Internet:
The Protocol Stack

Jeffery von Ronne

Department of Computer Science
Carnegie Mellon University

15-110 Unit 11B
November 9, 2011

Outline

Layers

Common Application Level Protocols

Net Neutrality

Layering Abstractions

game play “Write a Ruby function *play_baccarat()* that plays this game ...using the *shuffle* function ... ”

card mechanics “Define a Ruby function *shuffle(a)* ...calling your *bbs_seq* method ... ”

random numbers “Implement a function bbs_seq(x0,n) that returns an array of n pseudorandom numbers.”

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

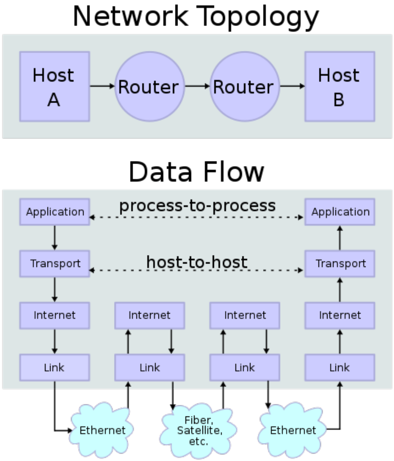
- agreement between communicating parties

syntax	how are the messages’ contents organized?
semantics	what do the messages mean?
synchronization	when are messages sent?

- standardized as a "Request for Comments (RFC)" by the Internet Engineering Task Force (IETF)

TCP/IP Reference Model

IETF’s Model, on which TCP/IP was based, uses four layers:



Source: http://en.wikipedia.org/wiki/File:IP_stack_connections.svg

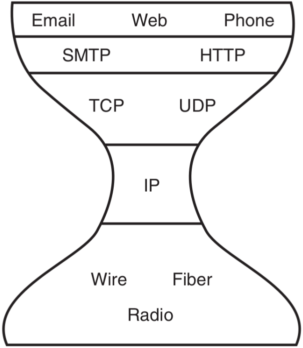
Example TCP/IP Stack Protocols

Application HTTP (web), SMTP (email), DNS (host naming), XMPP (Google Talk), SSH (secure shell); Rx (Andrew File System), RTP (Voice over IP)

Transportation Transmission Control Protocol(TCP), User Datagram Protocol (UDP)

Internet IP

Link 1000BaseT, DSL, Cabel Modem, WiFi, WiMax, LTE, HSPA+



Source:

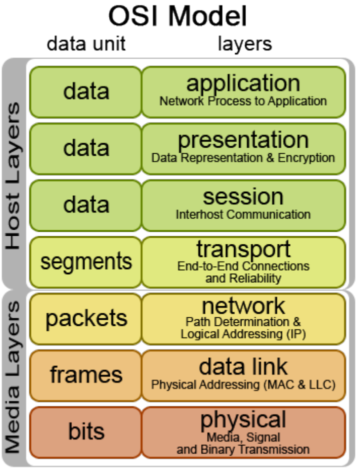
`http://www.bitsbook.com/excerpts/`

OSI Seven Layer Model

Contrast:

- IETF’s TCP/IP four-layer model
- ISO’s seven-layer Open Systems Interconnection (OSI) model

ISO is the International Organization for Standards (ISO)



OSI Model

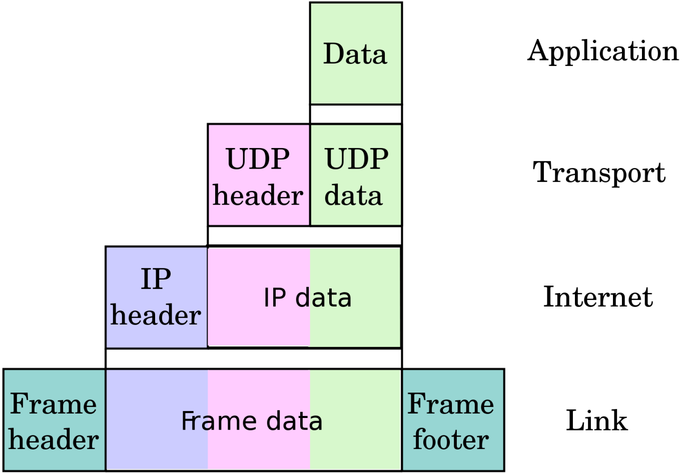
	data unit	layers
Host Layers	data	application Network Process to Application
	data	presentation Data Representation & Encryption
	data	session Interhost Communication
	segments	transport End-to-End Connections and Reliability
Media Layers	packets	network Path Determination & Logical Addressing (IP)
	frames	data link Physical Addressing (MAC & LLC)
	bits	physical Media, Signal and Binary Transmission

Source: (Dino.korah)

commons.wikimedia.org/wiki/File:Osi-model.png

Nesting of Data by Layers

In both OSI and TCP/IP, data of higher-level is encapsulated in the data of lower-levels.



Source: http://en.wikipedia.org/wiki/File:UDP_encapsulation.svg

TCP/UDP vs. IP

Internet Protocol (IP)

- delivers packets to IP address
- best effort delivery

User Datagram Protocol

- delivers packets to port at IP address
- 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

Example Requests to Each Layer

- Application Layer Requests
  - What is the IP address for www.cs.cmu.edu? (DNS)
  - What is the type and contents of http://www.cs.cmu.edu/ vonronne/hello.txt at 128.2.217.13 (HTTP)
- Transport Layer Requests (telnet example)
  1. open a TCP stream to port 80 at 128.2.217.13
  2. write to that stream:

     ```
     GET /~vonronne/hello.txt HTTP/1.1
     Host: www.cs.cmu.edu
     ```
  3. read header and data from that stream
- IP layer Request
  - send packet to 128.2.217.13
- Link layer Request
  - broadcast frame for wireless access point

Misfits

Secure Sockets Layer (SSL)
Transmission Layer Security (TLS)

- on top of TCP, below application layer
- looks mostly like TCP to upper layers
- encrypts the data packet to protect privacy
- sends encrypted data through TCP
- TLS can be activated on existing TCP connection

Virtual Private Networks / Tunneling

- take offsite computer and make it think it is plugged into a different physical network
- encryption to protect sensitive data
- openvpn
  - on top of SSL/TLS
  - below IP

Outline

Layers

Common Application Level Protocols

Net Neutrality

Hostnames

Domain Name Service (DNS)

- gets the IP address for a given name
- over TCP or UDP (port 53)
- protocol is defined by RFC 1034 and RFC 1035
- 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)
- RFC 2616
- can also send form data to the server
- support multiple requests per connection

E-Mail

Simple Mail Transfer Protocol (SMTP)

- RFC 821 (original), RFC 2821 (proposed update)
- pushes email to destination machine
- sender opens TCP connection to recipien
  - sends source and destination email address
  - sends email headers
  - sends email body
- receiver does one of the following:
  - puts the message in the user’s inbox "mail spool"
  - responds with an error (e.g., recipient doesn’t exist, mailbox full)
  - forwards the message to another host / email address
- spam
- encryption optional (through TLS)
- authentication optional

Telephony: Voice over IP and Voice over LTE

Session Initiation Protocol (SIP)

- TCP or UDP
- functionality of dialing, ringing, call-forwarding, etc.
- gets IP/port with which to communicate to your callee

Real Time Protocol (RTP)

- usually over UDP
- purpose: stream multimedia
- can carry data in different "payload types"
  - usually compressed and/or encrypted audio and/or video

Voice over LTE

- now: voice on cell phone is distinct from data
- future: voice will be carried using SIP and RTP

Extensible Messaging and Presence Protocol (XMPP)

Extensible Messaging and Presence Protocol (XMPP)

- uses
  - see if contacts are online
  - send instant messaging
  - maintain contact lists
- originally known as Jabber (used by jabber.org)
- also used by Google Talk
- servers can communicate with each other

Outline

Layers

Common Application Level Protocols

Net Neutrality

Content-Agnostic IP Routers?

- Do router’s know or need to know what kind of data they are carrying?

Content-Agnostic IP Routers?

- Do router’s know or need to know what kind of data they are carrying?

Content-Agnostic IP Routers?

- Do router’s 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

Content-Agnostic IP Routers?

- Do router’s 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.

Content-Agnostic IP Routers?

- Do router’s 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?
  - streaming media - quality of service (avoiding dropped calls)
  - prioritizing some content over others

Net Neutrality

- Should different kinds of packets be treated differntly?
- 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)

Benfits and Costs of 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
- Net neutrality
  - Are some packets more important than others?
  - Who decides and how? What are the costs?