Computer Networks

Protocols and Layering (§1.3)

David Wetherall (djw@uw.edu)
Professor of Computer Science & Engineering
UNIVERSITY of WASHINGTON

Networks Need Modularity

- The network does much 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
  - …

Networks Need Modularity

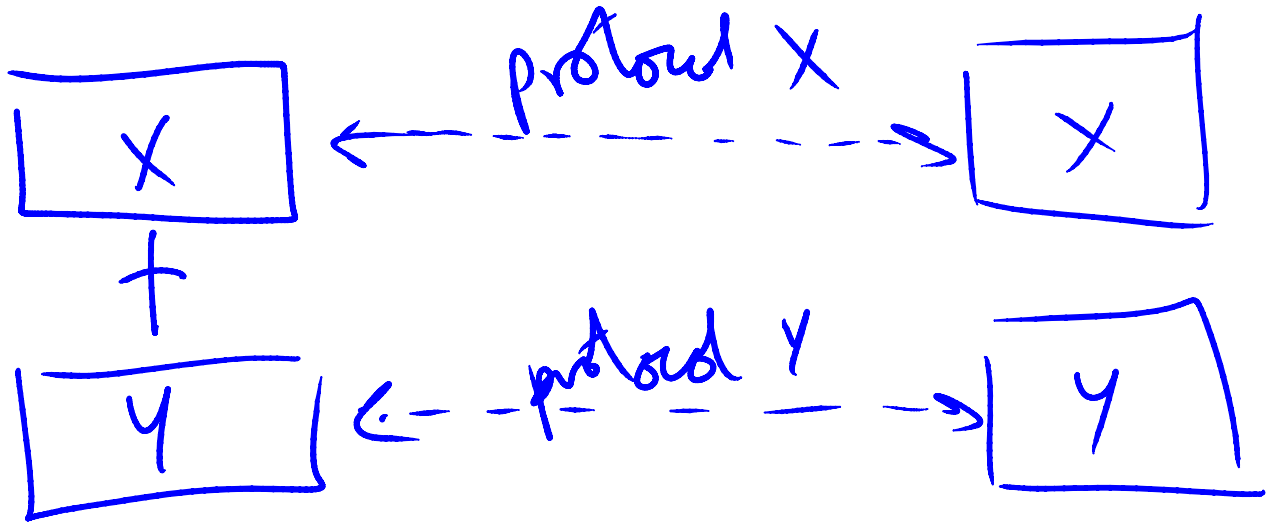
- The network does much for apps:
  - Make and break connections
  - ork
  - mation
  - ows
  - s
  - Secures information in transit
  - Lets many new hosts be added
  - …

We need a form of modularity, to help manage complexity and support reuse

Protocols and Layers

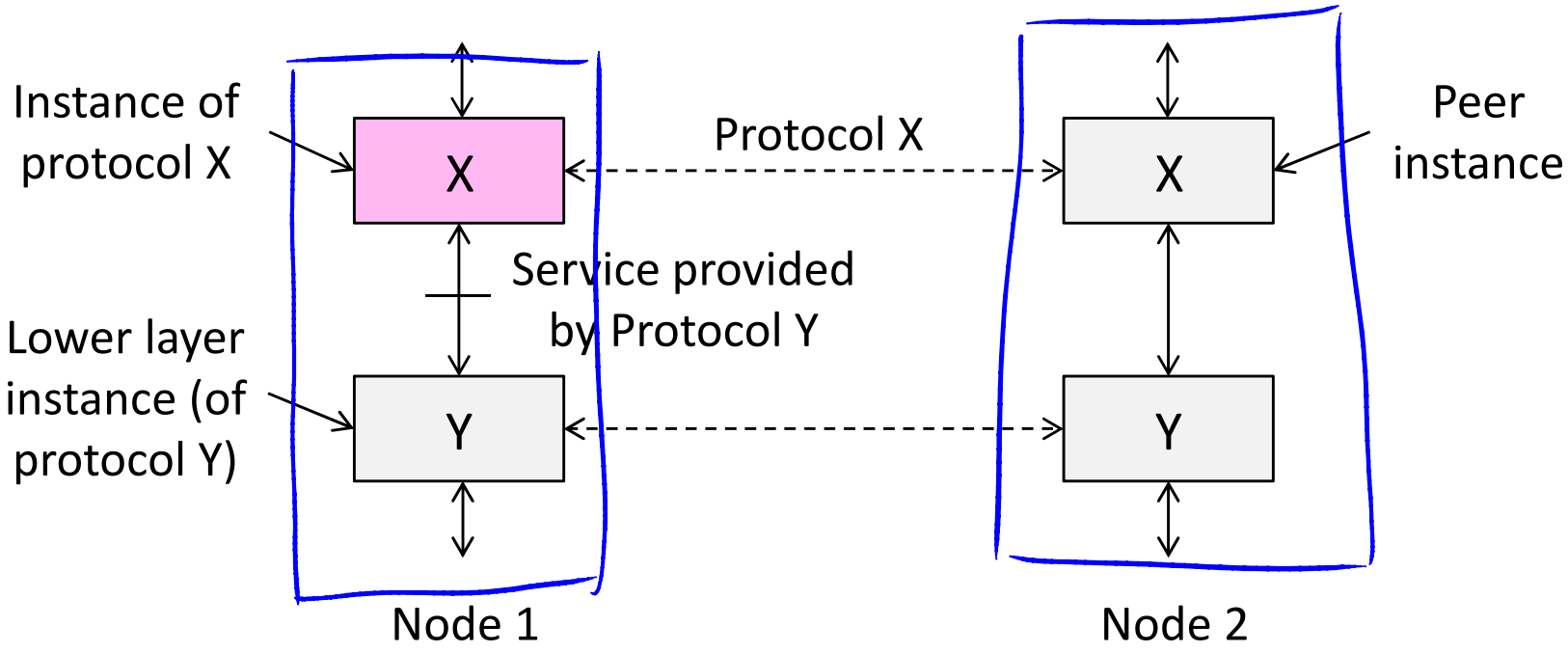
- Protocols and layering is the main structuring method used to divide up network functionality
  - Each instance of a protocol talks virtually to its peer using the protocol
  - Each instance of a protocol uses only the services of the lower layer

Protocols and Layers (2)



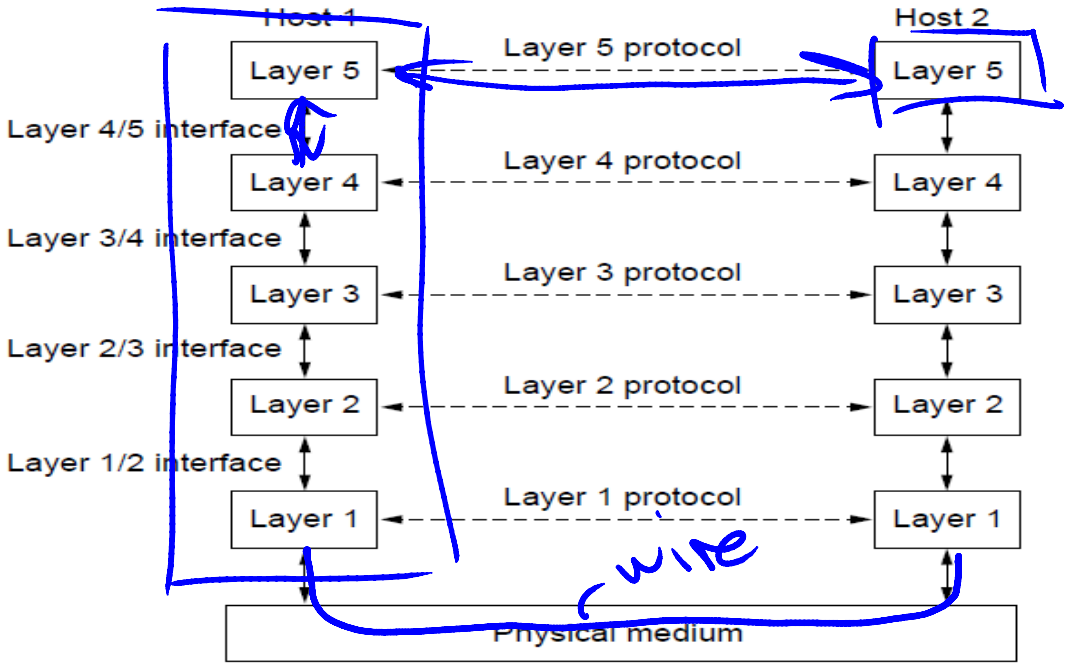
Protocols and Layers (3)

- Protocols are horizontal, layers are vertical



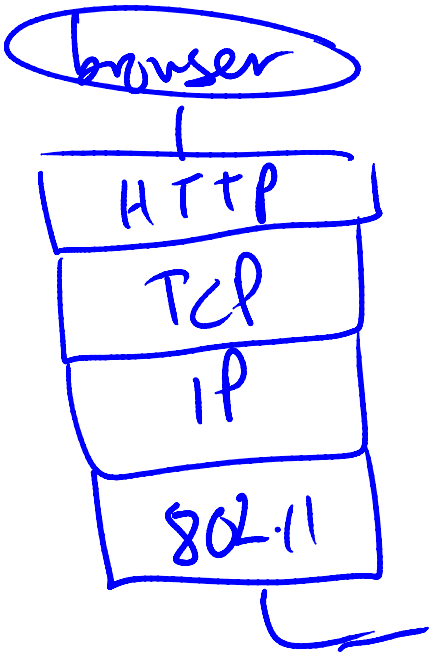
Protocols and Layers (4)

- Set of protocols in use is called a protocol stack



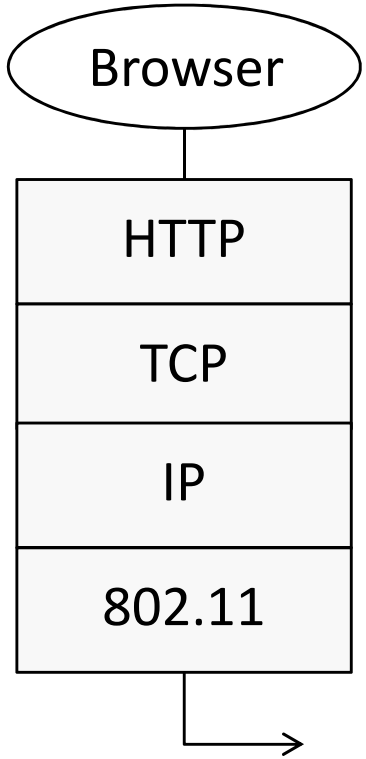
Protocols and Layers (5)

- 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



Protocols and Layers (6)

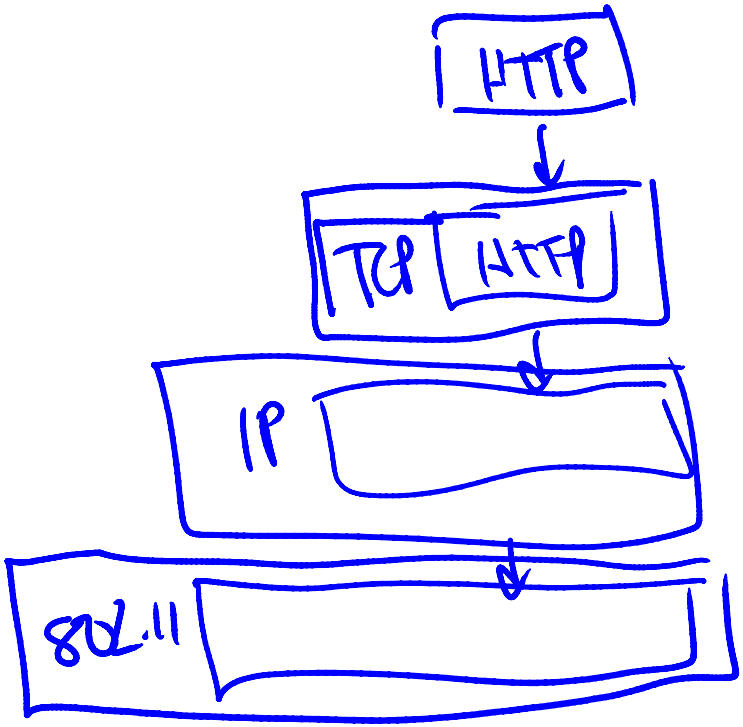
- Protocols you've probably heard of:
  - TCP, IP, 802.11, Ethernet, HTTP, SSL, DNS, … and many more
- An example protocol stack
  - Used by a web browser on a host that is wirelessly connected to the Internet

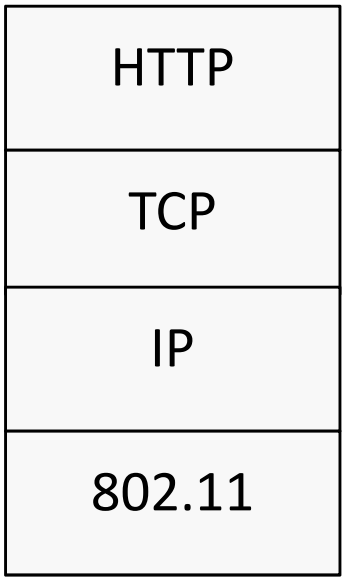


Encapsulation

- 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

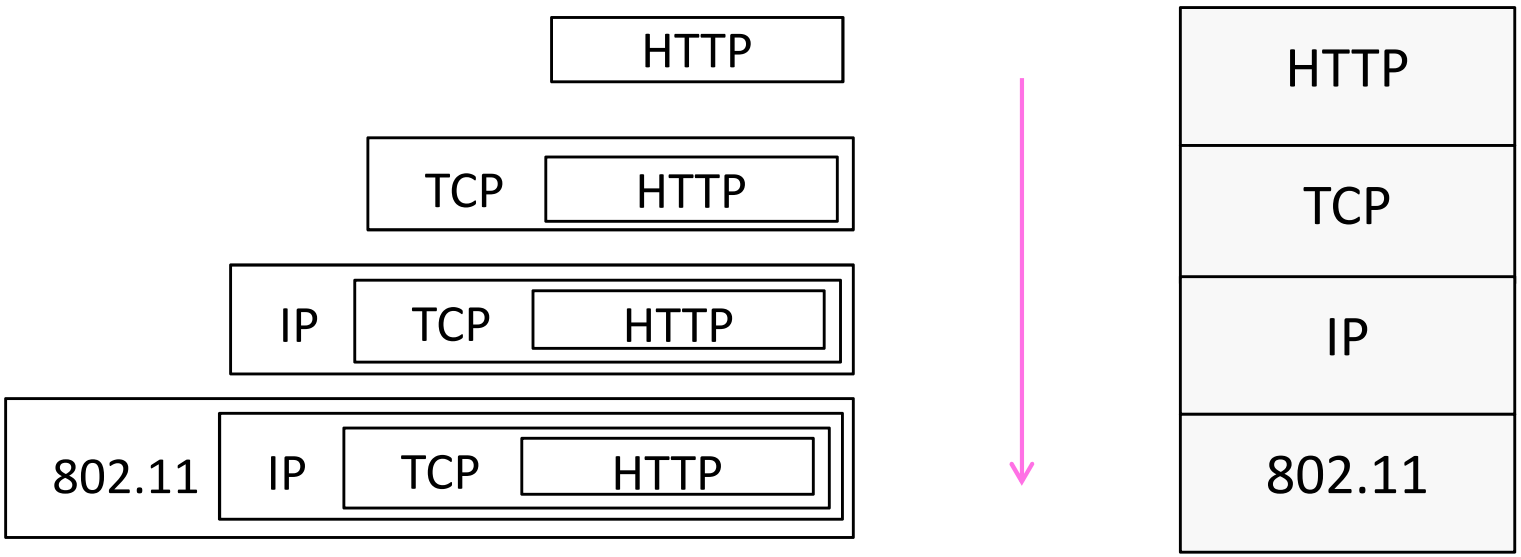
Encapsulation (2)



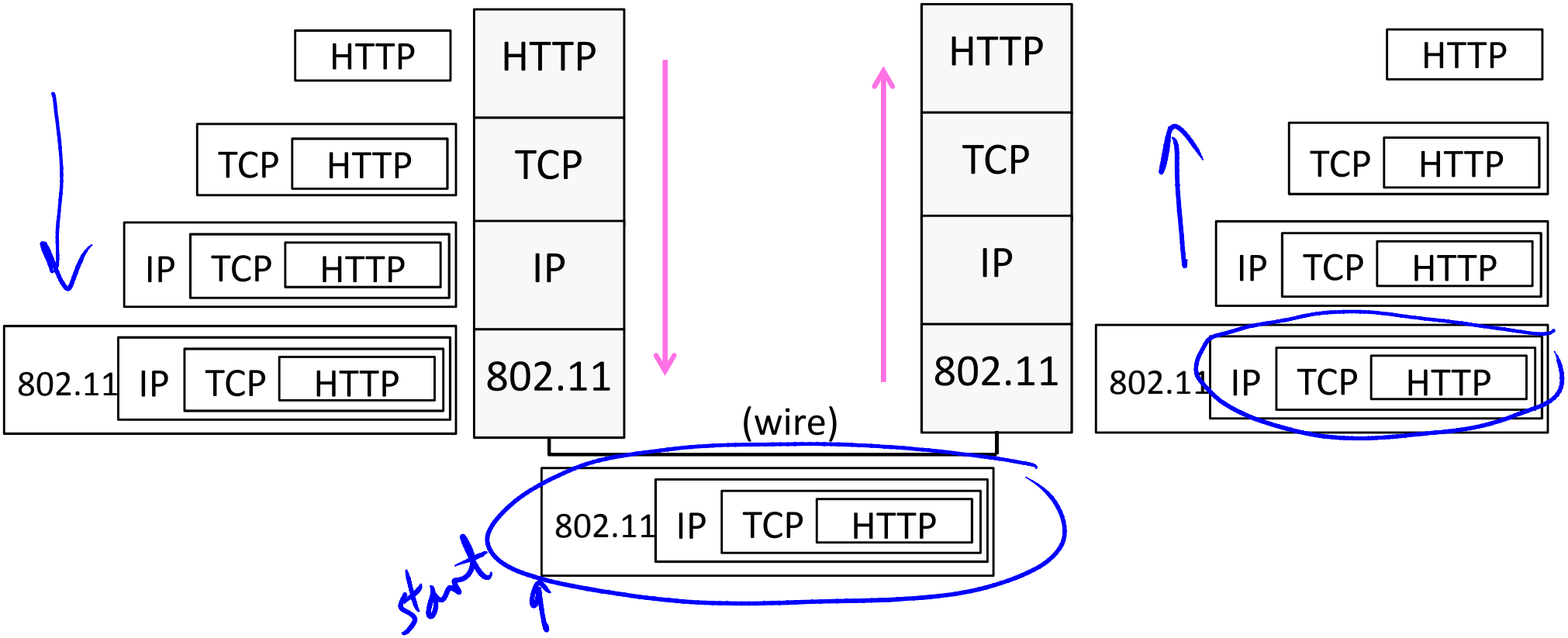


Encapsulation (3)

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

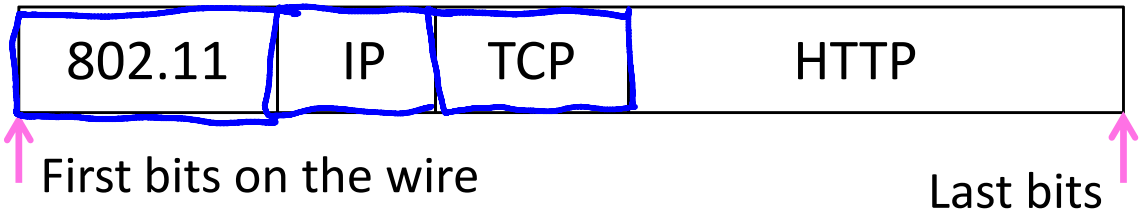


Encapsulation (4)



Encapsulation (5)

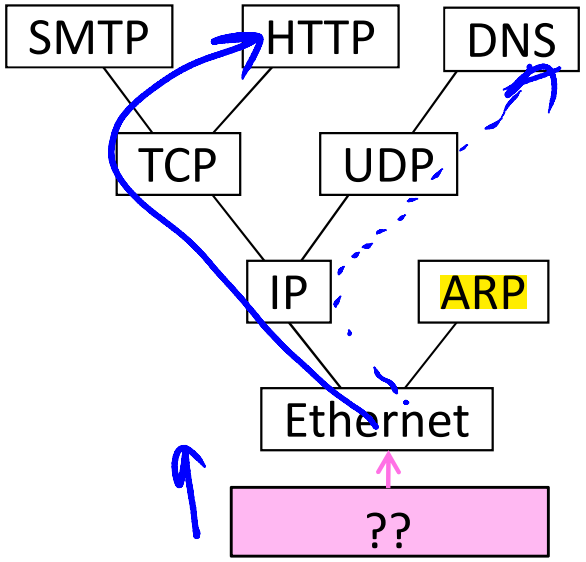
- Normally draw message like this:
  - Each layer adds its own header



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

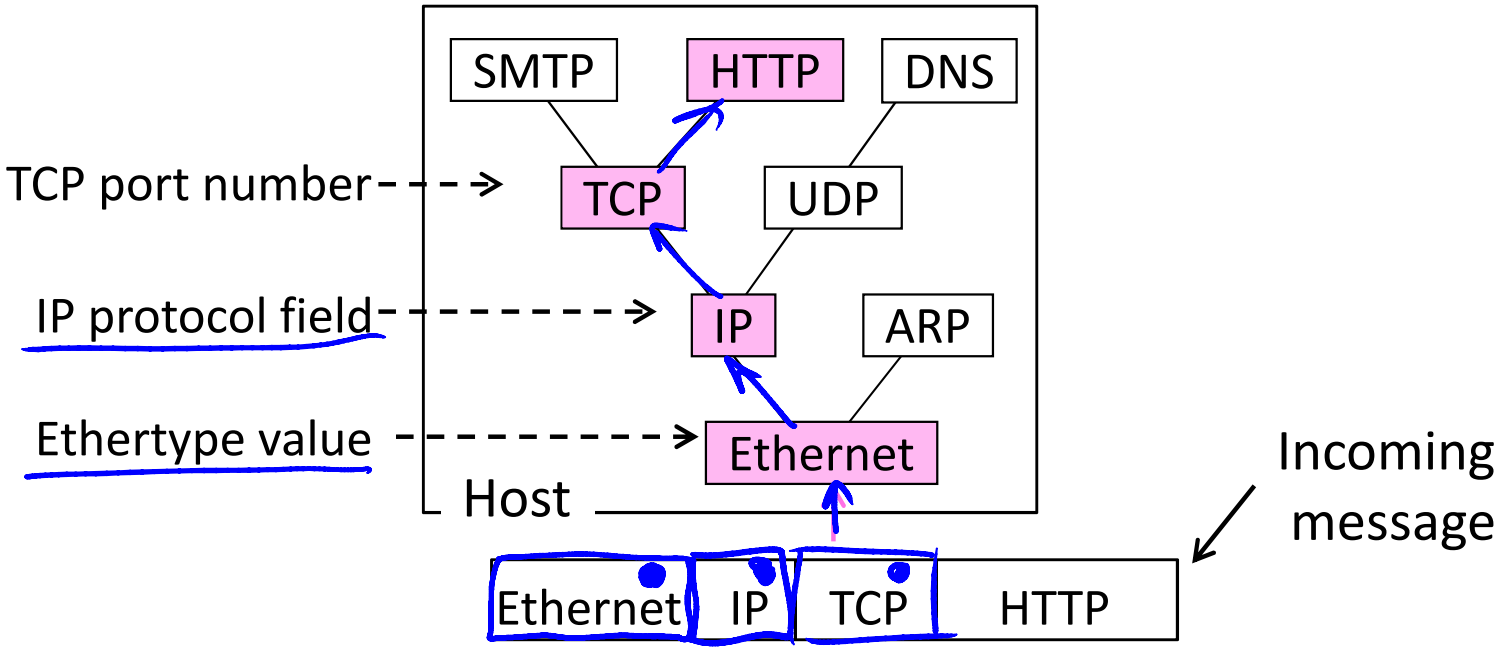
Demultiplexing

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



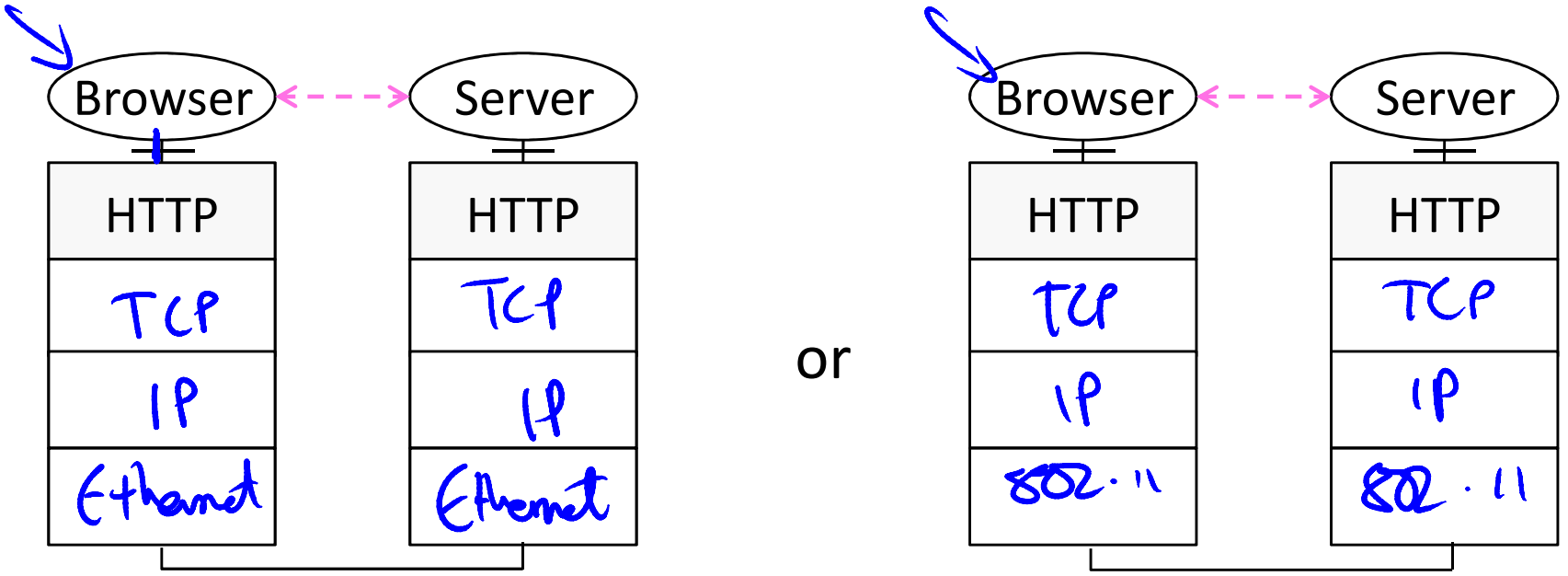
Demultiplexing (2)

- Done with demultiplexing keys in the headers



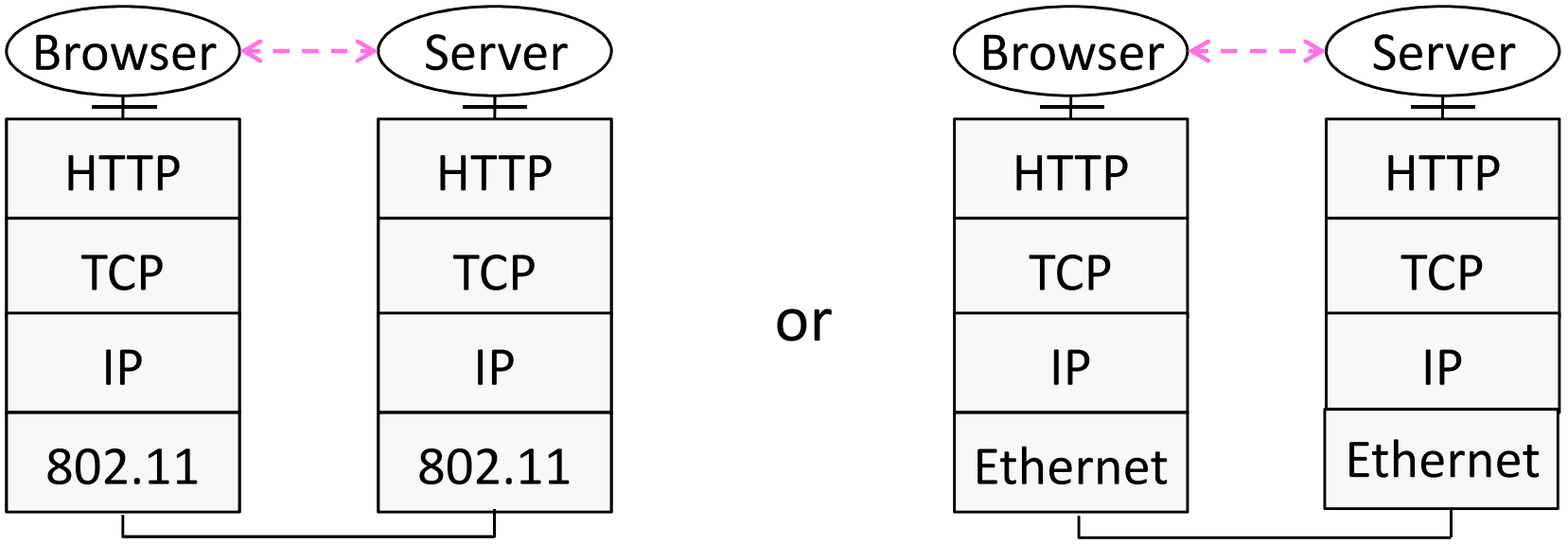
Advantage of Layering

- Information hiding and reuse



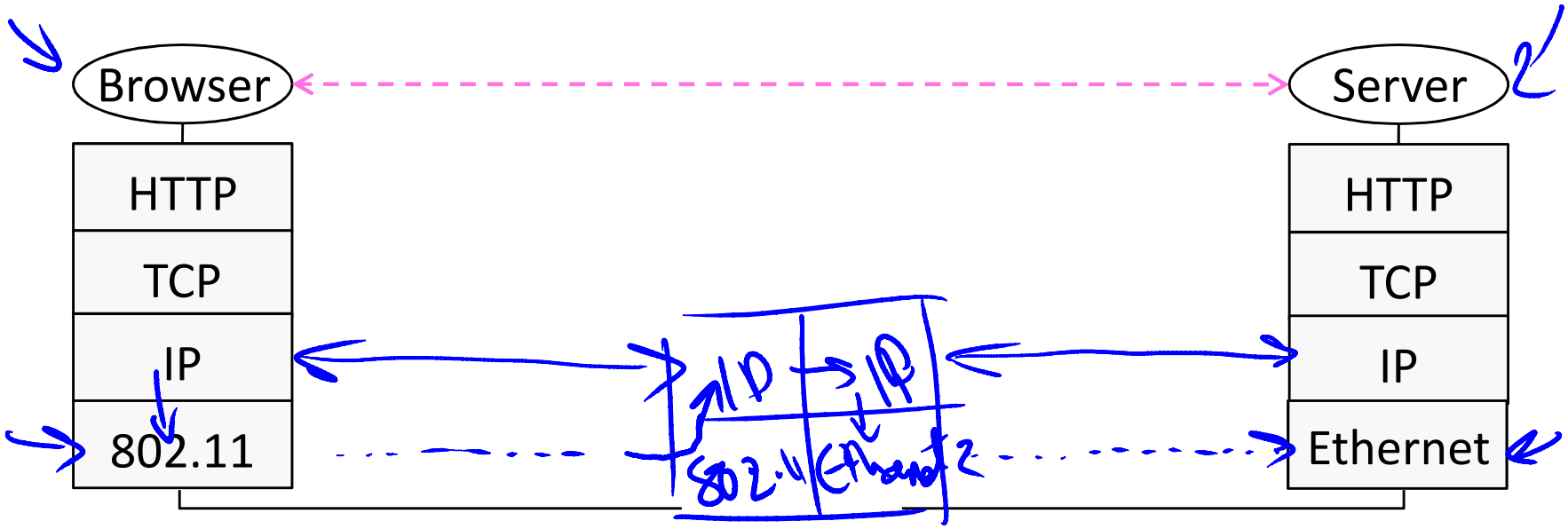
Advantage of Layering (2)

- Information hiding and reuse



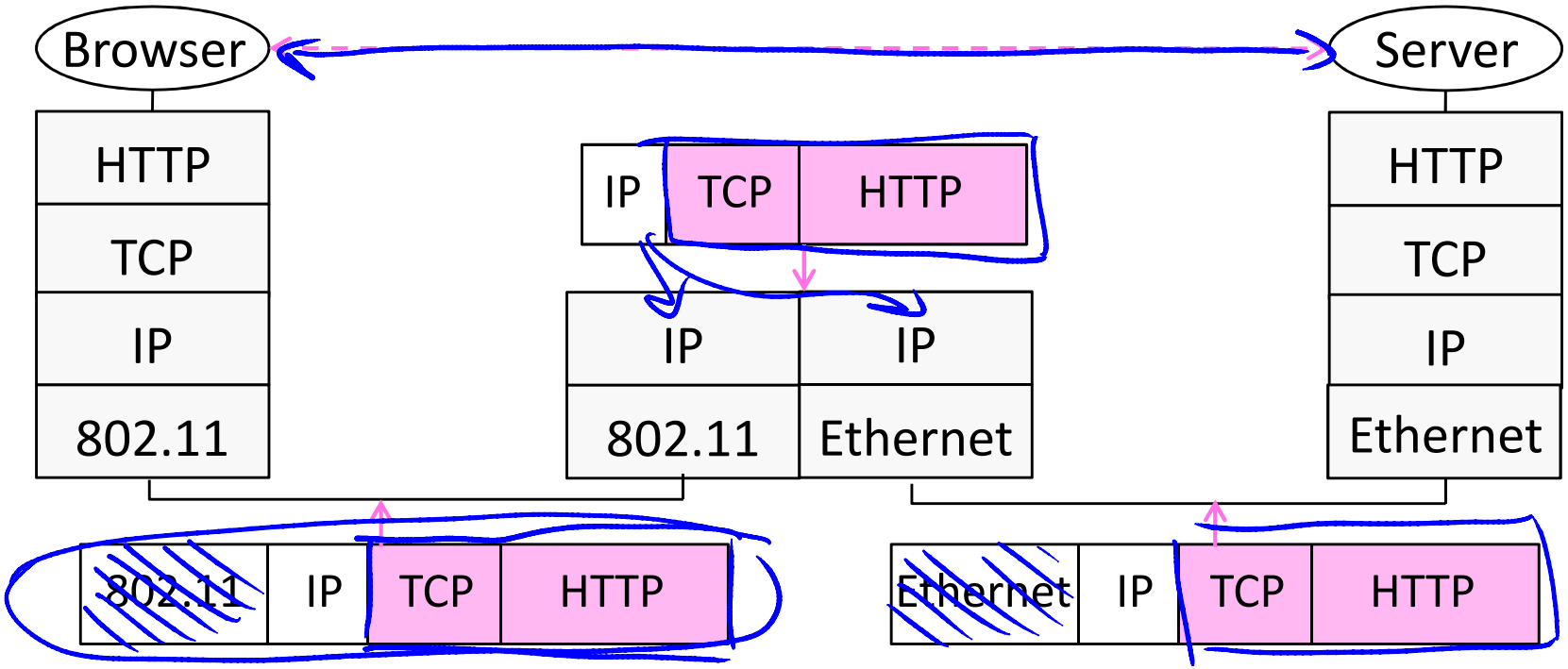
Advantage of Layering (3)

- Using information hiding to connect different systems



Advantage of Layering (4)

- 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!

END