Analysis Cooja

February 27, 2019

```
In [1]: #Modules to install via pip pandas,ipynb
        import pandas as pd
        import numpy as np

        import matplotlib.pyplot as plt
        import json
        from pprint import pprint
        import os
        import import_ipynb
        import sys
        sys.path.append('../')
        from functions import *
        from pandas.plotting import scatter_matrix
```

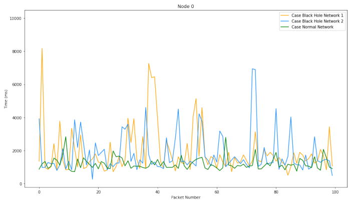
Using function to import datas

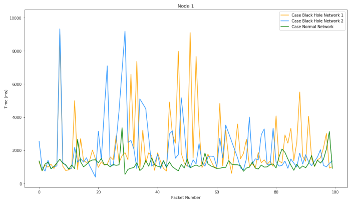
```
In [2]: dataList=coojaJsonImporter("./traces")
        test1BH1=dataList[0]
        test1BH2=dataList[1]
        testNorm =dataList[2]
        data=[]
        cases=[
              "Black Hole Network 1",
                "Black Hole Network 2",
            "Normal Network"
              ]
        colors = [ 'orange','dodgerblue', 'green','violet']
        for nodeList in dataList:
            data.append(createNodes(nodeList))


        #All data collection is in variable node that is a list of list of nodes
        #3 nets input x 9 nodes by net
        data[0][0].pkts[1:5]

        print(getPings(data))
```

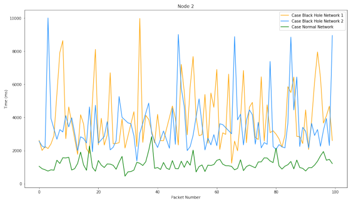
```
Importing test_1BH_2018-11-09_12_31_25.json
Importing test_nom_2018-11-09_08_55_11.json
Importing test_1BH_2018-11-09_14_37_46.json
[99, 100, 97]
```

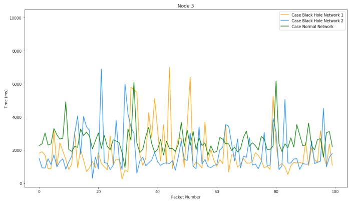
RTT Graph

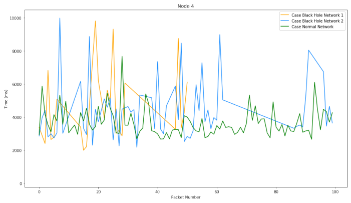
```
In [3]: #It will be necessary when adding more Cases
        fig, axs= plt.subplots(len(data[0]),1, figsize=(15,90),sharey=True, )
        for i in range(len(data)):
            for j in range(len(data[i])):
                axs[j].plot(data[i][j].pkts["pkt"],data[i][j].pkts["rtt"],label="Case " +str(ca
                axs[j].set_title("Node "+ str(j))
                axs[j].set_xlabel("Packet Number")
                axs[j].set_ylabel("Time (ms)")
                axs[j].legend()

        plt.show()
```

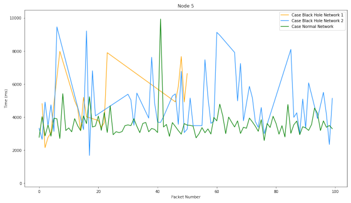

Node 0
Case Black Hole Network 1
Case Black Hole Network 2
Case Normal Network
Time (ms)
0
2000
4000
6000
8000
10000
Packet Number
0
20
40
60
80
100

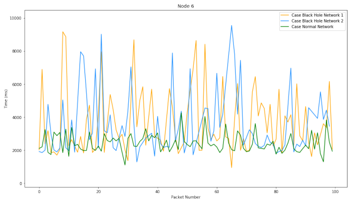

Node 1
Case Black Hole Network 1
Case Black Hole Network 2
Case Normal Network
Time (ms)
0
2000
4000
6000
8000
10000
Packet Number
0
20
40
60
80
100

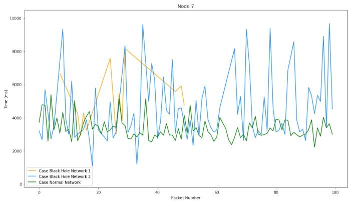

Node 2
Case Black Hole Network 1
Case Black Hole Network 2
Case Normal Network
Time (ms)
0
2000
4000
6000
8000
10000
Packet Number
0
20
40
60
80
100

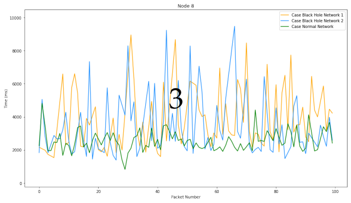

Node 3
Case Black Hole Network 1
Case Black Hole Network 2
Case Normal Network
Time (ms)
0
2000
4000
6000
8000
10000
Packet Number
0
20
40
60
80
100


Node 4
Case Black Hole Network 1
Case Black Hole Network 2
Case Normal Network
Time (ms)
0
2000
4000
6000
8000
10000
Packet Number
0
20
40
60
80
100


Node 5
Case Black Hole Network 1
Case Black Hole Network 2
Case Normal Network
Time (ms)
0
2000
4000
6000
8000
10000
Packet Number
0
20
40
60
80
100

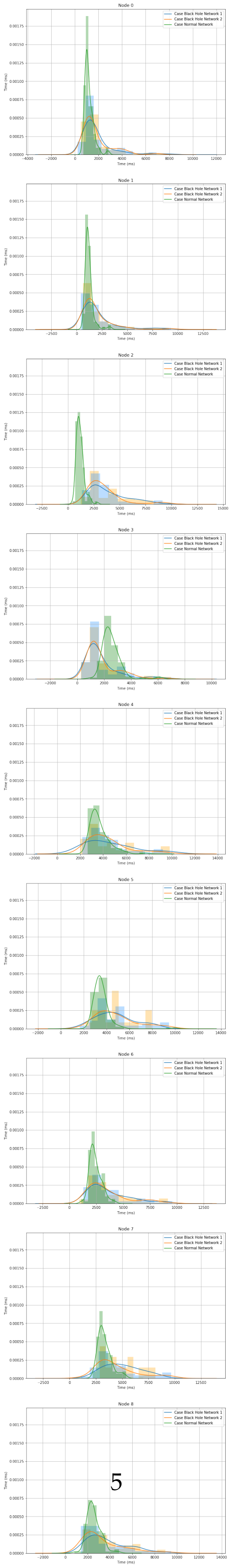

Node 6
Case Black Hole Network 1
Case Black Hole Network 2
Case Normal Network
Time (ms)
0
2000
4000
6000
8000
10000
Packet Number
0
20
40
60
80
100


Node 7
Case Black Hole Network 1
Case Black Hole Network 2
Case Normal Network
Time (ms)
0
2000
4000
6000
8000
10000
Packet Number
0
20
40
60
80
100


Node 8
Case Black Hole Network 1
Case Black Hole Network 2
Case Normal Network
Time (ms)
0
2000
4000
6000
8000
10000
Packet Number
0
20
40
60
80
100

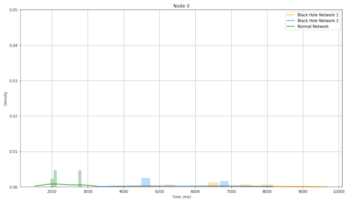
Distibution of the delay divided by Node in the differents Cases

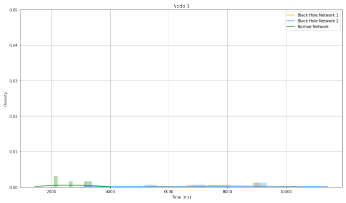
```
In [13]: fig, axs= plt.subplots(len(data[0]),1, figsize=(10,80),sharey=True, )
         for i in range(len(data)):
             for j in range(len(data[i])):
                 data[i][j].pkts["rtt"].plot.kde(
                     ax=axs[j],
                     label="Case " +str(cases[i]),

                 )
                 axs[j].set_ylabel("Time (ms)")
                 data[i][j].pkts["rtt"].hist(density=True,alpha=0.3,color=colors[i], ax=axs[j]
                 axs[j].set_title("Node "+ str(j))
                 axs[j].set_xlabel("Time (ms)")
                 axs[j].legend()
```

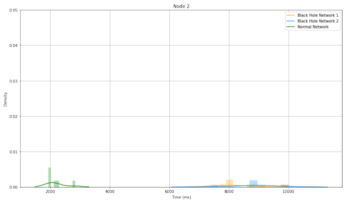

Node 0
Case Black Hole Network 1
Case Black Hole Network 2
Case Normal Network
Time (ms)
0.00000
0.00025
0.00050
0.00075
0.00100
0.00125
0.00150
0.00175
-4000
-2000
0
2000
4000
6000
8000
10000
12000
Time (ms)
Node 1
Case Black Hole Network 1
Case Black Hole Network 2
Case Normal Network
−2500
0
2500
5000
7500
10000
12500
Node 2
−2500
0
2500
5000
7500
10000
12500
15000
Node 3
−2000
0
2000
4000
6000
8000
10000
Node 4
−2000
0
2000
4000
6000
8000
10000
12000
14000
Node 5
Node 6
−2500
0
2500
5000
7500
10000
12500
Node 7
Node 8

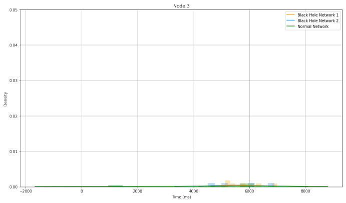
Density of outliers in every node by Case

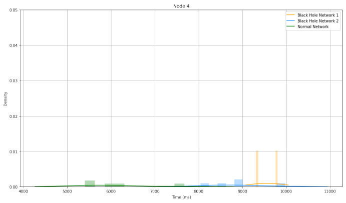
```
In [5]: fig, axs= plt.subplots(len(data[0]),1, figsize=(15,90),sharey=True, )
        for i in range(len(data)):
            for j in range(len(data[i])):
                out=getOutliers(data[i][j].pkts)
                if not out.empty | len(out)<2 :
                    axe=axs[j]
                    out["rtt"].plot.kde(
                    ax=axe,
                    label=cases[i],
                    color=colors[i]
                    )
                    axe.set_ylabel("Density")
                    out["rtt"].hist(density=True,alpha=0.3, ax=axe, color=colors[i])
                    axe.set_title("Node "+ str(j))
                    axe.set_xlabel("Time (ms)")
                    axe.legend()
                    axs[j].set_ylim([0, 0.05])
```

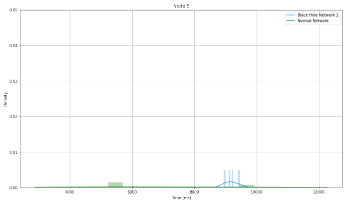

Node 0

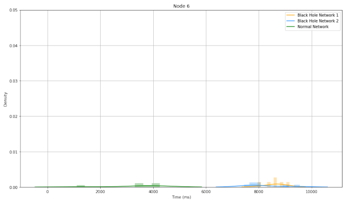

Node 1

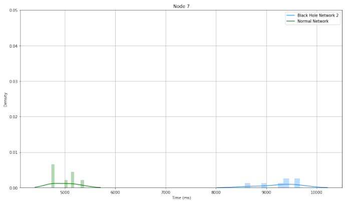

Node 2

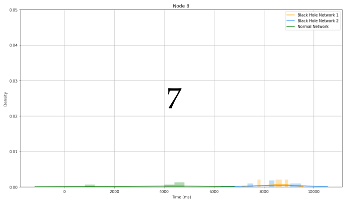

Node 3


Node 4


Node 5

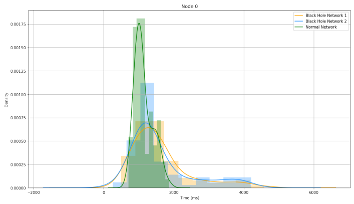

Node 6

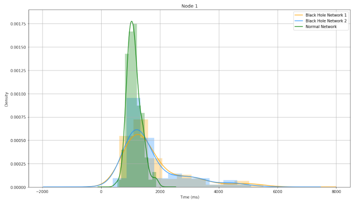

Node 7

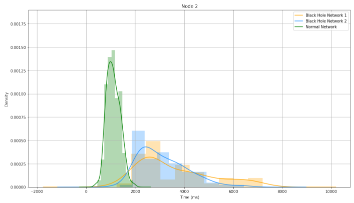

Node 8

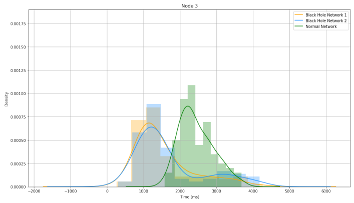
Density of delay without outliers in every node by Case

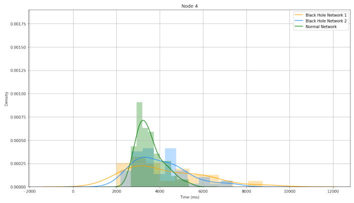
```
In [6]: fig, axs= plt.subplots(len(data[0]),1, figsize=(15,90),sharey=True, )
        for i in range(len(data)):
            for j in range(len(data[i])):
                out=getStdValues(data[i][j].pkts)
                if not out.empty :
                    ax=axs[j]
                    out["rtt"].plot.kde(
                    ax=ax,
                    label=cases[i],
                         color=colors[i]
                )
                    ax.set_ylabel("Density")
                    out["rtt"].hist(density=True,alpha=0.3, ax=ax, color=colors[i])
                    ax.set_title("Node "+ str(j))
                    ax.set_xlabel("Time (ms)")
                    ax.legend()
```

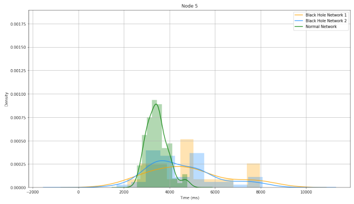

Node 0
Black Hole Network 1
Black Hole Network 2
Normal Network
Density
0.00000
0.00025
0.00050
0.00075
0.00100
0.00125
0.00150
0.00175
-2000
0
2000
4000
6000
Time (ms)

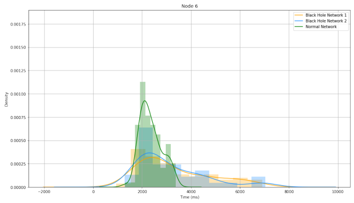

Node 1
Black Hole Network 1
Black Hole Network 2
Normal Network
Density
0.00000
0.00025
0.00050
0.00075
0.00100
0.00125
0.00150
0.00175
-2000
0
2000
4000
6000
8000
Time (ms)

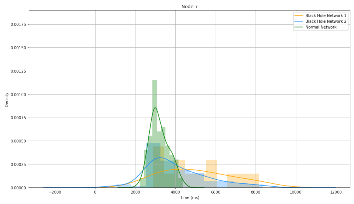

Node 2
Black Hole Network 1
Black Hole Network 2
Normal Network
Density
0.00000
0.00025
0.00050
0.00075
0.00100
0.00125
0.00150
0.00175
-2000
0
2000
4000
6000
8000
10000
Time (ms)

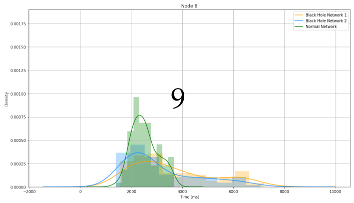

Node 3
Black Hole Network 1
Black Hole Network 2
Normal Network
Density
0.00000
0.00025
0.00050
0.00075
0.00100
0.00125
0.00150
0.00175
-2000
-1000
0
1000
2000
3000
4000
5000
6000
Time (ms)


Node 4
Black Hole Network 1
Black Hole Network 2
Normal Network
Density
0.00000
0.00025
0.00050
0.00075
0.00100
0.00125
0.00150
0.00175
-2000
0
2000
4000
6000
8000
10000
12000
Time (ms)


Node 5
Black Hole Network 1
Black Hole Network 2
Normal Network
Density
0.00000
0.00025
0.00050
0.00075
0.00100
0.00125
0.00150
0.00175
-2000
0
2000
4000
6000
8000
10000
Time (ms)


Node 6
Black Hole Network 1
Black Hole Network 2
Normal Network
Density
0.00000
0.00025
0.00050
0.00075
0.00100
0.00125
0.00150
0.00175
-2000
0
2000
4000
6000
8000
10000
Time (ms)


Node 7
Black Hole Network 1
Black Hole Network 2
Normal Network
Density
0.00000
0.00025
0.00050
0.00075
0.00100
0.00125
0.00150
0.00175
-2000
0
2000
4000
6000
8000
10000
12000
Time (ms)

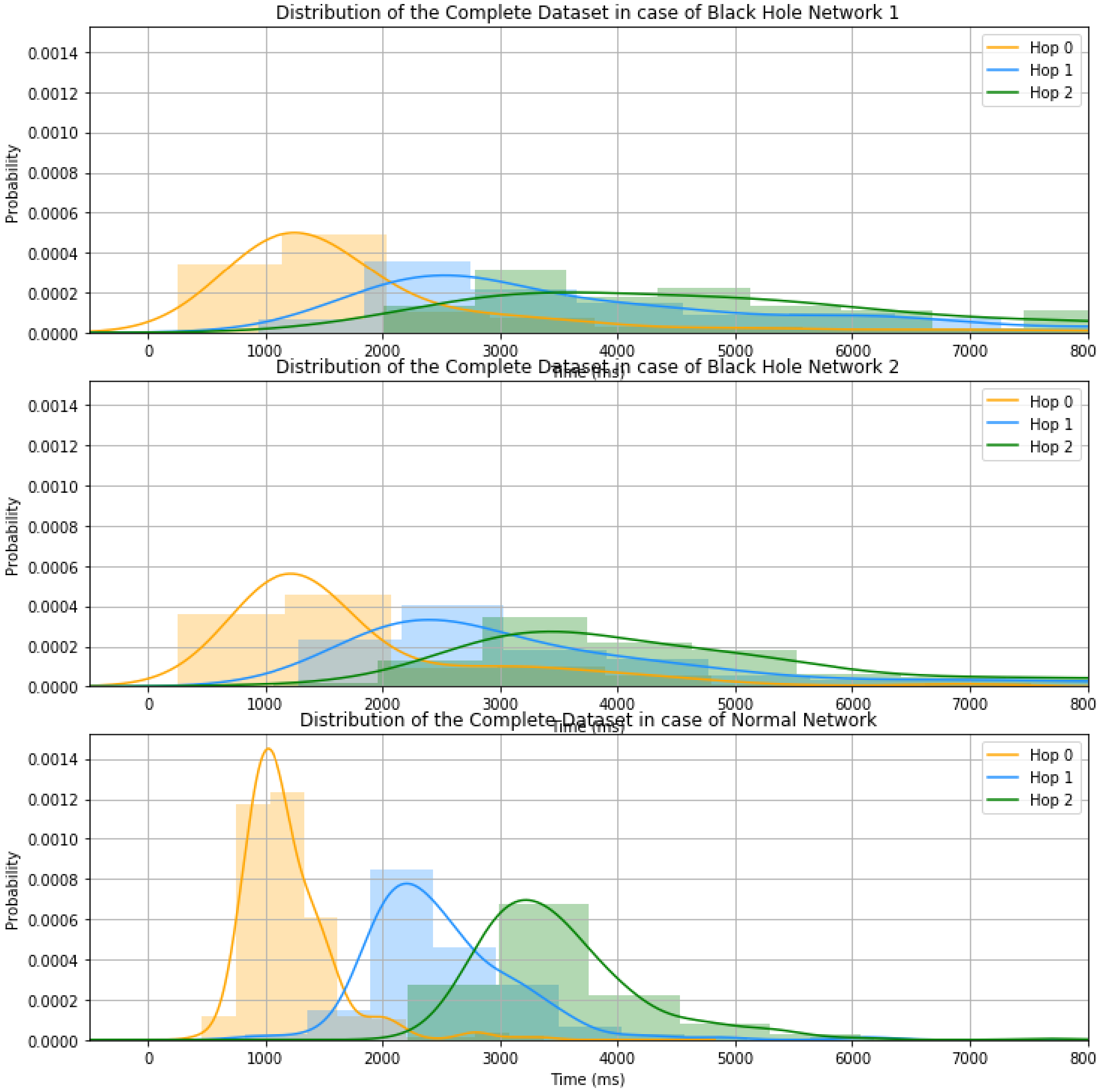

Node 8
Black Hole Network 1
Black Hole Network 2
Normal Network
Density
0.00000
0.00025
0.00050
0.00075
0.00100
0.00125
0.00150
0.00175
-2000
0
2000
4000
6000
8000
10000
Time (ms)

Data Preparation for Plot

```
In [19]: hoplist=[]
         df_a = pd.DataFrame( columns = ['pkt'])
         dataHop=[]

         listoflists = []
         for i in range(len(data)):
             sublist = []
             for j in range(3):
                 sublist.append((df_a))
             dataHop.append(sublist)
         #print (listoflists)

         for i in range(len(data)):
             col=[]
             for j in range(len(data[i])):
                 hop=data[i][j].hop-1

                 dataHop[i][hop]= pd.concat([dataHop[i][hop],data[i][j].pkts],sort=True)
```

Distribution of the delay in correlation with the Hops

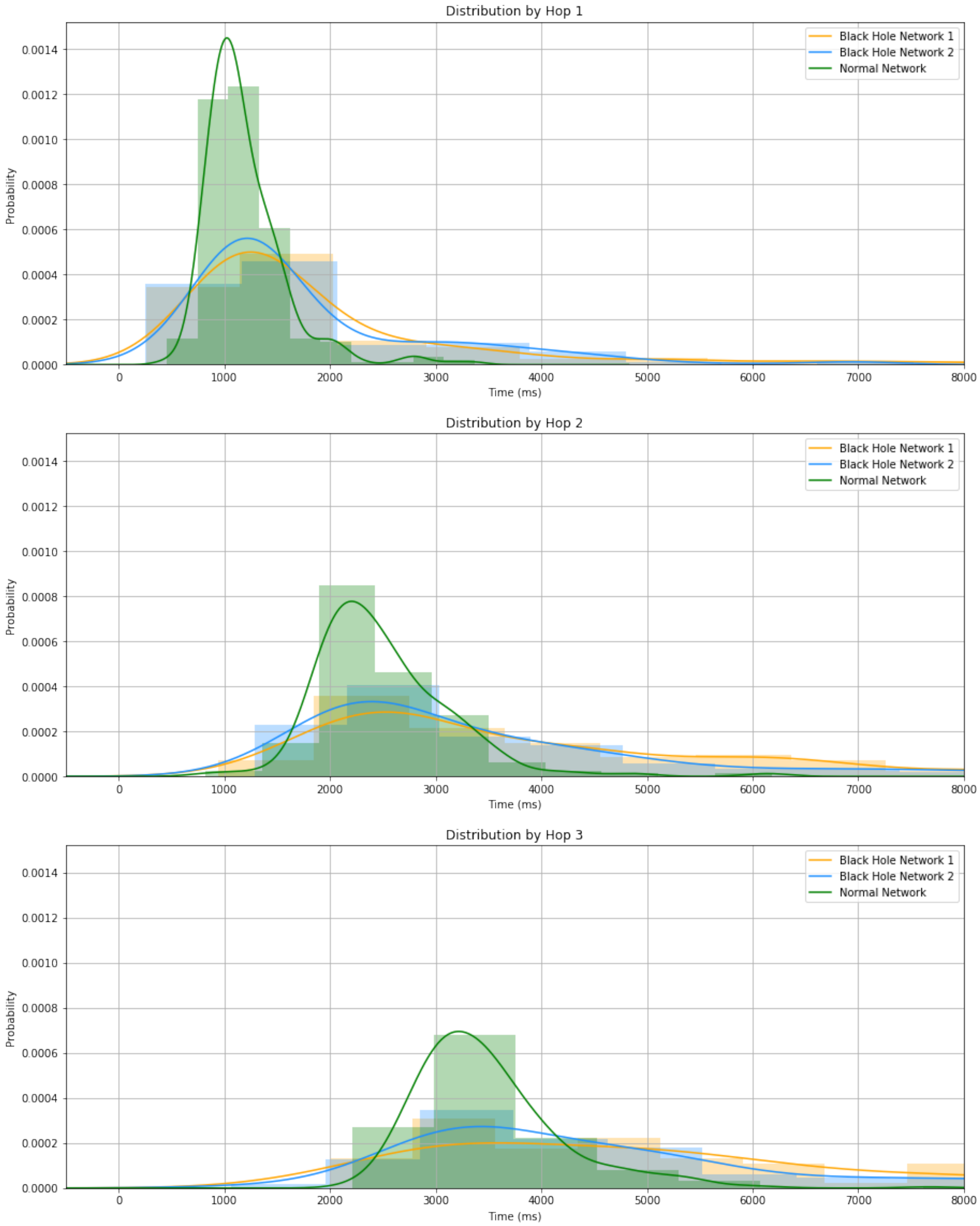
```
In [55]: dataHopT=[*zip(*dataHop)]
         #dataHopT=dataHop

         fig, axs= plt.subplots(len(cases),1,figsize=(10,10
                                                    )
                                ,sharey=True )
         plt.tight_layout()
         for i in range(len(dataHopT)):
             for j in range(len(dataHopT[0])):
                 axs[j].autoscale(enable=True)
                 dataHopT[i][j]['rtt'].plot.kde(
                     ax=axs[j],
                     label="Hop "+str(i),
                     color=colors[i]
                 )

                 dataHopT[i][j]["rtt"].hist(density=True,alpha=0.3, ax=axs[j],color=colors[i])
                 axs[j].set_title("Distribution of the Complete Dataset in case of "+ cases[j]
                 axs[j].set_xlabel("Time (ms)")
                 axs[j].set_ylabel("Probability")
                 axs[j].legend()
                 axs[j].set_xlim([-500, 8000])
```



Considering the clean data, produce a histogram of delay depending on the hop-distance from the root

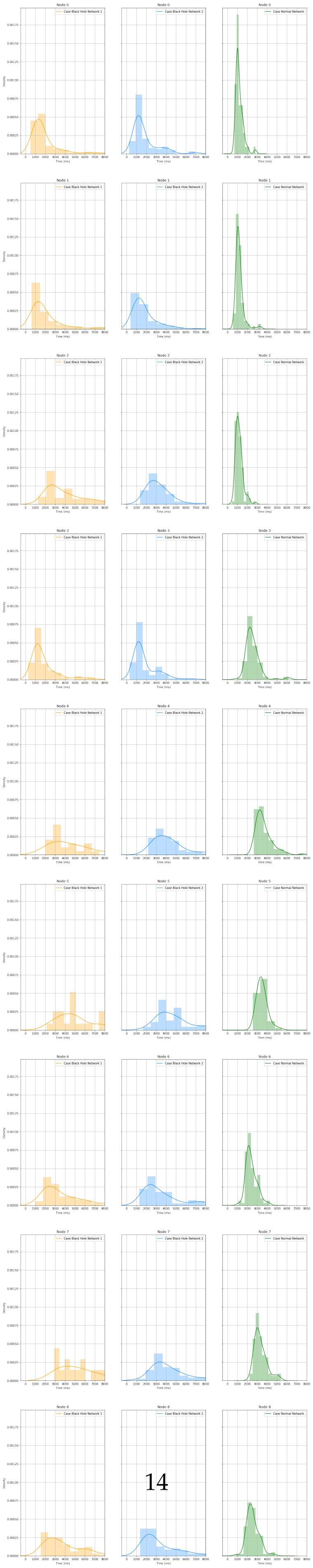
```
In [24]: fig, axs= plt.subplots(3,1, figsize=(15,20),sharey=True, )
         for i in range(len(dataHop)):
             for j in range(len(dataHop[i])):
                 dataHop[i][j]['rtt'].plot.kde(
                     ax=axs[j],
                     label=cases[i],
                     color=colors[i]
                 )

                 dataHop[i][j]["rtt"].hist(density=True,alpha=0.3, ax=axs[j],color=colors[i])
```

```
axs[j].set_xlabel("Time (ms)")
axs[j].set_ylabel("Probability")
axs[j].set_title("Distribution by Hop "+ str(j+1))
axs[j].legend()

axs[j].set_xlim([-500, 8000])
```



Kernel Density Estimation by Case by Node

```
In [27]: fig, axs= plt.subplots(9,3, figsize=(18,100),sharey=True, )
         for i in range(len(data)):
             for j in range(len(data[i])):
                 #print(i,j)
                 ax=axs[j][i]
                 data[i][j].pkts["rtt"].plot.kde(
                     ax=ax,
                     label="Case " +str(cases[i]),
                     color=colors[i]

                 )

                 ax.set_ylabel("Density")
                 data[i][j].pkts["rtt"].hist(density=True,alpha=0.3,color=colors[i], ax=ax)
                 ax.set_title("Node "+ str(j) )
                 ax.set_xlabel("Time (ms)")
                 ax.legend()
                 ax.set_xlim([-500, 8000])
```


Node 0
Case Black Hole Network 1
Node 0
Case Black Hole Network 2
Node 0
Case Normal Network
Node 1
Case Black Hole Network 1
Node 1
Case Black Hole Network 2
Node 1
Case Normal Network
Node 2
Case Black Hole Network 1
Node 2
Case Black Hole Network 2
Node 2
Case Normal Network
Node 3
Case Black Hole Network 1
Node 3
Case Black Hole Network 2
Node 3
Case Normal Network
Node 4
Case Black Hole Network 1
Node 4
Case Black Hole Network 2
Node 4
Case Normal Network
Node 5
Case Black Hole Network 1
Node 5
Case Black Hole Network 2
Node 5
Case Normal Network
Node 6
Case Black Hole Network 1
Node 6
Case Black Hole Network 2
Node 6
Case Normal Network
Node 7
Case Black Hole Network 1
Node 7
Case Black Hole Network 2
Node 7
Case Normal Network
Node 8
Case Black Hole Network 1
Node 8
Case Black Hole Network 2
Node 8
Case Normal Network
Density
0.00000
0.00025
0.00050
0.00075
0.00100
0.00125
0.00150
0.00175
0
1000
2000
3000
4000
5000
6000
7000
8000
Time (ms)

Division by Hop in the 3 cases

```
In [ ]:
```

```
In [ ]:
```