Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich
Spring Term 2014

Operating Systems and Networks
Assignment 9

Assigned on: **17th April 2014**
Due by: **24th April 2014**

1 UDP

Why is UDP used if applications do not want to have connection-oriented communications? Would it not have been enough to just let user processes send raw IP packets?

Answer:

No. IP packets contain IP addresses, which specify a destination machine. Once such a packet arrived, how would the network handler know which process to give it to? UDP packets contain a destination port. This information is essential so they can be delivered to the correct process

2 TCP Three-Way Handshake

In the lecture you learned how TCP uses three-way handshake to establish a session.

Imagine that a two-way handshake rather than a three-way handshake were used to set up connections. In other words, the third message was not required. Could there be any problem? If so, please illustrate with an example.

Answer:

One problem is that a deadlock is possible. For example, a packet arrives at A out of the blue, and A acknowledges it. The acknowledgement gets lost, but A is now open while B knows nothing at all about what has happened. Now the same thing happens to B, and both are open, but expecting different sequence numbers. Timeouts have to be introduced to avoid the deadlocks.

3 Sliding Windows

a) *Using 5-bit sequence numbers, what is the maximum size of the send and receive windows for the Go-Back-N and Seletive-Repeat algorithms?*

Answer: Some supplementary materials for ”go-back-N” and ”selective repeat”

`http://webmuseum.mi.fh-offenburg.de/index.php?view=exh&src=73` (Animates the impact of the Maximum Window size)

For the Go-Back-N algorithm, the size of the send and receive windows are 32 - 1 = 31 and 1, respectively. For the Selective-Repeat algorithm, the size of the send and receive windows are 32/2 = 16 and 32/2 = 16, respectively.

b) *Frames of 1000 bits are sent over a 1 Mbps satellite channel - propagation delay of 270 msec. Acknowledgements are always piggybacked. Headers are very short. Three bit sequence numbers are used. What is the maximum achievable channel utilization in frames per second for (a) protocol with Go-Back-N and (b) protocol with Selective-Repeat ?*

Answer: At 1Mbps, 1000 bits take 1 msec (neglect headers). The round trip time for a frame is 1+270+270+1 = 542 msec. (till ack is received by the sender).

(a) pipeline 7 frames (3 bit seq number) with roundtrip of 548 msec giving channel utilization 7/548 = 12.77 frames per second.

(b) pipeline 4 frames (window size is sequence-size /2) in 545 msec giving channel utilization 4/545 = 7.34 frames per second.

4 Maximum Throughput Computation

We assume that packet sequence number increases for each packet from a source to a destination and the maximum packet size is 1500 bytes. The packet sequence number for a flow starts from 0x0, and wraps around to 0x0 after the flow generates 2^x packets, where x is the length of packet sequence number. To get around the problem of sequence numbers wrapping around while old packets still exist one could use 64 bits to indicate sequence numbers.

a) *Theoretically, an optical fiber running at 75 Tbps. What maximum packet lifetime is required to make sure that future 75 Tbps networks do not have wraparound problems even with 64-bit sequence numbers? Assume that each byte has its own sequence number, as TCP does.*

Answer: A 75-Tbps transmitter uses up sequence space at a rate of 9.375×10^{12} sequence numbers per second. It takes 2 million seconds to wrap around. Since there are 86,400 seconds in a day, it will take over 3 weeks to wrap around, even at 75 Tbps. A maximum packet lifetime of less than 3 weeks will prevent the problem. In short, going to 64 bits is likely to work for quite a while.

b) *In practice, maximum packet lifetime is set to 2 minutes, and packets use 16-bit sequence numbers. What is the real throughput?*

Answer: Thoughput = $2^{16} \times 1500/120 = 800$ KBps.

5 UDP Socket Programming

Implement UDP client/server programs. The UDP client program sends a sentence (i.e. a character string which is entered by the user) to the UDP server program via UDP sockets. The UDP server displays the string, and sends it back to the UDP client that then displays the result.

Answer:

1. You can use any language of your taste (C language is a very good option). The idea of this problem is to give you some exposure to the basics of Socket programming.

2. Please use the standard socket library for the language of your choice. (i.e. socket(), recvfrom(), etc, if you decide to use C.)

3. The problem is asking you to build a UDP echo client and a UDP echo server. That is the client program should take any string input from the user(console), and send the users input to the server program. Upon receiving the packet sent by the client, the server displays the content of the packet (users input) and echoes/replies back to the client program. Finally, the client program should display the message (same as the users original input) it has received from the server.