

CMSC 417 Lecture #1 1/29/2016

Agenda

- ⇒ Syllabus (prereqs, books, late work, getting help)
- ⇒ Website (github)
- ⇒ Project 0 (github)
- ⇒ Piazza
- ⇒ Introduce Myself
- ⇒ Introduce TAs
- ⇒ Major Topics: Layering, Routing, Sockets, IP, BGP, R2P, TCP, DNS, Apps (HTTP, SMTP, ...), MAC, Wireless, SDN, MPLS?, Middleboxes?, NFV?
- ⇒ You can pick more topics

- ⇒ Intro / Layering
- ⇒ Sockets

CMSC 417 Spring 2016 Lecture #1 1/29/2016

Basic Networking / Layering

apps
process-to-process
host-to-host
n/w

in reality, layers 1-4
and 7 are all that's
talked about

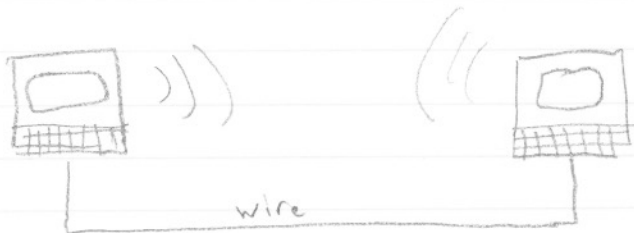
OSI model

- 7 apps (everything else)
- 6 presentation (data format)
- 5 session (logical connections)
- 4 transport (streams, big things, reliable)
- 3 network (packets w/ destinations)
dev-to-dev
- 2 link (frames, grouped bits w/ ends)
- 1 phy (physics, bits, voltages)

convergence layers, e.g., ARP, which we'll talk about later

In reality

| | |
|---|--------------------------|
| 7 | lots |
| 4 | TCP/UDP |
| 3 | IP |
| 2 | Ethernet 802.11 or 802.3 |
| 1 | lots |



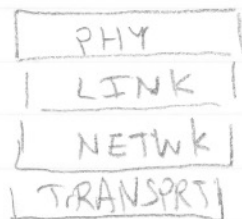
what if we had 3? 100?

what about multiple apps per computer?

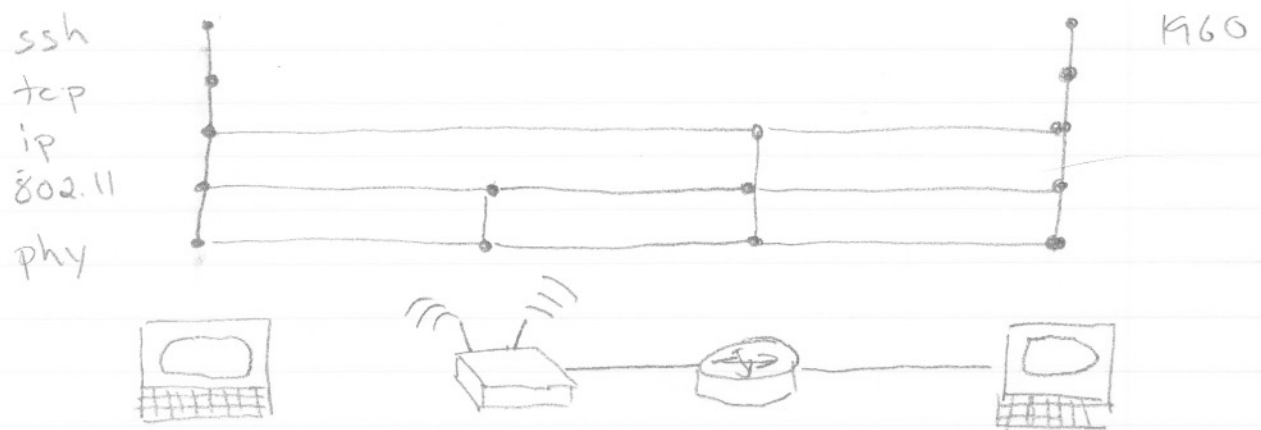
CMSC 417 Spring 2016 Lecture #1 1/29/2016

Layering continued

- ⇒ each layer typically has a header
- ⇒ layers are stacked



- ⇒ different headers are stripped/added as appropriate



- ⇒ separation of concerns
- ⇒ modularity
- ⇒ solve easier problems

Sockets

| | |
|------------------------------|------------|
| socket() / close() | sockets |
| connect() / send() / recv() | TCP client |
| bind() / listen() / accept() | TCP server |
| sendto() / recvfrom() | UDP |

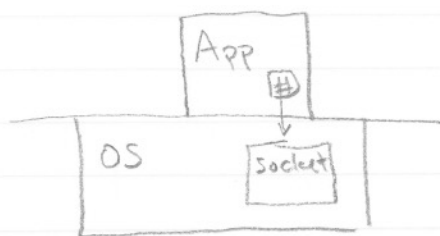
⇒ in/out parameters passed as pointers

Logically, a socket is something inside the OS.

⇒ socket() creates one

⇒ close() destroys one

Other calls tell the OS how it should make the object behave and allow you to retrieve information it's storing.



- ① allocate a socket and give me a pointer to it
- ② use the following local ^{and} or remote address
- ③ get my messages (if any)
- ④ send a message