

Lab6 (100 points)

Important: We will demo in class or online. For in class demos, we will go around the room. Be prepared to demo at the start of class. The submitted code will be used only to verify that you did not copy from others, to compile and re-run your program, to make sure you were indeed demonstrating your own code, and to grade for documentation of your code.

In this program we will start with Lab5. We are going to introduce 2 new concepts. First, we will have the receiver (destination) of the message ACK the message. This implies that we will have to have a sequence number in the message. This will be key value of **seqNumber:n** (where n is an integer). The first time the receiver gets a message, it will print out the message like in lab 5. It will then send an ACK back to the sender. If the receiver gets the message again, it will print out the sequence number, fromPort and a message saying "duplicate". It will then ACK the message again. This implies that we have a new message type called ACK. This will be key value of **type:ACK**

The second concept we will introduce is the concept of tracking the nodes (port numbers) from sender to receiver. This will be included in **every** message (whether it is an ACK or not). We will call this the "send-path". Every hop along the way will add its portNumber to the sendpath, separated by a comma. The destination will print out the send-path along with the message. This will be **send-path:<port1>, <port2>**

We will assume the following for this lab:

- Each toPort-fromPort will start at seq number 1
- There will be a maximum of 100 messages between any toPort-fromPort pair (don't have to check for this, it will help when creating data structures)
- The maximum message length will be 200 bytes (this includes all fields in the message)

The fields that **MUST** be included in the ACK are: send-path toPort fromPort seqNumber TTL (that it is acking) type:ACK location version

The output on the receiver will look something like:

sending an ack for seqNum 1 partner # 0, fromPort 20001

Name	Value
msg	"hello 20015"
version	6

toPort	20015
location	7
fromPort	20001
TTL	2
seqNumber	1
send-path	20001,20007
myLocation	15

The output on the sender (when they get an ACK) will look something like:
received an ACK for SeqNum 1 fromPort 20015

Name	Value
send-path	20015,20007
TTL	2
version	6
toPort	20001
fromPort	20015
seqNumber	1
type	ACK
location	7
myLocation	1

Everything else will stay the same from Lab6.

Submit well-documented and well indented code along with a README file explaining how to run the program, and a makefile. Submit it using GitHub

The grading rubric is as follows:

- Program correctness and robustness (what happens if I give garbage input): 80%
- Coding style (comments, indentations, README, Makefile): 20%