

```

1  #!/usr/bin/env python
2  import sys, parse, math
3
4  def main():
5
6      fileIn = open(sys.argv[1], 'r')
7
8      i = 0
9      flow = {}
10     # Le o arquivo com os dados da saida do primeiro parser
11     texto = fileIn.read()
12     # Pega as informacoes necessarias
13     throughput = [float(r.fixed[0]) for r in parse.findall("Vazao: {} Mbps\n", texto)]
14     delay = [float(r.fixed[0]) for r in parse.findall("Atraso: {} ms\n", texto)]
15     lostPckt = [float(r.fixed[0]) for r in parse.findall("Perda: {} % dos pacotes\n", texto)]
16
17     # Calcula a media, o desvio padrao e o intervalo de confianca para a vazao
18     avgTP = [sum(throughput)/len(throughput)] * len(throughput)
19     temp = [(a-b)*(a-b) for a,b in zip(throughput, avgTP)]
20     stddevTP = math.sqrt(sum(temp)/len(temp))
21
22     minTP = avgTP[0] - 1.96*(stddevTP/math.sqrt(len(temp)))
23     maxTP = avgTP[0] + 1.96*(stddevTP/math.sqrt(len(temp)))
24
25     print(minTP, avgTP[0], maxTP)
26
27     # Calcula a media, o desvio padrao e o intervalo de confianca para o atraso
28     avgDe = [sum(delay)/len(delay)] * len(delay)
29     temp = [(a-b)*(a-b) for a,b in zip(delay, avgDe)]
30     stddevDe = math.sqrt(sum(temp)/len(temp))
31
32     minDe = avgDe[0] - 1.96*(stddevDe/math.sqrt(len(temp)))
33     maxDe = avgDe[0] + 1.96*(stddevDe/math.sqrt(len(temp)))
34
35     print (minDe, avgDe[0], maxDe)
36
37     # Calcula a media, o desvio padrao e o intervalo de confianca para a perda
38     avgLP = [sum(lostPckt)/len(lostPckt)] * len(lostPckt)
39     temp = [(a-b)*(a-b) for a,b in zip(lostPckt, avgLP)]
40     stddevLP = math.sqrt(sum(temp)/len(temp))
41
42     minLP = avgLP[0] - 1.96*(stddevLP/math.sqrt(len(temp)))
43     maxLP = avgLP[0] + 1.96*(stddevLP/math.sqrt(len(temp)))
44
45     print(minLP, avgLP[0], maxLP)
46
47 if __name__ == '__main__':main()

```