Chris Banci

CS436

11/22/17

Homework 6 - Transport Layer

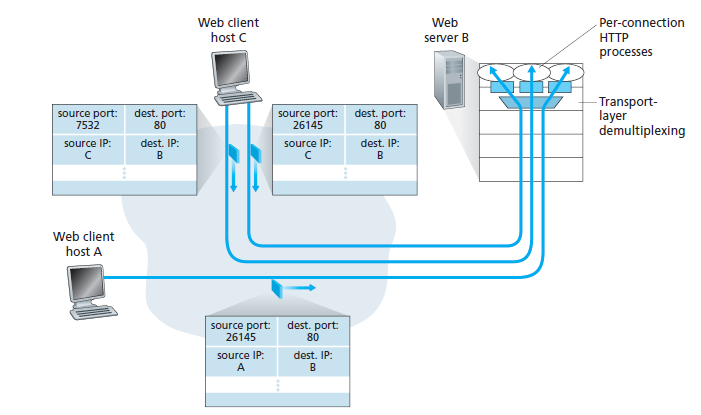
1. Primitives of transport service assume asymmetry between the two end points during connection establishment, one end (server) executes LISTEN while the other end (client) executes CONNECT. However, in peer to peer applications (i.e. *BitTorrent*) all end points are peers. There is no server or client functionality. How can transport service primitives be used to build such peer to peer applications? (4 pts)

To build a peer to peer applications using transport service primitives, before a connection is made between two peers, an application level mechanism can be used to inform each end point about who will act as the client and who will act as the server.

1. Consider a reliable data transfer protocol that uses only negative acknowledgments. 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? (4 pts)

A NAK-only protocol would be preferable when data is sent frequently and data loss is low because the NAK-only protocol will be able to instantly recognize when packets are dropped.

1. Consider the Figure below. What are the source and destination port values in the segments flowing from the server back to the clients’ processes? What are the IP addresses (use the host name) in the network-layer datagrams carrying the transport-layer segments? (6 pts)



Server B will return a response back to Host A using source port 80 and destination port 26145. Server B will return a response back to Host C using source port 80 and destination ports 26145 and 7532.

The IP addresses that are sent back to Host A and C from Server B will have the source IP set to the IP of the server and destination IP' set to the Host's IP addresses.

1. Consider the GBN protocol with a sender window size of 4 and a sequence number range of 1,024. Suppose that at time t, the next in-order packet that the receiver is expecting has a sequence number of k. Assume that the medium does not reorder messages. Answer the following questions: (4 pts)
2. What are the possible sets of sequence numbers inside the sender’s window at time *t*?

With the sender's window being of size 4, the possible sets of sequence numbers at time t is in the range between [K - 4] and [K].

1. What are all possible values of the ACK field in all possible messages currently propagating back to the sender at time *t*?

The possible values in all possible messages being propagating back to sender at time t is in the range between [K - 5] and [K - 1].

1. Some other policies for fairness in congestion control are Additive Increase Additive Decrease (AIAD), Multiplicative Increase Additive Decrease (MIAD), and Multiplicative. Increase Multiplicative Decrease (MIMD). Discuss these three policies in terms of convergence and stability.(6 pts)

In terms of convergence, AIAD and MIMD oscillates across the efficency line, but they will not converge. MIAD will converge just like AIAD. In terms of stability, none of these policies are stable.

1. Both UDP and TCP use port numbers to identify the destination entity when delivering a message. Give two reasons why these protocols invented a new abstract ID (port numbers), instead of using process IDs, which already existed when these protocols were designed. (4 pts)
2. Process ID's are not static; the system assigns a different ID each time a process is started.
3. A single process can have multiple channels for communication; process ID cannot be used to distinguish between the different channels.
4. What is the total size of the minimum TCP MTU, including TCP and IP overhead but not including data link layer overhead? (2 pts)

536 bytes (min default payload)

20 bytes (min TCP header size)

* 20 bytes (min IP header size)

-----------------------------------------

The total size of the minimum TCP MTU is 576 bytes.

1. Suppose that five measured SampleRTT values are 106ms, 120ms, 140ms, 90ms, and 115ms. (6 pts)
   1. Compute the EstimatedRTT after each of these SampleRTT values is obtained, using a value of α = 0.125 and assuming that the value of EstimatedRTT was 100ms just before the first of these five samples were obtained.

EstimatedRTT = (α) \* SampleRTT + (1-α) \* previousEstimatedRTT

* 106ms: 0.125 \* 106 + 0.875 \* 100 = 100.75ms
* 120ms: 0.125 \* 120 + 0.875 \* 100.75 = 103.16ms
* 140ms: 0.125 \* 140 + 0.875 \* 103.16 = 107.77ms
* 90ms: 0.125 \* 90 + 0.875 \* 107.77 = 105.55ms
* 115ms: 0.125 \* 115 + 0.875 \* 105.55 = 106.73ms
  1. Compute also the DevRTT after each sample is obtained, assuming a value of β = 0.25 and assuming the value of DevRTT was 5ms just before the first of these five samples was obtained.

DevRTT = (β) \* (|SampleRTT − EstimatedRTT|) + (1 − β) \* DevRTT

* 106ms: 0.25 \* |(106 - 100.75)| + 0.75 \* 5 = 5.06ms
* 120ms: 0.25 \* |(120 − 103.15)| + 0.75 \* 5.06 = 8.01ms
* 140ms: 0.25 \* |(140 − 107.76)| + 0.75 \* 8.01 = 14.07ms
* 90ms: 0.25 \* |(90 − 105.54)| + 0.75 \* 14.07 = 14.44ms
* 115ms: 0.25 \* |(115 − 106.71)| + 0.75 \* 14.44 = 12.90ms
  1. Compute the TCP TimeoutInterval after each of these samples is obtained.

TCP TimeoutInterval = EstimatedRTT + 4 \* DevRTT

* 106ms: 100.75 + 4 \* 5.06 = 120.99ms
* 120ms: 103.15 + 4 \* 8 = 135.15ms
* 140ms: 107.76 + 4 \*14.06 = 164ms
* 90ms: 105.54 + 4 \*14.42 = 163.22ms
* 115ms: 106.71 + 4 \*12.88 = 158.23ms

1. Suppose that the TCP congestion window is set to 18 KB and a timeout occurs. How big will the window be if the next four transmission bursts are all successful? Assume that the maximum segment size is 1 KB. (5 pts)

When a timeout occurs, the congestion window starts at 1 and threshold reset to 8kb.

The number of segments doubles after each transmission.

* 1st transmission = 1kb
* 2nd transmission = 2kb
* 3rd transmission = 4kb
* 4th transmission = 8kb

1. Consider Figure below. Assuming TCP Reno is the protocol experiencing the behavior shown above, answer the following questions. (7 pts)
2. Identify the intervals of time when TCP slow start is operating.

interval [1, 6] and interval [23, 26]

1. Identify the intervals of time when TCP congestion avoidance is operating.

interval [6, 16] and interval [17, 22]

1. After the 16th transmission round, is segment loss detected by a triple duplicate ACK or by a timeout?

The segment loss is detected by a triple duplicate ACK.

1. After the 22nd transmission round, is segment loss detected by a triple duplicate ACK or by a timeout?

The segment loss is detected by a timeout.

1. What is the initial value of ssthresh at the first transmission round?

The initial value at the 1st transmission round is 32.

1. What is the value of ssthresh at the 18th transmission round?

The value at the 18th transmission round is 21.

1. What is the value of ssthresh at the 24th transmission round?

The value at the 24h transmission round is 13.

