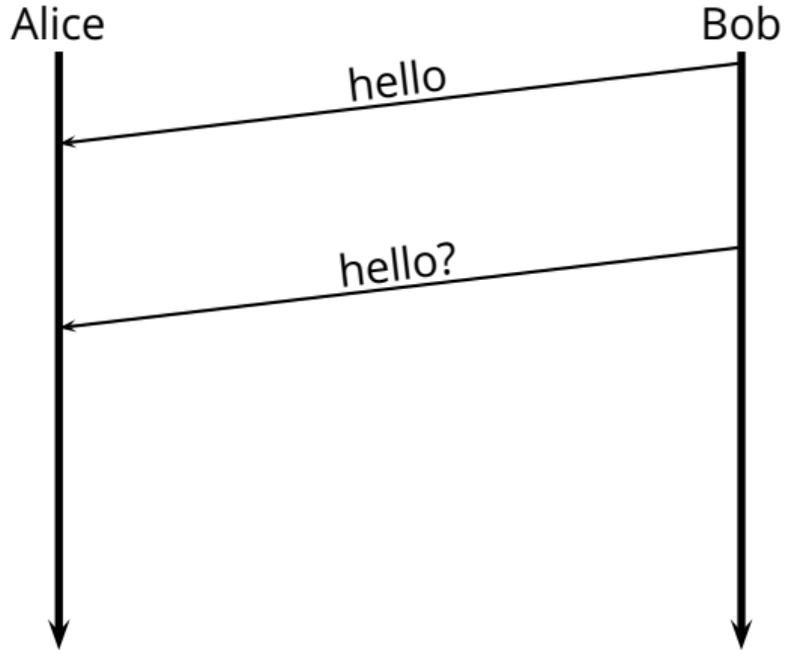
Bob



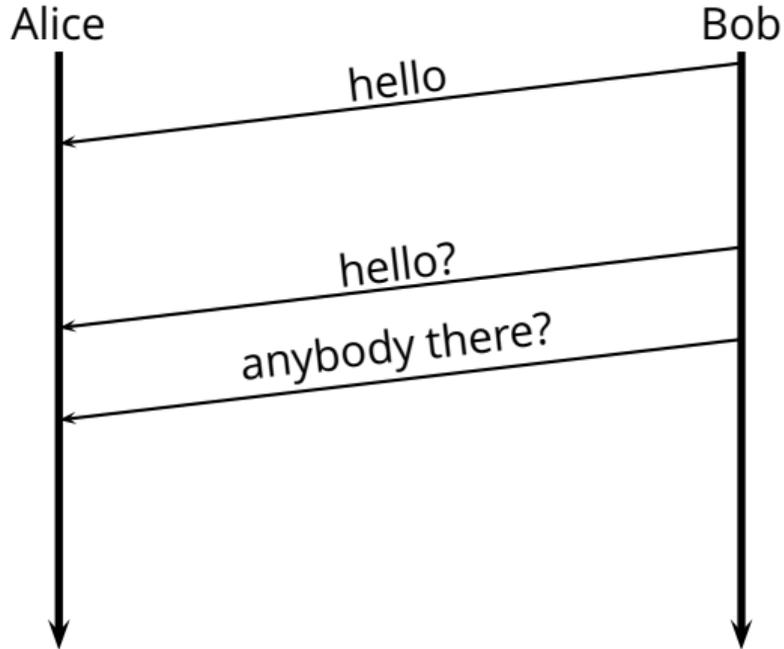
- Another run of the phone-call protocol



- Another run of the phone-call protocol



- Another run of the phone-call protocol



Communication Protocols: Principles

- A protocol is a lot like a program
 - ▶ in fact, it is a *distributed program*, where different processes can send messages to each other

Communication Protocols: Principles

- A protocol is a lot like a program
 - ▶ in fact, it is a *distributed program*, where different processes can send messages to each other
- It is an *executable* specification

Communication Protocols: Principles

- A protocol is a lot like a program
 - ▶ in fact, it is a *distributed program*, where different processes can send messages to each other
- It is an *executable* specification
- It must be *unambiguous*

Communication Protocols: Principles

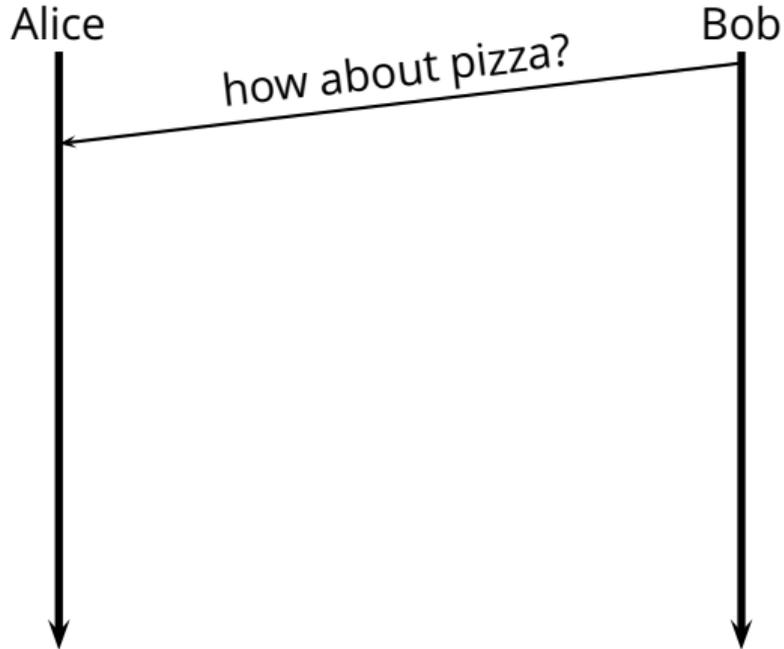
- A protocol is a lot like a program
 - ▶ in fact, it is a *distributed program*, where different processes can send messages to each other
- It is an *executable* specification
- It must be *unambiguous*
- It must be *complete*
 - ▶ i.e., it must include actions and/or responses for all possible situations and all possible messages

Communication Protocols: Principles

- A protocol is a lot like a program
 - ▶ in fact, it is a *distributed program*, where different processes can send messages to each other
- It is an *executable* specification
- It must be *unambiguous*
- It must be *complete*
 - ▶ i.e., it must include actions and/or responses for all possible situations and all possible messages
- A network protocol must also define all the necessary *message formats*

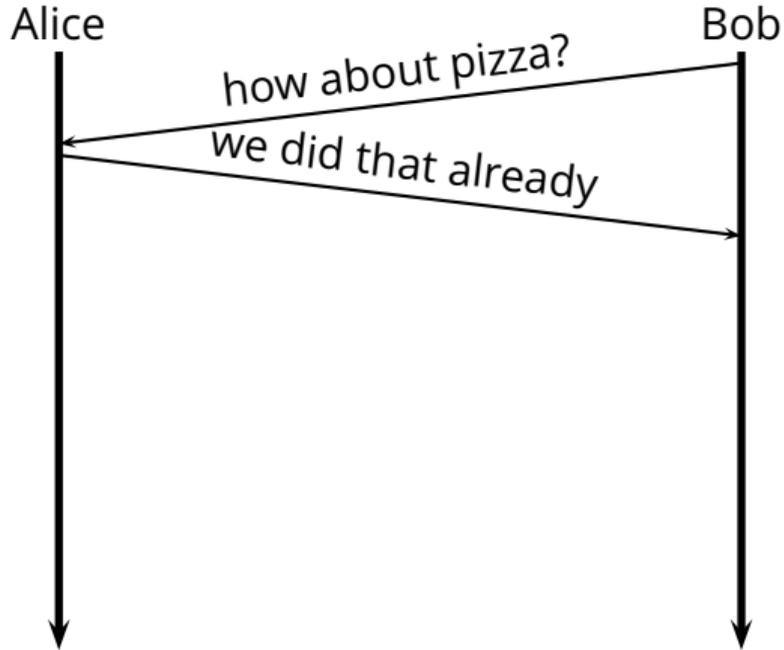
- Another protocol: deciding where to go for dinner

- Another protocol: deciding where to go for dinner

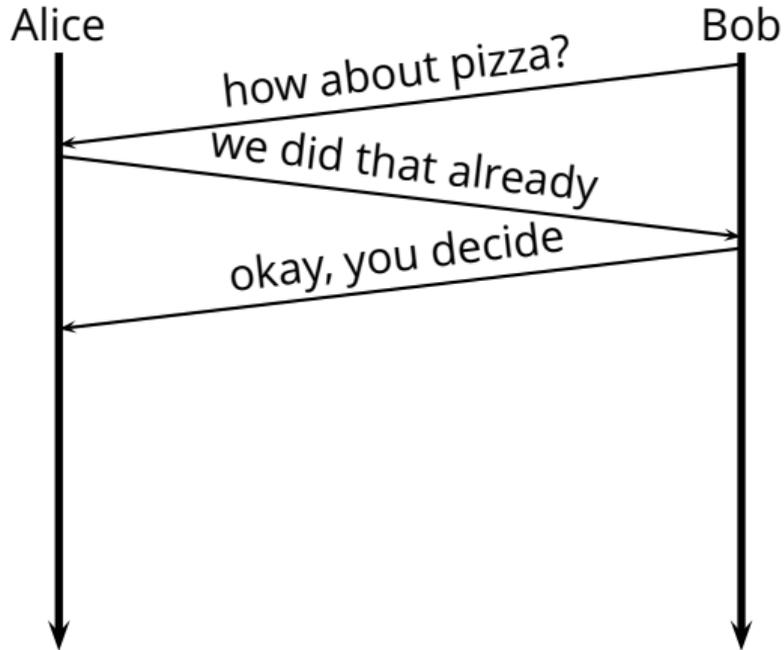


Communication Protocols (5)

- Another protocol: deciding where to go for dinner

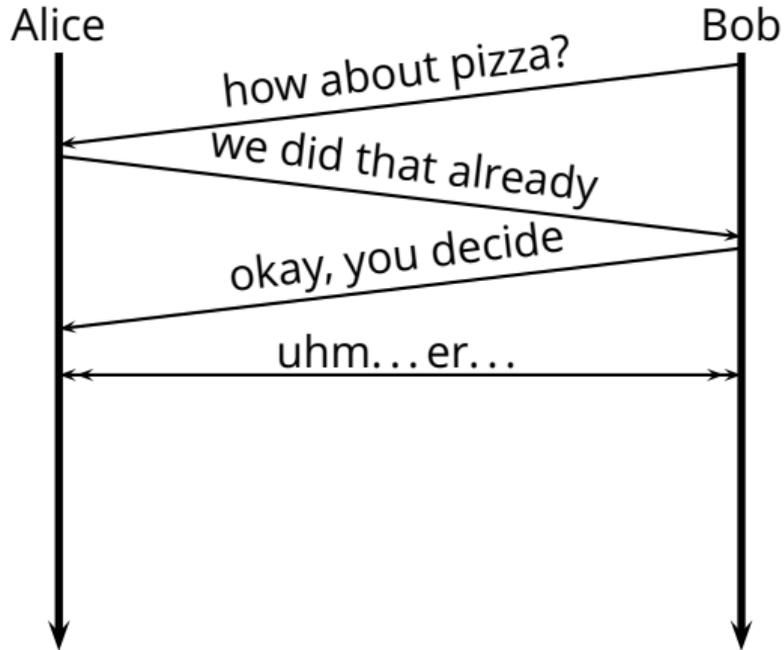


- Another protocol: deciding where to go for dinner



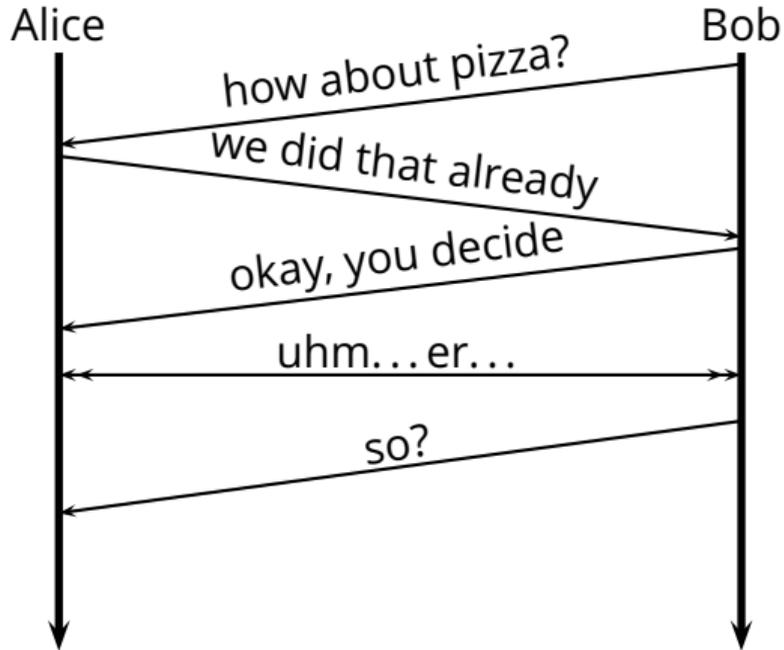
Communication Protocols (5)

- Another protocol: deciding where to go for dinner

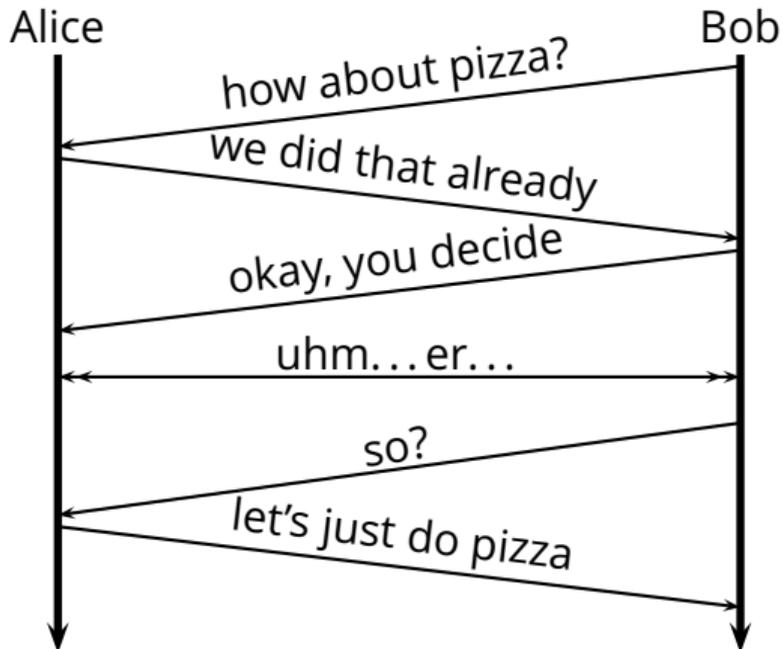


Communication Protocols (5)

- Another protocol: deciding where to go for dinner



- Another protocol: deciding where to go for dinner



- Alice calls Bob to decide where to go for dinner

- Alice calls Bob to decide where to go for dinner

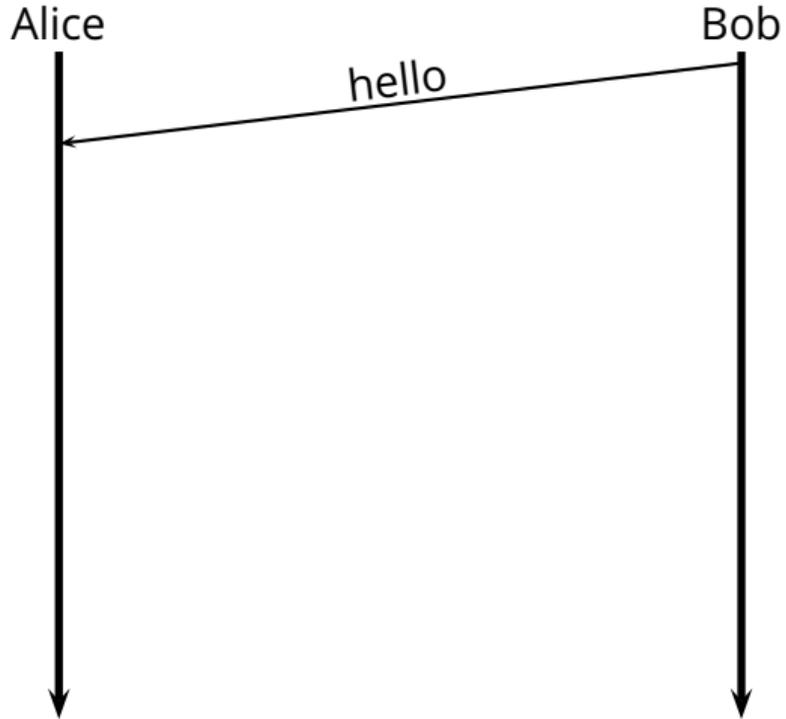
Alice



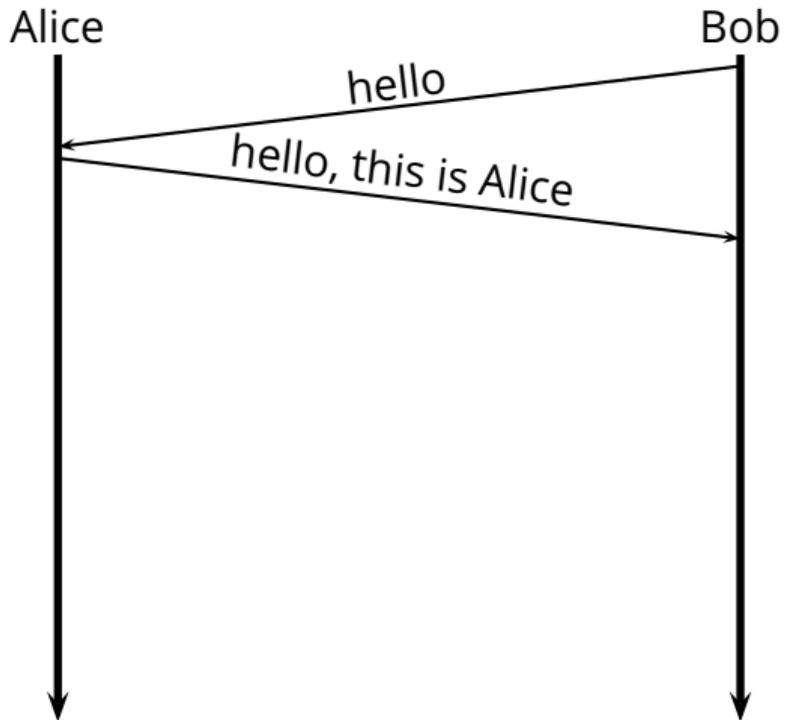
Bob



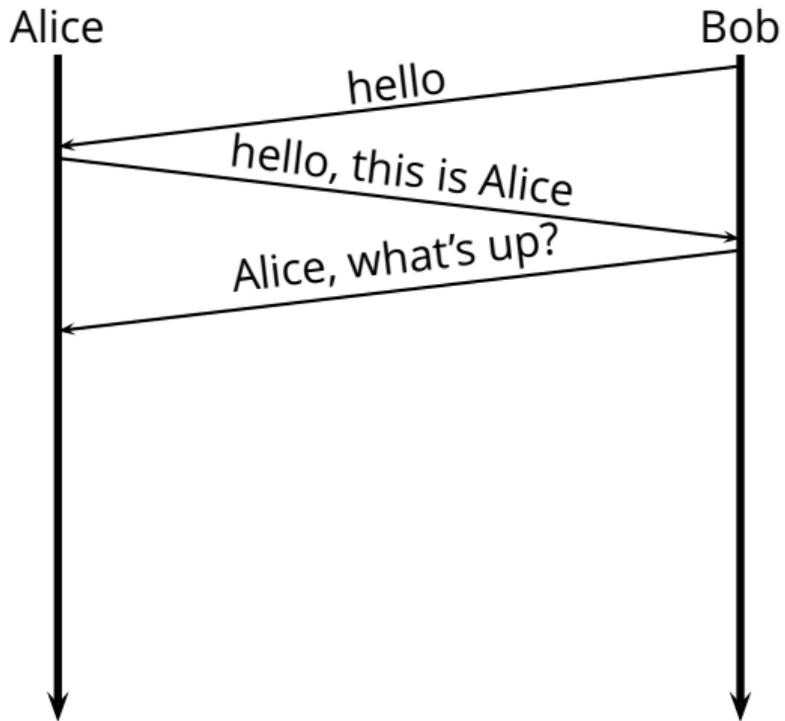
- Alice calls Bob to decide where to go for dinner



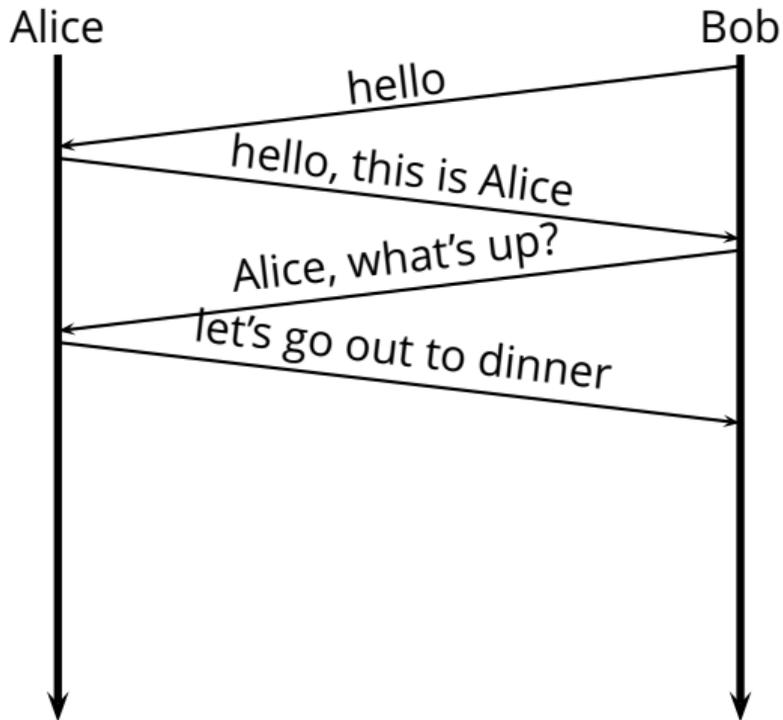
- Alice calls Bob to decide where to go for dinner



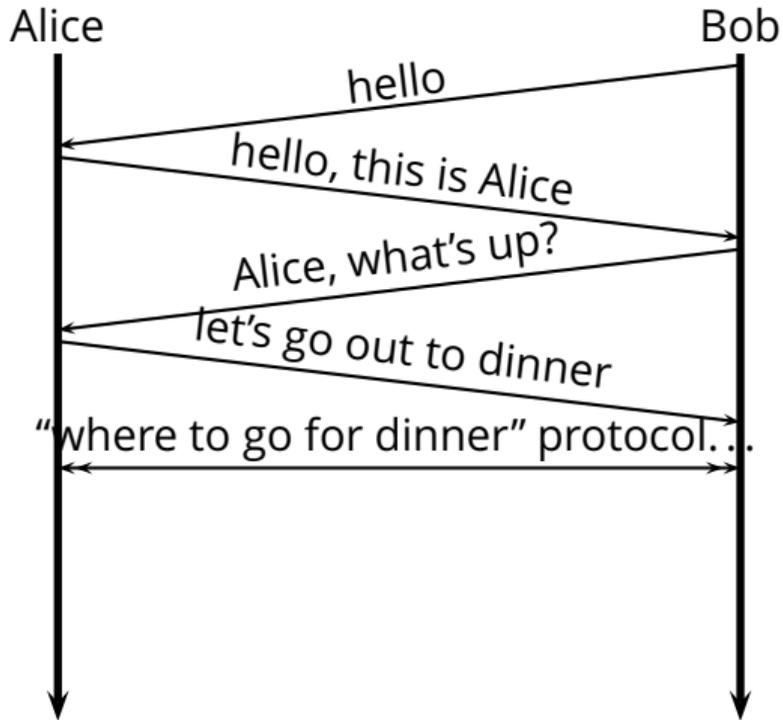
- Alice calls Bob to decide where to go for dinner



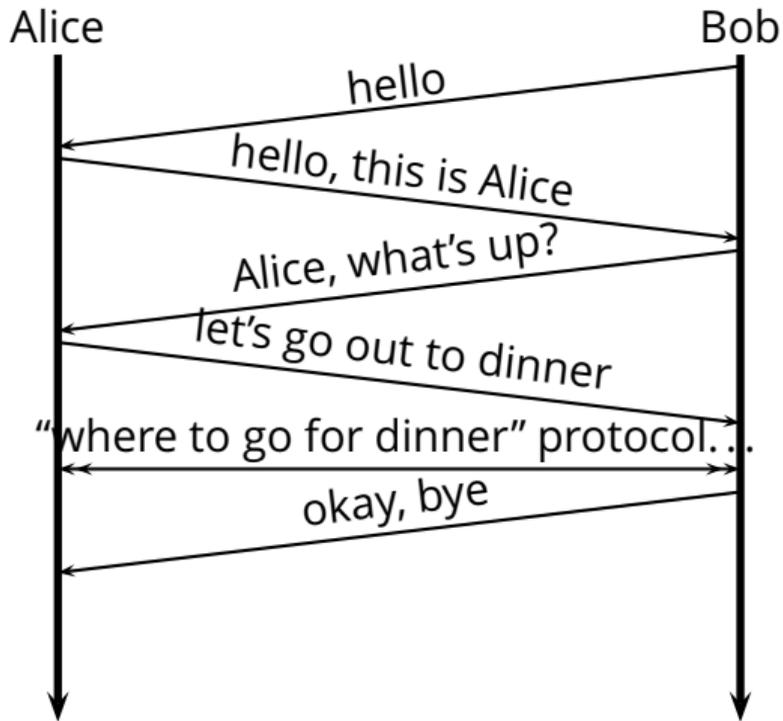
- Alice calls Bob to decide where to go for dinner



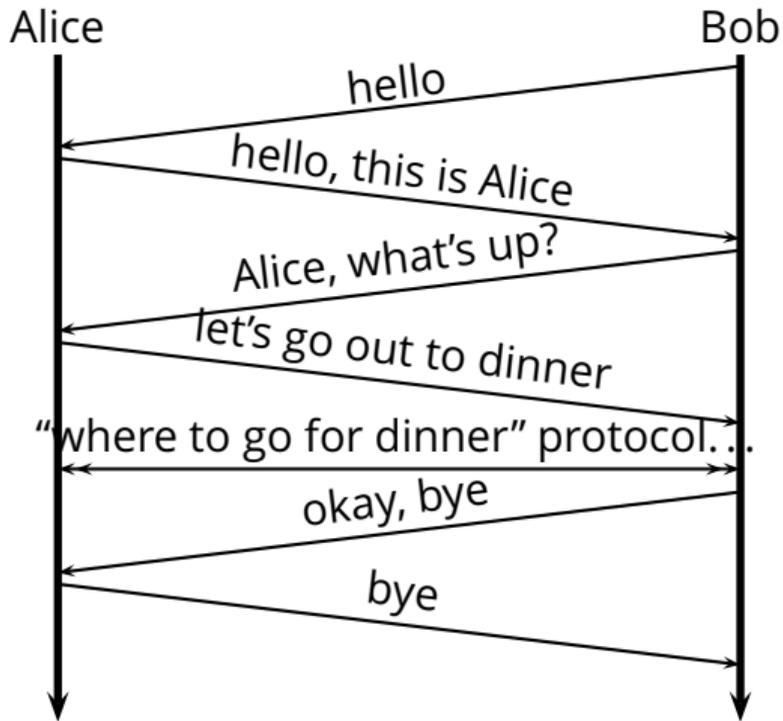
- Alice calls Bob to decide where to go for dinner



- Alice calls Bob to decide where to go for dinner

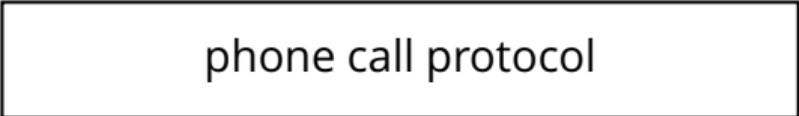


- Alice calls Bob to decide where to go for dinner



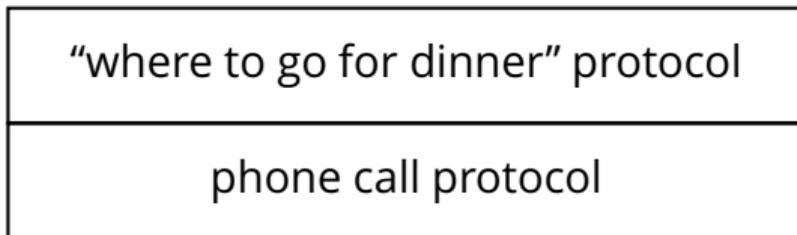
- Alice calls Bob to decide where to go for dinner

- Alice calls Bob to decide where to go for dinner

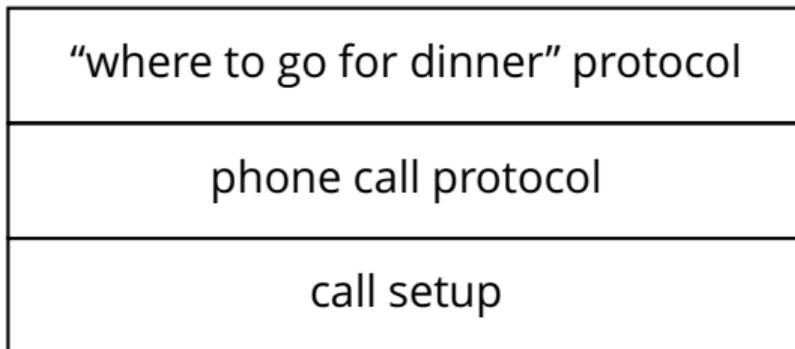


phone call protocol

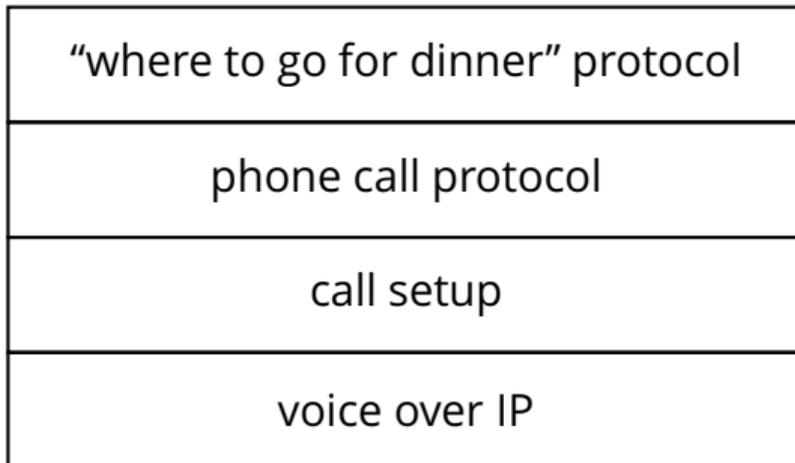
- Alice calls Bob to decide where to go for dinner



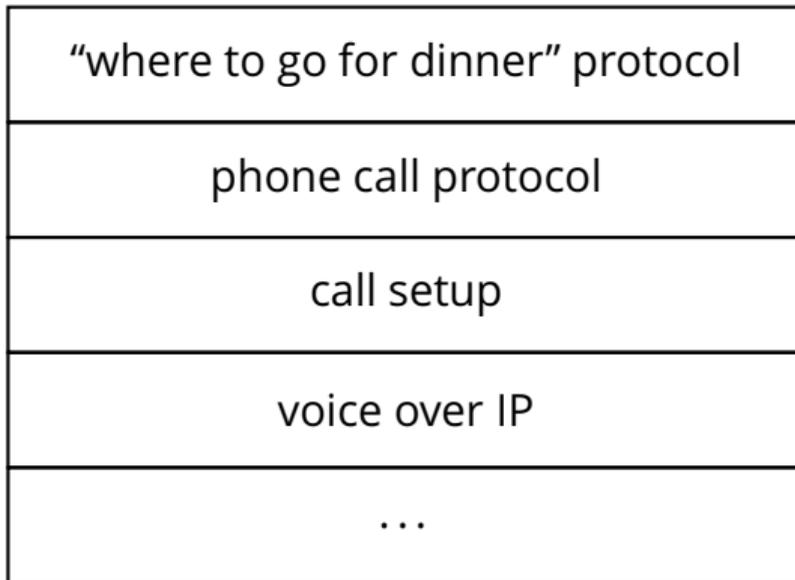
- Alice calls Bob to decide where to go for dinner



- Alice calls Bob to decide where to go for dinner



- Alice calls Bob to decide where to go for dinner



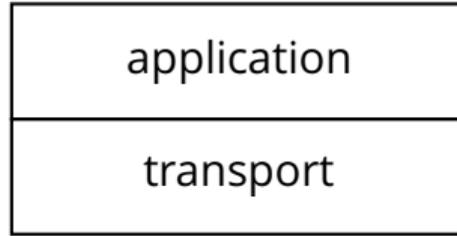
Internet Protocol Stack

Internet Protocol Stack

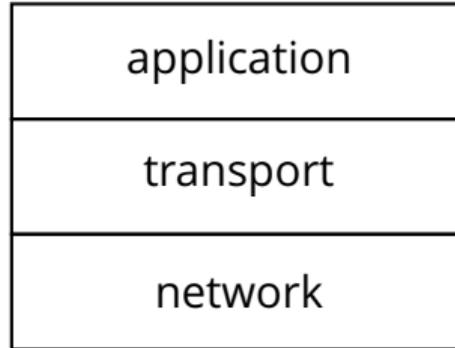


application

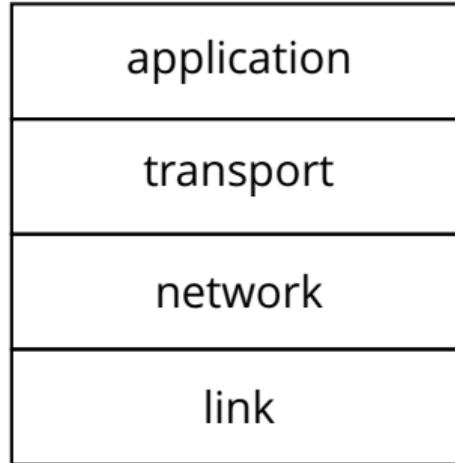
Internet Protocol Stack



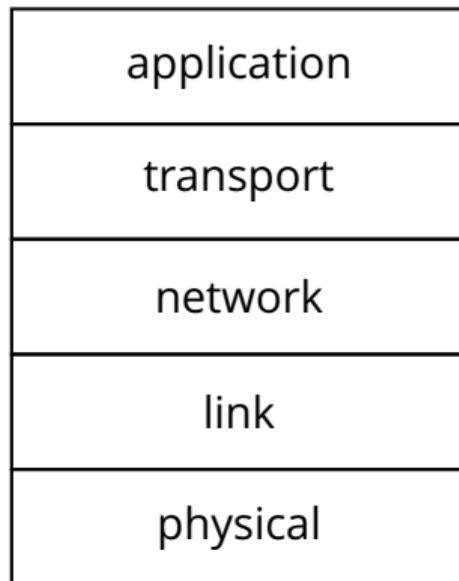
Internet Protocol Stack



Internet Protocol Stack



Internet Protocol Stack



- *Application* (e.g., HTTP, SMTP, and DNS)
 - ▶ application functionalities
 - ▶ application messages

- *Application* (e.g., HTTP, SMTP, and DNS)
 - ▶ application functionalities
 - ▶ application messages
- *Transport* (e.g., TCP and UDP)
 - ▶ application multiplexing, reliable transfer (TCP), congestion control (TCP)
 - ▶ datagrams (UDP) or segments (TCP)

- *Application* (e.g., HTTP, SMTP, and DNS)
 - ▶ application functionalities
 - ▶ application messages
- *Transport* (e.g., TCP and UDP)
 - ▶ application multiplexing, reliable transfer (TCP), congestion control (TCP)
 - ▶ datagrams (UDP) or segments (TCP)
- *Network* (IP)
 - ▶ end to end datagram, best-effort service, routing, fragmentation
 - ▶ packets (IP)

- *Application* (e.g., HTTP, SMTP, and DNS)
 - ▶ application functionalities
 - ▶ application messages
- *Transport* (e.g., TCP and UDP)
 - ▶ application multiplexing, reliable transfer (TCP), congestion control (TCP)
 - ▶ datagrams (UDP) or segments (TCP)
- *Network* (IP)
 - ▶ end to end datagram, best-effort service, routing, fragmentation
 - ▶ packets (IP)
- *Link* (e.g., Ethernet and PPP)
 - ▶ point-to-point or local broadcast communication
 - ▶ frames (or packets)

- *Application* (e.g., HTTP, SMTP, and DNS)
 - ▶ application functionalities
 - ▶ application messages
- *Transport* (e.g., TCP and UDP)
 - ▶ application multiplexing, reliable transfer (TCP), congestion control (TCP)
 - ▶ datagrams (UDP) or segments (TCP)
- *Network* (IP)
 - ▶ end to end datagram, best-effort service, routing, fragmentation
 - ▶ packets (IP)
- *Link* (e.g., Ethernet and PPP)
 - ▶ point-to-point or local broadcast communication
 - ▶ frames (or packets)
- *Physical*