

COS 440/540
Computer Networks
Homework 4
Due Wednesday, November 12th, 11:59 PM

Reliable UDP File Transfer

Learning Goals:

In this assignment you will gain experience using UDP for reliable file transfer between two cloud environments. It will give you an appreciation of what is involved in providing reliability in an unreliable network environment and experience in using the only alternative to TCP in the Internet environment.

Basic Assignment

You will implement a reliable file transfer mechanism between a VM on Jetstream and a VM on ArcSim using UDP (no credit for using two Jetstream or two ArcSim VMs). You will transfer a 1-MB file, logically decomposed into 1024 messages (datagrams), each of which is 1024 bytes. The server will read a 'chunk' from the file, attach a 'chunk' number to the message and send it to the client. The client will receive these messages and write them into the appropriate position within the file or a buffer as discussed below.

Because datagrams can get lost in the network and do not necessarily arrive in order you will need to implement a reliability mechanism to guarantee the file is correctly received. The basic approach to providing reliability should be as we discussed in class. The client maintains an Acknowledgment buffer (call it AckBuf) consisting of 1024-bytes initialized to '0's. As messages are received (file 'chunks') the corresponding 'chunk' position within the AckBuf is set to '1'. Periodically, the receiver sends the AckBuf to the server which retransmits all missing 'chunks'. This cycle continues until the entire file has been successfully transferred.

You are to implement two slightly different versions of the client and track the number of ack/retransmit cycles required to complete the transfer for each approach. First, have the client write the file 'chunk' into the appropriate position within the file as the chunk is received. Second, have the client write the chunks into their appropriate position within a buffer and write the entire buffer to the file when the transfer is completed. Please briefly discuss/explain your observations.

NOTE: Your program should be based on the material/approach discussed in class. A submission using a completely different approach will be viewed as non-compliant and receive a grade of 0.

Graduate Version:

Same as above except you will use a 4-MB file decomposed into 4096 messages of 1K bytes. The problem is that you cannot send a 4096-byte acknowledgement message because it will get dropped (due to the way cloud security treats datagram fragmentation).

This means you will need to use bits rather than bytes for your AckBuf. In this approach, you would create a 512-byte AckBuf where each *bit* represents the received/not-received status of the corresponding chunk. Thus, for example, bit 3 of byte 1 would represent the received/not-received status of message 11.

IMPORTANT NOTE: The UDP Server on Jetstream must bind to a port in the 60000 range and the client on ArcSim must bind to a port in the 8000 range! This means that the client must send the 'ready' message to the server using the port in the 60000 range and receive messages using a port in the 8000 range.

Restrictions: Do not use any command line arguments (e.g., for IP address, Port number, receive version). Rather, use #define at the top of your program.

Deliverables: Please upload to BrightSpace a **zip** file of your C source code and a brief explanation of your experimental results. This is due Wednesday, November 12th, 11:59 PM.