CS3205

Assignment 2 : Report

Author Name: N K Sagar Reddy

Roll Number: CS18B029

Date : 29th March 2021

Aim

The aim of this project is to emulate and analyze the AIMD algorithm for TCP congestion

Introduction

1) Congestion: Congestion is a state occurring in network layer when the message traffic is so heavy that it slows down the network response time

2) Due to congestion, delay increases, therefore decreasing performance and retransmission can be observed.

3) So, in order to control congestion, AIMD(Additive increase Multiplicative decrease) algorithm is used.

4) The AIMD algorithm has 3 phases
    - **Slow Start Phase**- starts slowly and growth of window is exponential till threshold is reached.

    - **Congestion Avoidance Phase**-After reaching the threshold,the congestion window is incremented by 1 per every RTT.

    - **Congestion Detection Phase**-If a congestion occurs, multiplicative decrement is implemented, either by going back to the Slow Start Phase or Congestion Avoidance Phase.
      The congestion can be detected through retransmission.
      There are 2 cases possible for retransmission -

      Case 1: Retransmission due to timeout
      Congestion possibility is high (Fast retransmission)
      Here we set the threshold to half of the congestion window, reset the congestion window to its initial value and jump to the **Slow Start Phase.**

      Case 2: Retransmission due to 3 duplicate Acks
      Congestion possibility is low (Fast recovery)
      Here we set both threshold and congestion window to half of the old congestion window and jump to **Congestion Avoidance phase.**

Experiment Details

- A modified AIMD algorithm is used.
- Go Back N Sliding protocol is used but with individual acknowledgements.
- Sender's MSS is 1 KB and each segment has a fixed length of one MSS.
- RWS(Receiver Window size) is set to 1 MB and it stays constant during the emulation.The congestion window is always interpreted as a multiple of MSS (1 KB).
- The congestion threshold is always set to 50% of the current congestion window.
- The Sender always has data to send to the receiver.

Simulation Setup

- The initial Congestion window is calculated using-

$$\mathbf{CW_{new} = K_i * MSS}$$

- Then the simulation would be in the exponential phase, during which the congestion window updates/increments itself after every acknowledgement.
  The following formula is used to update the congestion window. -

$$\mathbf{CW_{new} = min(CW_{old} + K_m * MSS, RWS)}$$

- Whereas in Linear growth phase the Congestion window is updated as follows -

$$\mathbf{CW_{new} = min(CW_{old} + K_n * MSS *(\frac{MSS}{CWold}) , RWS)}$$

- When a timeout occurs the congestion window gets reduced as follows-

$$\mathbf{CW_{new} = max(1, K_f * CW_{old})}$$

Entities/Parameters involved

The following are the parameters we take as input before we run the simulation.

1. **K_i -** This is used to calculate the initial congestion window as shown above.

$$1 \leq K_i \leq 4$$

2. **K_m -** This denotes the multiplier of the congestion window.(during exponential growth)

$$0.5 \leq K_m \leq 2$$

3. **K_n** - This denotes multiplier of the congestion window(during linear growth)

$$0.5 \leq K_n \leq 2$$

4. **K_f** - This denotes the multiplier of the congestion window (when a timeout occurs).

$$0.1 \leq K_f \leq 0.5$$

5. **P_s** - This is the probability for the occurrence of a timeout.

$$0.0001 \leq P_s \leq 0.01$$

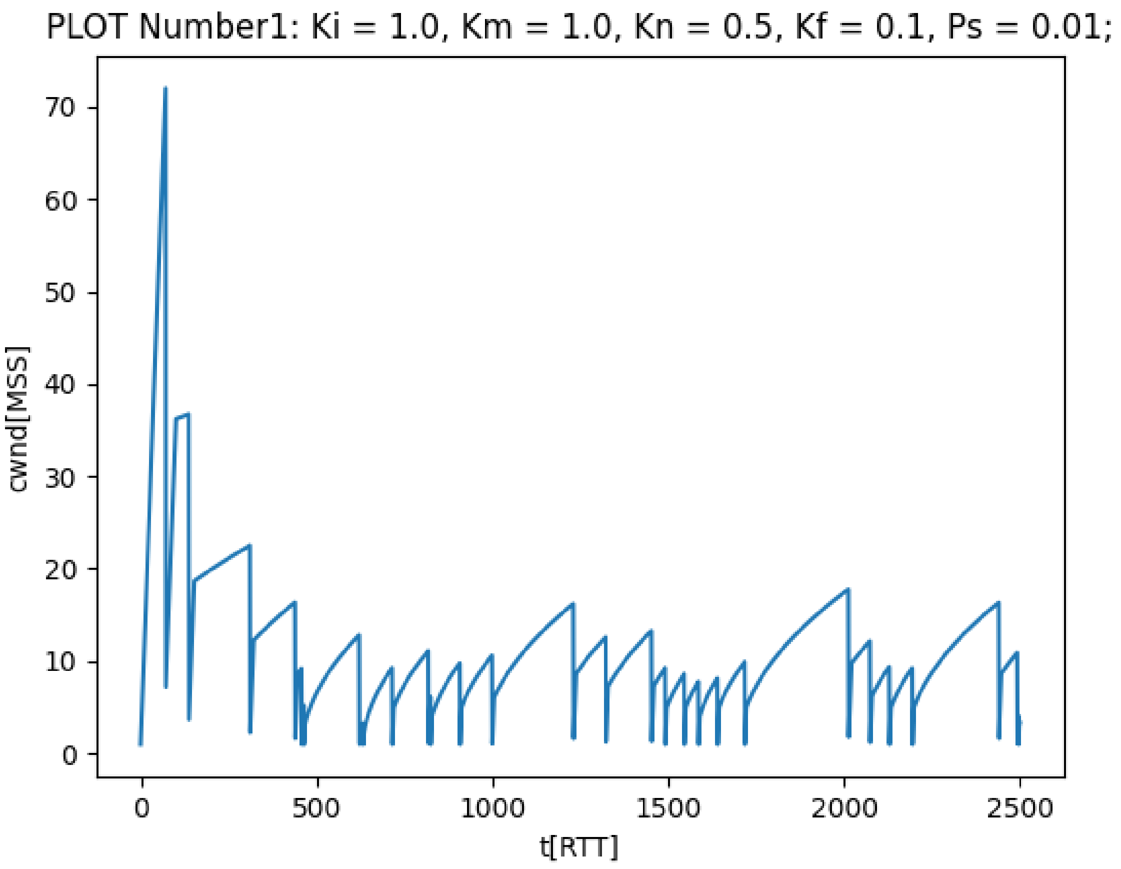
6. **T** - This denotes the number of segments to be processed.

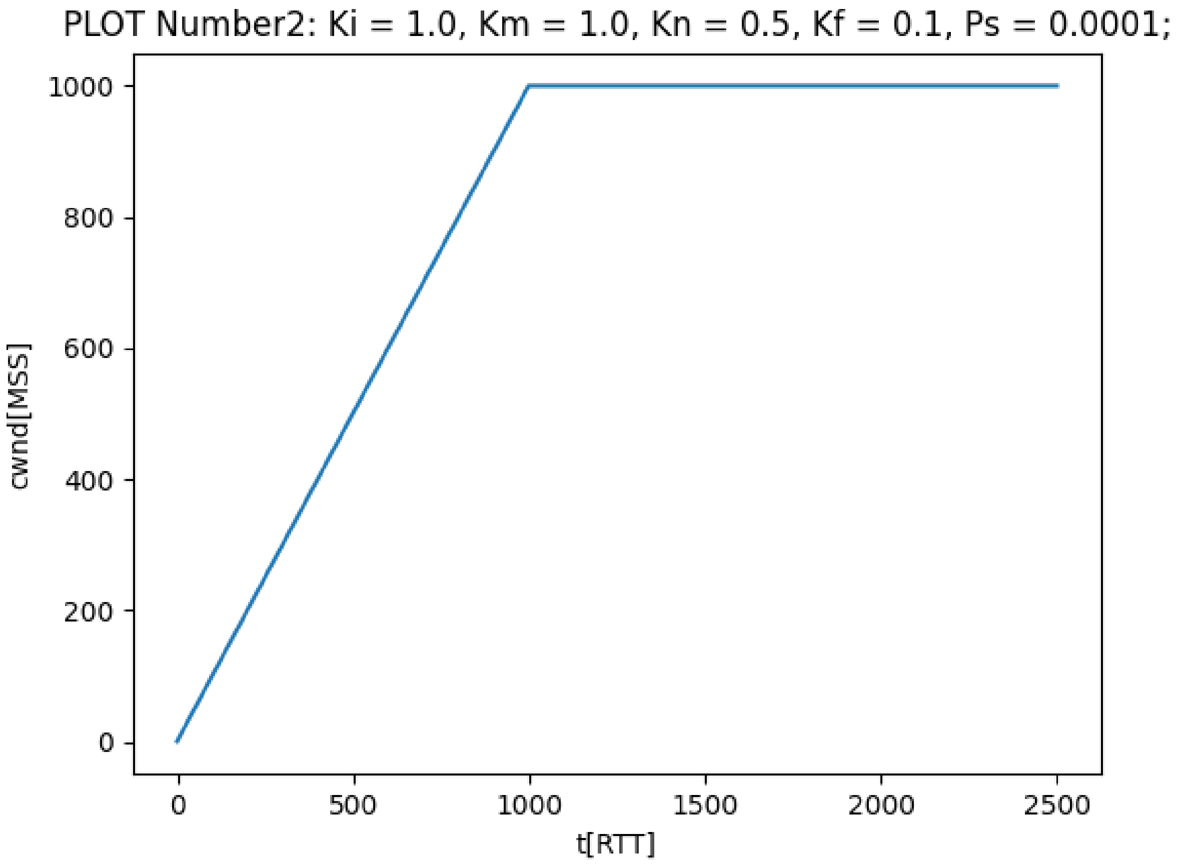
Additional details

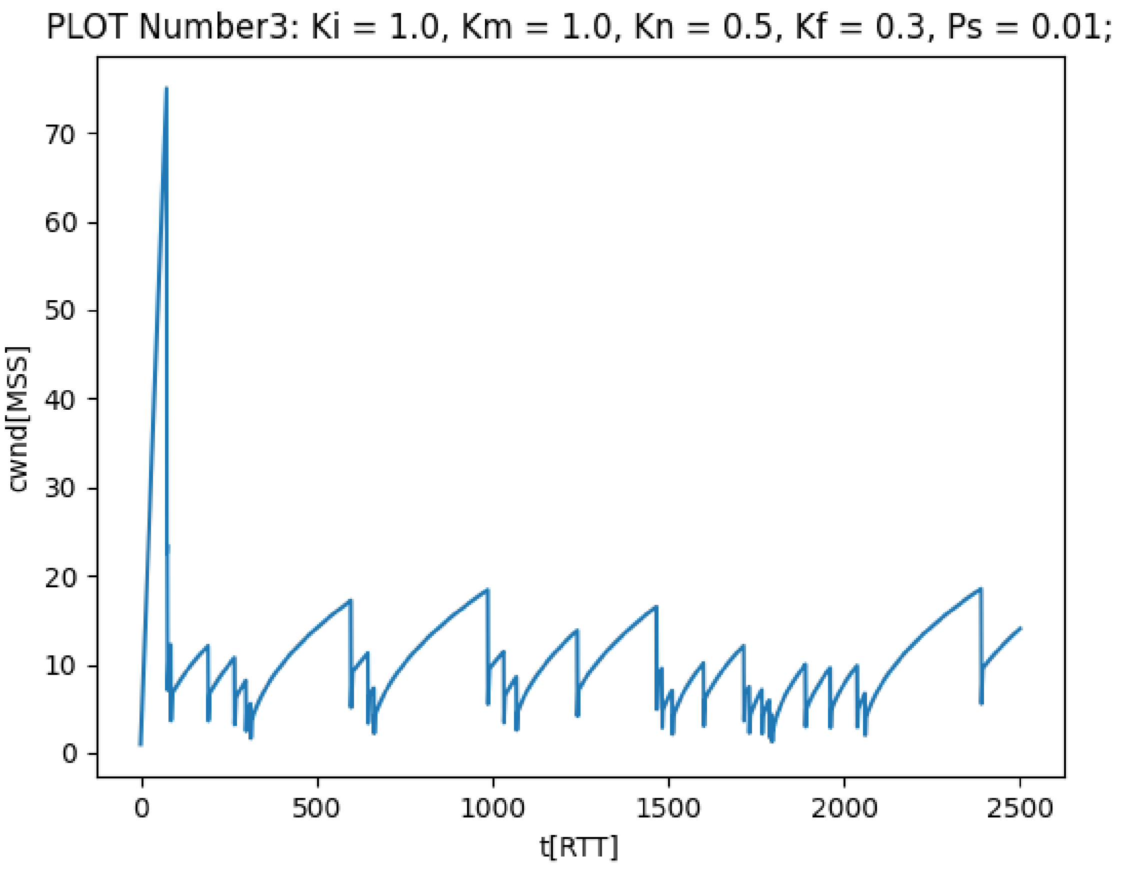
The simulations have been run using T = 2500 and by varying one/more of the first 5 parameters.

The variations are nothing but the endpoints of the given range of every parameter.

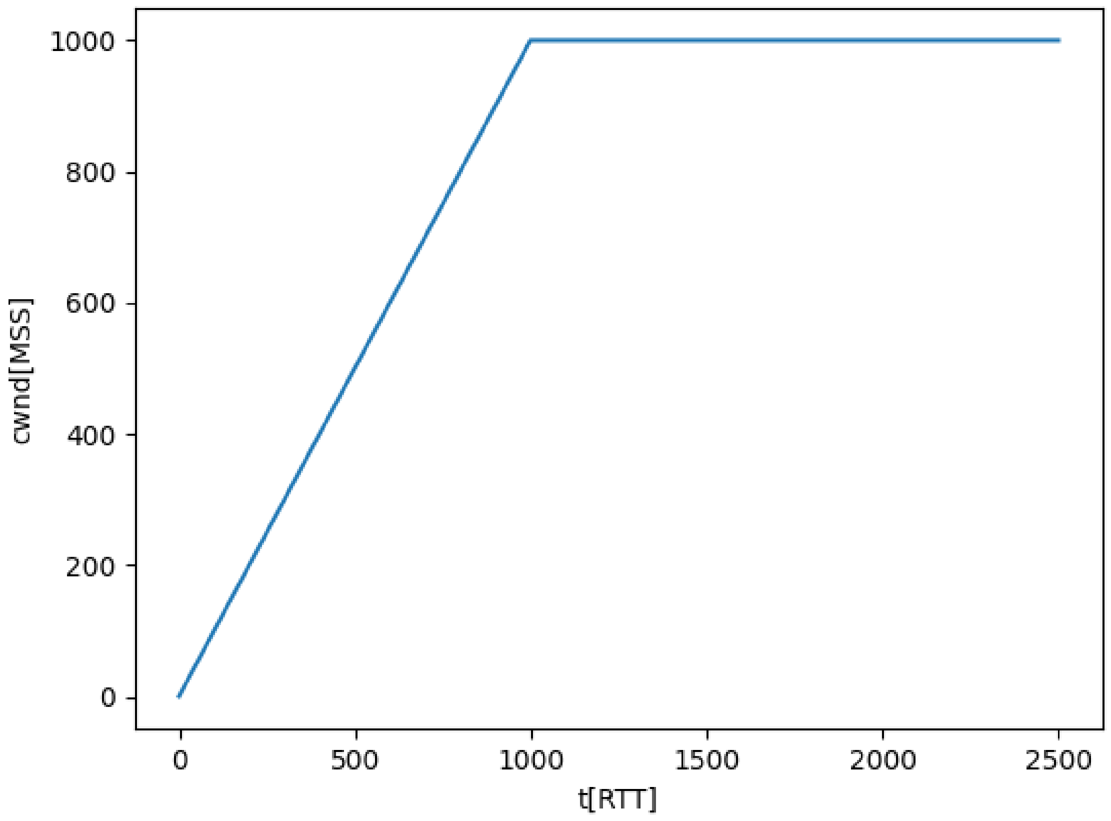
Results and Observations/Plots

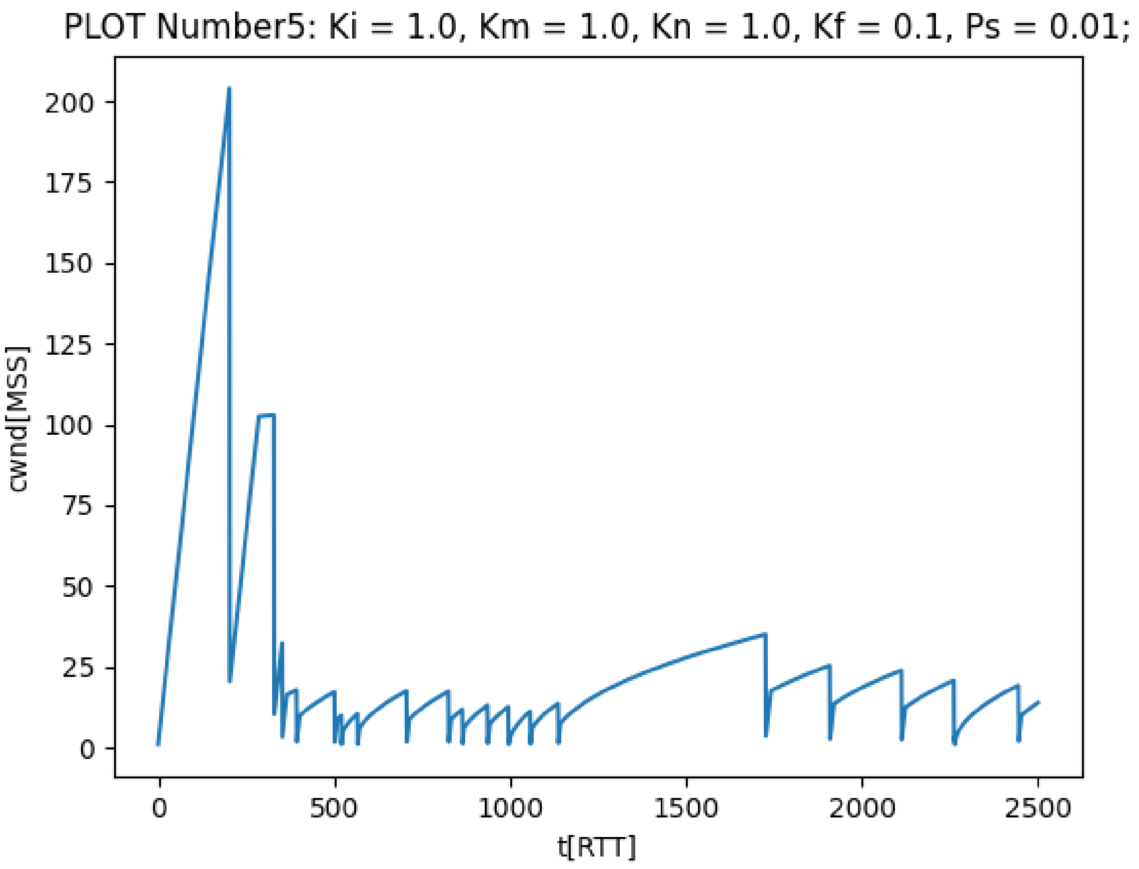

PLOT Number1: Ki = 1.0, Km = 1.0, Kn = 0.5, Kf = 0.1, Ps = 0.01;
cwnd[MSS]
t[RTT]
0
10
20
30
40
50
60
70
0
500
1000
1500
2000
2500

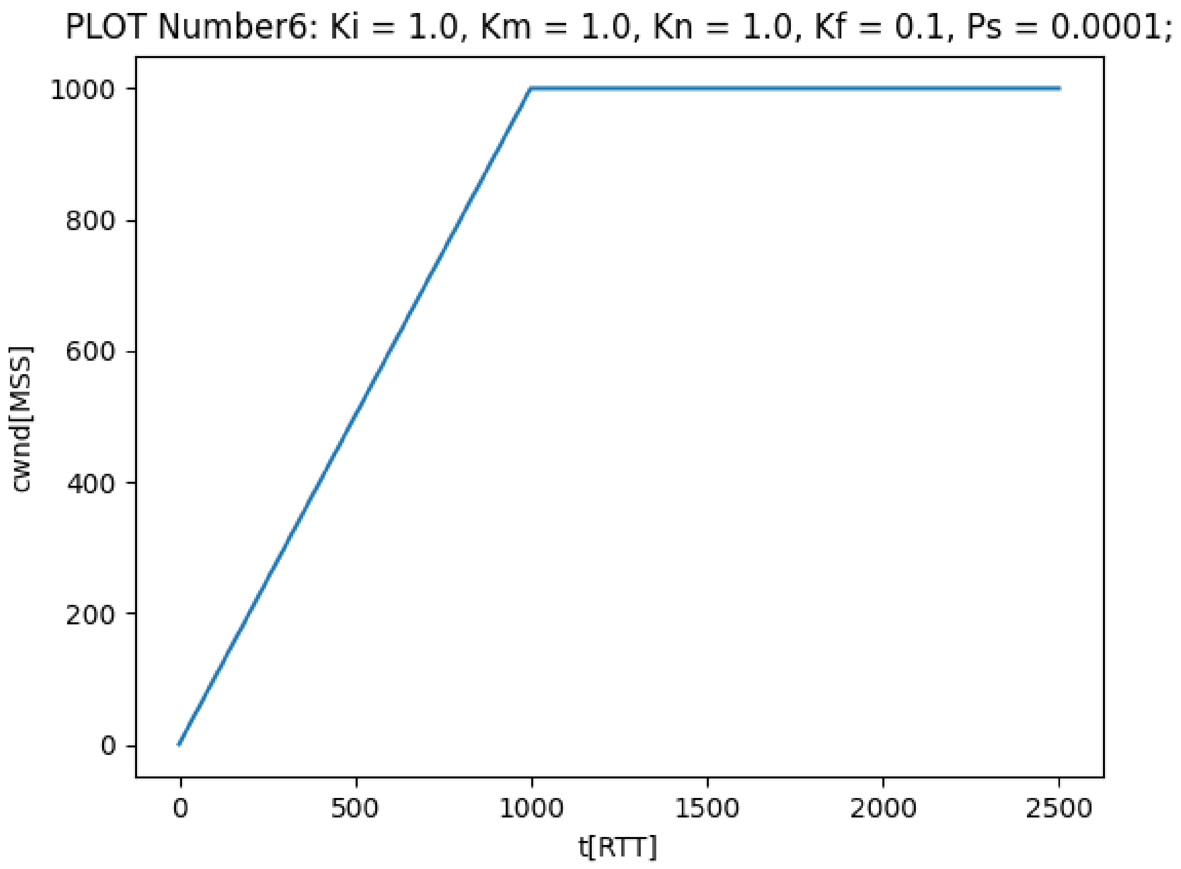

PLOT Number2: Ki = 1.0, Km = 1.0, Kn = 0.5, Kf = 0.1, Ps = 0.0001;
cwnd[MSS]
t[RTT]
0
200
400
600
800
1000
0
500
1000
1500
2000
2500

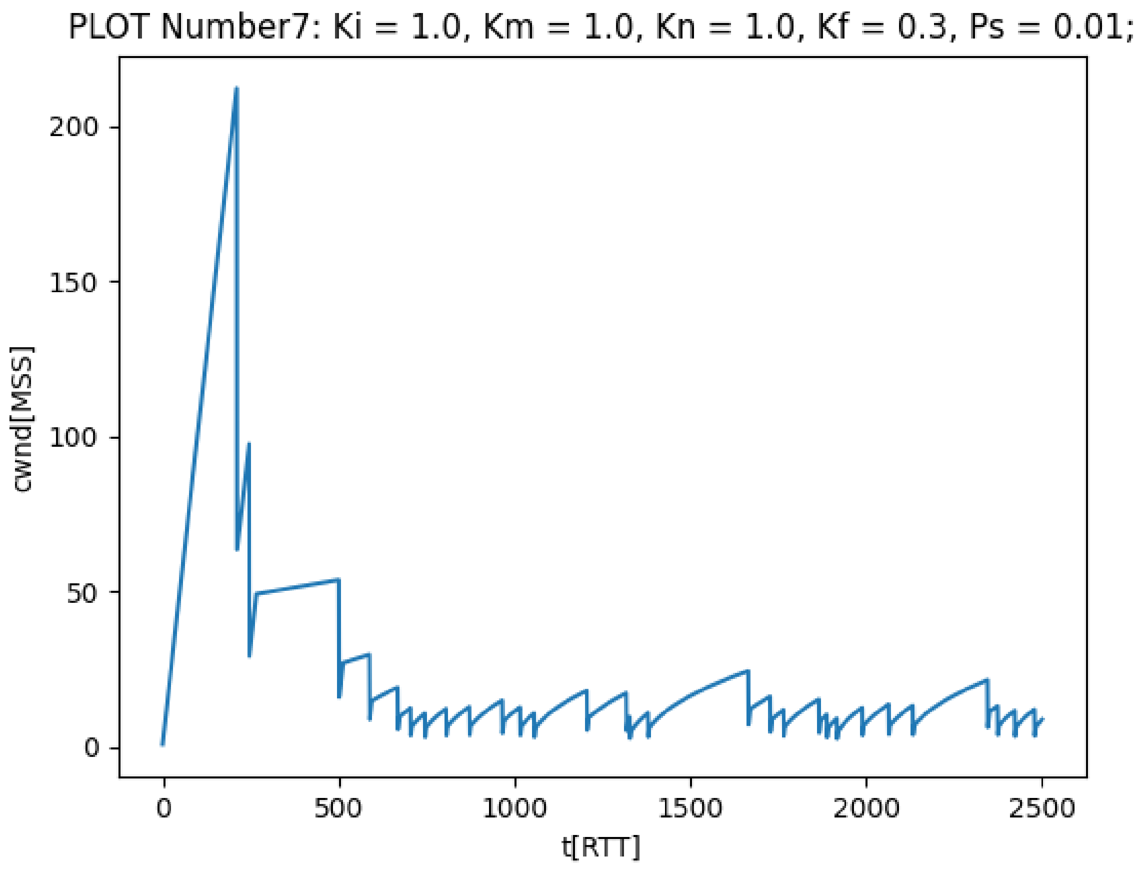

PLOT Number3: Ki = 1.0, Km = 1.0, Kn = 0.5, Kf = 0.3, Ps = 0.01;
cwnd[MSS]
t[RTT]
0
10
20
30
40
50
60
70
0
500
1000
1500
2000
2500

PLOT Number4: Ki = 1.0, Km = 1.0, Kn = 0.5, Kf = 0.3, Ps = 0.0001;

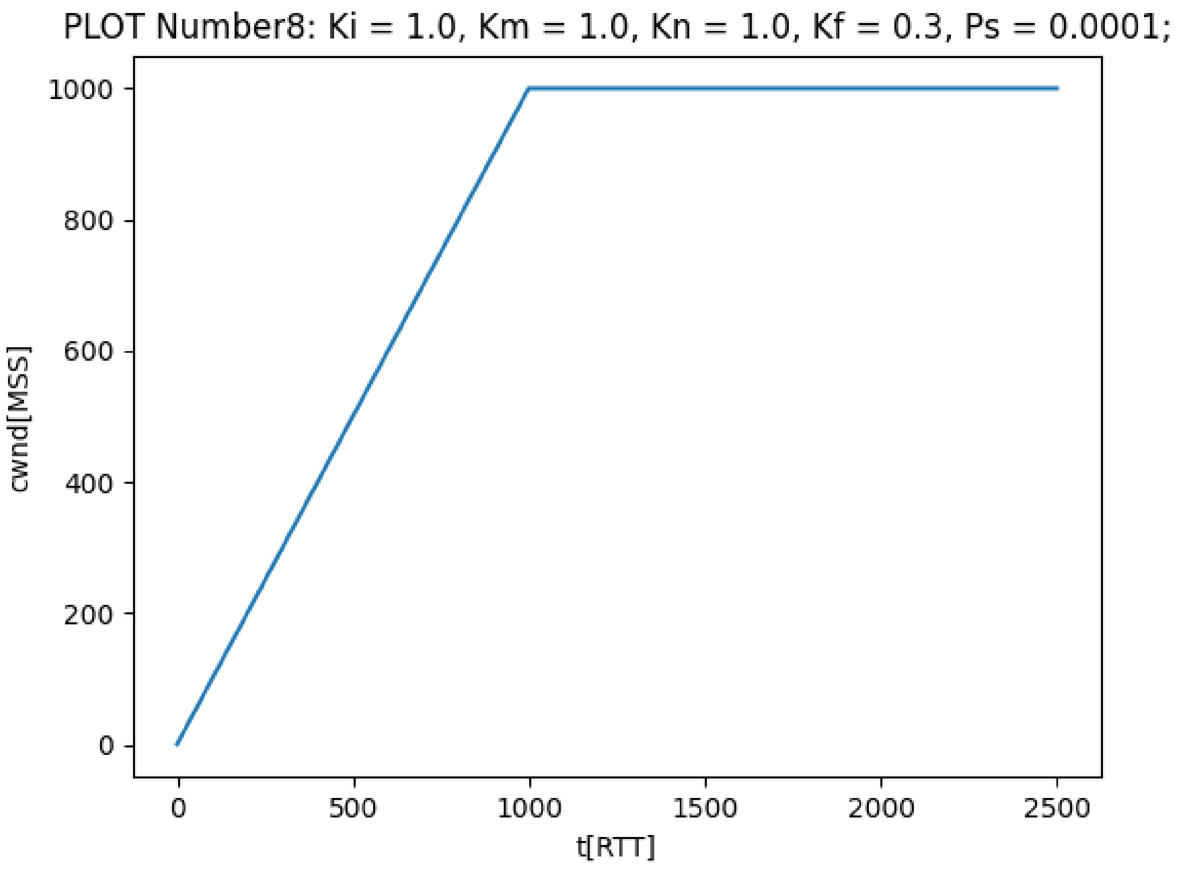


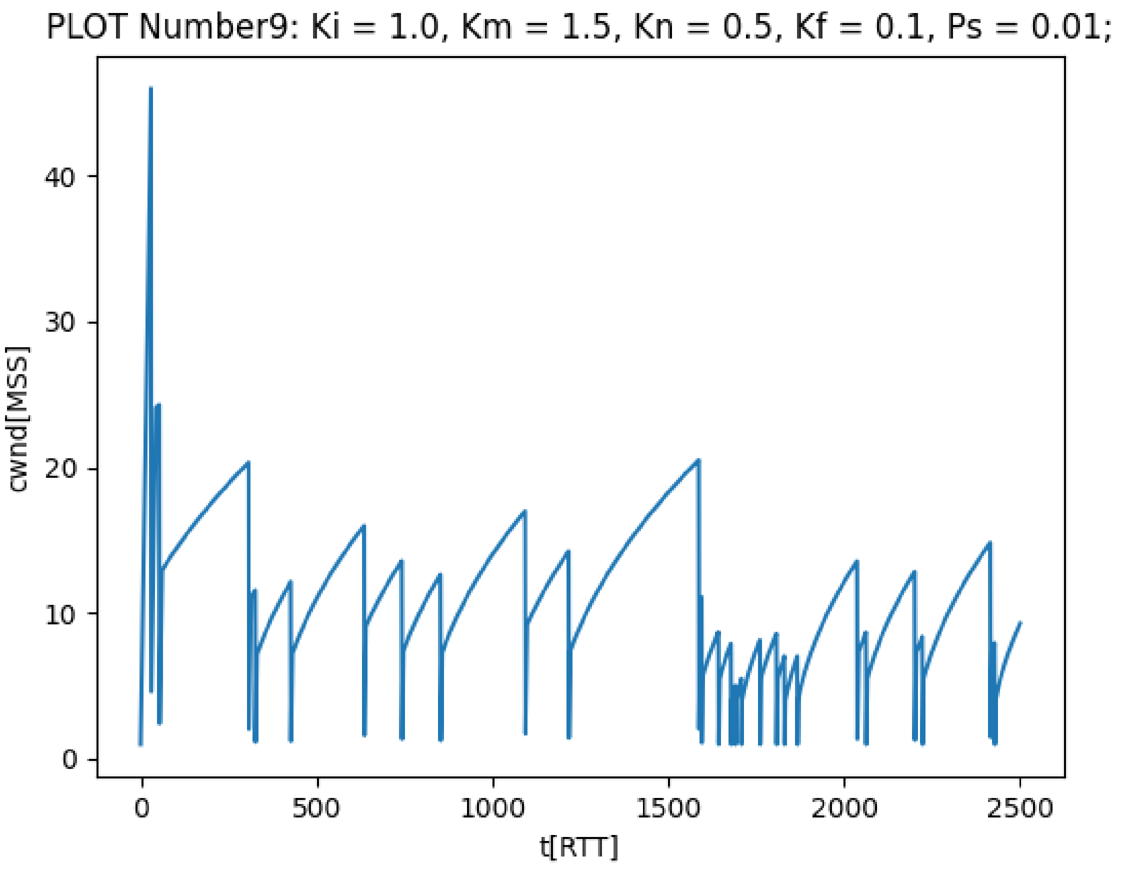

PLOT Number5: Ki = 1.0, Km = 1.0, Kn = 1.0, Kf = 0.1, Ps = 0.01;
cwnd[MSS]
t[RTT]
0
25
50
75
100
125
150
175
200
0
500
1000
1500
2000
2500


PLOT Number6: Ki = 1.0, Km = 1.0, Kn = 1.0, Kf = 0.1, Ps = 0.0001;
cwnd[MSS]
t[RTT]
0
200
400
600
800
1000
0
500
1000
1500
2000
2500

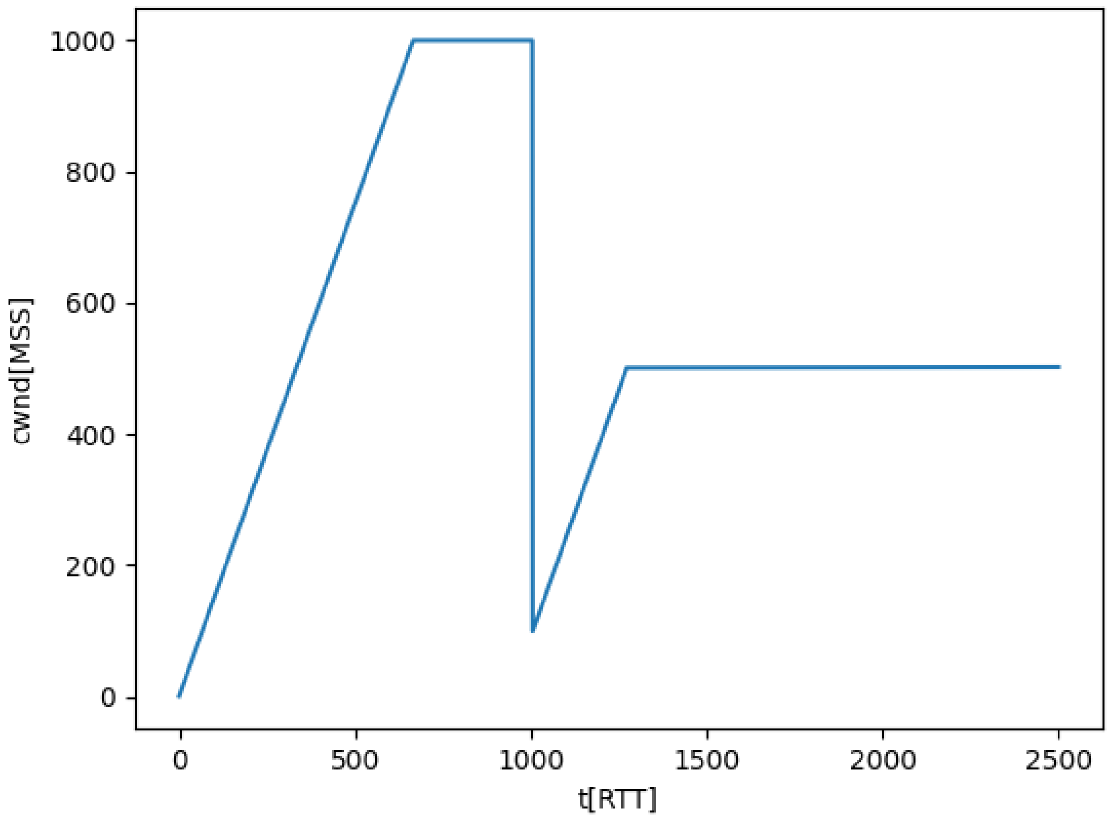


PLOT Number7: Ki = 1.0, Km = 1.0, Kn = 1.0, Kf = 0.3, Ps = 0.01;
cwnd[MSS]
t[RTT]
0
50
100
150
200
0
500
1000
1500
2000
2500

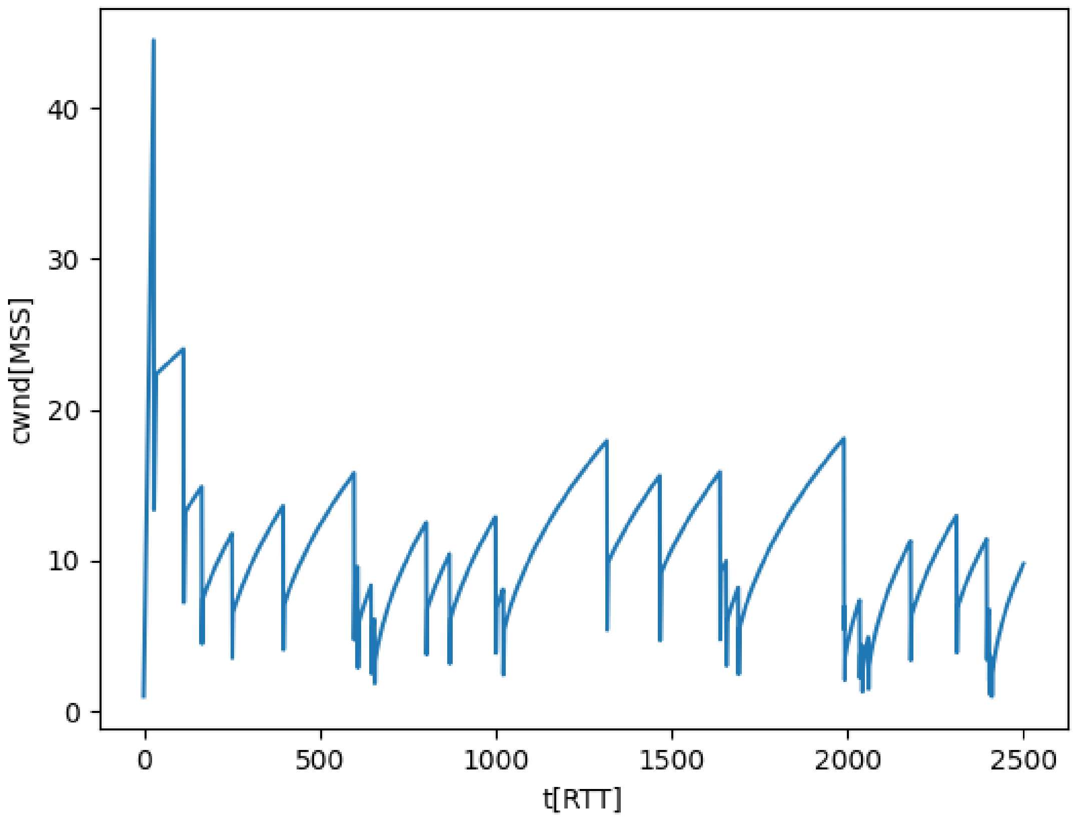

PLOT Number8: Ki = 1.0, Km = 1.0, Kn = 1.0, Kf = 0.3, Ps = 0.0001;
cwnd[MSS]
t[RTT]
0
200
400
600
800
1000
0
500
1000
1500
2000
2500


PLOT Number9: Ki = 1.0, Km = 1.5, Kn = 0.5, Kf = 0.1, Ps = 0.01;
cwnd[MSS]
t[RTT]
0
10
20
30
40
0
500
1000
1500
2000
2500

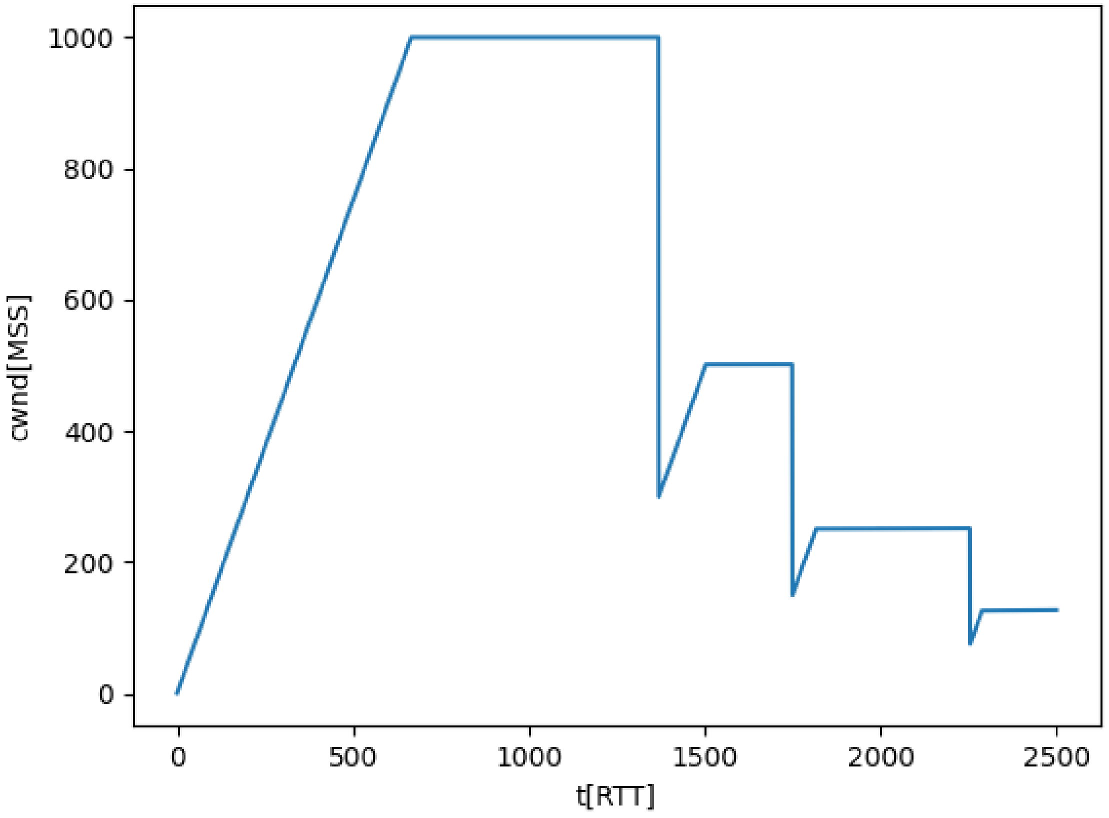
PLOT Number10: Ki = 1.0, Km = 1.5, Kn = 0.5, Kf = 0.1, Ps = 0.0001;

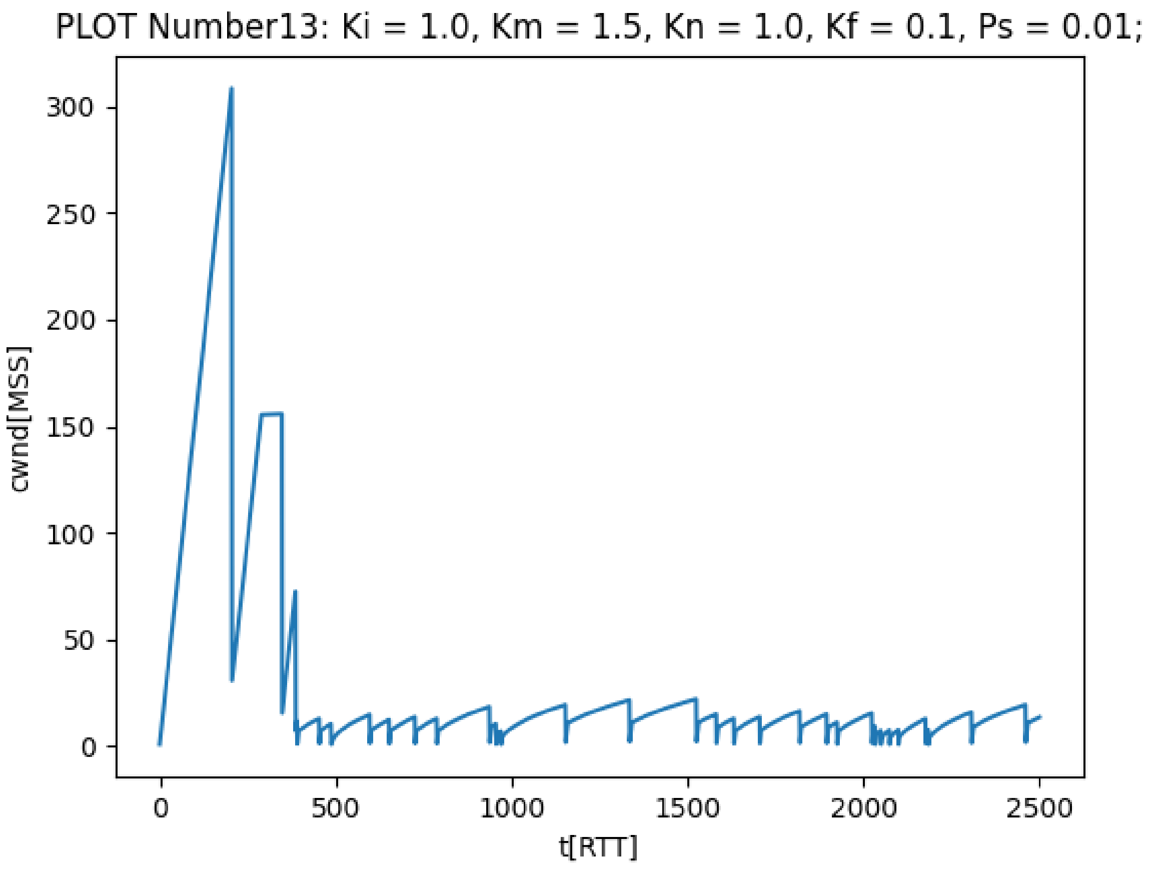


PLOT Number11: Ki = 1.0, Km = 1.5, Kn = 0.5, Kf = 0.3, Ps = 0.01;

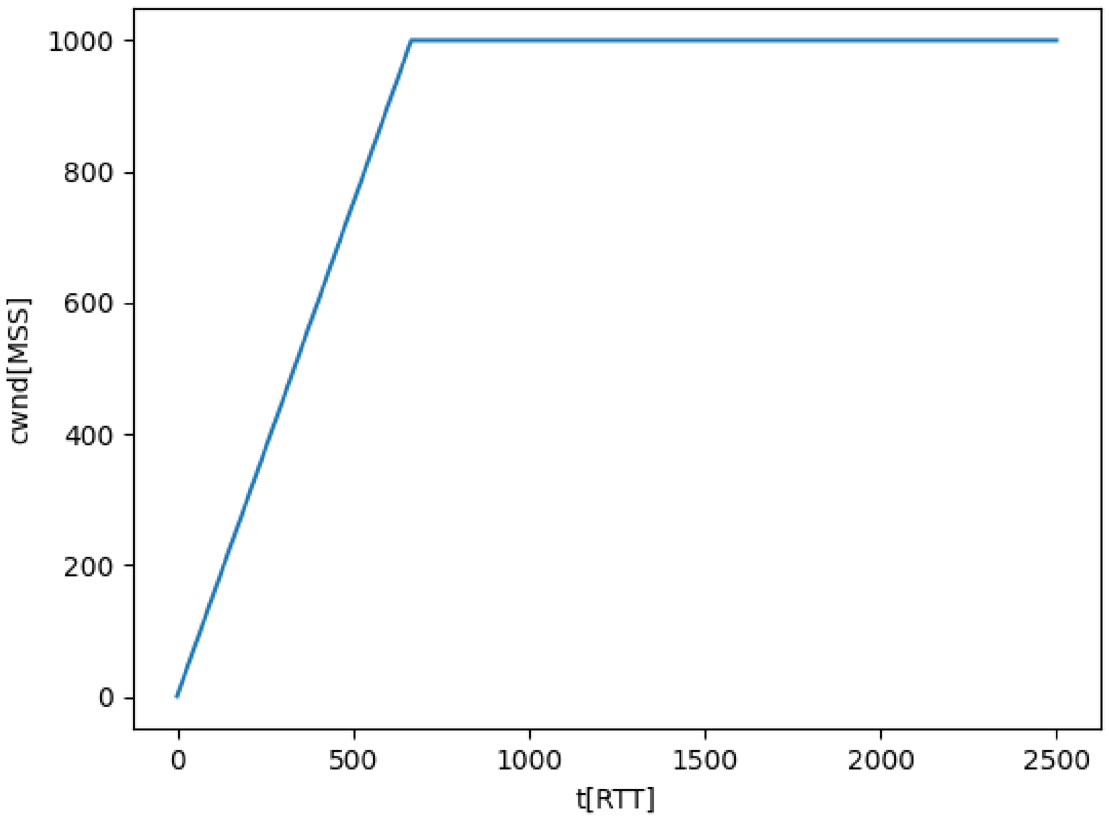


PLOT Number12: Ki = 1.0, Km = 1.5, Kn = 0.5, Kf = 0.3, Ps = 0.0001;

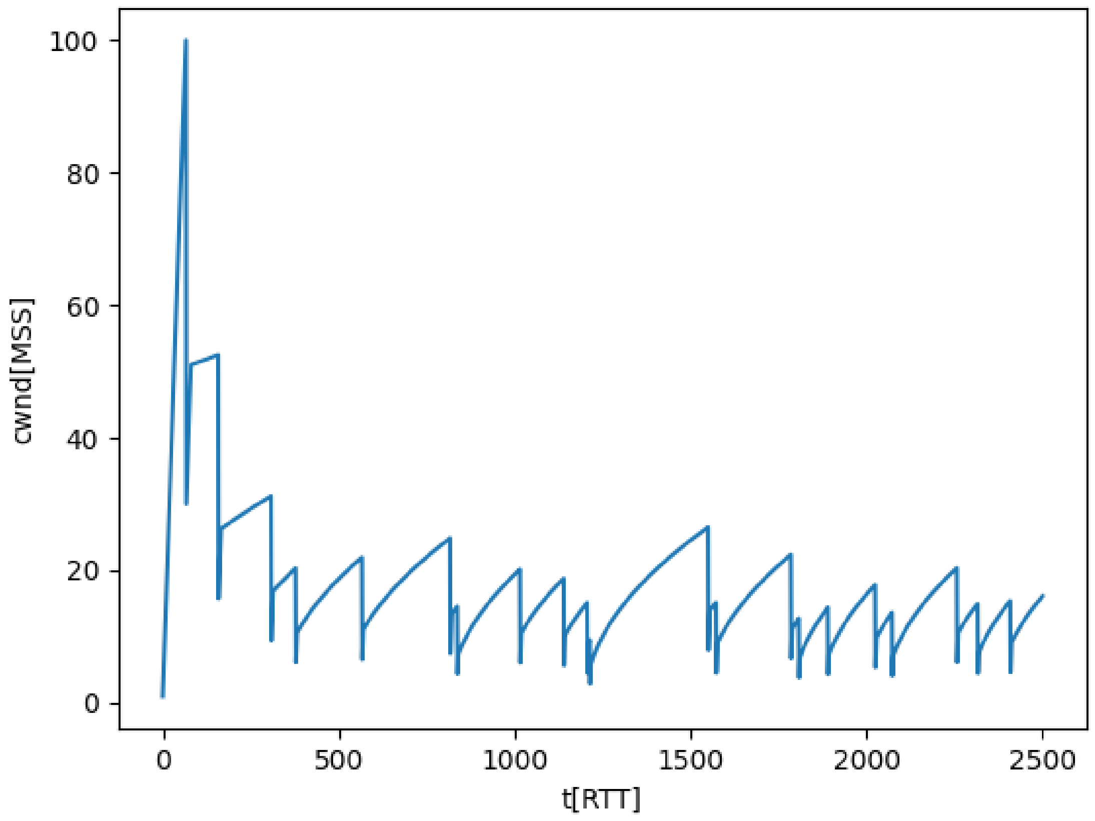



PLOT Number13: Ki = 1.0, Km = 1.5, Kn = 1.0, Kf = 0.1, Ps = 0.01;
cwnd[MSS]
t[RTT]
0
50
100
150
200
250
300
0
500
1000
1500
2000
2500

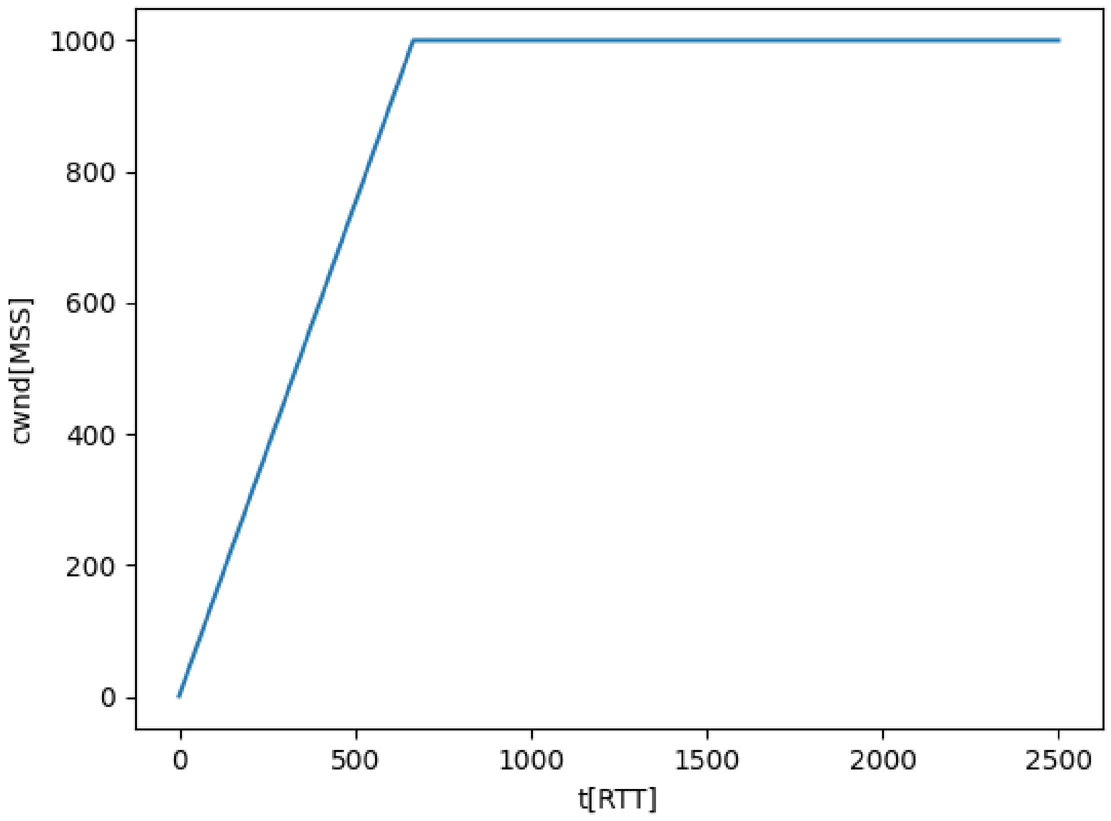
PLOT Number14: Ki = 1.0, Km = 1.5, Kn = 1.0, Kf = 0.1, Ps = 0.0001;

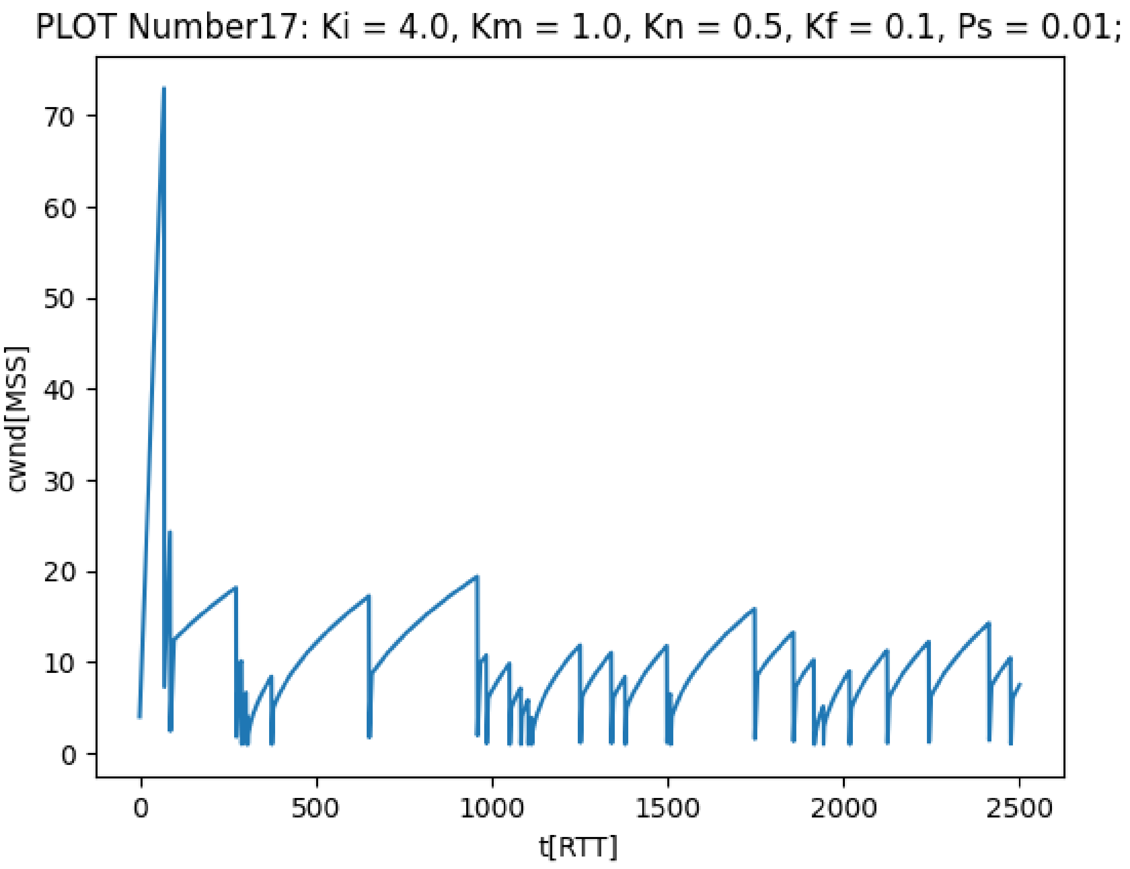


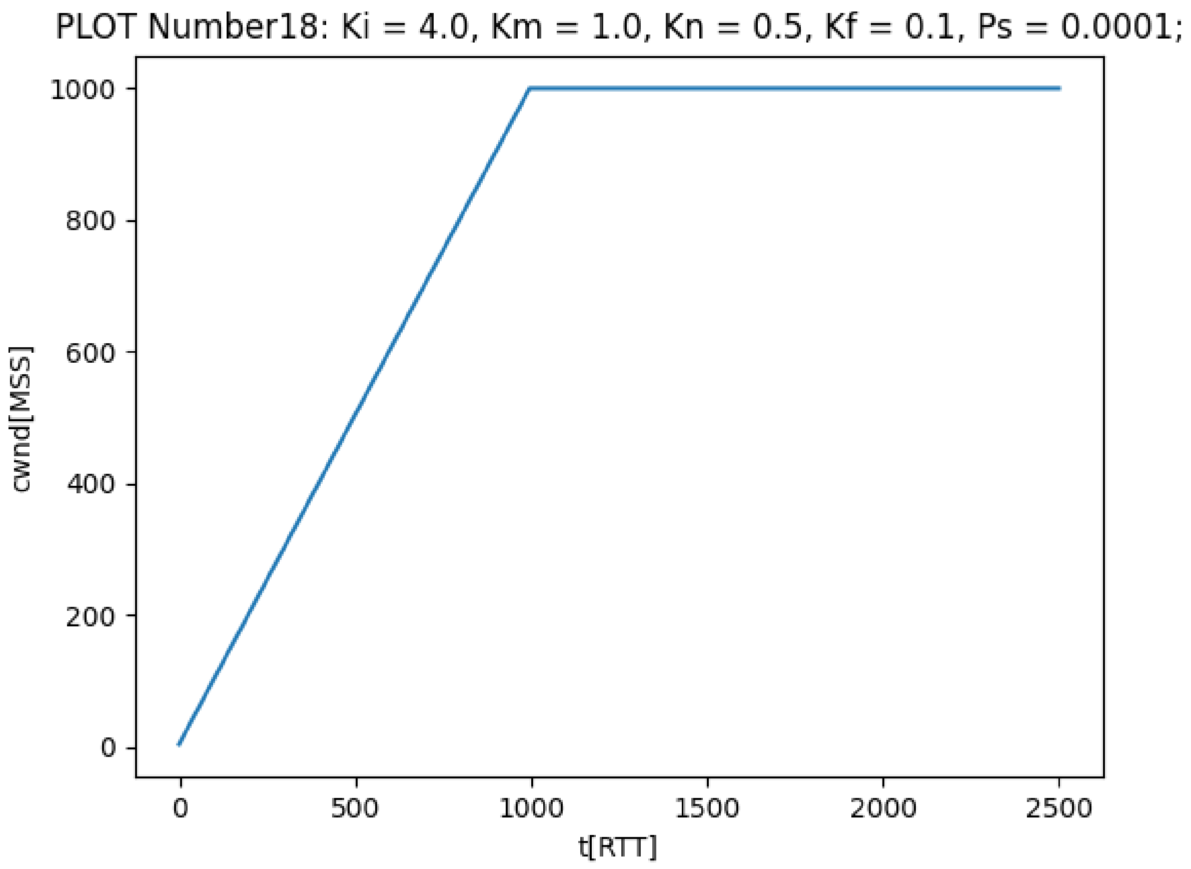
PLOT Number15: Ki = 1.0, Km = 1.5, Kn = 1.0, Kf = 0.3, Ps = 0.01;



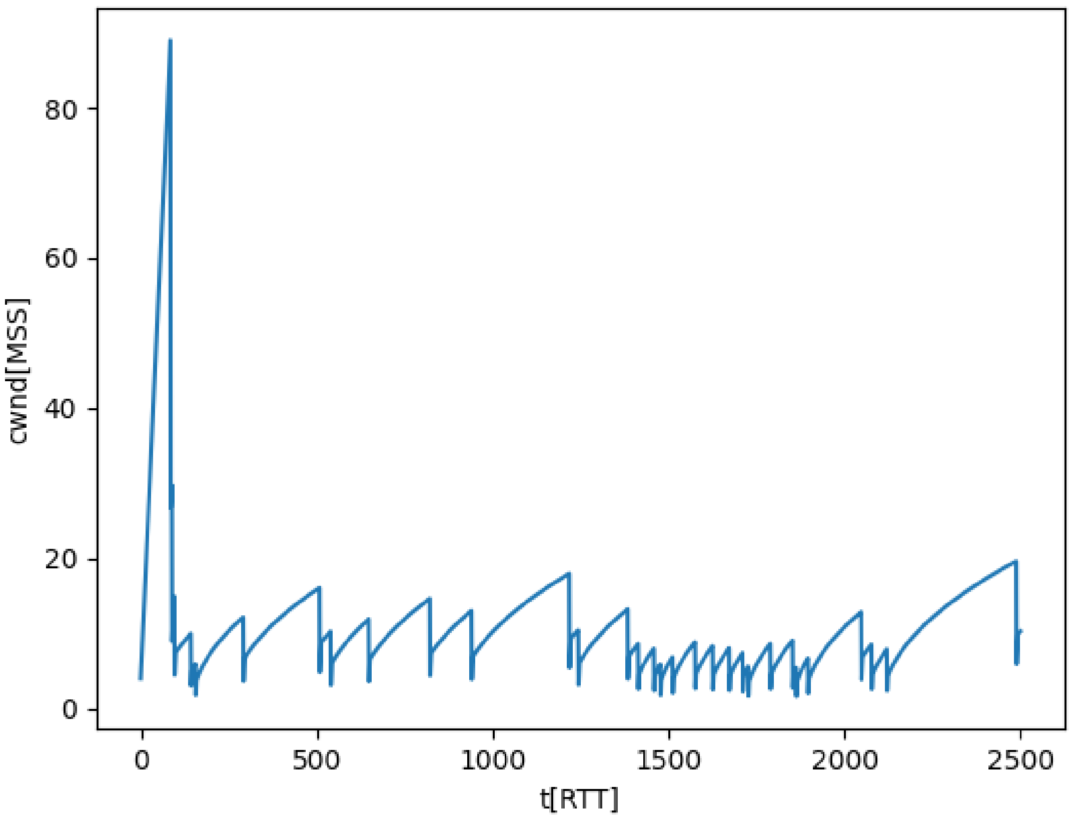
PLOT Number16: Ki = 1.0, Km = 1.5, Kn = 1.0, Kf = 0.3, Ps = 0.0001;

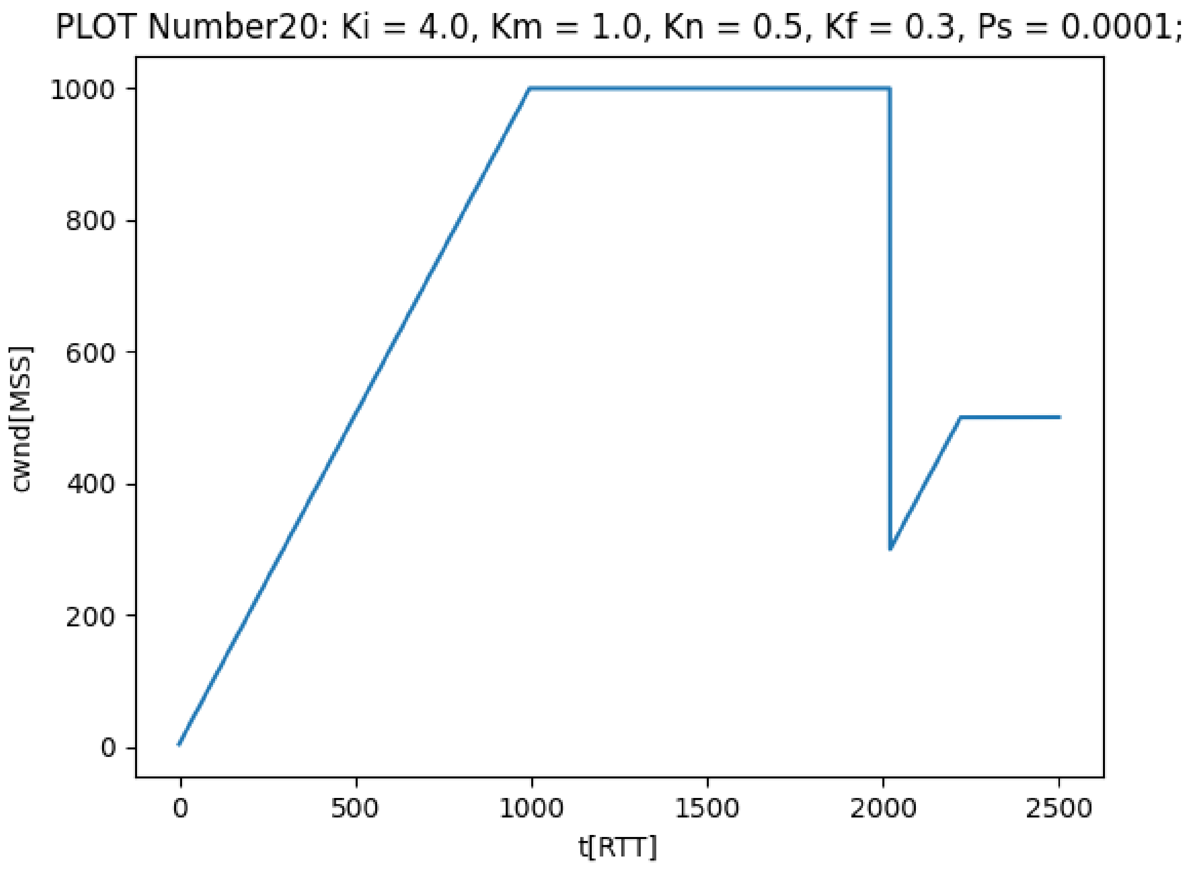



PLOT Number17: Ki = 4.0, Km = 1.0, Kn = 0.5, Kf = 0.1, Ps = 0.01;
cwnd[MSS]
t[RTT]
0
10
20
30
40
50
60
70
0
500
1000
1500
2000
2500

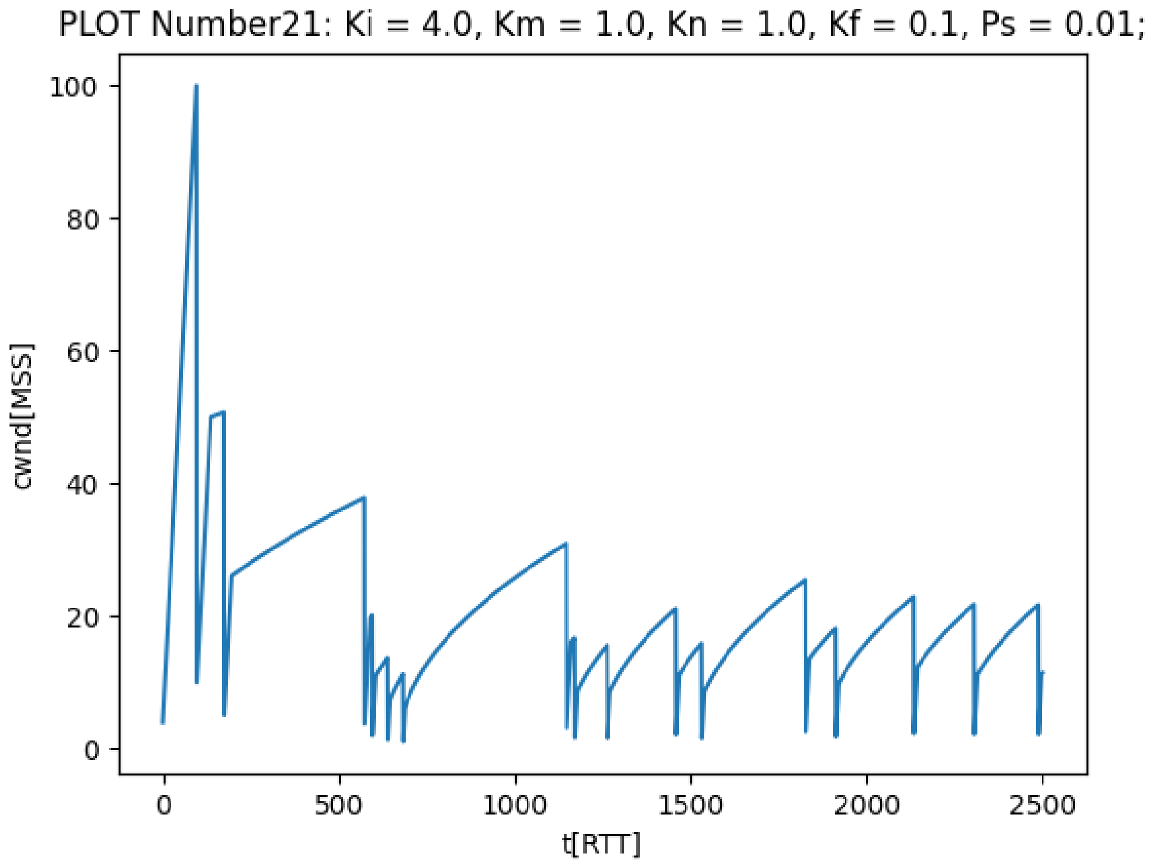

PLOT Number18: Ki = 4.0, Km = 1.0, Kn = 0.5, Kf = 0.1, Ps = 0.0001;
cwnd[MSS]
t[RTT]
0
200
400
600
800
1000
0
500
1000
1500
2000
2500

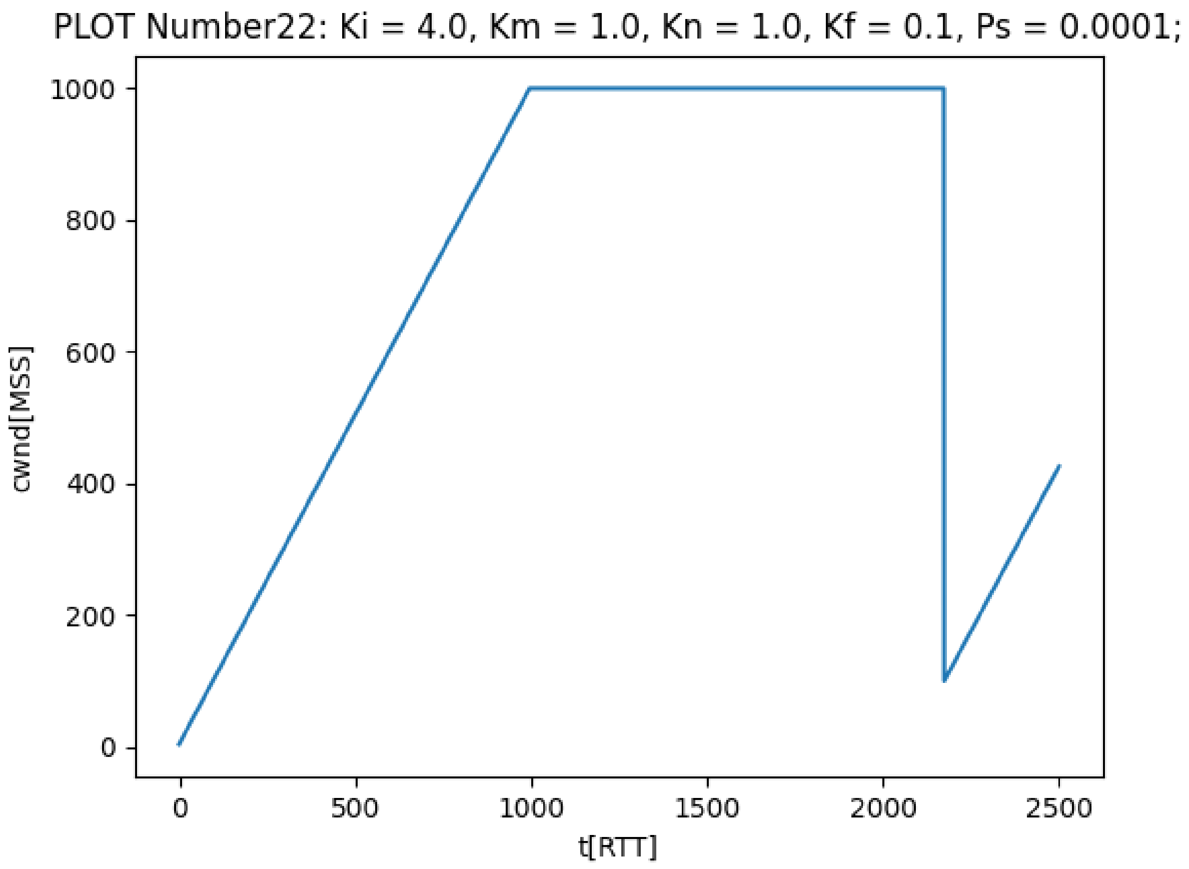
PLOT Number19: Ki = 4.0, Km = 1.0, Kn = 0.5, Kf = 0.3, Ps = 0.01;

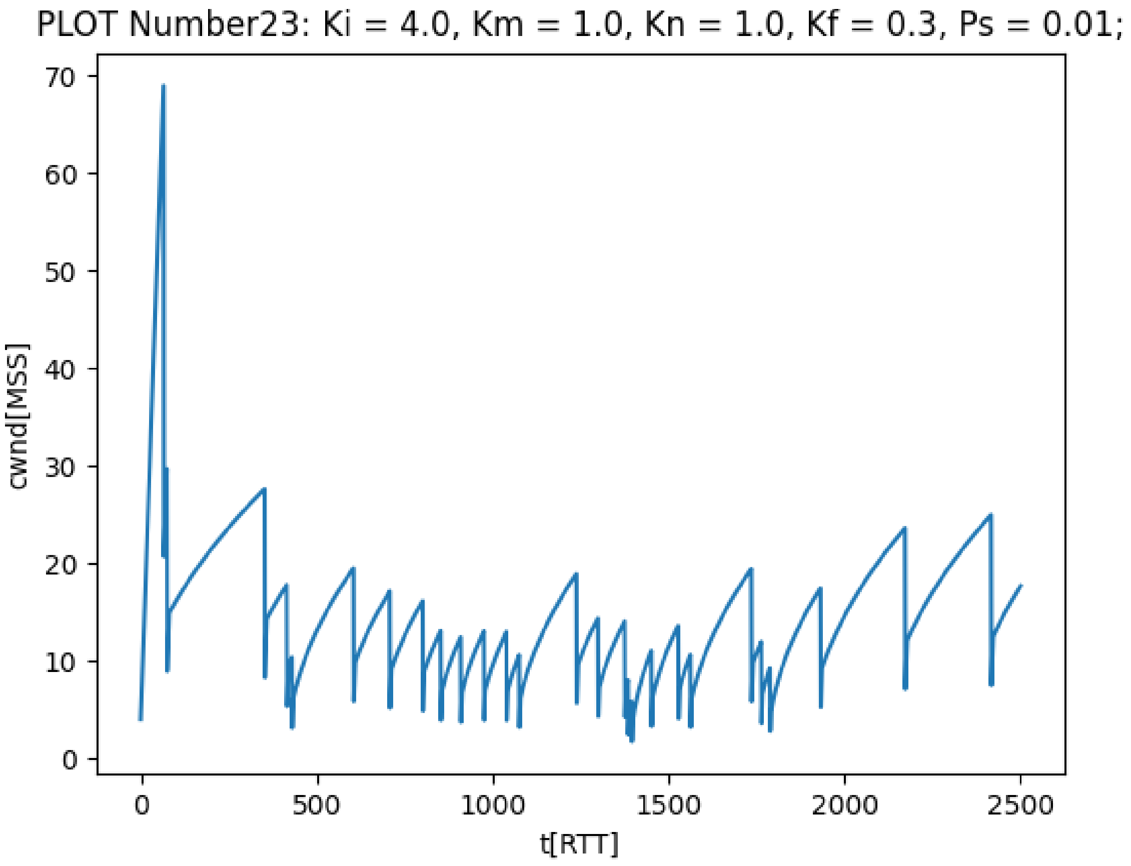


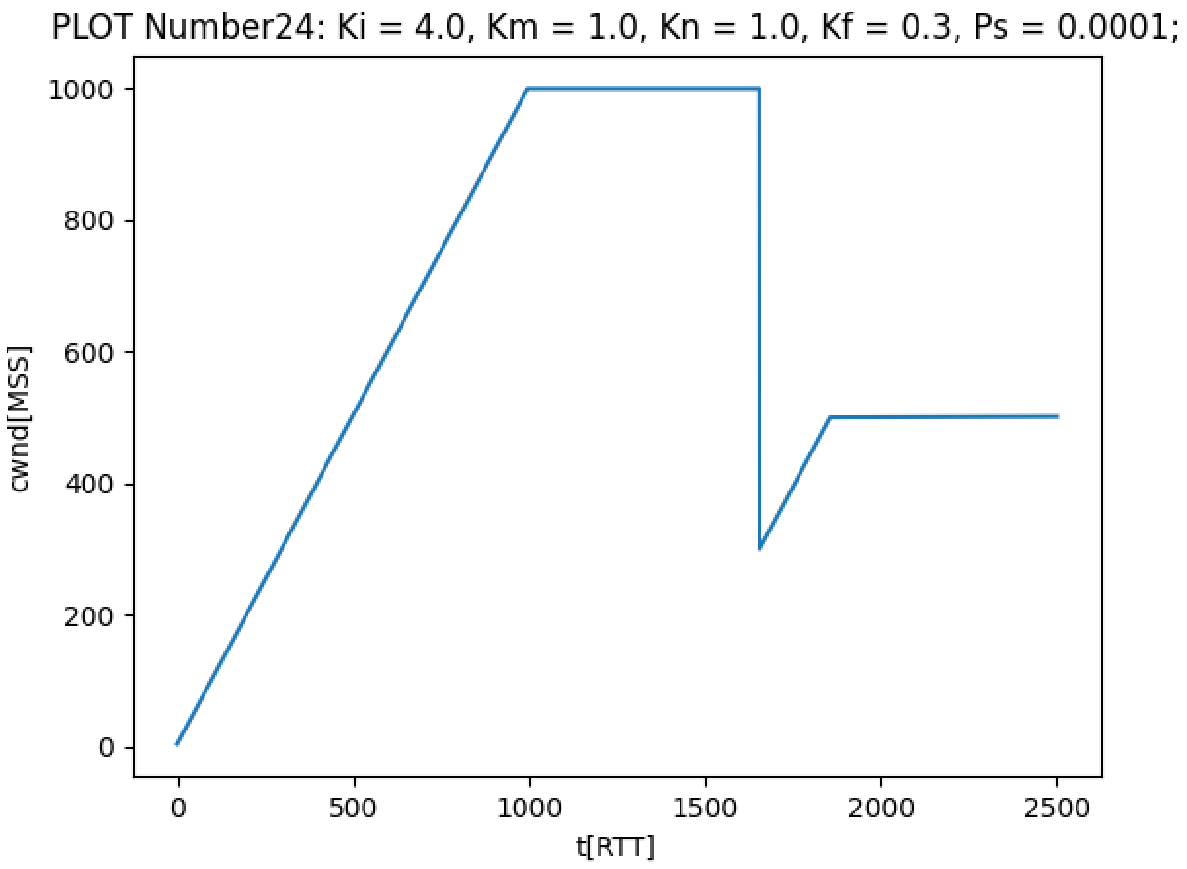


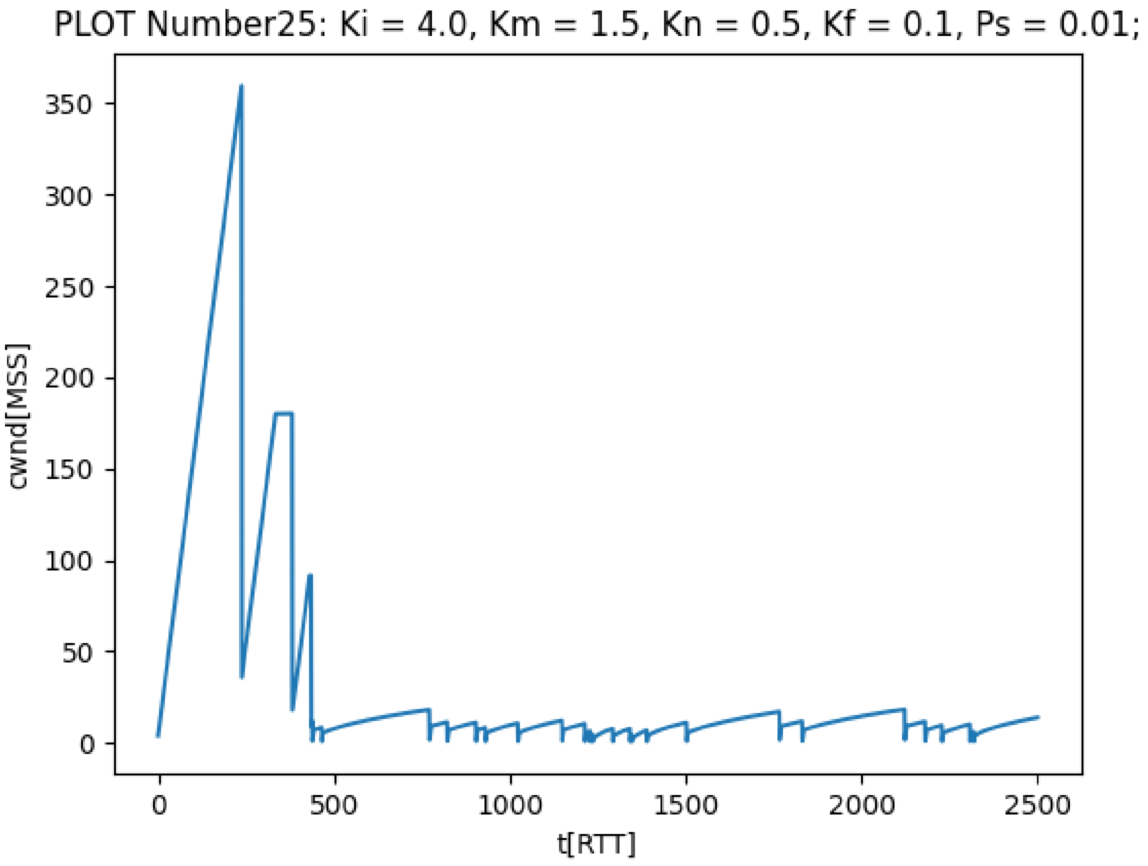
PLOT Number20: Ki = 4.0, Km = 1.0, Kn = 0.5, Kf = 0.3, Ps = 0.0001;
cwnd[MSS]
t[RTT]
0
200
400
600
800
1000
0
500
1000
1500
2000
2500


PLOT Number21: Ki = 4.0, Km = 1.0, Kn = 1.0, Kf = 0.1, Ps = 0.01;
cwnd[MSS]
t[RTT]
0
20
40
60
80
100
0
500
1000
1500
2000
2500

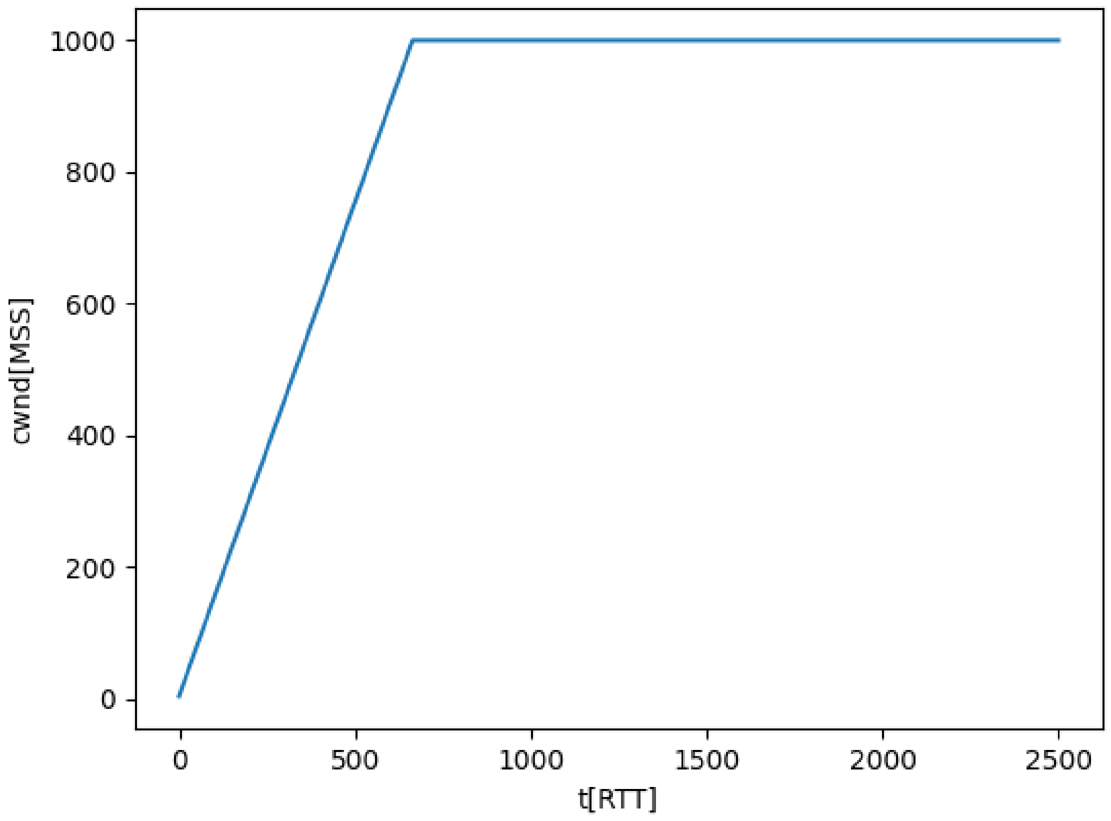

PLOT Number22: Ki = 4.0, Km = 1.0, Kn = 1.0, Kf = 0.1, Ps = 0.0001;
cwnd[MSS]
t[RTT]
0
200
400
600
800
1000
0
500
1000
1500
2000
2500

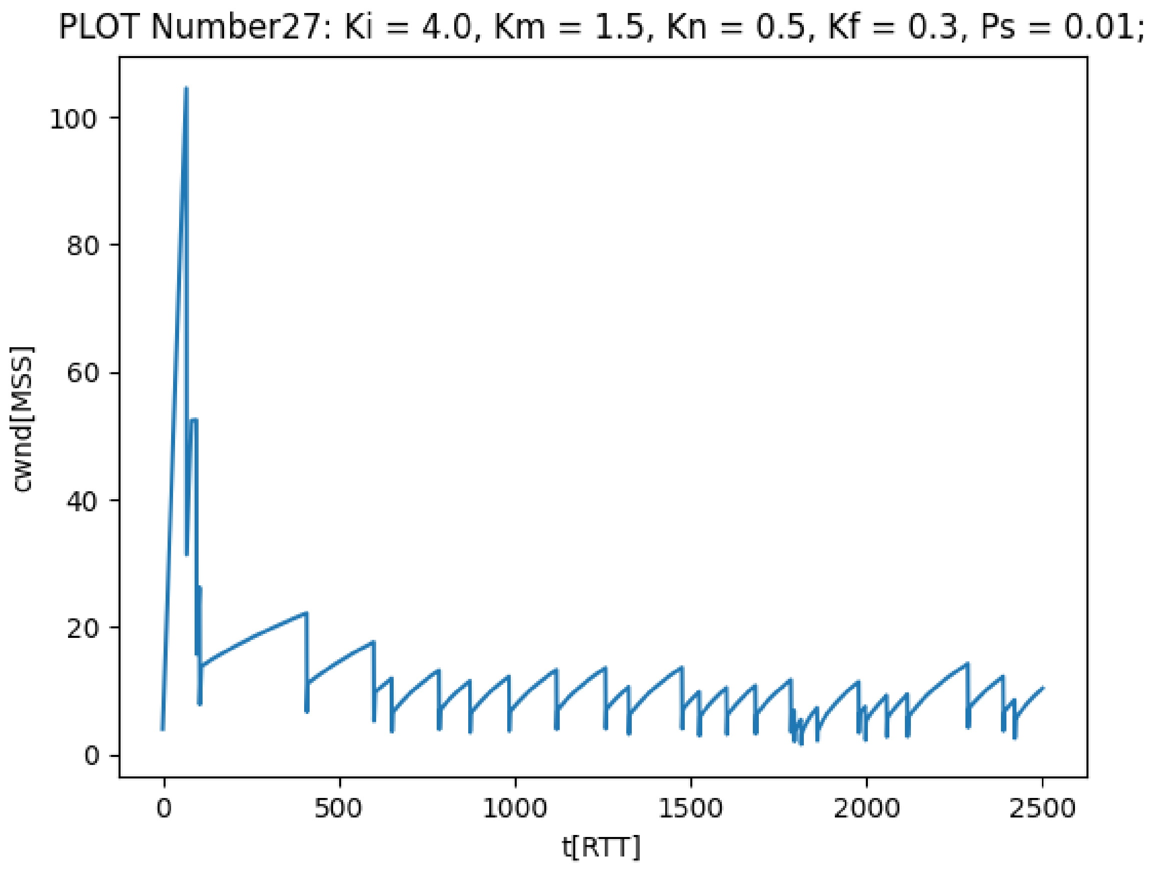

PLOT Number23: Ki = 4.0, Km = 1.0, Kn = 1.0, Kf = 0.3, Ps = 0.01;
cwnd[MSS]
t[RTT]
0
10
20
30
40
50
60
70
0
500
1000
1500
2000
2500


PLOT Number24: Ki = 4.0, Km = 1.0, Kn = 1.0, Kf = 0.3, Ps = 0.0001;
cwnd[MSS]
t[RTT]
0
200
400
600
800
1000
0
500
1000
1500
2000
2500

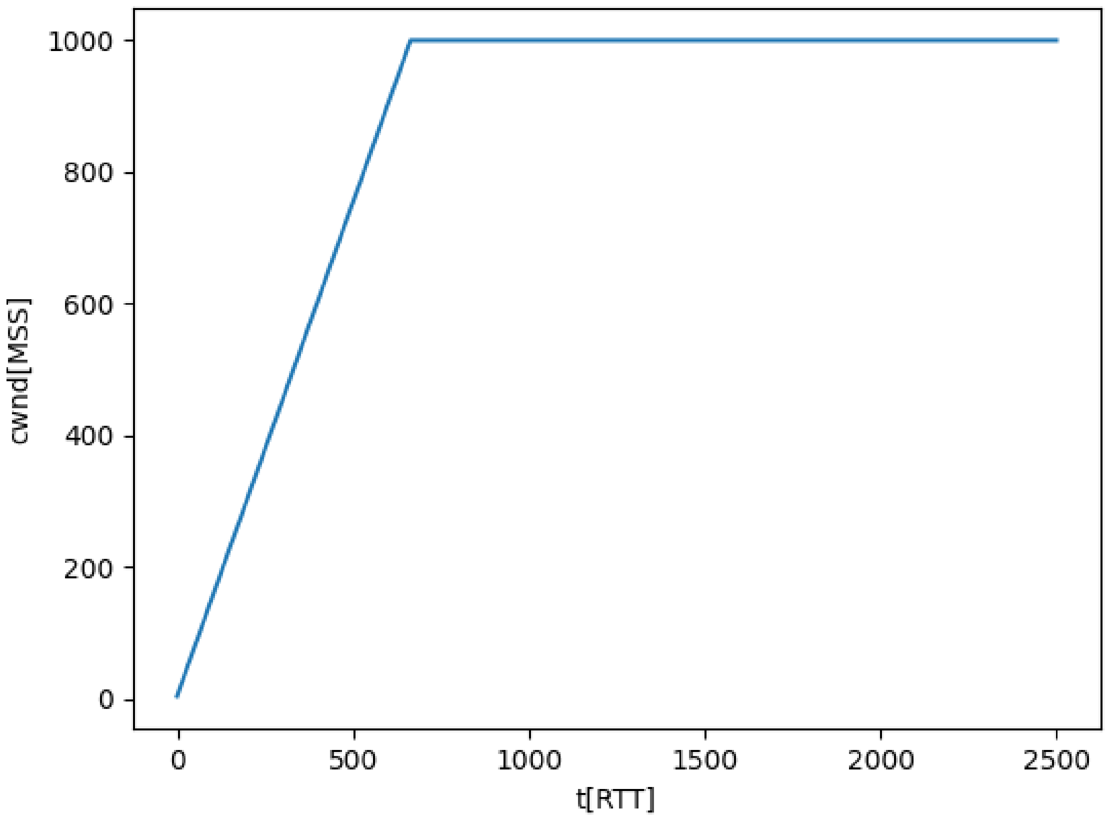

PLOT Number25: Ki = 4.0, Km = 1.5, Kn = 0.5, Kf = 0.1, Ps = 0.01;
cwnd[MSS]
t[RTT]
0
50
100
150
200
250
300
350
0
500
1000
1500
2000
2500

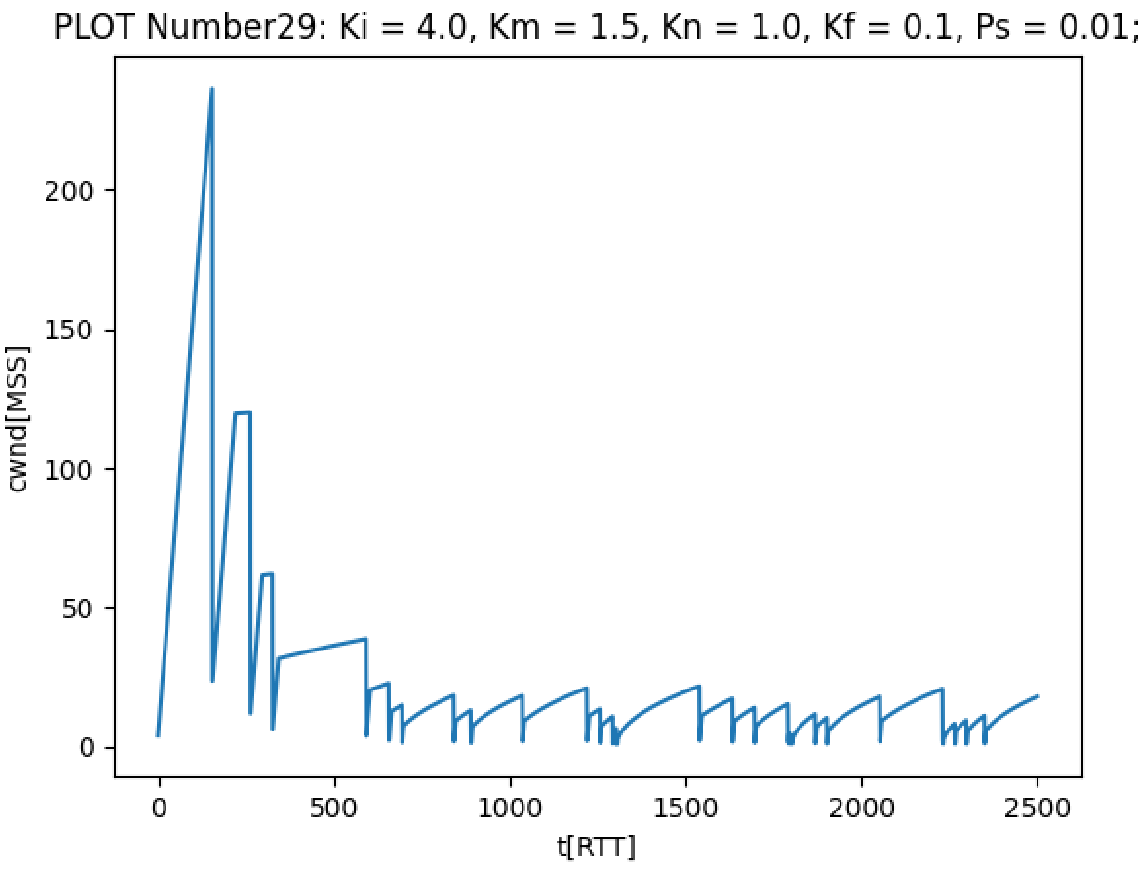
PLOT Number26: Ki = 4.0, Km = 1.5, Kn = 0.5, Kf = 0.1, Ps = 0.0001;



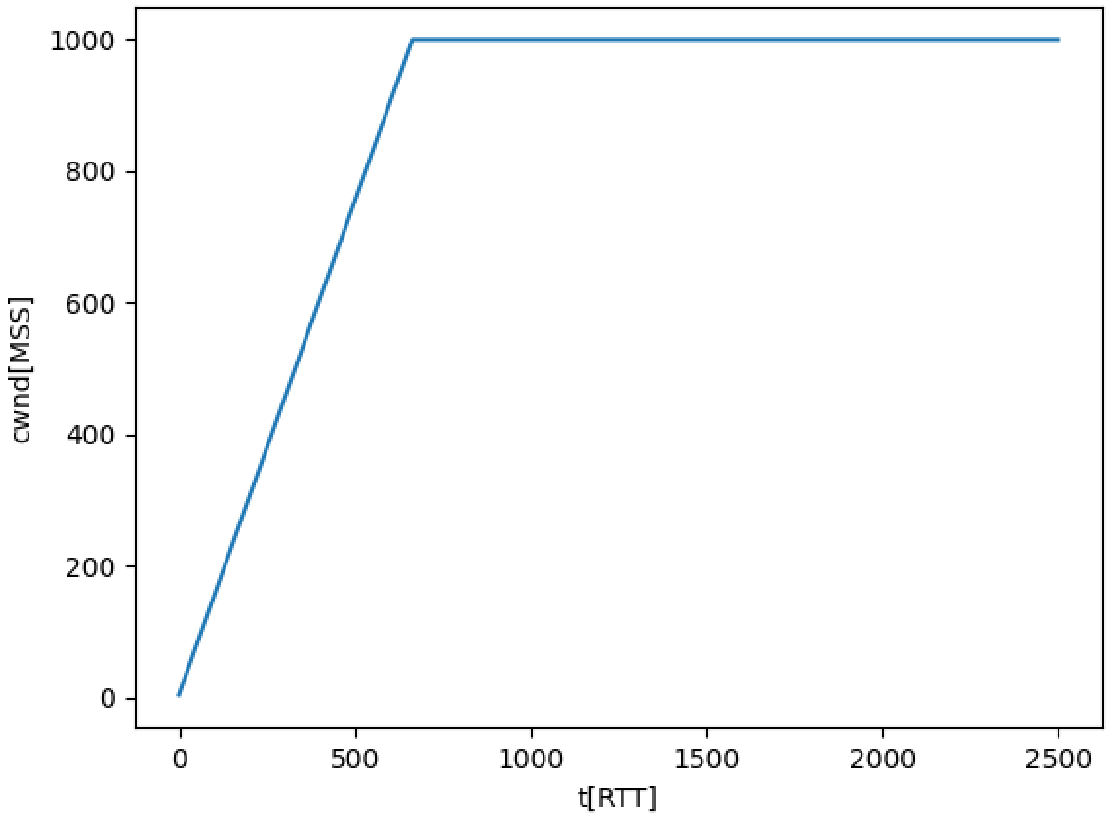

PLOT Number27: Ki = 4.0, Km = 1.5, Kn = 0.5, Kf = 0.3, Ps = 0.01;
cwnd[MSS]
t[RTT]
0
20
40
60
80
100
0
500
1000
1500
2000
2500

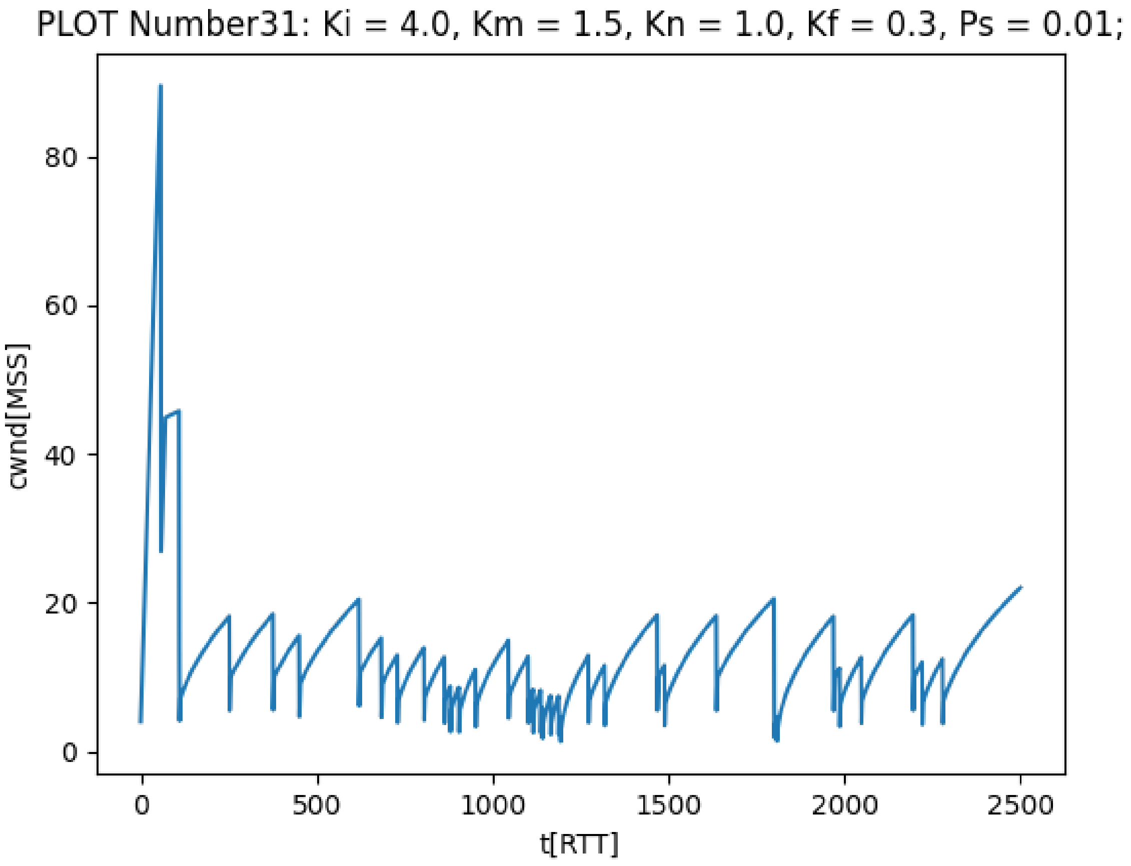
PLOT Number28: Ki = 4.0, Km = 1.5, Kn = 0.5, Kf = 0.3, Ps = 0.0001;

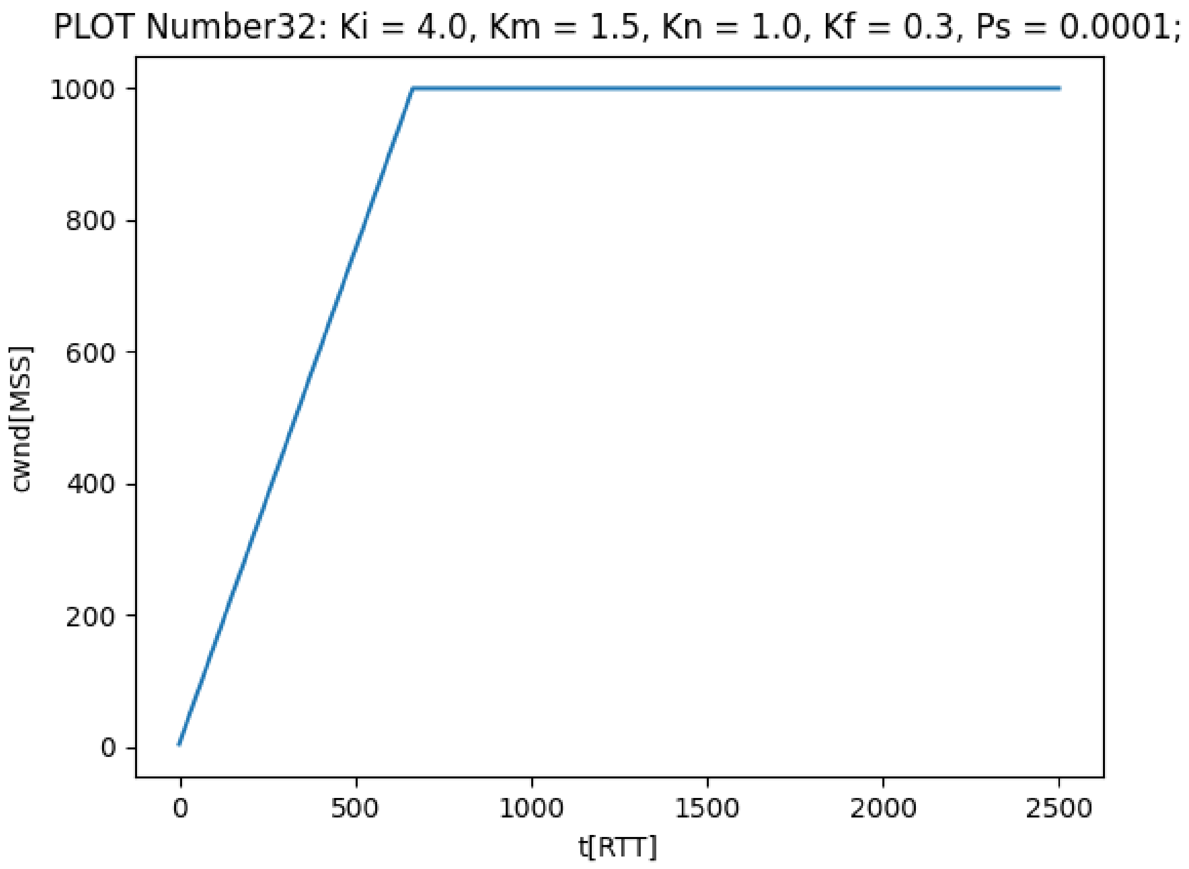



PLOT Number29: Ki = 4.0, Km = 1.5, Kn = 1.0, Kf = 0.1, Ps = 0.01;
cwnd[MSS]
t[RTT]
0
50
100
150
200
0
500
1000
1500
2000
2500

PLOT Number30: Ki = 4.0, Km = 1.5, Kn = 1.0, Kf = 0.1, Ps = 0.0001;




PLOT Number31: Ki = 4.0, Km = 1.5, Kn = 1.0, Kf = 0.3, Ps = 0.01;
cwnd[MSS]
t[RTT]
0
20
40
60
80
0
500
1000
1500
2000
2500



Learnings

- Each of the parameter plays an important role as we can see in the graph.
- By varying P_s between 0.01 and 0.0001 , as the change in probability is very big(100 times), we can clearly see that the graph is much more dependent on P_s

Conclusion

TCP Congestion control mechanism is an important aspect of TCP.
Congestion causes packets to be dropped on the network, and therefore leads to data loss and unreliable connection. Therefore effective congestion control is an important issue in the transport layer.

References

Slides on moodle/Lab Lecture video
Gfg - https://www.geeksforgeeks.org/tcp-congestion-control/