Questions not already in other documents

Mid Sem

3. b.

```
Consider a short, 10-meter link, over which a sender can transmit at a rate of 150
bits/sec
in both directions. Suppose that packets containing data are 100,000 bits long,
and packets
containing only control (e.g., ACK or handshaking) are 200 bits long. Assume that
N parallel
connections each get 1/N of the link bandwidth. Now consider the HTTP protocol,
and suppose
that each downloaded object is 100 Kbits long, and that the initial downloaded
object contains
10 referenced objects from the same sender. Would parallel downloads via parallel
instances
of non-persistent HTTP make sense in this case? Now consider persistent HTTP. Do
you expect
significant gains over the  non-persistent case? Justify and explain your answer.
```

Kurose UNIT 2 P 10

```
Note that each downloaded object can be completely put into one data packet. Let
Tp
denote the one-way propagation delay between the client and the server.
First consider parallel downloads using non-persistent connections. Parallel
downloads
would allow 10 connections to share the 150 bits/sec bandwidth, giving each just
15
bits/sec. Thus, the total time needed to receive all objects is given by:
(200/150+Tp + 200/150 +Tp + 200/150+Tp + 100,000/150+ Tp )
+ (200/(150/10)+Tp + 200/(150/10) +Tp + 200/(150/10)+Tp + 100,000/(150/10)+ Tp )
= 7377 + 8*Tp (seconds)
Now consider a persistent HTTP connection. The total time needed is given by:
(200/150+Tp + 200/150 +Tp + 200/150+Tp + 100,000/150+ Tp )
+ 10*(200/150+Tp + 100,000/150+ Tp )
=7351 + 24*Tp (seconds)
Assuming the speed of light is 300*106 m/sec, then Tp=10/(300*106)=0.03 microsec.
Tp
is therefore negligible compared with transmission delay.
Thus, we see that persistent HTTP is not significantly faster (less than 1
percent) than the
non-persistent case with parallel download.
```

4. a

```
Can a machine with a single DNS name have multiple IP" addresses ?
How could this occur ?
DNS uses UDP instead of TCP. Ifa DNS packet is lost, there is no automatic
recovery. Does this cause a problem, and ir
so, how is it solved ?
Can a computer have two DNS names that fall in
different top-level domains ? If so, give a plausible example. If not,
explain why not ?
```

```
Yes. In fact, in Fig. 7-3 we see an example of a duplicate IP address.
Remember that an IP address consists of a network number and a host
number. If a machine has two Ethernet cards, it can be on two separate net-
works, and if so, it needs two IP addresses.

DNS is idempotent. Operations can be repeated without harm. When a pro-
cess makes a DNS request, it starts a timer. If the timer expires, it just makes
the request again. No harm is done.

It is possible. www.large-bank.com and www.large-bank.ny.us could have
the same IP address. Thus, an entry under com and under one of the country
domains is certainly possible (and common).
```

5. a.

```
It is desired to send a sequence of computer screen images over an optical fiber.
The screen is 480 x
640 pixels, each pixel being 24 bits. There are 60 screen images per second. How
much bandwidth is
needed?
```

```
The data rate is 480 × 640 × 24 × 60 bps, which is 442 Mbps.
```

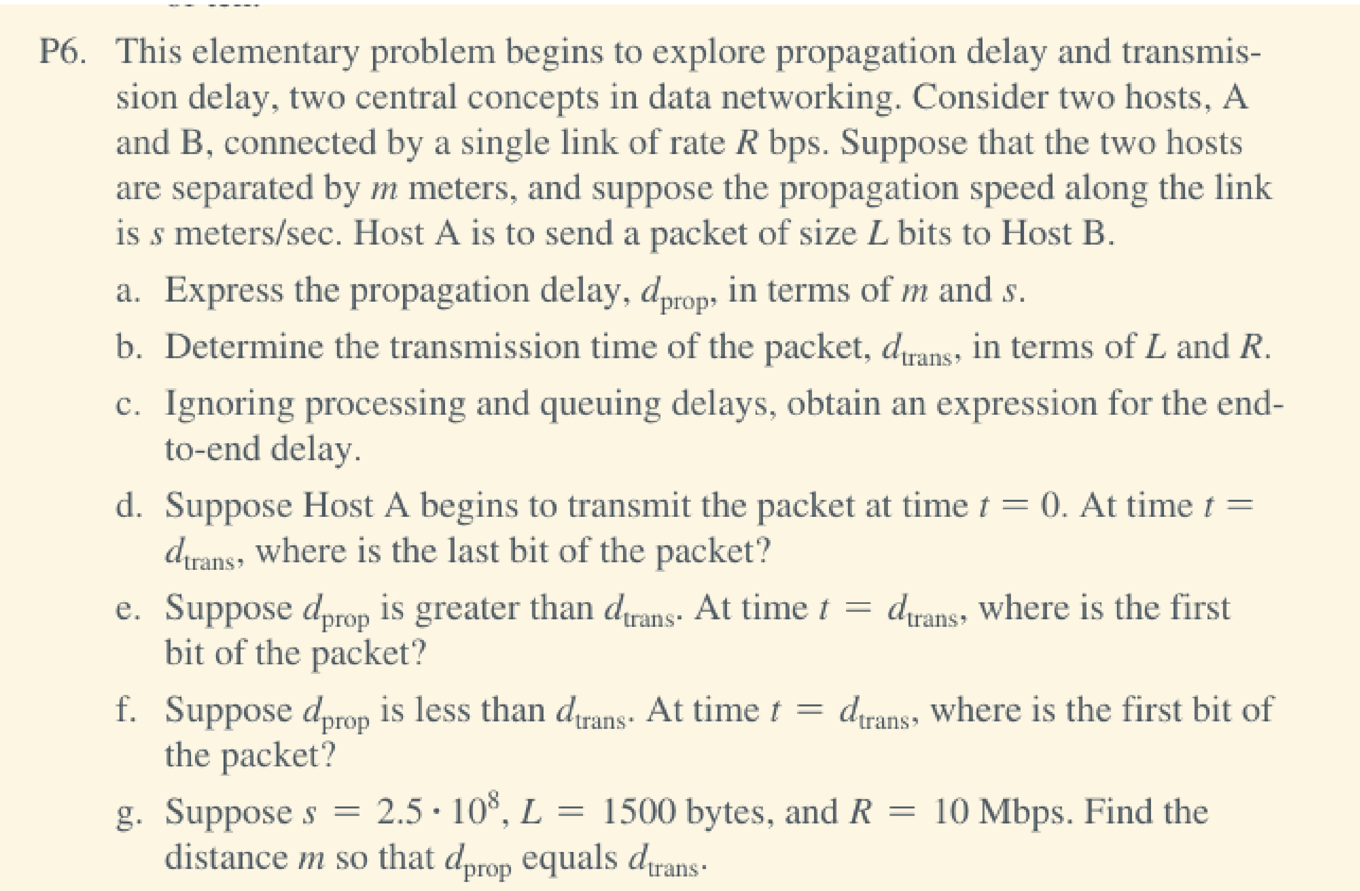
5. b.

```
Suppose Host A sends two TCP segments back to back to Host B over a
TCP connection. The first segment has sequence number 90; the second has
sequence number 110.
a. How much data is in the first segment?
b. Suppose that the first segment is lost but the second segment arrives at B. In
the acknowledgment that
Host B sends to Host A, what will be the acknowledgment number?
```

```
a) 20 bytes
b) ack number = 90
```

End Sem

2. a



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.

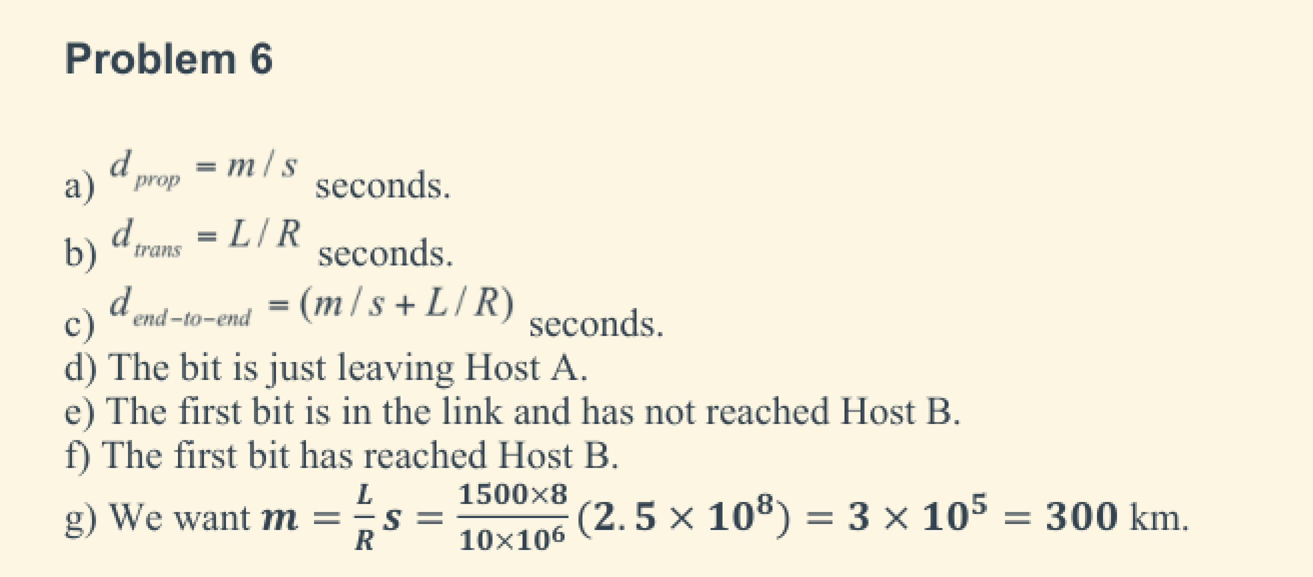
c. Ignoring processing and queuing delays, obtain an expression for the end-to-end delay.

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 $s = 2.5 \cdot 10^8$, $L = 1500$ bytes, and $R = 10$ Mbps. 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.

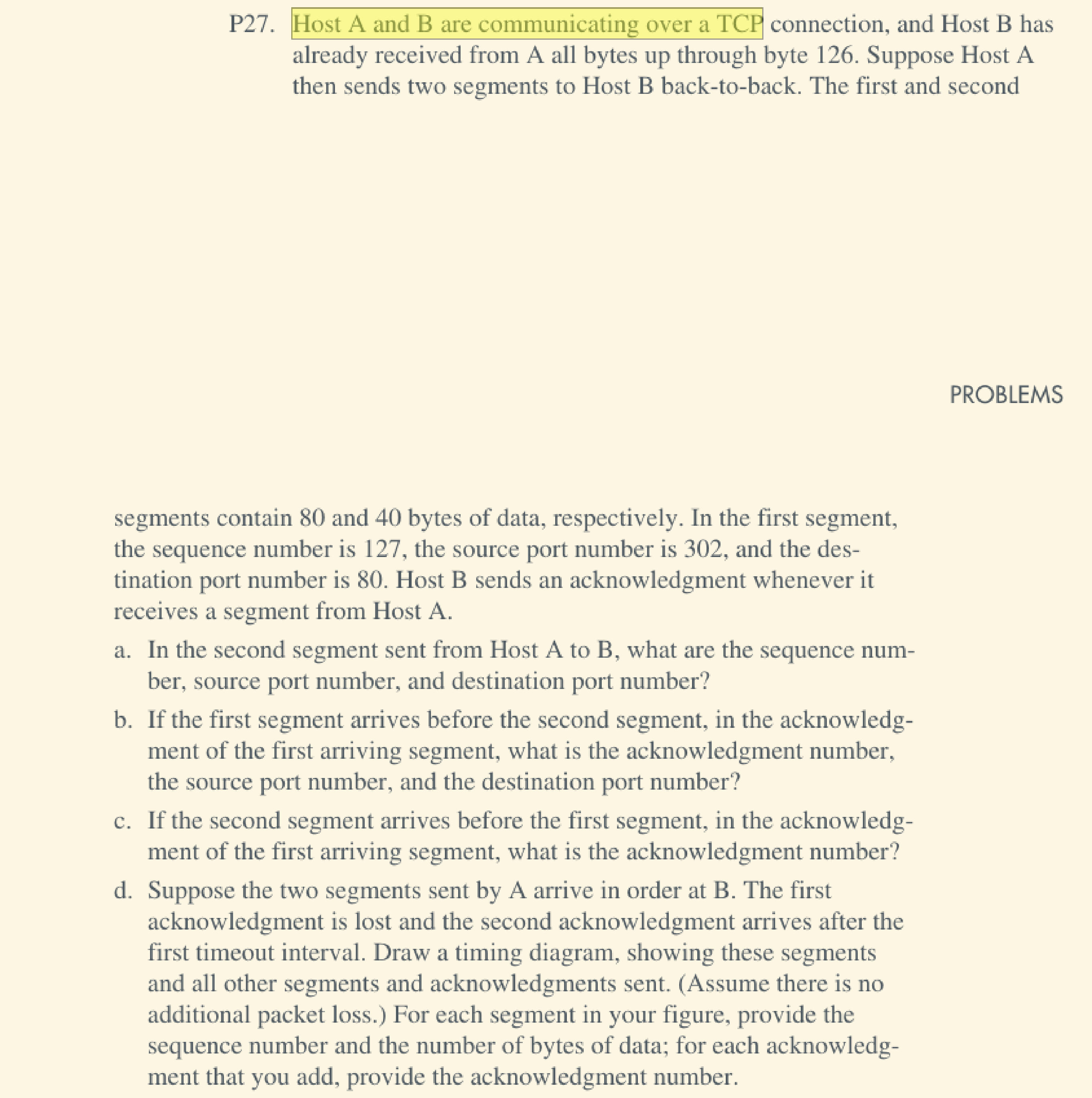
d) The bit is just leaving Host A.

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

f) The first bit has reached Host B.

g) We want $\boldsymbol{m} = \frac{\boldsymbol{L}}{\boldsymbol{R}}\boldsymbol{s} = \frac{\mathbf{1500\times8}}{\mathbf{10\times10^6}}(\mathbf{2.5}\times\mathbf{10^8}) = \mathbf{3}\times\mathbf{10^5} = \mathbf{300}$ km.

2. b



P27. Host A and B are communicating over a TCP connection, and Host B has already received from A all bytes up through byte 126. Suppose Host A then sends two segments to Host B back-to-back. The first and second

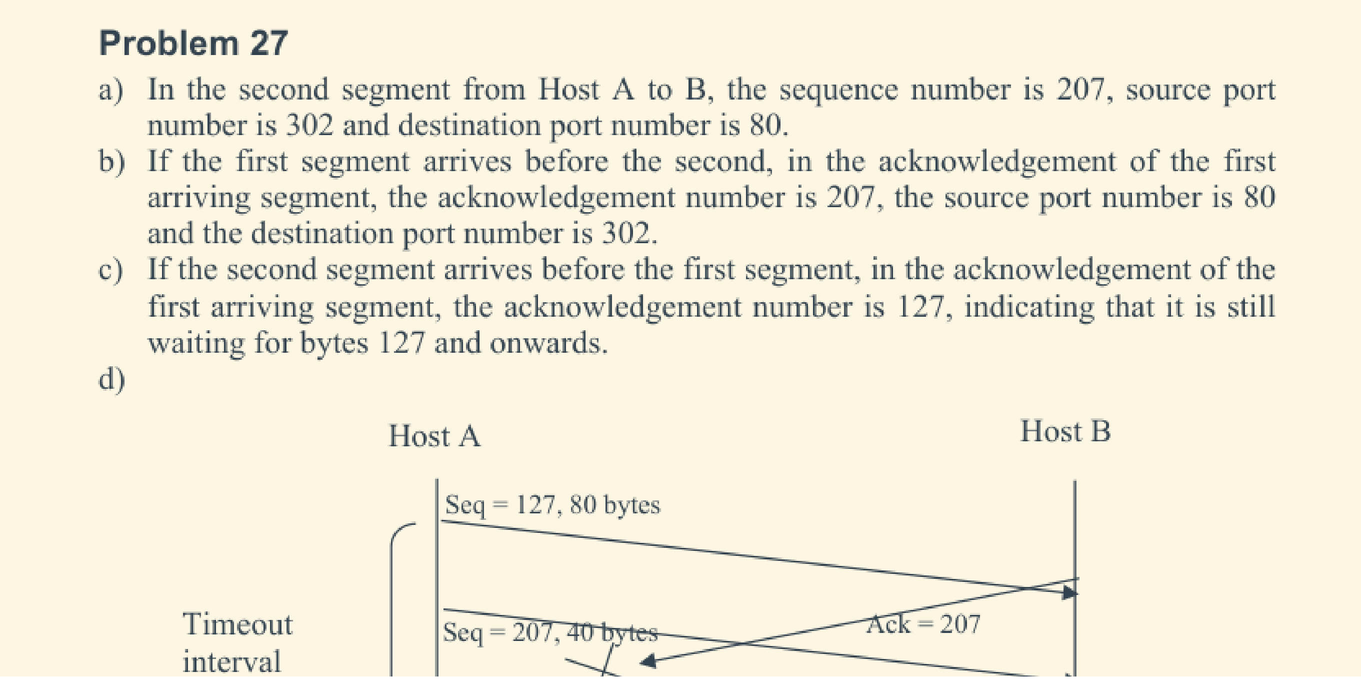
segments contain 80 and 40 bytes of data, respectively. In the first segment, the sequence number is 127, the source port number is 302, and the destination port number is 80. Host B sends an acknowledgment whenever it receives a segment from Host A.

a. In the second segment sent from Host A to B, what are the sequence number, source port number, and destination port number?

b. If the first segment arrives before the second segment, in the acknowledgment of the first arriving segment, what is the acknowledgment number, the source port number, and the destination port number?

c. If the second segment arrives before the first segment, in the acknowledgment of the first arriving segment, what is the acknowledgment number?

d. Suppose the two segments sent by A arrive in order at B. The first acknowledgment is lost and the second acknowledgment arrives after the first timeout interval. Draw a timing diagram, showing these segments and all other segments and acknowledgments sent. (Assume there is no additional packet loss.) For each segment in your figure, provide the sequence number and the number of bytes of data; for each acknowledgment that you add, provide the acknowledgment number.



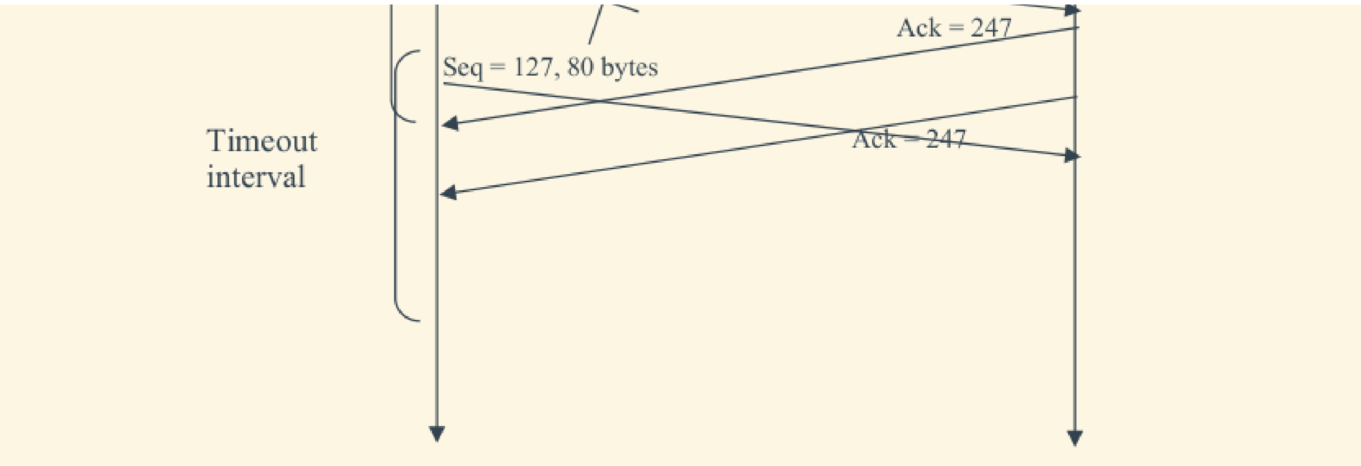
Problem 27

a) In the second segment from Host A to B, the sequence number is 207, source port number is 302 and destination port number is 80.

b) If the first segment arrives before the second, in the acknowledgement of the first arriving segment, the acknowledgement number is 207, the source port number is 80 and the destination port number is 302.

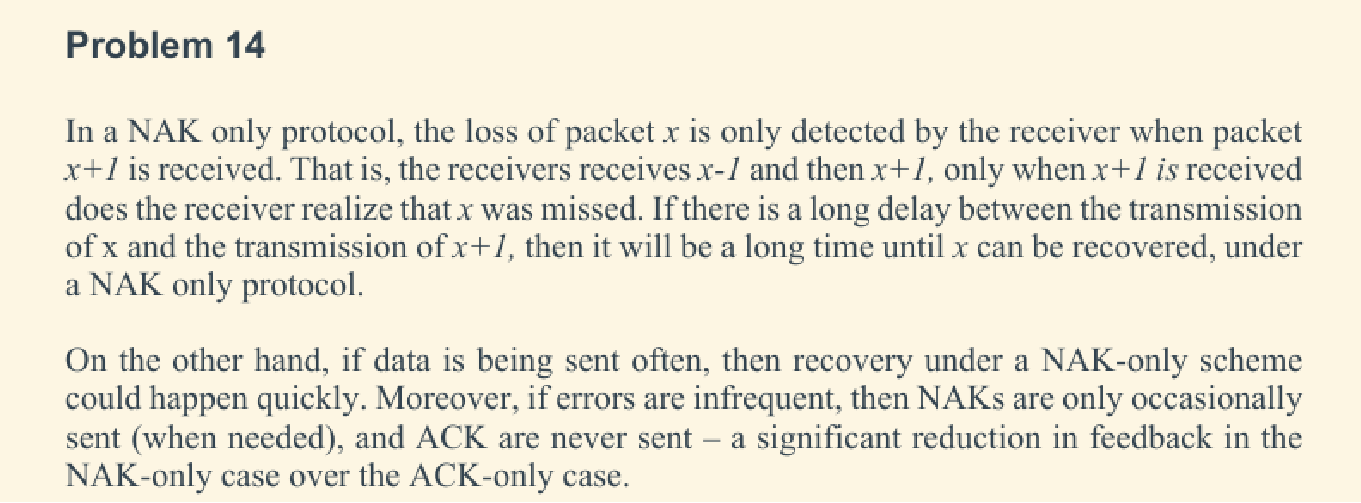
c) If the second segment arrives before the first segment, in the acknowledgement of the first arriving segment, the acknowledgement number is 127, indicating that it is still waiting for bytes 127 and onwards.

d)



3. a. i

```
Consider a reliable data transfer protocol that uses only negative acknowledg-
ments. Suppose the sender sends data only infrequently. Would a NAK-only
protocol be preferable to a protocol that uses ACKs? Why? Now suppose the
sender has a lot of data to send and the end-to-end connection experiences
few losses. In this second case, would a NAK-only protocol be preferable to
a protocol that uses ACKs? Why?
```



Problem 14

In a NAK only protocol, the loss of packet *x* is only detected by the receiver when packet *x+1* is received. That is, the receivers receives *x-1* and then *x+1,* only when *x+1 is* received does the receiver realize that *x* was missed. If there is a long delay between the transmission of x and the transmission of *x+1,* then it will be a long time until *x* can be recovered, under a NAK only protocol.

On the other hand, if data is being sent often, then recovery under a NAK-only scheme could happen quickly. Moreover, if errors are infrequent, then NAKs are only occasionally sent (when needed), and ACK are never sent – a significant reduction in feedback in the NAK-only case over the ACK-only case.