

Basic Concepts In Computer Networking

Antonio Carzaniga

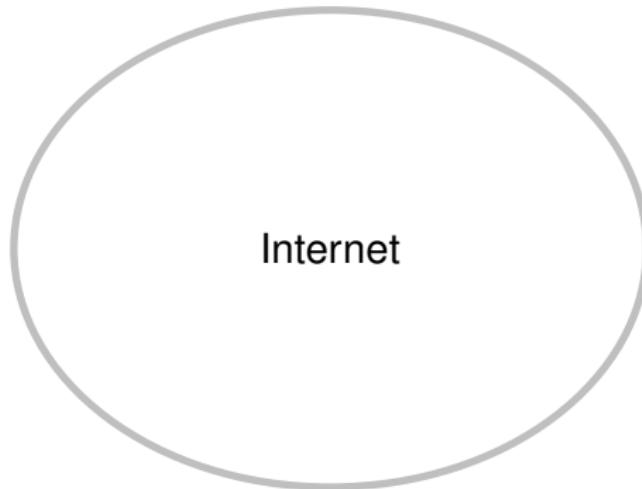
Faculty of Informatics
University of Lugano

March 11, 2005

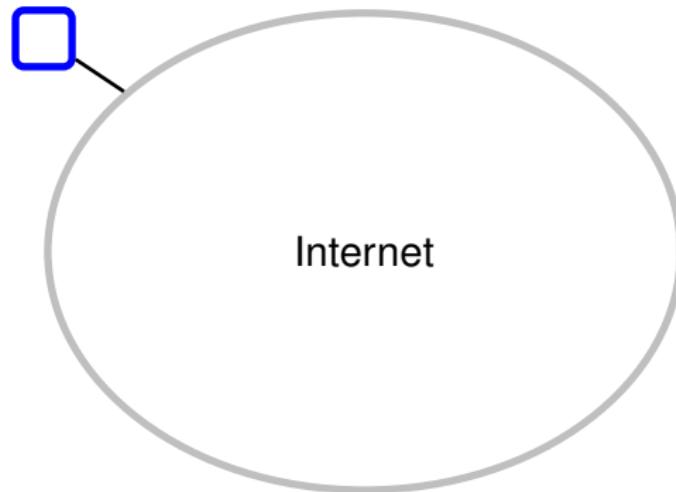
Outline

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

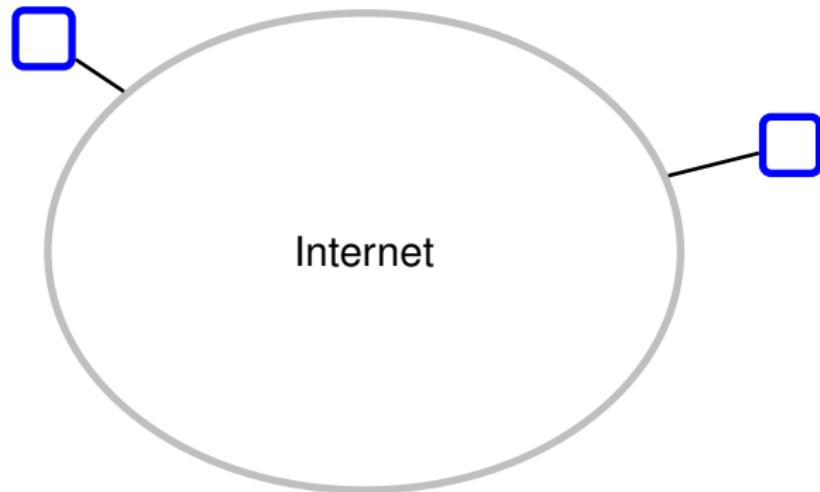
What is the Internet?



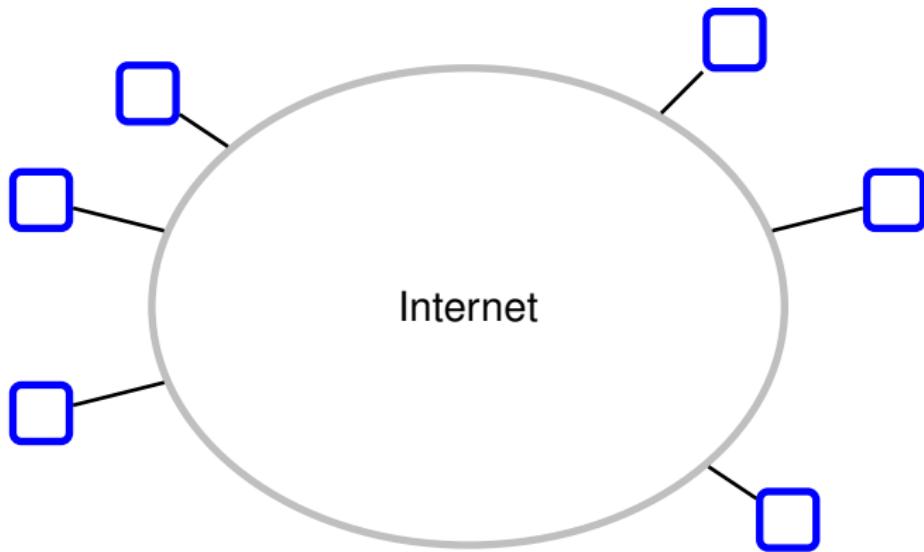
What is the Internet?



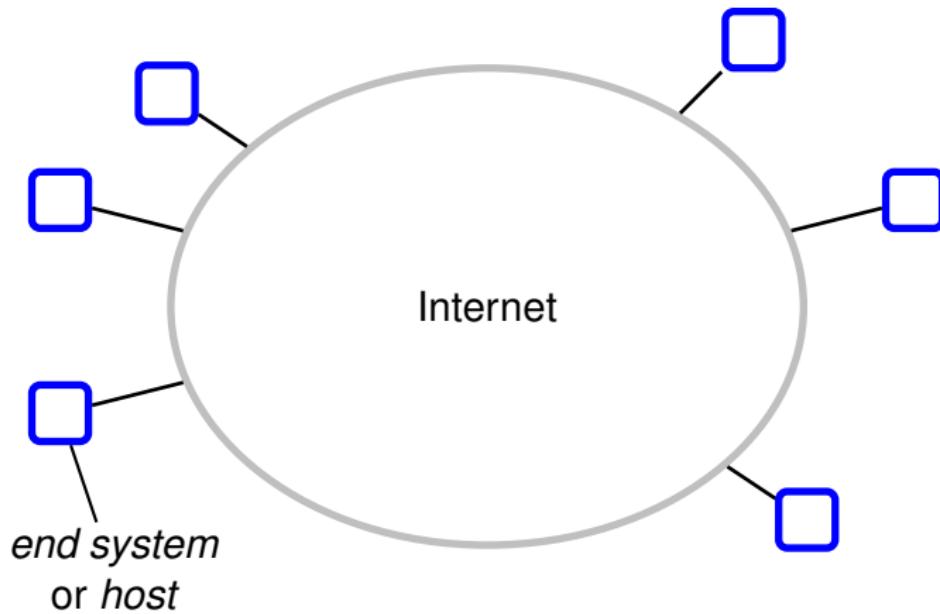
What is the Internet?



What is the Internet?



What is the Internet?



End Systems

- ◆ *end system or host* (□)

End Systems

- ◆ *end system or host* (□)
 - ▶ a computer

End Systems

- ◆ *end system or host* (□)
 - ▶ a computer
 - ▶ a server (well, that would also be a computer)

End Systems

- ◆ *end system or host* (□)

- ▶ a computer
- ▶ a server (well, that would also be a computer)
- ▶ a camera (a.k.a., webcam)

End Systems

- ◆ *end system or host* (□)

- ▶ a computer
- ▶ a server (well, that would also be a computer)
- ▶ a camera (a.k.a., webcam)
- ▶ a temperature sensor

End Systems

- ◆ *end system or host* (□)

- ▶ a computer
- ▶ a server (well, that would also be a computer)
- ▶ a camera (a.k.a., webcam)
- ▶ a temperature sensor
- ▶ a cell phone

End Systems

- ◆ *end system or host* (□)

- ▶ a computer
- ▶ a server (well, that would also be a computer)
- ▶ a camera (a.k.a., webcam)
- ▶ a temperature sensor
- ▶ a cell phone
- ▶ a PDA

End Systems

- ◆ *end system or host* (□)

- ▶ a computer
- ▶ a server (well, that would also be a computer)
- ▶ a camera (a.k.a., webcam)
- ▶ a temperature sensor
- ▶ a cell phone
- ▶ a PDA
- ▶ ...
- ▶ a car
- ▶ a television set
- ▶ a picture frame
- ▶ a toaster
- ▶ ...

End Systems

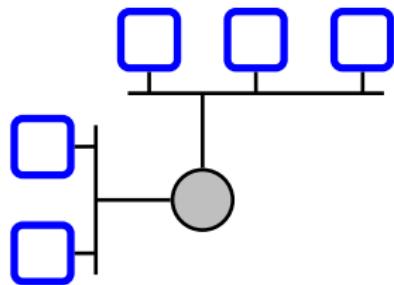
- ◆ *end system or host* (□)

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

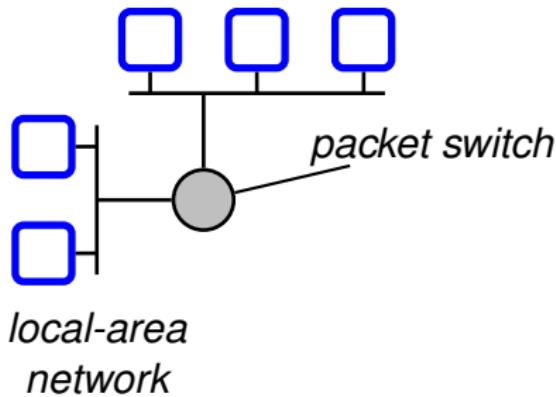
What is “Inside” the Internet?

-
-
-
-
-

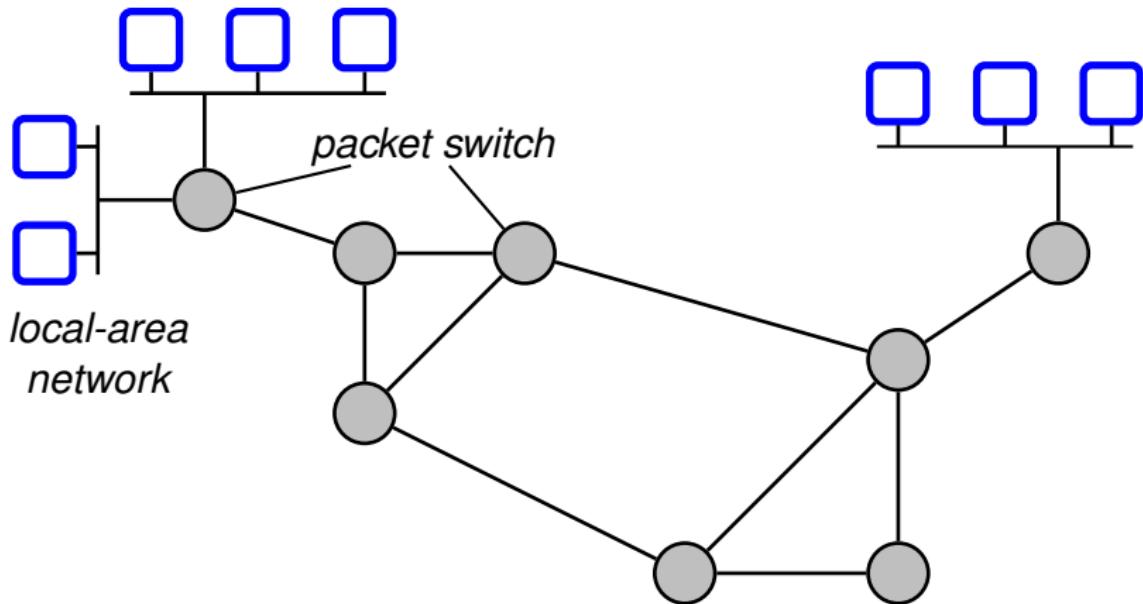
What is “Inside” the Internet?



What is “Inside” the Internet?



What is “Inside” the Internet?



Basic Concepts

- ◆ The Internet uses *packet switching*

Basic Concepts

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

Basic Concepts

- ◆ The Internet uses *packet switching*
- ◆ *Packet switch*: a *link-layer switch* or a *router*
- ◆ *Route*: sequence of switches that a packet goes through (a.k.a. *path*)

Basic Concepts

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

Packet Switching

- ◆ The Internet is a *packet-switched* network

Packet Switching

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

Packet Switching

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

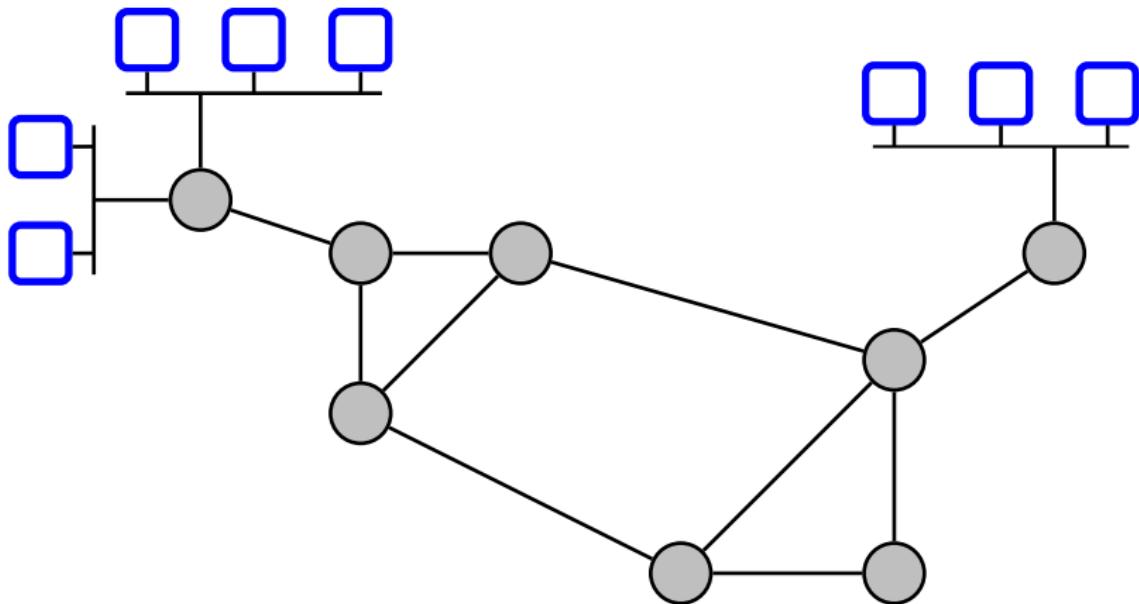
Packet Switching

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

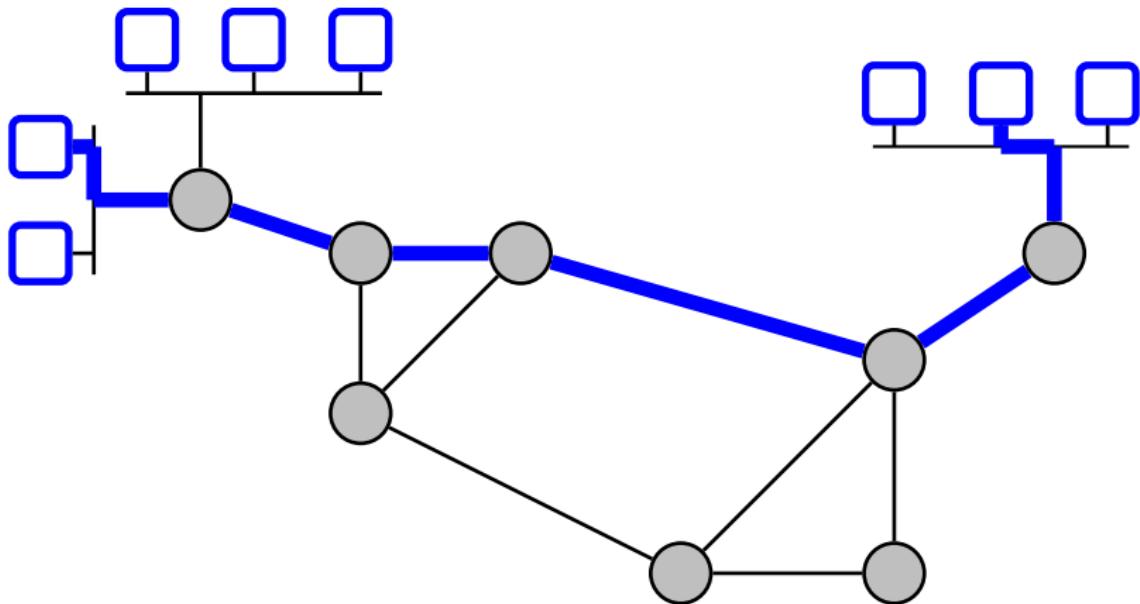
Packet Switching

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

Circuit Switching



Circuit Switching



Circuit Switching

- ◆ The telephone network is a typical circuit-switched network

Circuit Switching

- ◆ The telephone network is a typical circuit-switched network
- ◆ Communication requires a *connection setup* phase in which the network reserves all the necessary resources for that connection (links, buffers, switches, etc.)

Circuit Switching

- ◆ The telephone network is a typical circuit-switched network
- ◆ 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 that are dedicated to their connection for the entire duration of their conversation

Circuit Switching

- ◆ The telephone network is a typical circuit-switched network
- ◆ 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 that are dedicated to their 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 switching requires an expensive setup phase. On the other hand, once the connection is established, circuit switching is much more efficient in both space overhead and processing

Circuit vs. Packet Switching

- ◆ Circuit switching requires an expensive setup phase. On the other hand, once the connection is established, circuit switching is much more efficient in both space overhead and processing
- ◆ Packet switching does not incur any setup cost. However, it always incurs a significant processing cost for forwarding, as well as space overhead, because every packet must be self-contained

Circuit vs. Packet Switching

- ◆ Circuit switching requires an expensive setup phase. On the other hand, once the connection is established, circuit switching is much more efficient in both space overhead and processing
- ◆ Packet switching does not incur any setup cost. However, it always incurs a significant processing cost for forwarding, as well as space overhead, because every packet must be self-contained
- ◆ Circuit switching admits a straightforward implementation of quality-of-service guarantees, because network resources are reserved at connection setup time

Circuit vs. Packet Switching

- ◆ Circuit switching requires an expensive setup phase. On the other hand, once the connection is established, circuit switching is much more efficient in both space overhead and processing
- ◆ Packet switching does not incur any setup cost. However, it always incurs a significant processing cost for forwarding, as well as space overhead, because every packet must be self-contained
- ◆ Circuit switching admits a straightforward implementation of quality-of-service guarantees, because network resources are reserved at connection setup time
- ◆ Guaranteeing any quality of service with packet switching is very difficult

Circuit vs. Packet Switching (2)

- ◆ Circuit switching allows only a limited sharing of communication resources. In other words, circuit switching is an inefficient way to use the network

Circuit vs. Packet Switching (2)

- ◆ Circuit switching allows only a limited sharing of communication resources. In other words, circuit switching is an inefficient way to use the network
- ◆ Packet switching achieves a much better utilization of network resources, because it is designed specifically to share links (buffers, etc.)

Virtual Circuits

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

Virtual Circuits

- ◆ Idea: combine the advantages of circuit switching and packet switching
- ◆ The service model is connection oriented, and therefore requires a connection setup phase

Virtual Circuits

- ◆ Idea: combine the advantages of circuit switching and packet switching
- ◆ The service model is connection oriented, and therefore requires a connection setup phase
- ◆ The connection does not create a physical circuit, but rather a “virtual circuit”

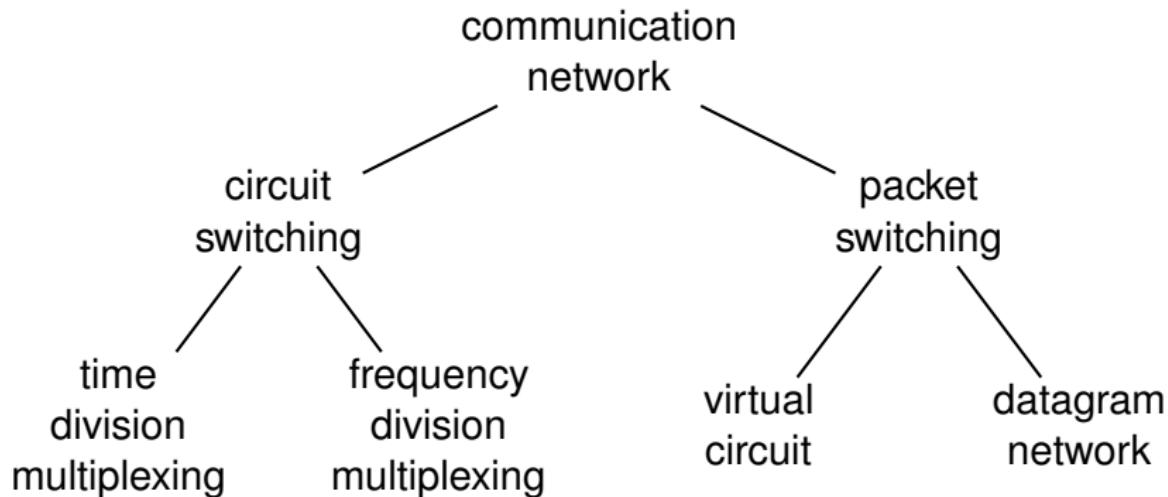
Virtual Circuits

- ◆ Idea: combine the advantages of circuit switching and packet switching
- ◆ The service model is connection oriented, and therefore requires 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

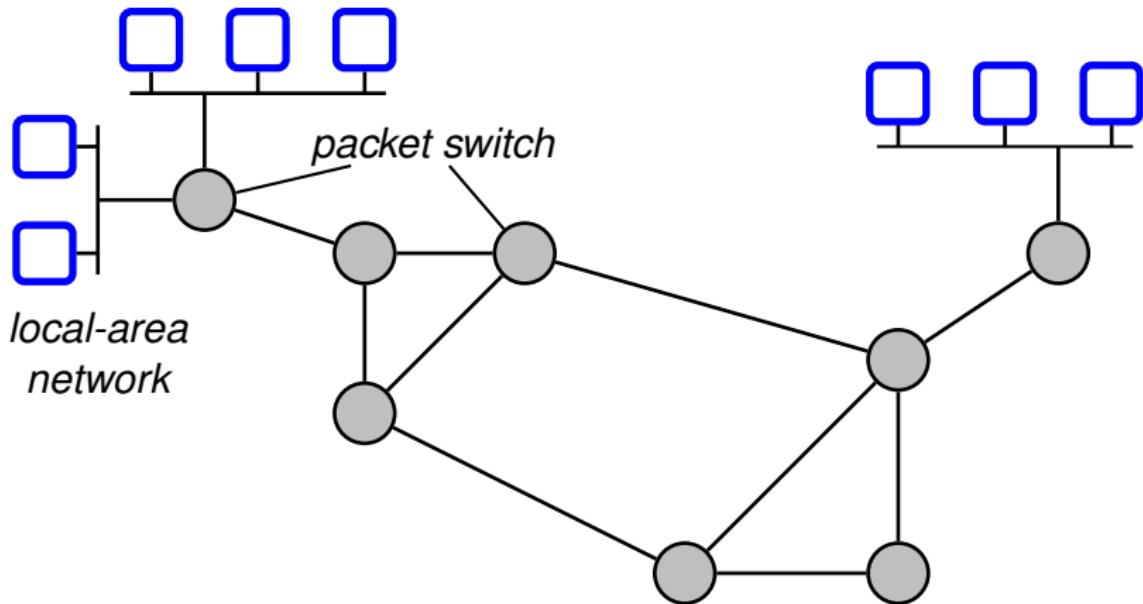
Virtual Circuits

- ◆ Idea: combine the advantages of circuit switching and packet switching
- ◆ The service model is connection oriented, and therefore requires 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 identifiers, which allow very fast processing (forwarding) and incur a space low-overhead

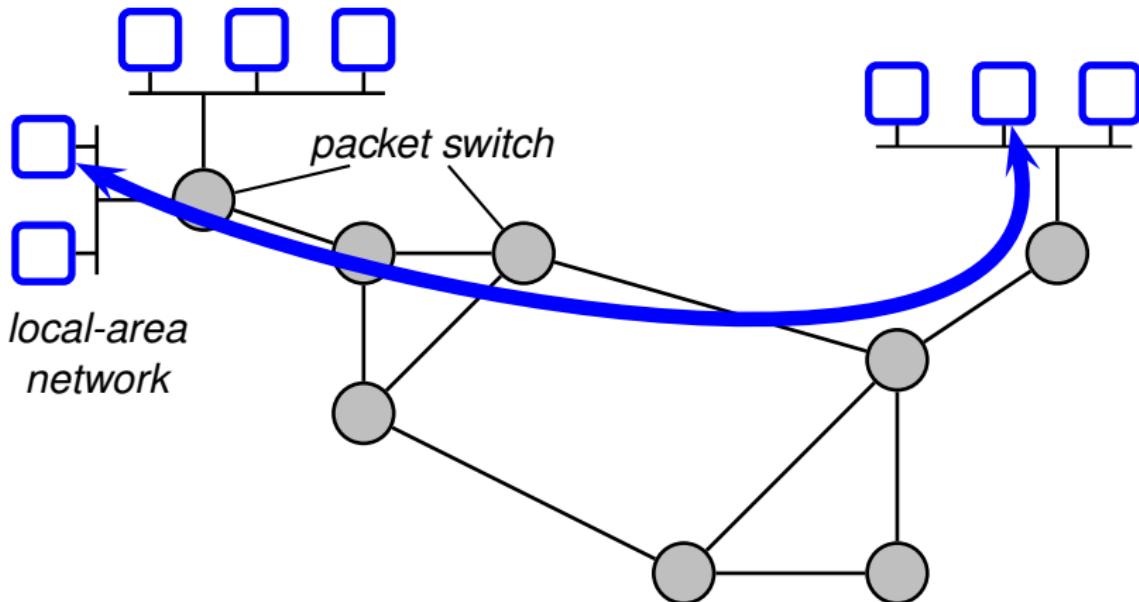
Taxonomy of Networks



Service Perspective



Service Perspective



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

Type of Service

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

Type of Service

- ◆ Two end systems can communicate through the Internet, but exactly what kind of communication service is that of the Internet?
- ◆ *Connection-oriented reliable*
 - ▶ virtual duplex communication channel ($A \leftrightarrow B$)—conceptually similar to a telephone service
 - ▶ information is transmitted “reliably” and in order

Type of Service

- ◆ Two end systems can communicate through the Internet, but exactly what kind of communication service is that of the Internet?
- ◆ *Connection-oriented reliable*
 - ▶ virtual duplex communication channel ($A \leftrightarrow B$)—conceptually similar to a telephone service
 - ▶ information is transmitted “reliably” and in order
- ◆ *Connectionless unreliable*
 - ▶ the network accepts “datagrams” for delivery—this is conceptually similar to the postal service
 - ▶ “unreliable” service (a.k.a. “best effort”)

Type of Service (2)

- ◆ How reliable is a “reliable” service?

Type of Service (2)

- ◆ 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.

Type of Service (2)

- ◆ 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)

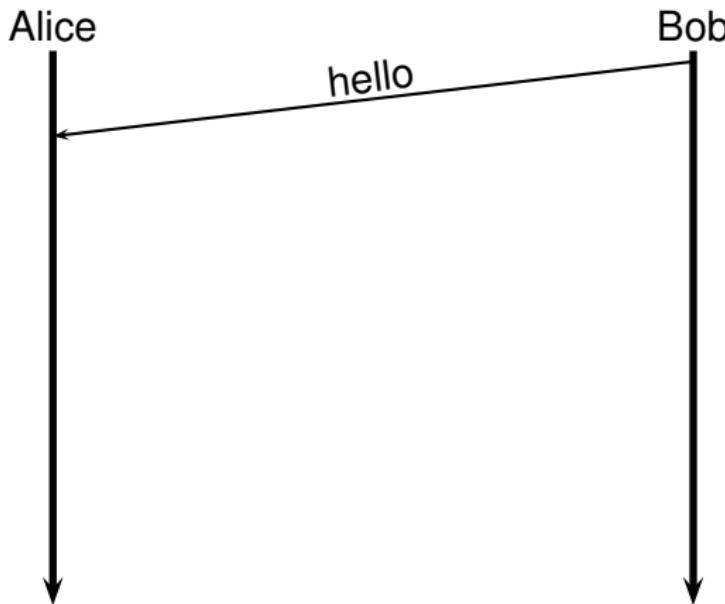
Communication Protocols

- ◆ 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?

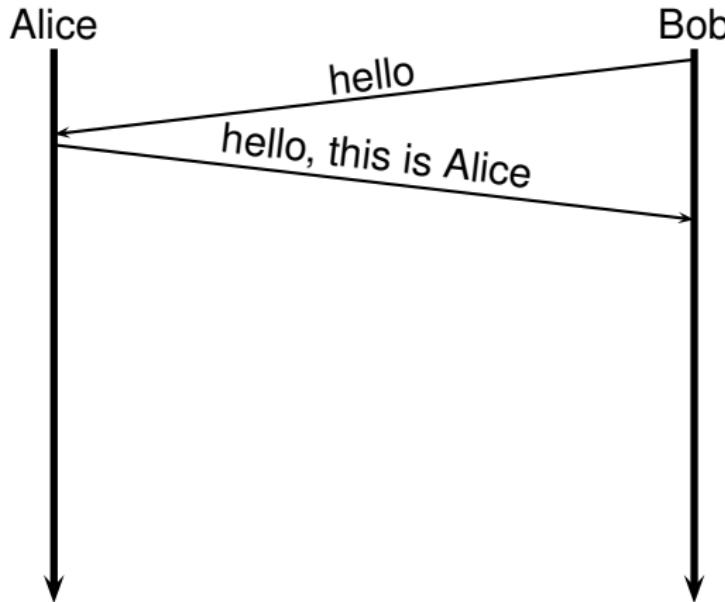
E.g., let's consider a phone call: Alice calls Bob



Communication Protocols

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

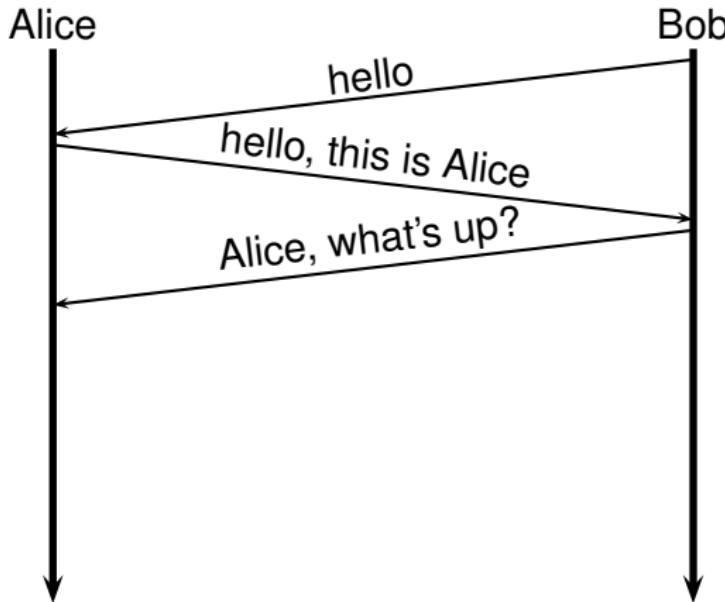
E.g., let's consider a phone call: Alice calls Bob



Communication Protocols

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

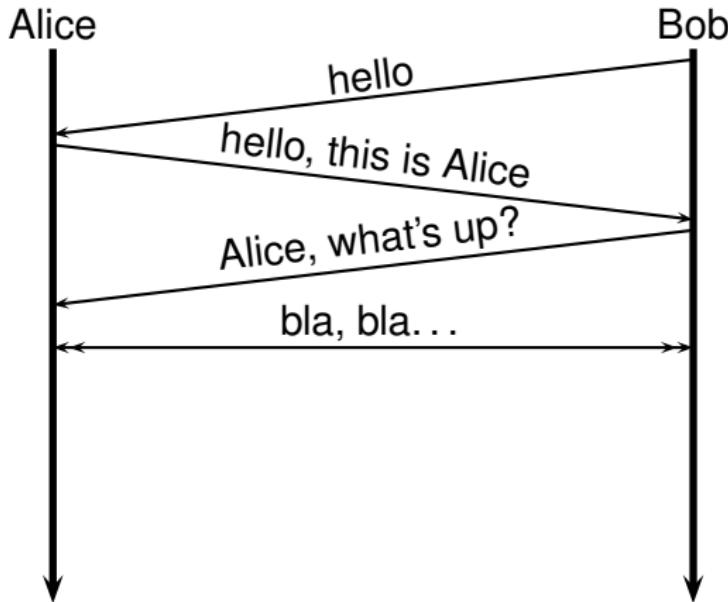
E.g., let's consider a phone call: Alice calls Bob



Communication Protocols

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

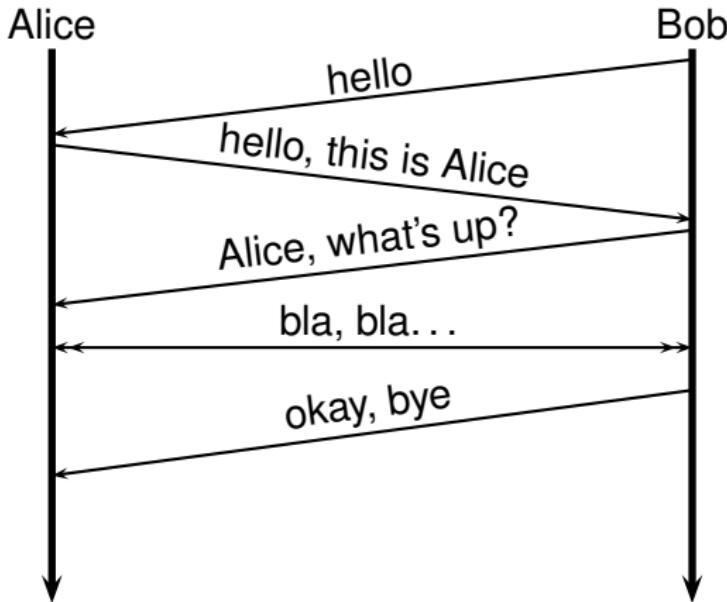
E.g., let's consider a phone call: Alice calls Bob



Communication Protocols

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

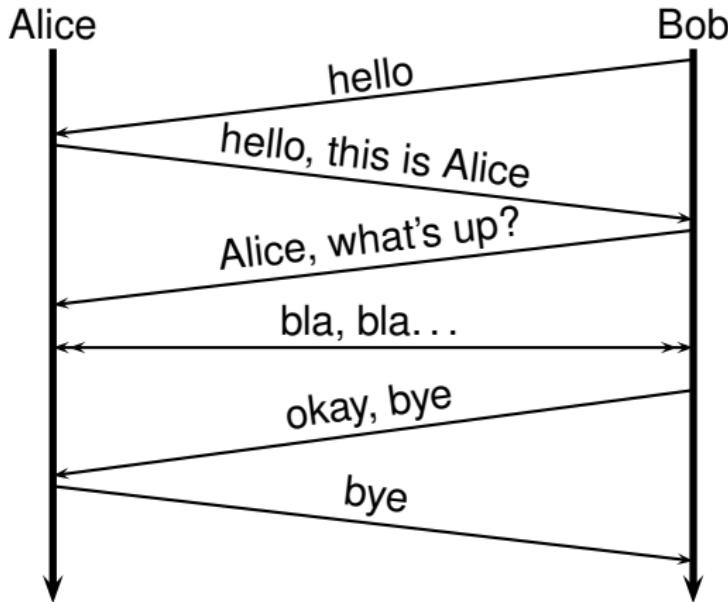
E.g., let's consider a phone call: Alice calls Bob



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

- ◆ Phases of the protocol
 - ▶ *handshake* establishes the identity and/or the context for a conversation
 - ▶ *conversation* free-form exchange
 - ▶ *closing* terminates the conversation

Communication Protocols

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

Communication Protocols (2)

- ◆ Another example: air traffic control

Communication Protocols (2)

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

Communication Protocols (2)

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

Communication Protocols (2)

- ◆ 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

Communication Protocols (2)

- ◆ 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
 - ▶ ...

Communication Protocols (2)

- ◆ 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

Communication Protocols (2)

- ◆ 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, goodbye

Communication Protocols (2)

- ◆ 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, goodbye
 - ▶ ... Center, request, Delta 800

Communication Protocols (2)

- ◆ 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, goodbye
 - ▶ ... Center, request, Delta 800
 - ▶ ... United 971, climb and maintain flight level 3-7-0

Communication Protocols (2)

- ◆ 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, goodbye
 - ▶ ... Center, request, Delta 800
 - ▶ ... United 971, climb and maintain flight level 3-7-0
 - ▶ flight level 3-7-0, United 971

Communication Protocols (2)

- ◆ 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, goodbye
- ▶ ... 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

Communication Protocols (2)

◆ 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, goodbye
- ▶ ... 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

Communication Protocols (2)

◆ 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, goodbye
- ▶ ... 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)

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)

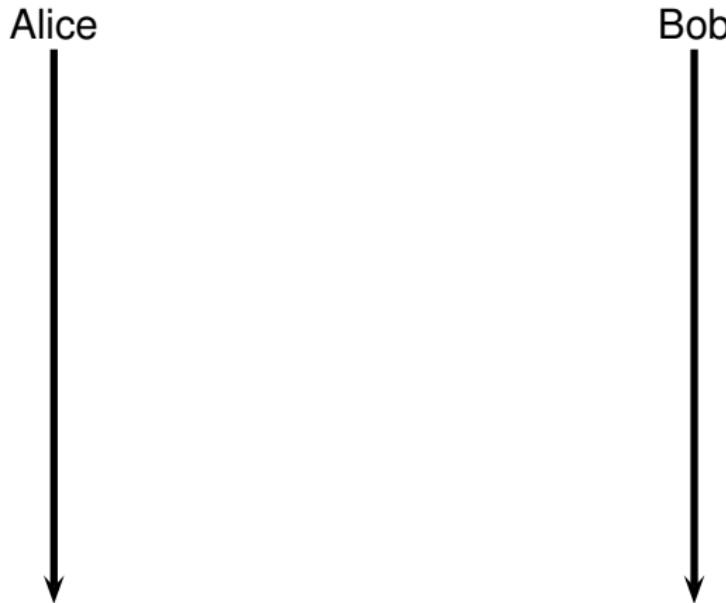
- ◆ A connectionless protocol
- ◆ Multi-party communication
- ◆ Medium access control (MAC) protocol
- ◆ Interleaved communication
- ◆ Acknowledgements
- ◆ Timeout and retransmission

Communication Protocols (2)

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

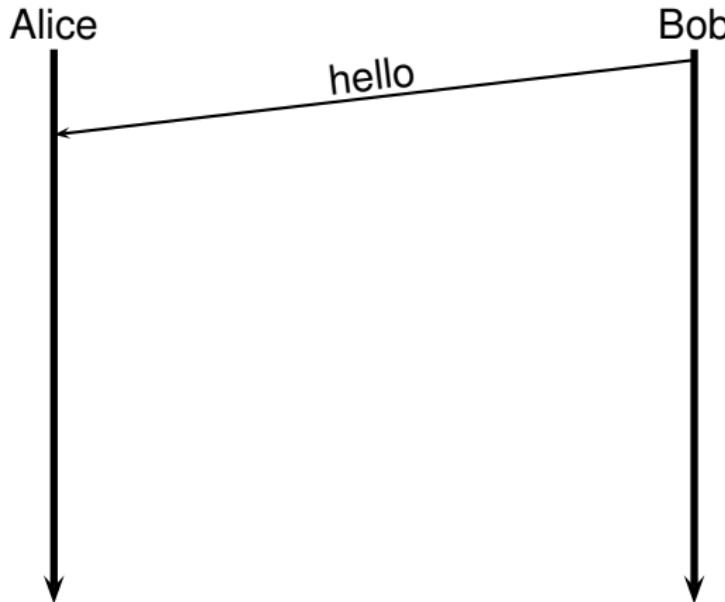
Communication Protocols (3)

- ◆ Let's revisit the phone-call protocol



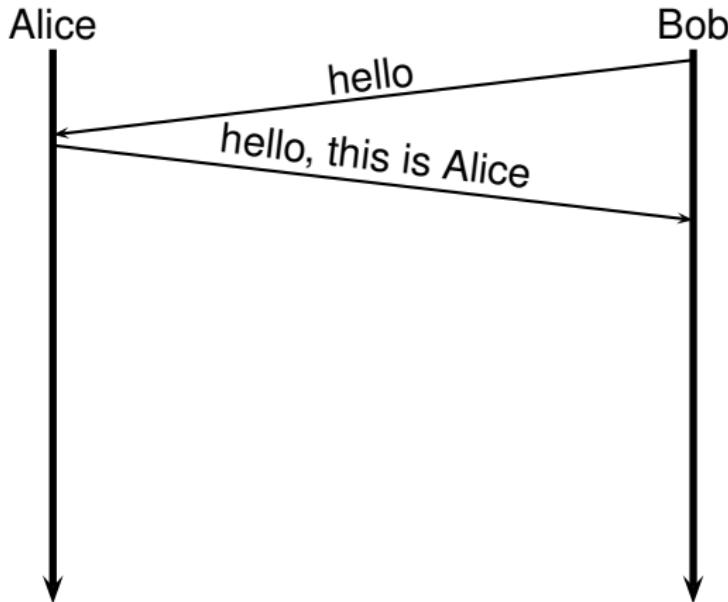
Communication Protocols (3)

- ◆ Let's revisit the phone-call protocol



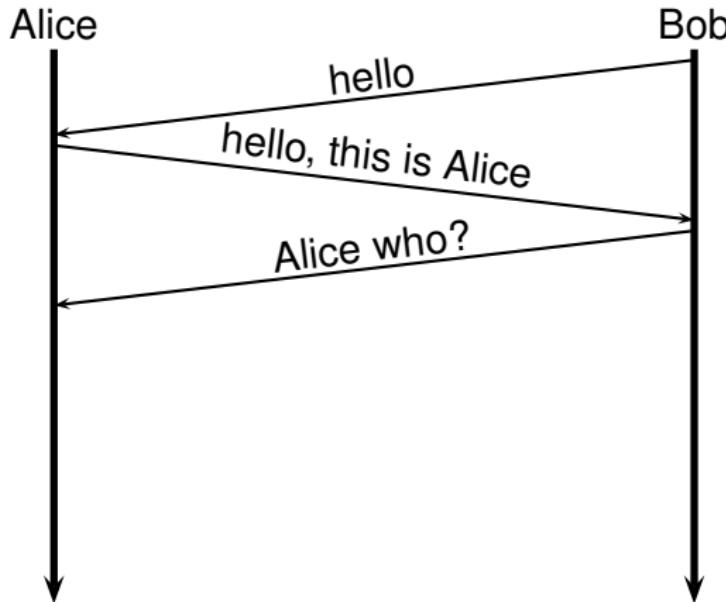
Communication Protocols (3)

- ◆ Let's revisit the phone-call protocol



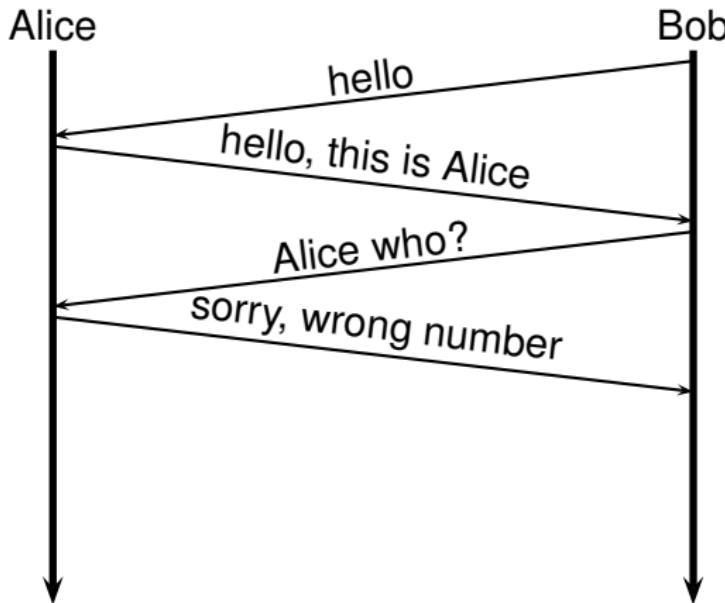
Communication Protocols (3)

- ◆ Let's revisit the phone-call protocol



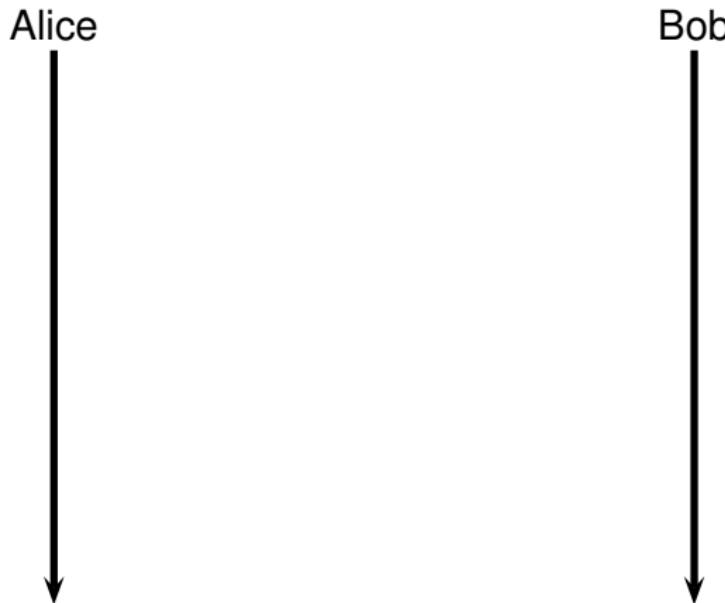
Communication Protocols (3)

- ◆ Let's revisit the phone-call protocol



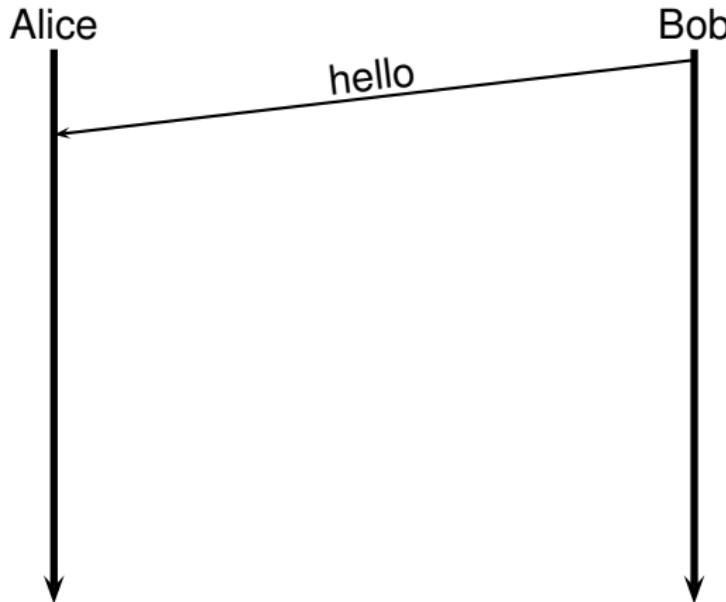
Communication Protocols (4)

- ◆ Another run of the phone-call protocol



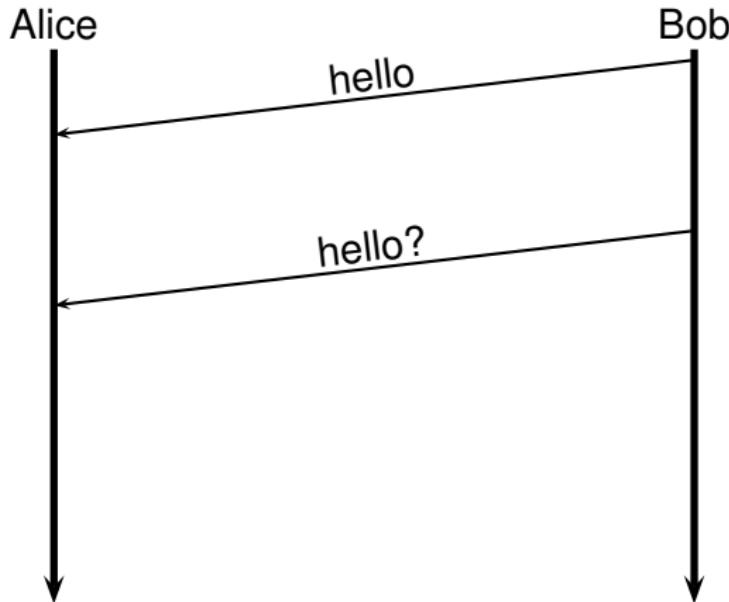
Communication Protocols (4)

- ◆ Another run of the phone-call protocol



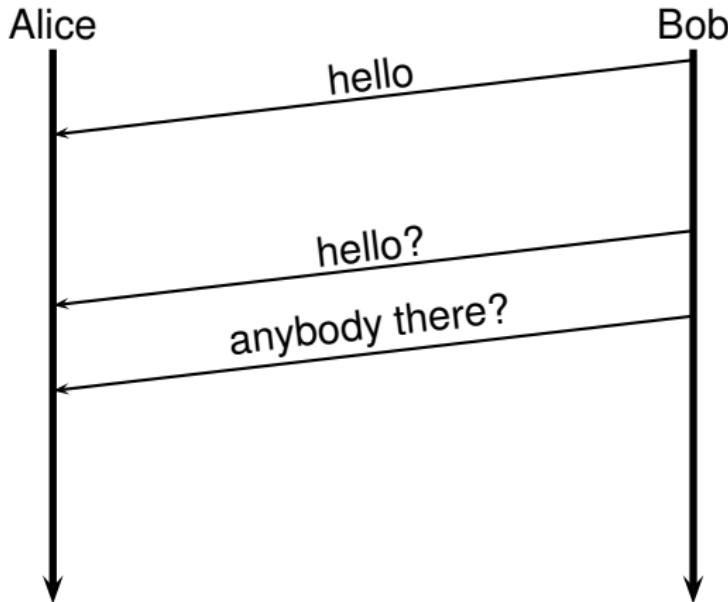
Communication Protocols (4)

- ◆ Another run of the phone-call protocol



Communication Protocols (4)

- ◆ 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

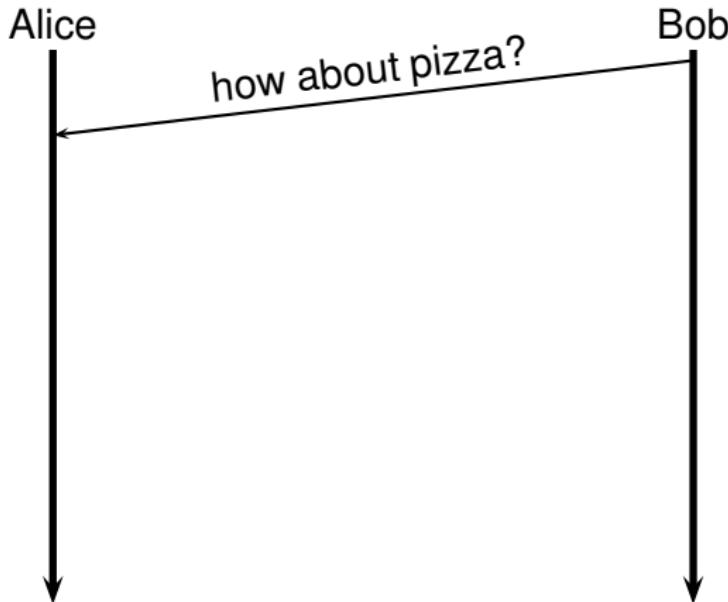
- ◆ A protocol is a lot like a program
 - ▶ in fact, it is a *distributed program*, where different processes can send messages to each other
- ◆ It is an *executable* specification
- ◆ It must be *unambiguous*
- ◆ It must be *complete*
 - ▶ i.e., it must include actions and/or responses for all possible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

Communication Protocols (5)

- ◆ Another protocol: deciding where to go for dinner

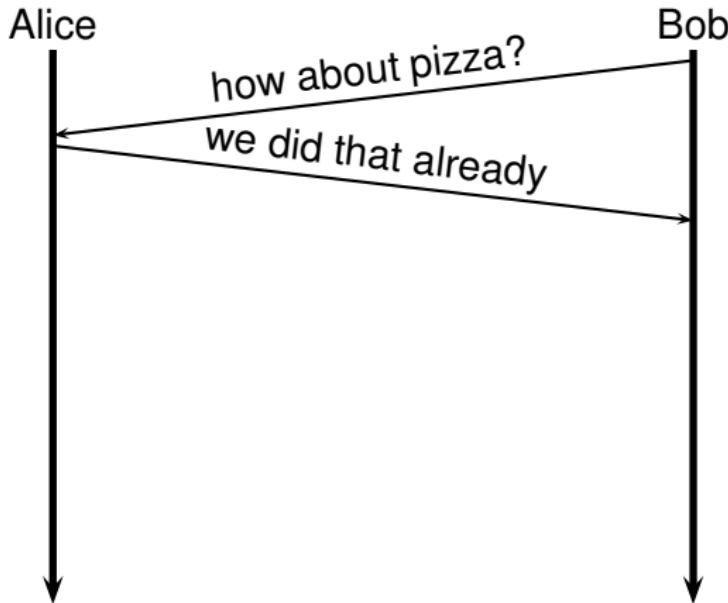
Communication Protocols (5)

- ◆ Another protocol: deciding where to go for dinner



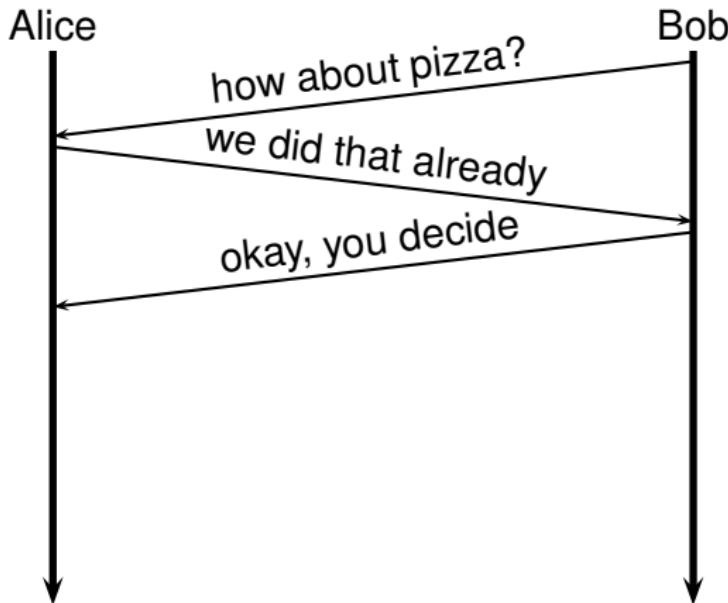
Communication Protocols (5)

- ◆ Another protocol: deciding where to go for dinner



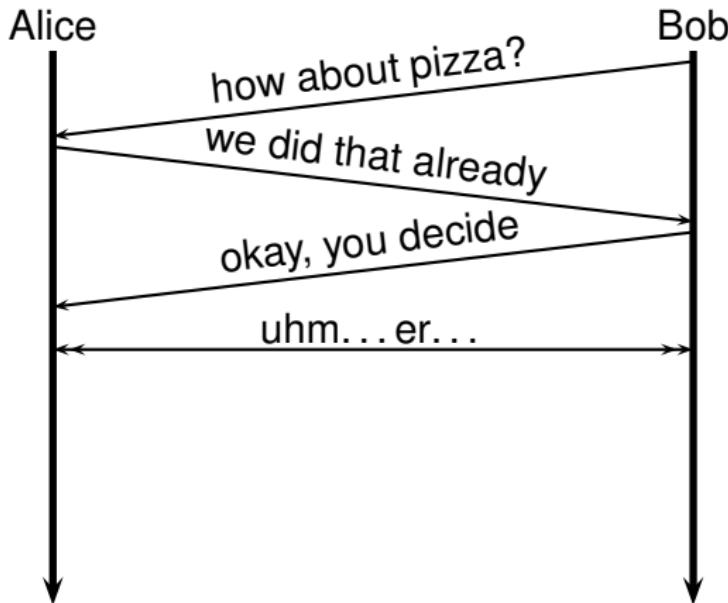
Communication Protocols (5)

- ◆ Another protocol: deciding where to go for dinner



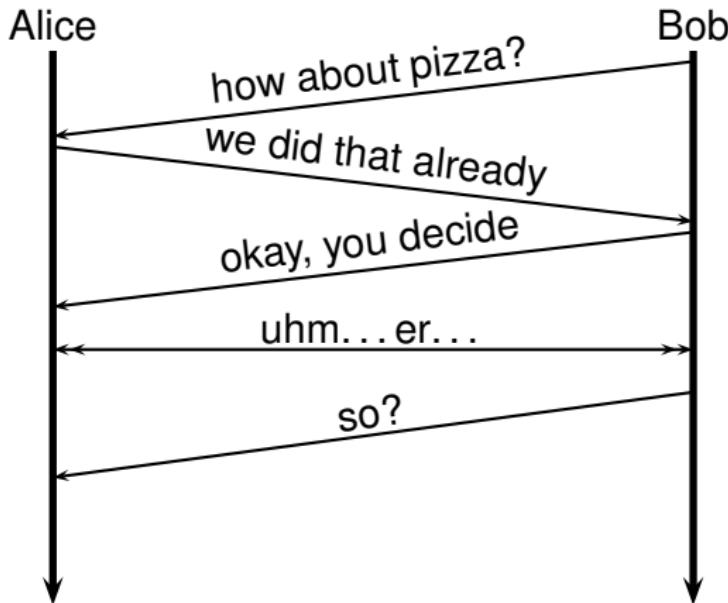
Communication Protocols (5)

- ◆ Another protocol: deciding where to go for dinner



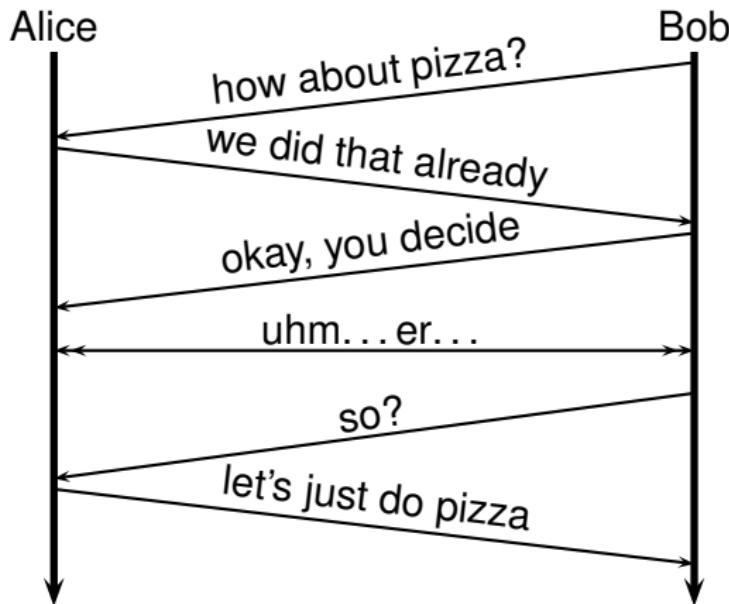
Communication Protocols (5)

- ◆ Another protocol: deciding where to go for dinner



Communication Protocols (5)

- ◆ Another protocol: deciding where to go for dinner

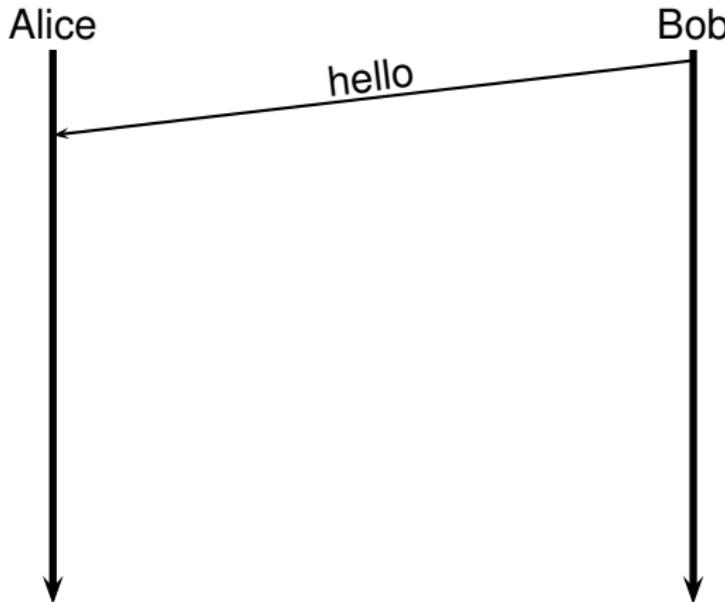


Protocol Layering

- ◆ Alice calls Bob to decide where to go for dinner

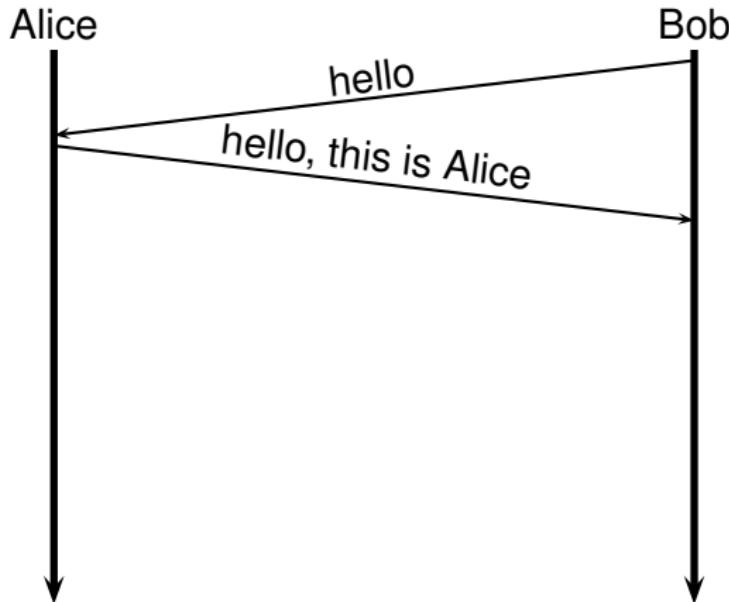
Protocol Layering

- ◆ Alice calls Bob to decide where to go for dinner



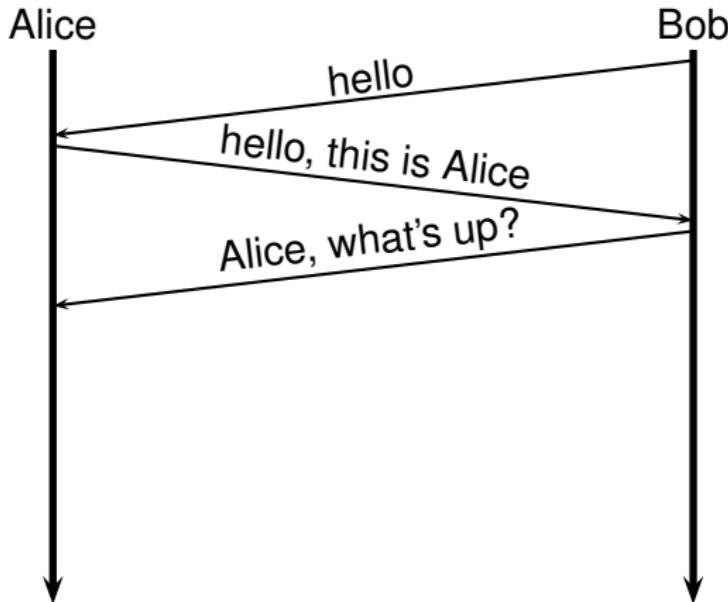
Protocol Layering

- ◆ Alice calls Bob to decide where to go for dinner



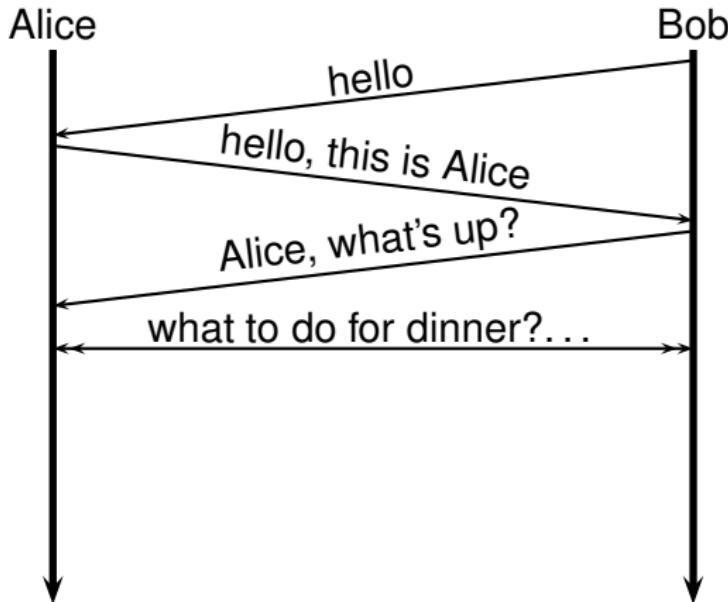
Protocol Layering

- ◆ Alice calls Bob to decide where to go for dinner



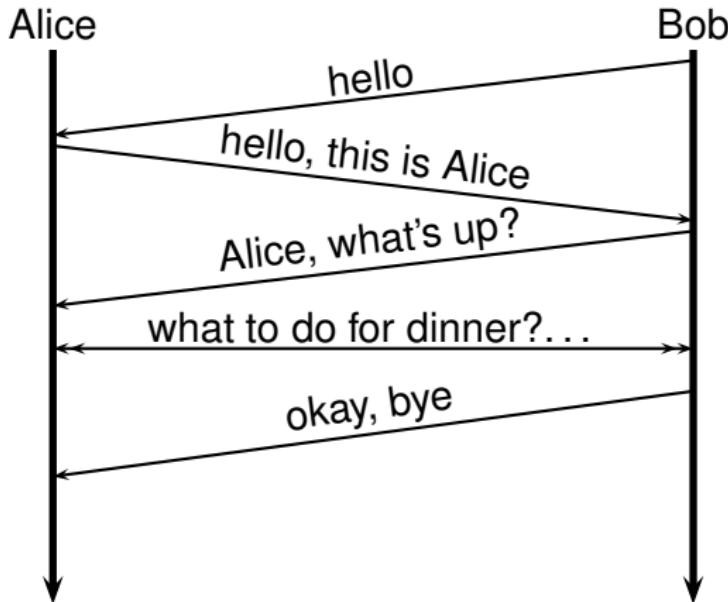
Protocol Layering

- ◆ Alice calls Bob to decide where to go for dinner



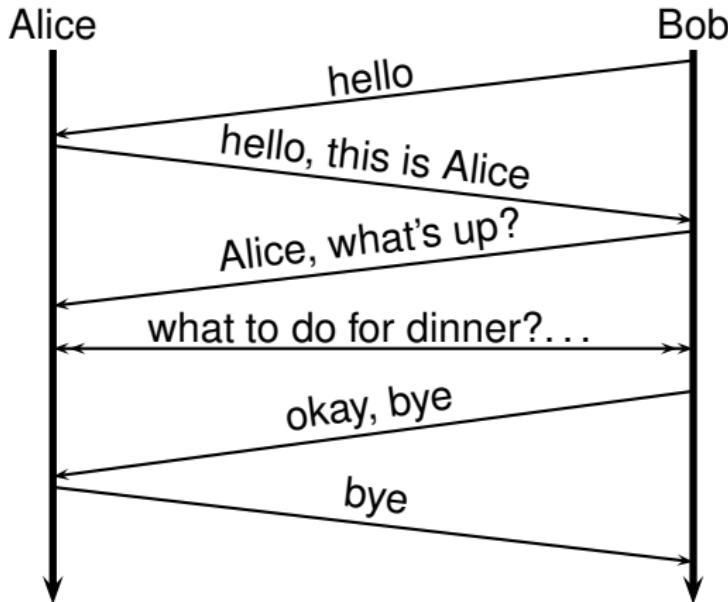
Protocol Layering

- ◆ Alice calls Bob to decide where to go for dinner



Protocol Layering

- ◆ Alice calls Bob to decide where to go for dinner



Protocol Layering

- ◆ Alice calls Bob to decide where to go for dinner

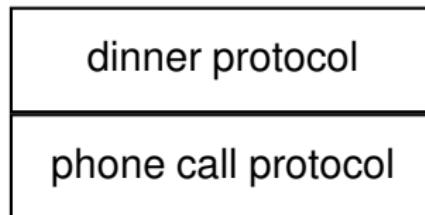
Protocol Layering

- ◆ Alice calls Bob to decide where to go for dinner

phone call protocol

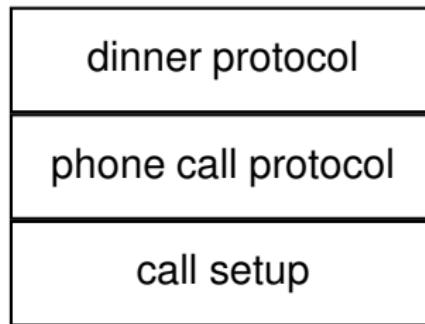
Protocol Layering

- ◆ Alice calls Bob to decide where to go for dinner



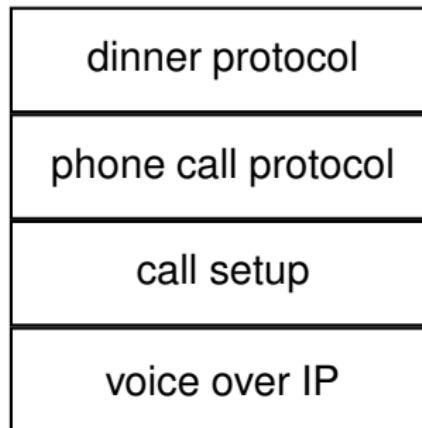
Protocol Layering

- ◆ Alice calls Bob to decide where to go for dinner



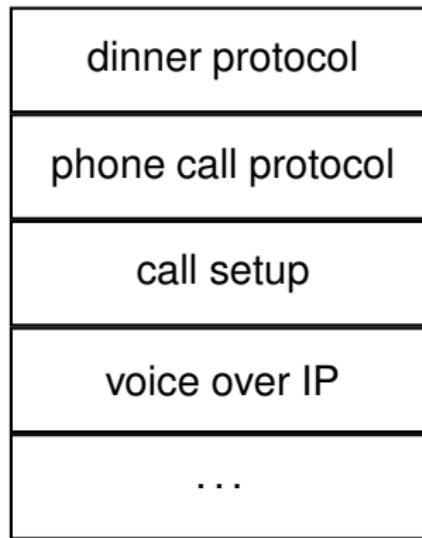
Protocol Layering

- ◆ Alice calls Bob to decide where to go for dinner



Protocol Layering

- ◆ Alice calls Bob to decide where to go for dinner

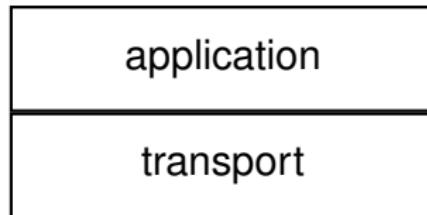


Internet Protocol Stack

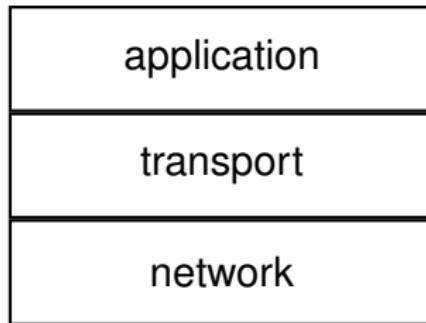
Internet Protocol Stack

application

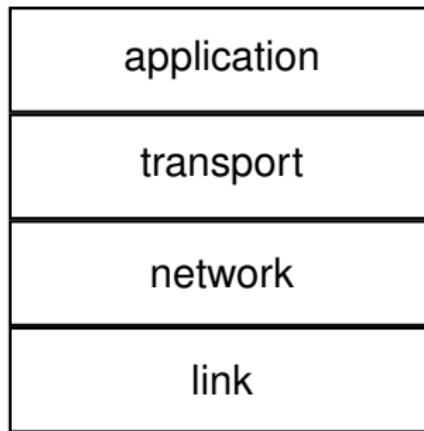
Internet Protocol Stack



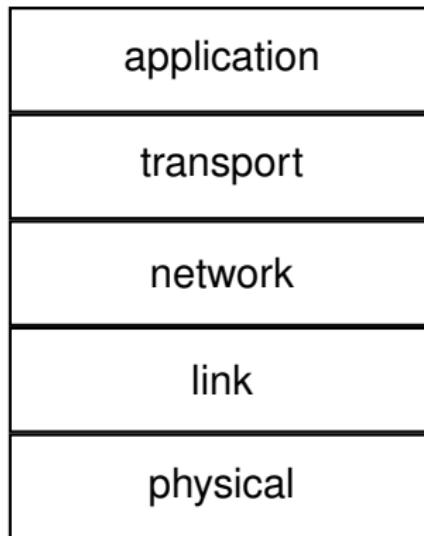
Internet Protocol Stack



Internet Protocol Stack



Internet Protocol Stack



Internet Protocol Stack (2)

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

Internet Protocol Stack (2)

- ◆ *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)

Internet Protocol Stack (2)

- ◆ *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)

Internet Protocol Stack (2)

- ◆ *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)

Internet Protocol Stack (2)

- ◆ *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*