

CSE 589 FALL 2017
PROGRAMMING ASSIGNMENT 2
IMPLEMENTING RELIABLE TRANSPORT PROTOCOLS
ANALYSIS REPORT

Student Name: VIKRANT

UB ID #: 50247689

ACADEMIC INTEGRITY STATEMENT 3

TIMEOUT SCHEME 3

MULTIPLE TIMERS IN SELECTIVE REPEAT USING SINGLE HARDWARE TIMER 4

EXPERIMENT 5

REFERENCE 5

ACADEMIC INTEGRITY STATEMENT

“I have read and understood the course academic integrity policy located under this link:

http://www.cse.buffalo.edu/faculty/dimitrio/courses/cse4589_fl4/index.html#integrity”.

TIMEOUT SCHEME

I have chosen timeouts 20.0, 12.0 and 20.0 for ABT, GBN and SR respectively.

TIMEOUT FOR ABT

The time out has been selected as 20, since it takes 5 logical units to travel from A to B and RTT is 10. I observed in my tests (sanity, advanced and basic) that a reduced TIMEOUT value below 20 was leading to errors and test failure due to early timeouts thereby indicating that the RTT and possibly some other factors in tests were leading to failure of the packet delivery in proper sequence to the Application layer of B.

TIMEOUT FOR GBN

The timeout has been selected 12, as below this timeout I found that packet loss was happening and my tests were failing.

TIMEOUT FOR SR

The timeout for SR has been chosen as 20, since at lower timeouts the tests were taking too long time, possibly due to large retransmission of packets. At timeout 20 the tests were completing in comfortable time.

MULTIPLE TIMERS IN SELECTIVE REPEAT USING SINGLE HARDWARE TIMER

PROBLEM:

In Selective Repeat protocol, how to use the single hardware timer to emulate the task of one individual timer per packet.

SOLUTION:

The approach I have taken to tackle this problem is to basically use a data structure that contains each packet (payload, sequence number, acknowledgment number, checksum), its sent and acknowledgement status and timer to calculate the time stamp for each packet while being resent.

In this implementation the flow correctly record when a timer interrupt occurs and it has occurred for which packet.

The implementation ensures that every time a packet is sent or when an ACK is received the time stamp is calculated and proper timer start and stop is managed.

I have implemented the timer based on discussions with TAs, Friends.

Following is in detail the approach I have taken to manage this.

- **Sbuff [2000]** is a struct buffer type variable that stores the packet, its sent and acknowledged status and the time stamp when it's sent.
- **sbuff[packet.acknum].time** is a float variable that captures the time stamp of each packet sent. The counter variable ensures that the time stamp is related to a particular packet sequence number.
- The time stored in the time variable is equivalent to the current simulator time plus the defined TIMEOUT Value. When a timer interrupt occurs, **currtime** variable check the current simulator time.
- A loop then checks that if the packet has been sent (indicated by **sbuff[i].sent==1** has been received and that the current time is bigger than the **sbuff[packet.acknum].time** value (indicating the timer interrupt has occurred after the **TIMEOUT** has happened)

- The `A_timerinterrupt` then resends the packets to B

EXPERIMENT

- Experiments not performed due to paucity of time.

REFERENCE

- <https://stackoverflow.com/questions/30420971/timer-in-selective-repeat-arq>
- <https://www.youtube.com/watch?v=5-5XTYSR7-s>
- https://www.youtube.com/watch?v=lOWzuG_CGog
- https://github.com/darshangm92/Reliable-Transport-Protocols/blob/master/cse589_Project_2/darshang/src/sr.c