IT200 Course Project
D2D Communication in NS-3

Gaurang Jitendra Velingkar
191IT113
Information Technology
National Institute of Technology Karnataka
Surathkal, India 575025
Email: gaurang.191it113@nitk.edu.in

Jason Krithik Kumar S
191IT123
Information Technology
National Institute of Technology Karnataka
Surathkal, India 575025
Email: jason.191it123@nitk.edu.in

Abhinav Dugar
191IT202
Information Technology
National Institute of Technology Karnataka
Surathkal, India 575025
Email: abhinavdugar.191it202@nitk.edu.in

Gaurav Singh
191IT218
Information Technology
National Institute of Technology Karnataka
Surathkal, India 575025
Email: gauravsingh.191it218@nitk.edu.in

***Abstract*—The wireless communication industry, driven by the manifold increase in data and devices, is demanding the attention of services that support fast and uninterrupted communication. A plethora of services use wireless communications, like mobile gaming, public safety, social networking, etc. Prioritising communication requests by devices in a dense environment becomes a necessity so that all resources that aid communication can be used intelligently. Device-to-Device communications is a new paradigm in cellular networks which allows user equipments (UEs) in close proximity to communicate using a direct link rather than having their radio signal travel through the base station. Sharding divides all the user equipments into separate clusters, each of which support communication that is independent of other shards. Signal to Interference and Noise Ratio or SINR is a measure of signal quality. This project works on sharding devices in a region based on the density of devices and finding the SINR value of the destination D2D node in a particular shard where each D2D pair starts communication at regular intervals of time.**

I. INTRODUCTION

Cellular network is now four generations old. The need for fast, multimedia-rich data exchange, along with high quality voice calls has been the driving force of this journey. With newer and more demanding applications growing their subscriber base at an exponential rate, there is an urgent requirement for more novel techniques to boost data rates and reduce latency. It is estimated that this demand will grow 500-fold by 2025. Current methods in cellular networks will not be able to meet the growing demand. There has to be better way to improve spectrum utilization, in next generation cellular networks. One prominent solution is D2D communication. D2D enables direct communication with another device without involvement of base station or a master cellular node. It differs from Bluetooth communication and other existing modes in terms of distance and data rates. Due to the physical proximity of devices, and the potential reuse gain and hop gain, D2D communications can significantly increase network spectral efficiency and energy efficiency. D2D will thus be of the primary additions of 5G cellular communications. D2D also enhances fairness, QOS matching and congestion control. It is particularly helpful for areas where signal strengths are weak.

A method for sharding, based primarily on the NS-3 application, Public Safety and Commercial Application has been devised. The Device to Device (D2D) pairs are sharded based on their density. Sharding or Clustering is basically an Unsupervised learning method that divides the data points into a number of specific batches or groups, such that the data points in the same groups have similar properties and data points in different groups. Since for sharding multiple clustering algorithms can be considered but here the implementation has been done using Density-based spatial clustering of applications with noise (DBSCAN) method.

Even though existing solutions involving Device to Device Communication and Clustering Algorithm such as K-means, PAM clustering and hierarchical clustering exist, not much research has been done with clustering in Device to Device Communication with the usage of DBSCAN Algorithm. Partitioning methods (K-means, PAM clustering) are severely affected by the presence of noise and outliers in the data. Moreover they are only suitable for well-separated clusters. DBSCAN algorithm overcomes the drawbacks of K-Means algorithm. DBSCAN algorithm identifies the dense region by grouping together data points that are closed to each other based on distance measurement and makes clustering much more efficient.

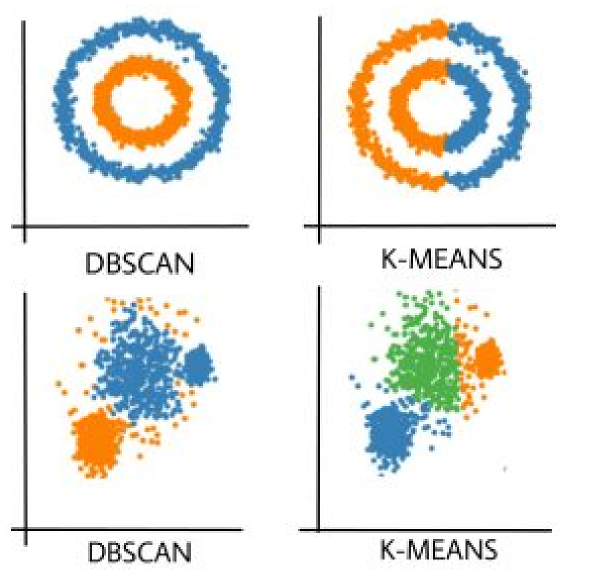


Fig. 1. DBSCAN vs K-Means

Key Contributions of our work are:

- Implemented a network topology consisting of various User Equipment's and Cellular Users
- Sharding of various devices has been done with the help of DBSCAN Clustering Algorithm
- Clustering has been done in such a manner that each shard contains at least one Cellular User
- SINR for each D2D pair for a shard is calculated and displayed

DBSCAN Clustering Algorithm

It groups 'densely grouped' data points into a single cluster. It can identify clusters in large spatial datasets by looking at the local density of the data points. It also does not require the number of clusters to be told beforehand, unlike K-Means, where we have to specify the number of centroids. DBSCAN requires only two parameters: epsilon and minPoints. Epsilon is the radius of the circle to be created around each data point to check the density and minPoints is the minimum number of data points required inside that circle for that data point to be classified as a Core point.

SINR Calculations

SINR stands for Signal to Interference and Noise Ratio. It is a measure of Signal Quantity and Interference and Noise Quantity. It is a very important measurement in terms of RF and sometime it is also called as SNR in absence of interference. It indicates how much desired signal is stronger compare to Noise and interference. Its unit is dB.

II. Literature Survey

Previous research in Computer Networks have focused on clustering to increase the number of users requesting users to using SINR interference models for better routing and channel allocation algorithms. Several such papers were looked at for inspiration and the network was simulated using ns3. In this paper, 2^{nd} paper's sharding concept of D2D and CU has been coupled with 1^{st} paper's clustering algorithm and SINR taking inspiration from 3^{rd} paper. In the "Summary of Literature Survey" Table I, are the papers which have been used for reference.

III. Problem Statement

Effective D2D Sharding based on density using DBSCAN clustering algorithm and finding SINR values of each D2D pair in a particular shard where each D2D pair starts communication at regular intervals.

Objectives

1) Make eNB, Cellular unit, and D2D nodes in different positions.
2) Cluster the nodes using DBSCAN clustering algorithm on the basis of density.
3) Generate GNUplot to get the nodes and the clusters in a graphical format.
4) After Sharding, find out the SINR values of each D2D pair starting communication in regular interval of one second.

IV. Methodology

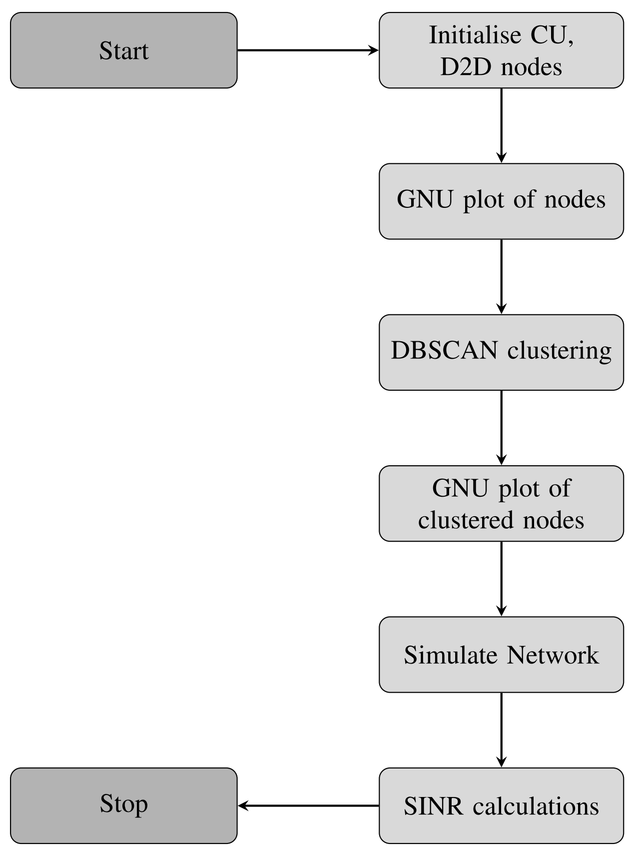


TABLE I
SUMMARY OF LITERATURE SURVEY

Sl. No.	Conference	Title	Merits
1.	21st Asia-Pacific Conference on Communications (APCC) - APCC2015 Special Technical Session	A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise	This clustering algorithm overcomes the problem of discovering clusters of arbitrary shape having minimal requirements of domain knowledge to determine the input parameter and performs significantly better than CLARANS clustering algorithms.
2.	KDD'96: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining	Using Clustering Techniques to Improve Capacity of LTE Networks	This paper deals with increasing the number of users requesting services by grouping nodes nearby as clusters and allocating a single bandwidth for that cluster. A cluster has one cu and more d2d nodes.
3.	2012 IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)	Experimental evaluation of measurement-based SINR interference models	Since SINR interference models is essential for better and more efficient channel assignment algorithms and routing algorithms. Since there is no comprehensive comparison on the accuracy of these measurement-based SINR interference models. This paper makes comparisons and find that it performs better than non SINR models.

A. DBSCAN

It is a clustering algorithm relying on a density-based notion of clusters which is designed to discover clusters of arbitrary shape. DBSCAN algorithm has two major parameters which are *eps* and minPts.
eps: *eps* defines the neighbourhood of a particular node. If the distance between two nodes is less than *eps* value, then they are considered to be the neighbours of each other.
minPts: *minPts* defines the minimum number of nodes within the *eps* radius. If the number of points around a node is less than *minPts*, then it is not considered as a core point. If a point is not considered as core point then its considered as noise. Only core points are taken to be parts of clusters generally.

The Algorithm for DBSCAN can be divided into the following parts:

1) Checking nearby points: The algorithm first, for each node, counts the number of nodes which are at a *minEps* distance from the given node. It creates a graph based on this, appending those elements connected with the given node. We create an array of vectors for this. So all elements within the *eps* radius of the particular node are pushed back to the vector having the index as the node.

2) Classify nodes: Then the algorithm picks up an arbitrary node from the dataset and checks if it has at least *minPoint* number of points near it. If it does, then it is known as a core point. Then it puts this node and its neighbours under the same cluster. It does this by using Depth First Search(DFS) on this node and classifying nodes which don't belong to any cluster. Doing this the clusters are expanded by recursively repeating this DFS. Each neighbouring node which is a core node is classified to the same cluster. Those nodes which are not in the vicinity of any cluster will be labeled as noise.

3) Make a cluster Vector: Create a vector of vector of datatype int. Iterate through each node and append it to the respective vector based on which cluster it belongs to. Thus, each index of the vector will contain all the nodes in a particular cluster.

B. Gnuplot

The project uses the Gnuplot library to generate the topology of the network as an image. The enB node and the CU nodes are first plotted by the function. The clusters are passed as a parameter to the function. The program loops through the cluster and plots all the nodes, clustered and unclustered.

A source file is generated initially, that when converted to an image displays all nodes before sharding. The source file should be manually converted into an image. A file, named *topologyInitial.png*, will be created in the root folder. Further, after sharding, another source file *topologyFinal.plt* will be created in the root folder. This can be converted into an image using the same command as above, but after changing the source file name.

C. SINR Calculation

The project used SideLinkTraces to generate the DownLink SINR statistics which is stored in an output text file. The output text file contained the following columns: *Time*, *CellId*, *IMSI*, *RNTI*, *RSRP*, *SINR*, *ComponentCarrierId*.

IMSI has been used as an unique ID for distinguishing the various nodes in the network and providing the SINR for the receiving node in a D2D pair. The International Mobile Subscriber Identity(IMSI) is a number that uniquely identifies every user in a cellular network.

SINR gives a measure of how strong a signal is with respect to the noise and interference present in the network. The formula for the calculation of SINR is given as:

$$SINR = \frac{SignalPower}{Noise + Interference} \tag{1}$$

After the creation of the DownLink SINR Statistics file , it has been opened at the our main *.cc file* during which we are able to map the ISMI values of the receiving nodes in the

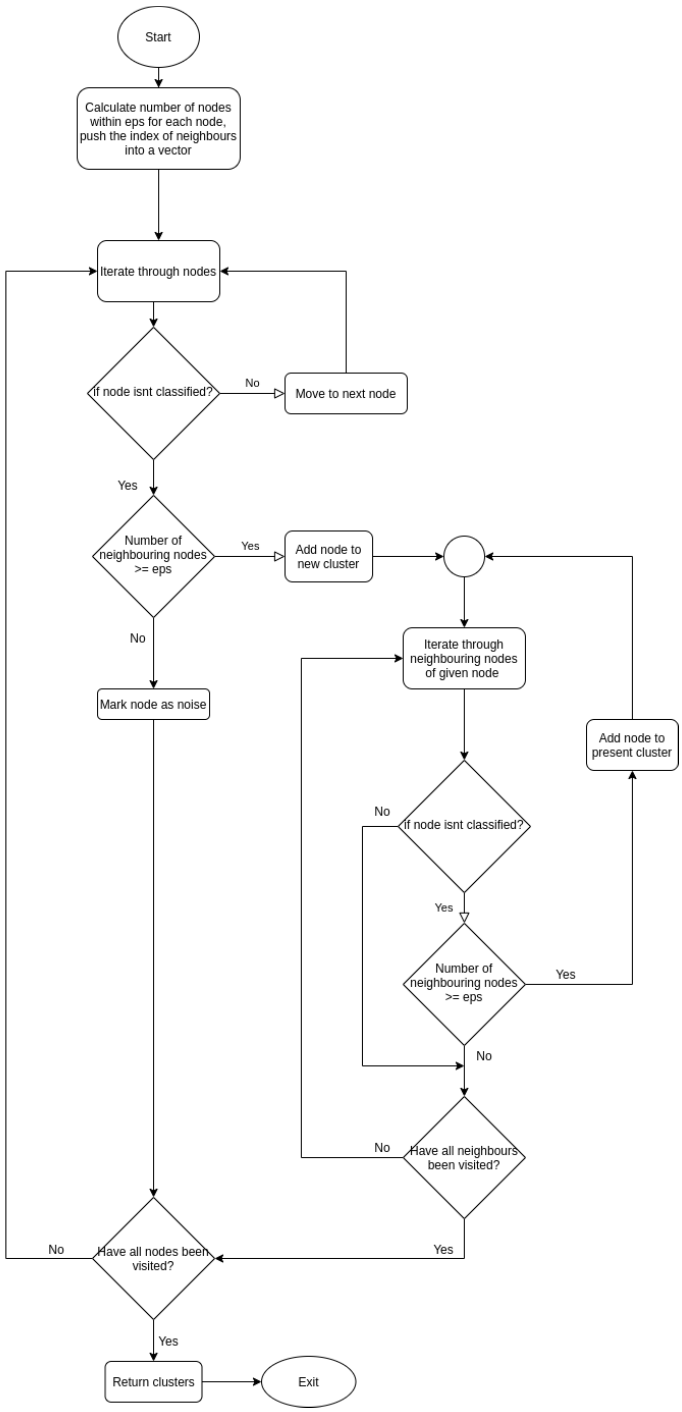


Fig. 2. DBSCAN FlowChart

D2D pairs to their respective SINR value. The SINR value of the destination of each D2D pair is taken and the value is calculated in decibels (dB), using the formula,

$$SINR = 10\log_{10}(sinr) \tag{2}$$

Here, *sinr* is the raw SINR value obtained from the text file

V. Results and Analysis

The following table, Table 3, gives an analysis of the DBSCAN vs the K-Means algorithm.

- The project uses DBSCAN because of its performance and accuracy when in the real world scenario when there's a lot of noise.
- The other clustering algorithms are severely affected by noise and work well for compact and well defined shapes but real life data may contain regularities.
- Due to the arbitrary shape of the clusters, and the data can contain noise, the DBSCAN algorithm is the best choice for real life applications.

TABLE II
DBSCAN vs K-Means

Sl. No.	DBSCAN	K-Means
1.	No need to specify number of Clusters	Need to specify number of Clusters
2.	Clusters may be of arbitrary shape and may not have same feature size.	Clusters should be approximately spherical/convex in shape and have same feature size.
3.	It efficiently handles outliners and noisy data sets.	It does not work well with outliners and noisy data sets.
4.	It does not efficiently scale for large or high dimensional data sets.	It works efficiently for large or high dimensional data sets.
5.	Does not work well for cluster datasets with large difference in densities.	Dataset of varying densities does not affect the it.

The results of the first experiment have been displayed in Fig. 3 and Fig. 4. There is one enB, two CU nodes and twenty UE nodes that are capable of D2D communication.

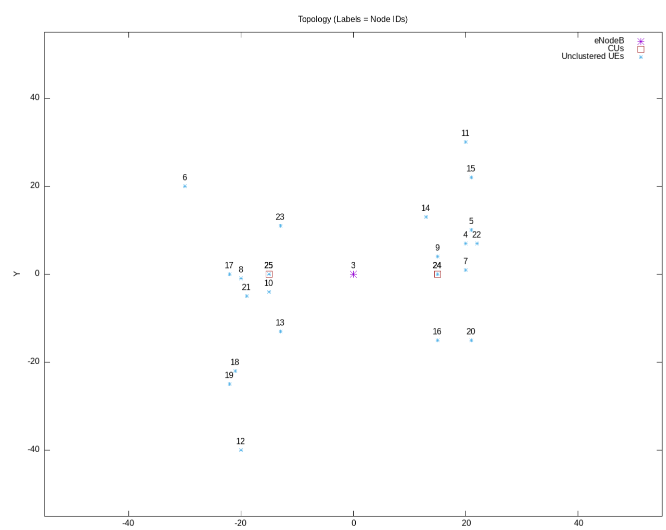


Fig. 3. Topology before sharding

TABLE III
SINR RESULTS

SI.No	Seconds	Active D2D Pairs	SINR @ The Destination (dB)
1	2.5	D2D_1	32.9448
2	3.5	$\text{D2D}_1, D2D_2$	32.9448, 32.2535
3	4.5	$\text{D2D}_1, D2D_2, D2D_3$	32.9448, 32.2535, 33.4119
4	5.5	$\text{D2D}_1, D2D_2, D2D_3, D2D_4$	32.9448, 32.2535, 33.4119,31.0695
5	6.5	$\text{D2D}_1, D2D_2, D2D_3, D2D_4, D2D_5$	32.9448, 32.2535, 33.4119, 31.0695, 34.0933

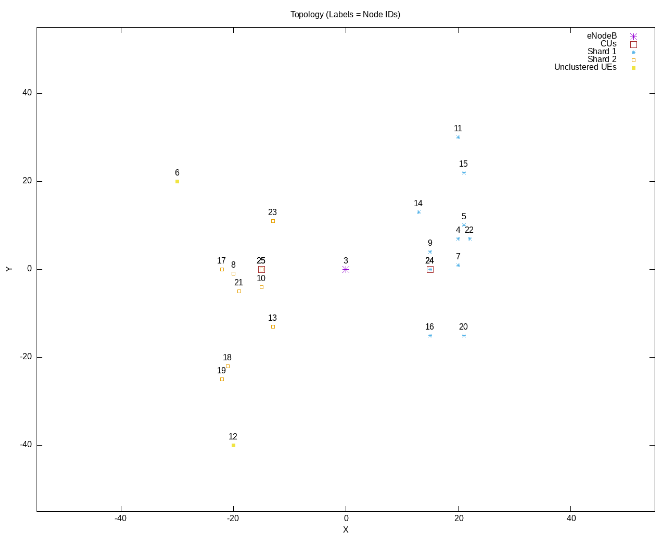
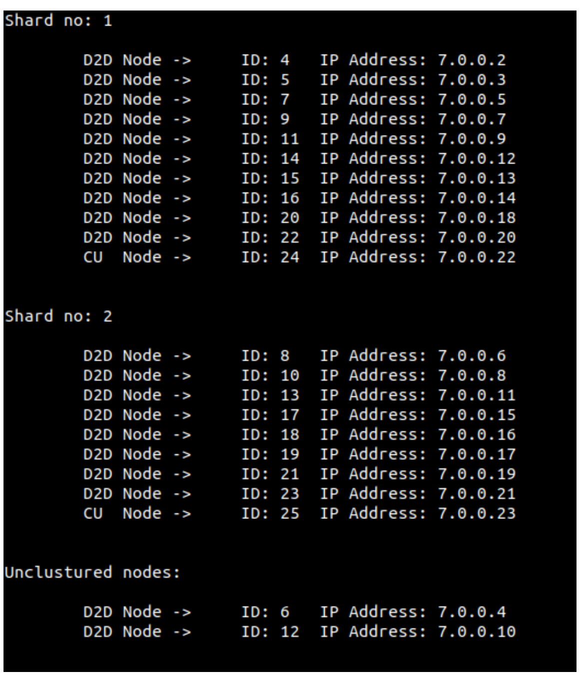


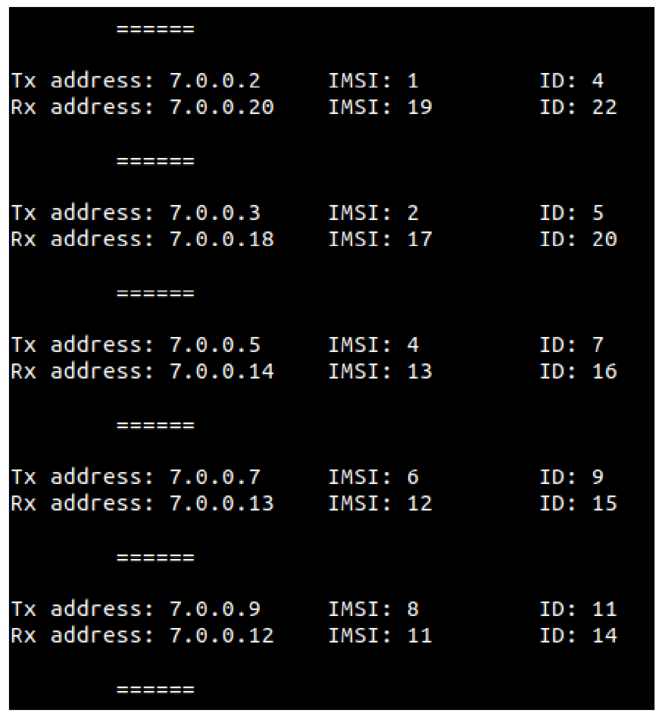
Fig. 4. Topology after sharding



```
Shard no: 1

        D2D Node ->     ID: 4   IP Address: 7.0.0.2
        D2D Node ->     ID: 5   IP Address: 7.0.0.3
        D2D Node ->     ID: 7   IP Address: 7.0.0.5
        D2D Node ->     ID: 9   IP Address: 7.0.0.7
        D2D Node ->     ID: 11  IP Address: 7.0.0.9
        D2D Node ->     ID: 14  IP Address: 7.0.0.12
        D2D Node ->     ID: 15  IP Address: 7.0.0.13
        D2D Node ->     ID: 16  IP Address: 7.0.0.14
        D2D Node ->     ID: 20  IP Address: 7.0.0.18
        D2D Node ->     ID: 22  IP Address: 7.0.0.20
        CU  Node ->     ID: 24  IP Address: 7.0.0.22


Shard no: 2

        D2D Node ->     ID: 8   IP Address: 7.0.0.6
        D2D Node ->     ID: 10  IP Address: 7.0.0.8
        D2D Node ->     ID: 13  IP Address: 7.0.0.11
        D2D Node ->     ID: 17  IP Address: 7.0.0.15
        D2D Node ->     ID: 18  IP Address: 7.0.0.16
        D2D Node ->     ID: 19  IP Address: 7.0.0.17
        D2D Node ->     ID: 21  IP Address: 7.0.0.19
        D2D Node ->     ID: 23  IP Address: 7.0.0.21
        CU  Node ->     ID: 25  IP Address: 7.0.0.23


Unclustured nodes:

        D2D Node ->     ID: 6   IP Address: 7.0.0.4
        D2D Node ->     ID: 12  IP Address: 7.0.0.10
```

Fig. 5. Sharding

Fig. 3 shows the topology before sharding. The algorithm divides all the nodes into two shards, one on the left and the other on the right based on density as shown in Fig. 4. Two nodes, with Node IDs, 6 and 12, which are away from the denser lot of nodes are considered to be noise and are not included in any shard by the algorithm. Fig 5 displays the terminal output for the same.



```
        ======

Tx address: 7.0.0.2     IMSI: 1         ID: 4
Rx address: 7.0.0.20    IMSI: 19        ID: 22

        ======

Tx address: 7.0.0.3     IMSI: 2         ID: 5
Rx address: 7.0.0.18    IMSI: 17        ID: 20

        ======

Tx address: 7.0.0.5     IMSI: 4         ID: 7
Rx address: 7.0.0.14    IMSI: 13        ID: 16

        ======

Tx address: 7.0.0.7     IMSI: 6         ID: 9
Rx address: 7.0.0.13    IMSI: 12        ID: 15

        ======

Tx address: 7.0.0.9     IMSI: 8         ID: 11
Rx address: 7.0.0.12    IMSI: 11        ID: 14

        ======
```

Fig. 6. D2D Pairs

Fig. 6 shows all D2D pairs. The D2D pairs become active sequentially, at regular intervals of one second, starting at 2.5 seconds. The IDs displayed are used to identify the D2D pairs in the output, and the IMSI is used to identify nodes from the *DlRsrpSinrStats.txt* that's generated.

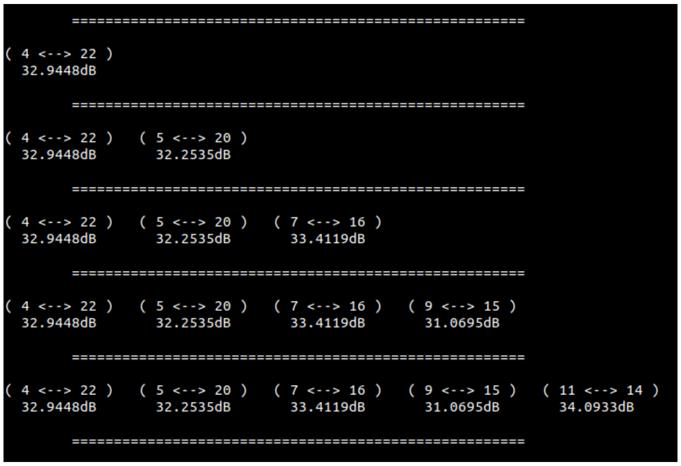


Fig. 7. SINR Result

Fig. 7 displays terminal output for the second part of the project. The SINR value for the destination node of each D2D pair is displayed right below the pair. The pairs are identified using the IDs of each node in the pair. The result is also summarised in Table 3. Any value of SINR above 30dB is considered to be good and signifies a good signal strength.

VI. Conclusion

This project explored the working of Public Safety Commercial Applications with ns-3 for sharding the available D2D pairs in such a manner that each shard contains atleast one Cellular User and any number of D2D pairs. DBSCAN, a density-based Clustering Algorithm was used to shard/cluster the nodes. Gnuplot library was used to generate the topology of the network. SINR values for the D2D pairs in a shard were calculated with the help of *SideLinkTraces*, using which the SINR values for all of the nodes in the network were obtained. The receiving node's SINR values after each interval were then converted to decibels and printed. DBSCAN algorithm is not suitable for higher dimensional data and data with varying density. Future work would include addressing these issues, and extending the algorithm to efficiently cluster data with higher dimensions and varying density. Excellent result was obtained in SINR, yet, more work is required on making this even better ensuring interference and noise is reduced as much as possible. The nodes were clustered based on density. The algorithm can be extended to take the application also into consideration during clustering. Public Safety Applications can be given more priority during communication.

References

[1] H. Kopka and P. W. Daly, *A Guide to LaTeX*, 3rd ed. Harlow, England: Addison-Wesley, 1999.

[2] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. *A density-based algorithm for discovering clusters in large spatial databases with noise.* In Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD'96). AAAI Press, 226–231.

[3] Maryam Hajjar, Ghadah Aldabbagh, Nikos Dimitriou, *Using Clustering Techniques to Improve Capacity of LTE Networks.* Kyoto, Japan: 21st Asia-Pacific Conference on Communications (APCC) - APCC2015 Special Technical Session, 2015.

[4] W. L. Tan, P. Hu and M. Portmann, *Experimental evaluation of measurement-based SINR interference models.* San Francisco, CA: 2012 IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2012.