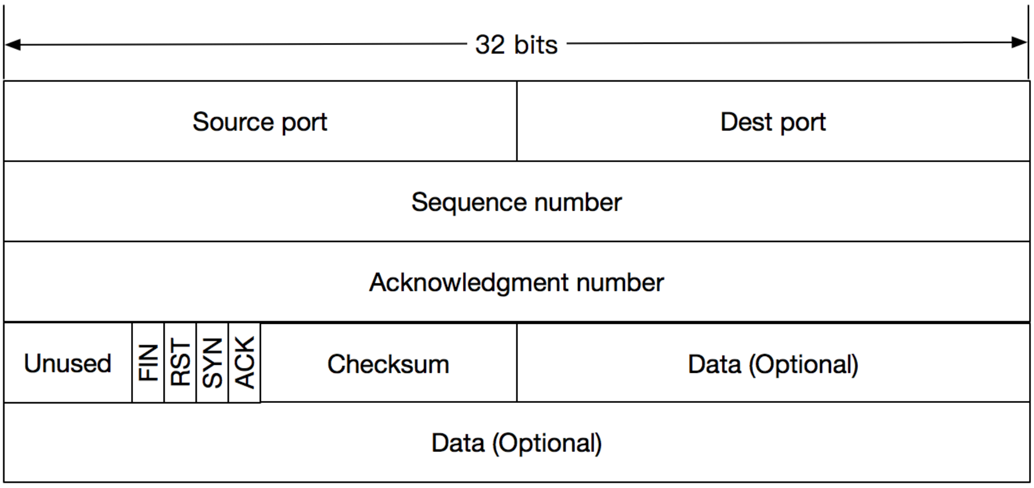
Report for STP Protocol

I implemented my STP protocol with the UDP socket API in python 3.
Example for transfer test0.pdf command.
1. Python3 receiver.py 31500 output.pdf
2. Python3 sender.py 127.0.0.1 31500 test0.pdf 500 100 4 0.1 0 0 0 0 0 0 100
Similar to TCP protocol, I used the first 10 bytes of the UDP data as STP header, which includes the information about sequence number, acknowledgment number, flag and checksum, while the remaining part of the UDP body will be STP data.
With on the information in STP header, I implemented all of the features required by our assignment, which include:

1. A three-way handshake
2. A four-segment connection termination.
3. Timeout operation
4. Fast Retransmit
5. Sequence number and acknowledgment number
6. Maximum segment size and maximum windows size
7. PLD Module, which supports drop, duplicate, reorder, bit errors and delay packets

The format of a STP packet is shown in the figure below:



...

The Source port and Dest port are parts of an UDP header. And the Sequence number and the Acknowledgement number both take 4 bytes, which have the same meaning with those of TCP protocol. The next one byte contains all of the flags that required by STP, which including ACK, SYN, RST and FIN, and all of them have the same meaning with those of TCP protocol. As for the 1-byte checksum, assume that the sum of every single byte (except Source port and Dest port) in a SCP packet is A, if A % 0x100 == 0xFF, the packet will be valid; otherwise it will be invalid. While the remaining part will be the optional STP data part.

One thing I want to discuss in my STP protocol is about error recovery. For now, in my implementation, if there is some errors (including network connection error, packet error and so on) during the handshake or termination process, my implementation will just print error message and quit with an exit code of 1. I think it may be better if I add some re-connection mechanism. Meanwhile, if there are some errors during data transferring, my implementation will always re-transfer the data packet. And I think one possible issue of this implementation is that, if there are some serious network error, my system may enter a dead loop of transferring data to a un-reachable address. And I think one possible solution is to add a maximum number of time for re-transferring a single packet. For example, assume the maximum number is 10, if my sender send try to re-transfer the same STP packet for 10 times without any success, the sender may just print out error message and quite the program with an exit code of 1.

As for the reference code, I only referred to some sample code about how to send and receive an UDP packet in python document. As for the implementation and application logic of STP itself, I did not refer to any other material except for the text book and the assignment requirement.

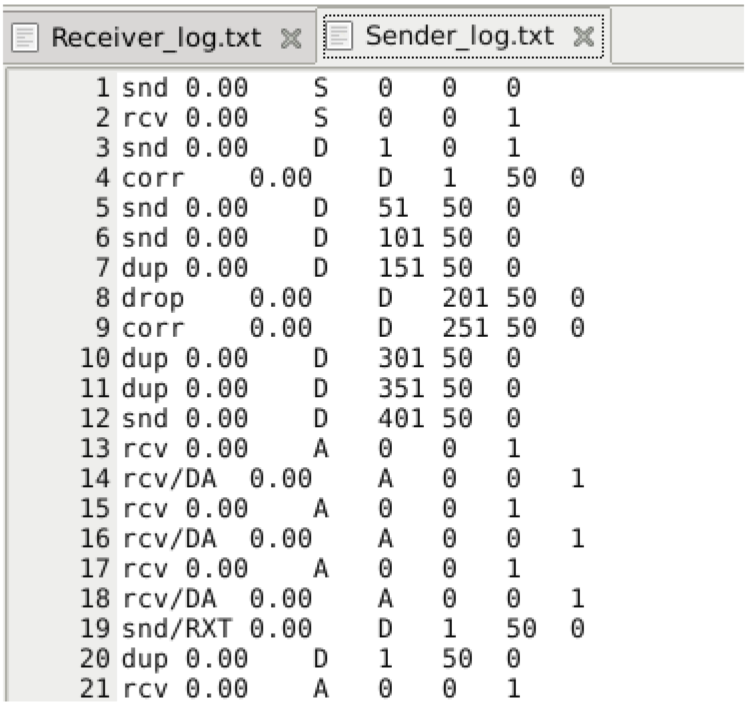
Answers to the 3 questions in Assignment:
(a)
The results are shown in Appendix A. When the pdrop is larger, there will be more packets that are lost during the transferring. For example, in the result of pdrop = 0.3, at time 5.51, the receiver received a packet with Sequence Number of 501 without receiving the packet with Sequence Number of 401, which means that the packet with Sequence Number of 401 seems to be dropped. Since the initial timeout time interval in our system is large, the packets lost here will cause much delay for the overall performance, it will take more time to transfer the whole file correctly when pdrop = 0.3

(b)
The results are shown in the table below:

Gamma	Number of Packets	Duration (s)
2	33143	24.87
4	31843	40.69
6	30008	59.25

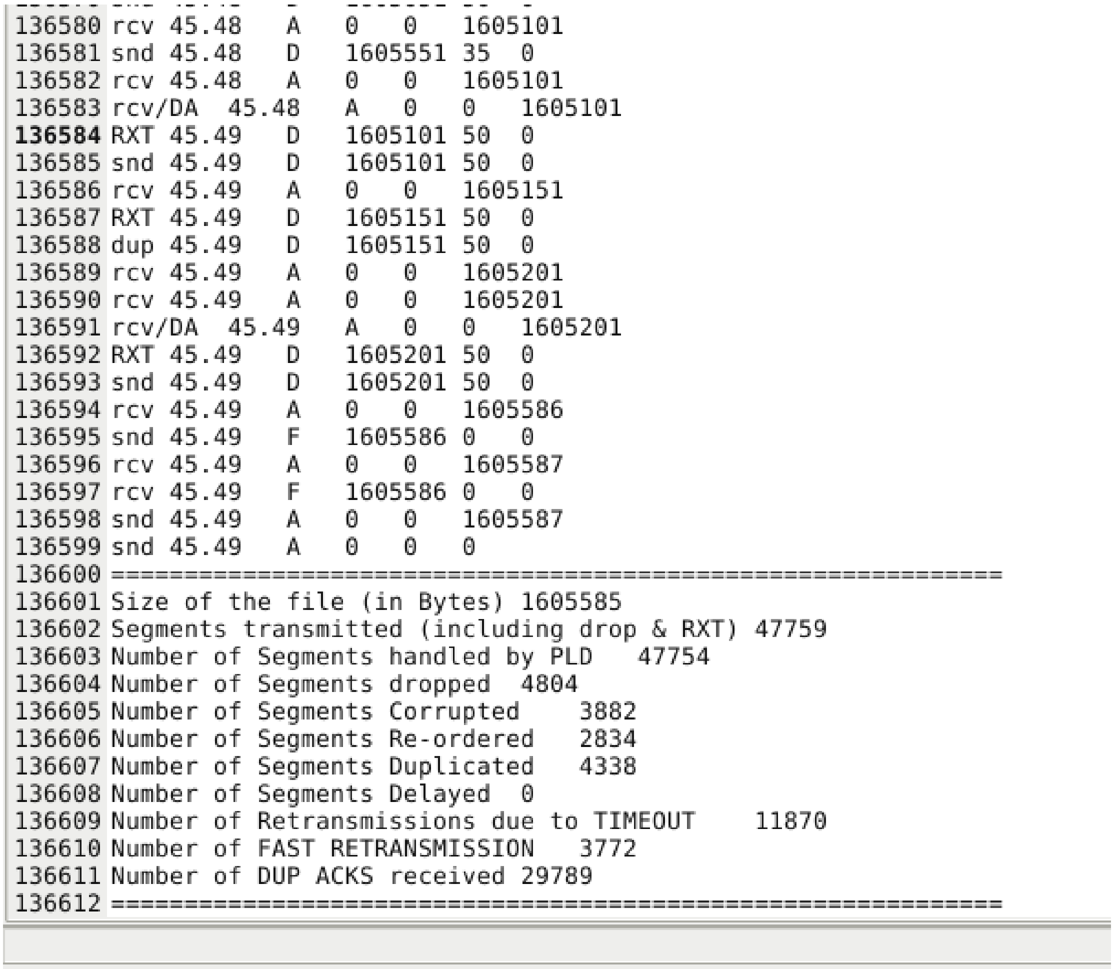
With the increasement of Gamma, the Number of Packets decreases slightly, while the duration increase largely. I think the main reason is that, Gamma will influence the speed of the change of the timeout time interval of the STP protocol. If the Gamma is large, the timeout time interval will change slowly toward the expected best time interval for STP protocol. As a result, since the initial timeout time interval for STP protocol is large, the duration will also be large is Gamma is large.

(c)
Results:
Sender:
Connection establishment + First 20:



```
snd 0.00    S   0   0   0
rcv 0.00    S   0   0   1
snd 0.00    D   1   0   1
corr    0.00    D   1   50  0
snd 0.00    D   51  50  0
snd 0.00    D   101 50  0
dup 0.00    D   151 50  0
drop    0.00    D   201 50  0
corr    0.00    D   251 50  0
dup 0.00    D   301 50  0
dup 0.00    D   351 50  0
snd 0.00    D   401 50  0
rcv 0.00    A   0   0   1
rcv/DA  0.00    A   0   0   1
rcv 0.00    A   0   0   1
rcv/DA  0.00    A   0   0   1
rcv 0.00    A   0   0   1
rcv/DA  0.00    A   0   0   1
snd/RXT 0.00    D   1   50  0
dup 0.00    D   1   50  0
rcv 0.00    A   0   0   1
```

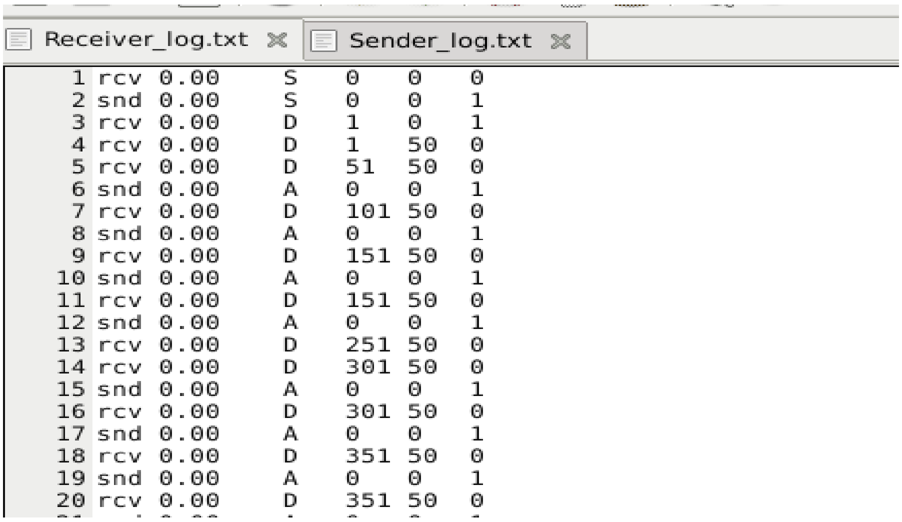
Last 20 and Summary:



```
rcv 45.48   A   0   0   1605101
snd 45.48   D   1605551 35  0
rcv 45.48   A   0   0   1605101
rcv/DA  45.48   A   0   0   1605101
RXT 45.49   D   1605101 50  0
snd 45.49   D   1605101 50  0
rcv 45.49   A   0   0   1605151
RXT 45.49   D   1605151 50  0
dup 45.49   D   1605151 50  0
rcv 45.49   A   0   0   1605201
rcv 45.49   A   0   0   1605201
rcv/DA  45.49   A   0   0   1605201
RXT 45.49   D   1605201 50  0
snd 45.49   D   1605201 50  0
rcv 45.49   A   0   0   1605586
snd 45.49   F   1605586 0   0
rcv 45.49   A   0   0   1605587
rcv 45.49   F   1605586 0   0
snd 45.49   A   0   0   1605587
snd 45.49   A   0   0   0
============================================================
Size of the file (in Bytes) 1605585
Segments transmitted (including drop & RXT) 47759
Number of Segments handled by PLD   47754
Number of Segments dropped  4804
Number of Segments Corrupted    3882
Number of Segments Re-ordered   2834
Number of Segments Duplicated   4338
Number of Segments Delayed  0
Number of Retransmissions due to TIMEOUT    11870
Number of FAST RETRANSMISSION   3772
Number of DUP ACKS received 29789
============================================================
```

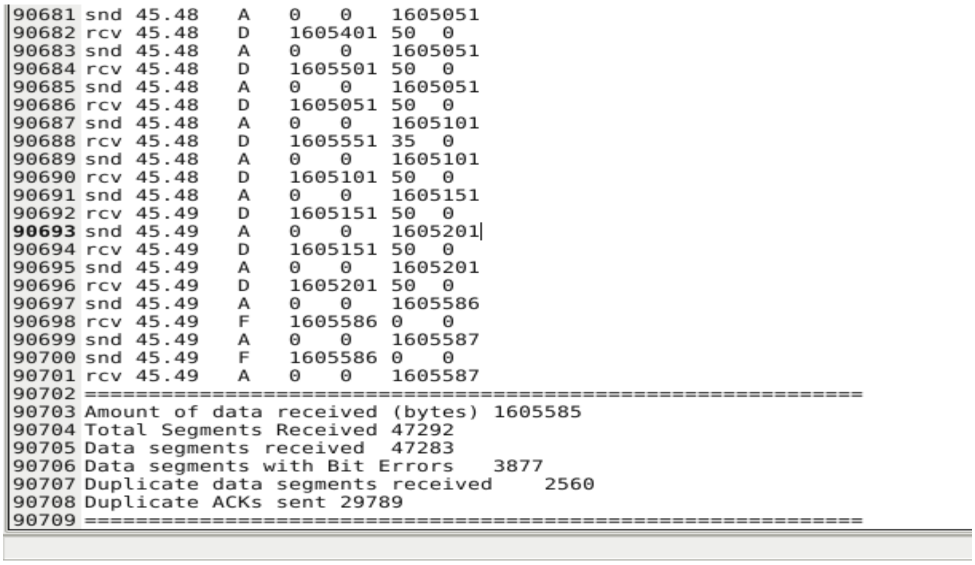
Receiver:

Connection establishment + First 20:



```
rcv 0.00    S   0   0   0
snd 0.00    S   0   0   1
rcv 0.00    D   1   0   1
rcv 0.00    D   1   50  0
rcv 0.00    D   51  50  0
snd 0.00    A   0   0   1
rcv 0.00    D   101 50  0
snd 0.00    A   0   0   1
rcv 0.00    D   151 50  0
snd 0.00    A   0   0   1
rcv 0.00    D   151 50  0
snd 0.00    A   0   0   1
rcv 0.00    D   251 50  0
rcv 0.00    D   301 50  0
snd 0.00    A   0   0   1
rcv 0.00    D   301 50  0
snd 0.00    A   0   0   1
rcv 0.00    D   351 50  0
snd 0.00    A   0   0   1
rcv 0.00    D   351 50  0
```

Last 20 and Summary:



```
snd 45.48   A   0   0   1605051
rcv 45.48   D   1605401 50  0
snd 45.48   A   0   0   1605051
rcv 45.48   D   1605501 50  0
snd 45.48   A   0   0   1605051
rcv 45.48   D   1605051 50  0
snd 45.48   A   0   0   1605101
rcv 45.48   D   1605551 35  0
snd 45.48   A   0   0   1605101
rcv 45.48   D   1605101 50  0
snd 45.48   A   0   0   1605151
rcv 45.49   D   1605151 50  0
snd 45.49   A   0   0   1605201
rcv 45.49   D   1605151 50  0
snd 45.49   A   0   0   1605201
rcv 45.49   D   1605201 50  0
snd 45.49   A   0   0   1605586
rcv 45.49   F   1605586 0   0
snd 45.49   A   0   0   1605587
snd 45.49   F   1605586 0   0
rcv 45.49   A   0   0   1605587
==============================================================
Amount of data received (bytes) 1605585
Total Segments Received 47292
Data segments received  47283
Data segments with Bit Errors   3877
Duplicate data segments received    2560
Duplicate ACKs sent 29789
==============================================================
```

The file has been successfully transferred. It took 28.40 seconds in total. I think the gamma is the main factor for the overall transferring time, since in receive, there are only 198 lines of log in the first 16.95 seconds, while there are more than 200,000 lines of log in the last 13 seconds, which means that in the first 16.95 second, the packets have been sent slowly while in the last 13 seconds the packets have been send quickly. As discussed in question (a) and (b), the main reason for it is that the initial timeout time interval is quite large, and the value of Gamma can control the speed of change of the timeout time interval.

Appendix A

pdrop = 0.1

```
rcv  0.00 S  0    0    0
snd  0.00 S  0    0    1
rcv  0.00 D  1    0    1
rcv  0.00 D  1    100  0
snd  0.00 A  0    0    101
rcv  0.00 D  101  100  0
snd  0.00 A  0    0    201
rcv  0.00 D  301  100  0
snd  0.00 A  0    0    201
rcv  0.00 D  401  100  0
snd  0.00 A  0    0    201
rcv  0.00 D  501  100  0
snd  0.00 A  0    0    201
rcv  0.00 D  601  100  0
snd  0.00 A  0    0    201
rcv  0.00 D  201  100  0
snd  0.00 A  0    0    701
rcv  0.00 D  701  100  0
snd  0.00 A  0    0    801
rcv  0.00 D  801  100  0
snd  0.00 A  0    0    901
rcv  0.00 D  901  100  0
snd  0.00 A  0    0    1001
rcv  0.00 D  1001      100  0
snd  0.00 A  0    0    1101
rcv  0.00 D  1101      100  0
snd  0.00 A  0    0    1201
rcv  0.00 D  1201      100  0
snd  0.00 A  0    0    1301
rcv  0.00 D  1301      100  0
snd  0.00 A  0    0    1401
rcv  0.00 D  1401      100  0
snd  0.00 A  0    0    1501
rcv  0.00 D  1501      100  0
snd  0.00 A  0    0    1601
rcv  0.00 D  1601      100  0
snd  0.00 A  0    0    1701
rcv  0.00 D  1701      100  0
```

```
snd  0.00 A     0     0     1801
rcv  0.00 D     1801        100  0
snd  0.01 A     0     0     1901
rcv  0.01 D     1901        100  0
snd  0.01 A     0     0     2001
rcv  0.01 D     2101        100  0
snd  0.01 A     0     0     2001
rcv  0.01 D     2201        100  0
snd  0.01 A     0     0     2001
rcv  0.01 D     2301        100  0
snd  0.01 A     0     0     2001
rcv  0.01 D     2401        100  0
snd  0.01 A     0     0     2001
rcv  0.01 D     2001        100  0
snd  0.01 A     0     0     2501
rcv  0.01 D     2501        100  0
snd  0.01 A     0     0     2601
rcv  0.01 D     2601        100  0
snd  0.01 A     0     0     2701
rcv  0.01 D     2901        100  0
snd  0.01 A     0     0     2701
rcv  0.01 D     3001        28   0
snd  0.01 A     0     0     2701
rcv  1.25 D     2701        100  0
snd  1.25 A     0     0     2801
rcv  2.38 D     2801        100  0
snd  2.38 A     0     0     3029
rcv  2.38 F     3029        0    0
snd  2.38 A     0     0     3030
snd  2.38 F     3029        0    0
rcv  2.38 A     0     0     3030
============================================================
==
Amount of data received (bytes) 3028
Total Segments Received     35
Data segments received      31
Data segments with Bit Errors     0
Duplicate data segments received      0
Duplicate ACKs sent     10
============================================================
==
```

pdrop = 0.3

```
rcv  0.00 S     0     0     0
snd  0.00 S     0     0     1
rcv  0.00 D     1     0     1
rcv  0.00 D     101  100  0
```

```
snd  0.00 A    0    0    1
rcv  0.00 D    201  100  0
snd  0.00 A    0    0    1
rcv  0.00 D    301  100  0
snd  0.00 A    0    0    1
rcv  5.51 D    1    100  0
snd  5.51 A    0    0    401
rcv  5.51 D    501  100  0
snd  5.51 A    0    0    401
rcv  5.51 D    801  100  0
snd  5.51 A    0    0    401
rcv  7.14 D    401  100  0
snd  7.14 A    0    0    601
rcv  7.14 D    1001      100  0
snd  7.14 A    0    0    601
rcv  8.80 D    601  100  0
snd  8.80 A    0    0    701
rcv  8.80 D    1101      100  0
snd  8.80 A    0    0    701
rcv  10.43     D    701  100  0
snd  10.43     A    0    0    901
rcv  10.43     D    1201      100  0
snd  10.43     A    0    0    901
rcv  11.98     D    901  100  0
snd  11.98     A    0    0    1301
rcv  11.98     D    1601      100  0
snd  11.98     A    0    0    1301
rcv  11.98     D    1701      100  0
snd  11.99     A    0    0    1301
rcv  14.91     D    1301      100  0
snd  14.91     A    0    0    1401
rcv  14.91     D    1801      100  0
snd  14.91     A    0    0    1401
rcv  16.27     D    1401      100  0
snd  16.27     A    0    0    1501
rcv  22.46     D    1501      100  0
snd  22.46     A    0    0    1901
rcv  22.46     D    2001      100  0
snd  22.46     A    0    0    1901
rcv  22.46     D    2101      100  0
snd  22.46     A    0    0    1901
rcv  22.46     D    2201      100  0
snd  22.46     A    0    0    1901
rcv  22.46     D    2301      100  0
snd  22.46     A    0    0    1901
rcv  23.59     D    1901      100  0
snd  23.59     A    0    0    2401
rcv  23.59     D    2701      100  0
snd  23.59     A    0    0    2401
```

```
rcv  23.59    D    2801      100  0
snd  23.59    A    0    0    2401
rcv  25.63    D    2401      100  0
snd  25.63    A    0    0    2501
rcv  26.54    D    2501      100  0
snd  26.54    A    0    0    2601
rcv  26.54    D    3001      28   0
snd  26.54    A    0    0    2601
rcv  28.18    D    2601      100  0
snd  28.18    A    0    0    2901
rcv  29.64    D    2901      100  0
snd  29.64    A    0    0    3029
rcv  29.64    F    3029      0    0
snd  29.64    A    0    0    3030
snd  29.64    F    3029      0    0
rcv  29.64    A    0    0    3030
============================================================
==
Amount of data received (bytes) 3028
Total Segments Received     35
Data segments received      31
Data segments with Bit Errors     0
Duplicate data segments received      0
Duplicate ACKs sent     18
============================================================
==
```