Networking and PA5

Hannah Davis,
with slides from Ariana Mirian and Nadia Heninger
Original Idea:

- Network is dumb
- Simple, robust service
- Shift complexity to endpoints
- Acts like postal system (packet-based) rather than traditional phone system (circuit-based)
OSI Layers
(Open Systems Interconnection)

- Application
  - End user layer
  - HTTP, FTP, Skype, SSH, SMTP, DNS
- Presentation
  - Syntax, byte order, compression, encryption
  - SSL, SSH, MPEG, JPEG
- Session
  - Connection establishment and maintenance
  - APIs, sockets
- Transport
  - End-to-end connections between processes
  - TCP, UDP
- Network
  - Addressing, routing between nodes
  - IP
- Data Link
  - Link management, frames
  - Ethernet, WiFi
- Physical
  - Physical wires
  - Photons, RF modulation
Internet Architecture

Application layer

Transport layer

Network layer

Link layer

Physical layer

NTP  DNS  SMTP  HTTP  FTP

UDP  TCP

IP

Cellular  WiFi  Ethernet

Radio  Copper  Fiber
Using the internet: A worked example

You connect your laptop to a cafe wifi network and type ucsd.edu into your browser’s URL bar. What happens?
Using the internet: A worked example

You connect your laptop to a cafe wifi network and type ucsd.edu into your browser’s URL bar. What happens?

You are waiting masked and outdoors for a takeout order from a cafe, and while you’re waiting you perch against the wall at a socially distant distance from everyone else, pull your laptop out, connect to the cafe’s wifi network and type ucsd.edu into the browser’s URL bar.
Using the Internet: a worked example

1. Your laptop uses DHCP (Dynamic Host Configuration Protocol) to bootstrap itself on the local network via WiFi over radio waves.
Using the Internet: a worked example

1. Your laptop uses **DHCP (Dynamic Host Configuration Protocol)** to bootstrap itself on the local network via **WiFi** over **radio waves**

   - New host has no **IP address**, doesn’t know who to ask
Using the Internet: a worked example

1. Your laptop uses **DHCP (Dynamic Host Configuration Protocol)** to bootstrap itself on the local network via **WiFi** over **radio waves**
   - New host has no **IP address**, doesn’t know who to ask
   - Broadcasts **DHCPDISCOVER** to 255.255.255.255 with its MAC address
Using the Internet: a worked example

1. Your laptop uses **DHCP (Dynamic Host Configuration Protocol)** to bootstrap itself on the local network via **WiFi** over **radio waves**
   - New host has no **IP** address, doesn’t know who to ask
   - Broadcasts **DHCPDISCOVER** to **255.255.255.255** with its **MAC address**
   - DHCP server responds with config:lease on host IP address, gateway IP address, DNS server information
Using the Internet: a worked example

2. Your laptop makes an **ARP** request to learn the **MAC address** of the local router.
Using the Internet: a worked example

2. Your laptop makes an **ARP** request to learn the **MAC address** of the local router.

- Every connection outside the local network will be encapsulated in a **link-layer** frame with the local router’s MAC address as the destination.
- Your laptop encapsulates each **IP** packet in a **WiFi** frame addressed to the local router.
2. Your laptop makes an **ARP** request to learn the **MAC address** of the local router.

- Every connection outside the local network will be encapsulated in a **link-layer** frame with the local router’s MAC address as the destination.
- Your laptop encapsulates each **IP** packet in a **WiFi** frame addressed to the local router.
- The router removes the **WiFi frame** and adds an **Ethernet frame** to forward them on its fiber connection to its upstream ISP, or to another part of the network.
- Each hop re-encodes the link layer for its own network.
Using the Internet: a worked example

3. Your laptop does a DNS lookup on ucsd.edu
   - It learned the IP address of a DNS server from the router or had one already hardcoded in (8.8.8.8)
   - The DNS request is tunneled through UDP packets which are themselves inside IP packets
Using the Internet: a worked example

3. Your laptop does a DNS lookup on ucsd.edu

- It learned the IP address of a DNS server from the router or had one already hardcoded in (8.8.8.8)
- The DNS request is tunneled through UDP packets which are themselves inside IP packets
- The DNS server responds with either “ucsd.edu has IP address x.x.x.x” or “I don’t know, but the nameserver at y.y.y.y might”
  - Follows a hierarchy upward: your ISP, then the .edu nameserver, then UCSD’s nameserver.
  - Eventually, get final IP address 132.239.180.101
Using the Internet: a worked example

4. Your laptop opens a TCP connection to IP address 132.239.180.101.

- Unlike UDP, has reliability.
Using the Internet: a worked example

4. Your laptop opens a TCP connection to IP address 132.239.180.101.

- Unlike UDP, has reliability.
- TCP is wrapped in IP which is wrapped in Ethernet
Using the Internet: a worked example

4. Your laptop opens a TCP connection to IP address 132.239.180.101.

- Unlike UDP, has reliability.
- TCP is wrapped in IP which is wrapped in Ethernet
- Each stop in the network checks its routing table against the destination IP address.
- E.g. sbcglobal.net-> att.net-> leve3.net -> cenic.net-> ucsd.edu
5. Your laptop sends an **HTTP GET** request across the **TCP** connection.
Using the Internet: a worked example

5. Your laptop sends an **HTTP GET** request across the **TCP** connection.
Using the Internet: a worked example

5. Your laptop sends an HTTP GET request across the TCP connection.

- If you’re using HTTPS, it performs a TLS handshake and encrypts the request before splitting it into TCP packets.
Using the Internet: a worked example

5. Your laptop sends an **HTTP GET** request across the **TCP** connection.

- If you’re using **HTTPS**, it performs a **TLS handshake** and **encrypts** the request before splitting it into **TCP** packets.
- Any future connections restart from step 3 with a new **DNS** request.
PA 5 introduction

- Scavenger hunt
- Begins tomorrow
- Starting point: your email
- Unzip the .tar file, and try to find hidden information
- The file will tell you what to submit
- Start early! It can be hard to tell how close you are to finishing.
- Be considerate of spoilers: try to use mainly OH and private Piazza posts for questions.
- Have fun!
Overview of tools you may need

- nc
- nmap
- ssh
- tcpdump
- wget
Overview of tools you may need

- Nc – allows you to make connections locally
- Nmap – scan ports/IPs (locally and externally)
- Ssh – connect to servers
- Tcpdump – view traffic on machine
- Wget – download of files from internet

- All of these have “man” pages!
NetCat (shoutout to Julia Evans)

**nc**
- Like `cat` for your network!
- It lets you create TCP (or UDP) connections from the command line & send/receive data.

**netcat**
- `nc -l PORT`
  - Start a server! This listens on PORT & prints everything received.

**Julia Evans**
- Wizardzines.com
- `nc IP PORT`
  - Be a client! Opens a TCP/UDP connection to IP: PORT.

**send files**
- Want to send a 100 GB file to someone on the same wifi network? Easy!
- Receiver:
  - `nc -l 8080` > file
- Sender:
  - `192.168.x.x`
  - `cat file | nc YOUR_IP 8080`

**make HTTP requests by hand**
- `printf 'GET / HTTP/1.1\nHost: example.com\n\n' | nc example.com 80`
- `type in any weird HTTP request you want! 😁`

I ❤️ that sending files trick! It works on your local network even if you're not connected to the internet!
Happy hunting!