Privacy Enabled Noise Free Data Collection in Vehicle Networks

MS Thesis Proposal

Anuj Dimri
School of Computing
University of Utah

Committee Members
Prof. Sneha Kasera (Chair)
Prof. Neal Patwari
Prof. Aditya Bhaskara

21 December 2017

Abstract

Preserving the privacy of a user who is collecting data is very important. Adding noise to the collected data can preserve the privacy but often leads to decrease in utility. If data is collected without adding noise then there would be no decrease in utility. In future, connected vehicles would become common and therefore could serve as a potential source of collecting large amounts of data. As part of this thesis we look into the aspect how noise free data can be collected in a privacy preserved manner in a vehicle network. We try to preserve privacy of the user by causing temporal and spatial variations using Random Delays and Indirections and collect the data in a noise free form. We take advantage of the mobility of the vehicles so that the adversary is unable to localize the users who collected or uploaded the data samples. We rely on vehicle and network simulators for trying different experiments and validating our approaches.

Contents

1 INRODUCTION

Many networked users and devices are interested in participating in distributed sensing and data collection for the purpose of betterment of human society or for earning some monetary rewards. For example, a car driver can report the current state of traffic, speed of vehicle, etc which would be beneficial to build a dynamic congestion map of a city. This information could further be used for mitigation purposes like routing the vehicles through less congested areas of a city thereby avoiding grave traffic congestion situations in a city. However, these participating nodes are concerned about their privacy and do not want the data collected by them to be associated with their identities. In many cases, a participating node reports its identity, its location, and its measurement (could be radio measurements, car velocity measurements, etc) to a central coordinator which then makes this data available for different applications. Such a data collection system does not preserve the location privacy of the participating users. If an adversary can correlate the data collected by a user with their identity then the adversary can know a lot of things about a user like driving patterns, places visited frequently and can even potentially track them. This is a serious privacy issue and users would not participate in data collection if there are no privacy guarantees. To preserve privacy of users researchers have used the concept of pseudonym ID's [12], group pseudonym and their extensions [11] but all of them have drawbacks as an adversary can still correlate the data collected with individual users. Another approach to preserve privacy is to add noise to the collected data [10] but this often leads to decrease in utility.

With all the major automobile companies diving into the business of connected and autonomous cars soon we would see a lot connected cars around us. A connected car has an internet connection and is able to communicate with the neighboring vehicles [1] by forming a vehicular network. These vehicular networks can serve as a potential source of collecting large amounts of data. The protocol used to make the vehicles communicate with each other is Dedicated Short Range Communication (DSRC). It operates in the 5.9GHz band and it's range is 1000 meters [7]. However, the range depends on the type of application and typically lies between 100-300 meters [8]. According to the US standards, vehicles participating in V2V (Vehicle to Vehicle) safety applications belong to Class C vehicles which have a communication range of 400 meters [8].

1.1 Motivation

If the privacy is preserved without adding noise in the data samples then there would not be any drop in the utility. In future, more and more vehicles would be able to communicate with each other and upload the data to a server. The server over a period of time can learn about a user thereby causing privacy issues. We specifically look how we can collect data without adding noise and preserve the privacy also in case of vehicle networks where the participating nodes are mobile. To solve this, we can take advantage of the mobility of vehicles as they move at a fast pace. We introduce temporal and spatial variations using the concept of Random Delays and Indirections in the data collection process itself. If an adversary receives a data sample after some delay he/she cannot associate that sample with the vehicle which is currently at the location where the data was collected. This is because the vehicle which collected the data would have moved to a new location of which the adversary has no idea. This approach gives us a two fold benefit of preserving the privacy and getting the data samples in a noise free form.

1.2 Thesis Statement

In this Masters thesis we are looking into Privacy Enabled Noise Free data collection approach in Vehicle Networks by introducing temporal and spatial variations using the concept of Random Delays and

Indirections.

1.3 Research Contribution

- Introduce Random Delays and Indirections in the data collection process.
- Implement different ways to cause Indirections and experiment with different Random Delays.
- Run simulations based on above techniques to get vehicle traces.
- Based on application type explore the trade-off between privacy and total delay which is acceptable for the message to be delivered.
- Evaluate our methods based on user defined privacy levels.

1.4 Challenges

- To simulate real vehicular environment and make the vehicles communicate with each other.
- To keep track of large number of vehicles and the messages which are being transmitted.
- Scaling issues due to broadcast nature of the protocol.
- Changing random delay and indirection parameters from a freeway/highway environment to a downtown environment.

1.5 Proposal Overview

The rest of the thesis proposal is organized as follows. Section 2 discusses the adversary model. Section 3 talks about our methodology, how we plan to preserve the privacy along with the privacy metrics. The simulation model is presented in Section 4 and Section 5 shows initial simulation results. Section 6 presents a timeline of future work and conclusion is presented in Section 7.

2 ADVERSARY MODEL

We assume that the central coordinator/server is the adversary. The adversary can read all the contents of the data which are uploaded to the server. The adversary also knows which particular user uploaded the data and the time at which the data was originally collected and uploaded to the server. In short the adversary knows the following-

(X,Y,t_0,t_{server},U,some measurements)

- X,Y - latitude, longitude where data was collected. In the rest of the proposal tuple of (X,Y) will be represented by Z.
- t_0-time at which data was collected.
- t_{server}-time at which data was uploaded to the server.
- U- Unique ID of the vehicle which uploads the data.
- Some measurements- application specific data like RSS values, speed of the vehicle, etc.

3 METHODOLOGY

3.1 Indirections

Indirections are used to introduce spatial variations. Each node has an option of uploading the data to the server with some probability or forward the data to one of its neighboring node. Let $p(I)$ be the probability of indirections. If $p(I) = 1$ then the node always uploads the data to the server. If $p(I) = 0.2$ then the node uploads the data to the server with probability 0.2 and with probability $1-p(I)$ it forwards the data to one of its neighbors. Thus the data would reach the server after multiple hops.

3.2 Random Delay

Random Delays are used to introduce temporal variations. The basic idea is not to send the data directly to the other node rather introduce some delay before sending. For each message hop this delay would be chosen between 1 to 'x' seconds. Because of this, when the data reaches the server after some time say 't', the original data collector could have traveled away from its original position in any direction and the data would be uploaded by some other node. Thus the adversary has no idea who the original data collector is. The adversary only knows the original location and time where the data was collected (Section 2), but does not knows who collected the data.

3.3 Problem Using Random Delay And Indirections

As the adversary is unaware of the location of the data collector, the adversary would try to localize the data uploader, say U. We assume that all the vehicles are able to communicate with each other. Privacy is lost if the server can localize U. The server adopts the following strategy to localize U-

- Server calculates the time difference $(t_{server} - t_0)$. This is the delay after which the data collected by the source node (S_0) at location Z and time t_0 reaches the server.
- The server calculates the distance that can be traveled in the above time difference at 'v' mph. (Server can find the value of 'v' based on the location where data was collected).
- Distance traveled by a vehicle is $r_0 = v * (t_{server} - t_0)$. We also need to take the range of DSRC protocol into consideration to calculate the value of 'r_0'. If the vehicles are moving in both the directions along a road network then the concept of random walk needs to be takes into consideration as the message can be transmitted in either direction of the traffic. Let us say that 'r' represents the distance traveled by the message packet after taking into consideration speed of vehicle, range of DSRC protocol and the concept of random walk.
- At time t_{server} the data uploader U can be anywhere in a circle of radius r centered at (Z).
- After 1 upload, the server knows that U can be anywhere inside the circle of radius r centered at (Z) and it can be any one of 'k' vehicles which are inside the above circle. If the area of the circle and 'k' are large then we can say that privacy is preserved of the uploader U.

But if the same uploader keeps uploading the data then the scenario changes. For each upload the server draws a circle as shown in Fig.1. When the server receives i^{th} data sample from the same uploader U it draws a circle C_i centered at Z_i. Based on the speed of the uploader and how frequently data samples are being uploaded the circles might intersect also.

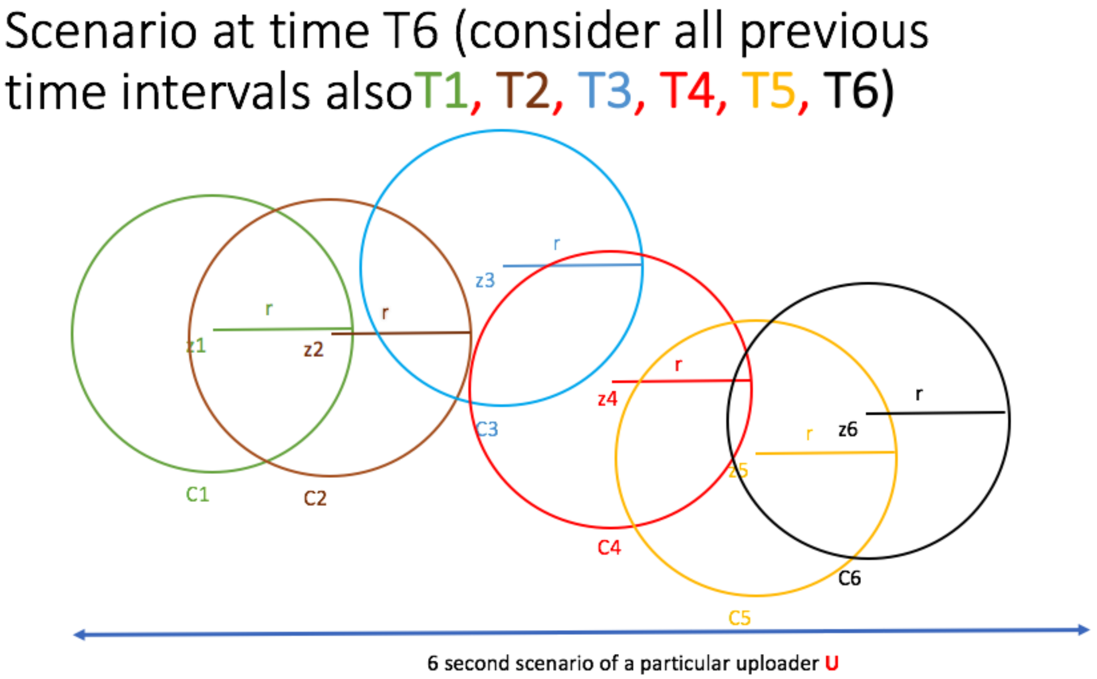

Scenario at time T6 (consider all previous time intervals alsoT1, T2, T3, T4, T5, T6)
r
z1
r
z2
z3
r
C3
r
z4
r
z6
r
z5
C1
C2
C4
C6
C5
6 second scenario of a particular uploader U

Figure 1: Scenario after 6 data uploads by a particular uploader U

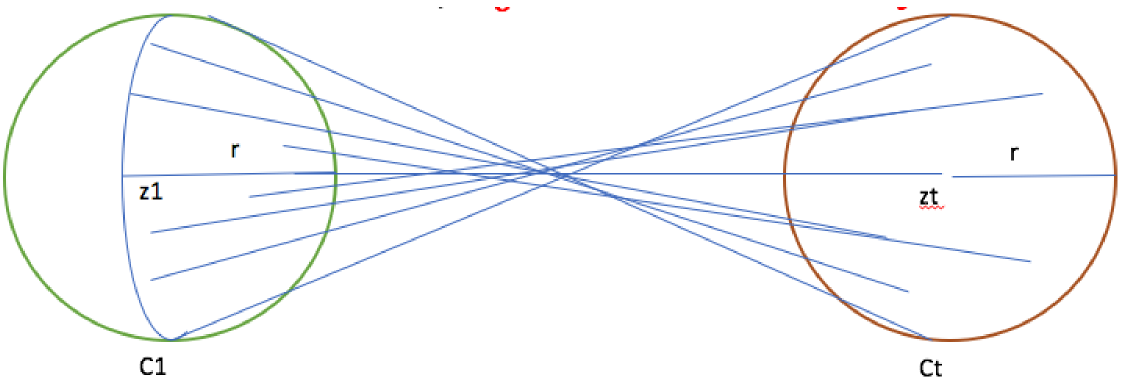


Figure 2: Set of possible points in C_1 which are at a distance of D_U^t from circle C_t.

Based on the speed of the uploader U and the number of times the upload was made by the same uploader (U) we can calculate the distance traveled by U in t seconds. Let this distance be D_U^t. At t^{th} second, circle C_t would be formed. So the uploader U could have traveled only D_U^t distance from circle C_t to C_1.

To calculate the speed of U when it uploads the i^{th} data sample, the adversary would take into consideration all the road segments which are inside circle C_i. The adversary would take note of the speed limits of the different roads which are inside C_i. The adversary now has a distribution of speeds and can take the value which has maximum probability. This is the speed of the uploader U.

Fig.2 shows the set of possible points in C_1 where the uploader U could have been if at present uploader U is in circle C_t. All the lines shown in the Fig.2 are of distance D_U^t. For simplicity, these lines are shown as straight lines but in reality these lines would lie on the road segments. These lines should pass through all the intermediate circles $(C_2, C_3, C_4 ... C_{(t-1)})$ because the uploader crossed these circles in the past. The general idea is that as U uploads more and more data (more circles will be formed), the number of possible points to reach a previous circle (C_1) keeps on decreasing. This can potentially be a privacy leak as the server would know where the uploader was in the past and can correlate the data received from that location in the past. In this way the adversary can know who was the data collector at any particular time and combine it with the data.

We aim to find for how long an uploader U can safely upload the data i.e. the adversary cannot localize U. As uploader U keeps on uploading the data the set of possible points where U could have been keeps on decreasing. After sufficient time say $(t + m)$ the set of possible points would be very less or below the privacy level set by the uploader U then it means that the adversary has localized uploader U. The uploader would stop uploading the data at time $(t + m)$ so that it does not looses it's privacy.

3.4 Privacy Evaluation Metrics

- **K-anonymity:** This is the most widely used privacy metric. It means that an individual can be any one of the 'k' people in a set [6]. The larger the value of 'k' is, more is the privacy guaranteed. In our problem 'k' is the number of different vehicles, so the uploader can be any one of the 'k' vehicles.

- **Area of region of interest:** With continuous uploads the area of consideration for the adversary keeps on decreasing. Based on the data received the adversary draws a circle where an uploader can be. Thus, after many uploads the area of the region becomes small thereby causing privacy issues.

Each user can define his/her own privacy level using the parameter ϵ. Lower the value of ϵ more is the privacy level. Based on the value of ϵ the value of 'k' in **k-anonymity** and **area of region of interest** also changes.

3.5 Algorithm

Algorithm 1 provides an approach where an uploader can safely upload the data to the server without any risk of its privacy being breached. If there is any threat to the uploader's privacy the uploader would stop uploading data to the server. The uploader has an option of defining it's own privacy level ϵ and as long as the ratio of number of possible points in the current set $(K_{common})$and number of possible point in the initial set $(K_{initial})$ is greater than privacy level (ϵ) set by the uploader the can safely upload the data. The ratio can also be calculated on the basis of the area of the region of interest as explained in Section 3.4

Algorithm 1 Privacy enabled approach

```
procedure MyProcedure
    $uploads \leftarrow$ number of uploads
    $V_U \leftarrow$ velocity of uploader (drawn from a distribution of speeds based
    on speed limits of the roads in the vicinity of Z)
    $D_U^t \leftarrow$ Distance traveled by uploader ‘U’ in ‘t’ seconds
    $K_{initial} \leftarrow$ Number of possible points in Circle 1 ($C_1$)
    $\epsilon \leftarrow$ User defined privacy level
    $t \leftarrow$ Current time
    while true do
        $++uploads$
        U uploads the data
        Server draws a circle of radius ’r’ based on $V_U$
        Calculate $D_U^t$
        $PossiblePoints$ = Set of possible paths of distance $D_U^t$ from circle t ($C_t$ to circle 1 ($C_1$) and passing through all the intermediate circles ($C_2, C_3, C_4 ... C_{(t-1)}$
        $K_{common}$ be the number of points in set $PossiblePoints$
        Ratio = $K_{common}/K_{initial}$
        if $Ratio <= \epsilon$ then
            STOP UPLOADING
            break
        $t \leftarrow t + 1$
```

4 SIMULATIONS

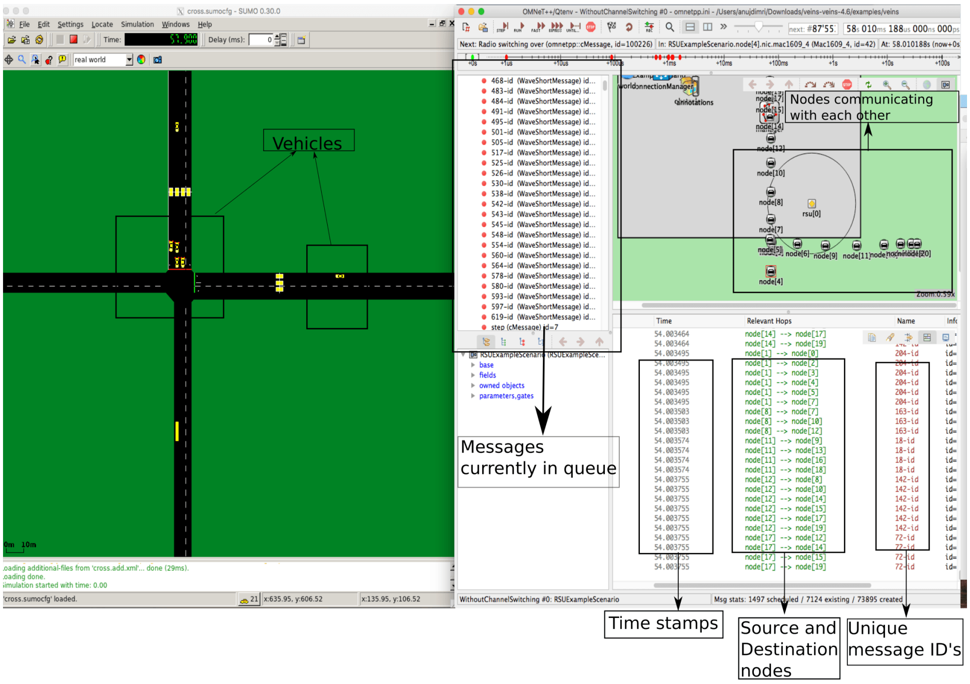
We are doing simulations to simulate the urban mobility environments where the vehicle move along the road segments, obey traffic rules, etc. We need to make the vehicles communicate with each other. The data needs to be sent to the neighboring nodes which are in the range of the vehicle. We follow DSRC protocol which is designed for making the vehicles communicate with each other. Next we need a framework to make the network and vehicular simulator talk to each other. We are using SUMO for vehicular simulation, OMNeT++ for network simulation and VEINS framework to make vehicular communication possible.

4.1 SUMO

SUMO stands for Simulation of Urban MObility[4]. It is used for vehicular simulations. It is equipped with a large variety of rich features, some of them are supporting a large number of fleet vehicles, traffic lights simulations, map generation, support for Open Street Maps, etc.

4.2 OMNeT++

OMNeT++ is an event based network simulator [3]. It is used for modeling both wired and wireless communication networks, protocol modeling, queueing networks,etc. OMNet++ provides infrastructure and tools for writing simulations.


Vehicles
Nodes communicating with each other
Messages currently in queue
Time stamps
Source and Destination nodes
Unique message ID's

```
468-id (WaveShortMessage) id...
483-id (WaveShortMessage) id...
484-id (WaveShortMessage) id...
491-id (WaveShortMessage) id...
495-id (WaveShortMessage) id...
501-id (WaveShortMessage) id...
505-id (WaveShortMessage) id...
517-id (WaveShortMessage) id...
525-id (WaveShortMessage) id...
526-id (WaveShortMessage) id...
530-id (WaveShortMessage) id...
538-id (WaveShortMessage) id...
542-id (WaveShortMessage) id...
543-id (WaveShortMessage) id...
545-id (WaveShortMessage) id...
548-id (WaveShortMessage) id...
554-id (WaveShortMessage) id...
560-id (WaveShortMessage) id...
564-id (WaveShortMessage) id...
578-id (WaveShortMessage) id...
580-id (WaveShortMessage) id...
593-id (WaveShortMessage) id...
597-id (WaveShortMessage) id...
619-id (WaveShortMessage) id...
step (cMessage) id=7
```

```
Time        Relevant Hops             Name
54.003464   node[14] --> node[17]
54.003464   node[14] --> node[19]
54.003495   node[1] --> node[0]       204-id
54.003495   node[1] --> node[2]       204-id
54.003495   node[1] --> node[3]       204-id
54.003495   node[1] --> node[4]       204-id
54.003495   node[1] --> node[5]       204-id
54.003495   node[1] --> node[7]       204-id
54.003503   node[8] --> node[7]       163-id
54.003503   node[8] --> node[10]      163-id
54.003503   node[8] --> node[12]      163-id
54.003574   node[11] --> node[9]      18-id
54.003574   node[11] --> node[13]     18-id
54.003574   node[11] --> node[16]     18-id
54.003574   node[11] --> node[18]     18-id
54.003755   node[12] --> node[8]      142-id
54.003755   node[12] --> node[10]     142-id
54.003755   node[12] --> node[14]     142-id
54.003755   node[12] --> node[15]     142-id
54.003755   node[12] --> node[17]     142-id
54.003755   node[12] --> node[19]     142-id
54.003755   node[17] --> node[12]     72-id
54.003755   node[17] --> node[14]     72-id
54.003755   node[17] --> node[15]     72-id
54.003755   node[17] --> node[19]     72-id
```

```
Msg stats: 1497 scheduled / 7124 existing / 73895 created
```

Figure 3: Vehicle simulator on left and network simulator on right.

4.3 VEINS

Veins is a framework used for vehicular network simulation [5]. It interacts with OMNeT++ and SUMO in parallel which are connected via TCP socket. As the vehicles move in SUMO the updates of their positions are made in the OMNet++ simulator as movement of nodes.

5 Initial Simulation Results

Fig.3 shows a running simulation. The left part of the image is the vehicular simulator where the vehicles are running on the roads and following traffic rules. The right part of the image is the network simulator where the nodes are moving and communicating with each other. The simulators are interacting with each other and the movement of the nodes in the network simulator is based on the movement of vehicles in the vehicle simulator. Fig.3 shows the messages which are currently scheduled but not delivered, timestamps, source and destination nodes and unique message ID's.

Fig. 4 shows the number of received and sent messages by each vehicle over a period of 1 minute. Due to the broadcast nature of DSRC protocol the number of messages received are greater than number of messages sent. The y-axis represent the number of messages and the x-axis represent different vehicles.

In the current simulation new messages are being generated and old messages are being forwarded. Random delay in the range (1-10 seconds) is being added before every message transmission. Whenever a new message is being created a unique ID is given to that message. But when the old messages are forwarded the unique ID does not changes so we can track the messages.

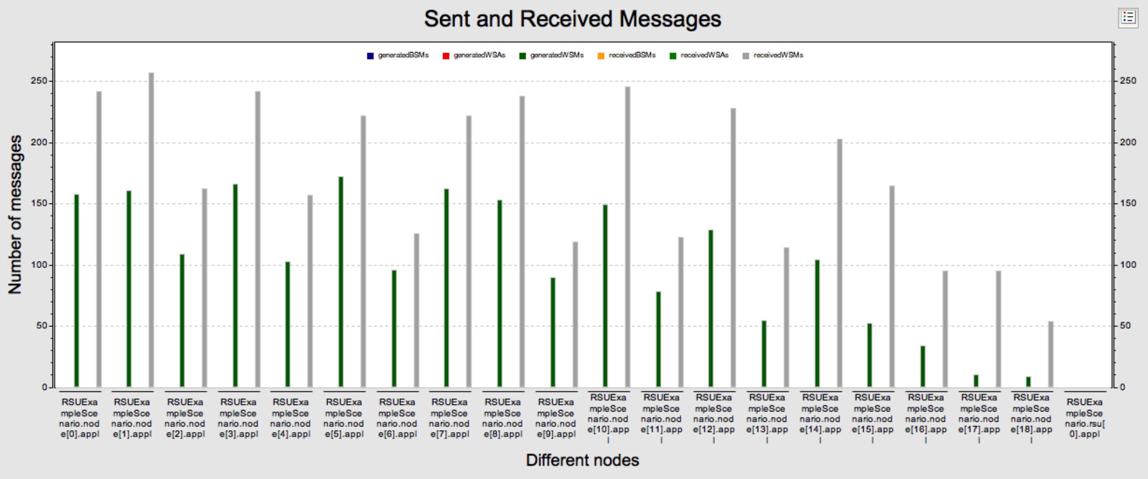


Figure 4: Number of messages sent and received by each node/vehicle.

Hop	Time (sec)	Random Delay (sec)
*node*0 creates message	10	1
$node0- > node3$	11	8
$node3- > node5$	19	6
$node5- > node8$	25	9
$node8- > node10$	34	3
$node10- > node12$	37	1
$node12- > node10$	38	2
$node10- > node14$	40	5
*node*14 uploads	45	-

Table 1: Example of random delay and indirections.

Due to the broadcast nature of DSRC protocol the message will be delivered to all the nodes which are within the range (range is set to 150 meters). But we consider only one of the neighbors for the transmission. Consider node1 has 4 neighbors - node2, node3, node4, node5. So node1 would send its data to all its 4 neighbors due to broadcast nature of DSRC protocol. But we would take only one of transmission under consideration out of all the 4 transmission. Let us walk through an example. The message under consideration has an unique ID - **'16-id'**. This message was created at t=10 seconds by node0.

The data which was collected by node0 at time t=10 seconds was uploaded to the server by node 14 at t=45 seconds after introducing random delays and indirections as shown in Table 1. The total delay is $45 - 10 = 35$ seconds. The vehicles are traveling at a speed of 25 meter/sec. So the distance covered in 35 seconds would be 875 meters. Thus the uploader can be at a distance of 875 meters from the point where data was originally collected. See Fig. 5 for details. The uploader can be anywhere on the roads inside the yellow boundary. Fig. 5 has 15 different vehicles inside the yellow box but in this case vehicles are running on only 2 roads and not on all 4 roads. **According to our privacy evaluation metrics we can say in this case the value of k (k-anonymity) is 15 and area of region of interest is**

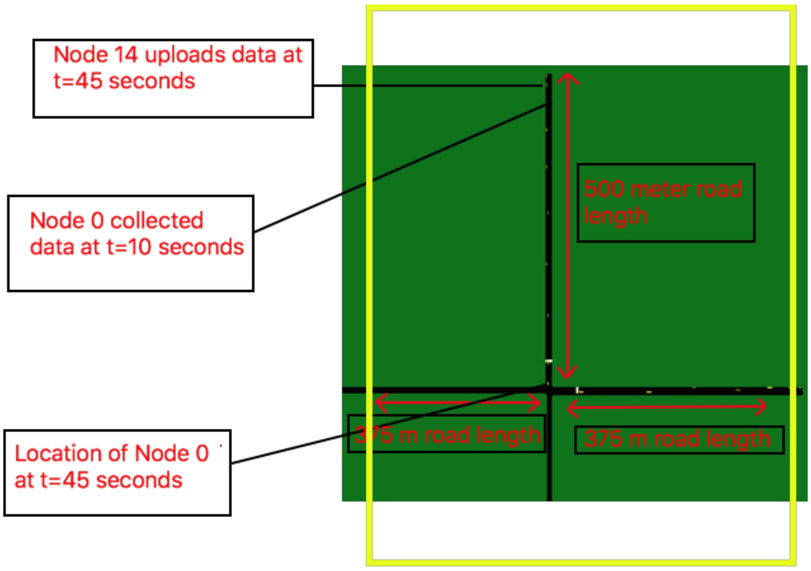

Node 14 uploads data at t=45 seconds
Node 0 collected data at t=10 seconds
Location of Node 0 at t=45 seconds
500 meter road length
375 m road length
375 m road length

Figure 5: Map showing time and location where a message was created and uploaded.

Milestone	**Expected Date of Completion**
Solve scalability issue	January 15*th*
Implement different Indirection approaches	February 15*th*
Experiment with freeway and downtown areas	March 1*st*
Evaluate Trade-off between privacy and total delay	March 15*th*
Introduce user define privacy level(ϵ)	April 1*st*
Thesis Writing	April 15*th*
Thesis Defense	April 25*th*

Table 2: Thesis Timeline.

875 meters. This is the scenario after only one upload. Next steps would be to see the scenario after multiple uploads.

6 Timeline

Timeline of the thesis is shown in Table 2.

7 Conclusion and Future Work

So far we have been successful in setting up the simulators and running the simulations. The vehicles are able to communicate with each other. At each second new messages are generated and old messages are also being forward. Due to the broadcast nature of the protocol all the neighboring nodes are receiving the messages. For indirections, we would choose any one of the neighbors and that neighbor has an option of uploading the message to the server or forward it to it's neighbors. Before each message transmission a random delay is being added. Initial results show that after one upload the adversary has a large area under consideration which has a lot other nodes also. But the adversary does not knows among these large number of nodes which is the uploader node and which is the source node. Future work, would be to find after how many uploads the adversary can localize the uploader. We would also experiment with different random delays and apply different algorithms to choose the neighboring nodes for indirections. Some of them can be to choose the neighboring nodes at random, choose the node which is traveling in the opposite direction of the source node, choose the node which is at the maximum distance from the source node, etc. It would be interesting to see how the scenario changes in case of freeways and downtown areas.

References

[1] 'Connected car,' Wikipedia, The Free Encyclopedia, "`https://en.wikipedia.org/w/index.php?title=Connected_car&oldid=813408145`", (accessed December 3, 2017).

[2] 'What's driving the connected car'. McKinsey & Company. "`https://www.mckinsey.com/industries/automotive-and-assembly/our-insights/whats-driving-the-connected-car`", (accessed December 3, 2017)

[3] OMNeT++ Network Simulator

[4] SUMO-Simulation for Urban Mobility

[5] 'Bidirectionally Coupled Network and Road Traffic Simulation for Improved IVC Analysis', Christoph Sommer, Reinhard German and Falko Dressler, *IEEE Transactions on Mobile Computing, vol. 10 (1), pp. 3-15*, January 2011.

[6] 'k-anonymity: a model for protecting privacy', L. Sweeney. *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems, 10 (5)*, 2002; 557-570.

[7] 'Vehicular Ad Hoc Networks and Dedicated Short-Range Communication', Jinhua Guo and Nathan Balon† University of Michigan - Dearborn

[8] 'Reliability Analysis of DSRC Wireless Communication for Vehicle Safety Applications', Fan Bai, H. Krishnan, *Intelligent Transportation Systems Conference, 2006*

[9] 'Dedicated Short-Range Communications (DSRC) Standards in the United States', Kenney, J. B, *Proceedings of the IEEE 99, no. 7 (2011)*

[10] 'Differential privacy: A survey of results', Cynthia Dwork, *International Conference on Theory and Applications of Models of Computation 2008*

[11] 'AMOEBA: Robust Location Privacy Scheme for VANET ', Sampigethaya, Krishna, et al., *IEEE Journal on Selected Areas in Communications 2007*

[12] 'Examining privacy in vehicular ad-hoc networks ', Da Silva, Daniel, et al.,*Proceedings of the second ACM international symposium on Design and analysis of intelligent vehicular networks and applications. ACM, 2012.*