Kent Mark

Cpre 489 – Homework 3

10/5/2021

Homework 3

Problem 23

* 1. A distribution scheme that could be used is a situation where the server the file to each client at a rate of . This rate is lower than each of the client’s download rates, since . Therefore, each client can receive at a rate of which would mean that the time for each client to receive the entire file is . Since all the clients receive the file in the overall distribution time is also .
  2. For this scenario consider a distribution scheme where the server sends the file to each, in parallel, at a rate of . This rate is less than the server’s link rate since . Also, since each client receives at a rate of the time for each client to receive the entire file is . All the clients receive the file in this time, so the overall distribution time is .
  3. From chapter 2 we are given

(Eq. 1)

Now let’s assume that . Then using Eq.1 we have and then combining our results from (a) we have (Eq. 2)

We can also show that

Once we combine Eq. 2 and Eq. 3 we get the desired result

Problem 24

* 1. We first define . We also assume (Eq. 1) We then divide the file into N parts with the part having size . The server with transmit the part to peer i at rate . Note that so that the aggregate server rate does not exceed the link rate of the server. Also have each peer i forward the bits it receives to each of the peer at rate We also know that the aggregate forwarding rate by peer i is . We havethe last inequality follows from Eq. 1. Therefore the aggregate forwarding rate of peer i is less than its link rate . In this distribution scheme, peer i receives bits at an aggregate rate of Thus each peer receives the file in
  2. We define . We also assume . Let . In this distribution scheme, the file is broken into N + 1 parts. The server sends bits from the part to the peer at a rate . Each peer i forwards the bits arriving at rate to each of the other peers. Additionally, the server sends bits from the part at rate to each of the N peers. The peers do not forward the bits from the part. The aggregate send rate of the server is this means the server’s send rate does not exceed its link rate. The aggregate send rate of peer i is . Thus, each peer’s send rate does not exceed its link rate. So in this distribution scheme, peer i receives bits at an aggregate rate of this means each peer receives the file in .
  3. We know from Chapter 2 we can use so we just combine this with what we got from (a) to (b) to get the desired result.

Problem 3

01010011

+ 01100110

10111001

10111001

+ 01110100

00101110

One’s Complement = 11010001

To detect errors the receiver adds four words (the three original words and the checksum). If the sum contains a zero the receiver knows there has been an error. All one-bit errors are detected, but two-bit errors can be undetected, for example if the last digit of the first word is converted to 0 and the last digit of the second word is converted to a 1.

Problem 16

Yes, it would. This causes the sender to send a number of pipelined data into the channel. There are problems to this approach because if data segments are lost in the channel, then the sender of rdt 3.0 won’t re-send those segments, unless there is some additional mechanism in the application to recover from loss.

Problem 20

Chart, diagram, schematic

Description automatically generated

Problem 21

A will need to be able to timeout and retransmit request messages to recover from any loss that occurs, A could also set copies of messages, so to detect duplicate request messages the protocol will have to use sequence numbers. Since this is a stop and wait protocol 1 bit sequence numbers will be used. The one doing the requesting, A, will subsequently have four states:

**“Wait for request 0 from above” –** A will wait for a call from above to request a unit of data, and upon receiving that request it will send message R0 to B, start a timer and transition to the next state which is “Wait for D0”. It should be noted that while in this current state A ignores anything it receives from B.

**“Wait for D0”** – Here A waits for a D0 data message from B. The timer will continuously run in this state, and if it expires A sends another R0 message, restarts the timer, and waits in this state. However, if a D0 message is received from B, A stops the timer and moves onto the next state which is “Wait for Request 1 from above” state and ignores anything from B.

**“Wait for Request 1 from above”** – Here A waits for a call from above to request a unit of data. When it receives a request from above it sends request message R1 to B, starts the timer and transitions to the “Wait for D1” state. While in this current state A ignores anything it receives from B.

**“Wait for D1” –** Here A waits for a D1 message from B. A timer always runs in this state and if it expires A sends another R1 message, restarts the timer, and waits in this state. If a D1 message is received from B, A stops the timer and moves to the “Wait for request 0 from above” state. Any message from B received in this current state is ignored.

The data supplier B only has two states:

**“Send D0” –** In this state, B continues to respond to received R0 messages by sending D0 and then staying in this state. If B receives a R1 message then it knows its D0 message has been correctly received. It will then get rid of this D0 data and moves to the “Send D1” state where it will use D1 to send the next requested piece of data.

**“Send D1”** – In this state B responds to received R1 messages by sending D1 and then staying in this state. If B receives a R1 message then it knows its D1 message has been received correctly and therefore moves to the “Send D1” state.