

APPENDIX

JACK CAI

CONTENTS

Code Snippet 1	1
Code Snippet 2	3
Code Snippet 3	3
Code Snippet 4	3
Code Snippet 5	3
Code Snippet 6	3
Code Snippet 7	3
Code Snippet 8	4
Code Snippet 9	4

CODE SNIPPET 1

```
1 import matplotlib.pyplot as plt
2 import numpy as np
3 import random
4
5 class User:
6     _window_size = None
7     _steps = None
8
9     def __init__(self, n):
10         self._window_size = n
11         self._steps = [n]
12
13     def Increase(self, alpha):
14         self._window_size += alpha
15
16     def Decrease(self, beta):
17         self._window_size = max(1, int(self._window_size * beta))
18
19     def Log(self):
20         self._steps.append(self._window_size)
21
22     def GetWindowSize(self):
23         return self._window_size
24
25     def GetSteps(self):
26         return self._steps
```

```
27
28 class Host:
29     _alpha = 0
30     _beta = 0
31     _max_window_size = 0
32     _clients = []
33
34     def __init__(self, seed, alpha=1, beta=0.5, max_window_size=200):
35         random.seed(seed)
36         self._alpha = alpha
37         self._beta = beta
38         self._max_window_size = max_window_size
39
40     def addClients(self, n):
41         for i in range(n):
42             self._clients.append(User(random.randint(5, 15)))
43
44     def Step(self):
45         sum_windows = 0
46         for client in self._clients:
47             sum_windows += client.GetWindowSize()
48
49         if (sum_windows > self._max_window_size):
50             # Congestion event
51             for client in self._clients:
52                 client.Decrease(self._beta)
53                 client.Log()
54             return
55
56         for client in self._clients:
57             client.Increase(self._alpha)
58             client.Log()
59
60     def Plot(self, x, y):
61         xpoints = np.array(self._clients[x].GetSteps())
62         ypoints = np.array(self._clients[y].GetSteps())
63
64         plt.plot(xpoints, ypoints)
65         plt.show()
66
67 if __name__ == "__main__":
68     my_host = Host(42069)
69     my_host.addClients(2)
70
71     for i in range(1000):
72         my_host.Step()
73     my_host.Plot(0, 1)
```

CODE SNIPPET 2

```

67 if __name__ == "__main__":
68     my_host = Host(42069)
69     my_host.addClients(200)

```

CODE SNIPPET 3

```

68 if __name__ == "__main__":
69     my_host = Host(42069, 1, 0.5, 5000)

```

CODE SNIPPET 4

```

68 if __name__ == "__main__":
69     my_host = Host(42069, 10, 0.5, 5000)

```

CODE SNIPPET 5

```

68 if __name__ == "__main__":
69     my_host = Host(42069, 1, 0.1, 5000)

```

CODE SNIPPET 6

```

68 if __name__ == "__main__":
69     my_host = Host(42069, 1, 0.9, 5000)

```

CODE SNIPPET 7

```

28 class Host:
29     _alpha = 0
30     _beta = 0
31     _max_window_size = 0
32     _clients = []
33     _step_count = 1

44 def Step(self):
45     sum_windows = 0
46     for client in self._clients:
47         sum_windows += client.GetWindowSize()
48
49     if (sum_windows > self._max_window_size):
50         # Congestion event
51         for client in self._clients:
52             client.Decrease(self._beta)
53             client.Log()
54         self._alpha = max(1, int(self._alpha * self._beta))
55         return
56
57     for client in self._clients:
58         client.Increase(self._alpha)
59         client.Log()

```

```
60     self._alpha += self._step_count
61     self._step_count += 1
```

CODE SNIPPET 8

```
28 class Host:
29     _alpha = 0
30     _beta = 0
31     _max_window_size = 0
32     _clients = []
33
44 def Step(self):
45     sum_windows = 0
46     for client in self._clients:
47         sum_windows += client.GetWindowSize()
48
49     if (sum_windows > self._max_window_size):
50         # Congestion event
51         for client in self._clients:
52             client.Decrease(self._beta)
53             client.Log()
54         self._alpha = max(1, int(self._alpha * self._beta))
55         return
56
57     for client in self._clients:
58         client.Increase(self._alpha)
59         client.Log()
60     self._alpha *= 2
61
```

CODE SNIPPET 9

```
5 class User:
6     _alpha = 0
7     _beta = 0
8     _window_size = None
9     _steps = None
10
11 def __init__(self, n, alpha, beta):
12     self._alpha = alpha
13     while (0 == beta):
14         beta = random.random()
15     self._beta = beta
16     self._window_size = n
17     self._steps = [n]
18
19 def Increase(self):
20     self._window_size += self._alpha
21
```

```
22     def Decrease(self):
23         self._window_size = max(1, int(self._window_size * self._beta))

28 class Host:
29     _max_window_size = 0
30     _clients = []
31
32     def __init__(self, seed, max_window_size=20):
33         random.seed(seed)
34         self._max_window_size = max_window_size
35
36     def addClients(self, n):
37         for i in range(n):
38             self._clients.append(User(random.randint(1, 10),
39                                     ↪ random.randint(1, 10), random.random()))
39
40     def Step(self):
41         sum_windows = 0
42         for client in self._clients:
43             sum_windows += client.GetWindowSize()
44
45         if (sum_windows > self._max_window_size):
46             # Congestion event
47             for client in self._clients:
48                 client.Decrease()
49                 client.Log()
50             return
51
52         for client in self._clients:
53             client.Increase()
54             client.Log()
```