EE122–Spring 2013 — Project 3 Report

Chaoran Yu & Zhiling Huang, 22903110 & 22607779

May 4, 2013

Please see the end of the document for all plots.

Task 1: Implement ARQ and Random Delay

1.1 The error control mechanism works as follows: sender $S2$ and receiver $D2$ both keep sliding windows of the same size. The window size is a tunable parameter.
Each time $D2$ receives a packet, it keeps the packet as long as it has not been received before so that $D2$'s sliding window can move faster, enhancing efficiency. If the sequence number of the received packet corresponds to the first sequence number contained in $D2$'s sliding window, it moves the window by 1 position (and possibly more if packets with sequence numbers immediately after the first one were already received before) to the right. $D2$ is a two-threaded application, meaning it simultaneously receives packets and sends ACK's back to $S2$. The ACK sent back is always the first sequence number in the window, regardless of the fact that packets with greater sequence numbers may have already arrived.
$S2$ processes ACK's sent back by $D2$ while at the same time sends packets to $D2$ via the router. The starting position of $S2$'s window always aligns with the most recent ACK number sent by $D2$. Each time $S2$ send WINDOW_SIZE packets in one shot to $D2$. Every time $S2$ receives a greater ACK, it moves its window further to the new position indicated by the new ACK.
By the foregoing procedure, flow 2 can achieve a loss-free transmission since $S2$ knows from the feedback what packets $D2$ expects and it can accordingly keeps sending lost packets until they are all properly received by $D2$.

1.2 The plot reveals that smaller windows size gives a higher efficiency. Since for larger window sizes, upon every failure, a larger number of packets need to be retransmitted, lowering the efficiency.

1.3 The first peak in the timeout, which is also the global maximum, is due to transmission of client1. After client1 has finished transmission, timeout decreases. All the

following local peaks are due to the instability of the network.

1.4 The load of queue1 is always 0 since priority is given to packets from client1. The load of queue2 starts increasing and stays at the maximum capacity of the queue, and finally decreases to 0.

Task 2: Congestion Control

2.1 The plot illustrates how AIMD modifies the window size. Every time the packet being expected is not received, window size halves itself. After WINDOW_SIZE packets are received in a row, window size is incremented by 1. Under weak connectivity, the window decreases more rapidly.

2.2 The efficiency of flow2 gradually decreases because flow2 is competing with flow1. Flow1 has a higher priority. The efficiency starts to climb after client1 has finished transmission.

2.3 With congestion control, the loads of two queues still follow the same trend as that observed in part 1.4 but the congestion control alleviate the problem of overflowing by slowing down the time it takes to blow up the queue.

Task 3: Observe with Wireshark

3.1 The IP address of my computer is 10.10.64.34.

3.2 Within the IP packet header, the value in the upper layer protocol field is *UDP*.

3.3 20 bytes are in the IPv4 header. 141 bytes are in the payload of the IP datagram.

3.4 The IP datagram has not been fragmented. This can be determined by looking at the "Fragment offset", which is 0.

3.5 (Contents are recorded in hexadecimal)
Version + Header length – 1B – 45
Differentiated services field – 1B – 00
Total length – 2B – 00 a9
Identification – 2B – 8a 4e
Flags (contained within Fragment offset) – 1B – 00
Fragment offset – 2B – 00 00
Time to live – 1B – 40
Protocol – 1B – 11
Header checksum – 2B – 58 0b
Source (Source GeoIP)– 4B – 0a 0a 43 b5
Destination (Destination GeoIP)– 4B – 0a 0a 40 22

3.6 (a) At the campus Airbears network, we examined THE same packet which has the following port information:
Source port: 65152
Destination port: 56790
The destination port is the port hard-coded in our program and the source port is supposedly the sending port of the router in the network.

(b) When one of the laptops is connected over DSL /hot spot, the same packet's source and destination port is the same as what were hard-coded since .