Protocols and Layers

Networks Need Modularity

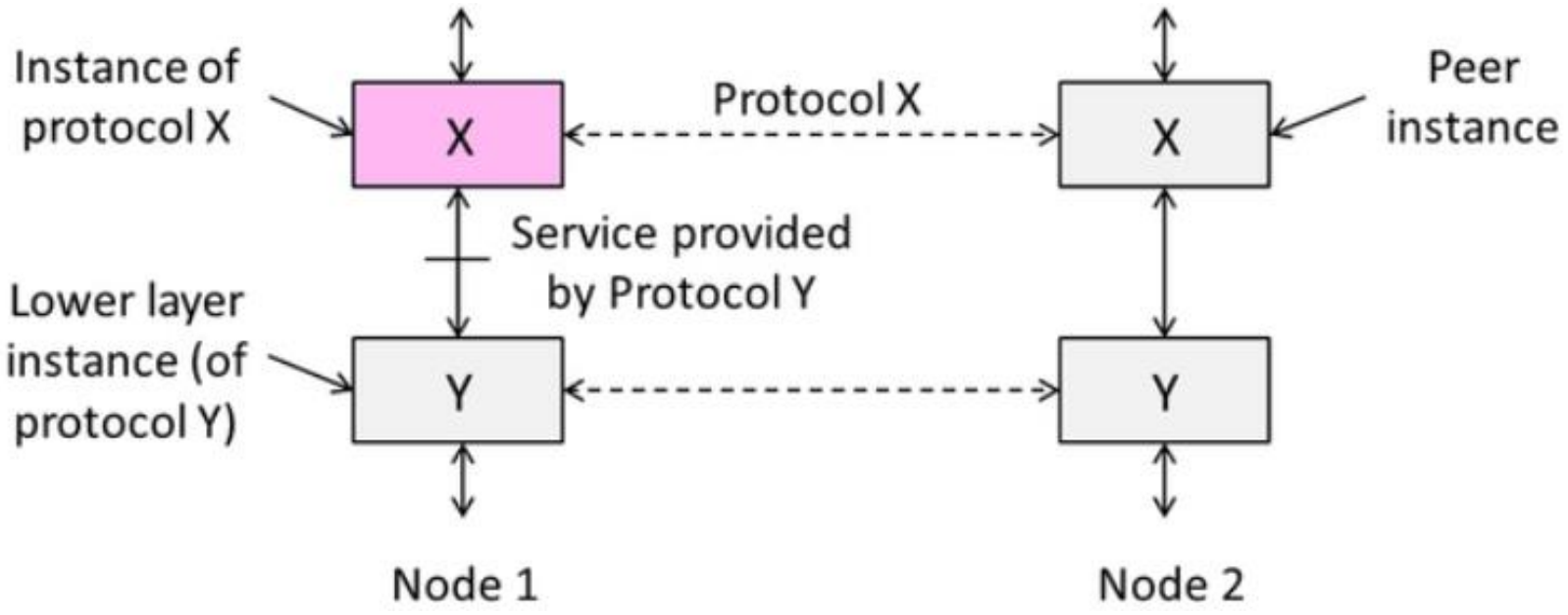
- The Network does much for apps:
  - Make and break connections
  - Find a path through the network
  - Transfer information reliably
  - Transfers arbitrary length information
  - Send as fast as the network allows
  - Share bandwidth among users
  - Secures information in transmit
  - Lets many new hosts be added
  - ….
- We need a form of 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
  - Each instance of a protocol talks virtually to its peer using the protocol
  - Each instance of a protocol uses only the services of lower layer

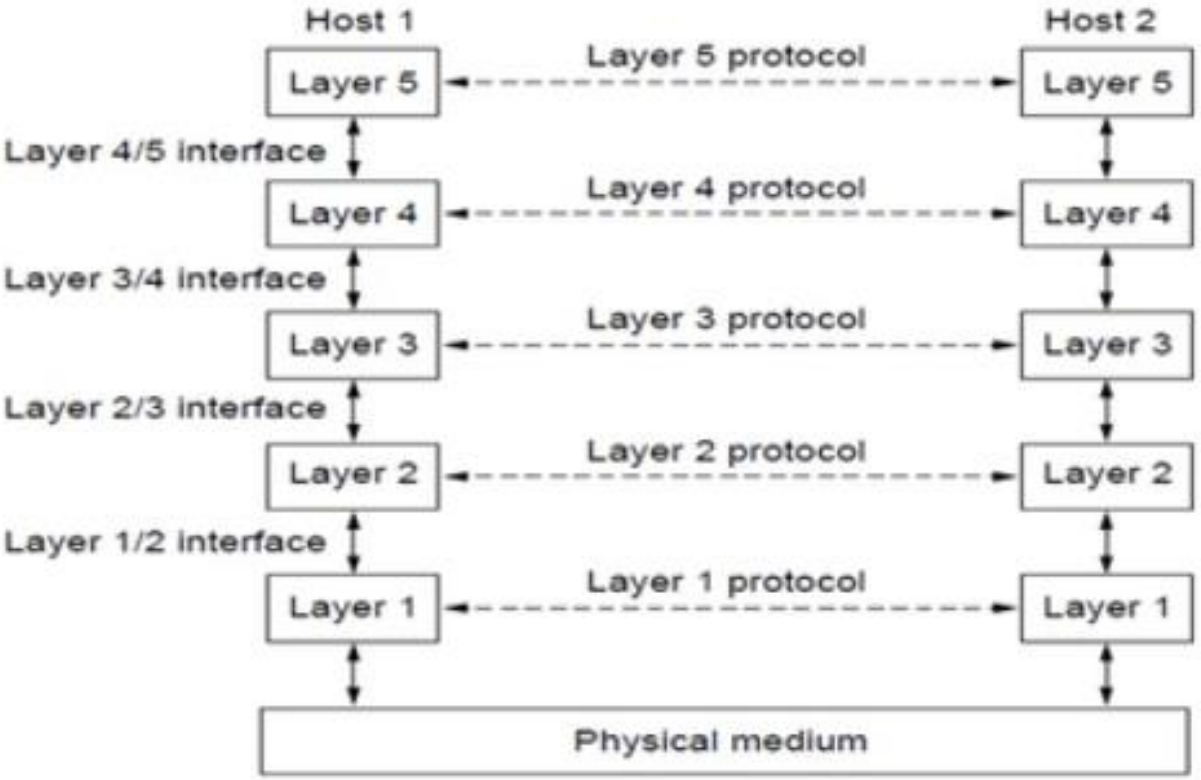
Protocols and Layers (2)

- Protocols are horizonal, layers are vertical



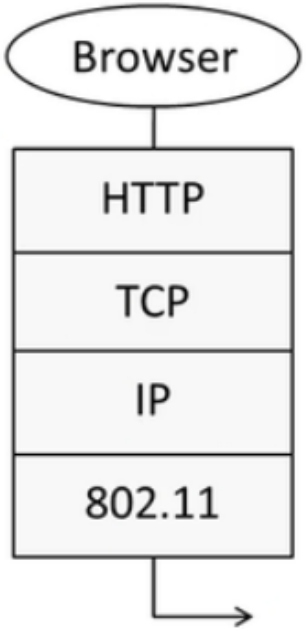
Protocols and Layers (3)

- Set of protocols in use is called a protocol stack



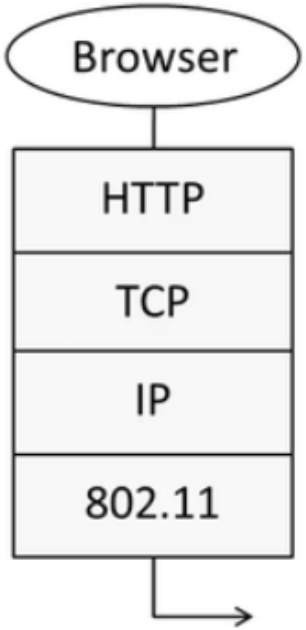
Protocols and Layers (4)

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



Protocols and Layers (4)

- 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

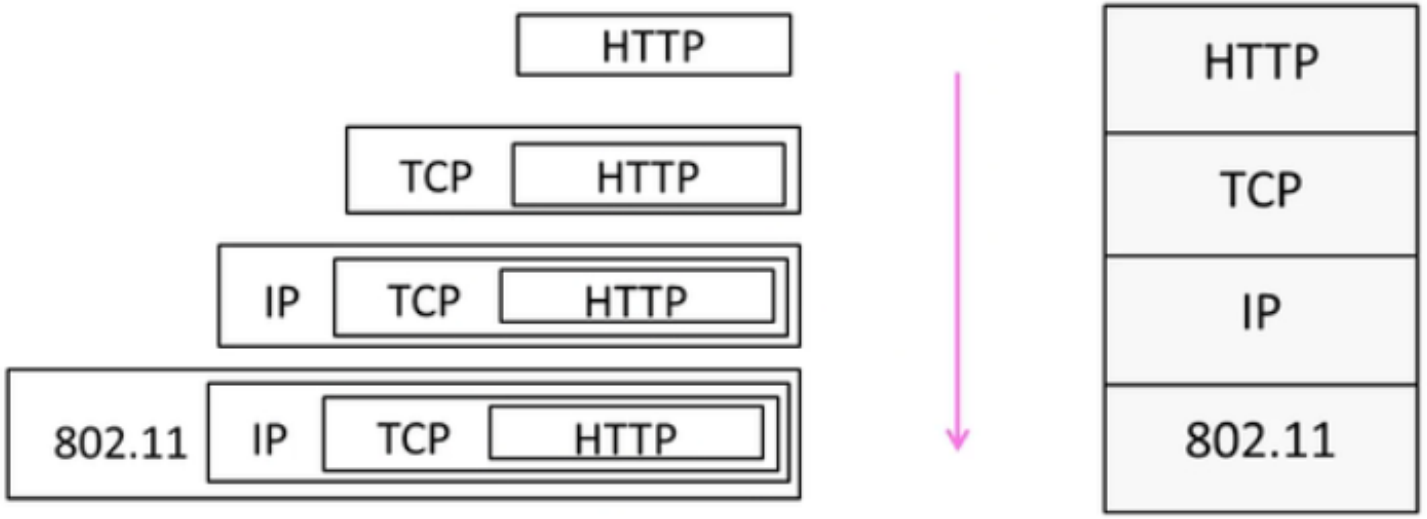


Encapsulation

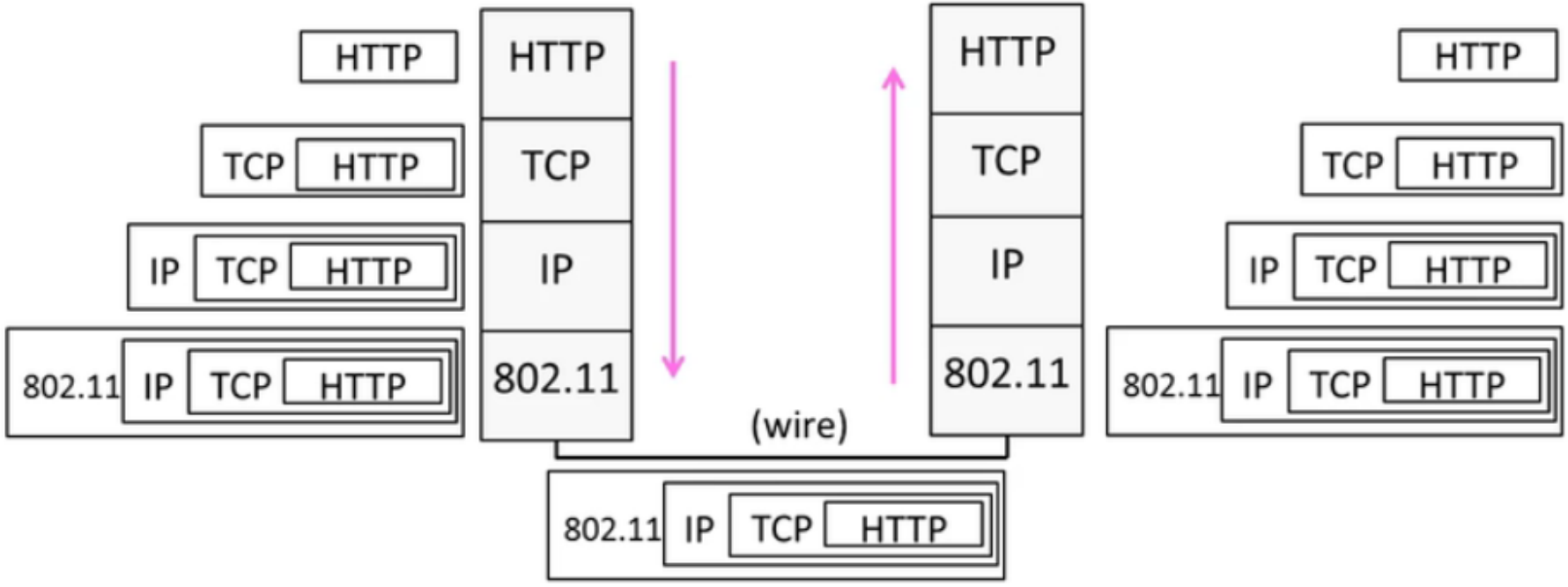
- Encapsulation is the mechanism used to effect protocol layering
- Lower layer wraps higher layer content, adding its own information to make a new message for delivery
- Like sending a letter in an envelope; postal service doesn’t look inside

Encapsulation (2)

- Messages "on the wire" begins to look like an onion
  - —Lowers layers are the outermost

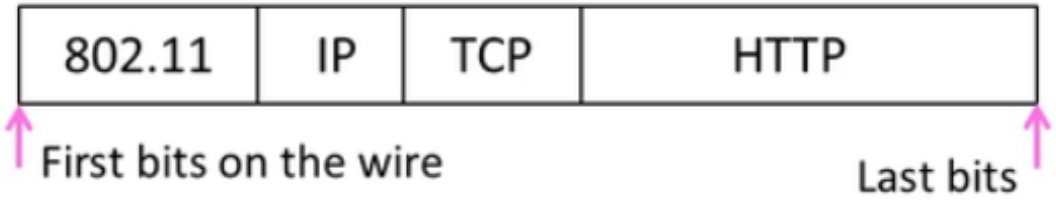


Encapsulation (3)



Encapsulation (4)

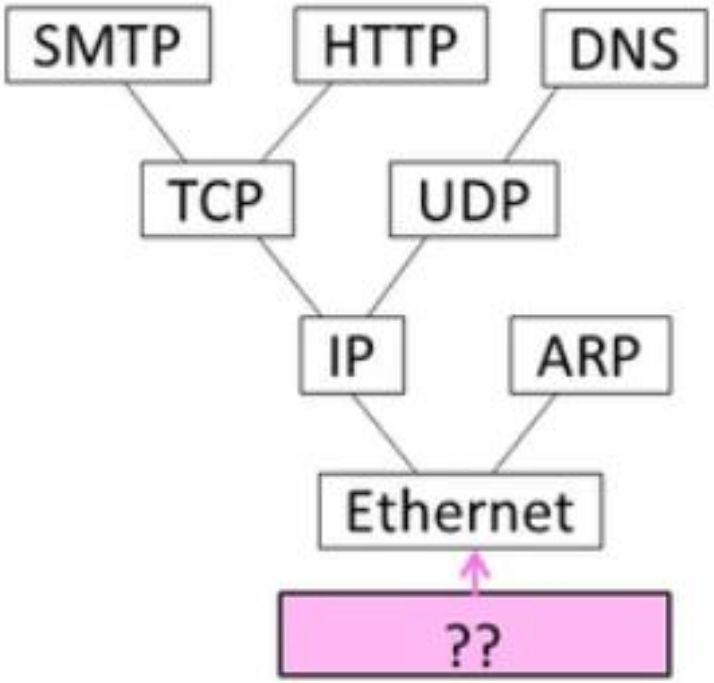
- Normally message draw like this:
  - —Each layer adds its own header



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

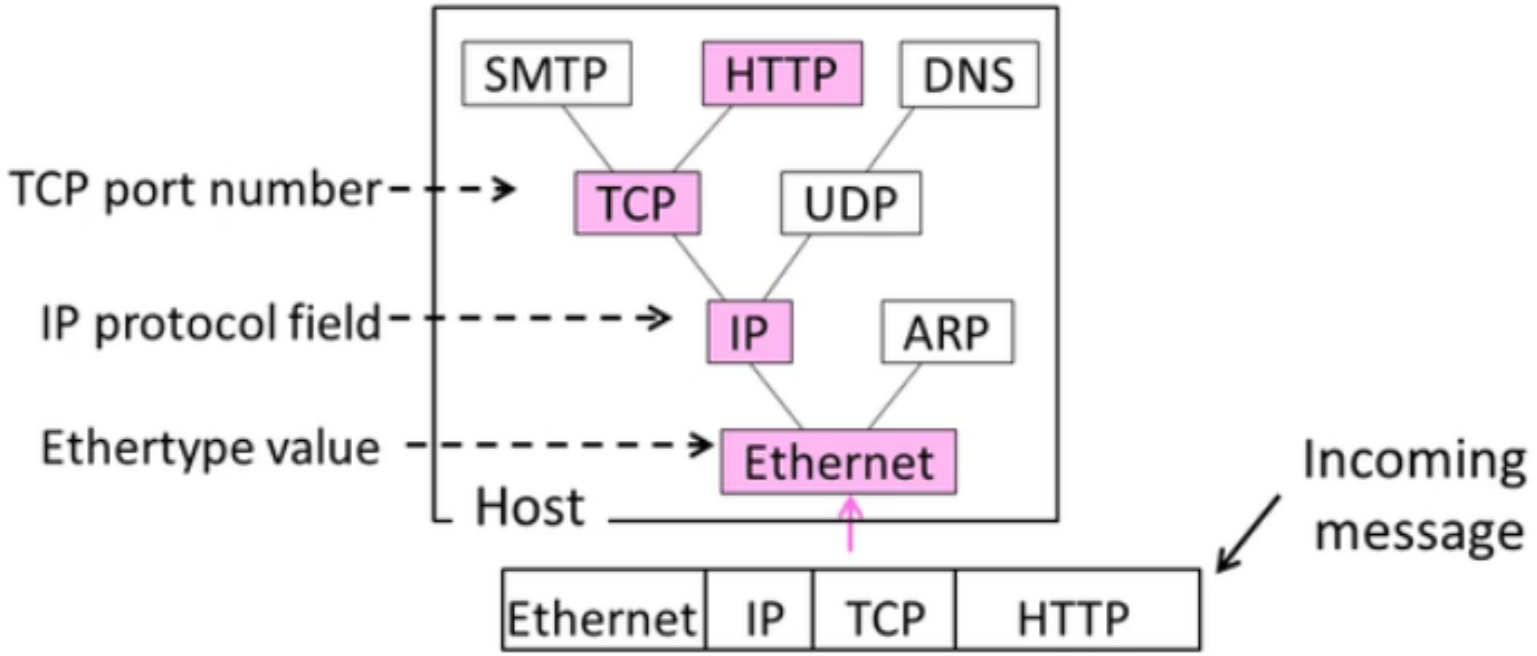
Demultiplexing

- Incoming message must be passed to the protocols that it uses



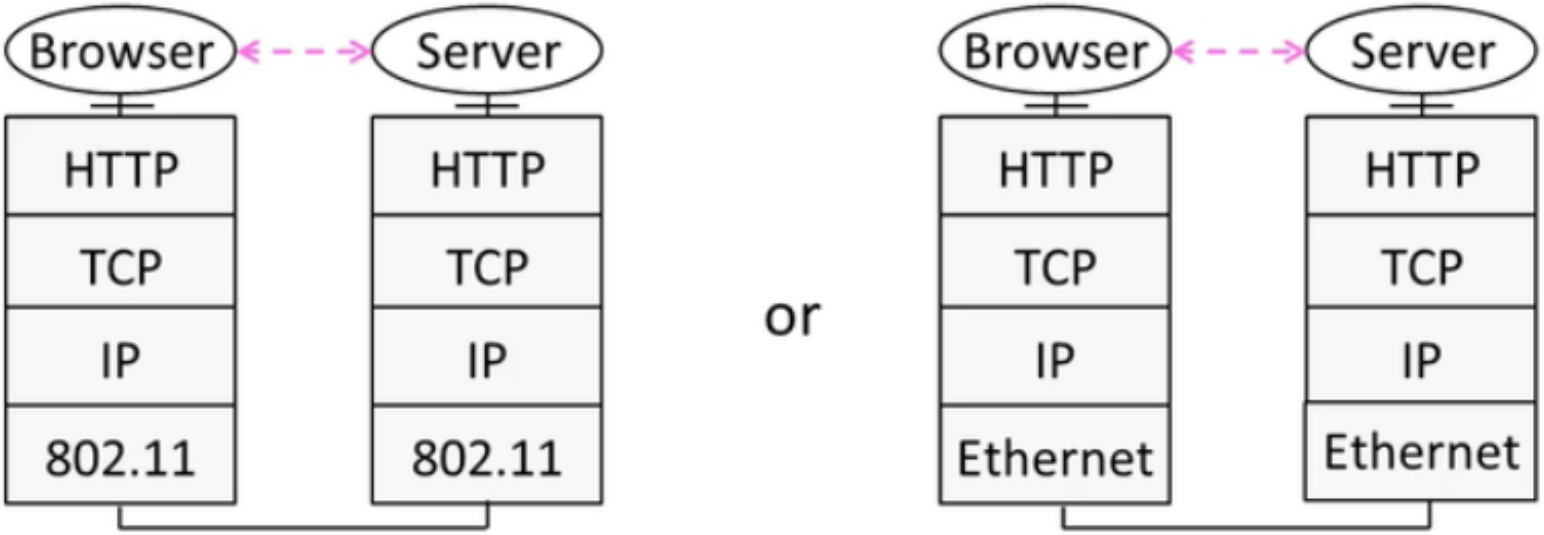
Demultiplexing (2)

- Done with demultiplexing keys in the header



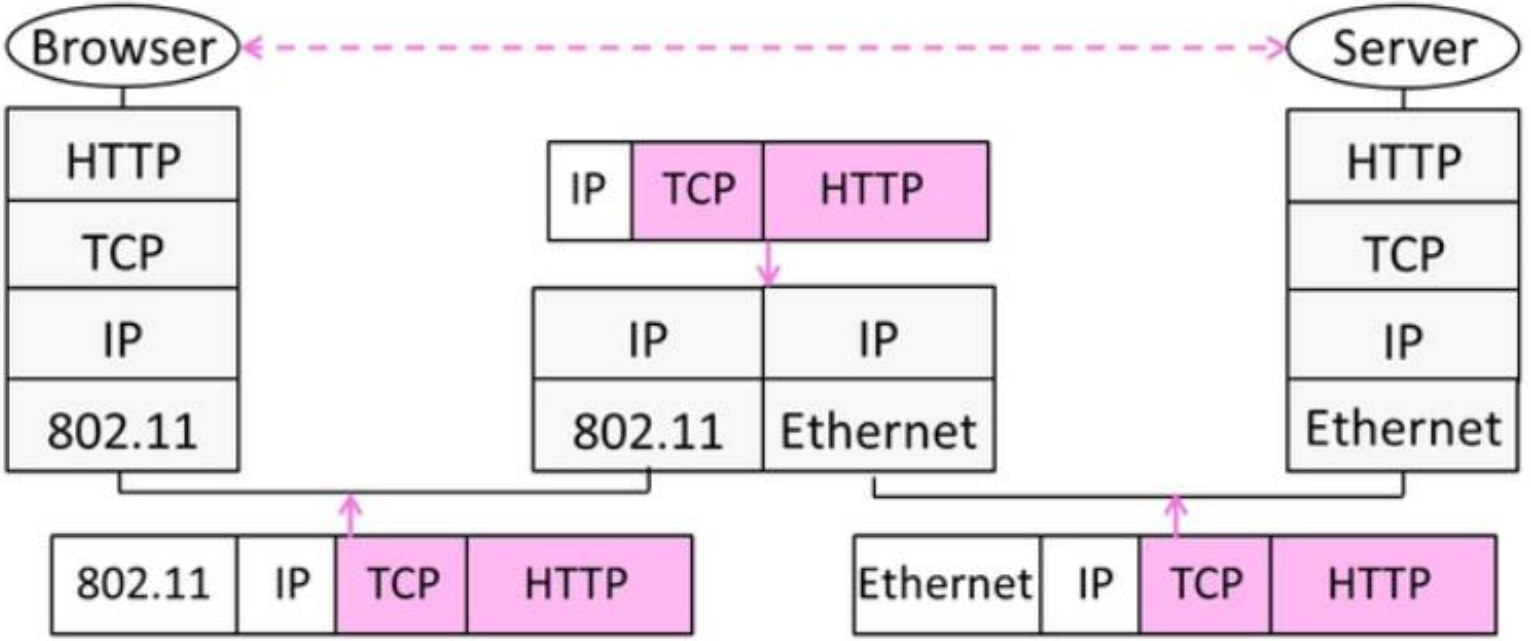
Advantage of layering

- Information hiding and reuse



Advantage of layering (2)

- Using information hiding to connect different systems



Disadvantage of layering

- Adds overhead
  - —But minor for long messages
- Hides information
  - —App might care whether it is running over wired or wireless!