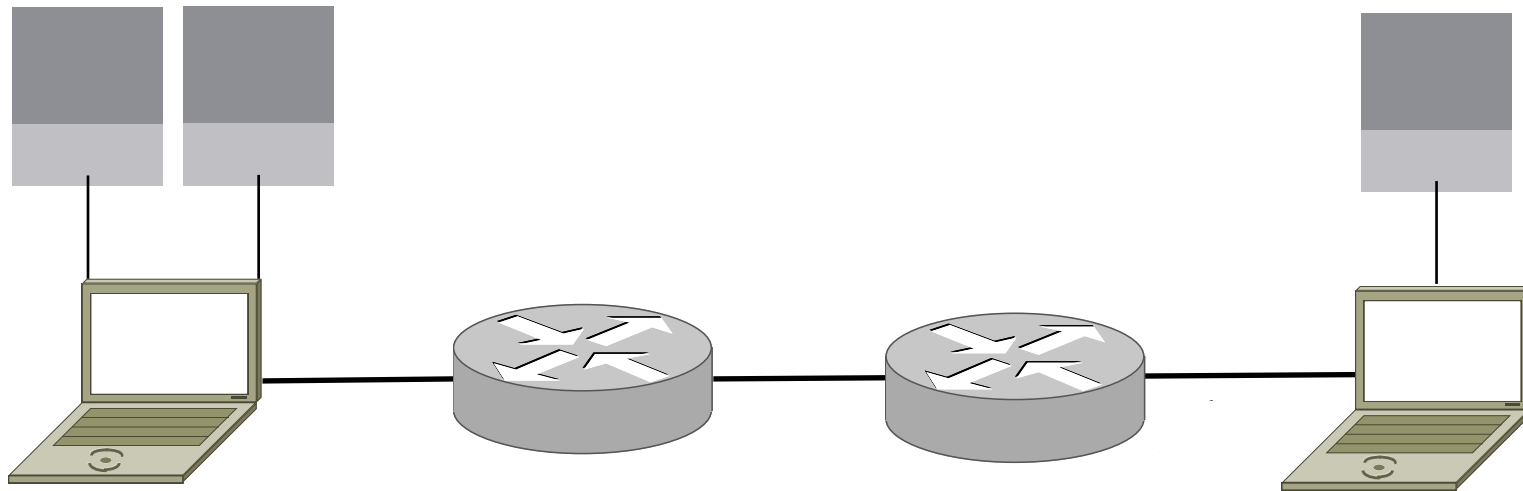
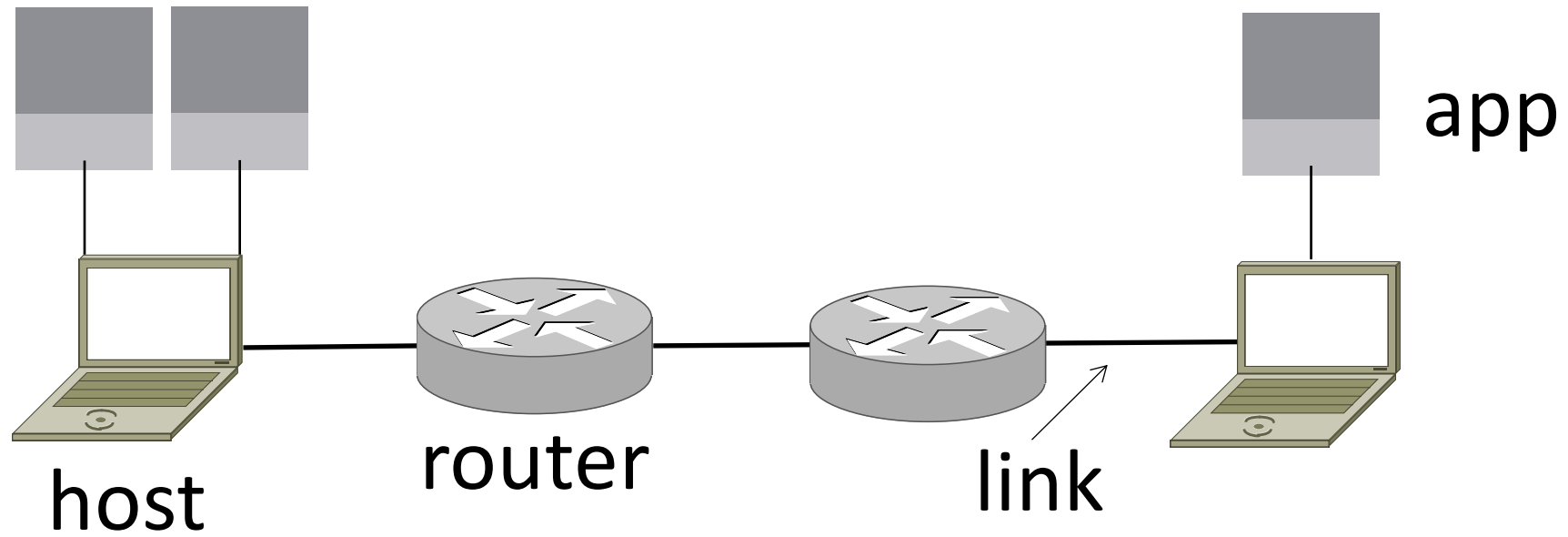


# Network Components

# Parts of a Network



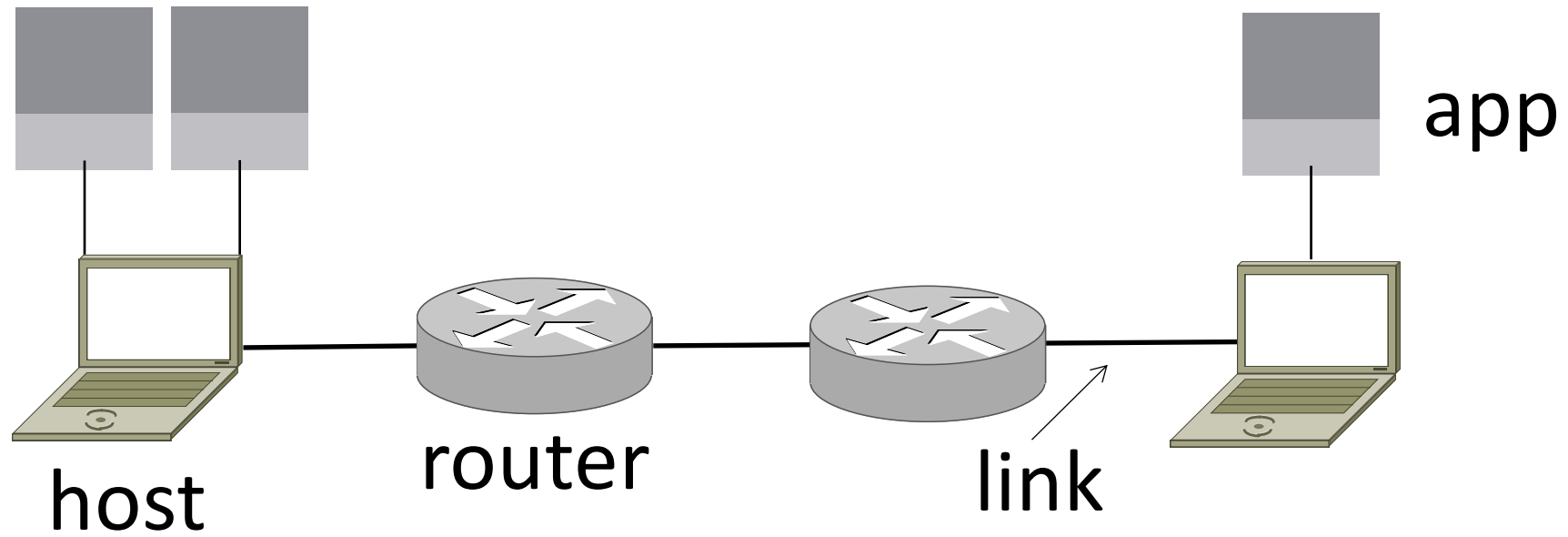
# Parts of a Network



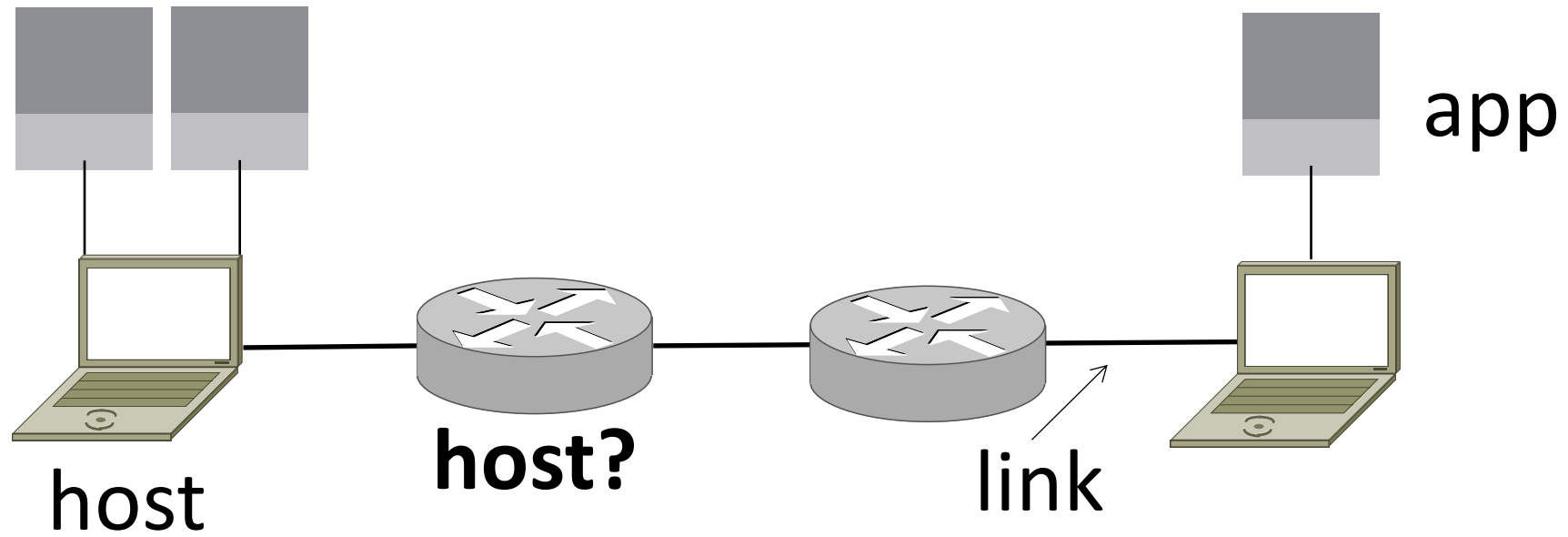
# Component Names

<b>Component</b>	<b>Function</b>	<b>Example</b>
<u>Application</u> , or app, user	Generates messages	Zoom, iTunes, Browser
<u>Host</u> , or end-system, edge device, node, source, sink	Runs the app	Laptop, mobile, desktop
<u>Router</u> , or switch, node, hub	Relays messages across links	Access point, cable/DSL modem
<u>Link</u> , or channel	Carries messages	Wires, wireless

# Parts of a Network



# Parts of a Network

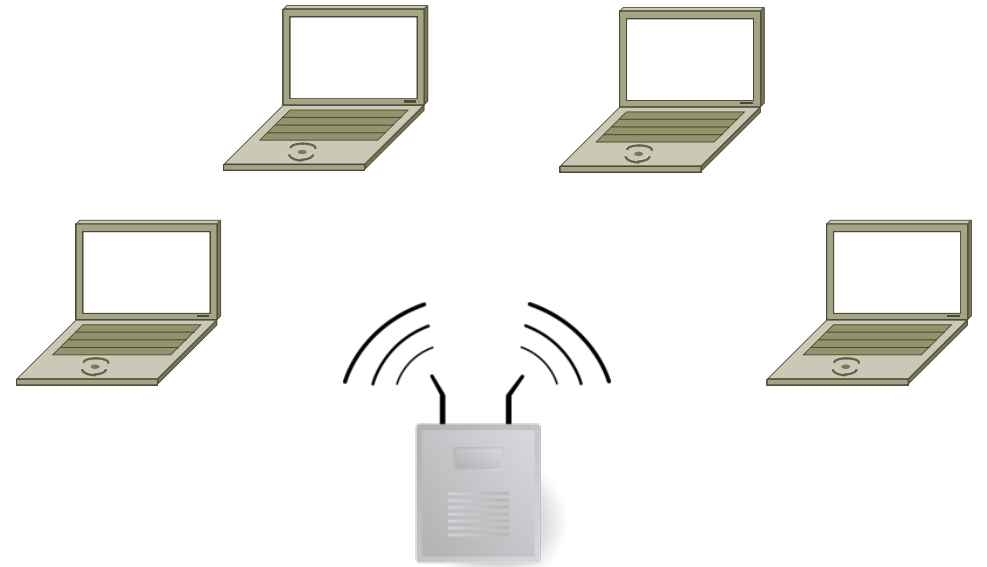


# Types of Links

- Full-duplex
  - Bidirectional
- Half-duplex
  - Bidirectional
- Simplex
  - unidirectional

# Wireless Links

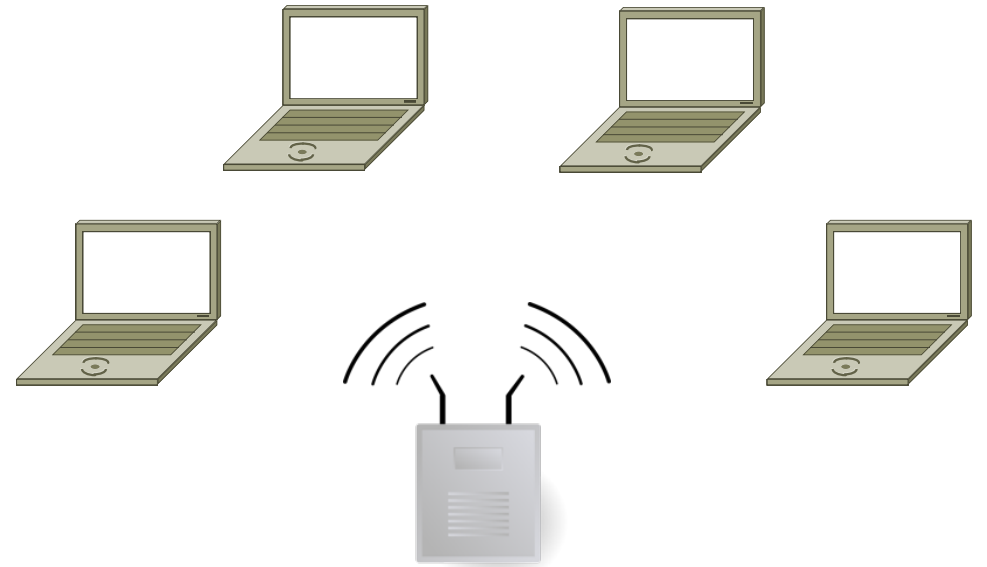
- Message is broadcast
  - Received by all nodes in range
  - Not a good fit with our model





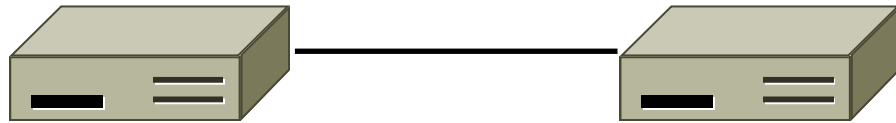
# Wireless Links (2)

- Often show logical links
  - Not all possible connectivity

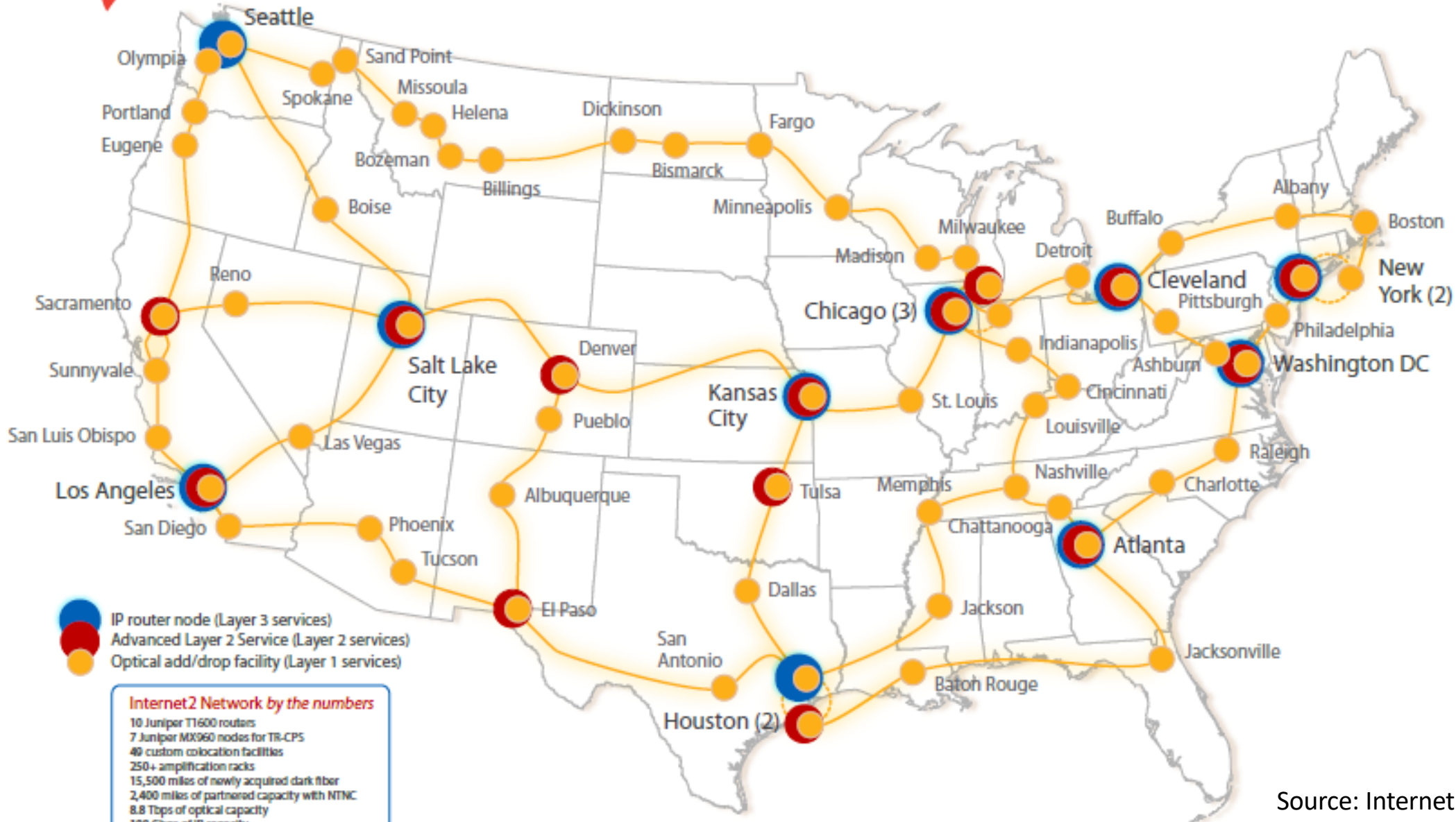


# A Small Network

- Connect a couple of computers



- Next, a large network ...



IN SUPPORT OF  
**U.S.UCAN**

NETWORK  
PARTNERS

**ciena**

**CISCO**

**INDIANA UNIVERSITY**

**infinera**

**JUNIPER**



Source: Internet2

Example Computer Networks?

# Example Computer Networks

- WiFi (802.11)
- Enterprise / Ethernet
- ISP (Internet Service Provider)
- Cable / DSL
- Mobile phone / cellular (2G, 3G, 4G, 5G)
- Bluetooth
- Telephone
- Satellite ...

# Computer network names by scale

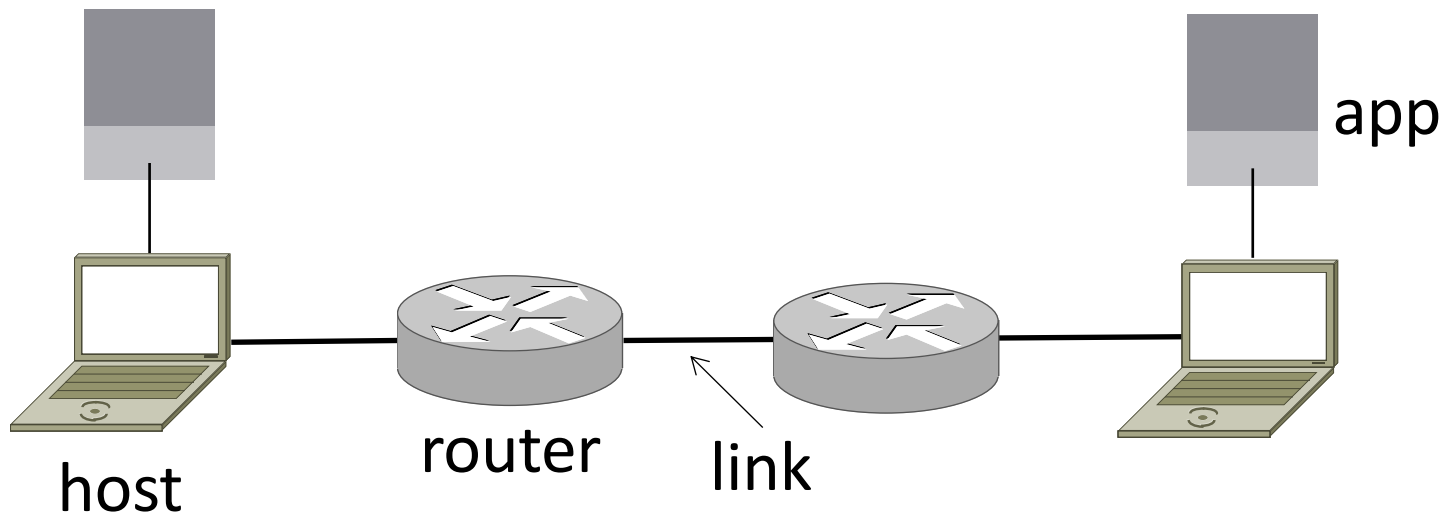
<b>Scale</b>	<b>Type</b>	<b>Example</b>
Vicinity	<u>PAN</u> (Personal Area Network)	Bluetooth (e.g., headset)
Building	<u>LAN</u> (Local Area Network) <u>DCN</u> (Data Center Network)	WiFi, Ethernet Ethernet
City	<u>MAN</u> (Metropolitan Area Network)	Cable, DSL
Country	<u>WAN</u> (Wide Area Network)	Large ISP
Planet	The Internet (network of all networks)	The Internet!

# Internetworks

- An internetwork, or internet, is what you get when you join networks together
  - Just another network
- The Internet (capital “I”) is the internet we all use

# Network Boundaries

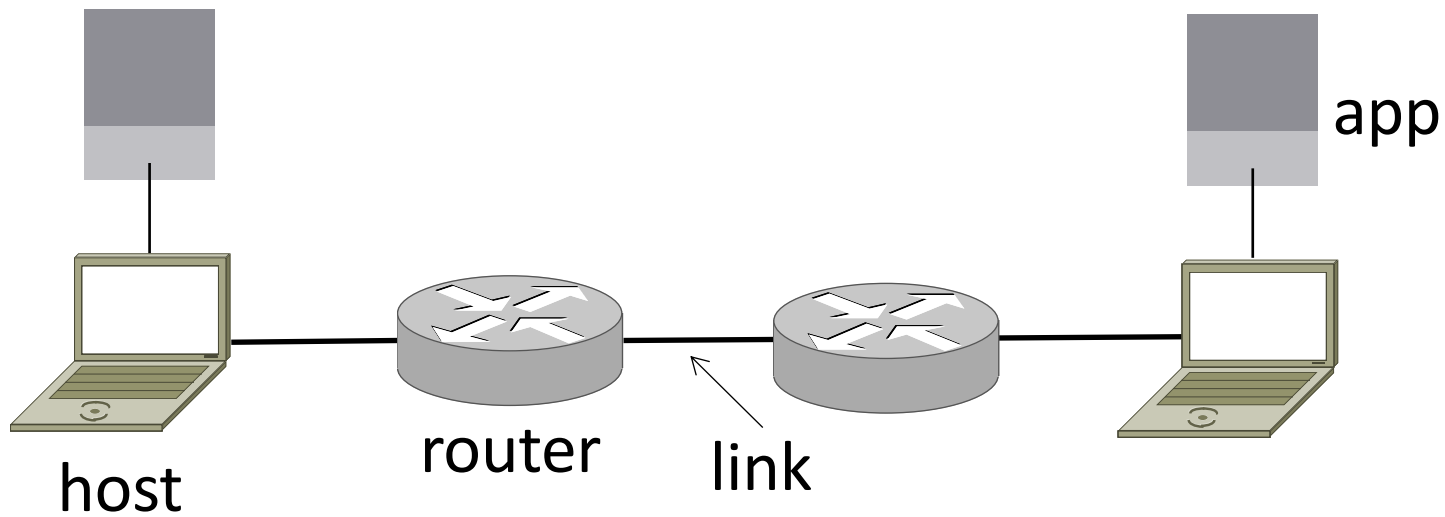
- What part is the “network”?





# Network Boundaries (2)

- What part represents an “ISP”?



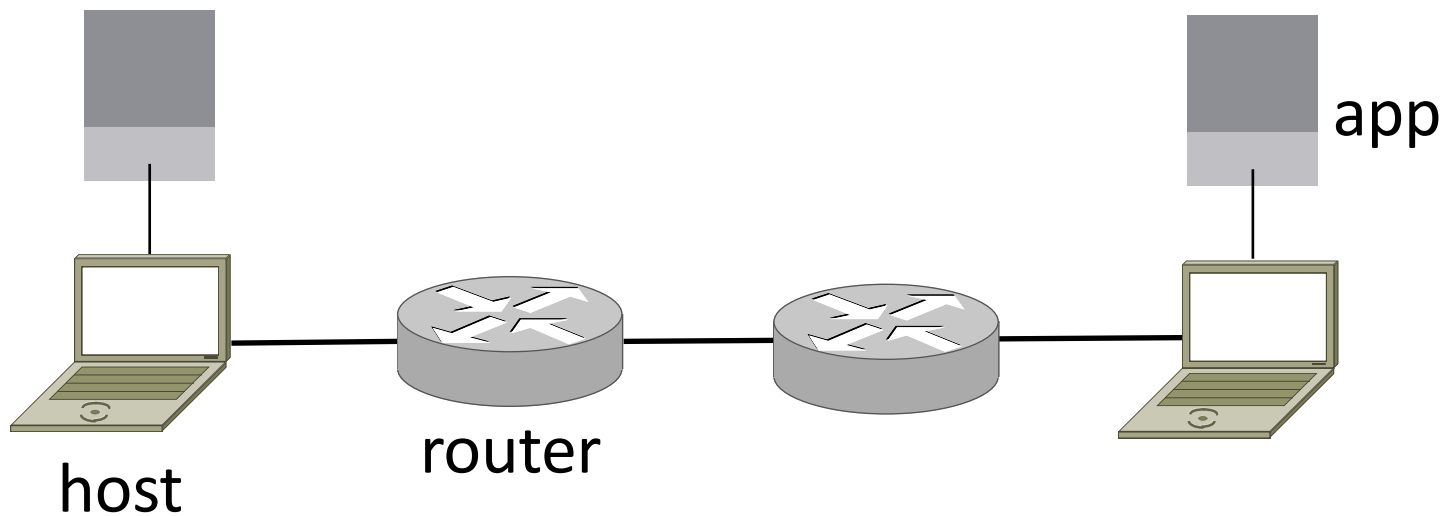
# Network Boundaries (3)

- Cloud as a generic network



# Key Interfaces

- Between (1) apps and network, and (2) network components



What should networks do for apps?

# What should networks do for apps?

- Make and break connections
- Find a path through the network
- Transfers information reliably
- Transfers arbitrary length information
- Send as fast as the network allows
- Shares bandwidth among users
- Secures information in transit
- Lets many new hosts be added
  - ...

# What should networks do for apps?

- Make and break connections
- Find the best path through the network
- Transfer data
- Transfer information
- Secure data flows
- Share resources
- Secures information in transit
- Lets many new hosts be added
- ...

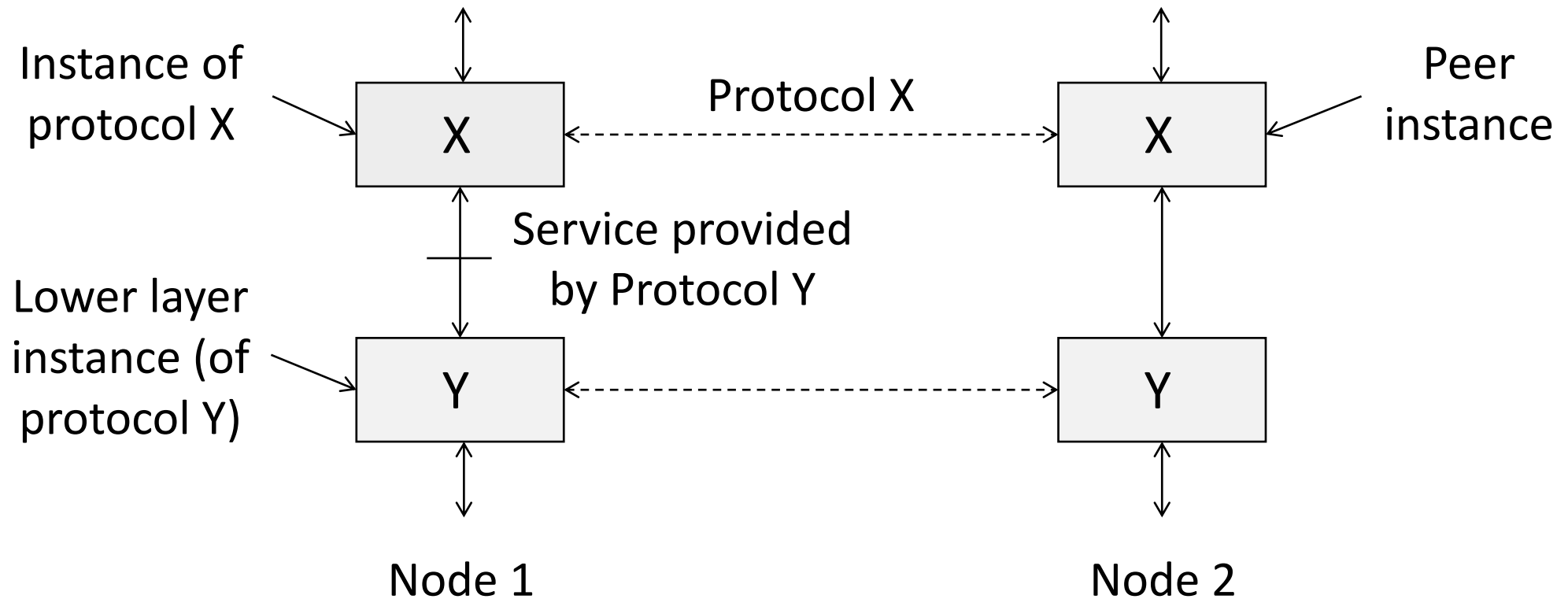
**We need modularity  
to help manage  
complexity and  
support reuse**

# Protocols and Layers

- Protocols and layering is the main structuring method used to divide up network functionality
  - Divide functionality in layers organized vertically
  - Each protocol implements the functionality of that layer
  - Each protocol instance talks *virtually* to its peer instances using the protocol
  - Each protocol instances uses only the services of the lower layer

# Protocols and Layers (2)

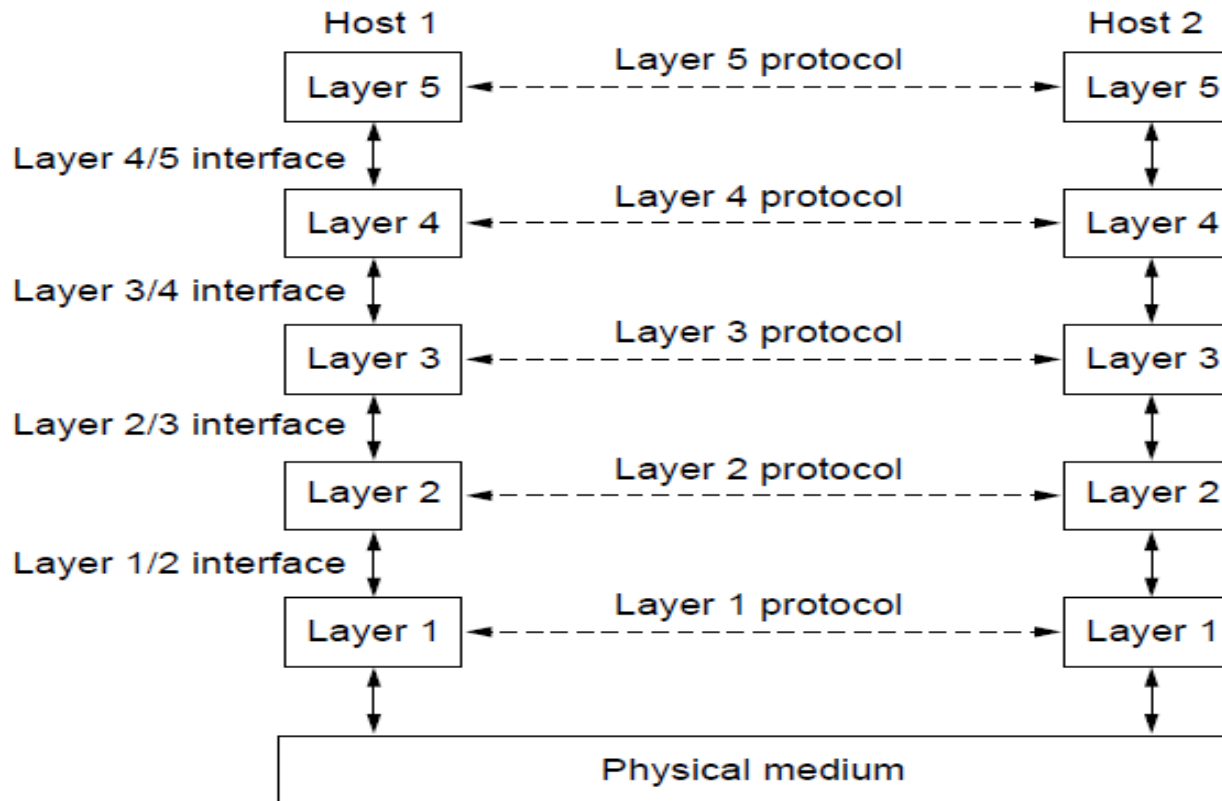
- Protocols are horizontal, layers are vertical





# Protocols and Layers (3)

- Set of protocols in use is called a protocol stack

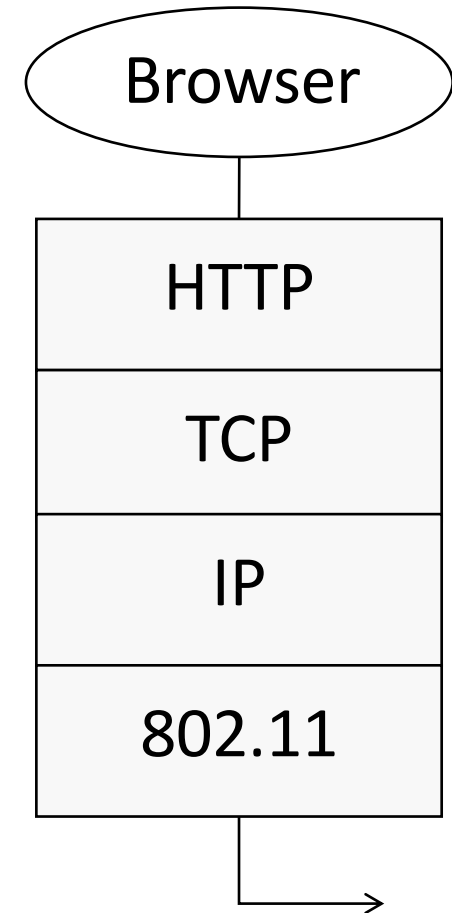


# Protocols and Layers (4)

- Protocols you've probably heard of:
  - TCP, IP, 802.11, Ethernet, HTTP, SSL, DNS, ... and many more

# Protocols and Layers (5)

- Protocols you've probably heard of:
  - TCP, IP, 802.11, Ethernet, HTTP, SSL, DNS, ... and many more
- An example protocol stack
  - Used by a web browser on a host that is wirelessly connected to the Internet

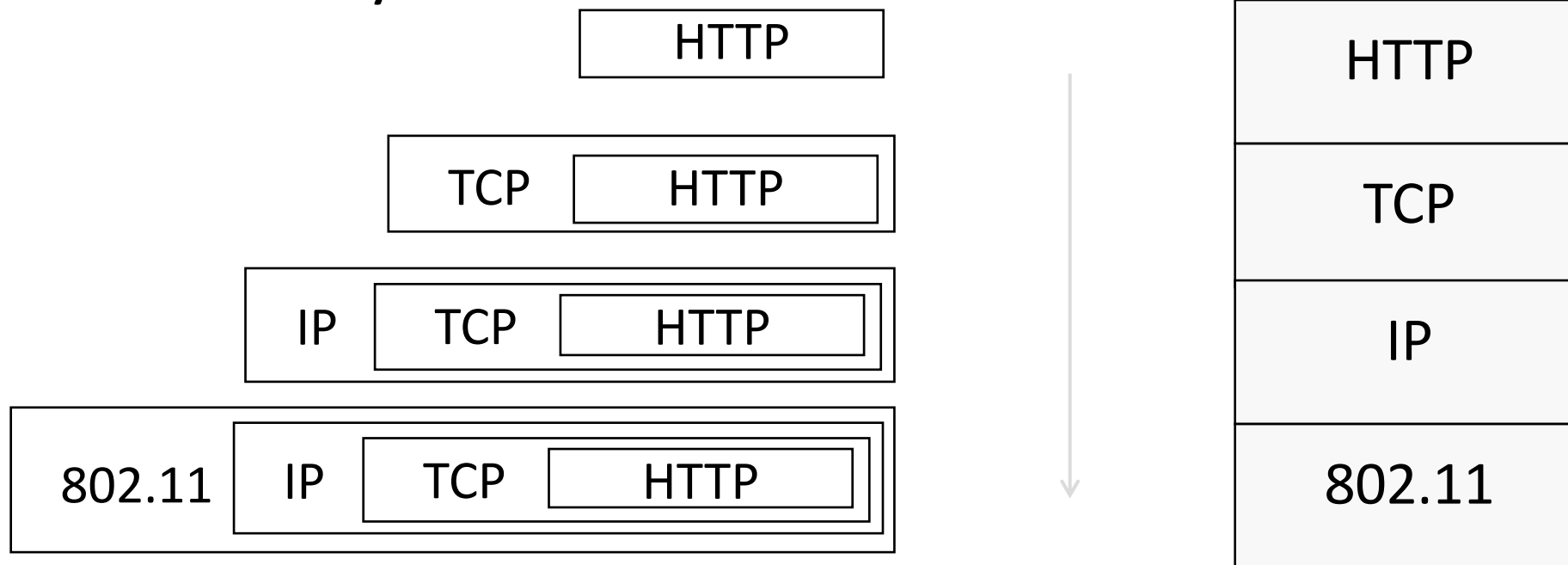


# Encapsulation

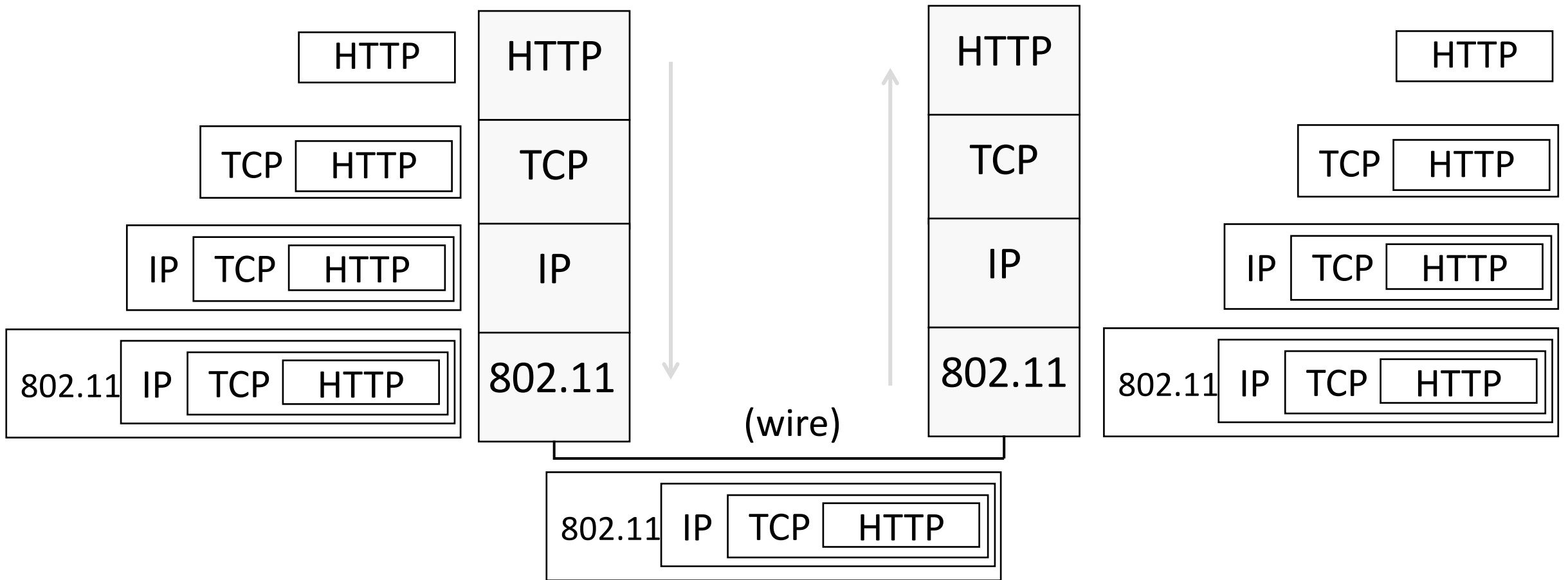
- Encapsulation is the mechanism used to effect protocol layering
  - Lower layer wraps higher layer content, adding its own information to make a new message for delivery
  - Like sending a letter in an envelope; postal service doesn't look inside

# Encapsulation (2)

- Message “on the wire” begins to look like an onion
  - Lower layers are outermost

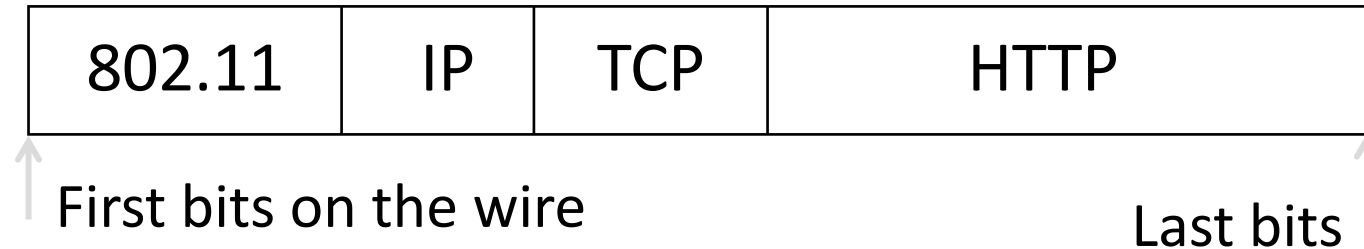


# Encapsulation (3)



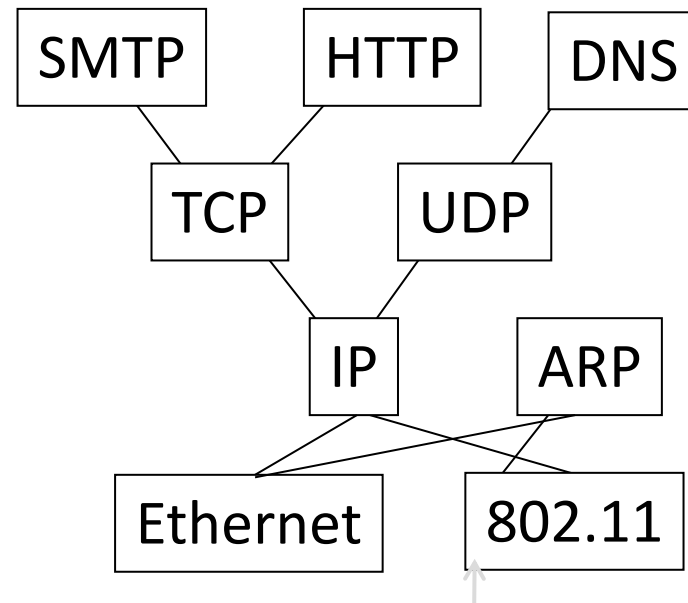
# Encapsulation (4)

- Normally draw message like this:
  - Each layer adds its own header



- More involved in practice
  - Trailers as well as headers, encrypt/compress contents
  - Segmentation (divide long message) and reassembly

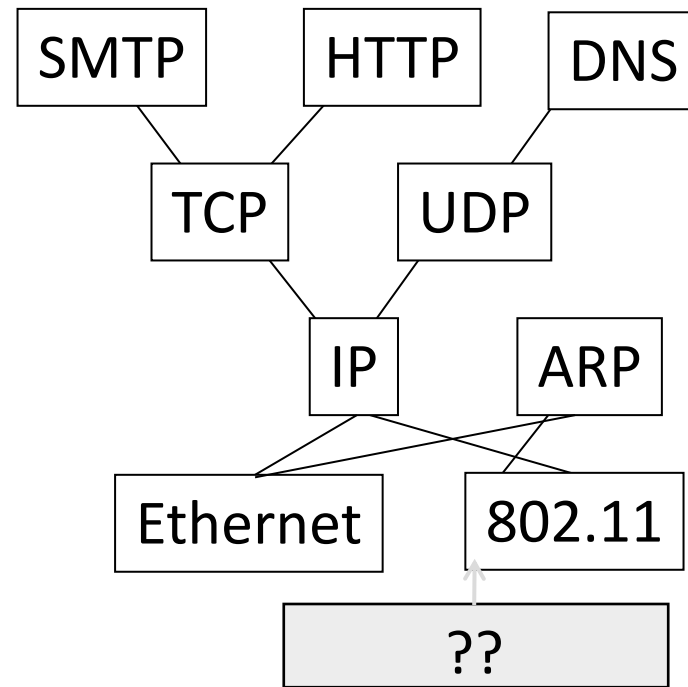
# Multiple protocols in a layer





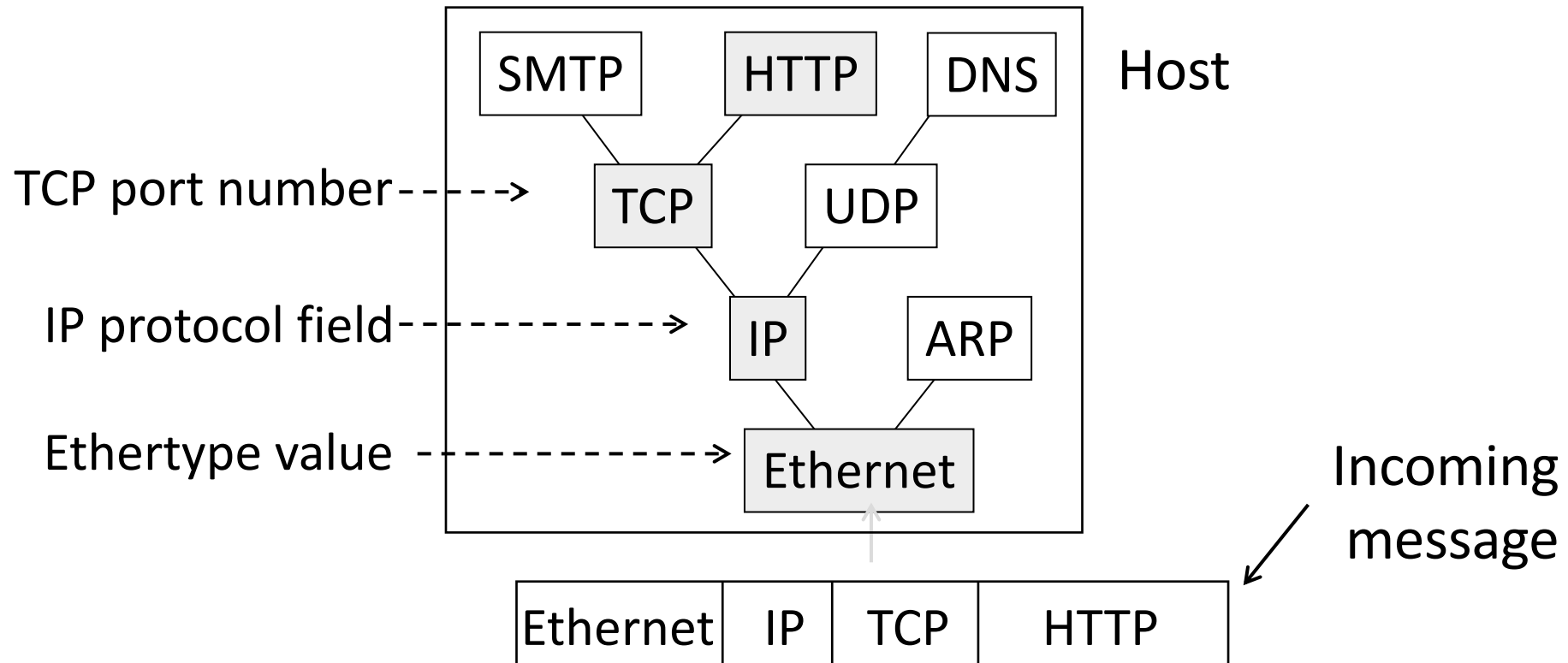
# Demultiplexing

- Pass incoming message to the protocols that it uses



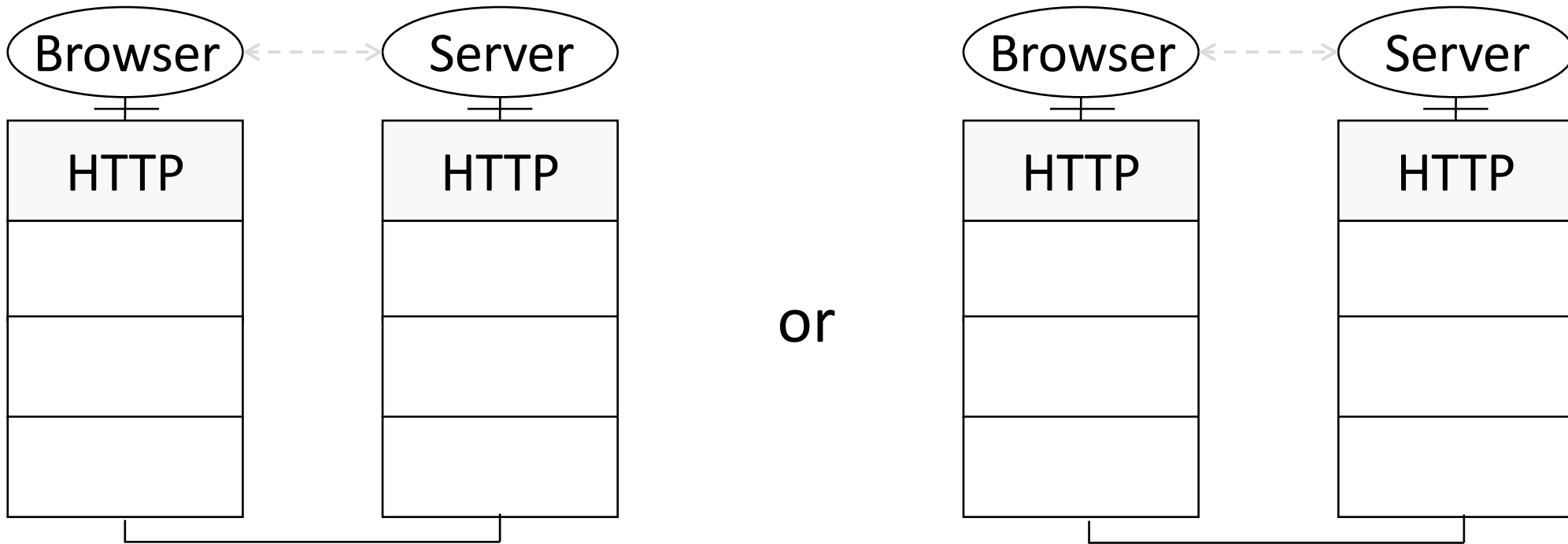
# Demultiplexing (2)

- Done with demultiplexing identifiers in the headers



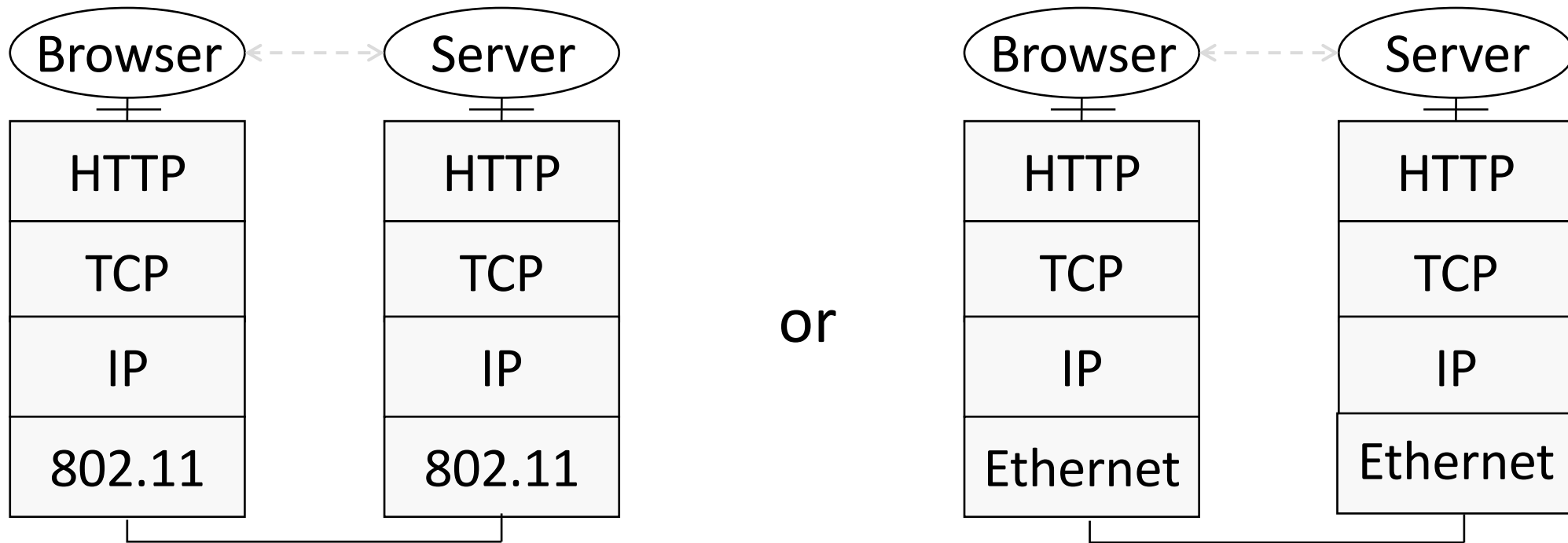
# Advantage of Layering

- Information hiding and reuse



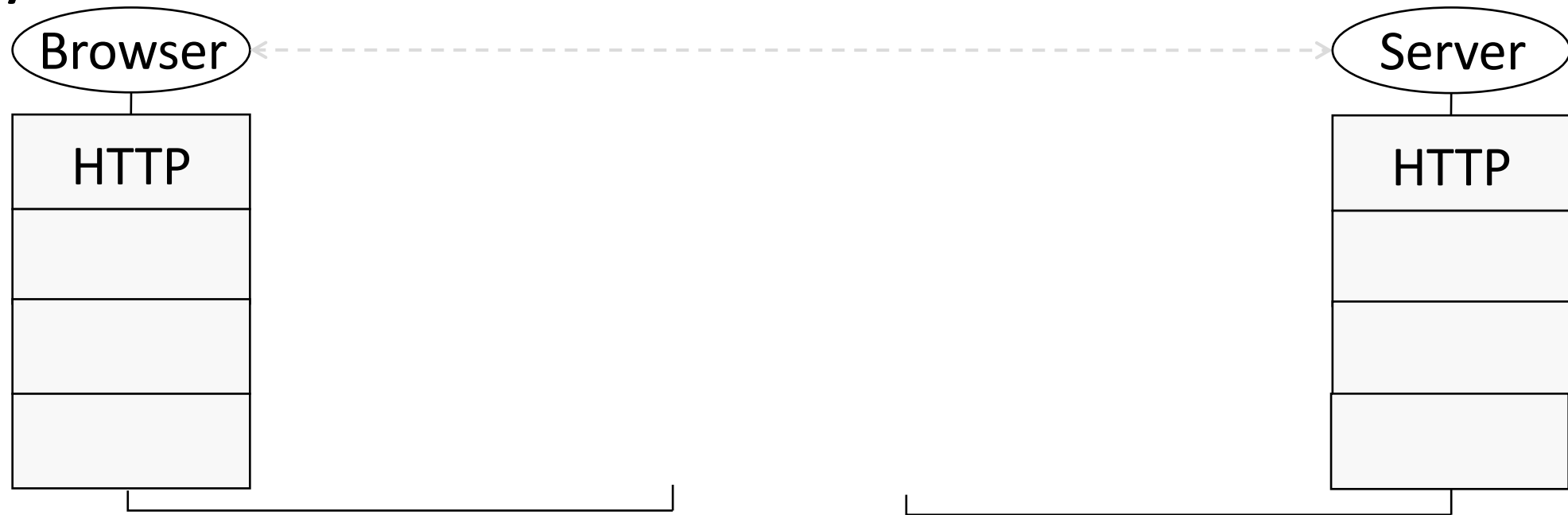
# Advantage of Layering (2)

- Information hiding and reuse



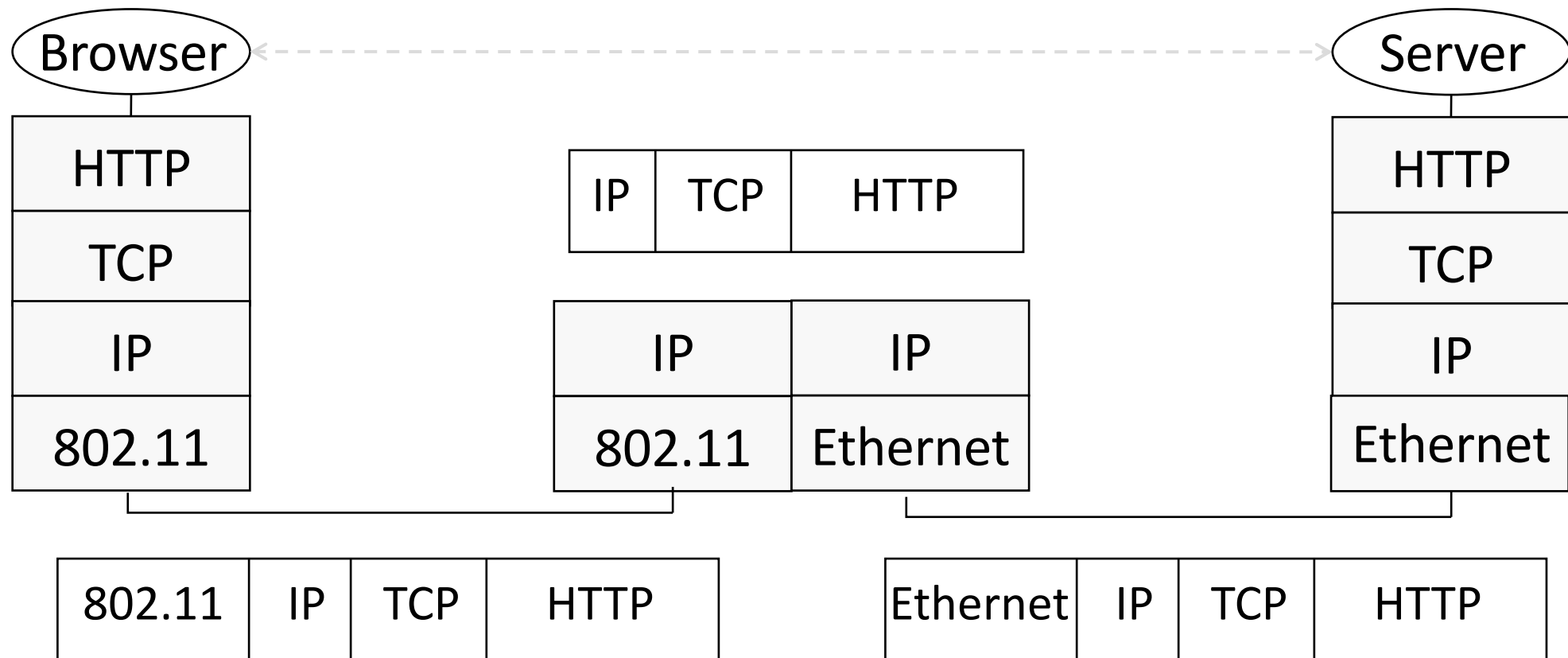
# Advantage of Layering (3)

- Using information hiding to connect different systems



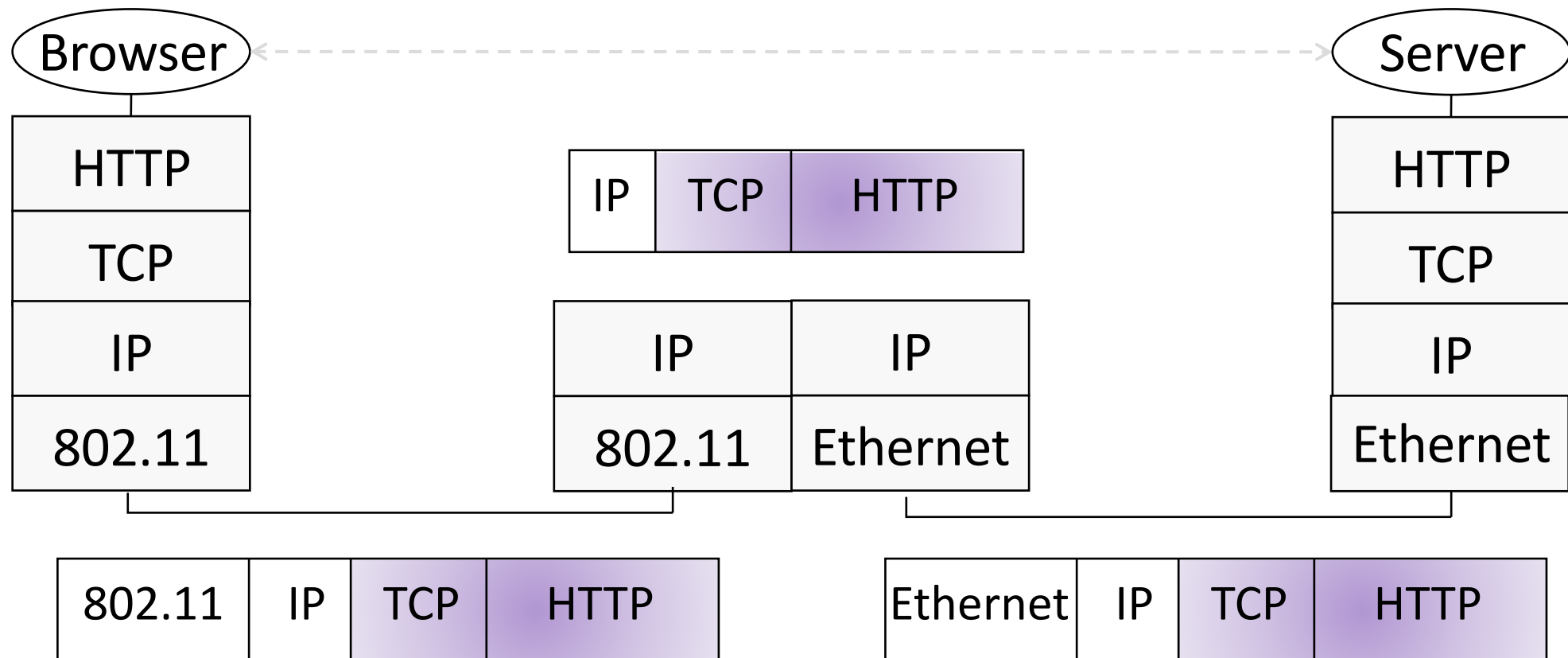
# Advantage of Layering (4)

- Information hiding to connect different systems



# Advantage of Layering (5)

- Information hiding to connect different systems



# Disadvantages of Layering

- ?



# Disadvantage of Layering

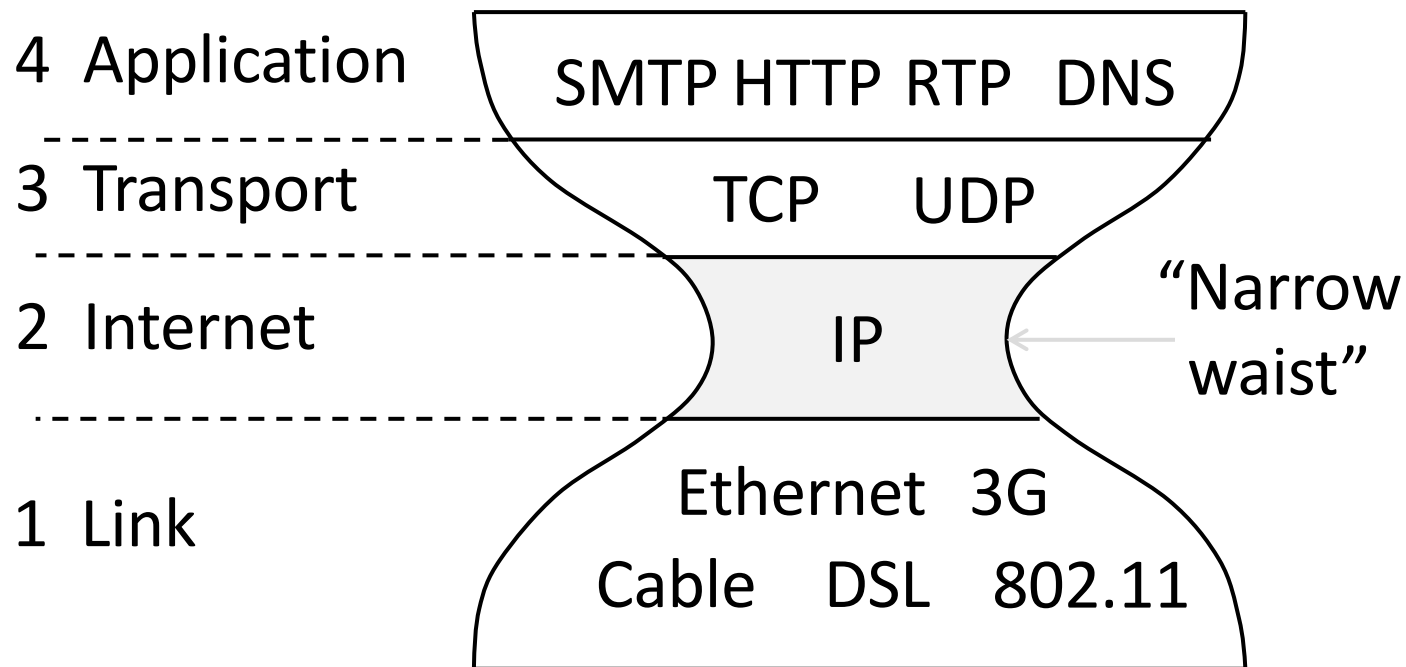
- Adds overhead
  - More problematic with short messages
- Hides information
  - App might care about network properties (e.g., latency, bandwidth, etc)
  - Network may need to know about app priorities (e.g., QoS)

# OSI Layers

Layer	Function	Example
<b>Application (7)</b>	Services that are used with end user applications	SMTP,
<b>Presentation (6)</b>	Formats the data so that it can be viewed by the user  Encrypt and decrypt	JPG, GIF, HTTPS, SSL, TLS
<b>Session (5)</b>	Establishes/ends connections between two hosts	NetBIOS, PPTP
<b>Transport (4)</b>	Responsible for the transport protocol and error handling	TCP, UDP
<b>Network (3)</b>	Reads the IP address from the packet.	Routers, Layer 3 Switches
<b>Data Link (2)</b>	Reads the MAC address from the data packet	Switches
<b>Physical (1)</b>	Send data on to the physical wire.	Hubs, NICs, Cable

# Protocols and Layering

- The real internet protocol stacks:



# Course Reference Model

- We mostly follow the Internet

5	Application	– Programs that use network service
4	Transport	– Provides end-to-end data delivery
3	Network	– Send packets over multiple networks
2	Link	– Send frames over one or more links
1	Physical	– Send bits using signals

# Lecture Progression

Middle → top → bottom

3. Application	- HTTP, DNS, CDNs
1. Transport	- TCP, UDP
2. Network	- IP, NAT, BGP
5. Link	- Ethernet, 802.11
4. Physical	- wires, fiber, wireless

Followed by more detail on cross-cutting elements:

- Quality of service, Security (VPN, SSL)