

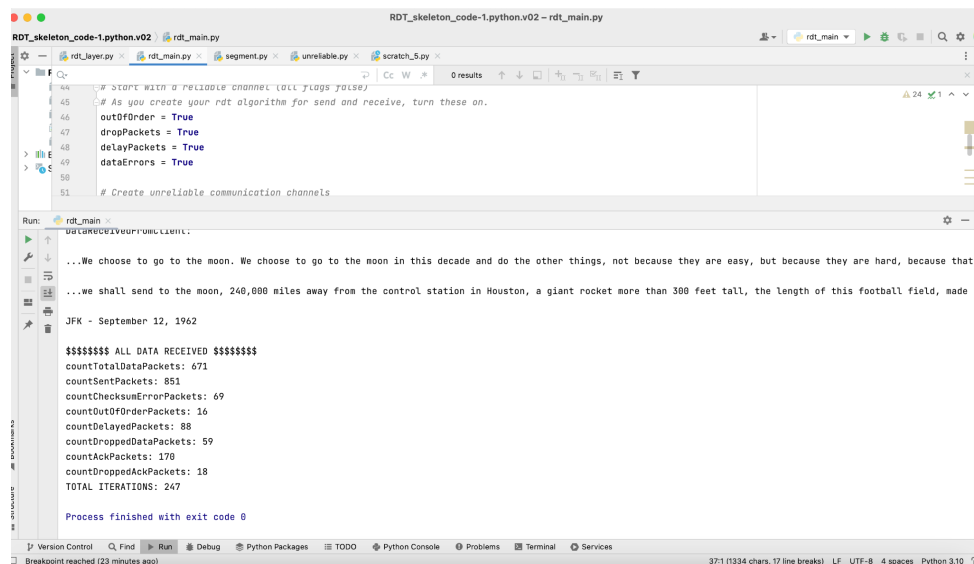
Daniel Arevalo

arevalda@oregonstate.edu

CS 372 Programming Assignment 3

This assignment was difficult, but actually fun and challenging. I spent a lot of time trying to optimize it, although I am still very far from optimized in comparison to the examples given.

Truthfully, it is still somewhat buggy. I have not figured out why, but about 1/10 of the run times of the larger data will trigger an inf loop.



```
RDt_skeleton_code-1.python.v02 - rdt_main.py
RDt_skeleton_code-1.python.v02 - rdt_main.py
rdt_layer.py x rdt_main.py x segment.py x unreliable.py x scratch_5.py x
# Start with a reliable channel (all flags false)
# As you create your rdt algorithm for send and receive, turn these on.
outOfOrder = True
dropPackets = True
delayPackets = True
dataErrors = True
# Create unreliable communication channels

Run: rdt_main
Welcome to the RDT layer.
...We choose to go to the moon. We choose to go to the moon in this decade and do the other things, not because they are easy, but because they are hard, because that
...we shall send to the moon, 240,000 miles away from the control station in Houston, a giant rocket more than 300 feet tall, the length of this football field, made
JFK - September 12, 1962

$$$$$$$$ ALL DATA RECEIVED $$$$$$$$
countTotalDataPackets: 671
countSentPackets: 851
countChecksumErrorPackets: 69
countOutOfOrderPackets: 16
countDelayedPackets: 88
countDroppedDataPackets: 59
countAckPackets: 178
countDroppedAckPackets: 18
TOTAL ITERATIONS: 247

Process finished with exit code 0
```

This is the smaller test data running with all flags set.

See next page for second example photo

The screenshot shows a Python IDE window titled "RDT_skeleton_code-1.python.v02 - rdt_main.py". The editor displays the following code:

```
43  
44 # Start with a reliable channel (all flags false)  
45 # As you create your rdt algorithm for send and receive, turn these on.  
46 outOfOrder = True  
47 dropPackets = True  
48 delayPackets = True  
49 dataErrors = True
```

The Run console shows the following output:

```
Client-----  
Sending segment: seq: 29, ack: -1, data: ver  
Sending segment: seq: 33, ack: -1, data: the  
Sending segment: seq: 37, ack: -1, data: lazy  
Server-----  
Sending ack: seq: -1, ack: 45, data:  
Main-----  
DataReceivedFromClient: The quick brown fox jumped over the lazy dog  
$$$$$$$ ALL DATA RECEIVED $$$$$$$  
countTotalDataPackets: 16  
countSentPackets: 21  
countChecksumErrorPackets: 3  
countOutOfOrderPackets: 2  
countDelayedPackets: 4  
countDroppedDataPackets: 0  
countAckPackets: 5  
countDroppedAckPackets: 2  
TOTAL ITERATIONS: 8  
  
Process finished with exit code 0
```

The IDE interface includes a sidebar with icons for Explorer, Search, and Run and Debug. The bottom status bar shows "Breakpoint reached (22 minutes ago)".

Larger test data with all flags set.