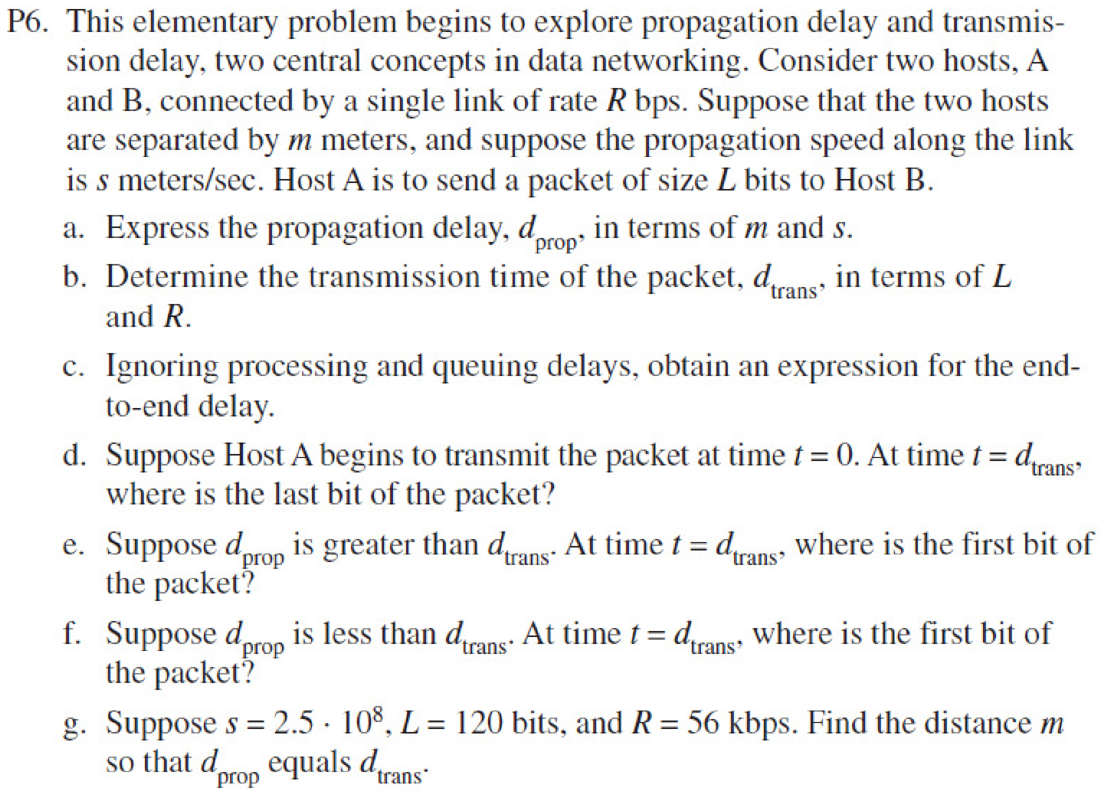
[Solution] CS5222 Assignment 1 (on Chapter 1 and 2)

Chapter 1



P6. This elementary problem begins to explore propagation delay and transmission delay, two central concepts in data networking. 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/sec. Host A is to send a packet of size L bits to Host B.

a. Express the propagation delay, d_{prop}, in terms of m and s.

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

c. Ignoring processing and queuing delays, obtain an expression for the end-to-end delay.

d. Suppose Host A begins to transmit the packet at time $t = 0$. At time $t = d_{\text{trans}}$, where is the last bit of the packet?

e. Suppose d_{prop} is greater than d_{trans}. At time $t = d_{\text{trans}}$, where is the first bit of the packet?

f. Suppose d_{prop} is less than d_{trans}. At time $t = d_{\text{trans}}$, where is the first bit of the packet?

g. Suppose $s = 2.5 \cdot 10^8$, $L = 120$ bits, and $R = 56$ kbps. Find the distance m so that d_{prop} equals d_{trans}.

Problem 6

a) $d_{prop} = m/s$ seconds.

b) $d_{trans} = L/R$ seconds.

c) $d_{end-to-end} = (m/s + L/R)$ seconds.

d) The bit is just leaving Host A.

e) The first bit is in the link and has not reached Host B.

f) The first bit has reached Host B.

g) Want

$$m = \frac{L}{R}s = \frac{120}{56\times10^3}\left(2.5\times10^8\right) = 536\text{ km}.$$

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 $8 \cdot 10^6$ bits long that is to be sent from source to destination in Figure 1.27. Suppose each link in the figure is 2 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 800 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.

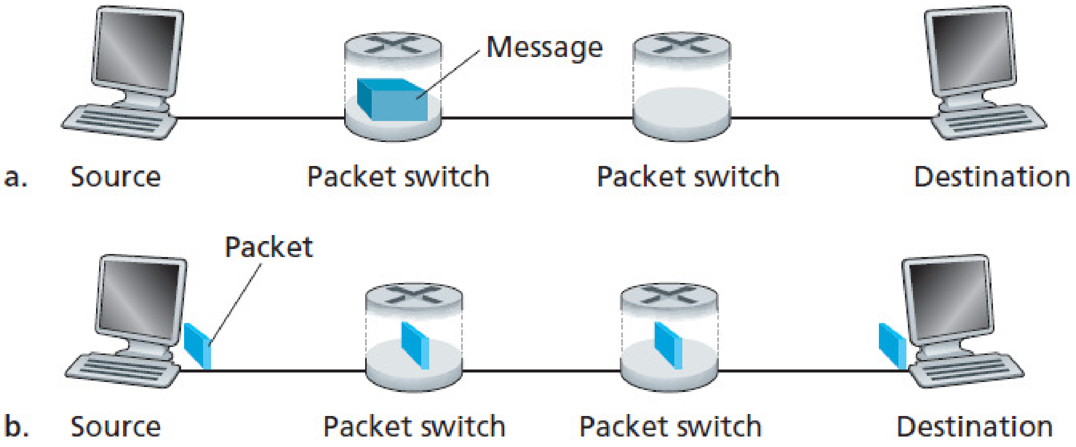


Figure 1.27 ♦ End-to-end message transport: (a) without message segmentation; (b) with message segmentation

Problem 31

a) Time to send message from source host to first packet switch = $\frac{8\times10^6}{2\times10^6}\sec = 4\sec$ With store-and-forward switching, the total time to move message from source host to destination host = $4\sec\times 3\ hops = 12\sec$

b) Time to send 1st packet from source host to first packet switch = . $\frac{1\times10^4}{2\times10^6}\sec = 5\ m\sec$. Time at which 2nd packet is received at the first switch = time at which 1st packet is received at the second switch = $2\times 5m\sec = 10\ m\sec$

c) Time at which 1st packet is received at the destination host = $5\ m\sec\times 3\ hops = 15\ m\sec$. After this, every 5msec one packet will be received; thus time at which last (800th) packet is received = $15\ m\sec + 799 * 5m\sec = 4.01\sec$. It can be seen that delay in using message segmentation is significantly less (almost 1/3rd).

d)

i. 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).

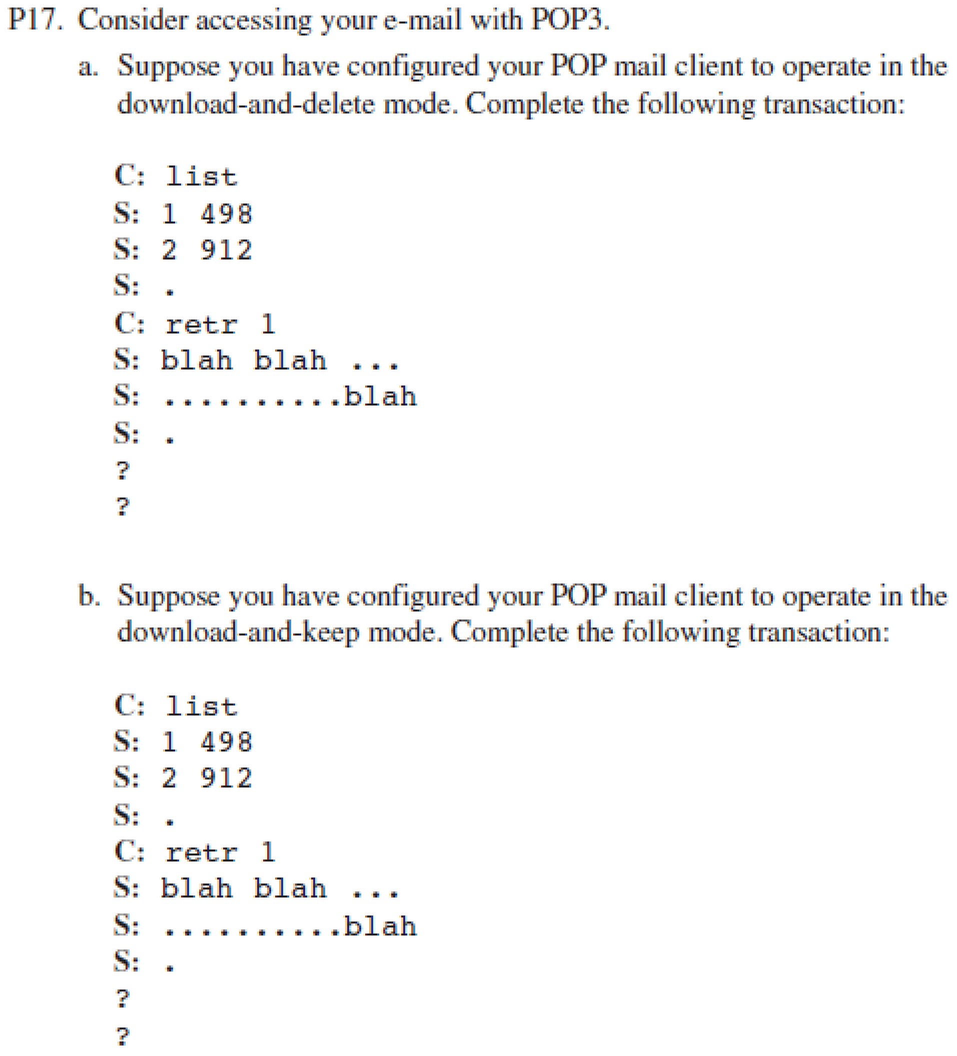
ii. Without message segmentation, huge packets (containing HD videos, for example) are sent into the network. Routers have to accommodate these huge packets. Smaller packets have to queue behind enormous packets and suffer unfair delays.

e)

i. Packets have to be put in sequence at the destination.

ii. 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.

Chapter 2



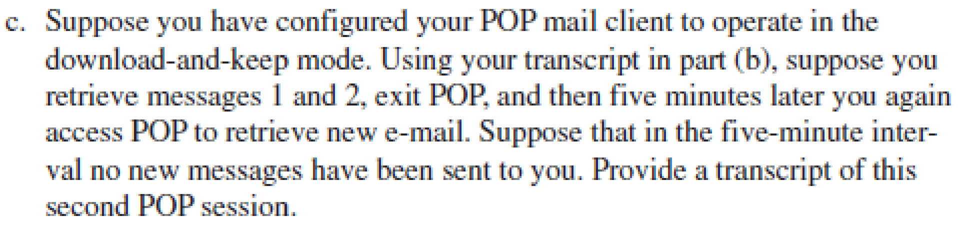
P17. Consider accessing your e-mail with POP3.

a. Suppose you have configured your POP mail client to operate in the download-and-delete mode. Complete the following transaction:

```
C: list
S: 1 498
S: 2 912
S: .
C: retr 1
S: blah blah ...
S: ..........blah
S: .
?
?
```

b. Suppose you have configured your POP mail client to operate in the download-and-keep mode. Complete the following transaction:

```
C: list
S: 1 498
S: 2 912
S: .
C: retr 1
S: blah blah ...
S: ..........blah
S: .
?
?
```



c. Suppose you have configured your POP mail client to operate in the download-and-keep mode. Using your transcript in part (b), suppose you retrieve messages 1 and 2, exit POP, and then five minutes later you again access POP to retrieve new e-mail. Suppose that in the five-minute interval no new messages have been sent to you. Provide a transcript of this second POP session.

Problem 17

```
a)  C: dele 1
     C: retr 2
     S: (blah blah …
     S: ………..blah)
     S: .
     C: dele 2
     C: quit
     S: +OK POP3 server signing off
```

```
b)  C: retr 2
     S: blah blah …
     S: ………..blah
     S: .
     C: quit
     S: +OK POP3 server signing off
```

```
b)  C: list
    S: 1 498
    S: 2 912
    S: .
    C: retr 1
    S: blah …..
    S: ….blah
    S: .
      C: retr 2
     S: blah blah …
     S: ………..blah
     S: .
     C: quit
     S: +OK POP3 server signing off
```

Question. Zoom (https://zoom.us/) is a videoconferencing application. After doing some research about its communication protocols, answer the following questions:

a) What kind of application-layer protocols do Zoom use? Please state three of them and briefly discuss their functions.

b) Is Zoom primarily based on a client-server or P2P architecture? Why?

c) Which transport layer protocol is Zoom primarily using? Why?

d) During our CS5222 lecture using Zoom, does a student's computer need to obtain the IP address of the lecturer's computer? If so, how? If not, why?

Answer:

Referenced literature is listed at the end of each answer, if any.

a) Primary application-layer protocols:

1) Zoom Proprietary Protocols for session initiation, control, and management. These protocols handle tasks such as user authentication, meeting setup, participant management, and feature controls (like screen sharing and recording) [1].

2) Real-Time Transport Protocol (RTP) and Secure RTP (SRTP): For the transmission of audio and video data, Zoom employs RTP, which is designed for end-to-end, real-time transfer of streaming media. To ensure security, Zoom uses SRTP, which adds encryption and message authentication to RTP, protecting the media streams against eavesdropping and tampering [2].

3) HTTPS/HTTP for various auxiliary functions such as accessing web services, downloading updates, and retrieving resources [3, 4].

4) WebRTC for browser-based connections [5].

5) Transport Layer Security (TLS) for encrypting communications between clients and servers [6].

b) Zoom primarily operates on a **client-server architecture** [7,8]. Here's why:

1) Centralized Servers: In Zoom's architecture, all participants in a meeting connect to Zoom's cloud servers. These servers handle the distribution of audio, video, and data streams between participants.

2) Scalability and Performance: Using centralized servers allows Zoom to optimize bandwidth usage and performance, especially in meetings with many participants. The servers can mix, process, and distribute streams efficiently.

3) Network Traversal: Client-server architecture simplifies connectivity across different networks and through firewalls or NAT devices, as clients establish outbound connections to well-known server addresses.

4) Security and Compliance: Centralized control over data flow enables Zoom

to implement robust security measures, such as encryption and compliance with various regulations.

While Zoom may employ some peer-to-peer (P2P) techniques in specific scenarios (like two-party calls to reduce latency), the predominant architecture is client-server to ensure reliability, scalability, and security.

c) Zoom primarily uses UDP (User Datagram Protocol) at the transport layer for real-time audio and video transmission [3]. UDP is preferred because:

1) Lower latency: UDP doesn't require connection establishment or error-checking, reducing delays.
2) Tolerates packet loss: In real-time communication, it's better to skip a damaged packet than wait for retransmission.
3) Suitable for real-time data: Audio/video streams can tolerate some data loss without significant quality degradation.

TCP is used for signaling, chat, and other non-real-time data where reliability is more important than speed.

d) No, a student's computer does not need to obtain the IP address of the lecturer's computer [7,8]. Here's why:

1) Communication Through Zoom Servers: In Zoom's client-server model, all participants connect to Zoom's cloud infrastructure rather than directly to each other. Audio, video, and data streams are managed and relayed by Zoom's servers.
2) Abstracted Connections: The application handles all connection details internally. Participants join meetings using meeting IDs and passwords without needing to know other participants' network information.
3) Security and Privacy: Not sharing IP addresses between clients enhances privacy and security, reducing the risk of direct attacks or unauthorized access.

How It Works:

1) Connection Establishment: When the lecturer and students join the Zoom meeting, each client establishes a secure connection to the Zoom servers over the Internet.
2) Data Transmission: The lecturer's audio and video streams are sent to the Zoom servers, which then distribute the streams to all participants.
3) Signaling and Control: All signaling (like joining/leaving the meeting and enabling/disabling features) is also managed through the servers.
4) Therefore, the student's computer relies on Zoom's infrastructure to receive the lecturer's audio and video without needing to know or obtain the lecturer's IP address.

[1]. https://intuji.com/how-does-zoom-work-video-conferencing-tech

[2]. https://explore.zoom.us/docs/doc/Zoom%20Encryption%20Whitepaper.pdf

[3]. https://support.zoom.com/hc/en/article?id=zm_kb&sysparm_article=KB0060548

[4]. https://explore.zoom.us/docs/doc/Zoom%20Connection%20Process%20Whitepaper.pdf

[5]. https://www.daily.co/blog/zoom-web-sdk-technical-notes/

[6]. https://support.zoom.com/hc/en/article?id=zm_kb&sysparm_article=KB0069186

[7]. https://andypickup.com/how-does-zoom-work-5964225e3708

[8]. https://www.zoom.com/en/blog/cloud-based-and-peer-peer-meetings/