Q1:

The file size must be converted from megabytes to kilobits in order to determine the download time.

a. Download time over a 32 kilobit/second modem:

• File size: 3 megabytes = 3,145,728 bytes = 25,165,824 bits = 25,165.824 kilobits

• Download speed: 32 kilobits/second

• Download time = File size / Download speed = 25,165.824 kilobits / 32 kilobits/second = 786.432 seconds

b. For a over 1 megabit/second modem:

• File size: 3 megabytes = 3,145,728 bytes = 25,165,824 bits = 25.165824 megabits

• Download speed: 1 megabit/second

• Download time = File size / Download speed = 25.165824 megabits / 1 megabit/second = 25.165824 seconds

c.

If we assume a maximum compression ratio of 1:6, then we have the following times for the 32 kilobit and 1 megabit lines respectively:

Subsequently, three megabytes will be reduced to one sixth megabyte

3 megabytes = 24 megabits

+ For the 32 kilobit/second modem:

T32K = 25,165.824 kilobits / (6\*32) kilobits/second ≈ 13,11 secconds.

+ For the 1 megabit/second modem:

T1M = 25.165824 megabits / (6 \* 1) megabits ≈ 4,2 seconds.

Q2:

When an application layer entity sends a message of L bytes to its peer process via an existing TCP connection, the message is encapsulated in a TCP segment with an additional 20 bytes of header. This TCP segment is then encapsulated into an IP packet, adding another 20 bytes of header. Finally, the IP packet is placed inside an Ethernet frame, which adds 18 bytes of header and trailer.

So, the total size of the transmitted message becomes L + 20 (TCP header) + 20 (IP header) + 18 (Ethernet header and trailer) = L + 58 bytes.

The proportion of the transmitted bits at the physical layer that represent the actual message information can be calculated as p = [L/(L+58)]\*100.

For different values of L, we get:

• For L = 75 bytes, approximately 56.38% of the transmitted bits represent the message information.

• For L = 350 bytes, approximately 85.77% of the transmitted bits represent the message information.

• For L = 1150 bytes, approximately 95.2% of the transmitted bits represent the message information.

Q3:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| B1 | | | | |
|  |  | Address | Port | Explaination |
| Step 1 | S1 => S5 | S1 | 1 | B1 receives a packet from S1 because S1 initiates the transmission. |
| Step 2 | S3 => S2 | S3 | 2 | B1 receives a packet from S3 as S3 broadcasts a packet. |
| Step 3 | S4 => S3 | S4 | 2 | B1 receives a packet from S4 because B2, knowing that S3 is on port 2, forwards the packet to port 2. |
| Step 4 | S2 => S1 | S2 | 1 | B1 receives a packet from S2 as S2 broadcasts a packet. |
| Step 5 | S5 => S4 |  |  | B1 does not receive a packet from S5 because B2, knowing that S4 is on network 3, does not forward the packet to networks 2 and 1. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| B2 | | | | |
|  |  | Address | Port | Explaination |
| Step 1 | S1 => S5 | S1 | 1 | B1 and B2 do not know the location of S5 initially, so the packet from S1 is broadcasted. B1 forwards the packet, allowing B2 to receive it and learn that S1 is on port 1. |
| Step 2 | S3 => S2 | S3 | 1 | S3 broadcasts a packet to S2. B2 receives this packet and learns that S3 is on port 1. |
| Step 3 | S4 => S3 | S4 | 2 | B2 receives a packet from S4 because S4 sends a packet to S3. |
| Step 4 | S2 => S1 |  |  | B1, knowing that S1 is on network 1, does not forward the packet from S2. Hence, B2 does not receive a packet from S2. |
| Step 5 | S5 => S4 | S5 | 2 | B2 receives a broadcast packet from S5. |

Q4:

a. There are two things needed to be changes:

First, the frame header needs to be modified to accommodate the list of frames to receive. It can be a fixed or a variable number of slots. NAK won’t be necessary because the receiver explicitly indicates which frames need to be transmitted. Second, the transmitter operation must change to retransmit frames according to the received list. If the list contains the m oldest frames that are yet to be received, then the list can be used to skip retransmissions of frames that have already been received.

b. The protocol’s performance is likely to improve, particularly in situations with high error rates or significant delays. This is because a single frame can request the retransmission of multiple frames. However, this comes at the cost of increased overhead in the header and greater protocol complexity compared to the standard Selective-Repeat ARQ.

Q5:

G = 10011

D = 1001010101

* step 1: Append four 0’s to the end of D to get a new D: 10010101010000.
* step 2: Now, we perform binary division of the new D by G.

10010101010000 10011

⊕ 1000110000

10011

-------

000011010

⊕

10011

-------

010011

⊕

10011

-------

000000000 => After the division, we find that the remainder is: 0000

* So, the value of R in this case is 0000.