L5: Protocols, Services, Layers

Sebastian Magierowski
York University

Outline

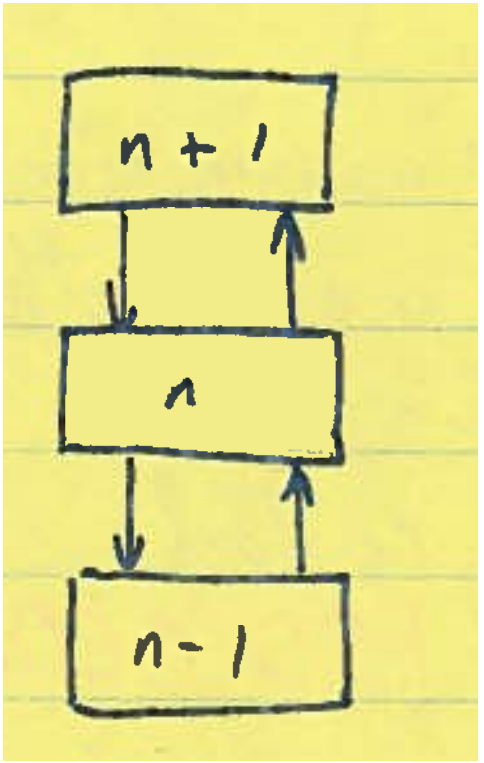
- Network Layering Terminology
  - protocols, services, peers, clients, etc.
- Network Protocol Examples
  - HTTP, TCP, DNS, UDP

Protocols

- For remote entities to establish working communications a set of rules needs to be in place
- Protocols are just these rules
  - HTTP
  - TCP
  - DNS
  - UDP
  - BGP
  - OSPF
  - etc., etc.,…
- In this lecture we look at some of these rules at work within the context of two other important ideas "layers" and "services"

5.1 Network Layers

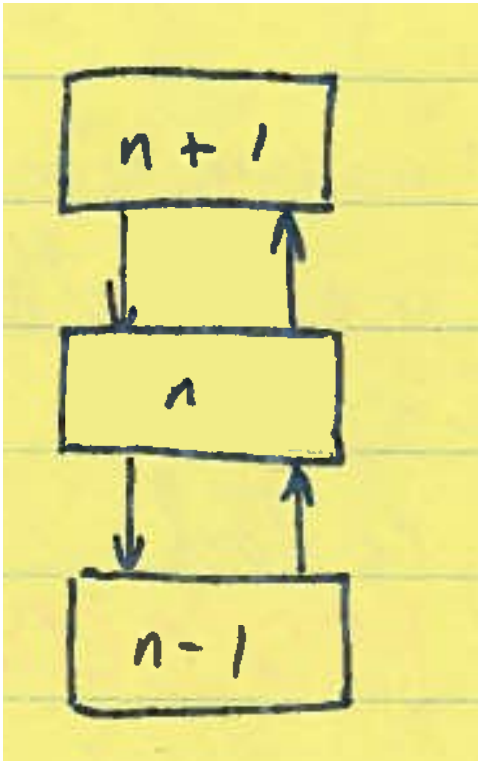
- Network functions are complex and require careful design
- To reduce design complexity network functions are organized as a stack of layers



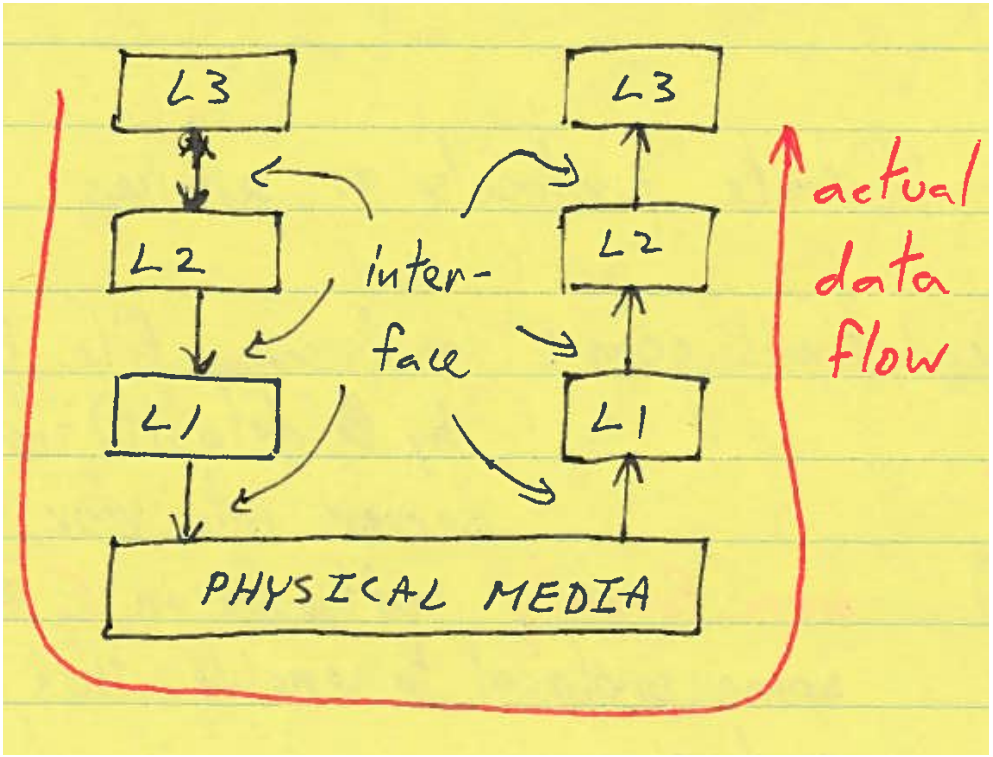
5.1 Network Layers

- Purpose of layer is to:

1. Provide "services" to layer above (e.g. routing, error control, connection-oriented link, etc.)
2. Shielding upper layers from implementation details (i.e. information hiding/abstract data types, etc.)

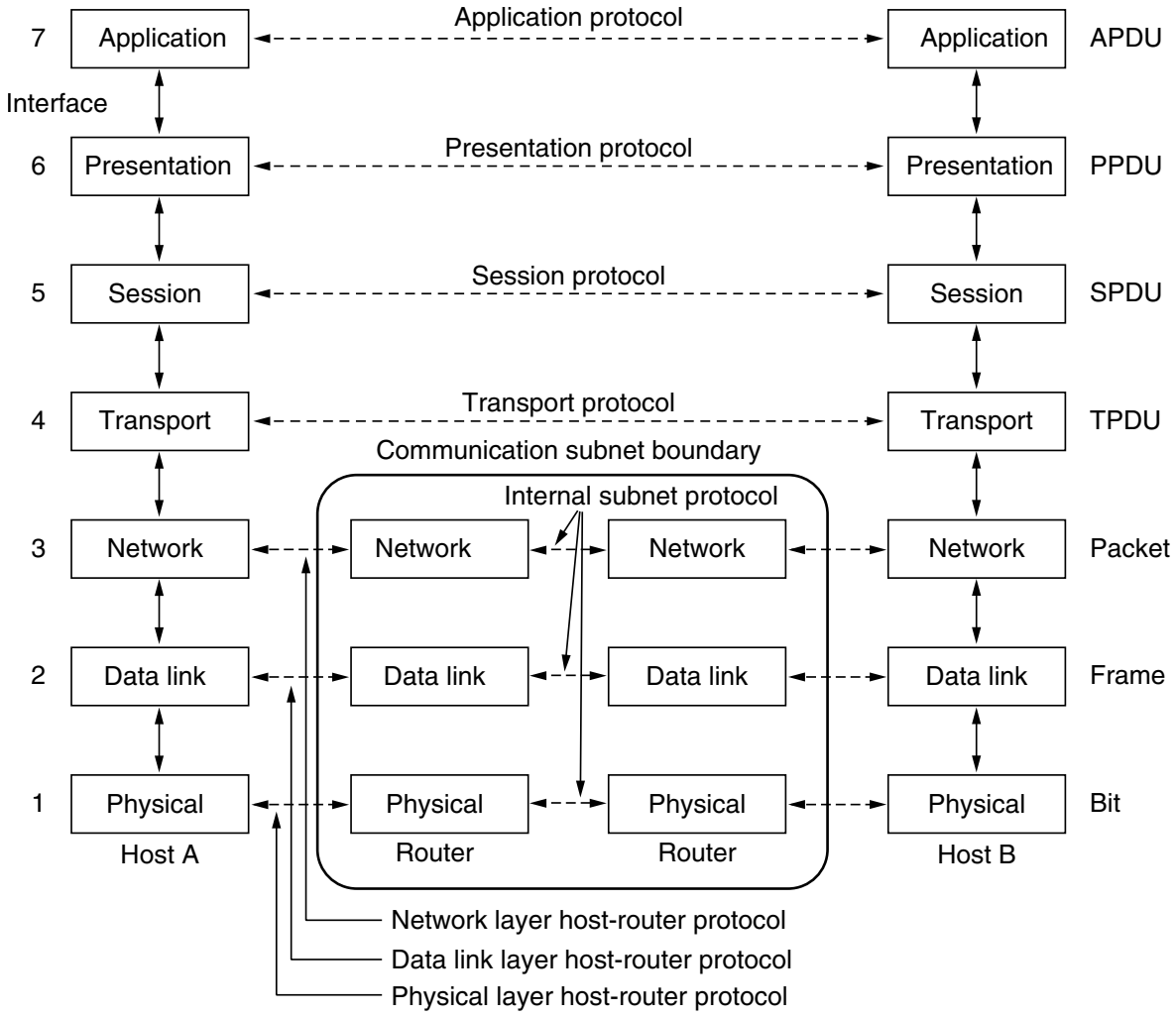


5.2 Layer Communication Flow



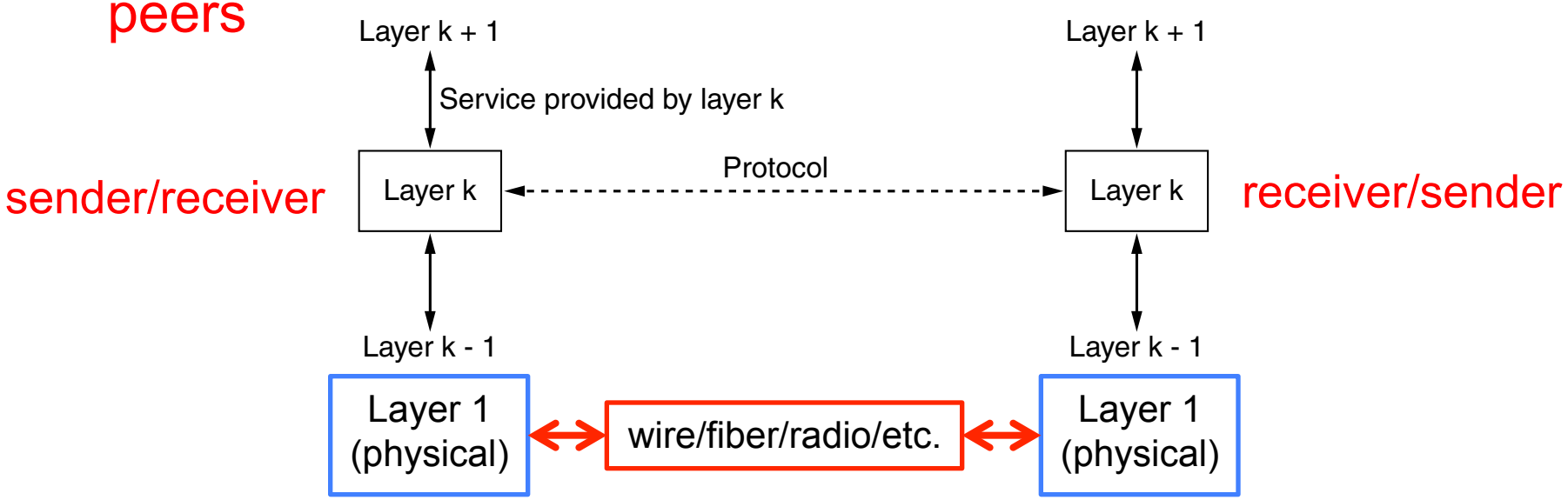
- "INTERFACE" lies between each pair of adjacent layers
- INTERFACE defines which services lower layers makes available to layer above

OSI Network Layers (A Preview)



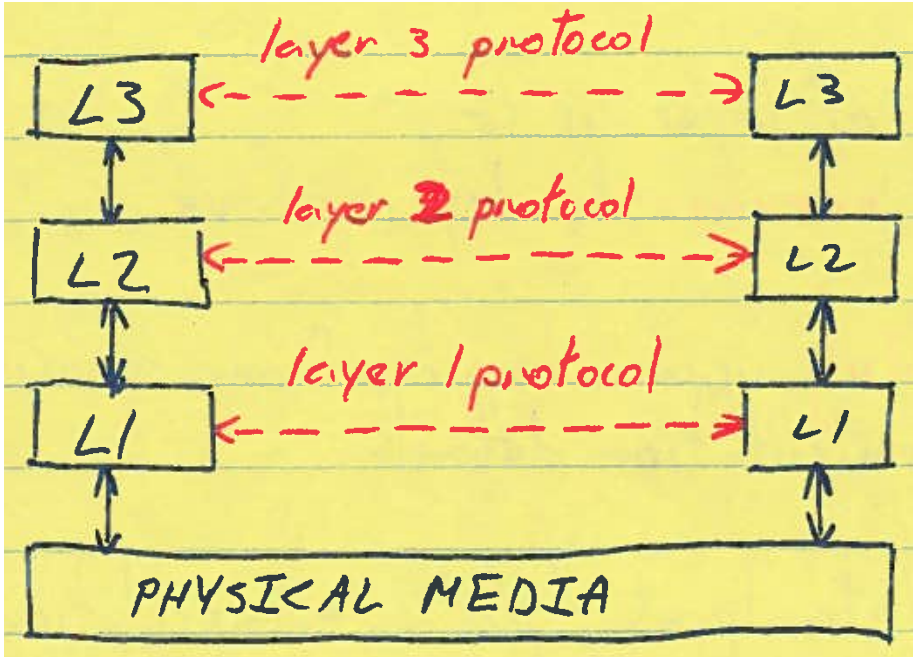
5.3 Peers and Protocols

- Corresponding layer on different machines are called peers



5.3 Peers and Protocols

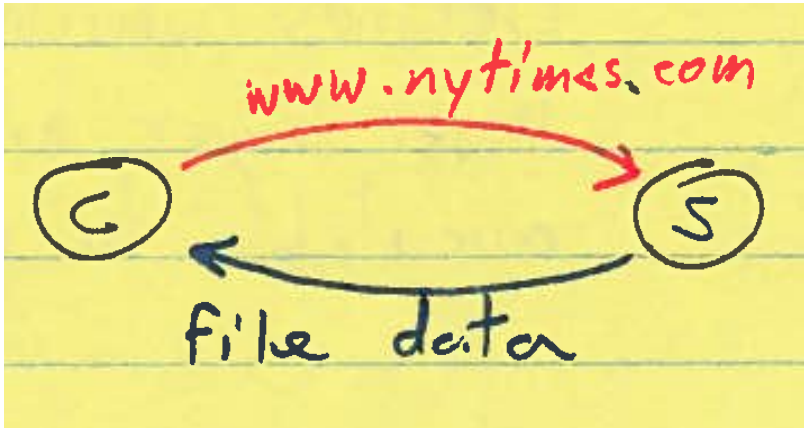
- In ABSTRACT we think of virtual communication that happens between peers



- Communication between peers adheres to specific rules called protocols (e.g. HTTP)
- Protocols are implementations of a service

5.4 Protocols In Action: Web Browsing

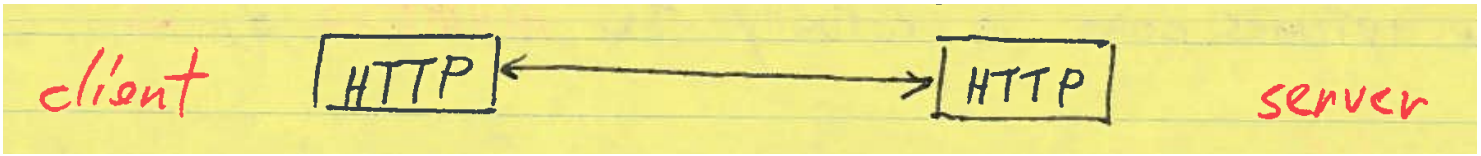
- I (client machine) want to grab a file (web page) from a server machine



- A whole bunch of rules (protocols) need to be followed to make this happen

5.4 Web Browsing

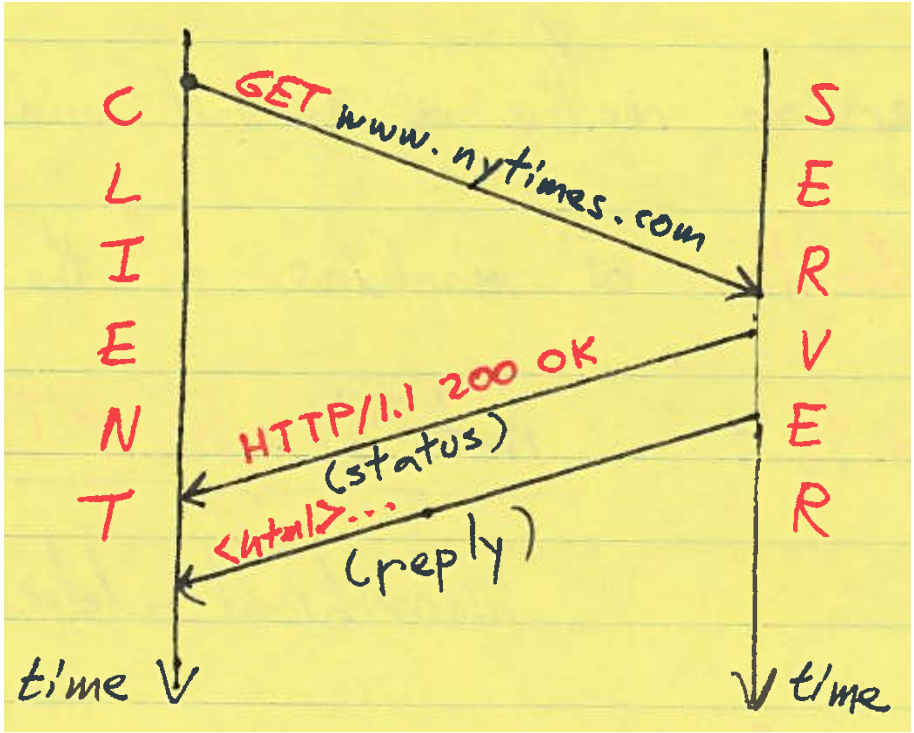
- At the top is HTTP



- In the abstract two HTTP entities are engaging in a virtual communication
  - The entities are using the HTTP protocol
  - Let's look at this in more detail…

5.5 HTTP

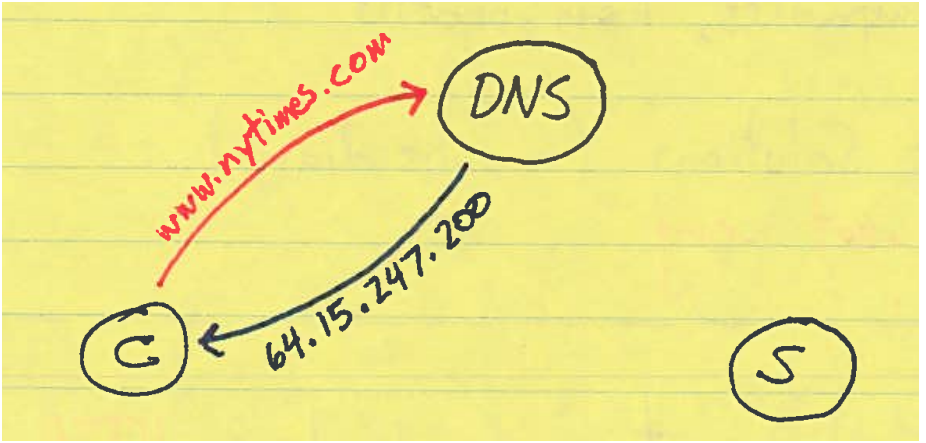
- To get a web page a specific set of rules is followed
  - Generally speaking HTTP is a “request-reply” protocol



- HTTP relies on the services of a bunch of other layers and their protocols…let’s see this in even more detail!

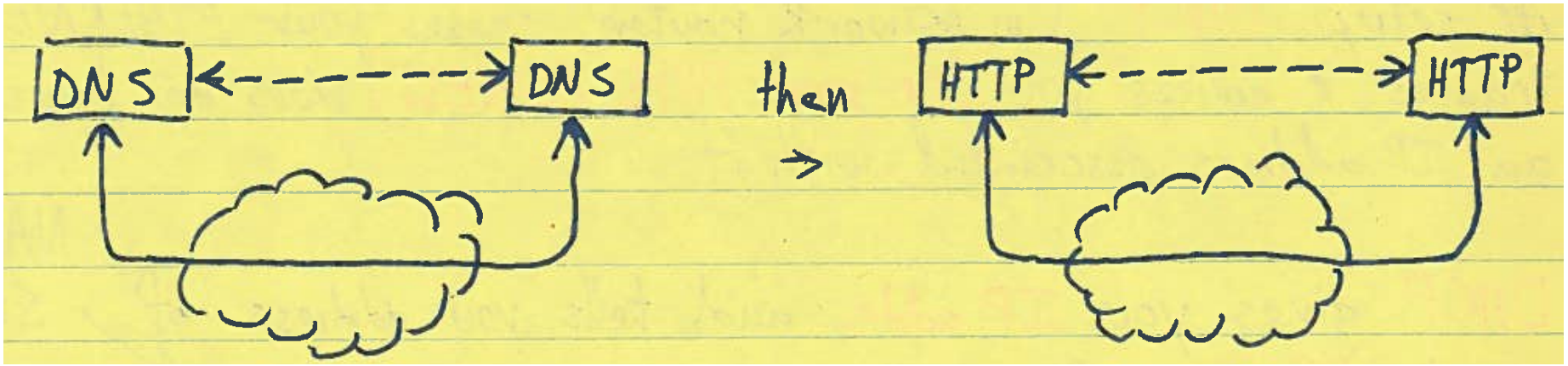
5.6 DNS (Domain Name System)

- How does C (the client) know where to send its messages (exactly)???????
  - www.nytimes.com is actually the server's address
  - But machines work with a 32-bit address (IP address)
    - not an alphanumeric one
  - So you need something capable of translating alphanumeric domain names to IP address
    - DNS does this
  - A few words on IP address
    - Often written in "dot-decimal" notation (for our consumption)
    - 2 level-hierarchy
      - netID.hostID



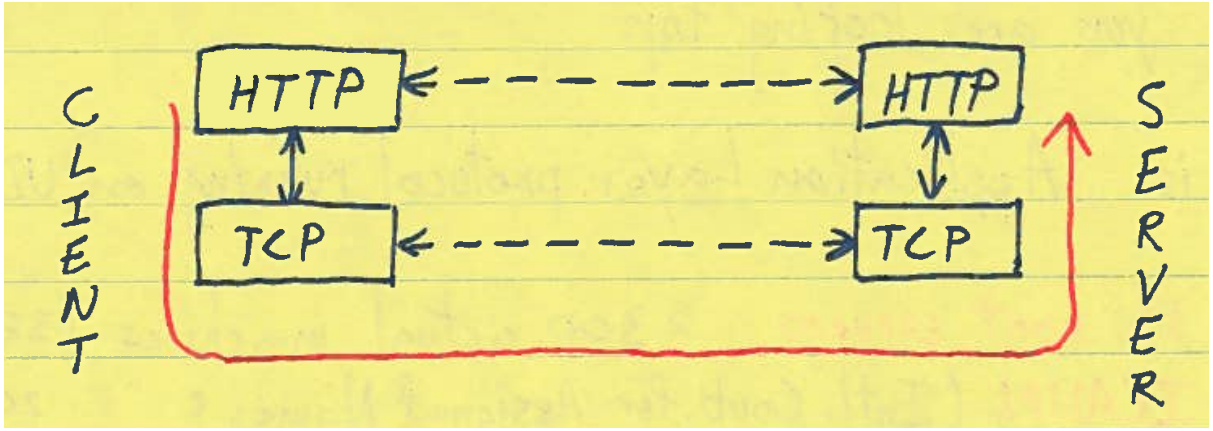
5.6 DNS

- In the end the DNS system runs the DNS protocol
  - It runs this on top of the UDP layer (which runs the UDP protocol)



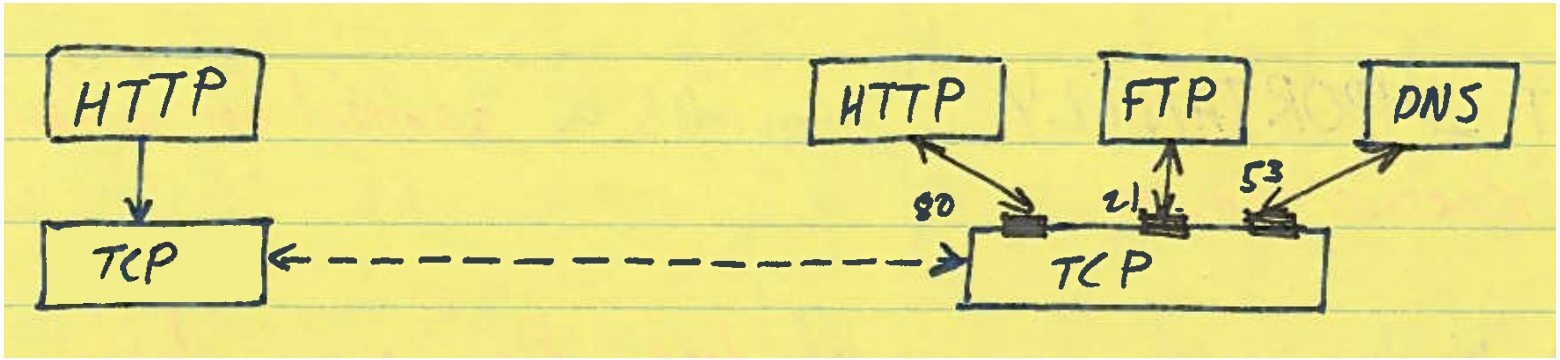
5.7 TCP

- HTTP employs the services of a lower layer and its protocol: TCP



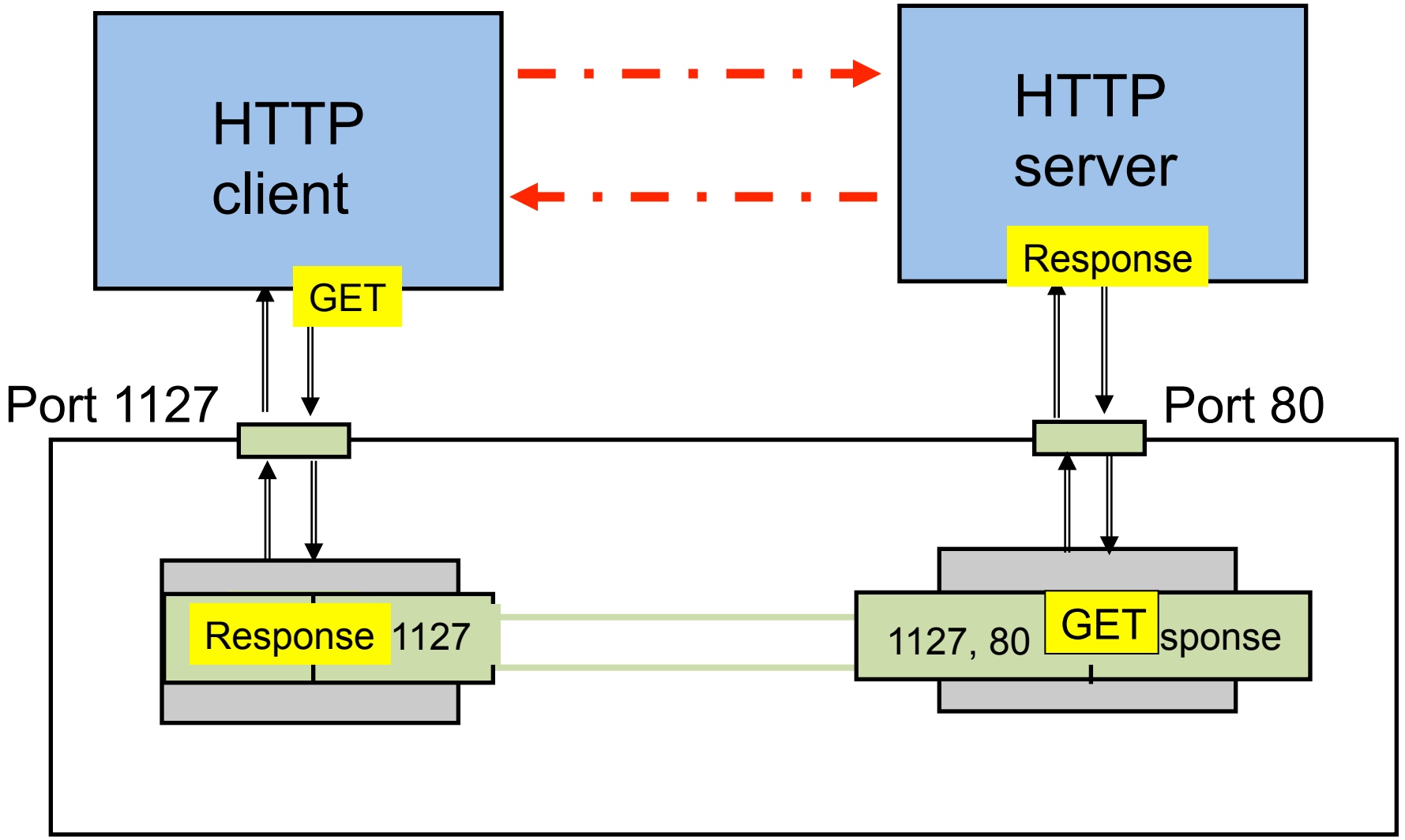
TCP Ports

- TCP can be serving a number of higher-level protocols simultaneously



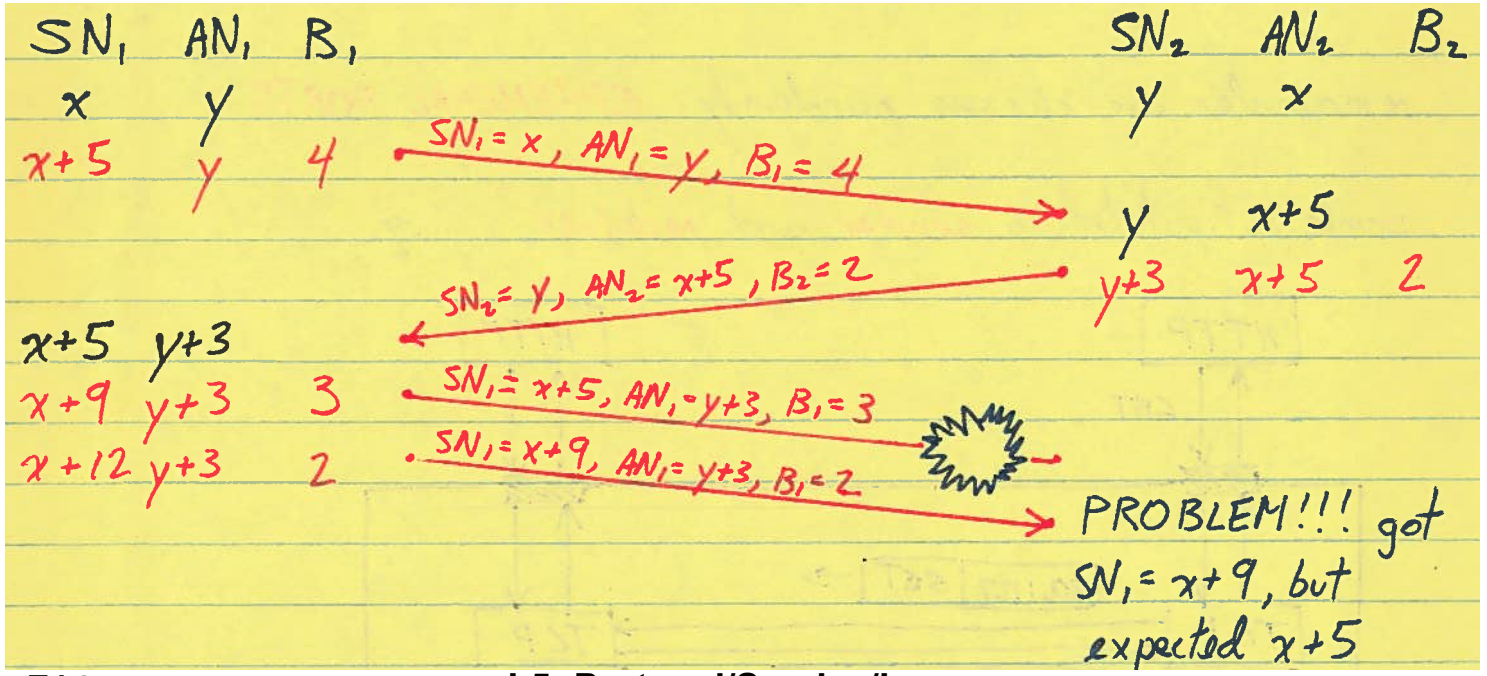
- Define TCP (Transport) address which can identify different higher-level protocols
  - Ports
  - 16-bit number (64k possible ports)
  - First 1k are reserved (remainder are ephemeral)

HTTP Uses Service of TCP



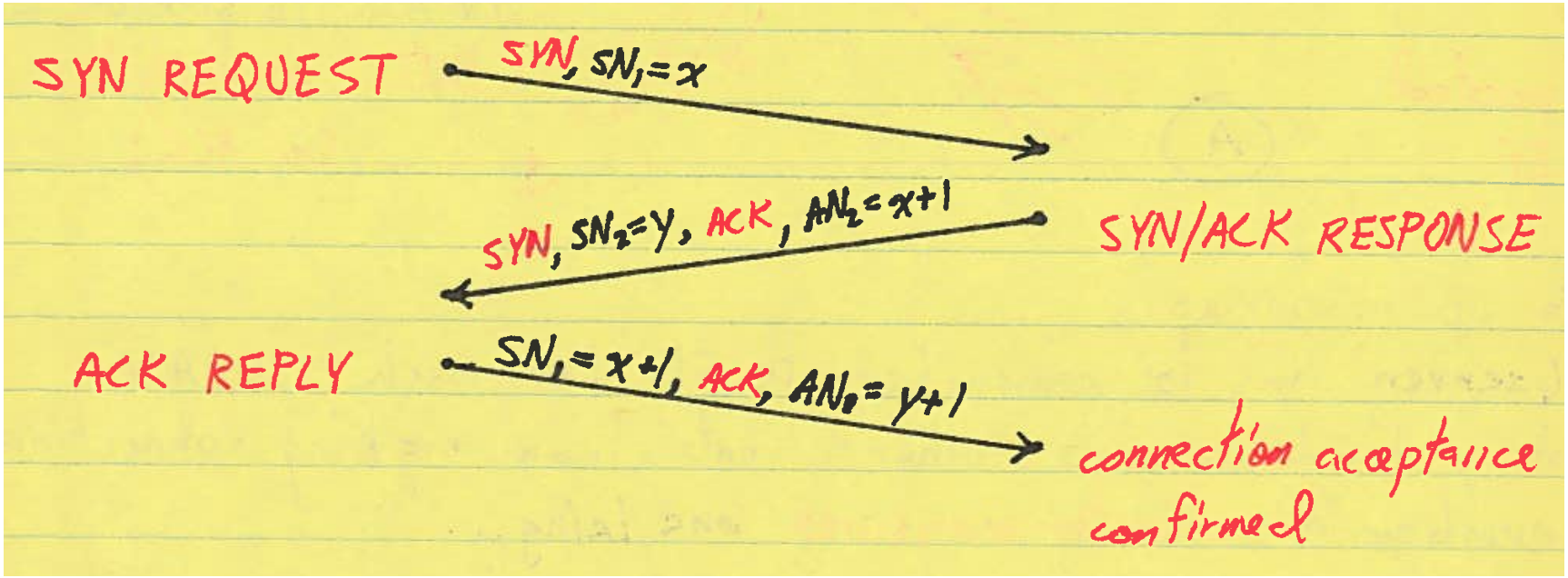
5.8 TCP Communications

- TCP provides connection-oriented byte stream service
  - Makes sure destination is available and reserves our access to it until disconnect (Layer 4 virtual circuit)
  - Delivers the bytes that a message is made up of in order
- How? Labels messages with sequence numbers



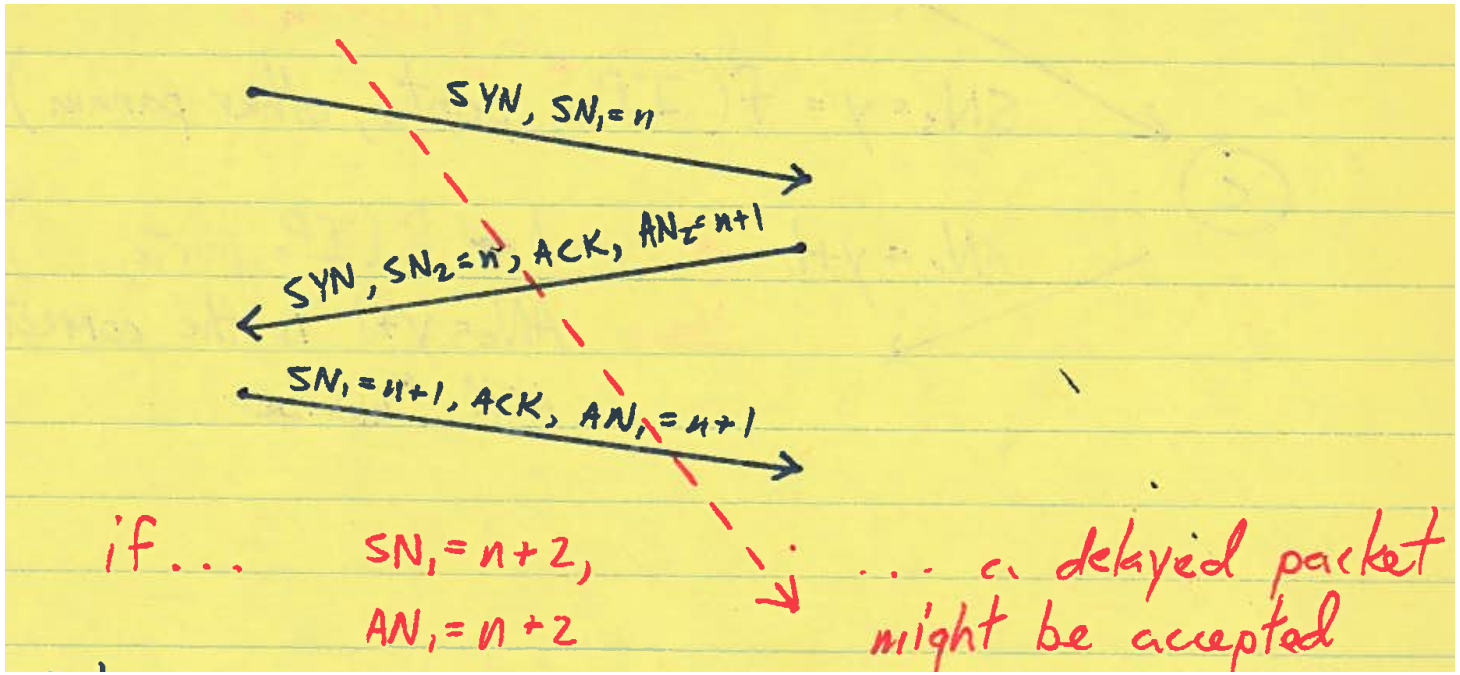
5.9 TCP Connection Establishment

- A 3-way handshake



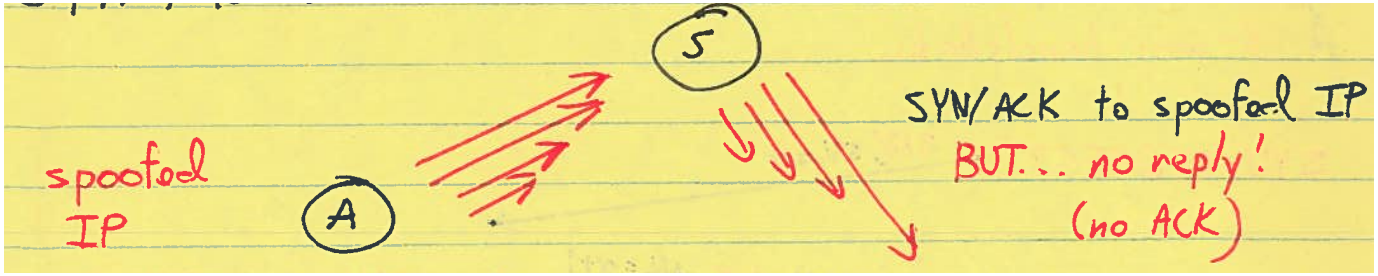
Predictable Sequence Numbers…

- …can cause problems



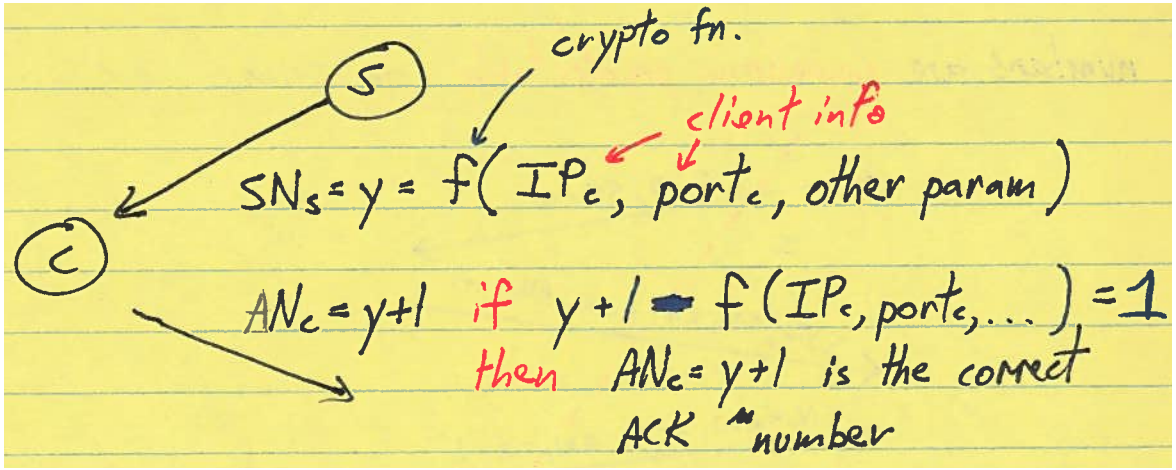
5.10 TCP Hacks

- SYN Flood



SYN Cookies

- Calculating SN's



Man-in-the-Middle Attack