Data Communication and Computer Networks

Mahalingam Ramkumar
Mississippi State University, MS

August 16, 2015
1 Introduction

2 Network Components
   • Functional Organization
   • Physical Organization
   • IP Address Structure
   • Autonomous Systems

3 Network Model
   • TCP/IP Model
   • Sockets
   • Tasks By Layer
   • Additional OSI Layers
Data Communication Networks

- Data (digital data in the form of bits)
- Communication (exchange of data)
- Network (the facilitator)
- For the purposes of this course:
  - bits (chunks, streams, or packets) exchanged between computers;
  - The network is the Internet
For advanced Juniors, Seniors and Graduate students
No hand-holding; no text book.
Broad topics/concepts will be discussed. You are expected to effectively utilize other online resources
C or C++ (for socket programming assignments)
Self-learn useful tools for implementing simulations in projects
  Choose your own tool (Python numpy/scipy, Matlab/Octave/Freemat etc.)
Self-learn tools like Wireshark, tcpdump, etc., for capturing and analysing network packets
Grading Policies

- Quizzes/Assignments 30%
- Exams 35% (15+20)
- Programming Assignments 10%
- Projects 15% (can earn bonus points)
- Participation 10%
Course Outline

- Overview of the Internet; layered models
- Data Communication Principles
- Internet Practices
Data Communication Principles

- Communication Theory
- Delays
- Switching
- Reliable Data Transmission
- Error Detection/Correction
- ARQ Protocols
- Issues in shared media
- Topology discovery and controlled flooding
Internet Practices

- Protocols in different Internet layers
  - DL/MAC layer protocols
  - IP protocol
  - TCP and UDP
  - Socket Library

- Applications
Why Digital?

- More Efficient
  - eg., analog vs digital TV
- Great equalizer — for transmission/reception, storage and processing
  - Does not matter if data is video, audio, image or document, or anything else
- Better suited for long range transmission
  - Can use repeaters
  - Why is this not suitable for analog transmission?
Introduction
Network Components
Network Model

Terminology

- Protocol (rigid rules/steps to be followed)
- Bit-rate (Kbps, Mbps, Gbps)
- Storage (KB, MB, GB, TB)
- Useful to be comfortable with powers of 2
  \[ 2^{10} = 1024 \approx 1000, \ 2^{20} \approx 1,000,000 \text{ (million)}, \ 2^{30} \approx 1,000,000,000 \text{ (billion)} \]
- IP (Internet Protocol) address, 32-bit unsigned number
- For convenience represented as 4 1-byte (8-bit) numbers (each in the range 0 to 255)
- Example, 129.36.42.207.
Some Interesting Quantities

- $2^8 = 256$ possible byte values (0 to 255)
- $2^{10} = 1024 \approx 1000$
- $2^{16} = 256 \times 256 = 65536$; number of port numbers
- $2^{20} \approx 1$ million. $2^{30} \approx 1$ billion.
- $2^{25} \approx$ number of seconds in a year.
- $2^{32} \approx 4$ billion (no. of IPv4 addresses, world population 1975)
- $2^{33} \approx 8$ billion (world population in 2020?)
- $2^{42} \approx 4$ trillion (federal budget in 2015)
- $2^{48} \approx 256$ trillion (number of unique MAC addresses)
- $2^{64} \approx$ grains of sand in all beaches in the world
- $2^{78} \approx$ number of water molecules in a Tbsp of water
- $2^{97} \approx$ number of stars in the observable universe
- $2^{128}$ ?
- $2^{266}$ number of baryons in the observable universe
Network Components

- Hosts (users of the network) & Network Infrastructure (provider)
- Postal Network: sender receiver; USPS
- Telephone network: caller and callee (subscribers); Telephone companies (network)
- Mobile Telephone: subscribers; service provider.
Network Components

- Network Hosts & Network Infrastructure
- Host computers: Clients and servers
- Infrastructure: Internet service providers (ISP)
  - ISPs in different tiers. Tier III ISPs rely on Tier II ISPs, and so on
  - Tier I ISPs are “Backbone operators.”
  - Some large users (e.g., Google, Netflix, Amazon, Yahoo) may connect directly to Tier I ISPs.
- Infrastructure components: Routers and Links (fiber, twisted pair, coax, wireless, satellite)
Internet Components

- Network Hosts & Network Infrastructure
- Host computers: Clients and servers
- Infrastructure: Internet service providers (ISP)
- ISPs in different tiers. Tier III ISPs rely on Tier II ISPs, and so on
- Tier I ISPs are “Backbone operators.”
- Some large users (e.g., Google, Netflix, Amazon, Yahoo) may connect directly to Tier I ISPs.
- Infrastructure components: Routers and Links (fiber, twisted pair, coax, wireless, satellite)
Internet Components

- Network Hosts & Network Infrastructure
- Host computers: Clients and servers
- Infrastructure: Internet service providers (ISP)
  - ISPs in different tiers. Tier III ISPs rely on Tier II ISPs, and so on
  - Tier I ISPs are “Backbone operators.”
  - Some large users (e.g., Google, Netflix, Amazon, Yahoo) may connect directly to Tier I ISPs.
- Infrastructure components: Routers and Links (fiber, twisted pair, coax, wireless, satellite)
Classical Telephone Network

- Telephones connected by a wire (twisted pair copper) to a local switching office (up to 10000 phones)
- Several local switching offices are connected to an area switching office (up to 1000 LSO per ASO)
- Area switching offices are interconnected (1000 ASOs)
- Each wire (from a phone to a switching office) is associated with a unique (phone) number — eg. (662) 323-xxxx
  - area code 662 identifies area switching office
  - 323 identifies local switching office;
  - every wire terminating at switching office 323 has a unique four digit number xxxx.
When the telephone receiver is lifted off the base the wire is activated (user hears a dial tone)

- User dials a destination number
- this is an instruction to the LSO to establish a path between the two phones

662 323 5678 ↔ 301 506 7259
662 323 5678 ↔ LSO 323 ↔ ASO 662 ↔ ASO 301 ↔ LSO 506 ↔ 301 506 7259

the path can be used for sending electrical signals between the two phones.
Classical Telephone Network

- Mouthpiece converts sound vibrations (pressure) to an electrical signal.
- Electrical signals conveyed over the wire (at the speed of light in copper) to the telephone at the other end.
- At the receiver, earpiece converts electrical signals to pressure vibrations (sound).
- Replacing the telephone on the hook terminates the connection.
- Telephone company keeps a record of the destination and duration of your call (for billing purposes).
Classical Telephone Network

- Mouthpiece converts sound vibrations (pressure) to an electrical signal
- Electrical signals conveyed over the wire (at the speed of light in copper) to the telephone at the other end.
- At the receiver, earpiece converts electrical signals to pressure vibrations (sound)
- Replacing the telephone on the hook terminates the connection.
- Telephone company keeps a record of the destination and duration of your call (for billing purposes).
Postal Network

- Users address mail/packages and drop them into boxes.
- Picked up by postal carriers
- Depending on the final destination the mail/package may be routed over several hops
- Possibly different modes of transportation at each hop (foot, bicycle, van, train, airplane, ···)
- Mail finally delivered to the destination address.
- Postal network uses **packet switching**. Telephone network uses **circuit switching**
Postal Network vs Telephone Network

- Postal network uses **packet switching**.
- Telephone network uses **circuit switching**.
The Internet at a Glance

- Hosts (computers), inter-connected routers (subnet);
- Every host has a unique IP address (32-bit address, about 4 billion unique addresses).
- Any host can send a packet (an IP packet) to any host.
- IP packets have a header indicating the IP addresses of the sender and the destination.
- IP packet also have a payload (any data that the sending host wants to send to the destination host)
- IP Packets are typically delivered over multiple hops — a router at each hop.
Routers and Links

- Every host has a direct connection to one router.
- A router can determine the best path to any other router.
- Does packet switching.
- At each link (between any two hops) IP packets are carried inside a data-link frame.
- The type of the data-link frame depends on the physical nature of the link (fiber, twisted pair, coax, wireless, satellite, etc.)
Packet Switching in the Internet

Sending process

Router C makes a choice to forward packets to E and not to D

Receiving process
Postal Network vs Internet

- The Internet is more like the postal network (and less like the telephone network)
  - Both employ packet switching
  - Street Address $\leftrightarrow$ IP address
  - Postal envelopes $\leftrightarrow$ IP packets;
  - Post office sorting facilities $\leftrightarrow$ routers
  - transportation modes (road+truck) $\leftrightarrow$ link + data-link frames
Postal Network vs Internet

- Postal envelopes carry a letter from sender to receiver; IP packets carry an *application message*;
- different transportation mechanisms over each hop
  - truck over road, train over rails, ships over oceans/waterways etc.
  - different types of data-link frames over different types of physical layers;
  - Ethernet over twisted pair, Wireless Ethernet over air, ALOHA over satellite links, FTTH over fiber, etc.
- The main difference is the speed — IP packets travel at the speed of light
- The Internet is postal network on steroids!
Virtual Link Between Hosts

- The speed of exchanges motivates sophisticated Internet based applications
- Applications running on hosts use the Internet to communicate with applications running on other hosts (possibly even at the other side of the globe)
- From the perspective of applications, the Internet enables a virtual link between hosts.
Functional Organization — Internet Layers

- Application Layer (AL)
- Transport Layer (TL)
- Network Layer (NL)
- Data-Link Layer (DL)
- Physical Layer (PL)

Each layer has a specific function.
Components in one layer can be modified without affecting other layers.
All Top Notch Donut Places (useful? mnemonic)
Lower and Upper Layers

- The two lower layers (PL and DL) are necessary for the functioning of the NL (to create a virtual link between hosts)
- The two upper layers (TL and AL) are required to use the virtual link
Lower Layers

- **Physical Layer** — convert data-link frames to electrical signals for delivery over a physical medium

- **Data-link Layer** — preparing data-link frames (which carry IP packets) for efficient utilization of the physical layer (physical layer could be unreliable)

- **Network layer** — determines how IP packets will need to be routed, depending on the destination IP address
Upper Layers

- **Transport Layer** — converts the unreliable “virtual link” between two hosts (thanks to Network layer) into a reliable *connection*, to carry *application messages*

- **Application Layer** — uses the reliable virtual link to send application messages (between an application running on one host and an application running on another host)
Apparent and Actual Data Paths

AM App. Message

AL → AL
Apparent and Actual Data Paths

AL  \longrightarrow  AM  \longrightarrow  AL

 TL  \longrightarrow  TH||AM  \longrightarrow  TL
Apparent and Actual Data Paths

- App. Message (AM)
- Trans. Segment (TH∥AM)
- Net. Packet (NH∥TH∥AM)

Diagram:

```
AL -----> AM -----> AL
↑        ↑        ↑
TL -----> TH∥AM -----> TL
↑        ↑        ↑
NL -----> NH∥TH∥AM -----> NL
```

Net. Packet: NL → TL → TH∥AM → AL
Apparent and Actual Data Paths

\[ \text{App. Message} \]

\[ \text{Trans. Segment} \]

\[ \text{Net. Packet} \]

\[ \text{DL Frame} \]

\[ \text{DL Frame} = \text{DH}||\text{NH}||\text{TH}||\text{AM}||\text{DF} \]
Apparent and Actual Data Paths

\[ \text{App. Message} \]

\[ \text{Trans. Segment} \]

\[ \text{Net. Packet} \]

\[ \text{DL Frame} = \text{DH} || \text{NH} || \text{TH} || \text{AM} || \text{DF} \]
Apparent and Actual Data Paths

AM  App. Message

TH||AM  Trans. Segment

Net. Packet

NL

NH||TH||AM

DL Frame

DL Frame = DH||NH||TH||AM||DF

Discussion on APPARENT AND ACTUAL DATA PATHS.
Apparent and Actual Data Paths

App. Message

<table>
<thead>
<tr>
<th>AL</th>
<th>AM</th>
<th>AL</th>
</tr>
</thead>
<tbody>
<tr>
<td>TL</td>
<td>TH</td>
<td></td>
</tr>
<tr>
<td>NL</td>
<td>NH</td>
<td></td>
</tr>
<tr>
<td>DL</td>
<td>DH</td>
<td></td>
</tr>
<tr>
<td>PL</td>
<td>PL</td>
<td>PL</td>
</tr>
</tbody>
</table>
Apparent and Actual Data Paths

**Virtual Link**

- **Net. Packet**
  - **NH||TH||AM**

- **DL Frame**
  - **DL Frame = DH||NH||TH||AM||DF**

**App. Message**

- **AM**

**Trans. Segment**

- **TH||AM**
Apparent and Actual Data Paths

\[ \text{NL} \rightarrow \text{NL} \rightarrow \text{DL} \rightarrow \text{PL} \rightarrow \text{PL} \]
Physical Organization

- Several hosts connected to a router in the same local area network (LAN)
- LANs may be interconnected by routers to form small networks
- Small networks may be interconnected to form larger networks (for example, an organization)
- Organizations connect to an Internet Service Provider (ISP)
- ISPs may form several tiers
- Lower tier ISPs connect to higher tier ISPs
- ISPs at the highest tier interconnected by the Internet backbones.
LAN

(a) Linear LAN

(b) Ring LAN
Interconnection of LANs
Interconnection of Networks
Interconnection of Networks
Interconnection of Networks
Interconnection of ISPs
IP Addresses

- 32-bit unsigned number,
- for example, 10101110 01001000 11000010 11001001
- which is 2,924,004,041
- more conveniently represented as 174.72.194.201
- About 4 billion unique addresses
- Another example, 130.18.205.15 is actually

\[
10000010 00010010 11001101 00001111 = 2,182,270,223 (1)
\]
IP Address Chunks

- Typically, consecutive IP addresses assigned to hosts in the same LAN
- For example, a LAN with less than 128 (but greater than 64) computers can be assigned addresses 130.18.205.0 to 130.18.205.127, which can all be collectively represented as

\[
10000010 \ 00010010 \ 11001101 \ 0xxxxxxx \quad (2)
\]

- First 32-7=25 bits are the same in all such addresses
- A short form for representing such collection is 130.18.205.0/25 (starting address, and the number of common bits for all addresses in the chunk).
- IP Prefix notation (to represent a chunk of \(2^r\) consecutive addresses, where \(r\) is an integer).
Assume an adjacent LAN has addresses in the range 130.18.205.128 to 130.18.205.255
Or prefix 130.18.205.128/25

10000010 00010010 11001101 1xxxxxxx  (3)

Both chunks can be consolidated as 130.18.205.0/24, or

10000010 00010010 11001101 xxxxxxxxx  (4)
IP Aggregation

```
130.18.205.0/25  R1
     |         | 130.18.205.0/24
     |         | R3
130.18.205.128/25 R2
     |         | 130.18.205.0/24
     |         | R4
     |         | 130.18.204.0/24
     |         | 130.18.204.0/23
     |         | R5
```

```
130.18.204.0/23
```
IP Aggregation

- **R1**: 130.18.205.0/25
- **R2**: 130.18.205.128/25
- **R3**: 130.18.205.0/24
- **R4**: 130.18.204.0/24
- **R5**: 130.18.204.0/23

The diagram shows the network topology with IP address aggregation.
IP Aggregation

- R1: 130.18.205.0/25
- R2: 130.18.205.128/25
- R3: 130.18.205.0/24
- R4: 130.18.204.0/24
- R5: 130.18.204.0/23

The network diagram illustrates the IP address aggregation with different IP addresses and routers (R1 to R5).
IP Aggregation

130.18.205.0/25

R1

130.18.204.0/24

130.18.205.0/24

R3

130.18.204.0/23

R5

130.18.204.0/24

R4

130.18.205.128/25

R2

130.18.205.0/24

130.18.205.128/25

R2

130.18.204.0/24

130.18.204.0/24

R4

130.18.204.0/24

R5
The Internet is an interconnection of autonomous systems (AS)
- ASes advertise IP prefixes to other ASes
- Every independent organization need not be an AS
- An AS may include any number of networks / organizations.
- What is the difference between an organizational network and an AS?
Interconnection of Autonomous Systems

Organization vs AS

- O1 \cdots O3 are organizations under AS A1;
- A_1 advertises prefixes belonging to O1 \cdots O3.
- Typically, ASes like A1 and A2 are ISPs;
- A3, A4, A5 are also organizations that use ISPs like A1, A2.
- But they have to be ASes because they have to specify some special policies to others, to reach computers in their organization.
- A4 and A5 have a “peering” arrangement
- Organization A3 uses two ISPs.

Example Topology

- O4
- O5
- O6
- O1
- O2
- O3
- A1
- A2
- A3
- A4
- A5
- A6
- A7
Structure of the Internet

Introduction
Network Components
Network Model

Functional Organization
Physical Organization
IP Address Structure
Autonomous Systems

Client
Corporate LAN
Telephone system
Server farm
Router
Regional ISP
POP
Backbone
NAP
Network Layer Models

- TCP/IP Model (5 layers)
- OSI (Open Systems Interconnection) Model (7 layers)
  - Application
  - Presentation Layer
  - Session Layer
  - Transport (TCP / UDP)
  - Network (IP)
  - Datalink / MAC (Medium Access Control)
  - Physical layer

- Presentation and Session Layers missing in TCP/IP Model
- All People Seem To Need Data Processing (7-layer mnemonic)
Physical Layer

- Hardware for physically carrying data
- Over wires, or wireless links
- modems, Ethernet/Wifi card, etc.
Data Link Layer

- send a packet of bits from one computer to another when a direct connection exists between the computers
- Depends on the nature of the physical medium used
- Two broad categories depending on if the physical medium is shared.
- For shared media DL protocols includes protocols for Medium Access Control (MAC).
  - Every computer that shares the physical medium should have a unique MAC layer address
  - Contention for channel access needs to be addressed
Network Layer

- Provide a virtual link between two computers.
- To send a packet from one computer to another
- A unique network address for every computer
- Routers relay packets over multiple hops
- Internet protocol (IP) — every computer has a unique IP address
Application and Transport Layer

- Protocols for applications/processes running on different computers to communicate with each other
- Rely on the “virtual link” facilitated by the lower layers.
Client-server model
  (Client establishes a connection with the server)
  Client sends a request
  Server responds
  (Close connection)

Some applications: E-mail, WWW, IM, FTP, File sharing, ···
Transport Layer

- Provides a “reliable connection” between processes running on different computers
- Takes care of many low-level details for creating, maintaining and closing connections
- Different processes running on a computer differentiated by unique process addresses (port numbers)
Application Developer View of Internet

- Every computer connected to the Internet has a unique IP address.
- Corresponding to each IP address, there may be many application processes willing to accept **connection requests** at a specific **port** number.
- Servers *listen* — wait for connection requests.
- Clients *initiate* connection requests.
- Once a connection is established, both client and server can send and receive (any number of) bytes.
- Connections can then be closed.
Client Server Applications

Server
0. Listen at some port number
2. Accept connection request
4. Process Query
5. Send Response
7/8. Close Connection

Client
1. Request Connection
3. Send Query
6. Receive response
7/8. Close connection
Addressing servers and Clients

- Client needs to know IP address and port number of the server.
- Usually clients know only the domain name (for e.g., yahoo.com)
- DNS (domain name system/service) — an application that translates domain names to IP addresses (like 411 Directory service);
- Port number depends on the type of application (standard port numbers)
- Connection request made by specifying IP address and port number of the server,
- Also conveys IP address and port number used by the client.
Sockets and Socket Programming

- Software library for network application development;
- Sockets are similar to file handles;
- File handles bound to a file name (and path); Sockets bound to socket address;
- Socket address has two components — an IP address and a port number;
- Two types of sockets - TCP (transport control protocol) and UDP (user datagram protocol);
- TCP sockets can be connected.
- The socket library provides various system calls like socket(), bind(), listen(), connect(), accept(), send(), recv(), close()
Socket (System) Calls

- `sd = socket(OPTIONS)`. Creates a socket. `sd` is a handle to the socket.
- `bind(sd,FROMADDRESS)` (local socket address).
- `listen()`
- `connect(sd,TOADDRESS)` (remote socket address).
- `sd = accept(sd)` (accept() returns a connected socket)
- `send(sd,buffer, num_bytes)`.
- `recv(sd,buffer, num_bytes)`.
- `close(sd)`.

These functions are the *interfaces provided by the transport layer* (to the application layer above).
Transport Layer Tasks

- Establish connections; reliable delivery of application data; differentiating between different communication instances using port numbers;
- Break up application data into message chunks; add a TRANSPORT header to each message (Transport packet = TRANSPORT header + application message)
- Send / receive transport packets to / from the network layer.
- Flow control: what happens if the sender has a high bandwidth connection and the receiver has a low bandwidth connection?
- Congestion control (helps NL in this process)
AL and TL

(Client) AL; port 1546
Transport Layer
NL; IP 120.5.246.47

App. Data
Trans. Packet = Trans Hdr + App Data
Net. Packet = Net Hdr + Trans Packet

(Server) AL; port 80
Transport Layer
NL; IP 201.234.56.78

Trans. Hdr (1546,80, flags, sequence/ack no. . . .)

Net. Hdr (120.5.246.47,201.234.56.78, . . .)
Network Layer Tasks

- Create IP packets (transport payload + IP header)
- Routing
- Determine best next hop to send an IP packet; the final destination for the received packet can be
  - itself — accept and provide contents (transport packet) to higher layer;
  - a host that is directly reachable (one hop away) — find data-link address of next hop in order to send the IP packet (enclosed in an appropriately addressed data-link frame).
  - a host that is not directly reachable (may be multiple hops away) — find data-link address of the next-hop router.
- Congestion control
- Fragment/reassemble IP fragments
DL and PL

- DL / MAC layer
  - Enclose IP packet (to be sent to next hop) in a DL / MAC frame
  - IP packet may be split into multiple smaller chunks (if the DL cannot handle large payloads)
  - MAC Frame = MAC Header + Payload (IP packet) + footer + framing flags at either end.
  - MAC header includes MAC addresses of sender and receiver (one hop away from sender)
  - MAC/DL footer for cyclic redundancy checks (CRC) — for error detection.
  - Provide the MAC frame to hardware (Ethernet card, modem) using interfaces to the hardware (drivers)

- Physical Layer: Convert bits to electrical/electromagnetic signals and send them over the link
Layer Hierarchy

**AL**

Data Packet

**TL**

- Destination IP
- Destination Port
- Origin IP
- Origin Port

**NL**

- Next Stop DL Address
- Origin DL Address

**DL**

- Conversion to Electrical Signals

**Pl**

**TH** - TCP HEADER, **NH** - IP HEADER, **DH** - MAC HEADER
Presentation Layer

- Syntax and semantics of information transmitted
- The same sequence of bytes can have different interpretations in different machines/CPUs — Big-endian vs little-endian
- All machines need to agree on some byte order (for representing 32-bit IP address and 16-bit port numbers)
- Network order, host order
- For TCP/IP translations are provided by some helper functions - htons(), htonl(), ntohs(), ntohl() included in the socket library.
Session Layer

- Establishing sessions
- Dialog control (who goes next?)
- Synchronization (continue seamlessly from previous connection)
- In TCP/IP model, dialog control and synchronization is the responsibility of the application layer
- Examples?