Plan

* What is encapsulation?
* Prog Lens
* Application to networks

Logistics

* PP prep report assignment due 3/30
* Hands-on (networking) due next Wednesday 3/24
* Read for Thursday: Ron
Layering & Encapsulation

Let's talk about U.S. mail

* English/Russian/Spanish
  * Room/Name
  * 32 Vason St
  * CA

- SF
- Truck
- Boston

- Mail
- Boston
Encapsulation

* Need to obey the requirements of the layer below.

  - size/weight
  - type of content
  - address well formatted
  - authentication/payment/rate limiting

* Sender needs to account for guarantees of layer below

  - reliability
  - speed/priority
Benefits

+ Scalability (address space)
+ Separation of concerns
+ Privacy (encryption)

Drawbacks

- Need to worry about layer below and what guarantees it does/doesn't provide
- Any layer can fail ⇒ problem
- Inefficiency ⇒
- "Dumb network"
Layering in Networks

Application - What do you want?
HTTP, Web, FTP, DNS, SSH
(English text of the letter)

Transport - How do you want it?
TCP = stream of bytes
UDP = individual packets
Which app on your computer gets which messages
(room # on envelope) = "port #"

Network - Where is it?
IP = address on Internet (12.5.72.80)

Link - How is it getting there?
Ethernet, Wi-Fi, DSL, Cable, ...

Physical
When you send data over Internet...

```
C-①JGET/home.htm
```

```
Ethernet
- src/dst
- chassis

IP
- header
- src/dst addr
- length
- version
- protocol

TCP
- header
- src/dst port
- checksum
- seq #

HTTP app data
```
Plan

* What is encapsulation?
* Prog lens
* Application to networks

Logistics

* PP prep report assignment due 3/30
* Hands-on (networking) due next Wednesday 3/24
* Read for Thursday: Ron
Layering & Encapsulation

Helpful to think about in context: U.S. mail

English/Spanish/Arabic/Swahili
("Application")

U.S. Mail 1st class (address, formatting, size, weight, postage)

U.S. Mail: SF to Boston

Fax: 123-456-7890

Email: info@company.com

Address:

SF, California 94111

Boston, Massachusetts 02101

Sign:

→
Encapsulation

* Need to obey the requirements of the layer below
  - address formatting
  - size and weight
  - stamp/payout
  - formatting of data

* Need to be aware of the guarantees of the layer below
  - speed/priority
  - damage/corruption
  - loss

<table>
<thead>
<tr>
<th>English language</th>
</tr>
</thead>
<tbody>
<tr>
<td>U.S. Mail 1st class</td>
</tr>
<tr>
<td>Container Truck</td>
</tr>
</tbody>
</table>
 Benefits

+ Modular - mix & match
+ Clear contract/
  "Separation of concerns"
+ Easy to change
  + changes on one level
  do not affect others — it's chill.

 Drawbacks

- Not as efficient / fast
- Have to obey rules of layer
  below even if inconvenient
- "Dumb network"
  + harder to implement in return
  functionality
Layers in Networks

**Application Layer** - What do you want?
- HTTP, HTTPS, SSH, SCP
- FTP, DNS, IMAP, POP

**Transport Layer** - How do you want it?
- TCP: streams of bytes in order, no loss
- UDP: individual packets (video, audio)
- port # 1-65535
- http://mit.edu:8080/

**Network Layer** - Where is it?
- IP = address on internet
  - 72.31.5.123
  - 32-bit addr (A)

**Link Layer** - How are packets moving across a link?
- Ethernet (aa: bb: cc: dd: ee: ff)
- 48-bit MAC

**Physical**

Diagram: Hourglass shape
HTTP

GET /home.html

IP header:
- src/dst IP
- checksum
- length
- version

TCP header:
- src/dst port

App data:

- app data

Eth header:
- src/dst MAC addr.