Chapter 1

2. The word protocol is often used to describe diplomatic relations. How does Wikipedia describe diplomatic protocol?

Ans: From Wikipedia: Diplomatic protocol is commonly described as a set of international courtesy rules. These well-established and time-honored rules have made it easier for nations and people to live and work together. Part of protocol has always been the acknowledgment of the hierarchical standing of all present. Protocol rules are based on the principles of civility.

3. Why are standards important for protocols?

Ans: Standards ensure **interoperability** between different devices and systems.
They enable **open development and innovation**.
They provide **reliability and consistency** in communication.
They support the **scalability** of the Internet.

5. Is HFC transmission rate dedicated or shared among users? Are collisions possible in a downstream HFC channel? Why or why not?

Ans: HFC bandwidth is shared among the users. On the downstream channel, all packets emanate from a single source, namely, the head end. Thus, there are no collisions in the downstream channel.

10. Describe the most popular wireless Internet access technologies today. Compare and contrast them.

Ans: There are two popular wireless Internet access technologies today:

a. Wifi (802.11) In a wireless LAN, wireless users transmit/receive packets to/from a base station (i.e., wireless access point) within a radius of a few tens of meters. The base station is typically connected to the wired Internet and thus serves to connect wireless users to the wired network.
b. 3G and 4G wide-area wireless access networks. In these systems, packets are transmitted over the same wireless infrastructure used for cellular telephony, with the base station thus being managed by a telecommunications provider. This provides wireless access to users within a radius of tens of kilometers of the base station.

12. What advantage does a circuit-switched network have over a packet-switched network? What advantages does TDM have over FDM in a circuit-switched network?

Ans: A circuit-switched network can guarantee a certain amount of end-to-end bandwidth for the duration of a call. Most packet-switched networks today (including the Internet) cannot make any end-to-end guarantees for bandwidth.

No congestion in circuit switched. No Packet loss.

FDM requires sophisticated analog hardware to shift signals into appropriate frequency bands.

In TDM no cross talk or interference.

In tdm efficient use of bandwidth: if a user has no data to send , others can use the bandwidth

In tdm simpler and cheaper hardware.

13. Suppose users share a 2 Mbps link. Also suppose each user transmits continuously at 1 Mbps when transmitting, but each user transmits only 20 percent of the time.

a. When circuit switching is used, how many users can be supported?
b. For the remainder of this problem, suppose packet switching is used. Why will there be essentially no queuing delay before the link if two or fewer users transmit at the same time? Why will there be a queuing delay if three users transmit at the same time?
c. Find the probability that a given user is transmitting.
d. Suppose now there are three users. Find the probability that at any given time, all three users are transmitting simultaneously. Find the fraction of time during which the queue grows.

Ans: a) 2 users can be supported because each user requires half of the link bandwidth.
b) Since each user requires 1Mbps when transmitting, if two or fewer users transmit simultaneously, a maximum of 2Mbps will be required. Since the available bandwidth of the shared link is 2Mbps, there will be no queuing delay before the link. Whereas, if three users transmit simultaneously, the bandwidth required will be 3Mbps which is more than the available bandwidth of the shared link. In this case, there will be queuing delay before the link.
c) Probability that a given user is transmitting = 0.2
d) Probability that all three users are transmitting simultaneously =

$$\binom{3}{3} p^3 (1-p)^{3-3}$$

= (0.2)^3 = 0.008. Since the queue grows when all the users are transmitting, the fraction of time during which the queue grows (which is equal to the probability that all three users are transmitting simultaneously) is 0.008.

14. Why will two ISPs at the same level of the hierarchy often peer with each other? How does an IXP earn money?

Ans: **Main reasons for peering:**

1. **Cost savings:** If ISP A and ISP B send their mutual traffic through an **upstream provider**, they both have to **pay** that provider for transit.
   By peering **directly**, they **avoid transit costs**.
2. **Improved performance:** Direct peering provides a **shorter path**, reducing **latency** and improving **throughput** between the two ISPs' networks.
3. **Traffic localization:** Keeps local or regional traffic **within the region**, instead of routing it unnecessarily across distant backbone networks.
4. **Mutual benefit:** Peering is often **settlement-free** — each ISP agrees to carry the other's traffic **only for their respective customers**.

IXPs earn revenue through:

1. **Port fees / connection fees:** Members pay for the **physical port** (e.g., 1 Gbps, 10 Gbps, 100 Gbps) they use to connect to the IXP's switch fabric.
2. **Membership or setup fees:** Some IXPs charge a **one-time joining fee** or **annual membership fee**.
3. **Cross-connect fees:** Charges for establishing **fiber cross-connects** between racks or colocation facilities.

19. Suppose Host A wants to send a large file to Host B. The path from Host A to Host B has three links, of rates R1 = 500 kbps, R2 = 2 Mbps, and R3 = 1 Mbps.

a. Assuming no other traffic in the network, what is the throughput for the file transfer?
b. Suppose the file is 4 million bytes. Dividing the file size by the throughput, roughly how long will it take to transfer the file to Host B?
c. Repeat (a) and (b), but now with R2 reduced to 100 kbps.

Ans: Here we treat the end-to-end throughput for a large file as the bottleneck (minimum) link rate.

a) Throughput = min{R1, R2, R3} = min{500 kbps, 2 Mbps, 1 Mbps} = **500 kbps**.

b) File size = 4,000,000 bytes = 32,000,000 bits = **32 Mb**.
Time = (file size) / (throughput) = 32 Mb/0.5 Mb/s=64 s **≈ 64 seconds.**

c) With R2= 100 kbps: throughput = min{500, 100, 1000} kbps = **100 kbps**.
Time = 32 Mb/0.1 Mb/s=320 s=5 min 20 s **≈ 320 seconds (5 min 20 s).**

22. List five tasks that a layer can perform. Is it possible that one (or more) of these tasks could be performed by two (or more) layers?

Ans: Five generic tasks are error control, flow control, segmentation and reassembly, multiplexing, and connection setup. Yes, these tasks can be duplicated at different layers. For example, error control is often provided at more than one layer.

23. What are the five layers in the Internet protocol stack? What are the principal

responsibilities of each of these layers?
Ans:
5. Application← Network applications (HTTP, SMTP, DNS)
4. Transport← TCP, UDP – process-to-process delivery
3. Network ← IP – routing of packets across networks
2. Link← Ethernet, Wi-Fi – node-to-node frame deliver
1. Physical← Bits on wire, radio, or fiber

24. What is an application-layer message? A transport-layer segment? A network-layer datagram? A link-layer frame?

Ans: Application-layer message: data which an application wants to send and passed onto the transport layer;
transport-layer segment: generated by the transport layer and encapsulates application-layer message with transport layer header;
Network-layer datagram: encapsulates transport-layer segment with a network-layer header;
link-layer frame: encapsulates network-layer datagram with a link-layer header.

25. Which layers in the Internet protocol stack does a router process? Which layers does a link-layer switch process? Which layers does a host process?

Ans: Router - layer 1- 3; Link-layer switch - layer 1-2; Host - 1-5

26. What is self-replicating malware?

Ans: Self-replicating malware is a type of malicious software that can copy itself and spread to other systems or files automatically, without needing the user to manually execute or send it.
Eg: Virus , worm , botnet malware

27. Describe how a botnet can be created and how it can be used for a DDoS attack

Ans: Common DDoS **modes** attackers can orchestrate, described at an abstract level:

- **Volumetric attacks** — overwhelm link capacity by having many bots transmit large volumes of traffic (UDP/TCP floods, ICMP floods). The goal: consume bandwidth between victim and the Internet.
- **Protocol attacks** — exploit weaknesses in protocols or stateful resources (e.g., SYN flood style behavior) to exhaust connection tables or processing capacity on routers/servers.
- **Application-layer attacks** — generate legitimate-looking application requests (e.g., many HTTP GETs) to overload web servers or databases; these are harder to distinguish from real clients.
- **Amplification/reflection (conceptual)** — attackers instruct bots to send small forged requests to open servers that reply with larger responses to the victim, multiplying traffic volume (DNS, NTP, and others have historically been abused). Discussed in networking as an interaction between many intermediaries and the victim that increases effective attack bandwidth.

P3. Consider an application that transmits data at a steady rate (for example, the sender generates an N-bit unit of data every k time units, where k is small and fixed). Also, when such an application starts, it will continue running for a relatively long period of time. Answer the following questions, briefly justifying your answer:

a. Would a packet-switched network or a circuit-switched network be more appropriate for this application? Why?
b. Suppose that a packet-switched network is used and the only traffic in this network comes from such applications as described above. Furthermore, assume that the sum of the application data rates is less than the capacities of each and every link. Is some form of congestion control needed? Why?

Answer: a) A circuit-switched network would be well suited to the application, because the application involves long sessions with predictable smooth bandwidth requirements. Since the transmission rate is known and not bursty, bandwidth can be reserved for each application session without significant waste. In addition, the overhead costs of setting
up and tearing down connections are amortized over the lengthy duration of a typical application session.
b) In the worst case, all the applications simultaneously transmit over one or more network links. However, since each link has sufficient bandwidth to handle the sum of all of the applications' data rates, no congestion (very little queuing) will occur. Given such generous link capacities, the network does not need congestion control mechanisms.

P5. car-caravan analogy in Section 1.4. Assume a propagation speed of 100 km/hour.

a. Suppose the caravan travels 175 km, beginning in front of one tollbooth, passing through a second tollbooth, and finishing just after a third toll booth. What is the end-to-end delay?
b. Repeat (a), now assuming that there are eight cars in the caravan instead of ten.

Answer: Tollbooths are 75 km apart, and the cars propagate at 175km/hr. A tollbooth services a car at a rate of one car every 12 seconds.

a) There are ten cars. It takes 120 seconds, or 2 minutes, for the first tollbooth to service the 10 cars. Each of these cars has a propagation delay of 25.7 minutes (travel 75 km) before arriving at the second tollbooth. Thus, all the cars are lined up before the second tollbooth after 27.7 minutes. The whole process repeats itself for traveling between the second and third tollbooths. It also takes 2 minutes for the third tollbooth to service the 10 cars. Thus the total delay is 57.4 minutes.
b) Delay between tollbooths is 8*12 seconds plus 25.7 minutes. The total delay is twice this amount plus 8*12 seconds, i.e., 53 minutes.

P7. We consider sending real-time voice from Host A to Host B over a packet-switched network (VoIP). Host A converts analog voice to a digital 64 kbps bit stream on the fly. Host A then groups the bits into 56-byte packets. There is one link between Hosts A and B; its transmission rate is 10 Mbps and its propagation delay is 10 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)?

Ans: Consider the first bit in a packet. Before this bit can be transmitted, all of the bits in the packet must be generated. This requires (56 * 8) / 64 * 10^3 sec = 7ms
The time required to transmit the packet is (56 * 8) / 10 * 10 ^ 6 sec = 44.8 micro seconds
Propagation delay = 10 msec.
The delay until decoding is $7m + 44.8\mu + 10m = 17.0448m$ sec.
A similar analysis shows that all bits experience a delay of 17.0448 msec.

P8. Suppose users share a 10 Mbps link. Also suppose each user requires 200 kbps when transmitting, but each user transmits only 10 percent of the time.

a. When circuit switching is used, how many users can be supported?
b. For the remainder of this problem, suppose packet switching is used. Find
the probability that a given user is transmitting.
c. Suppose there are 120 users. Find the probability that at any given time,

exactly n users are transmitting simultaneously. (Hint: Use the binomial distribution.)

Ans:

a) 50 users can be supported.

b) $p = 0.1$.

c) $\binom{120}{n} p^n (1-p)^{120-n}$.

P9. Consider the discussion in Section 1.3 of packet switching versus circuit switching in which an example is provided with a 1 Mbps link. Users are generating data at a rate of 100 kbps when busy, but are busy generating data only with probability p = 0.1. Suppose that the 1 Mbps link is replaced by a 1 Gbps link.

a. What is N, the maximum number of users that can be supported simultaneously under circuit switching?

b. Now consider packet switching and a user population of M users. Give a formula (in terms of p, M, N) for the probability that more than N users are sending data.

Ans:

a) 10,000

b) $\sum_{n=N+1}^{M} \binom{M}{n} p^n (1-p)^{M-n}$

P16. Consider a router buffer preceding an outbound link. In this problem, you will use Little's formula, a famous formula from queuing theory. Let N denote the average number of packets in the buffer plus the packet being transmitted. Let a denote the rate of packets arriving at the link. Let d denote the average total delay (i.e., the queuing delay plus the transmission delay) experienced by a packet. Little's formula is N = a * d. Suppose that on average, the buffer contains 100 packets, and the average packet queuing delay is 20 msec. The link's transmission rate is 100 packets/sec. Using Little's formula, what is the average packet arrival rate, assuming there is no packet loss?

Ans: find D:
queuing delay + transmission delay = total delay
Each packet takes 1/100 = 0.01sec to transmit
Dtotal = 20ms + 10ms = 30ms
Littles formula:
N = a * d, a = 100 / 30ms

P20. Consider the throughput example corresponding to Figure 1.20(b). Now suppose that there are M client-server pairs rather than 10. Denote Rs, Rc, and R for the rates of the server links, client links, and network link. Assume all other links have abundant capacity and that there is no other traffic in the network besides the traffic generated by the M client-server pairs. Derive a general expression for throughput in terms of Rs, Rc, R, and M.

Ans: Throughput = min{Rs, Rc, R/M}

P22. Consider Figure 1.19(b). Suppose that each link between the server and the client has a packet loss probability p, and the packet loss probabilities for these links are independent. What is the probability that a packet (sent by the server) is successfully received by the receiver? If a packet is lost in the path from the server to the client, then the server will re-transmit the packet. On average, how many times will the server re-transmit the packet in order for the client to successfully receive the packet?

Ans: Probability of successfully receiving a packet is: ps= (1-p)^N.
The number of transmissions needed to be performed until the packet is successfully received by the client is a geometric random variable with success probability ps. Thus, the average number of transmissions needed is given by: 1/ps . Then, the average number of re-transmissions needed is given by: 1/ps -1

P25. Suppose two hosts, A and B, are separated by 20,000 kilometers and are connected by a direct link of R = 5 Mbps. Suppose the propagation speed over the link is 2.5 # 108 meters/sec.

a. Calculate the bandwidth-delay product, R * d(prop).

b. Consider sending a file of 800,000 bits from Host A to Host B. Suppose the file is sent continuously as one large message. What is the maximum number of bits that will be in the link at any given time?

c. Provide an interpretation of the bandwidth-delay product.

d. What is the width (in meters) of a bit in the link? Is it longer than a football field?

e. Derive a general expression for the width of a bit in terms of the propagation speed s, the transmission rate R, and the length of the link m.

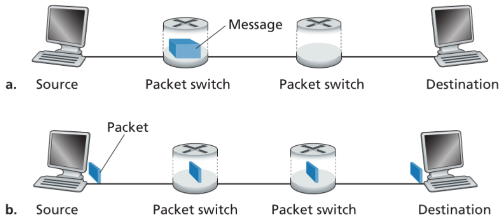
Ans: a) 400,000 bits

b) 400,000 bits

c) The bandwidth-delay product of a link is the maximum number of bits that can be in the link.

d) the width of a bit = length of link / bandwidth-delay product, so 1 bit is 125 meters long, which is longer than a football field

e) s/R



ure 1.27 ♦ End-to-end message transport: (a) without message segmentation; (b) with message segmentation

P31.

In modern packet-switched networks, including the Internet, the source host segments long, application-layer messages (for example, an image or a music file) into smaller packets and sends the packets into the network. The receiver then reassembles the packets back into the original message. We refer to this process as message segmentation. Figure 1.27 illustrates the end-to-end transport of a message with and without message segmentation. Consider a message that is 10^6 bits long that is to be sent from source to destination in Figure 1.27. Suppose each link in the figure is 5 Mbps. Ignore propagation, queuing, and processing delays.

a. Consider sending the message from source to destination without message segmentation. How long does it take to move the message from the source host to the first packet switch?

Keeping in mind that each switch uses store-and-forward packet switching, what is the total time to move the message from source host to destination host?
b. Now suppose that the message is segmented into 100 packets, with each packet being 10,000 bits long. How long does it take to move the first packet from source host to the first switch? When the first packet is being sent from the first switch to the second switch, the second packet is being sent from the source host to the first switch. At what time will the second packet be fully received at the first switch?
c. How long does it take to move the file from source host to destination host when message segmentation is used? Compare this result with your answer in part (a) and comment.
d. In addition to reducing delay, what are reasons to use message
segmentation?
e. Discuss the drawbacks of message segmentation.

Ans: a) Time to send message from source host to first packet switch = 10^6 / 5*10^6 = 0.2sec
With store-and-forward switching, the total time to move message from source host to destination host =$0.2 sec \times 3 hops = 0.6 sec$

b)transmission delay, L/R . L = 10000bits, R = 5*10^6. **t = 2msec**.

So it takes **2 ms** to move the first packet from the source host to the first switch.

Now timing the pipeline: the source sends packet 1 from t=0 to t=2 ms; the first switch, having received packet 1 at t=2 ms, begins transmitting it to the next switch from t=2 to t=4 ms. Meanwhile the source sends packet 2 from t=2 to t=4 ms. Therefore the second packet is fully received at the first switch at **4 ms**.

c)First packet reaches destination after : 3 x 2ms = 6ms

(Because it must be transmitted over 3 links one after another.)

After that, a new packet finishes every **2 ms** (since the pipeline is full).

T = 6 + (100-1)*2ms = 0.204ms

d) Without message segmentation, if bit errors are not tolerated, if there is a single bit error, the whole message has to be retransmitted (rather than a single packet).

Without message segmentation, huge packets (containing HD videos, for example) are sent into the network. Routers have to accommodate these huge packets. Smaller packets have to queue behind enormous packets and suffer unfair delays.

e) Packets have to be put in sequence at the destination.

Message segmentation results in many smaller packets. Since header size is usually the same for all packets regardless of their size, with message segmentation the total amount of header bytes is more.

P34. Skype offers a service that allows you to make a phone call from a PC to an ordinary phone. This means that the voice call must pass through both the Internet and through a telephone network. Discuss how this might be done.

Ans: The circuit-switched telephone networks and the Internet are connected together at "**gateways**". When a Skype user (connected to the Internet) calls an ordinary telephone, a circuit is established between a gateway and the telephone user over the circuit switched network. The skype user's voice is sent in packets over the Internet to the gateway. At the gateway, the voice signal is reconstructed and then sent over the circuit. In the other direction, the voice signal is sent over the circuit switched network to the gateway. The gateway packetizes the voice signal and sends the voice packets to the Skype user.

Chapter 2

R2. What is the difference between network architecture and application architecture?

Answer: Network architecture refers to the organization of the communication process into layers (e.g., the five-layer Internet architecture). Application architecture, on the other hand, is designed by an application developer and dictates the broad structure of the application (e.g., client-server or P2P).

R4. For a P2P file-sharing application, do you agree with the statement, “There is no notion of client and server sides of a communication session”? Why or why not?

Answer: No. In a P2P file-sharing application, the peer that is receiving a file is typically the client and the peer that is sending the file is typically the server.

R6. Suppose you wanted to do a transaction from a remote client to a server as fast as possible. Would you use UDP or TCP? Why?

Answer: You would use UDP. With UDP, the transaction can be completed in one roundtrip time (RTT) - the client sends the transaction request into a UDP socket, and the server sends the reply back to the client's UDP socket. With TCP, a minimum of two RTTs are needed - one to set-up the TCP connection, and another for the client to send the request, and for the server to send back the reply.

R10. What is meant by a handshaking protocol?

Answer: A protocol uses handshaking if the two communicating entities first exchange control packets before sending data to each other. SMTP uses handshaking at the application layer whereas HTTP does not.

R11. Why do HTTP, SMTP, and IMAP run on top of TCP rather than on UDP?

Ans: **TCP provides:**

1. **Reliable delivery:** ensures all bytes are delivered without loss.
2. **Ordered delivery:** preserves the sequence of bytes as sent.
3. **Error detection and retransmission:** automatically handles corrupted or lost segments.
4. **Flow control and congestion control:** prevents overwhelming the receiver or network.

UDP does not provide reliability or ordering, so using UDP would require each application to implement these mechanisms itself — unnecessary complexity for protocols like HTTP, SMTP, and IMAP.

R12. Consider an e-commerce site that wants to keep a purchase record for each of its customers. Describe how this can be done with cookies.

Answer: When the user first visits the site, the server creates a unique identification number, creates an entry in its back-end database, and returns this identification number as a cookie number. This cookie number is stored on the user's host and is managed by the browser. During each subsequent visit (and purchase), the browser sends the cookie number back to the site. Thus the site knows when this user (more precisely, this browser) is visiting the site.

R13. Describe how Web caching can reduce the delay in receiving a requested object. Will Web caching reduce the delay for all objects requested by a user or for only some of the objects? Why?

Ans: **Web caching** stores copies of previously requested web objects (like HTML pages, images, or videos) closer to the user, often in a **proxy cache** or **browser cache**.
When a user requests an object, the cache can **serve it directly** if it has a fresh copy, avoiding the need to fetch it from the original web server.
Will it reduce delay for all objects?
No. Only objects that are **already in the cache** will have reduced delay.
If the object is **not cached** or the cached copy is **stale**, the request still goes to the origin server, and delay is similar to a cache miss.

R19. Is it possible for an organization's Web server and mail server to have exactly the same alias for a hostname (for example, foo.com)? What would be the type for the RR that contains the hostname of the mail server?

Answer: Yes an organization's mail server and Web server can have the same alias for a host name. The MX record is used to map the mail server's host name to its IP address.

R21. In BitTorrent, suppose Alice provides chunks to Bob throughout a 30-second interval. Will Bob necessarily return the favor and provide chunks to Alice in this same interval? Why or why not?

Ans: It is not necessary that Bob will also provide chunks to Alice. Alice has to be in the top 4 neighbors of Bob for Bob to send out chunks to her; this might not occur even if Alice provides chunks to Bob throughout a 30-second interval.

R22. Consider a new peer Alice that joins BitTorrent without possessing any chunks. Without any chunks, she cannot become a top-four uploader for any of the other peers, since she has nothing to upload. How will Alice get her first chunk?

Ans: Recall that in BitTorrent, a peer picks a random peer and optimistically unchokes the peer for a short period of time. Therefore, Alice will eventually be optimistically unchoked by one of her neighbors, during which time she will receive chunks from that neighbor.

R23. What is an overlay network? Does it include routers? What are the edges in the overlay network?

Ans: **Overlay network:** A network built **on top of another network**. Nodes in the overlay are connected by **virtual links**, which correspond to **paths in the underlying physical network**.
The overlay network in a P2P file sharing system consists of the nodes participating in the file sharing system and the logical links between the nodes. There is a logical link (an “edge” in graph theory terms) from node A to node B if there is a semi-permanent TCP connection between A and B.
An overlay network does not include routers.

R24. CDNs typically adopt one of two different server placement philosophies. Name and briefly describe them.

Ans: One server placement philosophy is called **Enter Deep**, which enters deep into the access networks of Internet Service Providers, by deploying server clusters in access ISPs all over the world. The goal is to reduce delays and increase throughput between end users and the CDN servers.
Another philosophy is **Bring Home**, which brings the ISPs home by building large CDN server clusters at a smaller number of sites and typically placing these server clusters in IXPs (Internet Exchange Points). This Bring Home design typically results in lower maintenance and management cost, compared with the enter- deep design philosophy.

R26. In Section 2.7, the UDP server described needed only one socket, whereas the TCP server needed two sockets. Why? If the TCP server were to support n simultaneous connections, each from a different client host, how many sockets would the TCP server need?

Ans: Why UDP server needs only one socket
UDP is **connectionless**.
A single UDP socket can **receive from and send to any client**, because each incoming UDP datagram contains the client's address and port.

Why TCP server needs two sockets

TCP is **connection-oriented**.

The TCP server uses:

Listening socket – waits for incoming connection requests.

Connected socket – created **after accepting a client**, used for actual communication with that client.

TCP server supporting n simultaneous clients

For each new client connection, the server creates **one new connected socket**.

Total sockets required:

1 listening socket+n connected sockets=n+1 sockets.

R27. For the client-server application over TCP described in Section 2.7, why must the server program be executed before the client program? For the client-server application over UDP, why may the client program be executed before the server program?

Ans: For the TCP application, as soon as the client is executed, it attempts to initiate a TCP connection with the server. If the TCP server is not running, then the client will fail to make a connection. For the UDP application, the client does not initiate connections (or attempt to communicate with the UDP server) immediately upon execution

P18. a. What is a whois database?

Ans: a) For a given input of domain name (such as ccn.com), IP address or network administrator name, the whois database can be used to locate the corresponding registrar, whois server, DNS server, and so on.

P20. Suppose you can access the caches in the local DNS servers of your department. Can you propose a way to roughly determine the Web servers (outside your department) that are most popular among the users in your department? Explain.

Ans: We can periodically take a snapshot of the DNS caches in the local DNS servers. The Web server that appears most frequently in the DNS caches is the most popular server. This is because if more users are interested in a Web server, then DNS requests for that server are more frequently sent by users. Thus, that Web server will appear in the DNS caches more frequently.

P21. Suppose that your department has a local DNS server for all computers in the department. You are an ordinary user (i.e., not a network/system administrator). Can you determine if an external Web site was likely accessed from a computer in your department a couple of seconds ago? Explain.

Ans: Yes, we can use dig to query that Web site in the local DNS server. For example, “dig cnn.com” will return the query time for finding cnn.com. If cnn.com was just accessed a couple of seconds ago, an entry for cnn.com is cached in the local DNS cache, so the query time is 0 msec. Otherwise, the query time is large.

24-mid

1.IP is considered as a best-effort protocol. comment on that

Ans: **IP (Internet Protocol)** is called a **best-effort protocol** because it **does not guarantee delivery** of packets.

IP provides **“best-effort” delivery**, meaning it **tries to deliver packets** but makes no promises about success, order, or duplication.

2. How do the local dns servers get the IP addresses of root dns servers?

Ans: Local DNS servers (also called **recursive resolvers**) do not discover root DNS server IPs dynamically.

They are **preconfigured with a list of root server IP addresses** (both IPv4 and IPv6), called the **root hints file**.

When a resolver needs to start a query for an unknown domain, it **contacts one of these root servers** to begin the hierarchical lookup process

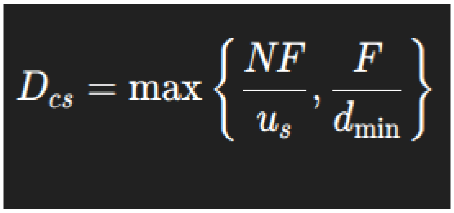
3. Consider distributing a file of 20 Gbps to 12 peers. The server has an upload rate of 10 Mbps, and each peer has a download rate of 2 Mbps, and an upload rate of 1 Mbps. Calculate the minimum distribution time for both client-server and P2P file distribution systems.

Ans: Given**:**

File size: F = 20 Gb
Number of peers: N = 12
Server upload rate: us = 10 Mbps
Each peer download rate: di = 2 Mbps
Each peer upload rate: ui = 1 Mbps

In a client-server system, the **server must send the entire file to each peer**.

Time to send to **all peers** is limited by **server upload rate**:



$$D_{cs} = \max\left\{\frac{NF}{u_s}, \frac{F}{d_{\min}}\right\}$$

Server upload: NF / us = 24000 s

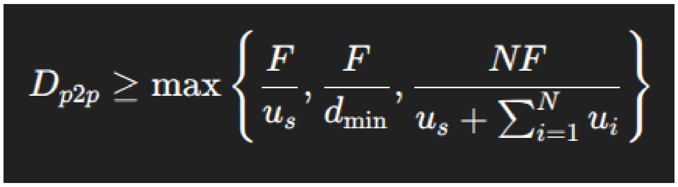
Peer download: F / dmin = 10000s

Dcs = max(24000, 10000) = 24000s

Client-server minimum distribution time: 24,000 s

P2P Distribution

For P2P, the **distribution time formula** is:



$$D_{p2p} \geq \max\left\{\frac{F}{u_s}, \frac{F}{d_{\min}}, \frac{NF}{u_s + \sum_{i=1}^{N} u_i}\right\}$$

Total upload : us + ui(1 - 12 peer) = 22Mbps

Dp2p = 10909 s

P2P minimum distribution time: ~10,909 s

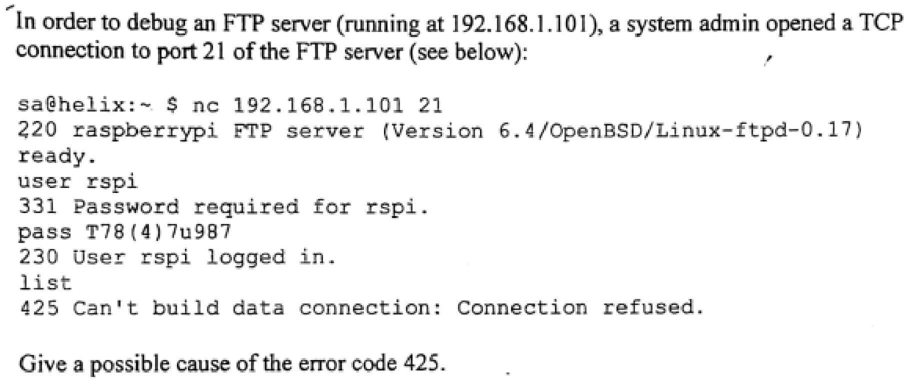
Q2. In order to prevent content piracy, some companies tend to disrupt P2P services by deploying decoys (i.e. misbehaving peers meant to disrupt downloads by other peers). Company A deployed a peer which advertises all chunks and sends fake content to others. On the other hand, company B deployed a peer which advertises all chunks but never sends any (fake or original) to others but repeatedly connects to other peers. Which method is more effective in disrupting P2P services and why?

Ans: **Effectiveness:** Company B’s method is **more effective** in disrupting P2P services.
Reason:

Fake data from Company A can be detected and discarded.

Non-responding peers from Company B **tie up peers’ connections and reduce swarm efficiency**, slowing down legitimate downloads more persistently.

25-mid



In order to debug an FTP server (running at 192.168.1.101), a system admin opened a TCP connection to port 21 of the FTP server (see below):

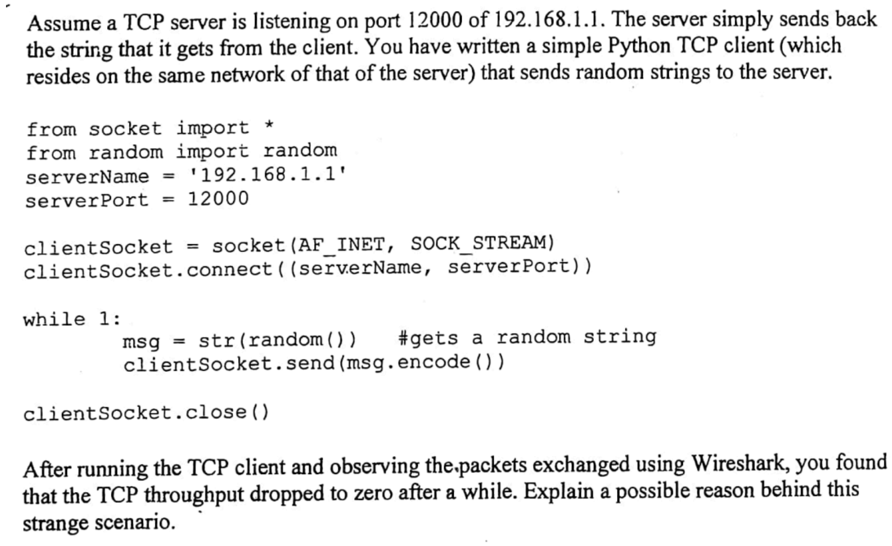
```
sa@helix:~ $ nc 192.168.1.101 21
220 raspberrypi FTP server (Version 6.4/OpenBSD/Linux-ftpd-0.17)
ready.
user rspi
331 Password required for rspi.
pass T78(4)7u987
230 User rspi logged in.
list
425 Can't build data connection: Connection refused.
```

Q1. Give a possible cause of the error code 425.

Ans: The error code 425 ("Can't build data connection: Connection refused") in FTP typically happens when the FTP server cannot establish a data connection back to the client.

Here's the most likely cause in this scenario:

The FTP server is using active mode, and the client's firewall is blocking the incoming data connection from the server.



Assume a TCP server is listening on port 12000 of 192.168.1.1. The server simply sends back the string that it gets from the client. You have written a simple Python TCP client (which resides on the same network of that of the server) that sends random strings to the server.

```
from socket import *
from random import random
serverName = '192.168.1.1'
serverPort = 12000

clientSocket = socket(AF_INET, SOCK_STREAM)
clientSocket.connect((serverName, serverPort))

while 1:
        msg = str(random())   #gets a random string
        clientSocket.send(msg.encode())

clientSocket.close()
```

Q2. After running the TCP client and observing the packets exchanged using Wireshark, you found that the TCP throughput dropped to zero after a while. Explain a possible reason behind this strange scenario.

Ans: The most likely reason the TCP throughput drops to zero is TCP flow control due to the server's receive buffer filling up.

- The server reads data from its receive buffer and writes it back to the client.
- But the client never calls recv(), so the client's receive buffer fills up with the echoed responses.
- Eventually, the client's receive buffer becomes full.
- Now the server cannot send more echoed data to the client.
- Since the server cannot send, it cannot clear its own receive buffer by processing more data from the client.

- This creates a deadlock: both sides have full buffers, and throughput drops to zero.