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

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

2 J. CAI

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

Code Snippet 2

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__":

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
```

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

4 J. CAI

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

Code Snippet 8

```
28 class Host:

29    _alpha = 0

30    _beta = 0

31    _max_window_size = 0

32    _clients = []
```

```
def Step(self):
44
           sum_windows = 0
45
           for client in self._clients:
46
                sum_windows += client.GetWindowSize()
47
48
                if (sum_windows > self._max_window_size):
49
50
                    # Congestion event
                    for client in self._clients:
51
                        client.Decrease(self._beta)
52
53
                        client.Log()
                    self._alpha = max(1, int(self._alpha * self._beta))
54
                    return
56
           for client in self._clients:
57
                        client.Increase(self._alpha)
58
59
                        client.Log()
           self.\_alpha *= 2
60
```

Code Snippet 9

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

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

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

CHERITON SCHOOL OF COMPUTER SCIENCE, UNIVERSITY OF WATERLOO *Email address*: jack.cai@uwaterloo.ca