

# Sharding based on application type of the D2D user and calculation of SINR

Kausthub Thekke Madathil - 191IT125  
Information Technology  
National Institute of Technology Karnataka  
Surathkal, India 575025  
Email: kausthubtm.191it125@nitk.edu.in

Puttaraja - 191IT139  
Information Technology  
National Institute of Technology Karnataka  
Surathkal, India 575025  
Email: puttaraja.191it139@nitk.edu.in

Sohanraj R - 191IT149  
Information Technology  
National Institute of Technology Karnataka  
Surathkal, India 575025  
Email: sohanrajr.191it149@nitk.edu.in

Neeraj Mirji - 191IT232  
Information Technology  
National Institute of Technology Karnataka  
Surathkal, India 575025  
Email: neerajmirji.191it232@nitk.edu.in

**Abstract**—A lot of disasters and natural calamities are happening now a days and their rate is also increasing day by day. A lot of people also die in this disasters because they don't come to know about these natural calamities before hand only. Therefore during such incidents it is fundamental to provide the first responders high performance and reliable communications, in order to improve their coordination capabilities and their awareness of the surrounding environment and allow them to promptly transmit and receive alerts on possible dangerous situations or emergency. Therefore D2D communication was introduced to carry out public safety communications at a faster rate. But now a days D2D communication is used for many other purpose as well. Thus base station has to check the purpose and then allocate resources for each D2D pair it consumes time and hence delays the communication process. So here we have come up with a novel idea that facilitates the public safety communications and the same we have implemented with the help of Network simulator 3. We have basically grouped all D2D users carrying out a public safety communication under a group and rest others into the other group using k -means algorithm. Therefore if a public safety communication is going on the eNB can easily come to know at the earliest and can give more reference to those communications by allotting best possible resources to carry out such communications.

## I. INTRODUCTION

### A. What is d2d communication ?

D2D communication is a new technology that enables direct communication between nearby devices without the involvement of the base station in the communication directly. It was introduced in the 4G itself but has become one of the pillar in 5G networks. It was basically introduced for Public safety application based communication but nowadays, almost all other fields also use d2d communication. Here unlike traditional communication, eNB is not directly involved in the communication i.e the communication link is not established by the base stations. It only allocates the resources and hence facilitate the communication. D2D communication was introduced under proximity services under the 3rd Generation

Partnership Project. It is also called as side link communication. Some of the D2D communication techniques are as follows

- Bluetooth 5.
- Wifi Direct
- LTE Direct

### B. What is sharding ?

Sharding is actually a DBMS based term which basically means clustering. i.e grouping things together. The word "Shard" means "a small part of a whole". Hence Sharding means dividing a larger part into smaller parts.

### C. What is SINR ?

SINR is basically Signal to Interference plus noise ratio. It is also sometimes called SNR which is Signal to Noise ratio. SINR is used to give theoretical upper bounds on channel capacity in wireless communication system. SINR is commonly used in wireless communication as a way to measure the quality of wireless connections. It has got more importance when it comes to wireless communication because in wireless communication signals are transmitted in all 360 degrees as there is no dedicated path. So lot of interferences happen between various signal leading to addition of noise. SINR basically gives the amount of unwanted interference and noise added to the signal.

### D. What is ns3-psc ?

1) NS3: ns3 is a discrete Network Simulator which is primarily developed for research and educational use. It has got powerful tools that help to visualize the simulations. It's a open source project and a free software. It is been made publicly available for research and development Ns3 is the third version of the network simulator. Ns 3 helps in implementing many real life based network protocols and it supports both ip and non ip based networks. In short Ns3 is

a boon for today's research and development.

2) *PSC*: PSC is basically Public Safety Communication which involves sharing of critical information to other fellow people. It also includes providing of timely event details to resident and community members to ensure their safety. If there is any public safety communication going, that communication should be given more priority as compared to other communication. They are the first responders. D2D communication was basically introduced to felicitate public safety communications.

## II. PROBLEM STATEMENT

Enb can be called a dumb device as it cannot keep on checking each request from the UEs to allocate resources. Hence it allocates similar resources irrespective of the type of application. So a device which is using social media is given similar preference as that of a public safety device, which must not be the case. So, by sharding/clustering devices based on application type, enb can easily differentiate between them and also tell from which shard the request is coming from. Hence the enb can easily allocate resources accordingly and the public safety devices can be given more priority.

### A. Objectives

The objectives of this project are mentioned below:

- To shard d2d devices based on their application
- To calculate the SINR values of the UE devices in the newly created shards

## III. METHODOLOGY

To achieve the above mentioned objectives we have used an open source project NS3-PSC which is basically used for network simulation. We have built this project keeping 'lte-sl-in-convrg-comm-model.cc' as the base code and we have built our code upon this file.

### A. Sharding based on application type :

We have created classes for d2d devices (d2d\_user) and CU devices (cu\_user) where we have defined attributes like 'position', 'application\_type' and 'cluster\_id' so that we can cluster the d2d users based on the application type. Fig.x shows the code for the classes. The sharding method that is proposed here can be divided into three steps;

1) *Differentiating d2d users based on application*: Every d2d device is given an application type randomly using the function rand(). Here in the code we have considered '1' for Social Media applications and '2' for Public safety applications. Later these d2d devices are sorted based on application types into two groups ( (i) Social Media (ii) Public Safety).

```
for all d2d_devices
{
    if (d2d_devices.application_type == 1)
        add it to the Social Media group;

    if (d2d_devices.application_type == 2)
        add it to the public safety group;
}
```

Fig. 1: Pseudo code for grouping

2) *Perform k-means for clustering*: Next we perform k-means on each group individually so that we get clusters/shards where d2d devices belong to the same application type. The k-means algorithm creates k centers/centroids i.e, k clusters which is returned back to the main function from which the information (cluster\_id) is stored along with the centers. The k-means algorithm's pseudo code is shown in Fig.2

```
create random centers;
do
{
    for all d2d_devices in a group
    {
        find the shortest center from the d2d_device;
        assign cluster id for the d2d_device;
        update the centers;
    }
} until( current centers == previous centers)

return centers;
```

Fig. 2: Pseudo code for k-means

3) *Assigning a CU device to each cluster*: Each cluster is assigned with a CU device. The assigning process is done using distances i.e. the closest CU device is assigned to a particular cluster and we have also ensured that the same CU device is not assigned twice i.e to two clusters.

After the sharding process is complete let us assume there are N D2D pairs (D2DN) in a shard where N ranges from 1, ..., N. The D2D pairs start communication in regular intervals i.e, the D2D1 pair starts data transfer at 2 second, the D2D2 pair starts data transfer at 3 second, the D2D3 pair starts data transfer at 4 seconds etc.

### B. SINR

SINR is basically Signal to Interference plus noise ratio. It is also sometimes called SNR which is Signal to Noise ratio. SINR is used to give theoretical upper bounds on channel capacity in wireless communication system.

$$Precision = \frac{SignalPower}{Noise + InterferencePower} \quad (1)$$

```

void PhySnirTrace (std::string context ,uint16_t cellId, uint16_t rnti, double rsrp, double sinr, uint8_t componentCarrierId)
{
    // the code goes here
}

Int main()
.....

Config::Connect
("/NodeList/*/DeviceList/0/$ns3::LteUeNetDevice/ComponentCarrierMapUe/*/LteUePhy/ReportCurrentCellRsrpSinr",
MakeCallback (&PhySnirTrace));

```

Fig. 3: code snippet for calculating SINR

We selected a particular shard for which we calculated the SINR values. SINR values of the nodes are calculated for the specific time mentioned in the question. For calculating the SINR values we have used the 'ReportCurrentCellRsrpSinr' which comes under LteUePhy which is written just after the simulation stops. This invokes a function called 'PhySnirTrace' which receives SINR as one of the parameters. This function is defined just above the main function. The code snippet of the same is shown in Fig.3.

#### IV. RESULTS AND ANALYSIS

When we run the project.cc file using command ./waf – run scratch/project , all the created D2D users will be assigned an application type based on which it will create two groups and in each of the group k means will be implemented to cluster the d2d users into smaller groups called shards. This happens in both the groups respectively. At last Cu's are also added to the shard on the basis of the nearest distance to shards created. The code produces the sharding output which is shown in Fig.4.

When the communication is initiated in shard two of the public safety group which has two d2d pairs, the function keeps on getting called again and again but gets executed at the following time intervals and prints the sinr value for d2d users as shown.

```

Time : 2.5 d2dNode : UE4
Time : 3.5 d2dNode : UE4 , UE6
Time : 4.5 d2dNode : UE4, UE6
Time : 5.5 d2dNode : UE4, UE6.

```

The calculated SINR is displayed as shown in Fig.5

#### V. CONCLUSION

The submitted NS3 code successfully first groups all the available d2d users on application based and small clusters will be created in each of the group. Now eNB can easily allocate resources to all d2d users on the basis of cluster id. If a D2D users is carrying out a public safety communication

then eNB will easily come to know based on the cluster id and hence they will be given more priority as compared to others. This saves a lot of time and also makes the communication process much faster and also serves the purpose of d2d communication.

The submitted code also calculates signal to noise plus interference ratio for all d2d users which receive the packets and hence gives the amount of unwanted interference and noise added to the signal during the communication process and thus helps in knowing the upper bounds on channel capacity.

#### INDIVIDUAL CONTRIBUTION

- **Kausthub Thekke Madathil (191IT125)** : Coding, code documentation, ideas, Report
- **Neeraj Mirji (191IT232)** : Coding, code documentation, ideas, Report
- **Puttaraja (191IT139)** : Coding and code documentation
- **Sohanraj R (191IT149)** : Coding and code documentation

The Gantt chart is shown in Fig.6.

#### IMPLEMENTED/BASE PAPER

##### REFERENCES

- [1] H. Kopka and P. W. Daly, *A Guide to L<sup>A</sup>T<sub>E</sub>X*, 3rd ed. Harlow, England: Addison-Wesley, 1999.
- [2] lte-sl-in-converge-comm-model.cc
- [3] <https://github.com/usnistgov/psc-ns3/blob/master/psc-3.0/ns3-psc-documentation.pdf>
- [4] <https://www.nsnam.org/docs/release/3.10/manual/html/lte.html>
- [5] [https://www.nsnam.org/doxygen/lte-ue-phy\\_8cc\\_source.html](https://www.nsnam.org/doxygen/lte-ue-phy_8cc_source.html)
- [6] <https://reasonabledeviations.com/2019/10/02/k-means-in-cpp/>
- [7] <https://www.youtube.com/watch?v=TuETIX7M2cY>

```

kausthub@kausthub-Inspiron-5570:~/psc-ns3-3.0.1$ ./waf --run scratch/project
Waf: Entering directory `~/home/kausthub/psc-ns3-3.0.1/build'
[2949/2999] Compiling scratch/project.cc
[2959/2999] Linking build/scratch/project
Waf: Leaving directory `~/home/kausthub/psc-ns3-3.0.1/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (8.353s)

*****
                        SOCIAL_MEDIA
*****

Shard 1 = {
    UE0 , IP_Address : 7.0.0.5
    UE1 , IP_Address : 7.0.0.6
    UE4 , IP_Address : 7.0.0.9
    UE5 , IP_Address : 7.0.0.10
    UE8 , IP_Address : 7.0.0.13
    UE9 , IP_Address : 7.0.0.14
    CU3 , IP_Address : 7.0.0.4
}

*****
                        PUBLIC_SAFETY
*****

Shard 1 = {
    UE2 , IP_Address : 7.0.0.7
    UE3 , IP_Address : 7.0.0.8
    CU1 , IP_Address : 7.0.0.2
}

Shard 2 = {
    UE6 , IP_Address : 7.0.0.11
    UE7 , IP_Address : 7.0.0.12
    CU2 , IP_Address : 7.0.0.3
}

*****
kausthub@kausthub-Inspiron-5570:~/psc-ns3-3.0.1$ 

```

Fig. 4: Screen shot of the output (sharding)

```

*****
SINR values for Public safety devices (Shard 2)
*****

UE4 belonging to D2D1
Time : 2.50021
RNTI : 4
rsrp : 2.11646e-11
SINR : 44619

UE4 belonging to D2D1
Time : 3.50021
RNTI : 4
rsrp : 2.11646e-11
SINR : 44619

UE6 belonging to D2D2
Time : 3.50021
RNTI : 15
rsrp : 1.87728e-12
SINR : 3957.66

UE4 belonging to D2D1
Time : 4.50021
RNTI : 4
rsrp : 2.11646e-11
SINR : 44619

UE6 belonging to D2D2
Time : 4.50021
RNTI : 15
rsrp : 1.87728e-12
SINR : 3957.66

UE4 belonging to D2D1
Time : 5.50021
RNTI : 4
rsrp : 2.11646e-11
SINR : 44619

UE6 belonging to D2D2
Time : 5.50021
RNTI : 15
rsrp : 1.87728e-12
SINR : 3957.66
kausthub@kausthub-Inspiron-5570:~/psc-ns3-3.0.1$ 

```

Fig. 5: Screen shot of the output (SINR)

Task ID	Task Description	Task Duration	Start Date	End date	11/10/2020	11/14/2020	11/16/2020	11/17/2020	11/19/2020	11/20/2020	11/23/2020	11/24/2020	11/25/2020	11/26/2020	11/27/2020	11/28/2020
1	Getting familiarize with LTE documentation and code	4	11/10/2020	11/14/2020												
2	Finding all the necessary docs and papers	2	11/14/2020	11/16/2020												
3	Understanding model1.cc and Kmeans Clustering	2	11/17/2020	11/19/2020												
4	Developing script for creating shards	3	11/20/2020	11/23/2020												
5	Analysing results of clustering	1	11/24/2020	11/25/2020												
6	Modified cluterling algorithm ( included kmeans)	1	11/25/2020	11/26/2020												
7	Referring various source files and video to develop script for SINR	1	11/25/2020	11/26/2020												
8	Modified script and including snippet for SINR	2	11/25/2020	11/27/2020												
9	Report Prepartion and compilation	1	11/27/2020	11/28/2020												

Fig. 6: ganttchart