## CSE/EE 461 – Lecture 2

## Protocols and Layering

---

## Last Time …

- Networks are used to share distributed resources
  - Key problems revolve around effective resource sharing
- Statistical multiplexing
  - It's well-suited to data communications



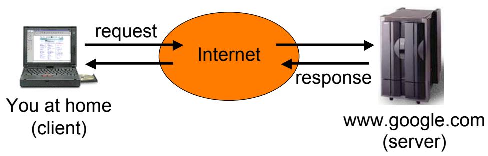
---

## This Lecture

1. A top-down look at the Internet
2. Mechanics of protocols and layering
3. The OSI/Internet models

## 1. A Brief Tour of the Internet

- What happens when you “click” on a web link?

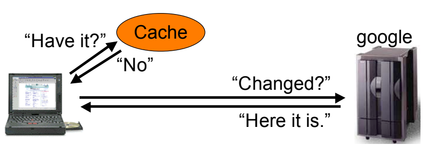



- This is the view from 10,000 ft …

CSE 461 L2.4

## 9,000 ft: Scalability

- Caching improves scalability

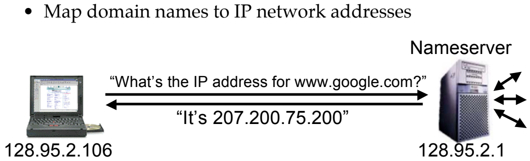



- We cut down on transfers:
  - Check cache (local or proxy) for a copy
  - Check with server for a new version

CSE 461 L2.5

## 8,000 ft: Naming (DNS)

- Map domain names to IP network addresses

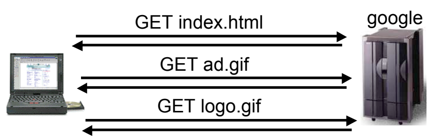



- All messages are sent using IP addresses
  - So we have to translate names to addresses first
  - But we cache translations to avoid doing it next time (why?)

CSE 461 L2.6

## 7,000 ft: Sessions (HTTP)

- A single web page can be multiple "objects"

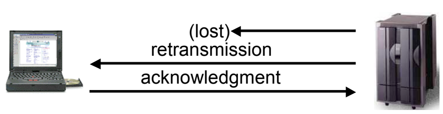



- Fetch each "object"
  - either sequentially or in parallel



## 6,000 ft: Reliability (TCP)

- Messages can get lost

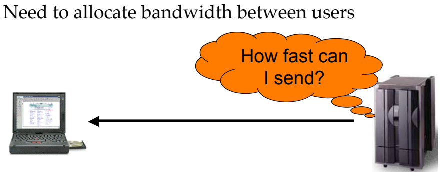



- We acknowledge successful receipt and detect and retransmit lost messages (e.g., timeouts)



## 5,000 ft: Congestion (TCP)

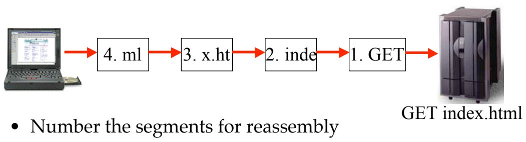
- Need to allocate bandwidth between users




- Senders balance available and required bandwidths by probing network path and observing the response

## 4,000 ft: Packets (TCP/IP)

- Long messages are broken into packets
  - Maximum Ethernet packet is 1.5 Kbytes
  - Typical web page is 10 Kbytes

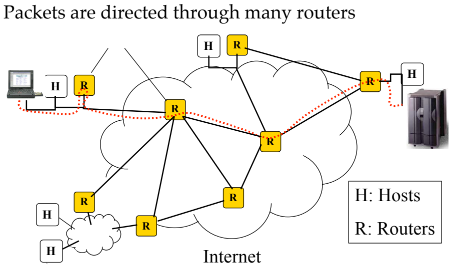



- Number the segments for reassembly



## 3,000 ft: Routing (IP)

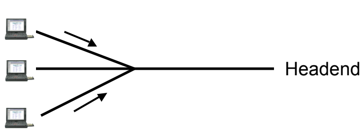
- Packets are directed through many routers






## 2,000 ft: Multi-access (e.g., Cable)

- May need to share links with other senders




- Poll headend to receive a timeslot to send upstream
  - Headend controls all downstream transmissions
  - A lower level of addressing (than IP addresses) is used … why?

## 1,000 ft: Framing/Modulation

- Protect, delimit and modulate payload as signal

| Sync / Unique | Header | Payload w/ error correcting code |
|---|---|---|

- E.g, for cable, take payload, add error protection (Reed-Solomon), header and framing, then turn into a signal
  - Modulate data to assigned channel and time (upstream)
  - Downstream, 6 MHz (~30 Mbps), Upstream ~2 MHz (~3 Mbps)



## 2. Protocols and Layering

- We need abstractions to handle all this system complexity

*A protocol is an agreement dictating the form and function of data exchanged between parties to effect communication*

- Two parts:
  - Syntax: format -- where the bits go
  - Semantics: meaning -- what the words mean, what to do with them
- Examples:
  - Ordering food from a drive-through window
  - IP, the Internet protocol
  - TCP and HTTP, for the Web

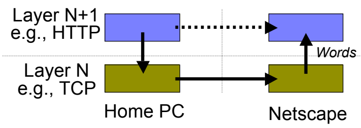


## Protocol Standards

- Different functions require different protocols
- Thus there are many protocol standards
  - E.g., IP, TCP, UDP, HTTP, DNS, FTP, SMTP, NNTP, ARP, Ethernet/802.3, 802.11, RIP, OPSF, 802.1D, NFS, ICMP, IGMP, DVMRP, IPSEC, PIM-SM, BGP, …
- Organizations: IETF, IEEE, ITU
- IETF (www.ietf.org) specifies Internet-related protocols
  - RFCs (Requests for Comments)
  - "We reject kings, presidents and voting. We believe in rough consensus and running code." – Dave Clark.

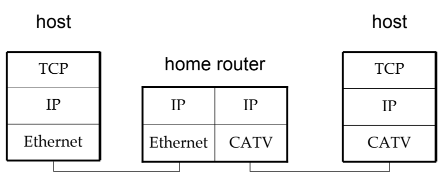
## Layering and Protocol Stacks

- Layering is how we combine protocols
  - Higher level protocols build on services provided by lower levels
  - Peer layers communicate with each other




CSE 461 L2.16

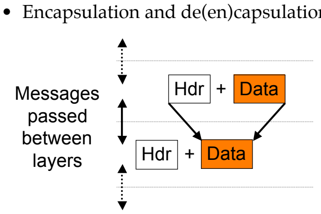
## Example – Layering at work




CSE 461 L2.17

## Layering Mechanics

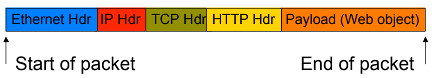
- Encapsulation and de(en)capsulation




CSE 461 L2.18

## A Packet on the Wire

- Starts looking like an onion!

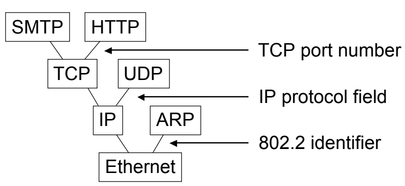



- This isn't entirely accurate
  - ignores segmentation and reassembly, Ethernet trailers, etc.
- But you can see that layering adds overhead



## More Layering Mechanics

- Multiplexing and demultiplexing in a protocol graph






## 3. OSI/Internet Protocol Stacks

Key Question: What functionality goes in which protocol?

- The "End to End Argument" (Reed, Saltzer, Clark, 1984):

  *Functionality should be implemented at a lower layer only if it can be correctly and completely implemented. (Sometimes an incomplete implementation can be useful as a performance optimization.)*

- Tends to push functions to the endpoints, which has aided the transparency and extensibility of the Internet.

## Internet Protocol Framework

| Model | Protocols |
| --- | --- |
| Application | Many (HTTP,SMTP) |
| Transport | TCP / UDP |
| Network | IP |
| Link | Many (Ethernet, …) |



| Ethernet Hdr | IP Hdr | TCP Hdr | HTTP Hdr | Payload (Web object) |
| --- | --- | --- | --- | --- |

## What's Inside a Packet

Top (start)

| | |
| --- | --- |
| Ethernet Header: | FROM=00:30:65:0a:ea:62,<br>TO=00:30:64:9a:11:22,<br>SIZE=200,… |
| IP Header: | FROM=128.95.1.32,<br>TO=28.2.5.1,<br>SIZE=200-SIZEOF(Ehdr) |
| TCP Header: | FROM=Port 5000,<br>TO=Port 80,<br>Byte#=23,<br>SIZE=200-SIZEOF(Ehdr)-SIZEOF(IPHdr) |
| HTTP Hdr: | HTTP v.1.0, Internet Explorer v5.1,… |
| Good Stuff | GET http://www.google.com |

Bottom (end)



## OSI "Seven Layer" Reference Model

| • Seven Layers: | Their functions: |
| --- | --- |
| Application | • Up to the application |
| Presentation | • Encode/decode messages |
| Session | • Manage connections |
| Transport | • Reliability, congestion control |
| Network | • Routing |
| Link | • Framing, multiple access |
| Physical | • Symbol coding, modulation |

## Key Concepts

- Protocol layers are the modularity that is used in networks to handle complexity
- The Internet/OSI models give us a roadmap of what kind of function belongs at what layer