

Efficient multi-hop connectivity analysis in urban vehicular networks



Mohammad A. Hoque^{a,*}, Xiaoyan Hong^b, Brandon Dixon^b

^a Department of Computing, East Tennessee State University, United States

^b Department of Computer Science, University of Alabama, United States

ARTICLE INFO

Article history:

Received 23 August 2013

Received in revised form 27 April 2014

Accepted 28 April 2014

Available online 5 May 2014

Keywords:

Vehicular network
Degree of connectivity
Network partitioning
Mobility
DSRC
Taxi

ABSTRACT

Vehicle to Vehicle (V2V) communication provides a flexible and real-time information dissemination mechanism through various applications of Intelligent Transportation Systems (ITS). Achieving seamless connectivity through multi-hop vehicular communication with sparse network is a challenging issue. In this paper, we have studied this multi-hop vehicular connectivity in an urban scenario using GPS traces obtained from San Francisco Yellow cabs. Our current work describes a new algorithm for the analysis of topological properties like connectivity and partitions for any kind of vehicular or mobile computing environment. The novel approach uses bitwise manipulation of sparse matrix with an efficient storage technique for determining multi-hop connectivity. The computation mechanism can be further scaled to parallel processing environment. The main contribution of this research is threefold. First, developing an efficient algorithm to quantify multi-hop connectivity with the aid of bitwise manipulation of sparse matrix. Second, investigating the time varying nature of multi-hop vehicular connectivity and dynamic network partitioning of the topology. Third, deriving a mathematical model for calculating message propagation rate in an urban environment.

© 2014 Elsevier Inc. All rights reserved.

1. Introduction

Over the past few years, connectivity in Vehicular Ad hoc Network (VANET) has been investigated by the researchers in different forms. One of the most common form of analysis is based on probabilistic modeling where some sort of simplified assumptions are made regarding the vehicular mobility pattern to determine the expected number of connected or reachable neighbors. This kind of analytical models show some properties of connectivity related to the overall traffic distribution and node density. Another type of analysis results from analyzing traces obtained from sophisticated vehicular traffic simulators where the robustness factor depends on the granularity of the microscopic mobility features. But still, this kind of microscopic mobility simulators fail even to capture the spatio-temporal variation of actual urban traffic. For this purpose, researchers are now more inclined towards utilizing real GPS trajectories from probe vehicles to capture the spatio-temporal characteristics of urban mobility patterns. But, problems still exist in this approach when the total number of probe vehicles are too small to be scaled to represent the mass traffic. However, if the number of probe vehicles is sufficiently large, for example if the entire fleet of public transportation or taxi cabs

are within the monitoring scope, this can essentially resemble the spatio-temporal features of the real time mass traffic. In addition, the calculations of vehicular connectivity entail complex computation using chain matrix multiplications. For example, every single moment the positions of the nodes change and form a new topology resulting into a different adjacency matrix. Existing algorithms for determining broadcast propagation rate in a vehicular network only account for connected network. In order to estimate message dissemination delay beyond the connected component using a store and forward mechanism, the algorithm needs to keep track of the changes in network partition due to the mobility of the vehicular nodes. This requires a more sophisticated algorithm that is capable to keep track of the newly connected nodes at every distinct time slot. From this perspective, ours is the first attempt to develop a new analytical approach and computational method for vehicular connectivity analysis using Boolean Matrix Multiplication.

On the other hand, Boolean Matrix Multiplication (BMM) has long been a topic of interest among the theoretical computer scientists. While researchers [28–30] have been trying to utilize BMM to determine transitive closure of a network since early seventies of the last century, not much effort has been put on leveraging the techniques of sparse matrix multiplication with BMM. As from practical sense, most vehicular ad hoc networks are sparse.

Hence, using dense matrix multiplication algorithms to determine connected components or transitive closures is obviously

* Corresponding author.

E-mail addresses: hoquem@etsu.edu (M.A. Hoque), hxy@cs.ua.edu (X. Hong), dixon@cs.ua.edu (B. Dixon).

inefficient. Our algorithm, apart from utilizing the techniques of sparse matrix multiplication, also considers an incremental approach for chain multiplication. This reduces the overall computational complexity for determining the transitive closure. Moreover, we reduce the storage cost by allocating only a single bit per each matrix element and increase the computational efficiency by incorporating bitwise operations on blocks of bits. We named this algorithm a Boolean Chain Matrix Multiplication (BCMM). From the perspective of vehicular communication and wireless networking, this is by far the first algorithm of this kind for analyzing multi-hop connectivity and network partitions.

It can be envisioned that in near future, enterprise business applications or commercial applications might be developed on top of DSRC platform targeting a particular class of vehicles in a specific geographical terrain. For example, a taxi cab company may use an internal fleetwide business application using V2V communication platform. Other examples of this type of selective multicast applications include commercial applications targeting vehicles of specific manufacturer or government entities trying to draw attention of a specific class of travelers, etc. Irrespective of the application scope, we present an efficient algorithm to determine the multi-hop connectivity within a large fleet of moving vehicles in a metropolitan area. We also described how this algorithm can be used to determine the network partitioning and broadcast propagation rate in a mobile taxi network. Our analysis results were based on live GPS traces obtained from the fleet of San Francisco yellow cabs [1]. The trajectories were made available through the Cabspotting project [3], a remarkable initiative of San Francisco Exploratorium [2]. The Cabspotting project is intended as a framework to help use the movement activity of commercial cabs to explore the economic, social, political and cultural issues that are revealed by the realistic GPS traces. Our analysis dealt with the entire fleet of 536 cabs generating over 10 million mobility traces within a period of one month. Our previous work [23] using these archived datasets showed interesting factors about taxi cab mobility, trip pattern, passenger hotspots, drivers empty cruise time and some basic analysis of data communication.

On the whole, majority of the prior researches on connectivity and partitioning were based on probabilistic modeling and simulating with mobility traces generated by well known traffic simulators, without using any real probe data or GPS traces. This makes our work different from all previous analysis. We also introduced the notion of saturated connectivity and a mathematical model to derive the message propagation rate in such a dynamic network. Our current work describes a complete method and step by step algorithm for the computation of topological characteristics in a highly dynamic environment. Moreover, ours is by far the first approach for determining multi-hop connectivity using sparse binary matrix manipulation with an efficient storage and computation mechanism. It has been already established by the researchers that the vehicular network can also act as a special form of Delay Tolerant Network (DTN) where information can be stored temporarily and forwarded as soon as a previously isolated node becomes reachable from a broadcasting node. Hence, in this way an emergency notification regarding traffic accidents or detour can be propagated throughout the entire urban metropolitan area. Here we investigated the propagation rate of such emergency messages within a taxi fleet covering a metro area. In addition, we also investigated the spatio-temporal behavior of network partitions and connectivity. The main contribution of this research is two fold. First, developing an efficient algorithm to quantify multi-hop connectivity with the aid of bitwise manipulation of sparse matrix. Second, investigating the time varying nature of multi-hop vehicular connectivity, dynamic network partitioning and message propagation in an urban environment. To the best of our knowledge, this is also the first approach to develop an efficient algorithm of

this kind which can be further scaled to parallel processing environment for performance improvement.

The subsequent sections are organized as follows: we discuss related work in Section 2, followed by our system model and data collection methodology in Section 3. Section 5 presents the steps of the proposed BCMM algorithm preceded by pre-processing of raw data in Section 4 and followed by applications of this algorithm in Section 7. Section 8 describes the details of results and analyses on vehicle connectivity and partitioning of the mobile nodes. Finally, we conclude in Section 9.

2. Related work

Several researchers came up with various algorithms and implementations of BMM [27–33]. Some of the recent researchers [27,34,35] introduced quantum computing based algorithm for BMM while some others [32,33,36] followed combinatorial algorithm. However, quantum computing is not implemented yet and still far from reality. The best combinatorial algorithm [36] so far gives the complexity of BMM bounded by $O(\frac{n^3}{\log^{2.25}(n)})$. Fischer and Meyer [28] theoretically showed that the upper bound of computational complexity for calculating transitive closure using BMM is $O(n^\alpha \cdot P(n))$ where $\alpha = \log_2 7$ and $P(n)$ is the number of bitwise operations needed. However, this algorithm is based on dense matrix multiplication and do not consider the sparseness of adjacency matrix for real network. Our BCMM algorithm is an incremental approach of computing chain multiplications of adjacency matrix to generate the transitive closure and multi-hop connectivity in a dynamic vehicular network.

Even though vehicular connectivity problem is an interesting topic for the VANET research community, but not much work has been done from the perspective of analyzing network partitioning and multi-hop reachability using real world traces. However, several analytical models have been developed using traffic simulators that shows the relationship between transmission range and node connectivity. Ukkusuri and Du [16] derived an analytical lower bound of average reachable nodes to maintain high connectivity, obtaining a relationship between the total node size and average number of reachable nodes. However, their analysis was based on a particular segment of freeway using data obtained from MITSIMlab, which is a traffic simulation framework. In our current work, we also derived the average number of reachable nodes from each node with an additional constraint added on the forwarding delay which includes the queueing, storing, transmission and propagation delay in each hop. Our traces are obtained from GPS trajectories of San Francisco Yellow Cabs covering the entire metropolitan area.

Ho et al. [20] utilized the Groovenet traffic simulator to generate traces of structured mobility and provided an analytical framework to investigate the k-hop connectivity in vehicular network. The authors also demonstrated the impacts of macro and micro mobility features on k-hop connectivity. They defined some useful metrics to evaluate the node connectivity in vehicular networks.

Fiore and Harri [22] studied the effects of node mobility on the topology of a vehicular network through comparative analysis between some of the well known stochastic and traffic stream mobility models. Their research dealt with the duration of peer-to-peer wireless links, node degree, number of partitions or connected components, average partition size, etc., in different kind of mobility. It is worth to mention that, our analysis also encompasses most these metrics but using real GPS traces instead of software generated trajectories.

Ferreira et al. [17] developed a framework named 'DIVERT' for large-scale traffic simulation and computation of node connectivity in vehicular sensor network. Using the DIVERT framework, the authors have demonstrated the temporal evolution of the average

degree of connectivity and presented an algorithm for computing the transitive closure to identify a connected component within the network. Their results from simulation traces show the same kind of trends that we obtained from our analysis on real traces, except that we had a very detailed picture of the connectivity evolution through hop-by-hop exploration. Also, our algorithm of generating the transitive closure and multihop connectivity is more efficient than theirs in terms of space-time complexity. Many researchers [18,19,21] also studied the variation of connectivity and component size (partition size) with respect to change of transmission range. Some of them [19] derived analytical models for connectivity simulating with CORSIM or other traffic simulators.

On the other hand, using real GPS traces, several interesting properties related to taxi mobility pattern has been investigated by the researchers. Most of these studies are based on analyzing archived traces from different taxi cab companies to explore hidden characteristics of urban mobility models. Some of these researchers tend to reveal new mobility models, taxi trips and usage pattern, etc., while others focus on clustering and hot spot identification.

Our previous work [23–26] based on the Cabspotting project described some statistical aspects of taxi mobility and trip patterns including instantaneous speed and directions, frequency of pickup and dropoff, distribution of hotspots, driver's empty cruise time, etc. Piorkowski et al. [6] utilized the Cabspotting data archived over a month to propose a parsimonious mobility model called Heterogeneous Random Walk (HRW) which captures some of the important mobility characteristics observed from the macroscopic level. A key feature of the model is that nodes follow independent and statistically equivalent mobility patterns, despite the presence of long-term clusters. They also evaluate the predictive power of the HRW model in the context of epidemic dissemination, which is one of the most prominent paradigms for routing in DTNs. Their work motivates the vehicular networking community to deeply investigate the taxi mobility traces for further research.

Shin and Park [5] used real-life location tracking data collected from the Taxi Telematics system developed in Jeju, Korea. Their analysis aimed at obtaining meaningful moving patterns of taxi cabs. They have extracted some interesting statistical factors such as taxi's driving type, driving time, driving area, pickup rate, trip duration, taxi usage ratio, service area and time, etc. Lee [4] analyzed a pick-up pattern of taxi service in the same geographical area aiming at clustering the pickup and drop off locations to develop a location recommendation service for empty taxis. The same author in another paper [6] analyzed both spatial and temporal statistics of taxi's waiting spots from the movement history. The authors also proposed an analysis framework or data model for taxi telematics system [14]. Again, Yang Yue et al. [13] described a deterministic single-linkage clustering algorithm and studied the spatio-temporal distribution of the taxi hotspots and travel interactions between these clusters. Their analysis resulted from as many as 596044 traces from the taxi cabs of Wuhan, China.

Cheng and Qu [15] proposed a service choice model for optimizing taxi service by reducing waiting time and maximizing revenue. Their analyses was done with the GPS traces of Singapore based taxi cabs. Bin Li et al. [11] explored a comparative analyses of different passenger-finding strategies using machine learning approach. Their work was based on more than two million traces generated by about five thousand taxi cabs of Hangzhou, one of largest cities of China. An interesting taxi mobility model has been proposed by Hongyu Huang et al. [12] using the GPS traces of more than 4000 taxis of Shanghai. The mobility model, named as META, considers the following parameters: turn probability, road section speed and travel pattern. The model generated traces were validated by real world traces and comparisons were made between two trace sets in terms of various features. These works provide

an insight to the possible dimensions of utilizing location tracking data for the purpose of taxi industry.

3. System model and data collection

The Cabspotting project tracks San Francisco's taxi cabs as they travel throughout the Bay Area. The data is transmitted from each cab to a central receiving station once in every minute, and then delivered in real-time to dispatch computers via a central server. This system broadcasts the cab call number, location and whether the cab currently has a fare. The cab locations are not stored by Yellow Cab, but only used in real-time to aid dispatch. Cabspotting server communicates to the Yellow Cab server and stores the data in a database, encoding the call number for privacy. The patterns traced by each cab create a living and always-changing map of city life. This project is intended for researchers to explore these issues in the form of a small experiment, investigation or observation. One of the most important component of this project is the API [7] that allows real time tracking information of individual cabs. Two other mentionable applications belonging to this project is the CabTracker [8] which averages the last four hours of cab routes into a map and the Time Lapse [9] which reveals time-varying patterns such as rush hour, traffic jams, holidays and unusual events.

3.1. Trace record

Each mobility trace record contains the following fields:

- (1) *Latitude & Longitude*: Two floating point values of the current GPS position of the cab.
- (2) *Occupancy status*: A binary value indicating the passenger occupancy status. A value of 0 indicates that the cab is free while 1 means hired by passenger.
- (3) *Timestamp*: Unix timestamp of the trace reception time.

3.2. Accumulation of trace records

Using the API we accumulated real time traces of these cabs over a time frame of more than 24 hours starting from July 17, 2011 11:01:09 PM to July 18, 2011 11:57:08 PM. A total of 2063 trace records was captured within this time frame. We also collected previously archived data for a period one month from CRAWDDAD [10] that was acquired through the same procedure. The archived records summed up to a total of more than 10 million traces organized in individual ASCII files for each of the 536 licensed yellow cabs. These trace files were simulated using our own developed application. We analyzed the traces both from the perspective of a single cab as well as from the perspective of the whole fleet.

3.3. Calculation of geographical distance

Previous work with GPS trace data and distances mostly considered Euclidian distance between two points. However, this calculation completely ignores the fact that the earth is round yielding incorrect results. The difference between Euclidian distance and a correct approach can be described in Fig. 1. According to the Euclidian distance, the distance between two points P1 and P2 would be equal to the cord P1P2, whereas the actual distance would be along the circular arc.



Fig. 1. Euclidian distance vs. actual geographical distance.

In our prior work [23] we investigated two algorithms, namely, the Spherical Law of Cosines and Equi-rectangular approximation, in calculating a geographical distance between two trace locations. Our implementations and usage of the two schemes suggested that, for more accurate precision level, the spherical cosine is better than the Equi-rectangular approximation. But for faster system performance the latter is preferred. In our current mathematical analysis, we used the latter in case of averaging one month’s data for all the cabs, which contained over 10 million records. While working with traces over 24 hour time span we used Spherical Cosine Law to get an accuracy level of less than one meter.

3.4. Degree of connectivity

We define the Degree of Connectivity (DoC) as the total number of nodes reachable from a particular node via any wireless path not longer than a given number(k) of hops. The Average Degree of Connectivity (ADoC) is the metric that characterizes the reachability of any random node with the network. Mathematically, ADoC specifies the average number of reachable nodes from a single source within a given path length. Hence, ADoC of a vehicular network with n nodes is defined by,

$$ADoC = \frac{\sum_{i=0}^n DoC}{n} \quad (1)$$

3.5. Network partition

We define the Network Partition as a connected component where any node can communicate with another node in the component through multi-hop communication. In other words, we can say that, there exists at least one path from any particular node to each of the other nodes within a partition or connected component. Obviously, the path length can never exceed the total number of nodes in that component. The size of the partition is determined by number of nodes in that partition. If the entire topology is connected, we get only one partition within the network. On the other hand, if any node is totally isolated from other nodes, this will be a partition of size one. Less number of partitions will lead to better connectivity and information dissemination.

3.6. Data structure for storing matrices

All the computations are done through manipulation of n -by- n matrices where n is the total number of mobile nodes in a metropolitan area. For simplification, we assume that the propagation delay for a single hop wireless broadcast is same for all nodes, given a specific transmission range. Hence we are only concerned with the degree of connectivity for each node instead of the actual physical distance between the nodes. The physical distance is only used to generate the adjacency matrix of the network by comparing with the transmission range. As the maximum transmission range for DSRC is 1000 m, most of the nodes have only a few neighbors directly connected through single hop communication. Hence, the adjacency matrix is basically a sparse boolean matrix. In order to reduce the complexity of the manipulation of large sparse matrices we use special data structures for storing the matrix. As we have only binary values as the matrix elements, we use string of n bits to store a row of a matrix as each row has n elements. We divide the rows into $\lceil n/b \rceil$ blocks where b is the size of each block. Normally the block size can be specified according to the processor architecture, that is, for a 64-bit computer the block size can be 64. We also keep track of the *Non-zero Blocks* using linked lists for each row. By the term *Non-zero Blocks* we refer to those blocks in a row which has at least one of the b bits set to 1. This helps us reduce unnecessary operation on the zero blocks. Fig. 2 shows a sample matrix with the non-zero blocks shaded.

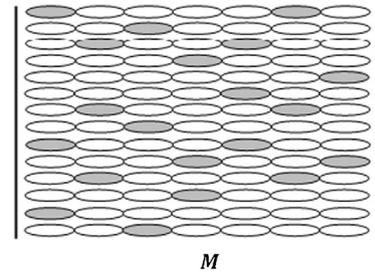


Fig. 2. Data structure of storing sparse matrix.

4. Preprocessing raw data

For any spatio-temporal analysis using GPS traces, one of the major steps involves preprocessing raw data and filtering out bad/invalid data. We developed our own strategy for interpolating and filtering the traces. Below we describe in brief how we preprocess the data for analysis.

4.1. Processing raw input

First, we process the input raw trace files into a data structure containing all node information. The GPS traces are organized into a set of ASCII text files, where each file corresponds to a single taxi cab. Each file contains different number of trace records with variable sampling frequency of broadcasting GPS data to the central repository. Each trace record comprises of several fields of data separated by a delimiter. The function `Get_Input(Trace file directory)` extracts each individual record from the directory of trace files and stores them into a data structure of nodes, where each node represents a taxi cab.

4.2. Determining node position vector

Next, we calculate the node positions for a given timestamp from the data structure using interpolation method. This step calculates the individual geographical positions (Latitude, Longitude) of each node for a specific time of interest. As the nodes generate traces randomly with an average sampling rate of around 30 seconds or less, we use a method of interpolating the closest samples to find the approximate position of the node at the specific time of experiment. We check for the samples one minute backward and forward and depending on the available samples we take the average of different interpolated and extrapolated values. Below we mention the possible different cases:

Case 1: In case, two samples are available during the total interval of 2 minutes, as shown in Fig. 3 and 4, we compute the interpolated (if two sample points are located in opposite side of the experimental timestamp) or extrapolated (if two sample points are located in same side of the experimental timestamp) position using the below formula:

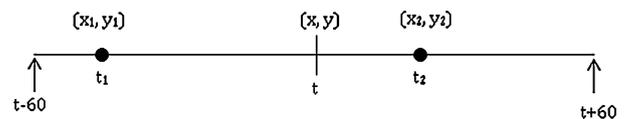


Fig. 3. Interpolation of two sample points.

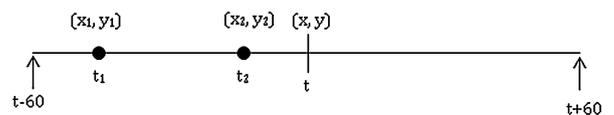


Fig. 4. Extrapolation of two sample points.

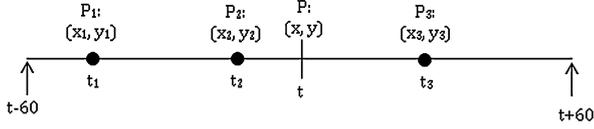


Fig. 5. Calculating average position from more than two sample points.

$$x = \frac{x_2 - x_1}{t_2 - t_1} t - t_1 \quad (2)$$

$$y = \frac{y_2 - y_1}{t_2 - t_1} t - t_1 \quad (3)$$

Case 2: If more than 2 samples are available within the interval (Fig. 5), we calculate the average of the different interpolation/extrapolation position acquired from several pairs of points. For example, in the above scenario where we have three consecutive points P1, P2 and P3, we calculate the position of P by:

- (1) interpolating P1 and P3;
- (2) interpolating P2 and P3, and
- (3) extrapolating P1 and P2.

Finally we take the average of the three values to minimize the error probability in the approximation.

Case 3: In case, less than two samples are available during the total interval of 2 minutes, we extend the sample searching interval in either or both direction to get at least one sample point in either direction from the experimental timestamp t .

5. Proposed algorithm

This section describes the proposed algorithm for multihop connectivity analysis. First we describe the steps required for computing the Degree of Connectivity. Then we demonstrate the steps using an example matrix.

5.1. Determining multi-hop degree of connectivity

The major steps involved in computing the k -hop Degree of Connectivity are as follows:

- (1) Generate adjacency matrix from the node position vector for a specific transmission range. The Adjacency Matrix is defined as $M = AdjMatrix(L, TX_Range)$;
- (2) Determine the k -hop reachability matrix M^k from adjacency matrix M . It is in fact the k -hop Transitive Closure calculated as $M^k = \prod_{j=1}^k M^j$;
- (3) Compute the k -hop degree of connectivity (k -hop reachability) DoC^k from M^k . DoC^k for node i , $(DoC_i)^k = \sum_{j=1}^{|V|} M^k(i, j) - 1$.

Below we elaborate each of these steps.

5.1.1. Building the adjacency matrix

In this step, we determine the adjacency matrix of the nodes from the node position vector and a specified transmission range. For our analysis of determining the k -hop reachability, we just consider the connectivity issue ignoring the actual physical distance as we are not determining the best routing path between two nodes; rather we are determining the existence of a path between two nodes. Hence we only manipulate a binary matrix in each of the step henceforth. Fig. 6 describes the algorithm for determining the binary adjacency matrix. From the algorithm, we can see the loop executes $\frac{n(n-1)}{2}$ times. Hence, the time complexity of determining the adjacency matrix is $O(|V|^2)$. While calculating the geographical distance between two node-positions, we consider the spherical cosine law which gives an accuracy within 1 meter. Also, as we are considering undirected graph, so $M[i][j] = M[j][i]$.

Procedure AdjMatrix{Node Position vector(L), TX_Range(R)}

```

n = |V|;
for i = 1 to n - 1
  for j = i + 1 to n
    if (GeoDistance(Li, Lj) ≤ R)
      M[i][j] = M[j][i] = 1;
    else
      M[i][j] = M[j][i] = 0;

```

Fig. 6. Algorithm for determining adjacent matrix.

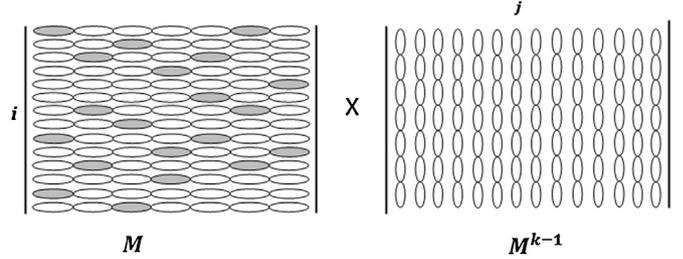


Fig. 7. Multiplying M with M^{k-1} to compute M^k .

Initialize Product $Mat_C = Adjacency\ Matrix\ M$;

```

BCMM(max-hop, block-size){
  Blocklist_A = Non_Zero_Blocks(C, block-size, Row-wise);
  Blocklist_B = Non_Zero_Blocks(C, block-size, Column-wise);
  for(k=1; k<=max-hop; k++)
  {
    if(k==1)
      Modified_Blocklist_A = Boolean_Multiply(Blocklist_A, Blocklist_B);
    else{
      Modified_Blocklist_A = Boolean_Multiply(Modified_Blocklist_A, Blocklist_B);
      if (Modified_Blocklist_A == NULL)
        break;
    }
  }
  Boolean_Multiply(Blocklist_A, Blocklist_B){
    Initialize Updated_Blocklist = NULL;
    for each row i of Blocklist_A
      for each column j of Blocklist_B
        for k=1 to size(Blocklist_A[i])
          if(Blocklist_A[i][k] & Blocklist_B[k][j] != 0){
            Insert block index containing (i,j) into Updated_Blocklist;
            Mat_C[i][j]=1;
            break;
          }
    return Updated_Blocklist;
  }
}

```

Fig. 8. Algorithm for bitwise matrix multiplication.

5.1.2. Determining the k -hop reachability matrix

Fig. 7 depicts the process of calculating the k -hop reachability matrix, $M^k = \prod_{j=1}^k M^j$. For each of the matrix multiplications, normally it would require $O(V^3)$ operations. So for $k - 1$ multiplications required to determine M^k , in the worst case it would require $O(kV^3)$. But our BCMM algorithm reduces the number of operation in several steps that can minimize the average complexity from typical matrix multiplication. The pseudocode of BCMM algorithm is shown in Fig. 8.

5.1.3. Computing DoC for k -hops

Using the following simple calculation, the k -hop degree of connectivity can be determined for each node. DoC^k for node i ,

$$DoC_i^k = \sum_{j=1}^{|V|} M^k(i, j) - 1$$

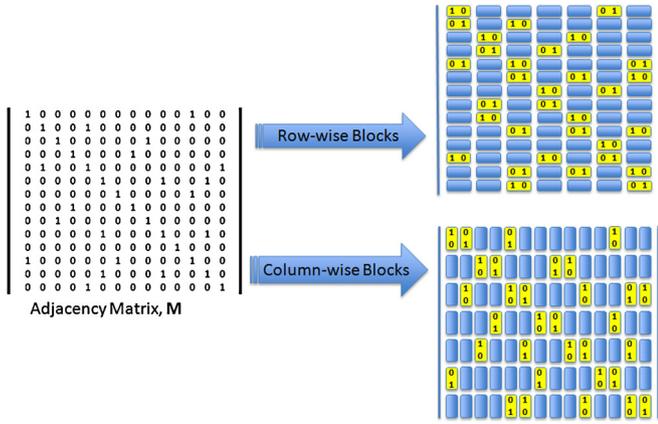


Fig. 9. Adjacency matrix stored in data structures. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

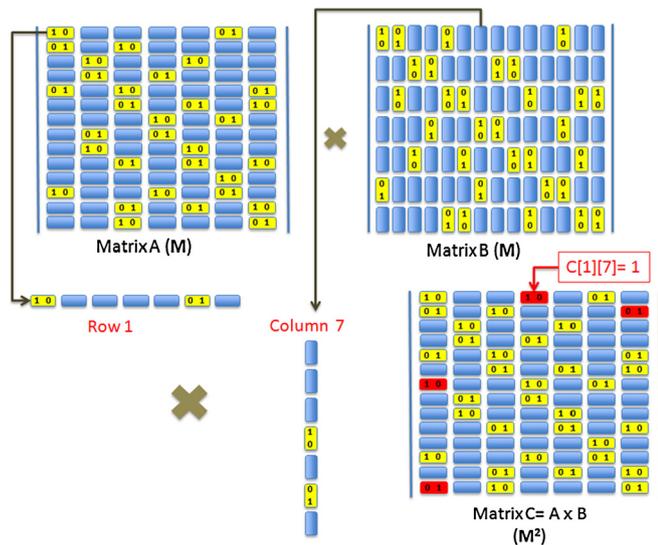


Fig. 10. Computation of $C = A \times B$ (first iteration). (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

Hence, the average DoC for the network with maximum

$$k\text{-hops} = \frac{\sum_{j=1}^{|V|} DoC_i^k}{|V|}$$

5.2. Demonstration of BCMM with an example matrix

In this subsection we demonstrate the details operations done in each step of the BCMM algorithm using illustrative figures.

We consider a topology of 14 randomly distributed nodes corresponding to a vehicular network at a specific timestamp. At first, the adjacency matrix, M is generated as per the algorithm described in Fig. 6. After that, the adjacency matrix is transformed into row-wise and column-wise blocks. Fig. 9 shows how the non-zero elements are stored in blocks of data structure. Here we considered a block-size of 2 for ease of demonstration. Practically the number of nodes will be several thousands and the block-size will be considered either 32 or 64, depending on the processor architecture. In the figure, the non-zero blocks are marked as yellow. These blocks actually participate in the multiplication, whereas the blue blocks do not contribute any value during the multiplication as they are all zeros.

Initially, both A and B matrices are assigned with the adjacency matrix M . During the first iteration of BCMM algorithm, M is multiplied to itself yielding M^2 as the product matrix C . This step

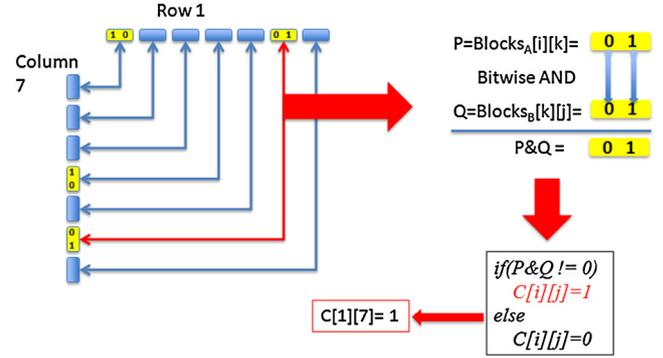


Fig. 11. Detail step of multiplying row with column.

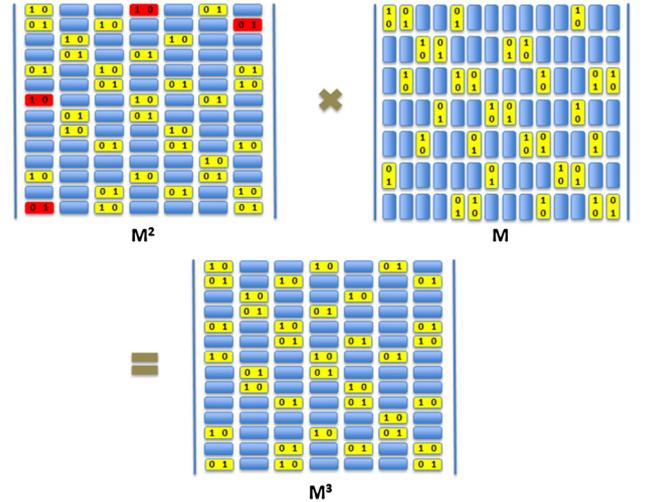


Fig. 12. Computing $M^3 = M^2 \times M$ (second iteration).

is shown in Fig. 10, where the red blocks are the blocks that have changed from A during the process. To elaborate, Fig. 11 shows the details of bitwise multiplication that takes place during the individual row-column multiplication. After the first iteration of BCMM algorithm, only the modified blocks are considered for the subsequent multiplication steps. After every iteration, C is assigned to A for next iteration. This incremental approach of chain multiplication is described in Fig. 12. During the second iteration, only 4 modified blocks from previous step are used for multiplication. As there is no new non-zero elements generated in M^3 , the transitive closure of M will be given by M^2 in this case. Once the transitive closure is determined, the algorithm terminates discarding any outstanding iteration.

6. Space-time complexity of the algorithm

In this section, we first point out how the complexity of sparse matrix multiplication was reduced with the aid of our unique data structure. Then we derive the theoretical complexity of our algorithm.

6.1. Techniques for reducing space-time complexity

- As the matrix is a Boolean matrix, we can use logical operation AND instead of multiplying two integers. As processors can execute logical operation faster than multiplication this reduces the hardware time consumption.
- Instead of considering each element as a single integer value, we use 32-bit integers to represent 32 consecutive Boolean elements of the matrix in a row. This makes the space complexity of the matrix reduce by a factor of 32. For 64-bit integers,

the space requirement reduces by a factor of 64. Moreover, a single bitwise AND operation of two 32-bit integers can now be equivalent to previous 32 logical AND operations. Hence, blocks with size b reduce both space and time complexity by a factor of b .

- We use a data structure to keep track of those bit-blocks in matrix M which have at least 1 bit set within the 32 bits. With the help of this data structure, we only manipulate with non-zero bit blocks. This means the number of blocks for manipulation would be at most $|E|$, where $|E|$ is the total number of direct links (or edges) in the topology. In practice, this number would be much less than $|E|$, because many of those blocks would have more than one bit set, as it is very unlikely that all the bits that are set would be distributed over distinct blocks. In the worst case, this number of non-zero bit-blocks in M would be equal to $|E|$; whereas in best case this would be $|V|$ as each node would have at least one bit set due to self connectivity. In Fig. 7, we represent the shaded blocks as valid blocks for manipulation.
- Likewise, instead of manipulating each of the valid blocks of M with $|V|$ columns of M^{k-1} , we keep track of the altered blocks of M^{k-1} from previous steps using another data structure, which practically discards many of the blocks from consideration in the current step, in a sparsely connected network.
- Further, if there are several valid blocks in a row i of M , we manipulate with column j of M^{k-1} until we get the first non-zero block after AND operation. This means we can determine the element (i, j) of M^k whenever we get a non-zero result from AND operation. This also implies that, in best case there will be only $O(V^2)$ operation needed for one multiplication step.

6.2. Generalized complexity analysis

In general, the total number of operations required to compute one element $M^k(i, j)$ in the matrix product M^k is as follows:

Number of operations to compute $M^k(i, j) = \sum$ Number of non-zero blocks in row i of $M \wedge \sum$ Number of altered blocks in column j of $M^{k-1} = N_i^r + N_j^c$.

Considering all rows of M ,

$$N^r = \sum_{i=1}^{|V|} N_i^r$$

where, N^r is bounded by $|V| \leq N^r \leq |E|$.

Again, considering all columns of M^{k-1} ,

$$N^c = \sum_{j=1}^{|V|} N_j^c$$

where, N^c is bounded by $0 \leq N^c \leq (DoC^{k-1} - DoC^{k-2})$.

Hence, the total number of operations required to complete the product matrix in each step is =

$$N \times (N^r + N^c)$$

where, $N = \frac{|V|}{b}$.

Therefore, the total number of operations required to compute $M^k = (k-1) \times N \times (N^r + N^c)$.

In our analysis, we are basically interested to observe the connectivity up to a certain number of hops (less than 30). Therefore, considering k as a small constant value we can see that the best case complexity to find M^k is $O(V^2)$ and the worst case would be $O(V(V+E))$.

7. Applications of the proposed algorithm

In this section, we describe two major applications of this algorithm to find out the network partitioning and the reachability of broadcast messages within a given amount of time. Both these applications use the bitwise multiplication of the sparse matrix data structure.

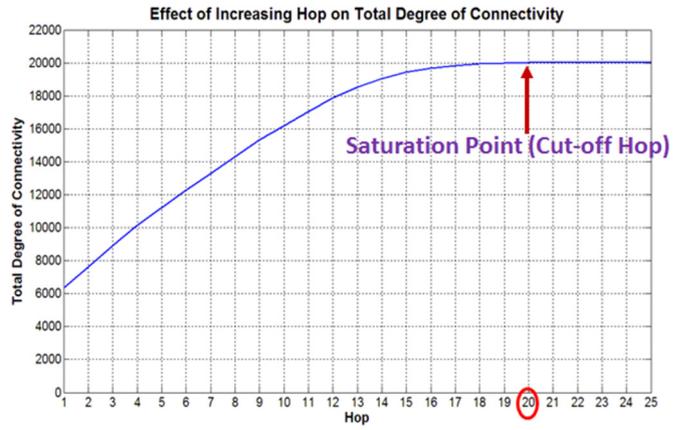


Fig. 13. Saturation point of multi-hop communication.

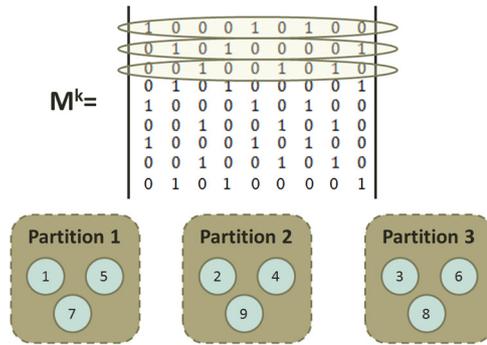


Fig. 14. Determining the network partition.

7.1. Determining network partitions

The steps of determining the network partition is very much similar to determining the k -hop transitive closure. The only thing is to find the minimum k for which $M^{k+1} = M^k$. At that point, M^k will give the full transitive closure of the topology. This minimum value of k , which we define as *Cutoff Hop*, can be a very important property of the wireless network because it determines the exact point when the degree of connectivity for the network gets saturated. After achieving this saturation point, no more nodes can be reached by any node even if the hop number is arbitrarily increased. This cutoff hop also determines the steepness of the curve that reflects the rate of DoC change with respect to hop increase. Fig. 13 shows the cutoff hop for a vehicular network which is 20. After 20 hops, the connectivity gets saturated which means we get a transitive closure of the topology.

In order to get the partition information from the full transitive closure, M^k , we extract the rows from the matrix where the total number of distinct row pattern gives the total number of partition and the arithmetic sum of the corresponding row will give the size of the partition (Fig. 14). This algorithm of determining the network partition is a novel approach.

7.2. Determining reachability of broadcast messages

Considering the dynamic nature of the topology of a vehicular network, it is very interesting to measure the propagation rate of an emergency broadcast message within a fleet of vehicles in a metropolitan area. In order to measure this we need to consider the variable positions of the fleet in terms of time. Let us assume that the vehicles of a specific fleet update their locations every p seconds and the broadcast message propagates h hops in each second. If there are n total vehicles and t time intervals to be con-

sidered, then the percentage of fleet that receives the message is given by the following equation:

$$\frac{\sum_{i=1}^n \sum_{j=1}^n M^t(i, j) \times 100}{n} \quad (4)$$

where M^t is defined by

$$M^t = \prod_{k=1}^t (M_k)^{p \times h} \quad (5)$$

Here M_k is the adjacency matrix for the topology at k -th time interval.

8. Results of spatio-temporal analysis of traces

In this section, we describe some of the key findings of our GPS based trace analysis to determine the probability of seamless connectivity within the taxi fleet. First we consider a node positioning scenario that corresponds to a particular time. Then we attempt to explore the time varying characteristics of the connectivity. Fig. 15 describes the geographical node positioning of taxi cabs (as observed from satellite) at a random experimental time for which we analyzed the V2V connectivity. The experimental time chosen for this snapshot was at 2:30 pm on June 5, 2008 which was a working day.

8.1. Average degree of connectivity for a specific time

Fig. 16 depicts the impact of transmission range and hop on average degree of connectivity for the mobile taxi network at a specific time (2:30 pm) on a working day. It is obvious that, increasing the wireless transmission range will have a significant impact on the Degree of Connectivity. The ADoC graph shown here corresponds to the snapshot of the whole taxi fleet at the said experimental time (Fig. 15).

From Fig. 16, it can be clearly observed that the average degree of connectivity is minimum for single hop connection, while longer transmission range corresponds to higher degree of connectivity. As we gradually increase the path length (hop count), more and more source-destination pairs become reachable via multi-hop communication which ultimately increases the ADoC of the network. All the curves show a near-linear rate of connectivity increase with the increment of path length up to a certain point when the curve becomes horizontal. This corresponds to the state when no more nodes can be explored with further hop increase. We refer this point as the saturation point. However, the slope of the curve depends on the transmission range, which implies that the longer the range the less number of hops are required to achieve maximum possible connectivity. The ADoC of a network after saturation indicates the portion of the fleet that can be reached from an arbitrary source node using multi-hop communication. The graph can also describe the percentage of the wireless coverage after a specific number of hops for any transmission range which may provide an estimate for the QoS provisioning of delay-sensitive applications.

8.2. Change of connectivity with time

In this subsection, we will show the variation of V2V connectivity with respect to time. For that, we have considered two different span of intervals, one is relatively short span which is the variation within 30 minutes and another is relatively long spanning over the whole day.

8.2.1. Short duration (30 minute)

In this case, we took a total of 30 sample snapshots within a half an hour duration, where the time difference between each successive snapshot is one minute. The selected time is from 2:30 pm to 3:00 pm on a business day. First we plot the change of connectivity from the perspective of a single node. Fig. 17 shows the change of connectivity for a random node with a transmission range of 300 m restricted by a maximum path length of 25 hops. This gives an idea about the rapid fluctuation of V2V connectivity for an individual node.

If we measure the standard deviation of connectivity change within this 30 minute interval for all the 536 nodes, we get an irregular scenario as described by Fig. 18. Here many of the nodes have high variance of connectivity while some others have less variance.

On the other hand, Fig. 19 shows the change of average connectivity with respect to time for different transmission ranges. This figure is a 3D projection of Fig. 16 where the topmost layer corresponds to transmission range of 1000 m and the bottommost layer represents the shortest transmission range of 300 m. From this figure, it can be observed that within a short span of time, the change of average connectivity is not significant even though a vehicle can move away more than a mile in the freeways within a minute. Even if the connectivity of individual node is varying a lot but when we take the average over all nodes it remains almost constant. The reason behind this phenomena is because, some cabs may lose connectivity while traveling out of the downtown or airport area while other cabs get connected when they get near a dense area. The bottom line is the average change of connectivity over the whole fleet almost remains stable within a short duration interval.

Fig. 20 describes a little bit more details of the above figure, where we can closely observe the change of average connectivity for each different transmission range and also measure the variation of connectivity with respect to time and hops. Each of the ribbons (stripes) correspond to a particular hop number which restricts the total path length within that number of hop.

8.2.2. Long duration (whole day)

In contrast with the short interval, a long duration average connectivity analysis results into reasonable fluctuation over the course of a day. This is quite natural because the fleet is not entirely utilized evenly throughout the day and also because of the impact of rush hours. From Figs. 21 and 22, we can easily observe that the maximum average connectivity is achieved during the afternoon and evening rush hours (4 pm and 6 pm). At this time most of the cabs can be found within the vicinity of downtown area for after office trips. For the transmission range, we considered a range of 300 m due to the reason that, in downtown area it might not be possible to reach too far because of the obstacles of high rise buildings and skyscrapers.

8.3. Network partitioning results at a specific timestamp

We attempt to identify the network partitions of the whole fleet of cabs based on the instantaneous positions at a certain time. Using the same topology snapshot as the previously analysis, the mobile taxi nodes distributed all across the city of San Francisco can be partitioned into various partitions based on their wireless connectivity between other nodes. For a specific transmission range of 300 m, it was found that, out of total 536 nodes, more than 20 percent of the nodes were isolated or disconnected from any other node. 16 partitions were found having 2 nodes and 9 with 3 nodes. The largest partitions found included 155 nodes,

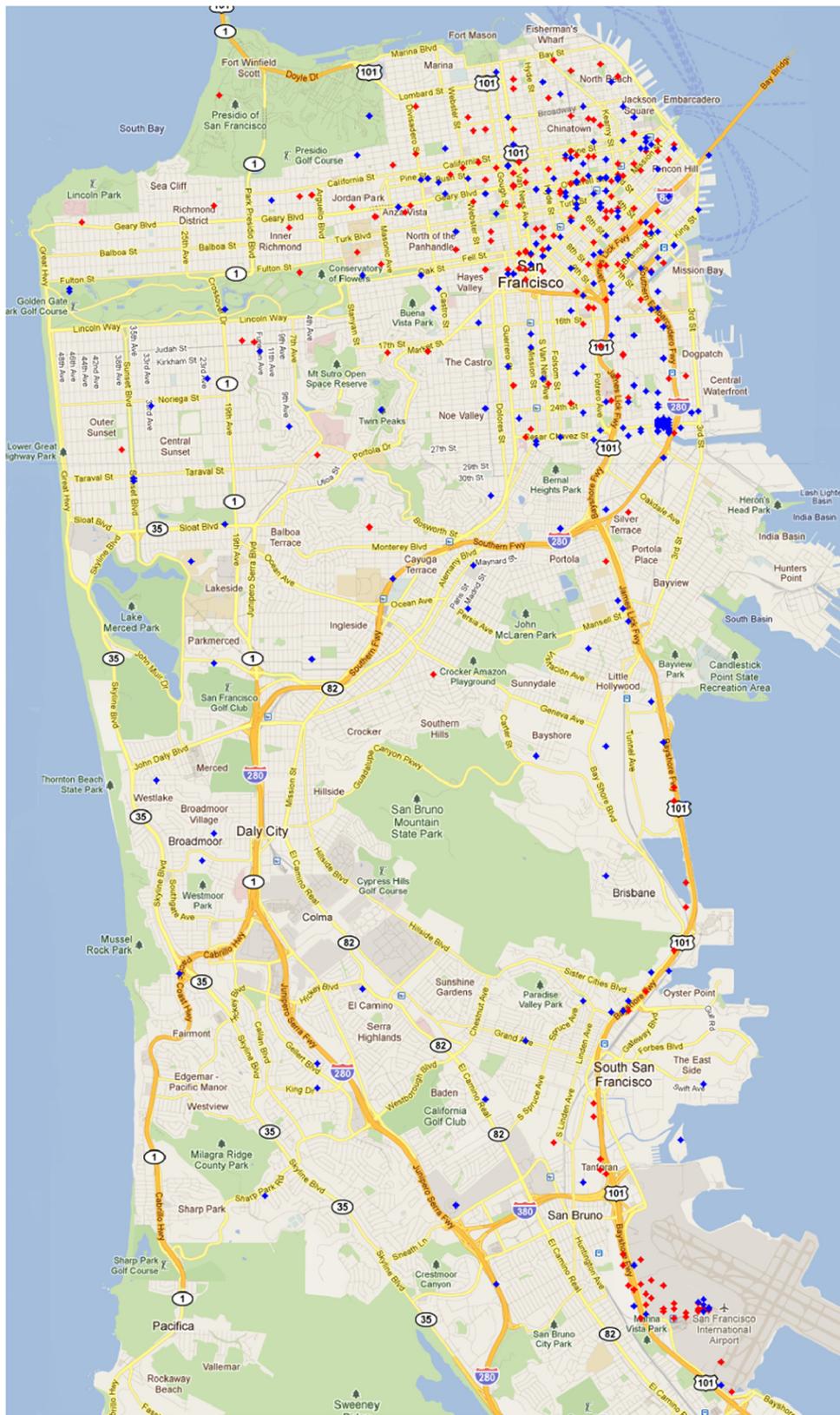


Fig. 15. Taxi node positions at a particular experimental time (blue and red dots refer to empty and occupied taxis respectively). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

which is located in the downtown area. The second largest partition with 120 nodes was found in the airport vicinity. Table 1 shows the distribution of nodes in various sizes of partitions for 300 m transmission range.

8.4. Change of partitioning over time

In order to capture the change of partitions over time we took two samples—one during mid-day (Fig. 23 – left) and another at

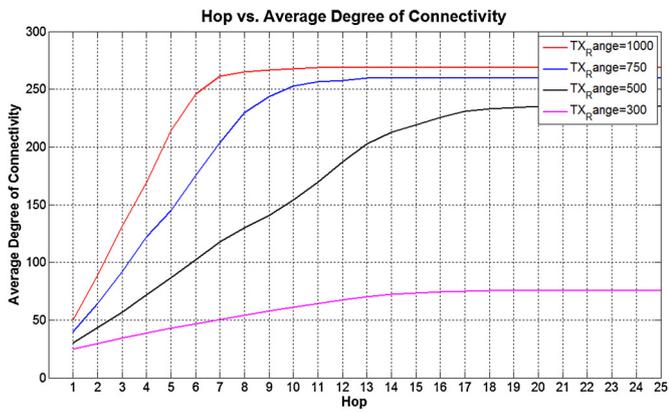


Fig. 16. Impact of path length and transmission range on ADoC.

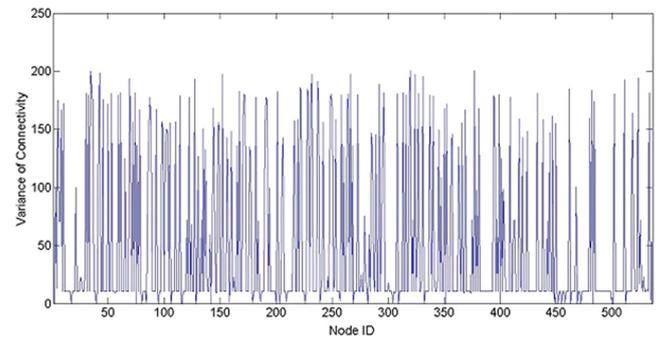


Fig. 18. Variance of connectivity for all the 536 nodes within half an hour.

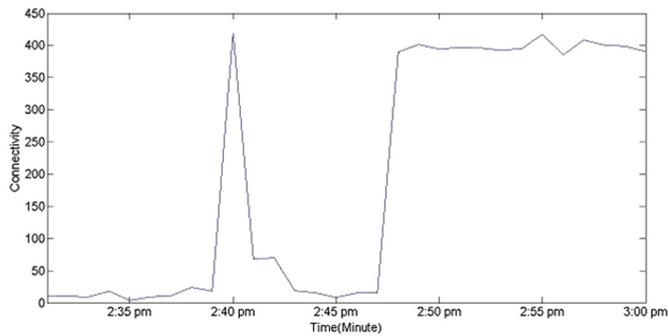


Fig. 17. Change of connectivity with respect to time for an individual node.

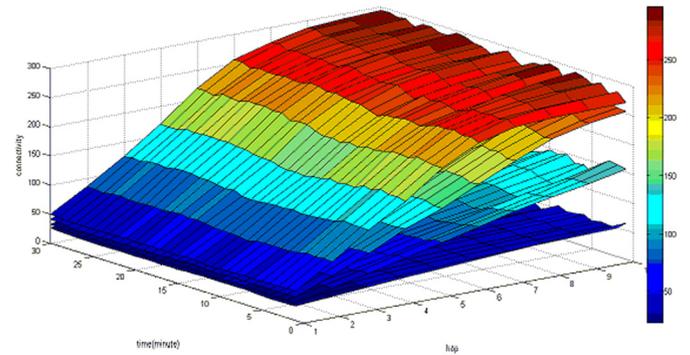
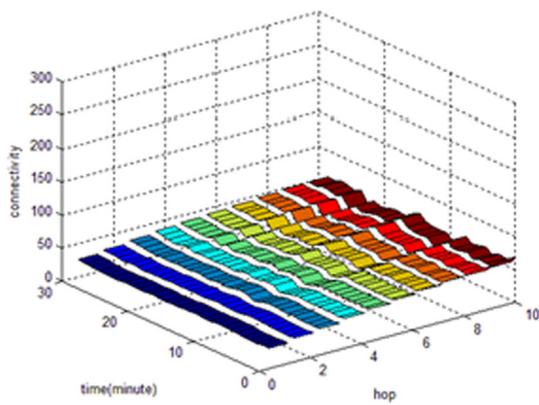
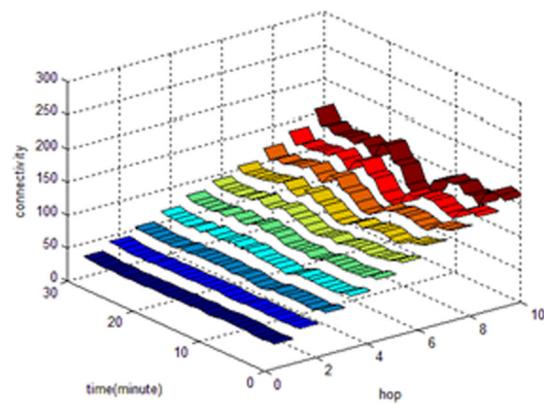


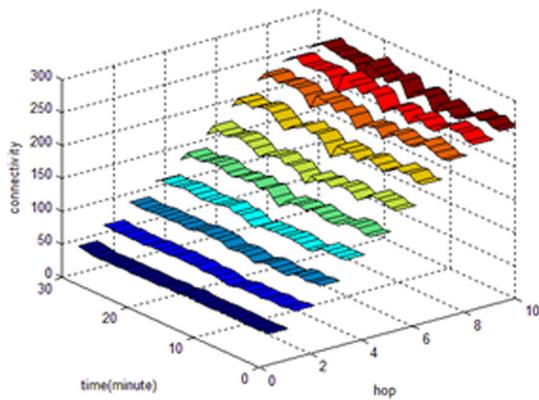
Fig. 19. Change of average connectivity with respect to time for different TX ranges.



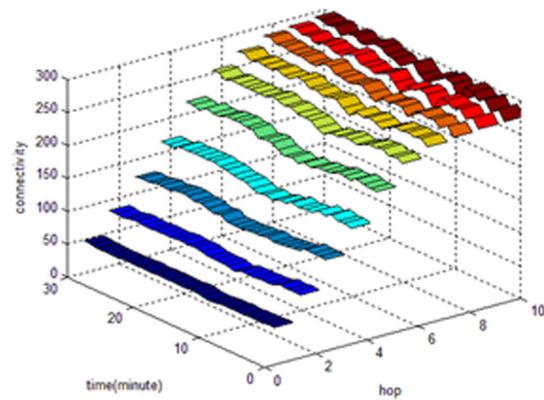
TX range 300m



TX range 500m



TX range 750m



TX range 1000m

Fig. 20. Change of average connectivity with respect to hop for different TX ranges.

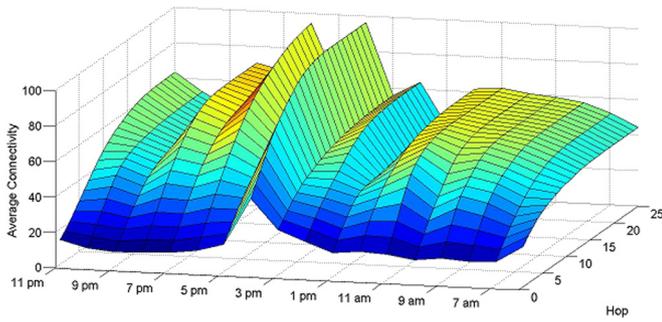


Fig. 21. Change of average connectivity with respect to time for 300 m TX range.

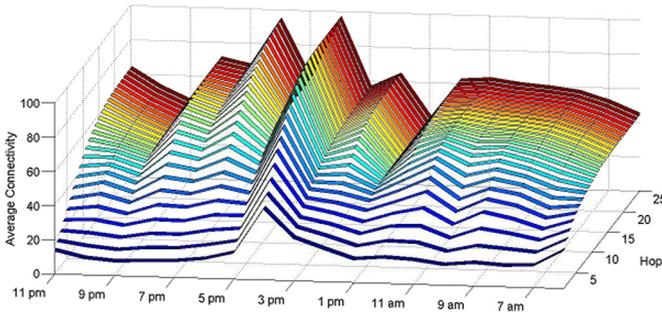


Fig. 22. Change of average connectivity with respect to hop for 300 m TX range.

Table 1
Partitioning of nodes for 300 m TX range.

Partition size (# of nodes)	Number of partitions	Total nodes
1	110	110
2	16	32
3	9	27
4	1	4
5	3	15
6	1	6
11	1	11
22	1	22
34	1	34
120	1	120
155	1	155
Total	145	536

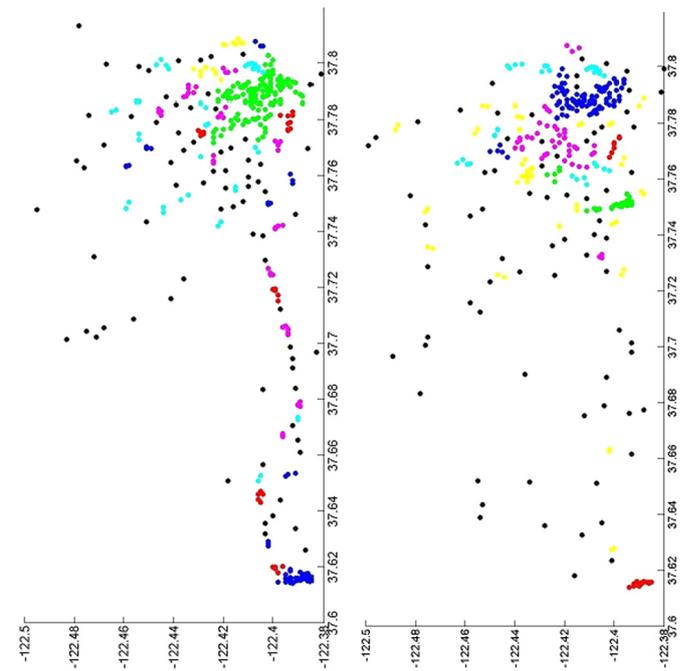


Fig. 23. Change of network partitions with respect to time. The left figure corresponds to noon (12 PM) and the right corresponds to midnight (12 AM).

Table 2
Number of partitions for different transmission ranges.

Partition size	Number of partitions			
	300 m	500 m	750 m	1000 m
1	110	69	46	31
2	16	11	10	10
3	9	7	3	1
4	1	1	2	1
5	3	0	0	1
6 to 10	1	1	3	3
11 to 20	1	0	0	1
21 to 30	1	1	1	1
31 to 50	1	1	1	1
51 to 100	0	0	0	0
101 to 150	1	0	0	0
151 to 350	1	0	0	0
350+	0	1	1	1
Total	145	92	67	51

mid-night (Fig. 23 – right). The nodes with the same color belongs to the same connected component or partition. In both the two parts of the figure, black dots represent isolated nodes that are not connected with any other node. If the plots are superimposed on the map of San Francisco, we can see that during the night the taxis are more scattered in the suburban area than during the daytime when taxis concentrate near the downtown or airport area. The dense upper right portion corresponds to the downtown area and the bottom cluster corresponds to airport.

8.5. Change of partitioning with transmission range

As the degree of connectivity varies along with transmission range, the partitioning also changes. Table 2 shows the distribution of nodes in different sizes of partitions for different transmission ranges. It is quite natural that, the number of isolated nodes (partitions with size 1) decreases as the transmission range increases. Also the total number of connected components is reduced at the same time. Fig. 24 shows the average size of partitions for different transmission ranges. The average partition size is less than 4 in case of 300 m range whereas in case of 1000 m it goes above 10.

Fig. 25 shows the difference between two partitioning results for the same time with different transmission range. On the left,

Average Partition Size vs. TX Range

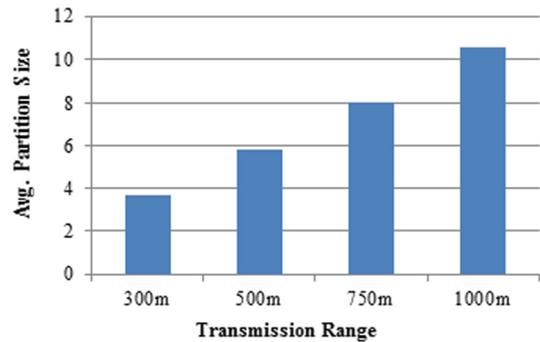


Fig. 24. Average size of partitions for different transmission ranges.

due to a transmission range of 1000 m, we have very large partitions which almost connects the whole city. On the right, due to shorter transmission range of 300 m, we can see lot of smaller partitions for the same node positions. Both the plots correspond to time of 11 AM for a business day.

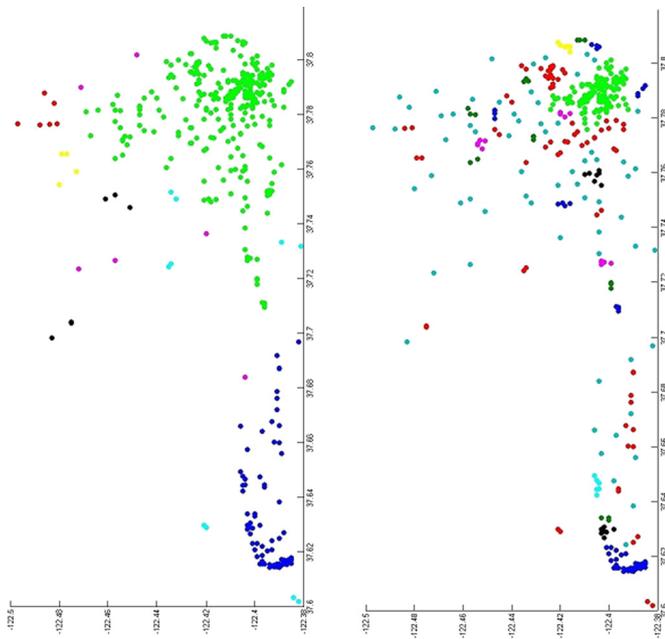


Fig. 25. Change of Network Partitions with respect to transmission range. The left figure corresponds to a TX range of 1000 m and the right corresponds to 300 m.

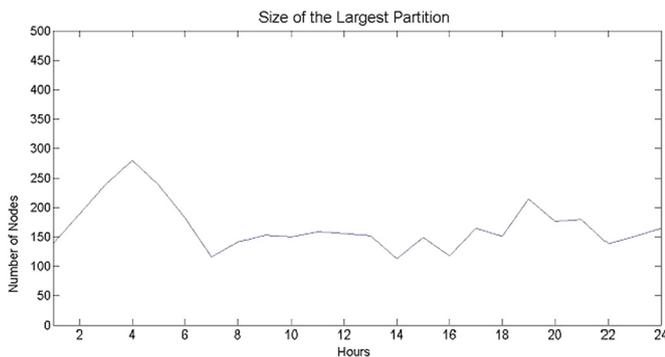


Fig. 26. Size of the largest partition for 300 m transmission range.

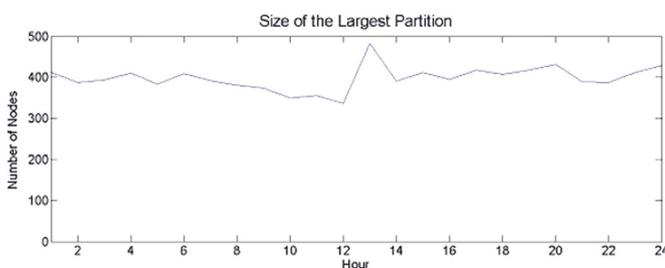


Fig. 27. Size of the largest partition for 1000 m transmission range.

8.6. Dimension of largest connected component

Figs. 26 and 27 show the comparison of partition size for the largest connected component in two different transmission range. For a range of 300 m, the largest connected component within the taxi network consist of 150 to 200 nodes on the average. Whereas in case of 1000 m transmission range, we can see bigger partition with around 400 nodes in the largest partition.

8.7. Broadcast propagation rate within a taxi fleet

We analyzed the broadcast message propagation rate within the entire fleet of 536 taxis of San Francisco Yellow Cabs. We

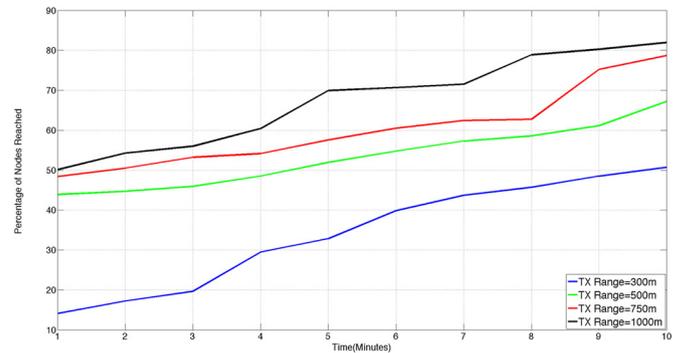


Fig. 28. Broadcast Propagation Rate Within a Taxi Fleet.

assume that the flooding propagates in a rate of 1 hop per second and the frequency of GPS location update is once in every minute. This means that the topology the considered to be changing every minute and within this interval the message propagates through neighbors. Since this rate of propagation depends on the source node, we take an average considering each node as source node. Fig. 28 shows the propagation rate for different transmission ranges. On an average, more than 50 percent nodes of the entire fleet receives the message in 10 minutes with 300 m transmission range. As the transmission range is increased, reachability also increases.

9. Conclusion

The paper presents a novel algorithm for spatio-temporal analysis of multi-hop V2V connectivity and network partitioning. With the efficient storage and computation procedure, we reduce the space-time complexity of our algorithm to a great extent. The paper also describes the results of our connectivity analysis from the real world GPS traces. Our results show that, on an average more than 70% vehicles can be communicated using multihop vehicular communication with reasonable transmission range in an urban environment. The analytical data presented in this paper revealed many new and useful features that can be helpful for wireless researchers, government organizations, taxi companies and even for the drivers or passengers. Even though our results might seem to be vulnerable to the impacts of line-of-sight (LOS), only about 1% of the geographical area considered within our analyses is vulnerable to building obstacles. This includes the downtown area where large buildings can significantly obstruct the LOS within 300 meter. Other areas do not contribute much in obstructing the vehicle communication within such a small range. Recent results from experimental measurement of LOS impact [37] on vehicular communications show that for a distance greater than 400 m the difference of the received signal strength between LOS and Non-LOS transmission is practically zero. This proves that our results of multi-hop connectivity analyses for transmission ranges greater than 300 m are literally free from the detrimental effects of building obstacles. Our future work will explore the clustering feature of mobility for V2V communications and for DSRC infrastructure configurations.

References

- [1] <http://www.yellowcabsf.com/>.
- [2] <http://www.exploratorium.edu/>.
- [3] <http://cabspotting.org/>.
- [4] Jung-hoon Lee, Analysis on the waiting time of empty taxis for the taxi telematics system, in: ICCIT, 2008.
- [5] In-Hye Shin, Gyung-Leen Park, Association analysis of location tracking data for various telematics services, in: ICCSA, 2010.

- [6] M. Piorowski, N. Sarafjanovic-Djukic, M. Grossglauser, A parsimonious model of mobile partitioned networks with clustering, in: IEEE COMSNETS, Bangalore, India, 2009.
- [7] <http://cabspotting.org/api>.
- [8] <http://cabspotting.org/client.html>.
- [9] <http://cabspotting.org/timelapse.html>.
- [10] <http://www.crawdada.org/>.
- [11] Bin Li, Daqing Zhang, Lin Sun, Chao Chen, Shijian Li, Guande Qi, Qiang Yang, Hunting or waiting? Discovering passenger-finding strategies from a large-scale real-world taxi dataset, in: IEEE Pervasive Computing, 2011.
- [12] H. Huang, Y. Zhu, X. Li, M. Li, M.-Y. Wu, META: a mobility model of Metropolitan TAxis extracted from GPS traces, in: IEEE WCNC, 2010.
- [13] Y. Yue, Y. Zhuang, Q. Li, Q. Mao, Mining time-dependent attractive areas and movement patterns from taxi trajectory data, in: GeoInformatics, 2009.
- [14] Junghoon Lee, Gyung-Leen Park, Design and implementation of a movement history analysis framework for the taxi telematics system, in: 14th Asia-Pacific Conference on Communications, APCC 2008, 14–16 Oct. 2008, pp. 1–4.
- [15] Shih-Fen Cheng, Xin Qu, A service choice model for optimizing taxi service delivery, in: 12th International IEEE Conference on Intelligent Transportation Systems, 4–7 Oct. 2009, ITSC'09, 2009, pp. 1–6.
- [16] Satish Ukkusuri, Lili Du, Geometric connectivity of vehicular ad hoc networks: analytical characterization, *Transp. Res. Part C, Emerg. Technol.* 16 (5) (October 2008) 615–634.
- [17] Hugo Conceio, Michel Ferreira, Joo Barros, On the urban connectivity of vehicular sensor networks, in: *Lecture Notes in Computer Science*, vol. 5067, 2008, pp. 112–125.
- [18] Maen M. Artimy, William J. Phillips, William Robertson, Connectivity with static transmission range in vehicular ad hoc networks, in: Proceedings of the 3rd Annual Communication Networks and Services Research Conference, 2005.
- [19] Saleh Yousefi, Eitan Altman, Rachid El-Azouzi, Mahmood Fathy, Analytical model for connectivity in vehicular ad hoc networks, *IEEE Trans. Veh. Technol.* 57 (6) (November 2008) 3341–3356.
- [20] Ivan Ho, Kin Leung, John Polak, Rahul Mangharam, Node connectivity in vehicular ad hoc networks with structured mobility, in: IEEE LCN, 2007.
- [21] Sunho Lim, Chansu Yu, Chita Das, A realistic mobility model for wireless networks of scale-free node connectivity, *Int. J. Mob. Commun.* 8 (3) (2010) 351–396.
- [22] Marco Fiore, Jerome Harri, The networking shape of vehicular mobility, in: Proceedings of ACM MobiHoc, 2008.
- [23] Mohammad A. Hoque, Xiaoyan Hong, Brandon Dixon, Analysis of mobility patterns for urban taxi cabs, in: IEEE International Conference on Computing, Networking and Communications, IEEE ICNC, 2012.
- [24] Mohammad A. Hoque, Xiaoyan Hong, Brandon Dixon, Innovative taxi hailing system using DSRC infrastructure, in: 22nd Annual ITS America Conference and Exposition, 2012.
- [25] Mohammad A. Hoque, Xiaoyan Hong, Brandon Dixon, Taxi hailing system using connected vehicle technology, in: 21st World Congress on Intelligent Transport Systems, 2012.
- [26] Mohammad A. Hoque, Protocol design in wireless networks: featuring channel access and vehicular communications, PhD dissertation, The University of Alabama, 2012.
- [27] Francois Le Gall, Improved output-sensitive quantum algorithms for boolean matrix multiplication, in: Proceedings of the 32nd annual ACM-SIAM symposium on discrete algorithms, SODA'12, 2012, pp. 1464–1479.
- [28] M.J. Fischer, A.R. Meyer, Boolean matrix multiplication and transitive closure, in: Proceedings of the 12th Annual Symposium on Switching and Automata Theory, 1971, pp. 129–131.
- [29] M.E. Furman, Application of a method of fast multiplication of matrices in the problem of finding the transitive closure of a graph, *Sov. Math. Dokl.* 11 (5) (1970) 1252 (English translation).
- [30] J.I. Munro, Efficient determination of the transitive closure of a directed graph, *Inf. Process. Lett.* 1 (2) (1971) 56–58.
- [31] M.D. Atkinson, N. Santoro, A practical algorithm for boolean matrix multiplication, *Inf. Process. Lett.* 29 (1) (1988) 37–38.
- [32] V.L. Arlazarov, E.A. Dinic, M.A. Kronrod, I.A. Faradzev, On economical construction of the transitive closure of a directed graph, *Sov. Math. Dokl.* (English translation) 11 (1970) 5.
- [33] V. Vassilevska Williams, R. Williams, Sub-cubic equivalences between path, matrix and triangle problems, in: Proceedings of the 51st Annual Symposium on Foundations of Computer Science, 2010, pp. 645–654.
- [34] A. Lingas, A fast output-sensitive algorithm for boolean matrix multiplication, *Algorithmica* 61 (1) (2011) 36–50.
- [35] F. Magniez, M. Santha, M. Szegedy, Quantum algorithms for the triangle problem, *SIAM J. Comput.* 37 (2) (2007) 413–424.
- [36] N. Bansal, R. Williams, Regularity lemmas and combinatorial algorithms, in: Proceedings of the 50th Annual Symposium on Foundations of Computer Science, 2009, pp. 745–754.
- [37] Rui Meireles, Mate Boban, Peter Steenkiste, Ozan Tonguz, Joao Barros, Experimental study on the impact of vehicular obstructions in VANETs, in: Vehicular Networking Conference (VNC), 2010 IEEE, IEEE, 2010, pp. 338–345.