CSEP 561 – Protocol Layers

David Wetherall
djw@cs.washington.edu
Q: How can we break down the design of networks into simpler pieces?

• The Internet (a network of networks) is complex!
• We want a modularity to help understand and evolve it

• Its constituent networks may be simple (“two home computers and a wire”) or complex themselves (over)
• Its components (“boxes” like hosts) split functionality across a key boundary – the network itself

• Key idea for describing functionality:
  – Modules called protocols that are layered on each other to provide higher-level virtual communications
Example network: NTT backbone
Components of a Network (“Hardware”)

- **Links** carry information (bits)
  - Wire, wireless, fiber optic, smoke signals …
  - May be point-to-point or broadcast
- **Switches** move bits between links
  - Routers, gateways, bridges, CATV headend, PABXs, …
- **Hosts** are the communication endpoints
  - PC, PDA, cell phone, tank, toaster, …
  - Hosts have names
- **Applications** make use of the network at hosts
  - Facebook, iTunes, VoIP phones, cameras, …

- Much other terminology too:
  - channels, nodes, intermediate systems, end systems, etc.
Functionality of Networks ("Software")

- Key idea: modularity in the form of layered protocols
- Protocols are modules that provide specific functionality
  - But composed in a constrained way
- Higher layers build on (hide) lower layers
  - Provide virtual communication at higher levels

- We’ll look at an informal example first, then formalize.
Example: Navigating with your mobile

• You open the map, your mobile uses its GPS location to fetch the map over the network for wherever you are.

• This is the view from 10,000 ft …
9,000 ft: Application logic (e.g., caching)

“Have it?”

Cache

“No”

“Changed?”

“Here it is.”

- Caching improves performance:
  - Lowers bandwidth, latency when in cache
  - Easy to check for newer version
8,000 ft: Naming (DNS)

“What’s the IP address for maps.google.com?”
“It’s 173.194.33.10”

- All messages are sent using IP addresses
  - So we have to translate names to addresses first
  - But we cache translations to avoid next time
7,000 ft: Sessions (HTTP)

- The “map” is likely to be multiple objects
  - Do HTTP fetches either sequentially or in parallel

128.95.2.106

GET tile1.jpg

GET tile2.jpg

GET tile3.jpg

Server
173.194.33.10
6,000 ft: Reliability (TCP)

- Messages can get lost, so we acknowledge receipt and detect and retransmit lost messages (e.g., timeouts)
5,000 ft: Congestion (TCP)

• Senders determine how fast they can send by probing network path and observing the response
4,000 ft: Packets (TCP/IP)

- Long messages must be broken into packets
  - Typical packet < 1.5 KB; typical web page is 10 KB
  - Number the segments for reassembly
3,000 ft: Routing (IP)

- Packets are directed through many routers
2,000 ft: Multi-access

- Need to share wireless channels among mobiles
  - “Basestation” of 3G network controls schedule of which mobile can send on what frequencies at which times
1,000 ft: Framing/Modulation

- Channels carry analog signals, not digital bits
  - Need to modulate carrier signal with data to send
  - Typically delimit start/end of packet and add redundancy to protect against corruption by noise
Protocols and Layering

- **Protocols** are modules that accomplish a specific function
- **Layering** is how we combine protocols
  - Higher level protocols build on services provided by lower levels
  - Peer layers communicate virtually with each other
  - Result is a protocol stack

![Diagram showing the relationship between Layer N+1 (e.g., HTTP) and Layer N (e.g., TCP) with Mobile and Server connected by arrows indicating communication.](image)

Protocol defines what goes over this interface

---

djw // CSEP 561, Winter 2012
Layering Mechanics

- Encapsulation (add header going down – and don’t touch data) and decapsulation (extract data going up)

Analogy is a piece of mail: envelope (header) wraps the contents (data); it is added before transit and removed and discarded after
A Packet on the Wire

- Starts looking like an onion!

- This isn’t entirely accurate
  - ignores segmentation and reassembly, trailers, etc.
- But you can see that layering adds overhead
Inside a Packet (detailed view)

802.11 Header:
FROM=00:30:65:0a:ea:62,
TO=00:30:64:9a:11:22,
SIZE=200,…

IP Header:
FROM=128.95.1.32,
TO=28.2.5.1,
SIZE=200-SIZEOF(80211hdr)

TCP Header:
FROM=Port 5000,
TO=Port 80,
Byte#=23,
SIZE=200-SIZEOF(80211hdr)-SIZEOF(IPHdr)

HTTP Hdr:
HTTP v.1.0, Internet Explorer v5.1,…

Good Stuff
GET http://www.google.com

Read rows from Top (start) to Bottom (end)
Left to Right
Example – Layering at work

We can connect across systems with different lower layers by virtual communicating between peers.
More Layering Mechanics

- Multiplexing and demultiplexing in a protocol graph
- Demultiplexing requires a demux key
OSI “Seven Layer” Reference Model

- Seven Layers:
  - Application
  - Presentation
  - Session
  - Transport
  - Network
  - Link
  - Physical

Their functions:
- Your call
- Encode/decode messages
- Manage connections
- Reliability, congestion control
- Routing
- Framing, multiple access
- Symbol coding, modulation

Guidance to help us come up with protocols at different layers
Internet Protocol Framework

Model

Protocols

Application
Transport
Network
Link

HTTP
SMTP
RTP
DNS
TCP
UDP
IP
ICMP
DSL
SONET
802.11
Ethernet

Simplified model, reverse-engineered from practice
Internet Protocol Framework Redux

Model

Protocols with IP as the “narrow waist”
Protocol Standards

- There are many network functions, thus many protocols
  - E.g., IP, TCP, UDP, HTTP, DNS, FTP, SMTP, NNTP, ARP, Ethernet/802.3, 802.11, RIP, OSPF, 802.1D, NFS, ICMP, IGMP, DVMRP, IPSEC, PIM-SM, BGP, …
- Each is standardized for interoperability
  - Not how to build a good product. Why?
- Organizations:
  - IETF (Internet): RFCs, e.g., RFC 2460 is IPv6
  - IEEE (communications): 802 standards, e.g., 802.11 is WiFi
  - ITU (telecommunications) “letter recs”, e.g., G.992.5 is ADSL
  - Various forums like 3GPP (cellular), e.g., R6 is UMTS w/ HSPA
Layering and wireless

• Models are not a good fit to wireless – they were developed in the context of early wired networks

• Concepts don’t always translate well, adding complexity
  – e.g., what is a link? And it has a time-varying capacity!
  – e.g., hosts are mobile now. That will mess with routing!

• Layering hides information that is often useful to improve wireless performance (see example over)
  – Much “cross-layer” wireless research

• Despite these downsides, the modularity of protocols has proved valuable across networking including wireless
Example: Opportunistic routing

Classic layered model
1. Pick entire route
2. Send over each link

Cross-layer scenario
1. Send over a link
2. Pick the next step

Wireless broadcast happens to be received a long way away