CS419: Computer Networks

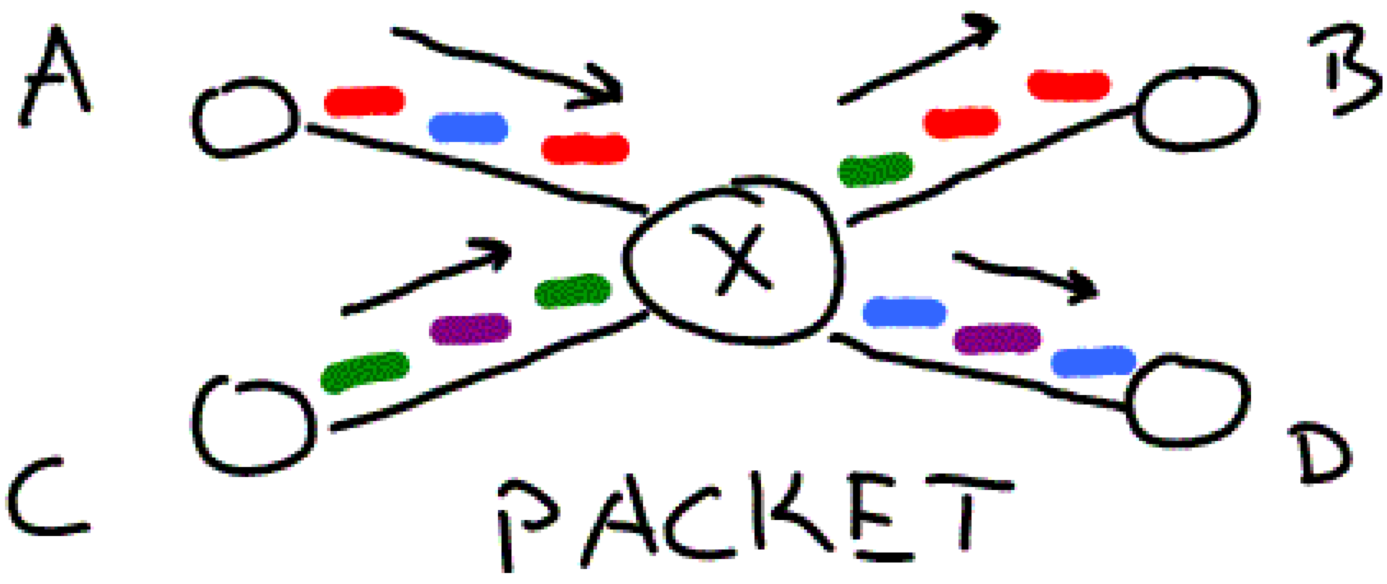
Lecture 1 (part 2): Jan 26, 2004

Intro to Computer Networking

Remember this picture?

CS419

- How did the switch know to forward some packets to B and some to D?

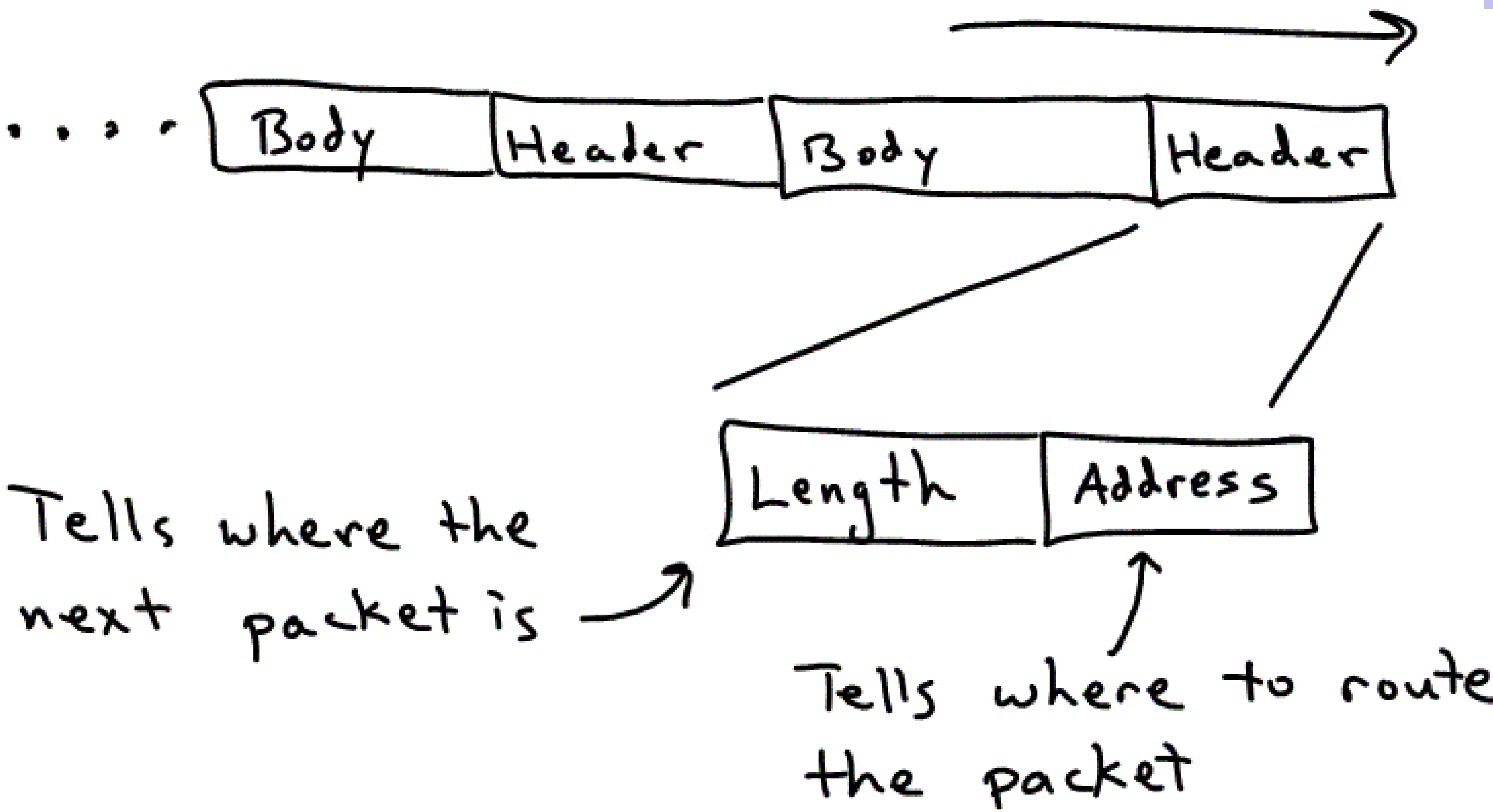


From the address in the packet header…

- A packet has a header and a body
  - and, sometimes, a trailer
- The header says:
  - Where the packet is going (*address*)
  - How big the packet is (*length*)
  - Some other stuff

Packets

CS419



Like an envelope?

CS419

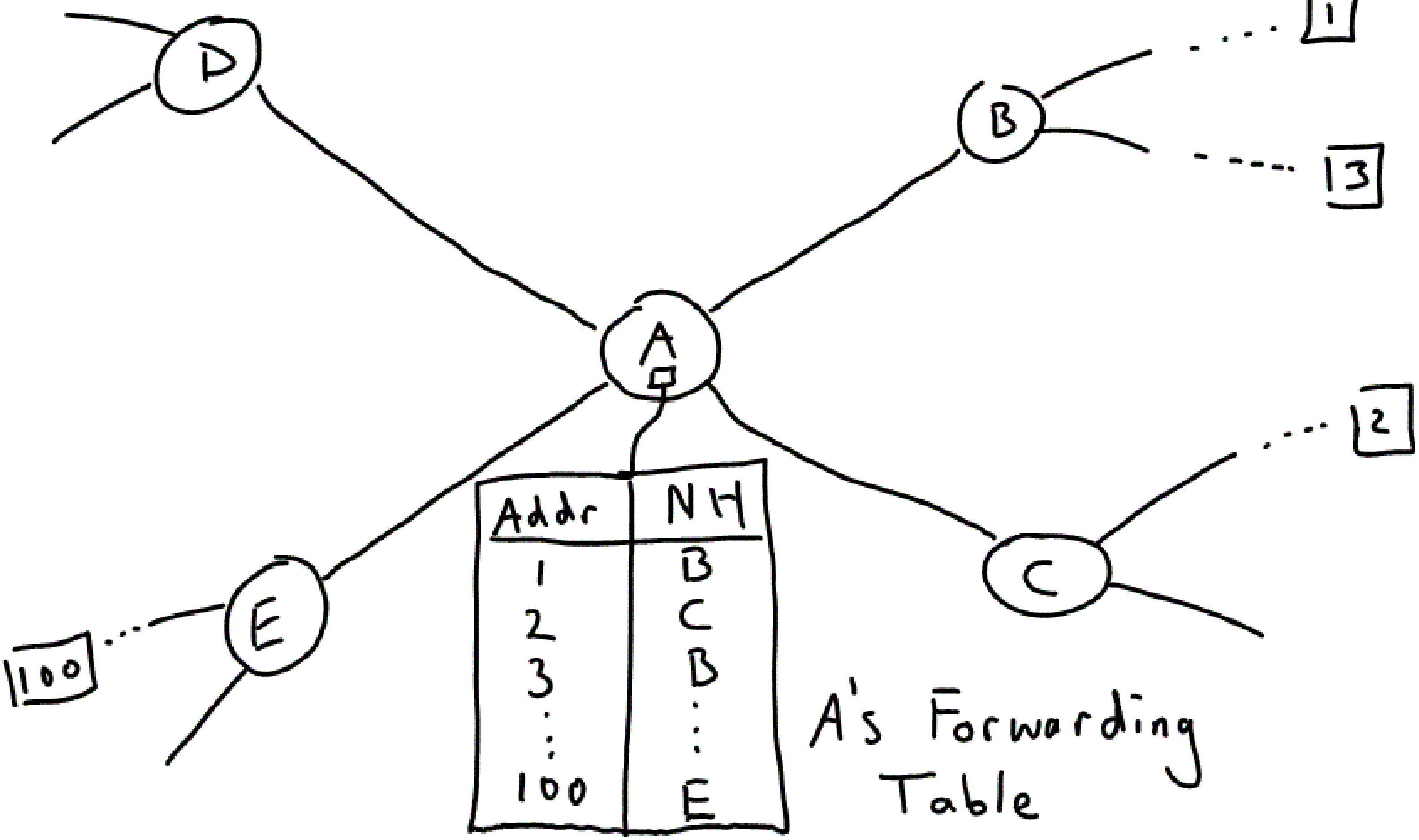
- The address field is somewhat analogous to the address on an envelope
  - And the contents of the envelope would then be like the packet body
- But this analogy doesn't work for the length field!

Forwarding Table

- Routers (or switches) have a forwarding table
  - Router is a forwarding box that operates on IP packets
- This table is indexed by the address in the header, and tells which next hop to send the packet to
- Addresses can be hierarchical (like phone numbers)

Forwarding Table

CS419



Forwarding tables and routing algorithms

CS419

- How did the forwarding table get there?
- Typically a routing algorithm is run among the routers, and this algorithm establishes the contents of the forwarding table
- In this class, we'll look in detail at address structures and routing algorithms

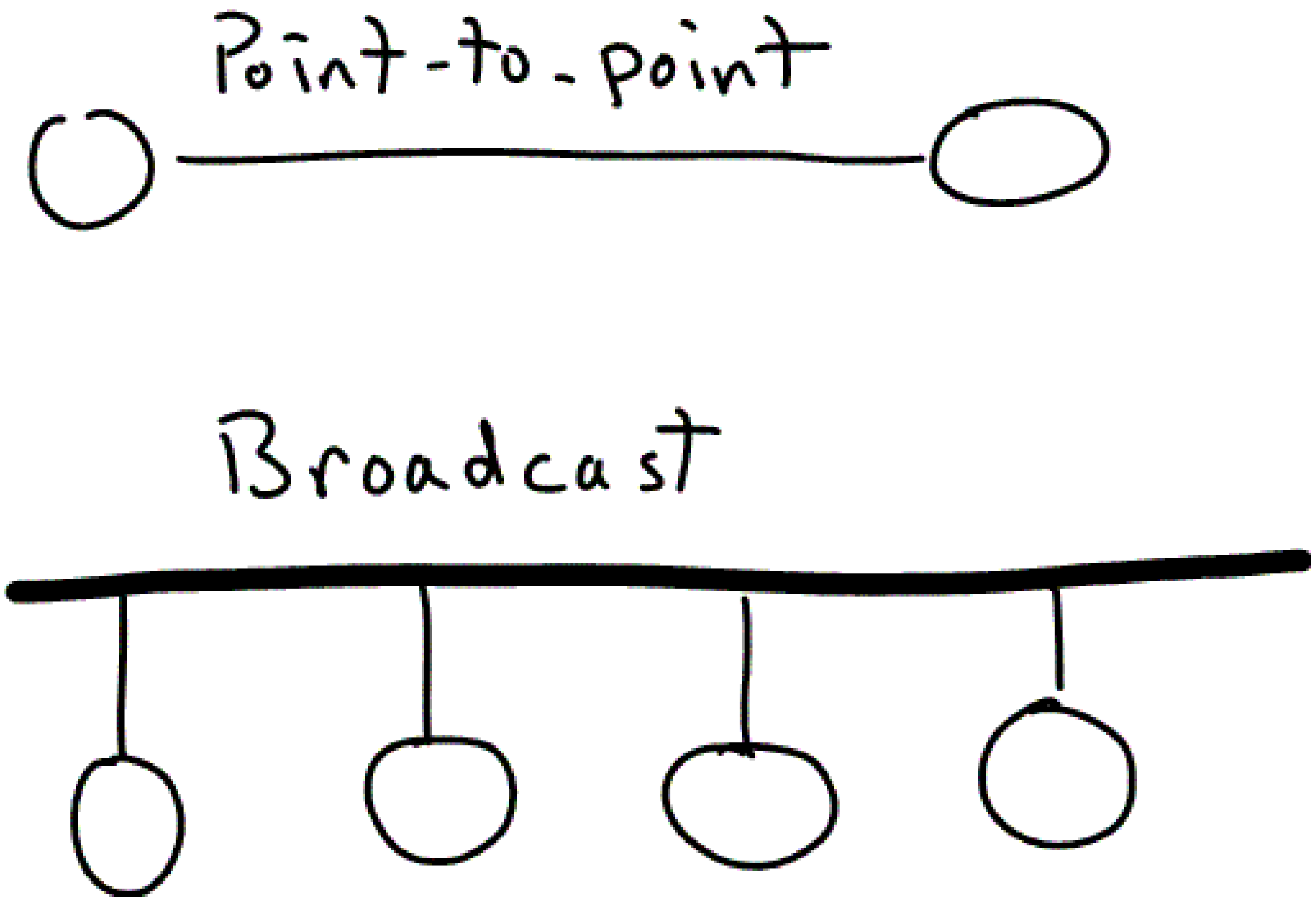
Two kinds of links

CS419

- Routers and hosts in the Internet are typically connected by two types of links
- We’ve been looking at pictures of point-to-point links
- The other common kind is the broadcast link
  - Usually Ethernet

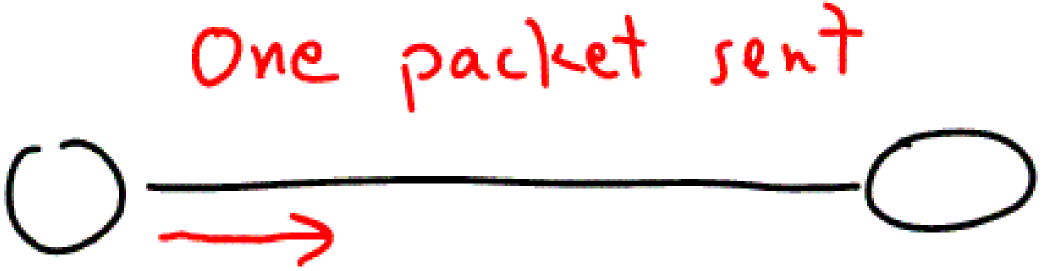
Point-to-point and broadcast links

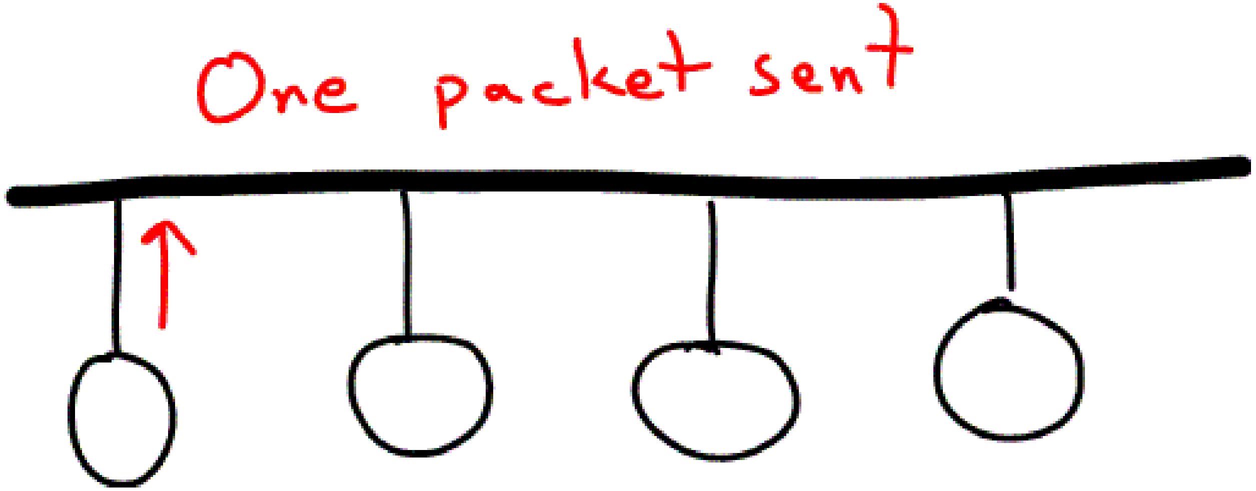
CS419



Point-to-point and broadcast links

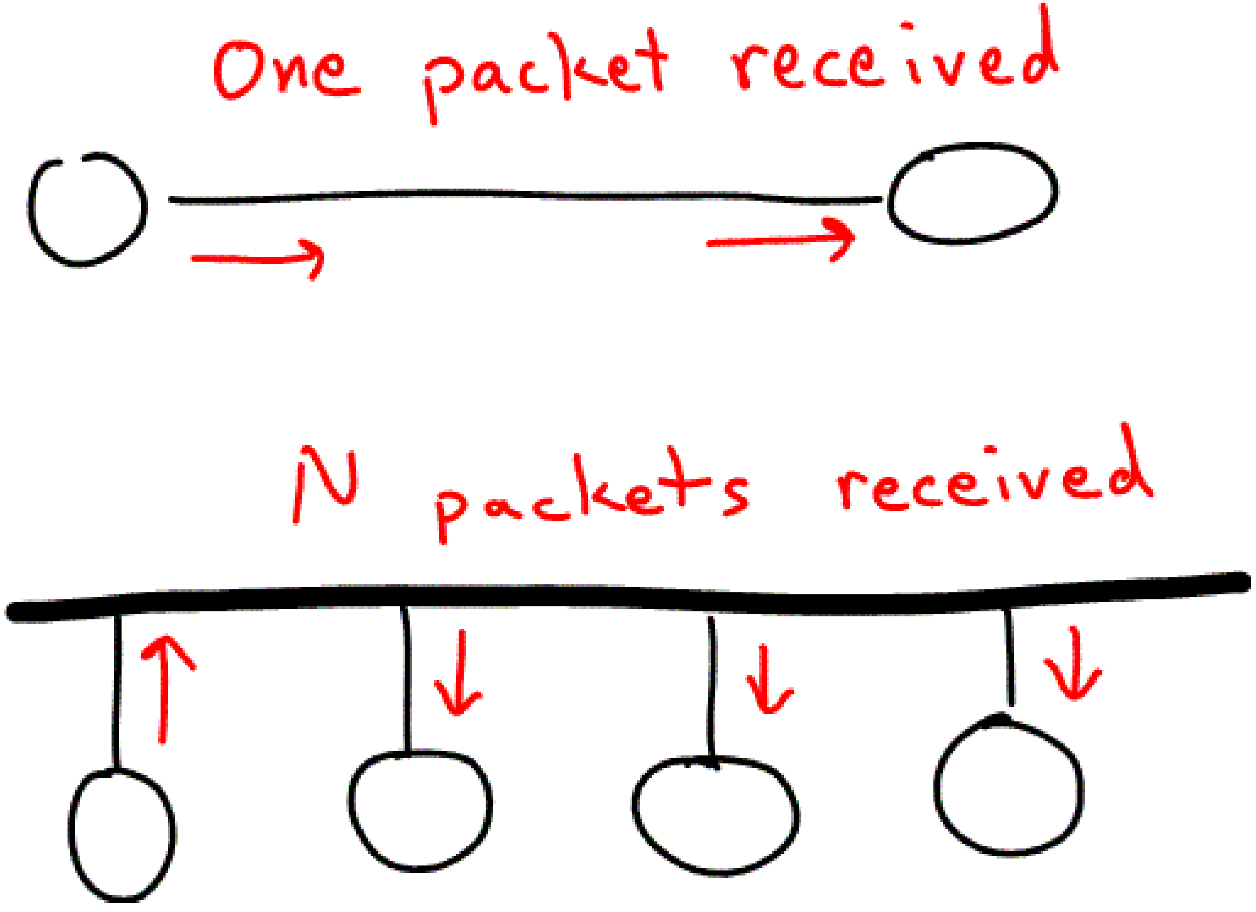
CS419





Point-to-point and broadcast links

CS419



Broadcast link (Ethernet)

CS419

- Well, N packets are “seen”, not really received
- The Ethernet hardware filters out packets that are not for “self”
  - By examining the Ethernet address
- The operating system (OS) never sees the packet (no packet interrupt)
- Though Ethernet does have multicast and broadcast address

Ethernet addresses *and* IP addresses???

CS419

- As you all know, the Internet is a network of networks
  - That's why its called the *Inter*net
- This introduces the concepts of:
  - Interface
  - Encapsulation

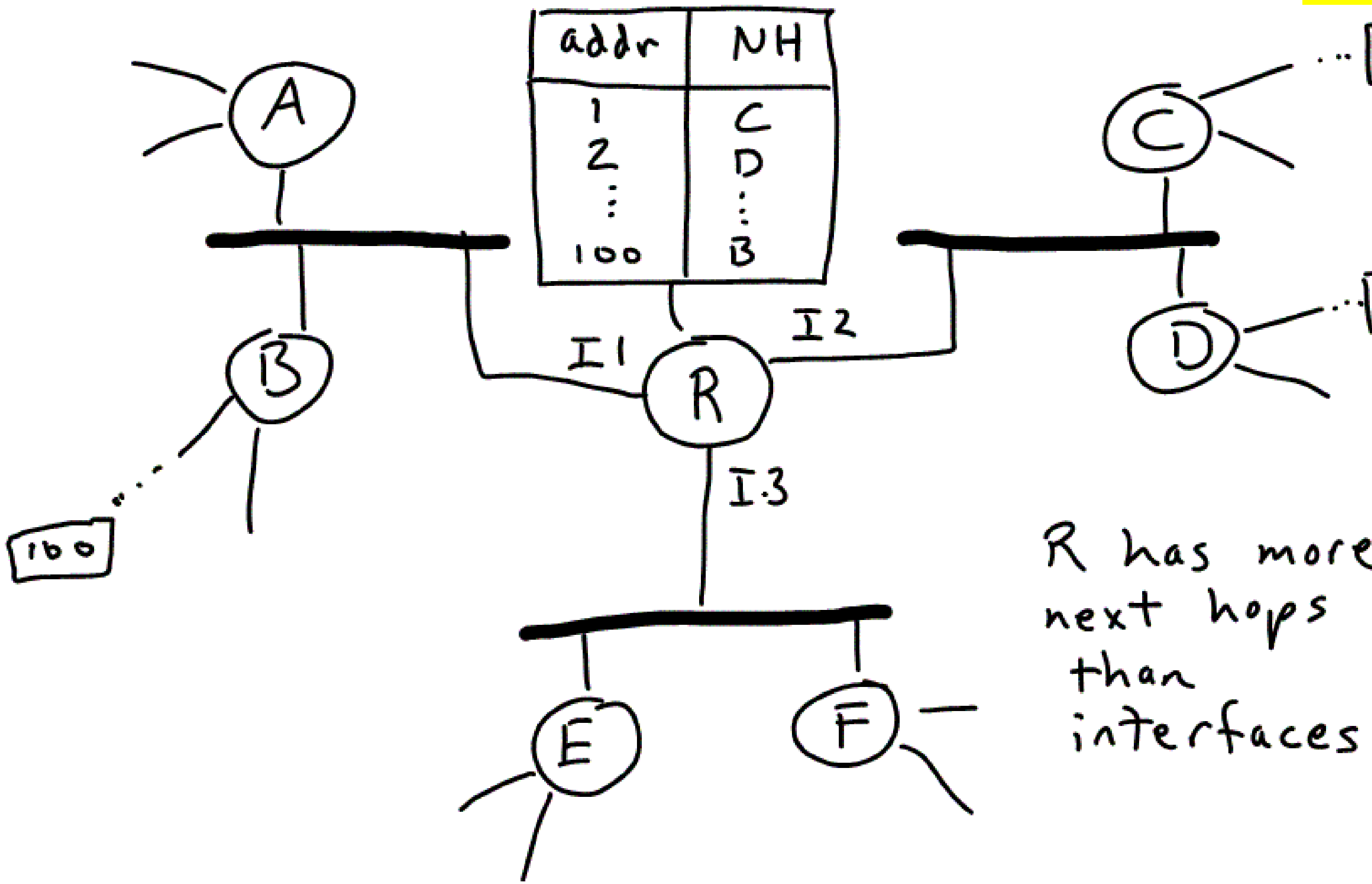
Next hop and interface (and logical interface!)

CS419

- Next hop is the next router on the path to the destination host
  - Or may be the destination host itself
- Interface is the input/output port over which the next hop can be reached
  - May be physical (an actual wire)
  - Or logical (multiple interfaces on an actual wire)

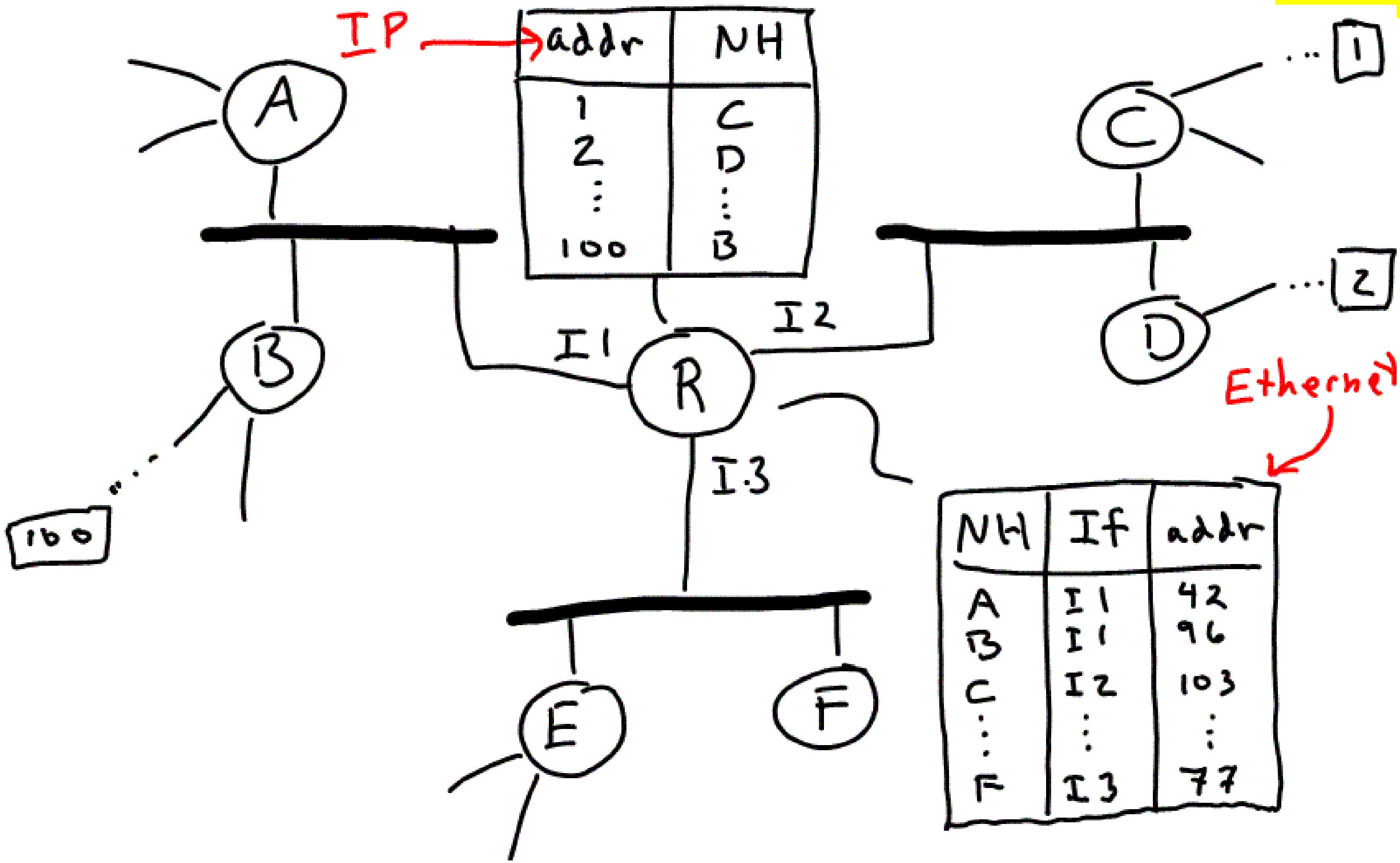
Next hop and interface (and logical interface!)

CS419



So the router has another table (neighbor table)

CS419



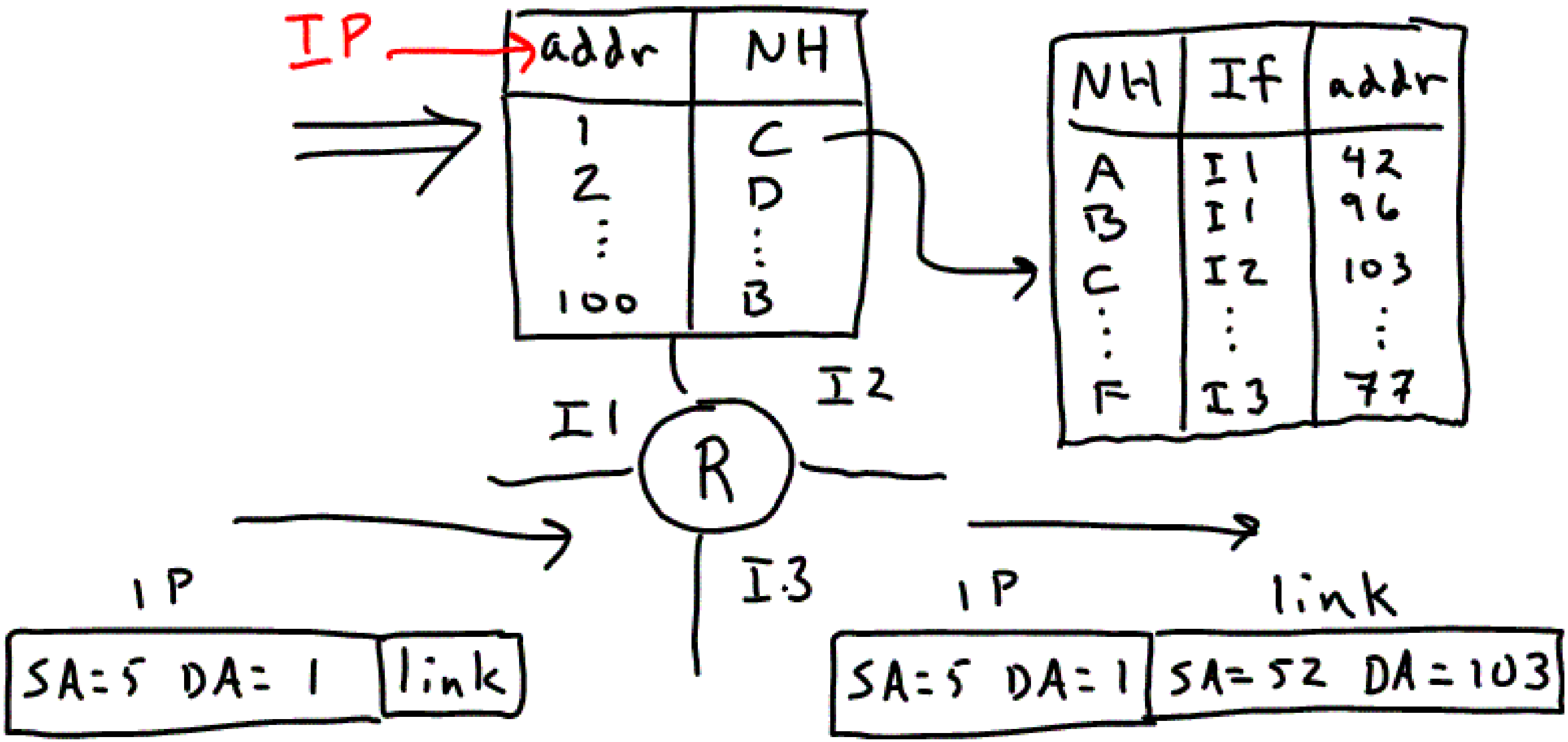
Router packet forwarding procedure:

CS419

- Look up dest IP address in received packet
  - Obtain Next Hop router (its IP address)
- Look up Next Hop router in the Neighbor Table
  - (with a pointer from the forwarding table entry)
  - Obtain iface (interface) and “link” address of Next Hop router
- *Encapsulate* IP packet in link packet and send over iface

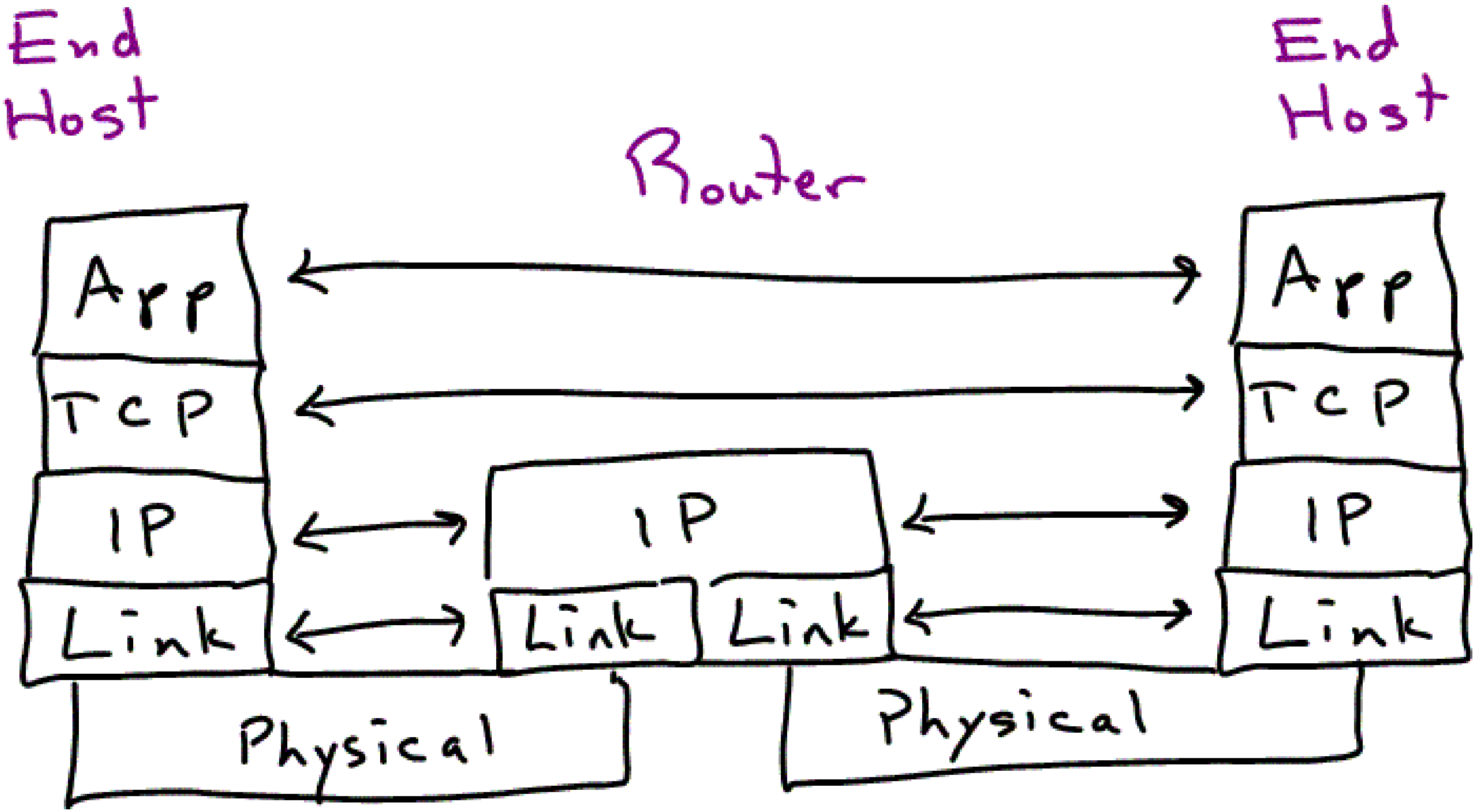
Router packet forwarding procedure:

CS419



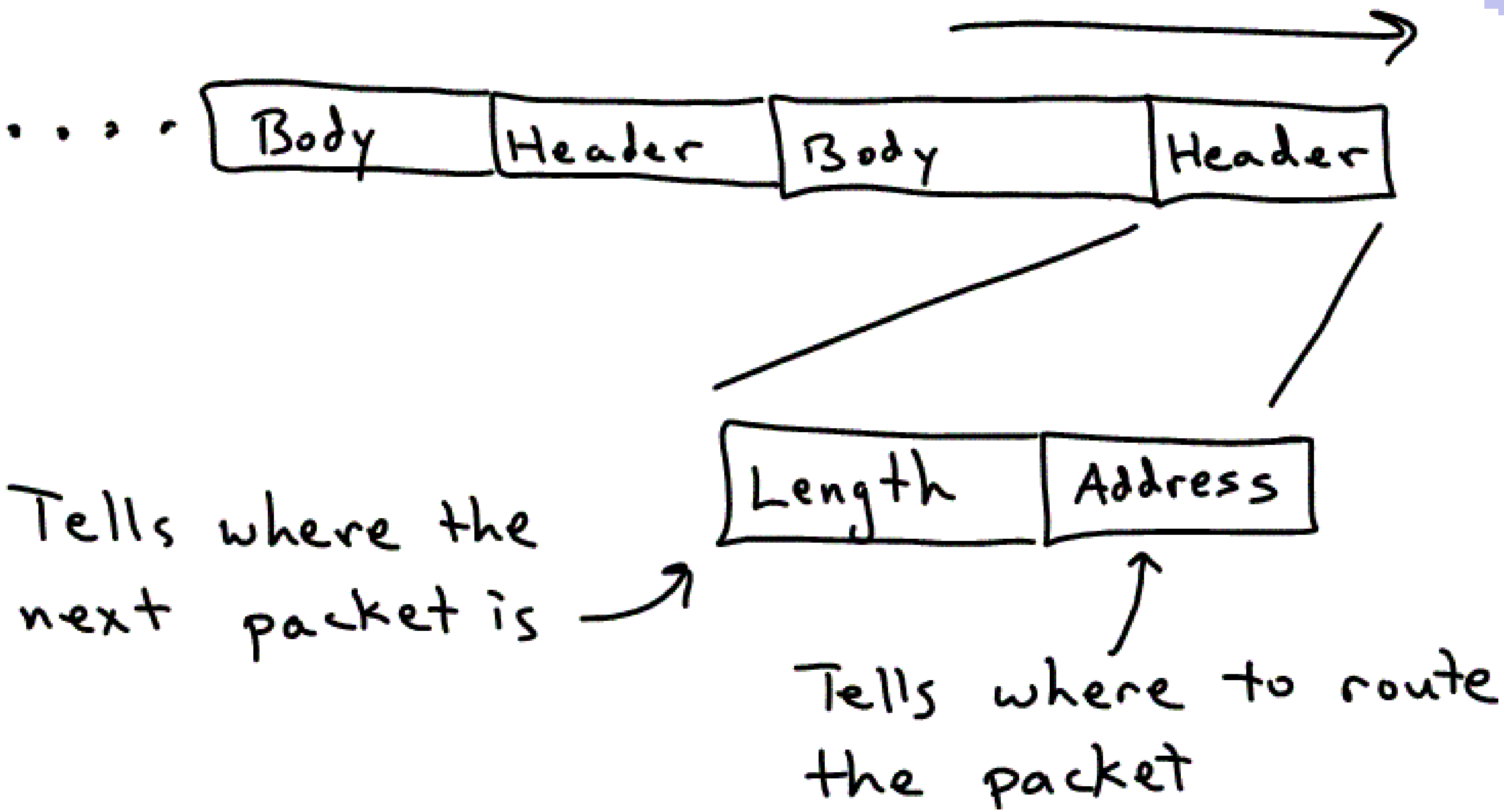
Protocol Layers

CS419



Packets (revisited)

CS419



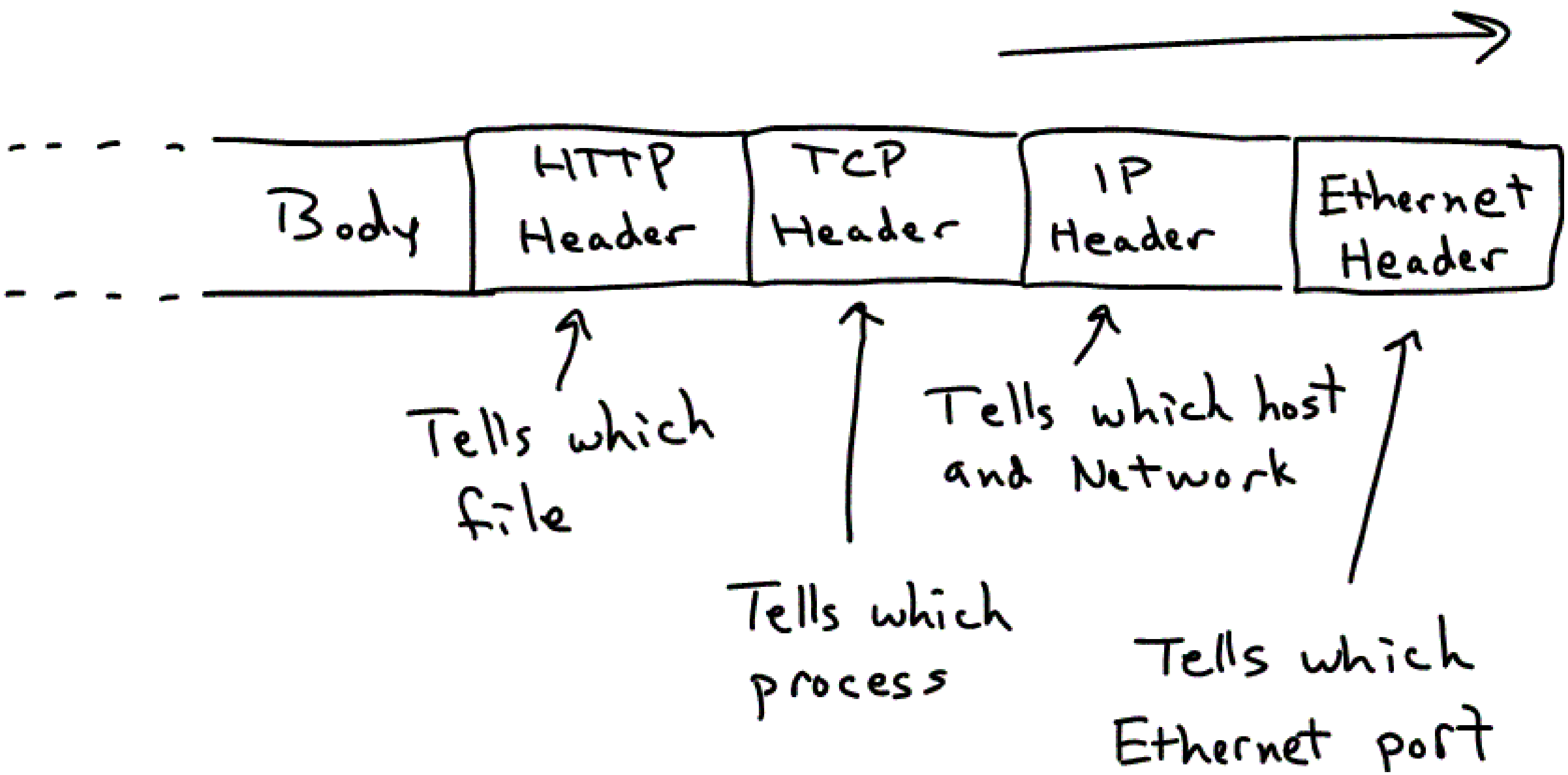
But, what is “where”?

- To an Ethernet, “where” is an Ethernet port
  - Ethernet address
- To the Internet, “where” is a host computer on a network
  - IP address
- To a host computer, “where” is a process
  - TCP or UDP port
- To a process, “where” may be a file
  - HTTP URL

A stack of headers

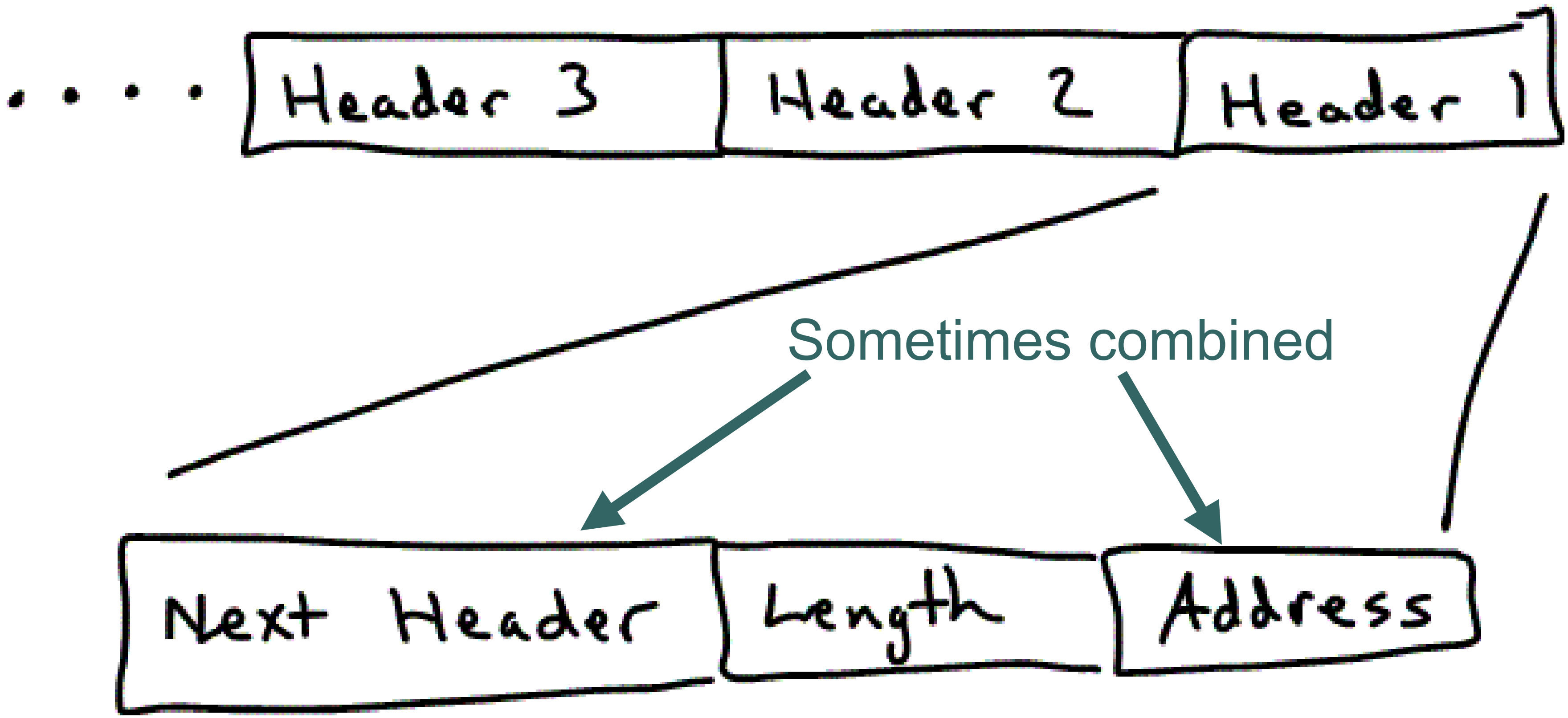
CS419

- To deal with all these “wheres”, a packet in fact contains a stack of headers:



A stacked header requires one more field: “next header”

CS419

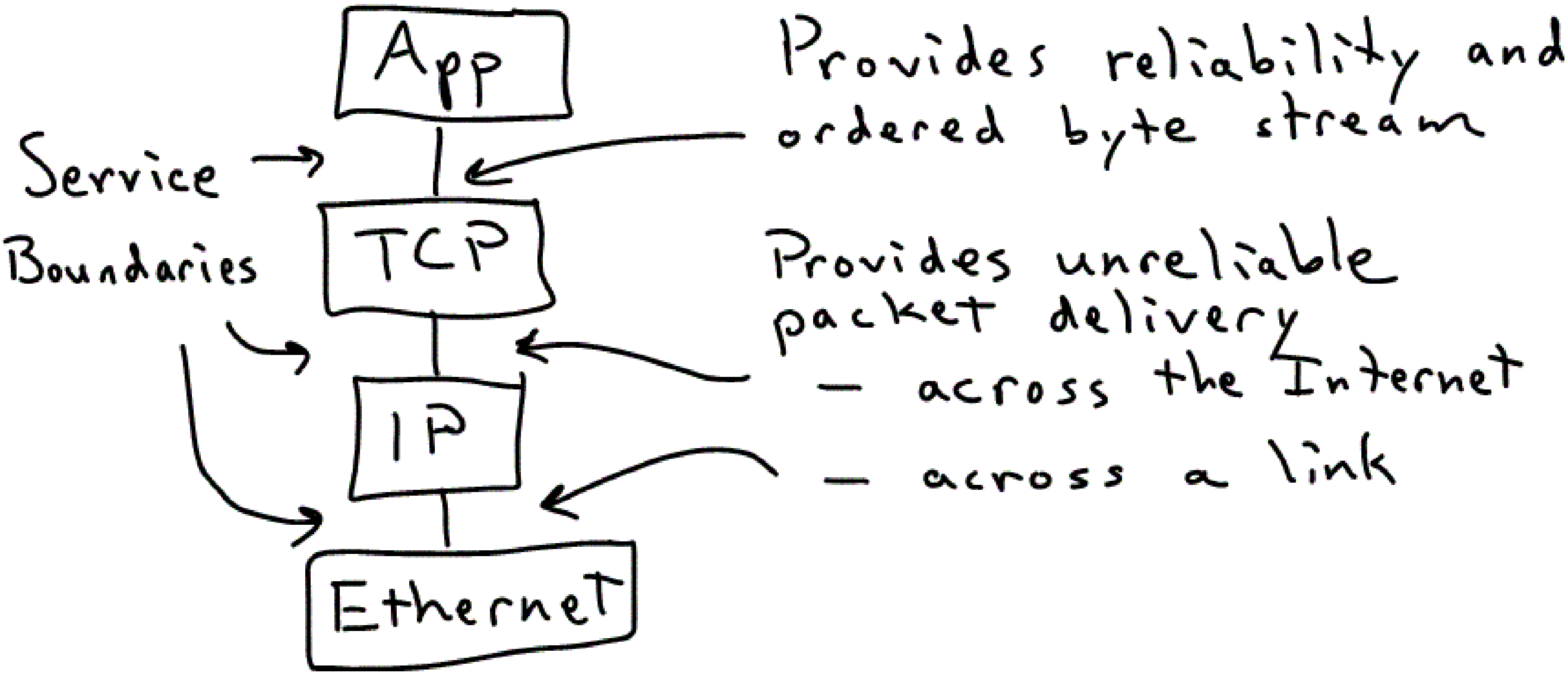


Header stack as protocol services

- Except for the physical layer protocol, protocol peers communicate with each other by talking to a lower layer
  - HTTP peers use TCP, TCP peers use IP, etc.
- We say that each protocol provides a *service* to the layer above it
  - Often there is a *service interface* that defines the service

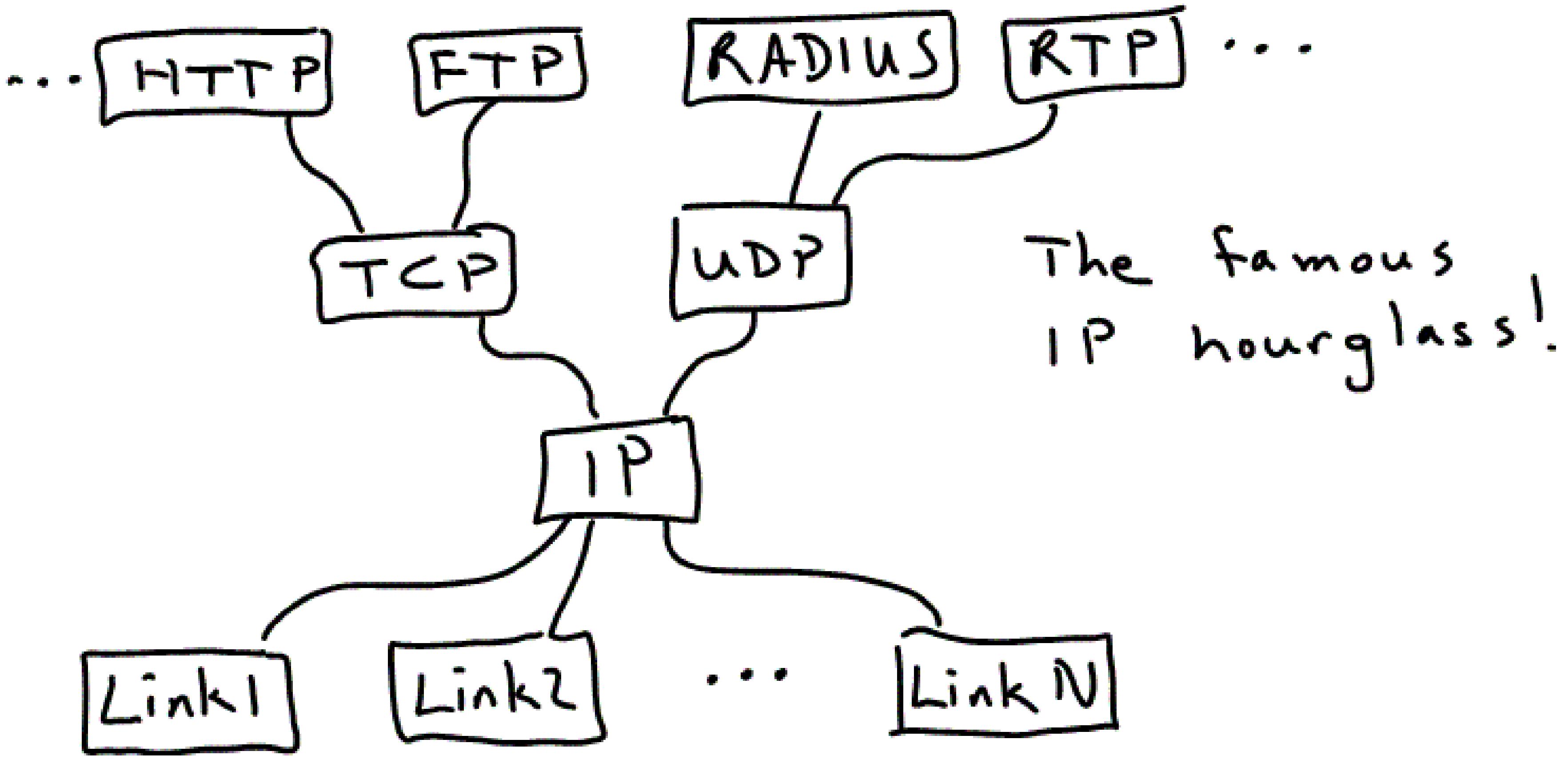
Protocol services

CS419



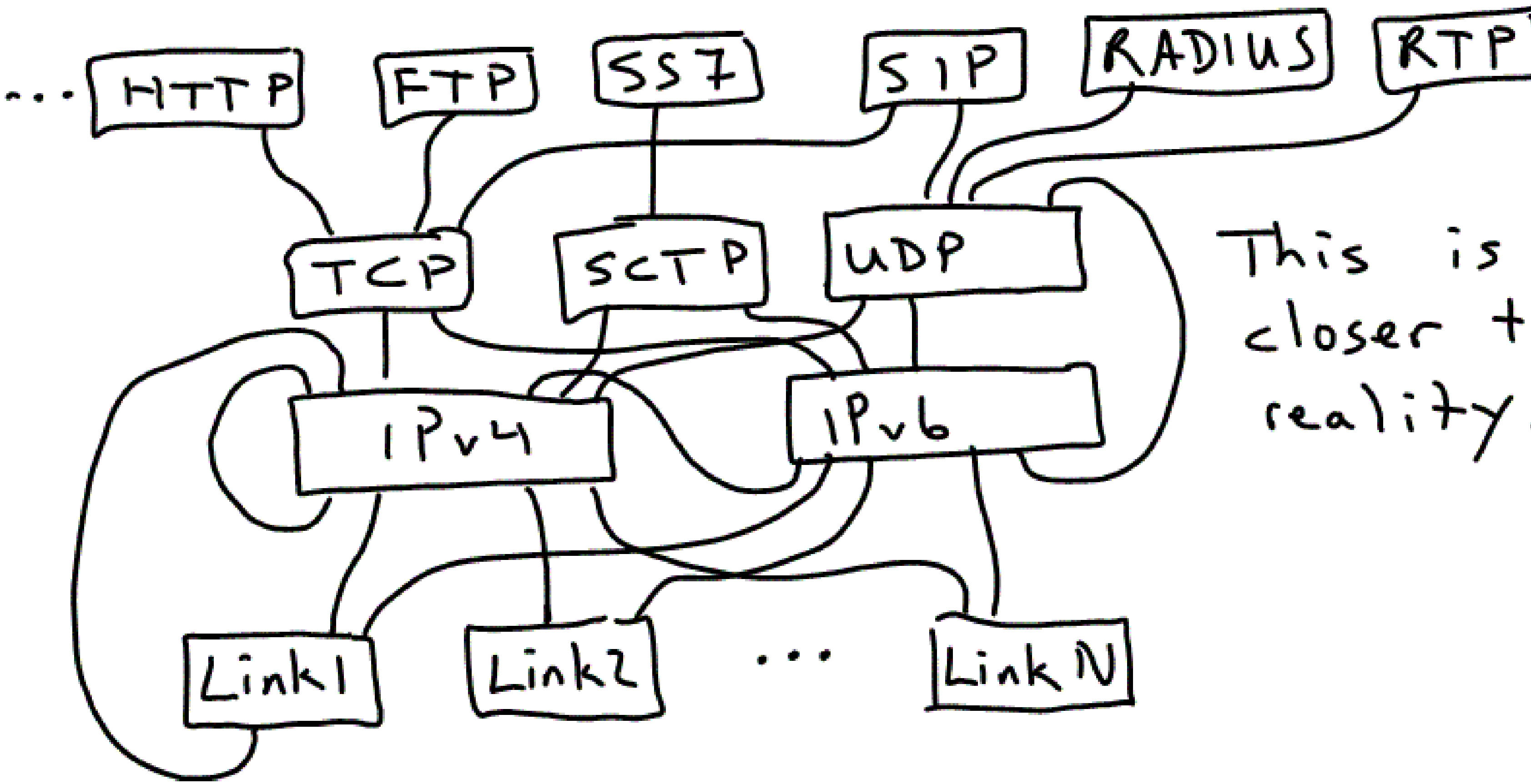
Services as a protocol graph

CS419



Services as a protocol graph

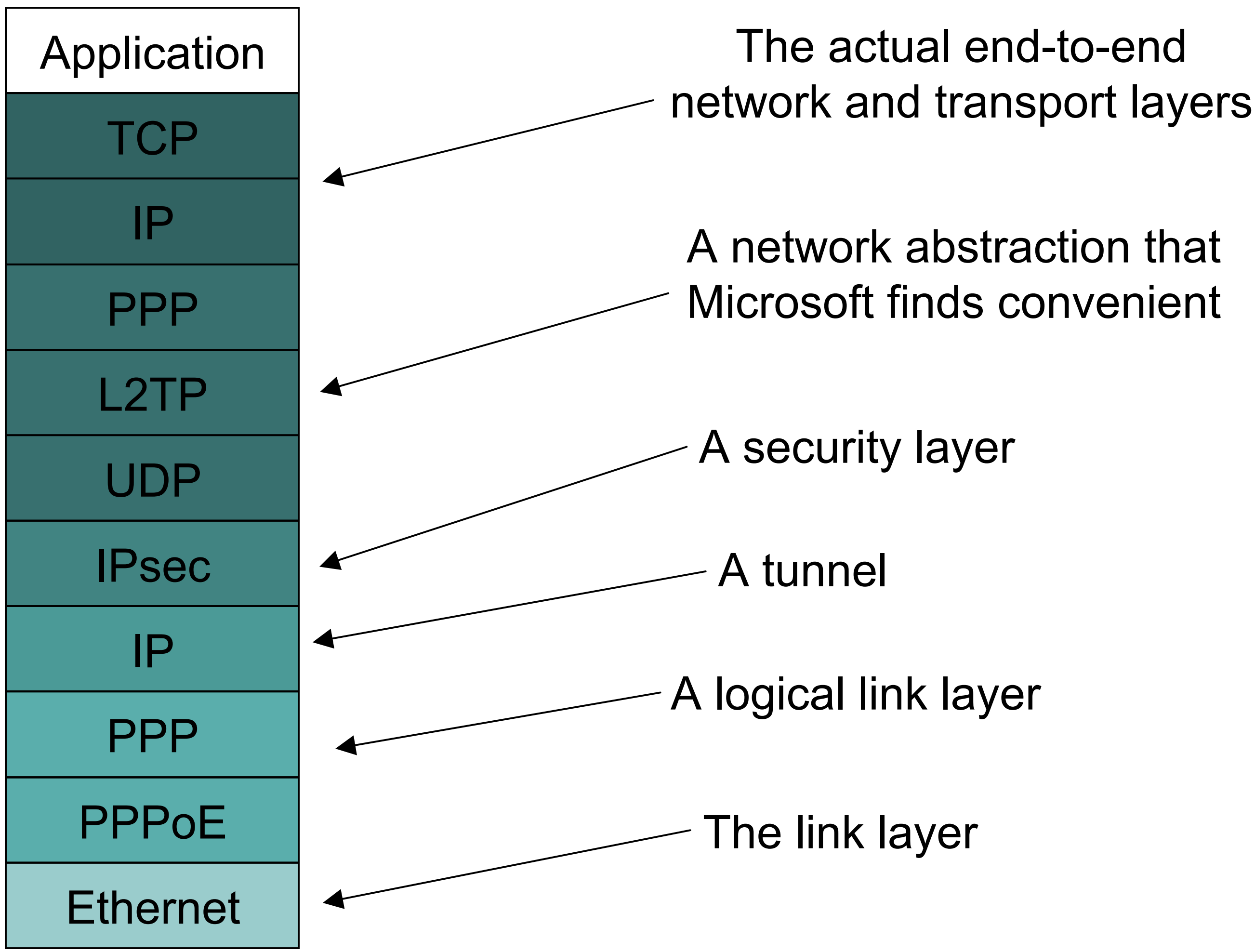
CS419



This is closer to reality!!!

Example Microsoft VPN stack

CS419



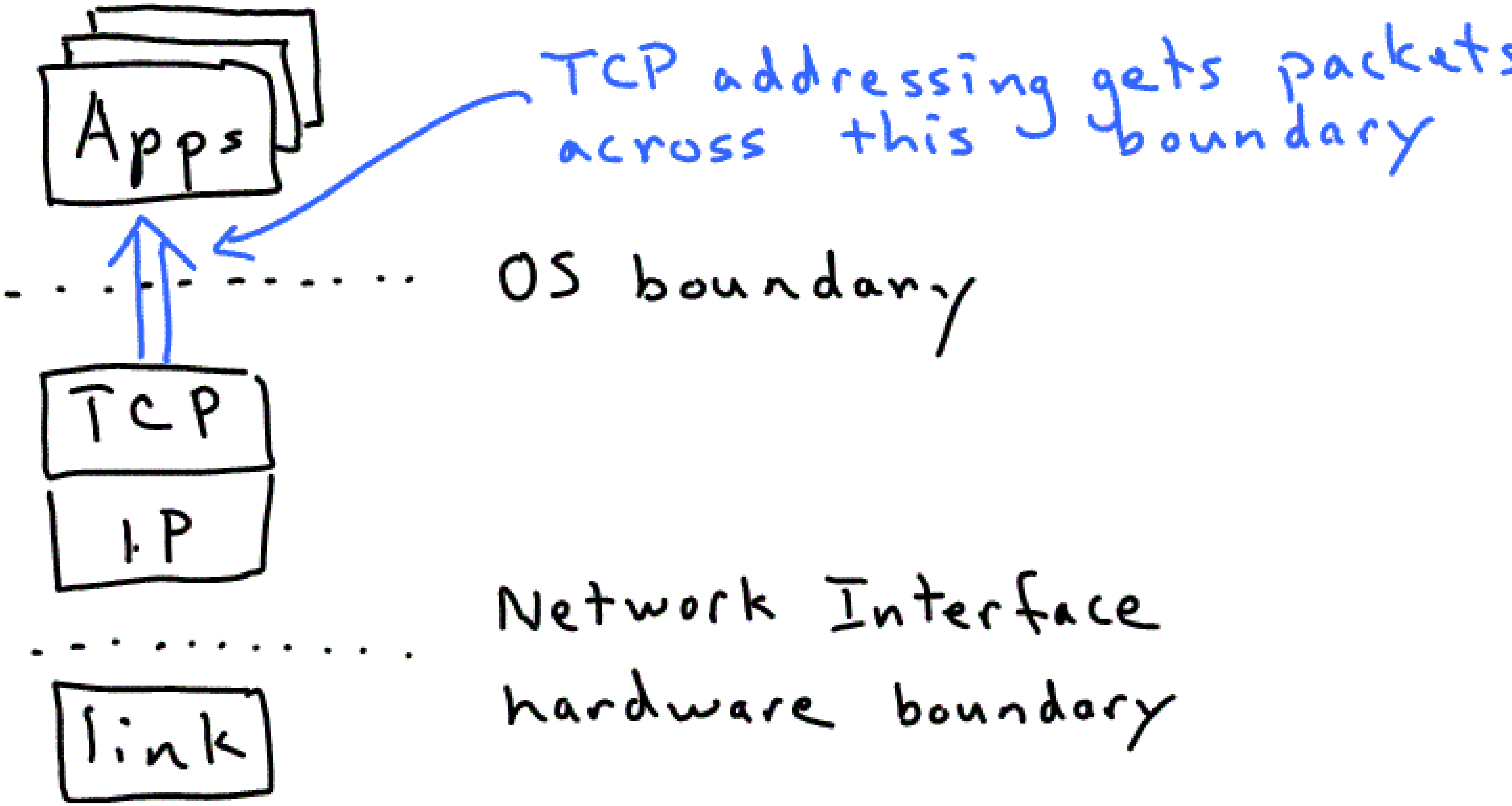
Example Microsoft VPN stack

Application
TCP
IP
PPP
L2TP
UDP
IPsec
IP
PPP
PPPoE
Ethernet

TCP:	Transport Control Protocol
IP:	Internet Protocol
PPP:	Point-to-Point Protocol
L2TP:	Layer 2 Tunneling Protocol
UDP:	User Datagram Protocol
IPsec:	Secure IP
PPPoE:	PPP over Ethernet

Protocol layers revisited

CS419



Summary of lecture 1

- *Packet networks* are more flexible than *circuit networks*
  - But have "QoS" issues of *delay (latency)*, *dropping*, and *jitter* (versus *blocking* for circuit networks)
- Fancy *queuing* can help, but ultimately traffic sources have to slow down to avoid *congestion*

Summary of lecture 1

- Delay has three components, queuing, propagation, and transmit
- Large *Delay x Bandwidth Product* pipes are becoming more common
- *Packets* have *headers* that tell *where* the packet is going, and how *long* it is (among other things)

Summary of lecture 1

CS419

- *Routers* have *forwarding tables* that select the *next hop* in a *path* to an *address*
  - And *neighbor tables* that tell which *interface* and link address to use to get to the next hop
- *Encapsulation* is used to get the IP packet from one router to another over a link

Summary of lecture 1

CS419

- Protocols are *layered*, with each layer providing a communications *service* to the layer above
- The layering is complex, with *tunnels* that allow protocols to be layered over themselves
- IP is a special layer at the waist of the Internet hourglass.

Next Lecture: IP

CS419

- Because of IP's special position in the Internet, it seems reasonable to start with IP, then work down and up…