

Internet: The Protocol Stack

Jeffery von Ronne

Department of Computer Science
Carnegie Mellon University

April 9, 2011

Layering Abstractions

It is often useful to divide large systems into layers, takes advantage of the facilities provided by lower layers and is independent of higher layers.

Example: A Dice Game

game play “Create a function `hazard(main)` that simulates one round of hazard ...”

dice “... includes a function `roll_pair` that your should call to roll a pair of dice.”

random numbers You can get random numbers with Ruby’s `rand` function.

Standardization using API's and Protocols

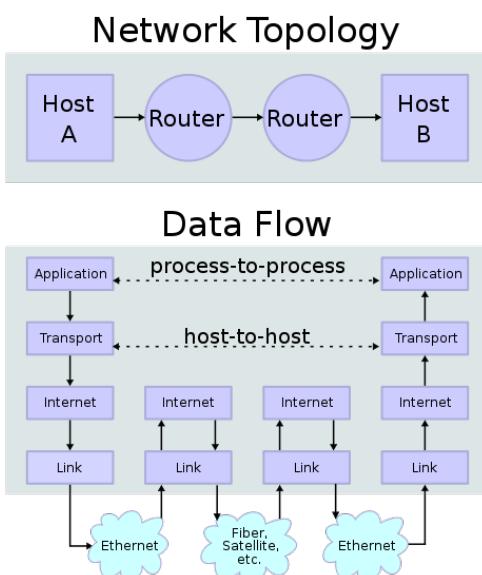
Application Programming Interface (API)

- method names, parameters, and their meaning
- provided by a lower layer to a higher layer
- may have multiple implementations

Communications Protocol

- agreement between communicating parties
 - syntax** how are the messages' contents organized?
 - semantics** what do the messages mean?
 - synchronization** when are messages sent?
- sometimes standardized as a "Request for Comments (RFC)" by the Internet Engineering Task Force (IETF)

Layers of the TCP/IP Reference Model



wikipedia.org/wiki/File:IP_stack_connections.svg

Link Layer

- transmission between adjacent hosts

Internet Layer

- logical transmission of packets between two hosts

Transport Layer

- logical connection between application processes

Application Layer

- communication of the application

Question: Which Layer?

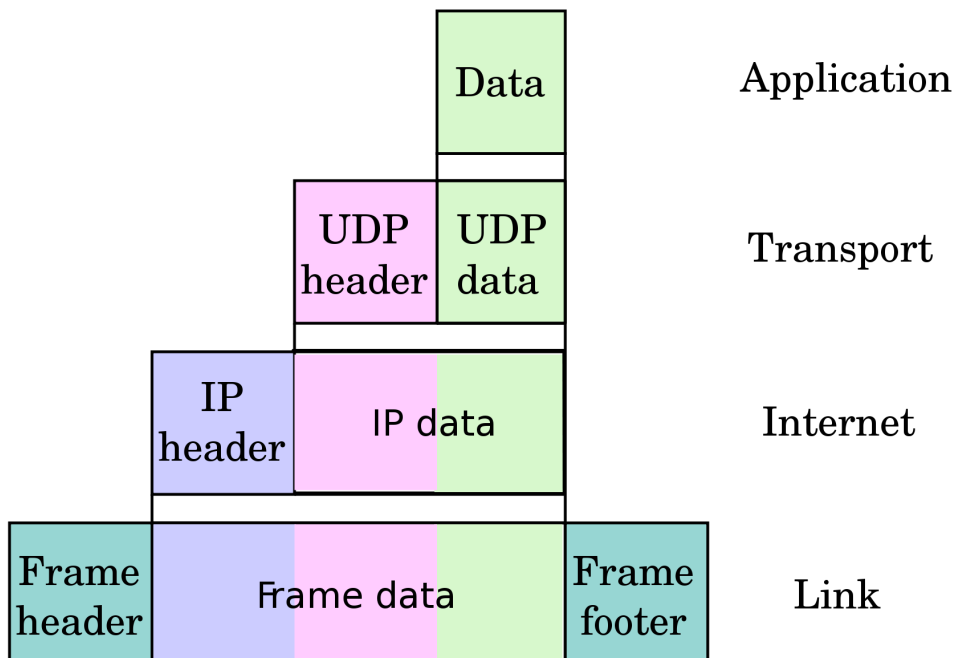
Suppose ...

The Jet Propulsion Laboratory develops a new protocol describing how frames containing data can be transmitted over radio waves directly between a satellite orbiting earth and one orbiting mars.

To which of the layers in the four-layer internet model would this protocol belong?

Nesting of Data by Layers

Data of higher-level is encapsulated in the data of lower-levels.



TCP/UDP vs. IP

Internet Protocol (IP)

- delivers packets to IP address
- best effort delivery

User Datagram Protocol (UDP)

- delivers packets to port at IP address
- port identifies
- still best effort delivery

Transmission Control Protocol (TCP)

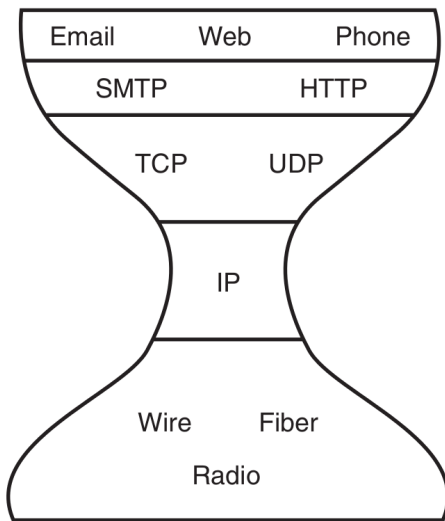
- creates a reliable bi-directional stream (source address/port and destination address/port)
 - acknowledgements, resend, reassembly in correct order, error detection
 - connection must be opened and closed
 - established with three-way handshake
 - flow/congestion control

Reliable Communication with TCP

Suppose Alice and Bob are the TCP implementations of two computers.

- Alice is asked to send a message to Bob.
- Alice breaks the message into several packets.
 - Each packet includes parity information, so Bob can check it for accuracy.
 - Packets are sent via IP.
- Bob receives the packets.
 - If Bob is missing a packet or receives a corrupt packet, he can request retransmission.
 - If the packet is OK, Bob sends an acknowledgement.
- If Alice doesn't get an acknowledgement, she will retransmit.
- Bob assembles the incoming packets in order and provides the message to the appropriate application.

Examples of TCP/IP Stack Protocols

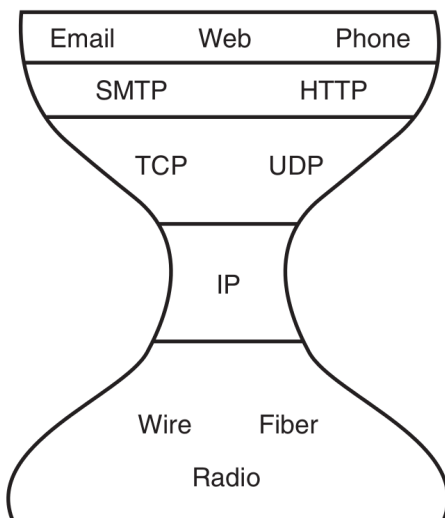


<http://www.bitsbook.com/excerpts/>

Application

- Hypertext Transfer Protocol (HTTP)
- Simple Mail Transfer Protocol (SMTP)
- Domain Name System (DNS)
XYZ.com → w.x.y.z
- Secure Shell (SSH) Protocol
ssh unix.andrew.cmu.edu
- Voice Over IP (Phone calls)
 - Session Initiation Protocol (SIP)
 - Real-time Transport Protocol (RTP)

Examples of TCP/IP Stack Protocols

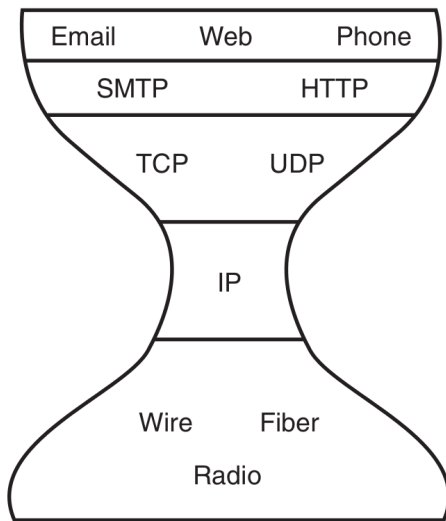


<http://www.bitsbook.com/excerpts/>

Transport

- Transmission Control Protocol (TCP)
- User Datagram Protocol (UDP)

Examples of TCP/IP Stack Protocols

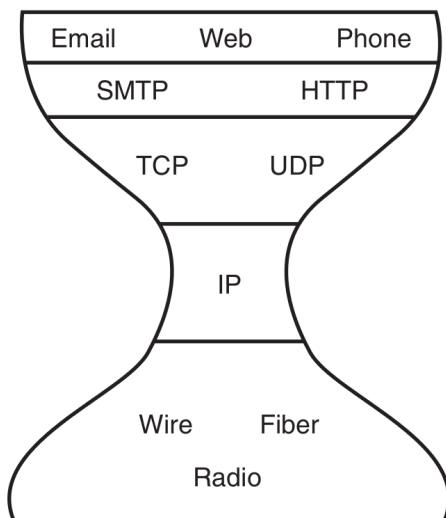


<http://www.bitsbook.com/excerpts/>

Internet

- IPv4
- IPv6

Examples of TCP/IP Stack Protocols



<http://www.bitsbook.com/excerpts/>

Link

- 1000BaseT
 - Gigabit Ethernet
- Data Over Cable Service Interface Specification
 - Cable Modems
- Long Term Evolution (LTE)
 - 4G cell phone
- 802.11N (Wi-Fi)
 - Wireless Ethernet

Hostnames

Domain Name Service (DNS)

- gets the IP address for a given name
- over TCP or UDP
- hierarchical

root name servers knows how to find dns servers for each top-level domain (e.g., "edu")

top-level domain servers know how to find dns servers for each second-level domain (e.g., "cmu.edu")

second-level domain servers know how to find each host in directly in the second-level domain (e.g., "www.cmu.edu") and how to find dns servers for each third-level domain (e.g., "andrew.cmu.edu")

...

Web

Hypertext Transfer Protocol (HTTP)

- retrieves documents (in HTML and other formats)
- over TCP (port 80)
- can also send form data to the server
- support multiple requests per connection

An HTTP Request

```
http://www.cs.cmu.edu/~vonronne/hello.txt
```

TCP connection to port 80 at 128.2.217.13:

```
GET /~vonronne/hello.txt HTTP/1.1  
Host: www.cs.cmu.edu
```

Content-Agnostic IP Routers?

- Do routers know or need to know what kind of data they are carrying?
 - No. Everything is just an IP packet with a destination, and some data
 - But they could, e.g., look at the data inside the packet and prioritize based on the contents.
- Why might an ISP do this?
 - quality of service (avoiding dropped calls)
 - prioritizing some content over others

Net Neutrality

- Should different kinds of packets be treated differently?
- The principle of net neutrality advocates no restrictions by ISPs or governments on consumers' access to networks that participate in the Internet.

Net neutrality means simply that all like Internet content must be treated alike and move at the same speed over the network. The owners of the Internet's wires cannot discriminate. This is the simple but brilliant "end-to-end" design of the Internet that has made it such a powerful force for economic and social good.

– Lawrence Lessig and Robert W. McChesney (Washington Post, June 8, 2006)

Benefits and Costs of Legislating Net Neutrality

Pros

- control of data
- digital rights and freedoms
- competition and innovation
- preservation of internet standards
- end-to-end principle

(from Wikipedia)

Cons

- property rights
- innovation and investment
- counterweight to server side non-neutrality
- bandwidth availability
- opposition to legislation

Summary

- introducing abstraction layers is often an effective strategy for reducing software complexity
- In TCP/IP, the protocol layering, allows the application code to be written as if it was passing data directly to an application on another machine without worrying about how the data gets to its destination.
- IP is the neck of the hourglass.
 - many application protocols exist above IP
 - new ones can be introduced: everything just works
 - to the routers, everything is just IP packets
 - IP can be carried on various network technologies