AISE 4430

Intro to Networking, Security, and IoT Systems

Arnav Goyal

2025-2026

Contents

1 Introduction to Networking

1.1 What is the Internet

The **internet** is essentially billions of connected computing devices, the internet features the following things:

- **Hosts** - end systems that run apps at the internet's edge
- **packet switches** - forward *packets* of data to routers and switches
- **communication-links** - fiber, copper, radio, satellite all featuring a *bandwidth*
- **Networks** - collections of devices, routers and links, managed by some organization

As such we can say that the internet is a network of networks featuring a bunch of interconnected ISPs. These connections often feature protocols, which are a set of rules based on how to communicate things.

The internet can also be thought of as a sort of *infrastructure* that provides services to applications such as the web, streaming, multimedia, teleconferencing, email, etc. The internet essentially provides a **programming interface** to distributed applications

1.2 What is a protocol?

All communication happens between computers, and this communication is governed by a **protocol** , which defines the format, order of messages sent, and actions taken on message transmission and reception.

1.3 A closer look

At the local level, one access ISP can feature many networks, such as school, a home, a company, etc. Along with cell phone services. But at the internet level we have *millions* of ISPs, how can we connect them together?

The idea is to have a few global transit ISPs, that connect together, this forms a sort of *central core* of the internet. Furthermore, **content provider networks** such as Google Microsoft, or Amazon might run their own network to bring services closer to end users.

1.4 Layering

The internet is quite a complex system, and layering it properly can help to understand it better. Overall from top to bottom, these are the layers of the internet, described by the OSI model of computing

1. Application Layer
   - Supports network apps, IMAP, SMTP, HTTP
2. Transport Layer
   - process to process data transfer TCP, UDP
3. Network Layer
   - routing of data from source to destiation, IP, routing protools
4. Link Layer
   - Data transfer between neighbourinig network elements (Etheret, Wifi, etc)
5. Physical Layer
   - Physical Bits transmitted across a wire, deals with maintaining low BER as possible across several physical communication mediums.

2 Network Performance

Networks consist of devices sending packets back and forth to each other, understanding this will help us determine and characterize a networks performance

2.1 Routers

Routers represent a interchange, on the road a packet takes, it essentially sends the packet to the right place based on where it needs to go.

We can model a router as below:

1. A flop + logic stage that accepts the data and does some processing on the packet t_{proc}
2. A FIFO that queues the packet with queueing delay, t_{queue}
3. A TX that sends the packet out after a delay t_{trans}
4. A channel where the packet propagates through, reaching the destination after t_{prop}

Note that our FIFO is a fixed length, and incoming packets can be dropped if the FIFO is full! This packet might be resent by the previous node, or by the system or not at all.

2.2 Router Delay

Overall the total time spent at a node (router) before reaching the other node is defined below, sometimes this quantity is also referred to as t_{nodal}. Each term in this expression below will be explained further.

$$t_{\text{router}} = t_{\text{proc}} + t_{\text{queue}} + t_{\text{trans}} + t_{\text{prop}}$$

The flop stage, resulting in t_{proc} usually involves checking for bit errors, and algorithms for determining the proper destination, all of this typicall takes less than a millsecond

The FIFO stage results in a delay t_{queue} and it depends on the current usage of the FIFO, supposing we have a transmission rate of R bits per second, and each packet in the FIFO has a length of L bits, this queueing delay becomes dependent on n the number of packets before the current one in the FIFO, if the FIFO is empty when the packet arrives, n is equal to 0, essentially this is the delay to transmit all packets before the one of interest:

$$t_{\text{queue}} = n \cdot \frac{L}{R} = n \cdot t_{\text{trans}}$$

The Tx ends up resulting in t_{trans}, if we take our variables from earlier, R for transmission rate in bits per second, and L for packet length, we can see that the transmission delay is as described earlier in the FIFO stage delay.

$$t_{\text{trans}} = \frac{L}{R}$$

Lastly, we have the propagation delay* of the packet through the chosen channel. We can describe this delay with d which is the length of the physical link, and s which is the propagation speed through this link. Typically we assume $s \approx 2\text{E}8$. Altogether we get:

$$t_{\text{prop}} = \frac{d}{s} = \frac{d}{2 \times 10^8}$$

3 The Application Layer

Network apps include the web, texts, emails, youtube, etc. Here we will learn about the layer that they operate on

Network apps are essentially programs that run on different end systems, and communicate over a network. Developers of these apps dont actually need to write software for network-core devices (such as routers). Network-core devices do not run applications, instead they talk to end systems that actually do

3.1 Network Architectures

This introduces the **client-server** paradigm, which describes the above:

Server:

- The **Server** is an always-on host
- It features a permanent IP adress
- and is often in data-centers for faster scaling

Clients:

- Are end systems, user devices
- They can communicate directly with the server only, clients do not talk to other clients
- they might have dynamic IP adresses
- Examples: HTTP, IMAP, FTP

There is also an alternative, the **peer-to-peer** paradigm, where:

- This is no always-on server
- end systems communicate directly with each other
- peers request service from other peers, and return the service to other peers as well
  - This means that this is *self-scalable*, new peers bring new servicie capacity as well as new service demands
- Peers are intermittently connected and change IP adresses with complex management systems
- Example: P2P File Sharing (torrents)

3.2 Communication of Processes

a **process** is a program running within a host. Within the host two processes can communicate through the OS, but processes residing on different hosts must communicate by exchanging messages.

We define the term **client-process** as the process that initiates communication, and the term **server-process** as the process that waits to be contacted. Applications with P2P architectures have both client and server processes.

Processes communicate by sending messages to and from a **socket**, which can be thought of as a *door*. A sending process will:

1. Shove the message out of the door
2. Rely on the transport infrastructure outside of the door to deliver the message to the destination door on the other side

Furthermore, we need to address these processes in other to recieve messages properly. Thus each process must have an *identifier*

This identifier is called the IP address, and it belongs to the host device, that can have multiple processes running at the same time. Furthermore, there are also port numbers such as 80 or 25, that are used as part of the IP address.

3.3 Application Layer Protocols

Now that we know how processes should communicate, what sorts of things should be defined in a protocol used at the application layer?

1. Types of Messages Exchanged
   - A request, response, etc
2. Message Syntax
   - What is included in the message, and how fields are delineated
3. Message Semantics
   - What do each of the fields in the message signify?
4. Rules for when/how processes can send or respond to messages

Some examples of open protocols are HTTP and SMTP, an example of a closed (proprietary) protocol is skype

So what transport services does an app need to actually work?

1. Data Integrity
   - Some appds like file transfer apps require completely reliable data transfer, others can tolerate some loss

2. Timing
    - Some apps (ex. games) require low delay to be effective
3. Throughput
    - Some apps require minimum amount of throughput to be effective
    - Other apps, called *elastic* make use of whatever throughput they can get
4. Security

3.4 Transport Layer Protocols

The application layer heavily depends on the services of the layer below it. The transport layer features TCP and UDP as the main protocols.

TCP provides reliable transport, flow and congestion control, and is connection oriented, however it does not provide timing, a minimum throughput or any security. UDP provides unreliable data transfer, and none of the other benefits of TCP.

4 Sample Questions

4.1 Chapter 1

4.1.1 Problem 1

Consider two hosts, A and B, connected by a single link of rate R bps. Suppose that the two hosts are separated by m meters, and suppose the propagation speed along the link is s meters/s. Host A is to5. If $t_{\text{prop}} > t_{\text{trans}}$, at $t = t_{\text{trans}}$ where is the first bit? send a packet of size L bits to host B.

1. Express t_{prop} in terms of m and s

From the notes earlier we know

$$t_{\text{prop}} = \frac{m}{s}$$

2. Determine transmission time, t_{trans} of the packet in terms of L and R

$$t_{\text{trans}} = \frac{L}{R}$$

3. Ignoring queueing and processing times, what is the end-to-end delay?

$$t_{\text{router}} = t_{\text{trans}} + t_{\text{prop}} = \frac{L}{R} + \frac{m}{s}$$

4. Suppose host A begins transmission at $t = 0$, at $t = t_{\text{trans}}$ where is the last bit of the packet?

Time to transmit the whole packet is t_{trans} so after this time, the last bit will be at the start of the channel/physical link

5. If $t_{\text{prop}} > t_{\text{trans}}$, at $t = t_{\text{trans}}$ where is the first bit?

Here this means that the propagation delay for a single bit, takes longer than transmission of the entire packet, thus the first bit would be on the wire but not reached host B yet

6. If $t_{\text{prop}} < t_{\text{trans}}$, at $t = t_{\text{trans}}$ where is the first bit?

Here this means that the propagation delay for each bit takes less time than transmission of the whole packet, thus the first bit would have reached host B by now, since the entire packet has been transmitted at $t = t_{\text{trans}}$ and propgation time is less than this.

4.1.2 Problem 2

In this problem we consider sending voice from Host A to Host B over a packet-switched network (for example, Internet phone). Host A converts analog voice to a digital 64 kbps bit stream on the fly. Host A then groups the bits into 48-byte packets. There is one link between Host A and B; its transmission rate is 1 Mbps and its propagation delay is 2 msec. As soon as Host A gathers a packet, it sends it to Host B. As soon as Host B receives an entire packet, it converts the packet's bits to an analog signal. How much time elapses from the time a bit is created (from the original analog signal at Host A) until the bit is decoded (as part of the analog signal at Host B)?

From the question we have:

$$A \rightarrow \text{NODE} \rightarrow B$$

at A we have a 64Kbps bit-stream, grouped into packets of width 48 B.

so the total number of bits in a packet is, and the first packet takes this long to generate completely

$$t_{\text{first packet}} = \frac{48 \times 8}{64\text{K}} = 6\text{ms}$$

Then we need to consider the nodal time

$$t_{\text{nodal}} = t_{\text{trans}} + t_{\text{prop}} = \frac{48 \times 8}{1 \times 10^6}$$

The total time is then:

$$t_{\text{total}} = t_{\text{first packet}} + t_{\text{nodal}}$$

4.1.3 Problem 3

Consider the queuing delay in a router buffer (preceding an outbound link). Suppose all packets are L bits, the transmission rate is R bits, and that N packets simultaneously arrive at the buffer every LN/R seconds. Find the average queuing delay of a packet. (Hint: the queuing delay for the first packet is zero; for the second packet L/R; for the third packet 2L/R. the Nth packet has already been transmitted when the second batch of packets arrives)

The queuing delay for any packet is dependent on N

$$t_{\text{queue}} = N \cdot \frac{L}{R}$$

If N packets arrive every $\frac{NL}{R}$ seconds, the average queing delay for a packet is the following:

$$t_{\text{queue, avg}} = \sum_i t_{\text{queue, packet i}} \cdot \frac{1}{N}$$

{

$$t_{\text{queue, avg}} = \frac{1}{N} \cdot \left(\frac{L}{R}\right) [0 + 1 + 2 + \cdots + N - 1]$$

This simplifies to

$$t_{\text{queue, avg}} = \frac{1}{N} \cdot \frac{N(N-1)}{2} \cdot \frac{L}{R} = \frac{N-1}{2} \cdot \frac{L}{R}$$

4.1.4 Problem 4

Suppose two hosts A and B are seperated by 10 thousand km, and are connected by a direct link of R = 1 Mbps. Suppose to propagation speed is 2.5 E 8 m/s

1. Calculate the bandwith-delay product

$$R \times t_{\text{prop}} = 1\text{M} \times \frac{10000\text{E3 m}}{2.5\text{E8 m/s}}$$

2. Suppose we are sending a large file 400,000 bits wide from A to B, what is the maximum number of bits that will be in the link at any given time?

The max number of bits on the link at any given time can be found by remembering that distance = speed $\times$ time, here the speed is the propagation speed of a bit, and the time is the transmission time for one bit.

therefore we get this one expression for the width of a bit:

$$l_{\text{bit}} = \frac{s}{R}$$

Where s is bit propagation speed of a single bit in m/s, and R is the transmission bandwidth in bps

Then we can divide the link length by the bit-length to get the total number of bits on the link at any given time.

$$n_{\text{bits}} = \frac{d}{l_{\text{bit}}}$$

4.1.5 Problem 5

Suppose users share a 3 Mbps link. Also suppose that each user requires 150 kbps when transmitting, but each transmits only 10 percent of the time

1. When circuit switchinng is used, how many users can be supported?

For circuit switching the number of users supported q can be found here

$$q_{\text{circuit switching}} = \frac{R_{\text{link}}}{R_{\text{user transmit}} = \frac{3000}{150} = 20}$$

2. For the remainder of this problem suppose packet switching is used, find the probability that a given user is transmitting

Each transmits 10% of the time, this is given from the question

3. The probability that n users out of all m users are simultaneously transmitting on a link, when each user has a p chance of transmitting is

$$P_n = \binom{m}{n} \times p^n \times (1-p)^{m-n}$$

Where,

$$\binom{n}{k} = \frac{n!}{k! \cdot (n-k)!}$$

The probablity that there are 10 users out of 15 transmitting simultaneously is given by the below expression. For part c) we suppose 120 users total, what is the probabiliity that 55 are transmitting at once

$$P_{10} = \binom{120}{55} \times (0.1)^{55} \times (0.9)^{120-55}$$

the chance that there are 55 or more users transmitting is

$$1 - \sum_{n=0}^{55} \binom{120}{n} \cdot (0.1)^n \cdot (0.9)^{120-n}$$