

RDT 3.0 Simulation

Victoria Joy Scott

85449069

Victoriajoyscott@ufl.edu

Compilation:

```
javac network.java
javac receiver.java
javac sender.java
```

-or-

```
javac *.java
```

Execution: My code has default values for all 3 java files. If passed no arguments the programs will default to port 9069 on localhost. Sender.java will read from test.txt.

```
java network
java receiver
java sender
```

-or-

```
java network 1234
java receiver localhost 1234
java sender localhost 1234 myFile.txt
```

Code structure:

Sender.java is a while loop that executes once per word. Inside there are more loops that will consistently send the same message until the expected response is received. Once the proper ACK is received sender.java will construct the new message to send. When a period is detected the code will enter termination mode. Once the proper ACK is received sender gracefully terminates.

Receiver.java implements a similar structure. Receiver.java will continue onto the next ACK once it receives a message with the proper sequence number and checksum. The proper message is concatenated to the final output message. Upon receiving a -1 termination byte the message is printed and the program gracefully terminates.

Network.java has two classes. Receiver_ and sender_ are the two threads that my network will accept. Receiver_ must always be connected first. When a packet comes in the network uses the Math.random() function to determine if the packet should be PASS, DROP, or CORRUPT. DROP will send the message

"ACK2", CORRUPT adds one to the checksum and PASS gives the message forward unchanged. When the -1 byte is received from the sender -1 is passed to the receiver then the program gracefully terminates.

Examples:

I demonstrate the default values explained above in this screenshot. The receiver's ACKs are alternating and the total number of packets sent counting logically. The message received is then printed. The sender is sending and resending packets as expected and the network is behaving as the go-between.

```

ACK1, 6, no more packets to send
vria_scott_85449069 java sender
ACK0, 0, send Packet0
ACK0, 1, resend Packet0
ACK0, 2, resend Packet0
ACK0, 3, resend Packet0
ACK0, 4, resend Packet0
ACK0, 5, resend Packet0
ACK1, 6, send Packet1
ACK0, 7, send Packet0
ACK0, 8, send Packet0
ACK1, 9, send Packet1
vria_scott_85449069

victoria_scott_85449069 java receiver
Waiting 0, 1, 0 0 189 We, ACK1
Waiting 0, 2, 0 0 189 We, ACK1
Waiting 0, 3, 0 0 189 We, ACK1
Waiting 0, 4, ACK2, ACK1
Waiting 0, 5, 0 0 189 We, ACK1
Waiting 0, 6, 0 0 188 We, ACK1
Waiting 1, 7, 1 1 655 should, ACK0
Waiting 0, 8, 1 1 656 should, ACK1
Waiting 0, 9, 0 2 199 be, ACK1
Waiting 1, 10, 1 3 912 together, ACK0
Message: We should be together.
victoria_scott_85449069

78 s.close();
79 System.exit(1);
80 }
81 split = input.split(" ");
82 }
83 }
84
85 //proper ACK received update message
86

No results found for 'sender'

Received: ACK0 PASS
Received: Packet1, 3, DROP
Received: ACK0 DROP
Received: Packet1, 3, PASS
Received: ACK1 PASS
victoria_scott_85449069 java network
Hello from a receiver!
Ready to connect sender
Hello from a sender!
Received: Packet0, 0, CORRUPT
Received: Packet0, 0, CORRUPT
Received: ACK1 PASS
Received: Packet0, 0, CORRUPT
Received: ACK1 CORRUPT
Received: Packet0, 0, DROP
Received: ACK1 DROP
Received: Packet0, 0, CORRUPT
Received: ACK1 DROP
Received: Packet0, 0, PASS
Received: ACK0 PASS
Received: Packet1, 1, PASS
Received: ACK1 DROP
Received: Packet1, 1, CORRUPT
Received: ACK1 PASS
Received: Packet0, 2, PASS
Received: ACK0 PASS
Received: Packet1, 3, PASS
Received: ACK1 PASS

```

This screenshot demonstrates the non-default values.

```

ACK1, 6, send Packet1
ACK1, 7, resend Packet1
ACK0, 8, send Packet0
ACK1, 9, send Packet1
vria_scott_85449069 java sender localhost 4567 message.txt
ACK0, 0, send Packet0
ACK0, 1, resend Packet0
ACK0, 2, resend Packet0
ACK0, 3, resend Packet0
ACK1, 4, send Packet1
ACK1, 5, resend Packet1
ACK0, 6, send Packet0
ACK0, 7, resend Packet0
ACK0, 8, resend Packet0
ACK0, 9, resend Packet0
ACK0, 10, resend Packet0
ACK0, 11, resend Packet0
ACK0, 12, resend Packet0
ACK1, 13, send Packet1
vria_scott_85449069

Waiting 0, 9, 0 2 199 be, ACK1
Waiting 1, 10, 1 3 912 together, ACK0
Message: We should be together.
victoria_scott_85449069 java receiver
Waiting 0, 1, 0 0 189 We, ACK1
Waiting 0, 2, 0 0 189 We, ACK1
Waiting 0, 3, 0 0 189 We, ACK1
Waiting 0, 4, 0 0 188 We, ACK1
Waiting 1, 5, 1 1 656 should, ACK0
Waiting 1, 6, 1 1 655 should, ACK0
Waiting 0, 7, 0 2 199 be, ACK1
Waiting 1, 8, ACK2, ACK0
Waiting 1, 9, 0 2 199 be, ACK0
Waiting 1, 10, 0 2 200 be, ACK0
Waiting 1, 11, 0 2 199 be, ACK0
Waiting 1, 12, 0 2 199 be, ACK0
Waiting 1, 13, 0 2 200 be, ACK0
Waiting 1, 14, 1 3 912 together, ACK0
Message: We should be together.
victoria_scott_85449069

78 s.close();
79 System.exit(1);
80 }
81 split = input.split(" ");
82 }
83 }
84
85 //proper ACK received update message
86

No results found for 'sender'

sender

Received: Packet0, 2, PASS
Received: ACK0 PASS
Received: Packet1, 3, PASS
Received: ACK1 PASS
victoria_scott_85449069 java network 4567
Hello from a receiver!
Ready to connect sender
Hello from a sender!
Received: Packet0, 0, CORRUPT
Received: ACK1 PASS
Received: Packet0, 0, CORRUPT
Received: ACK1 DROP
Received: Packet0, 0, CORRUPT
Received: ACK1 CORRUPT
Received: Packet0, 0, PASS
Received: ACK0 PASS
Received: Packet1, 1, CORRUPT
Received: ACK0 PASS
Received: Packet1, 1, PASS
Received: ACK1 PASS
Received: Packet0, 2, PASS
Received: ACK0 DROP
Received: Packet0, 2, DROP
Received: ACK0 DROP
Received: Packet0, 2, PASS
Received: ACK0 DROP
Received: Packet0, 2, CORRUPT
Received: ACK0 DROP
Received: Packet0, 2, PASS
Received: ACK0 DROP
Received: Packet0, 2, PASS
Received: ACK0 CORRUPT
Received: Packet0, 2, CORRUPT
Received: ACK0 PASS
Received: Packet1, 3, PASS
Received: ACK1 PASS
victoria_scott_85449069

```

Limitations:

My program will exit if there is a period anywhere in the final word. Periods should not be in the middle of words unless they are the final word in the message.

I was having an issue compiling my code but I think it resolved. I had my classes sender and receiver named the same as my java codes. I changed my class names to sender_ and receiver_ respectively. This fixed the compilation bug. If you run into the bug during testing execute the code this way:

```
javac network.java
```

```
java network 1234
```

```
javac receiver.java
```

```
java receiver localhost 1234
```

```
javac sender.java
```

```
java sender localhost 1234 myFile.txt
```