Mobility Aware Data Delivery in Vehicular ad hoc Networks

Syed Muzakkir Ahmed, Mohammed Eunus Ali, Tanzima Hashem
Date Submitted: 3 October 2013
Date Published: 17 October 2013

Updated information and services can be found at:

These include:

Subject Classification  Vehicular Technology

Keywords  Multi hop data delivery; VANET; Real-time traffic;

Submitting Author's Comments
I always appreciate positive criticism to improve my work.

Comments  You can respond to this article at:
https://engine.lib.uwaterloo.ca/ojs-2.2/index.php/pptvt/comment/add/670/0

Copyright  Copyright © Date Submitted: 3 October 2013 Syed Muzakkir Ahmed et al. This is an open access article distributed under the Creative Commons Public Domain Declaration, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.
Mobility Aware Data Delivery in Vehicular ad hoc Networks

Syed Muzakkir Ahmed, Mohammed Eunus Ali, and Tanzima Hashem

Abstract

Vehicular ad hoc networks (VANETs) are promising for multi-hop data delivery between a source and a destination vehicle (or a node) because of their specific characteristics such as high level of mobility in constrained and predictable city/highway road networks. Multi-hop data delivery is useful for many real-life applications, such as a driver or a passenger in the moving vehicle may be interested to query for a sale in the shopping mall through fixed location-based service provider, or to know about available parking spaces or current traffic conditions of a region. In each cases, there is a need to handle real-time traffic information to efficiently transfer data between a source and a destination node. Existing methodologies of multi-hop data delivery mainly focus on predicted/approximate traffic and cannot adopt to dynamically changes of traffic condition. In this paper, we propose efficient data delivery technique that works based on real-time traffic information. We develop a novel methodology, Mobility Aware Data Delivery (MADD) that considers global (predicted/approximate) and local (real-time) traffic conditions. We develop three approaches: global window based approach (MADD-G), local window based approach (MADD-L) and hybrid approach (MADD-H). We carried out extensive experiment to demonstrate the effectiveness and efficiency of MADD-H and MADD-L with baseline approach MADD-G and other techniques GPSR, GPCR and RBVT-R. Simulation results show that the MADD-H outperforms other approaches in terms of number of hops, average delivery time and average delay.

Keywords

Multi hop data delivery, VANET, Real-time traffic.

Syed Muzakkir Ahmed is with the Department of Computer Science and Engineering, Military Institute of Science and Technology, Mirpur Cantonment, Mirpur-1216 Dhaka, Bangladesh e-mail: (syed.muzakkir@gmail.com).

Dr. Mohammed Eunus Ali and Dr. Tanzima Hashem are with the Department of Computer Science and Engineering, Bangladesh University of Engineering and Technology (BUET), Dhaka-1000, Dhaka, Bangladesh e-mail: (eunus@cse.buet.ac.bd and tanzimahashem@cse.buet.ac.bd)

Manuscript received July 12, 2013; revised September XX, 2013.
I. INTRODUCTION

Vehicular ad hoc networks (VANETs) [1] have received considerable attention both from academics (e.g., Network on Wheels [2], Sevecom [3], VIC’s [4]) and industries (e.g., Car-to-Car [5], eSafety [6]) in recent years. VANETs are envisioned as next generation intelligent vehicular networking technologies where moving cars are treated as nodes. VANETs that work on short range wireless communications, are capable of working in two environments: vehicle talking to vehicle (V2V) or vehicle talking to roadside infrastructure (V2I) [7]. They are also seen favourable for future data delivery networks for many applications (e.g., traffic safety, traffic jams, location-based services) as future smart cars are equipped with GPS, sensors, computing and communication devices in order to harness the power of technologies.

VANETs are promising for multi-hop data delivery because of their specific characteristics such as high level of mobility in constrained and predictable city/highway road networks. It makes end-to-end connection possible between a source and a destination which is otherwise impossible to achieve in a sparse network. Multi-hop data delivery is useful for many applications where a moving vehicle may want to query a fixed location-based service provider, or a region several miles away from the vehicle’s current position. In the first case, the vehicle may be interested for a sale in a shopping mall or for the price of fuels of a gas station. Whereas in the second case there is no fixed location-based server, and the vehicle may be interested to know the available parking spaces or current traffic conditions in a region. In this scenario, the query is first forwarded to a vehicle in the region and then broadcasted to all participants. In both cases, there is a need of addressing the data delivery challenges by collecting real-time traffic information and adopting dynamically changes of traffic condition for timely disseminating data to the recipients.

Efficient routing in VANETs is essential to deliver the data to a destination with minimum delay. The traditional routing protocols for mobile ad hoc networks (MANETs) are not suitable for VANETs due to their typical characteristics like fast movement of vehicles and frequent network partitions. To address this issues, number of geographical protocols, e.g., greedy-face-greedy (GFG) [8], greedy other adaptive face routing (GOAFR) [9], and greedy perimeter stateless routing (GPSR) [10] have been developed in VANETs. Despite better path stability geographical forwarding does not perform well in a city-based environment as it considers nodes in Euclidean space, which is not suitable for data delivery in a constrained network space, e.g., a road network. With this motivation, a number of road-based routing protocols such as MDDV [11], Gytar [12], CAR [13], and VADD [14] have been designed for city-based environment. The major limitations of these approaches is that they are based on historical data and thus
assume static traffic pattern.

Later, Nzouonta et al. in [15] has demonstrated two road-based vehicular traffic (RBVT) routing schemes: RBVT-R and RBVT-P on a city-based road network. The major limitations of these two approaches are: (i) route discovery process works only on the approximate traffic information and does not contemplate the dynamically changing traffic pattern while delivering the data from a source to a destination, and (ii) flooding in the network is relatively higher. We argue that the traffic in roads is more dynamic and therefore, there is always a need of real-time i.e. local information to accurately transfer data between a source to a destination.

Fig. 1. (a) Traffic scenario of a city environment where a source vehicle $S$ is forwarding a packet to a destination $D$ at time stamp $t_1$ (b) Traffic scenario changes when the forwarded packet reaches to intersection point $I_4$ at time stamp $t_2$.

An example is depicted in Fig. 1 to illustrate how to find a suitable path and adopt in dynamic change of traffic pattern environment. Fig. 1 (a) shows from a source $S$, we like to deliver a packet towards a destination $D$ through number of road segments. Source $S$ can forward the packet through different paths like $I_1 \rightarrow I_4 \rightarrow I_7 \rightarrow I_8$, $I_1 \rightarrow I_2 \rightarrow I_5 \rightarrow I_8$, $I_1 \rightarrow I_4 \rightarrow I_5 \rightarrow I_8$, etc. Apparently, it seems that among these path, $I_1 \rightarrow I_2 \rightarrow I_5 \rightarrow I_8$ is the shortest path basing on traffic density. But within a short time the traffic condition in the forwarding direction path may change. Therefore, we cannot progress towards a destination point with predicted/approximate traffic information only. In the next example we find the changing traffic condition in different time stamp. In this case, source $S$ calculates optimal route path $I_1 \rightarrow I_4 \rightarrow I_7 \rightarrow I_8$ at time stamp $t_1$. In Fig. 1 (b) shows that when the packet reaches at intersection $I_4$ at time stamp $t_2$, the forwarding packet find the path $I_4 \rightarrow I_7 \rightarrow I_8$ has low traffic density. In this case, the alternative path in the changing scenario can be $I_4 \rightarrow I_5 \rightarrow I_8$. After comprehending this scenario, we can firmly stand into our arguments that the traffic conditions in the road are continuously changing. We term this as dynamic traffic condition.
In this paper, we address the dynamic changing of traffic pattern by handling efficient data delivery in real-time traffic environment. We develop a novel mobility aware data delivery (MADD) for VANETs. We introduce a new concept of global window based on approximate/predicted traffic information and local window based on real-time traffic information to handle dynamic change of traffic in city-based road networks. In our approach, traffic light posts are positioned in the road intersection point and assumed to have similar processing and communication capabilities as vehicles. Traffic posts use periodic beacon messages to update the traffic conditions of their surrounding roads. Each traffic post stores real-time traffic information obtained from the received beacon messages that are sent from different neighboring traffic posts. In addition, it can handle stalemate situation when there is no traffic within vicinity. Static traffic light post positioned in the road intersection point can temporarily hold data and transfer when its find suitable carrier vehicle within vicinity.

We adopt carry and forward approach [14] that ensures the packet delivery in a sparse network, where a vehicle carries the packet until it encounters a new suitable vehicle in its communication range and forwards the packet to the new vehicle. In this paper, we develop three approaches: global window based approach (MADD-L), local window based approach (MADD-L) and hybrid approach (MADD-H). First, we trigger our methodology with global window approach (MADD-G) that gives intuitive idea to predict/approximate the traffic scenario by limiting a boundary global window from a source to a destination. In the second approach, we utilize this intuition and progress on the real-time traffic by defining local window in local window based approach (MADD-L). The key idea of this approach is to limit the boundary space within immediate neighbours. Lastly, we blend the two concept of global (predict/approximate) and local (real-time traffic) information to find a best possible solution to reach towards destination in hybrid approach (MADD-H).

In summary, the contribution of this paper are as follows:

- We address the problem of dynamic change of traffic pattern by defining local window and collecting real-time traffic information.
- We propose three approaches: global window based approach (MADD-G), local window based approach (MADD-L) and hybrid approach (MADD-H).
- We conduct extensive set of experiments to demonstrate the effectiveness and efficiency of our proposed approaches in different scenarios.

The remaining part of this paper is organized as follows. In Section II, we give a brief summary of all related works including their strengths and weaknesses. In Section III, we structure MADD model. In Section IV, we present our MADD methodology that includes three approaches. Section V gives the
performance evaluation and comparison of our approaches. Lastly, in Section VI we conclude.

II. RELATED WORK

Data delivery through efficient routing has been a major research topic in VANETs. In the early stage of development in VANETs, routing protocols (e.g., AODV [16], DSDV [17], DSR [18] and OLSR [19]) for MANETs were used. But these traditional routing protocols of MANETs could not satisfy the requirement of VANETs [1], [13] due to its typical characteristics like fast movement of vehicles, frequent network partitions. As a result, researchers have designed many routing protocols to have smooth data delivery technique. These protocols can be categorized on topology based, position based, geo-cast based, cluster based, broadcast based and infrastructure based. In this section, we mainly touch upon the existing routing protocols including its strength and weaknesses that relates to multi-hop data delivery technique in VANETs.

A. Position Based Routing

Karp and Kung [10] introduced Greedy Perimeter Stateless Routing (GPSR) for wireless network. Lochert et al. identified several limitations of GPSR to implement in VANET especially in city environment and proposed GSR [1] that combines position based routing with topological knowledge. But in low traffic density, it makes difficult to find end-to-end connection in preselected path. Thus, Lochert et al. identified a new methodology of data delivery technique, Greedy Perimeter Coordinator Routing (GPCR) [20] which works without having access to road maps and became more realistic in city environment.

Position-based routing for Inter Vehicle Communication System (IVCS) in a built-up city environment faces greater challenges because of potentially more uneven distribution of vehicular nodes, constrained mobility, and difficult signal reception due to radio obstacles such as high-rise buildings. Seet et al. [21] proposes a new position-based routing scheme called Anchor-based Street and Traffic Aware Routing (A-STAR), designed specifically for IVCS in a city environment. Though A-STAR achieves obvious network performance improvement compared with GSR and GPSR, but routing path may not be optimal because it is along the anchor path following the city streets.

B. Road Based Routing

Wu et al. [11] proposed MDDV, a mobility-centric approach for data dissemination in vehicular networks designed to operate efficiently and reliably despite the highly mobile, partitioned nature of
these networks. MDDV is designed to exploit vehicle mobility for data dissemination, and combines the idea of opportunistic forwarding [5][6], trajectory based forwarding [8] and geographical forwarding [9]. MDDV uses road map and traffic density into considerations, that is how the source node extends its trajectory to the destination. The node carrying the message which is closer to the destination is called the message head. The message head is responsible to forward the message to destination. The role of message head might change if the message do not find its neighbor within the communication range and ultimately message can be lost. MDDV has limitation of using static traffic density and road map network.

Greedy Traffic Aware Routing (GYTAR) [12] is an improved routing protocol for VANETs in City Environments. It contains two modules: Junctions selection, forwarding data between two junctions. A packet will pass through junctions to reach its destination. In junction selection process a value is given to each junction by comparing the traffic density between the current junction and the next candidate junction and the curve metric distance to the destination. The junction with highest value will be chosen for packet forwarding. In second module each vehicle maintains a table which contains position, velocity and direction of each neighbor vehicle and the table is updated periodically. Thus, when a packet is received, the forwarding vehicle computes the new predicted position of each neighbor using the table and then selects the next hop neighbor which is closer to the destination junction which may cause packets in a local optimum. To overcome this problem GYTAR uses store and forward strategy.

Later, Zhao and Cao addressed the issues to support multi-hop delivery technique in sparsely connected network by Vehicle Assisted Data Delivery (VADD) tool in their work [9]. VADD uses the idea of carry and forward, where a moving vehicle carries a packet until a new vehicle moves into its vicinity and forwards the packet. VADD handles important event while forwarding a packet at an intersection. For this in the intersection mode, the VADD designed the forwarding protocols: Location First Probe (L-VADD), Direction First Probe (D-VADD) and Hybrid Probe (H-VADD). The idea of Direction First Probe (D-VADD) can be extended to MD-VADD, in case when the carrier can deliver the packet meeting multiple roads connected to an intersection. On the other hand, H-VADD technique make a trade-off between both location and direction of a vehicle while selecting a road. VADD has many important strengths; guaranteeing an end-end connection in a sparse network, use of carry and forward approach with predictable mobility. However, it does not sustain long in real time traffic environment.

Nzouonta et al. in [10] has demonstrated two Road-based vehicular traffic (RBVT) routing schemes: RBVT-R and RBVT-P on city-based road network. RBVT-R is a reactive routing protocol which initiates a route discovery packet and broadcast along roads to the destination and in turn destination unicast
a route reply packet back to source. While RBVT-P is a pro-active routing protocol which periodically disseminate all its status and create a route connectivity information. The major limitations of these two approaches are: (i) route discovery process works only on approximate the traffic information (ii) thus it does not contemplate the dynamically changing traffic pattern while delivering the data from a source to a destination, and (iii) flooding in the network is relatively higher.

Khokhar et al. in [22] proposes a Fuzzy-assisted social-based routing (FAST) protocol that bring a new concept of adopting of social behaviour of humans on the road to make optimal and secure routing decisions. FAST uses prior global knowledge of real-time vehicular traffic for packet routing from the source to the destination. In FAST, fuzzy inference system leverages friendship mechanism to make critical decisions at intersections which is based on prior global knowledge of realtime vehicular traffic information. However, it is not much useful to dynamically assess real time traffic and for fastest multi hop data delivery application. In contrast to our work, it does not account both global and local knowledge of traffic to make realistic decision.

C. Geographic Forwarding

Dhurandher et al. in his work [23] proposes geographic routing over VANETs called GROOV, which takes into account varying topographies and densities of highways as well as cities. To increase reliability, GROOV calculates transmission feasibility for each node, based on link quality (average acceleration), range weight (weightage to relative positions of nodes) and direction, instead of traditional greedy forwarding, in the selection of the next relay node. Taking volatility of critical city intersection scenarios into account, GROOV calculates new node coordinates of vehicles at intersections to make best route selections at intersections and thus, routes the data packet through the path directed at the intended recipient. This prevents the occurrence of a routing loop, thereby, decreasing delay and increasing packet delivery ratio. To compare GROOV with our approach, it is evident that GROOV does not consider a combination of predicted and real-time traffic information rather it mainly improves on the traditional geographical forwarding technique by considering intersection of road and densities. GROOV has a limitation of grouping the vehicles in the intersection area to make a routing decision which may lead to dead-end due to the absence of traffic in the intersection area for sometime.

In each aforementioned cases, there is a limitation of ignoring combined knowledge of global (predict/approximate) and local (real-time) traffic information for handling dynamic traffic density. In this paper, we trade off between predicted and real-time traffic and develop a state of art: MADD to efficiently handle the dynamic traffic and assess a suitable path for multi hop data delivery. In the next section,
we formally introduce MADD to facilitate easy understanding on the core issues of our methodology subsequently.

III. MADD Model

This section presents the mobility aware data delivery (MADD) model that portrays system model, key concepts, and system overview of MADD approaches.

A. System Model

We envisage our system model as depicted in Fig. 2 which shows a portion of city road environment. We consider each vehicle is equipped with some devices: GPS for determining its own location; short range wireless equipment to communicate with its neighboring vehicles; on board sensors for determining vehicle’s speed and direction; digital maps providing city-based road network. In addition, Vehicle’s communication devices are compatible to communicate and process real-time traffic information with neighboring traffic post. We assume that traffic light posts or road side unit (RSU) are installed in every road intersections. They have embedded system application which are capable of computing and collecting real-time traffic information. We also consider in a stalemate traffic condition where the traffic light post can keep the deliverable data packet for some time till it finds suitable vehicle within its vicinity to deliver. Our system model is designed to work basically in vehicle to vehicle (V2V) dissemination approach.

Traffic post periodically relays an initiating message in different directions of the road segments. Traffic post appreciates total accumulated traffic scenario by forwarding the initiating packet in different directions with the assistance of vehicle’s communication devices. This traffic scenario may include average speed of the vehicle, direction, and number of the vehicles within the current road segment. In this fashion, the traffic pattern of one intersection can be relayed and disseminated to neighboring segments.

B. Key Concepts

In order to develop methodologies for MADD in VANETs, we structure our work basing on two concepts: First, a vehicle is exposed to real-time traffic information to its neighboring road segments than its distant road segments through the traffic light post; second, as the data packet proceeds more nearer to the destination vehicle, the carrier becomes more up-to-date about traffic patterns of the road segments connecting the destination. Initially, the source vehicle adopts carry and forward approach to deliver the packet to the next suitable vehicle within its communication range. When the packet completes its travel path in the first leg of the initial road segment and reaches to the immediate intersection point, then it
requires the routing decision to choose next road segment. For this, it seeks the assistance of traffic post
to get the real-time traffic scenario to choose the direction of next road segment. This way it completes
the next leg of the travel path and proceeds nearer to the destination.

Taking above two key concepts, in the next paragraph we define key terms: global window and local
window.

1) Global Window: We need to determine a suitable path with good traffic density from a source $S$
to a destination $D$. It is quite impractical to cover all domain space of a city road network for finding
a suitable path from $S$ to $D$. Therefore, our proposition is to limit the boundary space and look for the
suitable path having good traffic within our domain space. Based on this perception, we like to define
the global window which can limit our domain space to find a better solution. Global window includes a
boundary space covering non overlapping $k$-shortest paths from a source to a destination where source
node has different exit paths. We use the most popular Dijkstra’s shortest path algorithm to find the
$k$-shortest paths.

In this paper, we refer to $k$-shortest paths function to identify maximum extents which is termed as
global window covering all $k$-shortest paths from a source node $S$ to a destination $D$.

To further explain how we obtain the boundary space of global window, let us consider two points($S, D$)
in pre-loaded grid map shown in Fig. 3. The grid map of city area depicts the road segments and
road-intersection points. From a source $S$ to a destination $D$, there may be wide range of paths which
can even fall outside the domain space of the grid map. But we need to limit the boundary to have
manageable domain space where we can find better solution. Therefore, we find possible \textit{k-shortest paths} \( \{p_1, p_2, \ldots, p_k\} \) from \( S \) to \( D \) where \( k \) is at least 3 (in case of grid map) and non-overlapping. To maximize the boundary space covering from a source to a destination, we take 3 different exits \( (p_1, p_2 \text{ and } p_3) \) of road segments from the source node which allow to find non-overlapping paths. Then, we define the global window taking the maximum extents \( (GW_{LB}, GW_{RT}) \) of \( k \)-shortest paths having a set of \( m \) intersection points \( \{I_1, I_2, \ldots, I_m\} \). This global window will assist us to derive suitable path for predicted traffic in our solution approach. However, we also need to assess suitable path for real-time traffic which requires to restrict views in immediate neighbours. Therefore, we discuss the concept of another term local window in subsequent section.

2) \textbf{Local Window}: The traffic in the road are always dynamic. The predicted traffic assessed by source might change once the source node \( S \) progress towards destination \( D \). Our main idea is that real-time traffic must fall within the road segments of immediate neighbours. Therefore, we further limit the boundary space within immediate neighbours. Fig. 4 depicts how a source node \( S \) proceeding towards destination \( D \) by selecting its each leg of travel path \( P_S \) by defining local window. First source node \( S \) draw rectangle global window \( GW \). It has predeicted/approximated traffic information within its window. As the source node \( S \) progress towards destination, it proceeds basing on real-time traffic information. For this, the carrier restricts its view within immediate neighbours and draw a rectangle taking the maximum extents of its immediate neighbouring road segments which is known as local window. As carrier moves...
from one leg to another leg of travel path, the local window is shifted till the carrier reaches towards destination.

![Diagram showing grid map of a city, global window (GW), local window (LW), source (S), destination (D), and intersection points (Ij)].

Fig. 4. Source $S$ within its global window ($GW_{LB}$, $GW_{RT}$) progress towards destination $D$ by selecting its subsequent leg of travel path $P_S$ by defining local window ($LW_{LB}$, $LW_{RT}$). The dotted arrow represents the tentative paths whereas the continuous arrow represents selected path $P_S$.

C. System Overview

Fig. 5 depicts the system overview to assess both predict/approximate and real-time traffic conditions. When a user request for delivering data from a source to a destination, it identifies points (source/destination) in the pre-loaded grid map. In the next step, it finds $k$-shortest paths between a source and a destination. Then, it takes maximum extents of $k$-shortest paths to draw a rectangle and then define the global window boundary space as described in problem formulation section. Now to deliver the data from a source to a destination it adopts one of the three approaches: MADD-G, is a global window based approach which approximate total traffic scenario within global window; MADD-L, a local window based approach assess its immediate neighbours situation within local window (real-time traffic); MADD-H blends two approaches MADD-G and MADD-L which assess both global (approximate) and local (real-time) traffic information.

IV. OUR APPROACH

After comprehending above MADD model, we like to formulate the three approaches; global window based approach (MADD-G), local window based approach (MADD-L) and hybrid approach (MADD-H). In following subsection we will discuss in details the methodology including the algorithm of three different approaches.
A. Global Window Based Approach (MADD-G)

In this approach, a source node collect traffic density of all road segments within the global domain space to deliver the data to a destination. We term this approach, global window based approach i.e. mobility aware data delivery within global window (MADD-G). Here, the source assesses the best possible path basing on the predicted traffic density throughout the global window. To explain this approach, first we describe the methodologies and then discuss the algorithm subsequently.

1) Methodology: This approach performs its methodology in few steps: (i) request for traffic density information(RDI) (ii) collect density information by the source node (iii) deliver predicted data (PDD) i.e. data delivery basing on predicted traffic density information (iv) lastly, update path information (PUI). The steps are explained in details as follows.

- **Request Density Information(RDI):** When a source node needs to deliver a packet to a destination node, the source node requires to initiates a request packet to the destination. This packet includes a header with source and destination location, boundary locations of global window i.e. minimum and maximum \((x, y)\) points, location of all intersection nodes it finds within the global window and the time stamp of sending the request. We assume the grid map have all the locations of intersections nodes. It only needs to determine the list of intersection nodes which fall within global window. Then, source node needs to receive the traffic density information which are dynamic. Fig. 6(a) illustrates source node \(S\) sends request packet to all intersection nodes \(I_1, I_2, I_4, I_5, I_7, I_8\) within
global window.

- **Collect Density Information (CDI):** After receiving the request for sending the density information, all the intersection nodes disseminate their neighbouring traffic density status to the source node. In Fig. 6(b) shows the source node $S$ receives the density information of all intersection nodes within the global window. By considering all the intersection nodes as list of vertices and their corresponding road segment as edge, a neighbouring list is created in table as shown Table I. The table includes the list of intersection nodes, its neighbours’ list and the traffic density information. For example, intersection node $I_1$ has neighbours $I_2$ and $I_4$ with traffic density 1 and 3 respectively.

<table>
<thead>
<tr>
<th>Node</th>
<th>List of Neighbours (Density)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$I_1$</td>
<td>$I_2(1), I_4(3)$</td>
</tr>
<tr>
<td>$I_2$</td>
<td>$I_1(1), I_5(2)$</td>
</tr>
<tr>
<td>$I_4$</td>
<td>$I_1(3), I_5(1), I_7(2)$</td>
</tr>
<tr>
<td>$I_5$</td>
<td>$I_2(2), I_4(1), I_8(1)$</td>
</tr>
<tr>
<td>$I_7$</td>
<td>$I_4(2), I_6(2)$</td>
</tr>
<tr>
<td>$I_8$</td>
<td>$I_5(1), I_7(2)$</td>
</tr>
</tbody>
</table>

- **Deliver Predicted Data (PDD):** In this step, the source node after collecting all predicted density information, needs to deliver data to the destination. It uses Dijkstra’s shortest path algorithm and find the best route basing on traffic density to reach towards a destination. From the Table I,
we can prepare the graph $G = (V, E)$ and apply the connection weight from the traffic density information to each road segment as a list of edges. Now we can apply the shortest path algorithm to find the best possible route to destination. We term it as predicted data delivery since the source collects the density information which might change once it progresses towards destination. From the Fig. 6(b), we can find that after collecting traffic density information, the best path is calculated by the source node is $I_1 \rightarrow I_4 \rightarrow I_7 \rightarrow I_8$. However, carrier vehicle has two alternative options if it finds any difficulties to progress basing on the predicted route: (i) choose alternative route where the traffic is available within the vicinity (ii) hold the packet in traffic post i.e. by intersection node for sometime whenever a vehicle is unable to deliver the packet due to non-availability traffic within its vicinity. Third stage basically addresses the stalemate traffic scenario which is otherwise obvious in many situation.

- **Update Path Information (PUI):** While progressing towards destination the path information need to be updated periodically as the vehicle do not remain constant rather it is continuously changing. Path update has basically two functionalities: (i) update the route information every time the packet crosses one road segments and comes near to the next intersection (ii) inform the source node about the stalemate situation in a particular road segment e.g. when no-vehicle available.

By using the above concept we can formulate a function for MADD-G. Suppose if a data is to be delivered from a source node $S$ to a destination node $D$, then using prediction based technique the function for MADD-G is denoted as $f_G$ and can be defined as follows.

$$f_G = f_{RDI}(S, I_N, D) + f_{CDI}(D, I_N) + f_{PDD}(S, D) + f_{PUI}(I_N, D)$$  \hspace{1cm} (1)$$

where $f_G$ is calculated basing on four functions:

- $f_{RDI}$ represents the request for density information from $S$ to $D$ covering all intersections $I_N$
- $f_{CDI}$ represents collect density information backward from $D$ to $S$ including all intersections $I_N$
- $f_{PDD}$ represents data delivery basing on predicted density information and
- $f_{PUI}$ represents path update information which adjust route information.

2) Algorithm MADD-G: We first formulate the steps of MADD-G and then we present the algorithm. The step by step procedure are given below :

- **Step 1:** Identify *k-shortest paths* (where *k* paths are non-overlapping) based on the distance covering a source and a destination node and the source node have at least 3 different exit paths.
- **Step 2:** Define global window space covering maximum extent of *k-shortest paths* inclusive.
- **Step 3:** Collect traffic density (number of vehicles in a road segment at particular timestamp)
from the intersection point (traffic post) by sending probe signal periodically to neighbouring road segments.

- **Step 5**: Relay local traffic density information to its neighbor and disseminate to all intersection point within the global window space.

- **Step 6**: Build a path applying connection weight (where the number of vehicle represents as connection weight) within the graph.

- **Step 7**: Continue the steps until we receive the vehicle density information of all road segments covering the global window space.

**TABLE II. NOTATIONS USED AND THEIR MEANINGS (MADD-G)**

<table>
<thead>
<tr>
<th>Notation</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S, D$</td>
<td>Source and the destination vertex</td>
</tr>
<tr>
<td>$I_n$</td>
<td>List of intersection node $I_n={I_1, I_2, \ldots, I_n}$</td>
</tr>
<tr>
<td>$K$</td>
<td>List of shortest paths where $K = {S, I_1, I_2, D}, {S, I_4, I_5, I_6, D}$ etc.</td>
</tr>
<tr>
<td>$k$</td>
<td>Number of shortest paths where $k$ is at least 3</td>
</tr>
<tr>
<td>$Q$</td>
<td>Set of node list</td>
</tr>
<tr>
<td>$GW_{RT}, GW_{LB}$</td>
<td>Upper right top and lower left bottom point of global window</td>
</tr>
<tr>
<td>$P$</td>
<td>Ordered Path list according to their average traffic density</td>
</tr>
<tr>
<td>$NL$</td>
<td>List of neighbouring nodes</td>
</tr>
</tbody>
</table>

**Illustration of the Algorithm:** Algorithm 1 shows the steps of global window based approach. It takes the following parameter as input: a source node $S$, a set of intersection node list $I$ and a destination node $D$. Receiving a source, list of intersections and a destination node as input, it identifies $k$-shortest paths in Line 1.4. In the initial steps from Lines 1.5-1.9, first it calculates two extents i.e. lower left bottom and upper right top points. Based on the extents, it defines the global window. In the second step from Lines 1.11-1.14, it sends request packet to all intersections within the global window for traffic density information. Next step, in the Lines 1.15-1.19, it collects density information from all of intersection nodes, store it in $NL$ and maintain a list of table along with the density. Then, in Lines 1.21-1.28, from the neighbouring list and density table, it finds shortest path according to their traffic density and store in path list $P$. Lastly, it returns the ordered path list from the algorithm. We summarize the notations in Table II which are used in this algorithm.
Algorithm 1: Global Window Based Approach (S, I, D)

1.1 Initialize K, R and Q to an empty list
1.2 Initialize a priority queue P
1.3 Initialize (GW_{LB}, GW_{RT}) to NULL
1.4 K ← FindKshortestpath(S, I, D)
1.5 for each shortest path k_i of K do
   1.6   cur_extent ← FindMinMaxExtent(K_i)
   1.7   if cur_extent is wider than (GW_{LB}, GW_{RT}) then
   1.8       (G_{LB}, G_{RT}) ← cur_extent
   1.9
1.10 DefineGlobalWindow(GW_{LB}, GW_{RT})
1.11 for each intersection node I_i of I do
1.12   if any node is inside (GW_{LB}, GW_{RT}) and I_i ≠ D then
1.13     RequestDensityInfo(I_i)
1.14
1.15 for each I_i of node list of I do
1.16   while I_j = Imm_neighbour(I_j) do
1.17     if NL[I_i][I_j] ≠ (I_i, I_j) then
1.18       Enque(I_i, NL[I_i][I_j], Find_tr_density(I_i, I_j))
1.19
1.20 Q ← insert(I)
1.21 while Q ≠ NULL do
1.22   if u ≠ D then
1.23     u ← extract_min(Q)
1.24     R ← AddNode(R, u)
1.25     for each node v is a neighbour of u do
1.26       P_i ← UpdateShortestPath(u, v, w)
1.27       Enqueue(P_i, UpdateDensity(u, v))
1.28
1.29 B. Local Window Based Approach (MADD-L)

When we predict the traffic scenario in global domain space, it is unable to provide nearly accurate real-time picture. Because, when the carrier passes one leg of travel path and proceeds to another intersection
point, it may find the traffic density has changed from what was comprehended earlier. Therefore, we
term this approach \textit{local window based approach} i.e mobility aware data delivery using local window
\textit{(MADD-L)}. The key idea of \textit{local window based approach} is that we consider real-time traffic must remain
within the local window. The motive behind defining the local window space is to determine the traffic
density of immediate neighboring segment from a traffic post or intersection point. After completing the
delivery of a packet carrier from current road segment, the local window is shifted to the next boundary
space where the traffic density more. The methodology of this approach are discussed subsequently.

1) \textbf{Methodology}: This approach performs its methodology in two steps: (i) collect neighbour density
information\textit{(CNI)} (ii) deliver data based on real-time traffic \textit{(RDD)} i.e. data delivery basing on real-time
traffic information. The detail methodology of this approach is illustrated further with an example.

- \textbf{Collect Neighbour Information\textit{(CNI)}}: In this step, the every carrier vehicle before proceeding
towards next neighbouring road segment seek for traffic density of its immediate neighbours. Thus
the carrier vehicle decides to proceed towards more traffic available road segment. Collecting
local surrounding information has much superiority over the global distant (predicted/approximate)
information which is MADD-G. Fig. 7(a) illustrates the source node $S$ first defines the local window
as described in problem formulation Section 3.2 taking the neighbouring intersection nodes $I_1$, $I_2$
and $I_4$. Then it progress each leg of travel path and shifts the local window as shown in Fig. 7(b)
till it reaches towards a destination node $D$.

- \textbf{Deliver Data Based on Real-time Traffic \textit{(RDD)}}: Based on the local traffic information, the carrier
vehicle first decides to select its next route. Fig. 7 depicts a scenario where carrier $c_1$ at some
point of time finds his next route of travel path $I_1 \rightarrow I_4$. This step has few functionalities: (i) select the road segment where the density is more (ii) select the road segment preferably which direction is towards destination (iii) stay with present carrier vehicle if it does not find any suitable next carrier within its vicinity but present carrier changing the road segment towards the forwarding direction or even it moves little away from destination but within global window (iv) leave the packet to nearest traffic post once its unable find to forward to next carrier. In Fig. 8(a) illustrates carrier vehicle $c_1$ does not find any carrier in the next two road segments either $I_4 \rightarrow I_5$ or $I_4 \rightarrow I_7$. Since the carrier vehicle $c_1$ is proceeding towards next road segments which is in the forwarding direction, therefore data packet decides to stay in $v$. In the next Fig. 8(b) depicts another scenario where carrier vehicle $c_2$ is proceeding outside the global window thus carrier vehicle $v$ transfer data packet to immediate traffic post i.e. intersection point $I_4$.

![Fig. 8](image_url)

Fig. 8. MADD-L in real-time data delivery handling stalemate traffic condition (a) stay with present carrier vehicle when there is unavailability of traffic in forwarding road segment and the present carrier moving to next road segment (b) present carrier may leave the packet to nearest traffic post due to unavailability of traffic in the next road segment.

By using the above concept we can formulate a function for MADD-L. Let us see an example before we formulate function for MADD-L. Fig. 9 depicts a road segment where carrier vehicle $c_r$ decides its next carrier from its neighbouring road segment. Carrier vehicle $c_1$ positioned in road intersection point $I_1$ need to identifies its next carrier. It has three neighbouring road segments $I_2$, $I_3$ and $I_4$ in three different directions. Based on the traffic density it decides to transfer its carrier to next vehicle $c_2$.

If a packet is to be delivered from a source node $S$ to destination node $D$, then using locality-based technique to find nearest neighbour, the function for MADD-L is denoted as $f_L$ and it is defined as
follows.

\[ f_L = \sum_{i=1}^{N} \{ f(C_k)_i + f(D_i) \} \] (2)

where the notations of above functions are defined as follows:

- \( f(C_k) \): Collect density information of all neighbours of \( i \)th intersection node
- \( f(D_i) \): Deliver the data to best immediate neighbours in \( i \)th intersection node
- \( i \): Number of intersection nodes traveled in \( i \) steps where \( i = 1 \) to \( N \)
- \( k \): Path or route direction number for example from \( S \) to \( D \) there are number of intersection points and every intersection point has at least 3 neighbouring road segments in three different forwarding directions.

Fig. 9. Carrier vehicle \( c_1 \) collect local density through intersection point \( I_1 \), then decide to find next route \( k_2 \) among three routes basing on traffic density

2) Algorithm MADD-L: We first formulate the steps of MADD-L and then we present the algorithm. The step by step procedure are given below:

- **Step 1**: Define local window space covering source vehicle and its immediate neighbouring segments.
- **Step 2**: Forward the packet from source vehicle to its immediate neighbouring vehicle until it reaches to destination checking following conditions:
  - **Condition 1**: Select the road segment where the density is more.
  - **Condition 2**: Select the road segment preferably which direction is towards destination.
\begin{itemize}
\item \textit{Condition 3}: Stay with present carrier vehicle if it does not find any suitable next carrier within its vicinity even it moves little away from destination but within global window.
\item \textit{Condition 4}: Leave the packet to nearest traffic post once it unable find to forward to next carrier.
\end{itemize}

\begin{table}
\centering
\caption{Notations used and their meanings (MADD-L)}
\begin{tabular}{|l|l|}
\hline
Notation & Meaning \\
\hline
\(I_n\) & List of intersection node \(I_n=\{I_1, I_2, \ldots, I_n\}\) \\
\& \textit{Neighbours} & List of neighbouring intersection nodes with point \\
\textit{tr\_density} & List of traffic density of neighbouring intersection nodes \\
\(LW_{RT}, LW_{LB}\) & Upper right top and lower left bottom point of local window \\
\(P\) & Path list according the sequence of intersection node visited \\
\hline
\end{tabular}
\end{table}

\begin{itemize}
\item \textbf{Illustration of the Algorithm}: Algorithm 2 shows the steps of local window based approach. This algorithm initially takes the following parameter as input: a source node \(S\), a destination node \(D\), and a list of intersection node \(I\). In the Line 2.8, it identifies the immediate neighbours of source node. Form Lines 2.9 - 2.19, it gradually visits its immediate neighbours and progress towards destination. First, it defines its local window within its immediate neighbours. Then, assessing the traffic density, it selects the path to proceed towards destination. Lastly, the selected neighbours are stored in the priority queue \(P\) which is returned at the end of the algorithm.
\end{itemize}
Algorithm 2: Local Window Based Approach (S, I, D)

2.1 Initialize \((LW_{RT}, LW_{LB})\) to NULL
2.2 Initialize \(P\) to an empty list
2.3 Initialize \(tr\_density\) to NULL
2.4 Initialize \(cur\_node\) to NULL
2.5 Initialize \(cur\_density\) to NULL
2.6 Initialize \(cur\_neighbour\) to NULL
2.7 \((LW_{LB}, LW_{RT})\) ← NULL
2.8 \(cur\_node\) ← find_neighbour(S)
2.9 while \(cur\_node \neq D\) do
2.10 for each neighbour \(j\) of \(cur\_node\) do
2.11 \(update(LW_{LB}, LW_{RT})\) ← find_extent(cur_node, neighbour(j))
2.12 Define_Localwindow(LW_{LB}, LW_{RT})
2.13 for each neighbour \(k\) of \(cur\_node\) do
2.14 \(tr\_density[k]\) ← find_density(neighbour(k))
2.15 if \(tr\_density[k]\) is promising than \(cur\_density\) then
2.16 \(cur\_density\) ← \(tr\_density[k]\)
2.17 \(cur\_neighbour\) ← neighbour(k)
2.18 Enqueue(P, cur_neighbour, cur_density)
2.19
2.20 return \(P\);

C. Hybrid Approach (MADD-H)

Both the aforementioned approaches consider their domain space either locally or globally. They have a limitation of ignoring combination of both local or global traffic situation while approaching towards destination. Therefore, the alternative course of action can be combining both local and global approaches. Our rationale is, we just can not progress with only real-time traffic ignoring the surrounding traffic situation of immediate neighbours from a source node leading to a destination. Thus, we term this approach hybrid approach i.e hybrid mobility aware data delivery (MADD-H). The main idea of MADD-H has two aspects: (i) considering local window within the immediate neighbours of source node where real-time traffic exists and (ii) considering the predicted scenario within the global window (excluding
the local window) i.e. it starts from the immediate neighbours of source node and covers all intersection nodes and then lead up to destination.

The key concepts of MADD-H are illustrated in Fig. 10 in time perspective. Let us consider carrier vehicle $c_r$ is proceeding towards intersection node $I_1$. It has the neighbouring intersection nodes $I_2$, $I_3$ and $I_4$. It first defines local window $LW$ within its neighbouring intersection nodes and collect traffic density which is considered in current time $t$. Thus the blue rectangle depicts the local window defined in current time. While the carrier vehicle also needs to acquire the surrounding distant traffic information which is outside the local window but within global window. The surrounding distant traffic information are considered as predicted/approximate one which are dynamic. The global window $GW$ provide traffic information starting from time $t-1$, $t-2$,..etc which shown in red color rectangle.

We can further explain this approach by formulating a function for MADD-H. For determining MADD-H, we combine both the function of MADD-L and MADD-G where we applied weight $W_L$ and $W_G$ to each function respectively. The function for MADD-H is defined by $f_H$ as follows.

$$f_H = W_L * f_L + W_G * f_G \quad (3)$$

![Fig. 10. MADD-H considers the local window in current time $t$ while global window considers in predicted/approximated time ($t-1$, $t-2$,..etc).](image)

The calculation of $f_L$ and $f_G$ will be little different in time perspective as depicted in Fig. 10. Suppose at time $t$ from source $S$ to its immediate neighbour $N_S$ (where $N_S$ might have three neighbours $N_A$, $N_B$, $N_C$ etc). Therefore we count the number of steps $i=1$ to $M$. The local window based function $f_L$ is defined
in current time $t$ as follows. Other notations used are similar to function $f_L$ as described in local window based approach.

$$f_L(t) = \sum_{i=1}^{M} \{f(C_k)_i + f(D)_i\} \quad (4)$$

and from immediate neighbours of $S$ i.e. $N_S$ to destination $D$, the global window based function $f_G$ is defined as follows. Other notations used are similar to function $f_G$ as described in global window based approach.

$$f_G(t-n) = f_{RD1}(I_N, D) + f_{CD1}(D, I_N) + f_{PDD}(S, D) + f_{PU1}(I_N, D) \quad (5)$$

Since $f_L$ is calculated basing real-time traffic density of immediate neighbours, therefore the cost of weight $W_L$ is considered much higher, whereas the cost of weight $W_G$ is considered lower value since its density is taken at $(t-n)$ time period to make a trade off between global and local traffic density.

Using the above function of each individual approaches, in the next section we evaluate their performances and compare the experimental results MADD-H and MADD-L with the base line approach MADD-G to identify the best approach.

V. PERFORMANCE EVALUATION

In this section, we evaluate the performance of three $MADD$ approaches, namely MADD-G, MADD-L, and MADD-H. We measure the performance of multi hop data delivery in different intensity of dense traffic environment. We use the similar concept of $VanetMobisim$ [24], [25] using both macroscopic and microscopic traffic model. $Macroscopic$ model [26] treats traffic at large scale where simulation takes place on a section-by-section basis rather than by tracking individual vehicles. For this, we use a portion of city segment which depicts the vehicular mobility of Manahattan grid-based model [27]. At the $microscopic$ level, it uses intersection management, multiple lanes, roads, speed, traffic posts etc. Our MADD tool is developed in java environment which includes vehicular mobility generations and simulate runtime mobility aware multi hop data delivery to implement and test our proposed MADD algorithms.

To evaluate the performance of our approaches in contrast to other methodologies, we compare with most widely used geographical forwarding schemes GPSR [10] and GPCR [20], and road based routing scheme RBVT-R [15]. The rationale for incorporating three techniques to compare and contrast with our methodologies are presented as follows. GPSR is elementary geographical forwarding schemes which progress on local information and follows greedy technique. On the other hand, GPCR is an enhancement of GPSR which uses city road map forming a planer graph. Both uses local information which will facilitate to compare with our technique, MADD-L. Again, RBVT-R is road based reactive routing
technique which discover routes on demand and has much similarity with our baseline approach MADD-G except having an additional procedure of maintaining/updating path information basing on real time traffic.

A. Evaluation Criteria

We compare the performance of MADD-L and MADD-H with the base line approach MADD-G. MADD-G basically predict/approximate the traffic conditions as worked in traditional routing schemes and thus we consider MADD-G is our baseline approach. It discovers route on demand but cannot handle dynamically changing traffic pattern efficiently. MADD-L works on its immediate neighbours by restricting its view within local window. It transfers data to its immediate neighbour after collecting real-time traffic density status. Though it does not allow the packet to be delivered to a vehicle which is moving outside of the global window boundary but it ignores the surrounding status of its immediate neighbours. Therefore, we introduce a third approaches MADD-H, that blends the idea both MADD-L and MADD-G by taking account not only on its immediate neighbours but also the prediction based traffic information leading towards destination. Thus, we conduct our experiment to compare the performance of MADD-L and MADD-H with our baseline approach MADD-G.

In previous section, we defined the function \( f_H \) for MADD-H where the weight factors \( W_L \) and \( W_G \) are mentioned in equation 3. While conducting the experiment we need to identify the performance of MADD-H varying the weight factors considering total weight value 1. For this, in the initial stage of experiment we varied the weight factor \( W_L \) ranging from 0.5 (minimum value) to 0.9 (maximum value) while \( W_G \) ranging from 0.1 (minimum value) to 0.5 (maximum value) to observe the performance of MADD-H. Our rationale for considering relatively higher values of \( W_L \) than \( W_G \) are: (i) function of \( f_L \) of MADD-L accounts real-time traffic condition (ii) function \( f_G \) of MADD-G accounts predicted/approximate traffic condition. After identifying the best weight factors of \( W_L \) and \( W_G \) for MADD-H, we continue our experiment in subsequent stages.

Furthermore, we simulate various traffic environment since we argue that the multi hop data delivery largely depends on the availability of traffic in the road network. We observe both unevenly and evenly distribution of traffic in our simulation area and then compare the overall performance.

B. Metrics

In this subsection we present the following metrics for comparing the performance of three different approaches.
• **Number of hops** - To deliver the packet from a source node to a destination node the number of hops required.

• **Number of stay in traffic post** - Number of car carrying the packet fail to find his neighbourhood car due to the absence of traffic within his vicinity, thus can deliver the packet to its nearby traffic post.

• **Average delay** - Average delay time caused due to either holding the packet by traffic post or by the carrier vehicle itself when it unable to find traffic within its vicinity.

• **Average delivery time** - Total average time taken to transfer the packet from a source node to a destination node.

• **Delivery success rate** - Number of times data successfully delivered from a source node to a destination node in contrast to total number of simulation runtime. This metric also leads to identify the data delivery failure rate.

### C. Simulation Setup

Our simulation scenario works on *Manhattan grid* based model shown in Fig. 11. It uses 7000 m x 7000 m rectangle area where the grid is constructed by horizontal and vertical road segments. We place the traffic post in every intersection to collect the traffic information from its neighbour and keep data if unable to deliver due to non-availability of traffic for some time. We consider the number of vehicle mobility generations 500, 650 and 800 in *unevenly distribution of traffic* within our simulation area. We term the 800 vehicles as relatively higher dense, 650 vehicles as medium dense and 500 vehicles as low dense traffic scenario.

![Manhattan grid based Layout](image)

**Fig. 11.** Manhattan grid based Layout

Besides, we have also considered other two categories of density i.e. very low and very high in *evenly*
distributed traffic environment during simulation to assess the off-peak and peak hours. At times, there are situation when there are very low traffic in off-peak hours or huge traffic during the peak hours all over the city area. Here, we observed two metrics: average delivery time and average delay in traffic posts. We discarded first 100 sec of vehicle movement generations to have more realistic performance in our results. Inter vehicle communication range is considered taking the general guideline of Dedicated short range communication [28] standard from 500 m to maximum 1000 m. To ensure smooth and reliable communication, we specify 250 m as our vehicle communication range due to the presence of high rise buildings and obstacles within the city area.

Furthermore, we depict the simulation scenario as shown in Table IV.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simulation area</td>
<td>7000m x 7000m</td>
</tr>
<tr>
<td>Number of Lanes</td>
<td>2</td>
</tr>
<tr>
<td>Number of vehicles</td>
<td>Unevenly distributed - 500(low), 650(medium) and 800 (high) density</td>
</tr>
<tr>
<td>Number of vehicles</td>
<td>Evenly distributed traffic 250 (very low) and 1200 (very high) density</td>
</tr>
<tr>
<td>Communication range</td>
<td>250m</td>
</tr>
<tr>
<td>Vehicle velocity</td>
<td>50 to 60 km per hour</td>
</tr>
<tr>
<td>Simulation time</td>
<td>400s</td>
</tr>
</tbody>
</table>

D. Simulation Results

The Simulation results is presented in two stages. First, we tried to visualize the normal traffic situation in city environment. We observe for three different densities of vehicle nodes to identify the overall impact of traffic density in five metrics. We start the initial simulation minimum twenty times then calculate the average on each metrics to obtain the expected results. However, we ran maximum simulation time for 50 times in two key metrics (average delivery time, average delay) to identify the variances of the results. The results are formulated as follows.

1) Unevenly Distributed Traffic Conditions : In unevenly distributed traffic environment, we started our simulation in two phases. In the first phase we consider varying weight factors of $W_L$ and $W_G$ in two key metrics i.e. average delivery time and average delay to identify the best weight factor for MADD-H. In the second phase we have incorporated the best weight factor and carried out detail experiment
to evaluate in five metrics and compare MADD-H with MADD-L, MADD-G and other methodologies GPSR, GPCR and RBVT-R.

The first phase of the simulation results are presented in two key metrics as follows.

- **Average delivery time** - We have observed the performance of MADD-H by varying the weigh factors of MADD-H as shown in Table V. The result shows that the average delivery time relatively higher in two cases: (i) at maximum value of $W_L$ and minimum value of $W_G$; (ii) at minimum value of $W_L$ and maximum value of $W_G$. But MADD-H needs maximum delivery time in first case. MADD-H performs best while considering $W_L$ equivalent to 0.7 and $W_G$ equivalent to 0.3 weight factors respectively.

<table>
<thead>
<tr>
<th>Density</th>
<th>MADD-H</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$W_L(0.9)$</td>
</tr>
<tr>
<td></td>
<td>$W_G(0.1)$</td>
</tr>
<tr>
<td>Low</td>
<td>224</td>
</tr>
<tr>
<td>Medium</td>
<td>102</td>
</tr>
<tr>
<td>High</td>
<td>67</td>
</tr>
</tbody>
</table>

- **Average delay** - During evaluation of average delay, we have also varied the weigh factors of MADD-H and observed few changes comparing to average delivery time. At higher weight of $W_L$ and lower weight of $W_G$, average delay is comparatively less. This is due to delivering data by assessing real traffic conditions will certainly require less delay. However, at 0.7 weight factor of $W_L$ and 0.3 weight factor of $W_G$ still gives best result. The results are projected in Table VI.

<table>
<thead>
<tr>
<th>Density</th>
<th>MADD-H</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$W_L(0.9)$</td>
</tr>
<tr>
<td></td>
<td>$W_G(0.1)$</td>
</tr>
<tr>
<td>Low</td>
<td>28.31</td>
</tr>
<tr>
<td>Medium</td>
<td>9.02</td>
</tr>
<tr>
<td>High</td>
<td>2.18</td>
</tr>
</tbody>
</table>

During the second phase, in unevenly distributed traffic conditions the simulation results are presented in five metrics considering fixed weight factors of MADD-H (incorporating best weight factors from previous results). The results are as follows.
• Number of hops - Fig. 12 shows number of hops required to transfer a packet from a source to a destination. During the runtime of simulation it is observed that with the intensity of the traffic, the number of hops increases and it becomes very much easy for source packet to deliver to a destination node with the increasing density of traffic. We plot the number of hops in terms of three categories of density and MADD-L requires less number of hops than MADD-G. There are two rationales for this: (i) MADD-L progress by assessing more dynamic traffic (ii) it follows route where it finds more traffic density. In case of MADD-H, we can much optimize the overall number of hops comparing to MADD-L and MADD-G, thus it allows fastest data delivery in overall results. In case of GPSR and GPCR, we find they requires more number of hops as they do not follow the path on the road network. Sometimes their progress on local information unnecessary delay the optimum solution. However, RBVT-R performs better than both MADD-G and MADD-L in low density of traffic but with the increasing of density, both MADD-H and MADD-L required less number of hop than RBVT-R. This may be due to the increase of traffic density, both techniques can have better adaptability to find optimum solution.

![Bar chart showing number of hops in different traffic densities for MADD-L, MADD-H, GPSR, GPCR, and RBVT-R.](image)

Fig. 12. Number of Hops required to transfer data from a source to a destination in MADD-L, MADD-G, MADD-H, GPSR, GPCR and RBVT-R.

• Number of stay in traffic post - Due to non-availability of traffic in immediate vicinity sometimes it becomes difficult to find suitable route and packet remain stand still. To handle this type of exceptional situation, we allow the the packet carrier to be handed over to its immediate traffic post and stay for sometime. This might cause some delay but we observed that it has two advantages: (i) avoid routing loops (ii) ensuring guaranteed delivery towards destination. We run the twenty times of simulation and make the average of number of stay. We evaluated this for our three approaches.
Since other techniques do not have this method therefore it is limited to our methodologies only. However, we have measured the delay occurrences for keeping packet carrier in average delay metric. The results projected in Fig. 13 where we find MADD-H required less number of stay in traffic posts in contrast to MADD-L and MADD-G. Since MADD-G calculates static traffic scenario beforehand, therefore after traveling few legs, it finds difficulty to transfer packet due to non-availability of traffic in predicted path. We also observed that with the increasing intensity of traffic, specially in the high density of traffic the average stay can be reduced significantly in MADD-H.

![Fig. 13. Number of stay in traffic post in MADD-L, MADD-G and MADD-H](image)

- **Average delivery time** - We compute the average delivery time of three approaches in contrast to other three techniques GPSR, GPCR and RBVT-R. First, we recorded the results after running twenty times of simulation and projected in Fig. 14. We observe, delivery time of MADD-H is much less than MADD-G and MADD-L. Since MADD-G, progress on predicted traffic density therefore, it takes more time to adjust its travel path. We also find that MADD-H can be further optimized and reduced to almost 57% less by increasing from low to medium density and nearly 82% from low to high density of traffic. While evaluating the results of our approaches in contrast to GPSR, GPCR and RBVT-R, we find MADD-G take less time comparing with GPSR and GPCR but more time than RBVT-R. This is due to calculating the path basing on prediction based information by MADD-G. But MADD-L and MADD-H required less delivery time than RBVT-R. In addition MADD-H performs much better in medium and high density of traffic. It can be reduced to half delivery time in high density traffic comparing with MADD-L and RBVT-R. The reason is MADD-H not only gives more preference on local information i.e. real-time traffic but also account for
both surrounding information of immediate neighbours and distant information where its ultimate
destination locates. While running simulation for fifty times we did not find major differences to
record the results.

![Graph](image)

Fig. 14. Average delivery time required to transfer data from a source to a destination in MADD-G, MADD-L, MADD-H,
GPSR, GPCR and RBVT-R (a) projects twenty times simulation results (b) projects fifty times simulation results.

- **Average delay** - We compute average delay in two occasions: (i) delay occurred in traffic post by
dividing total stay period in traffic posts by number of stay and (ii) delay caused by carrier vehicle
of holding it due to non-availability of traffic within its vicinity. Then we project the results in
Fig. 13. Since the overall stay in traffic post is very less in MADD-H in contrast to MADD-L and
MADD-G, therefore average delay is comparatively less in MADD-H than MADD-L and MADD-
G. Again, we observe average delay in terms of percentage in respect of delivery time in three
categories of density. We find that the average delay in low traffic density in three approaches 13%
to 16%. It can be reduced to 25% less in medium density and becomes 10% to 12% and further
reduced to less than 5%. However, We observe that MADD-H have 53% to 29% delay in contrast
to MADD-G and have 81% to 48% delay in contrast to MADD-L with the increasing density of
traffic. While comparing our approaches with other three techniques we find (i) MADD-G counts
more delay than other three techniques due to progress approximate information (ii) MADD-L
performs better than GPSR and GPCR (iii) MADD-L had more delay than RBVT-R in low density
but counts less delay in medium and high density. (iv) in all cases MADD-H outperforms all others.

- **Delivery success rate** - In Fig. 16, we observe that the delivery success rate is less in low density
of traffic which is very much obvious. Since in many occasions carrier vehicle do not find suitable
traffic even after transferring the carrier to immediate traffic post. However, the success rate is
increased with the increasing density of traffic. While compare the MADD-H and MADD-L with
baseline approach MADD-G and other techniques GPSR, GPCR and RBVT-R in terms of success rate we find both MADD-H and MADD-L performs better than others. Overall result shows that MADD-H have higher efficiency in high density of traffic and attains to almost 100% in terms of delivery success rate.

2) Evenly Distributed Traffic Conditions: In the second stage of simulation, the performance of algorithm is being observed in comparatively very low and very high density with uniform traffic situation throughout the domain space. Our intention is to observe the situation during the peak and off-peak hours. We usually find situation like very high traffic and very low traffic in the road during peak and off-peak hour in almost all the road network in the city environment. In this scenario, we found the results are quite different than first stages of simulation results. Three approaches along with other three techniques
have very meagre differences. However, in contrast to MADD-G, MADD-L, GPSR, GPCR and RBVT-R, MADD-H performs better. The results are projected below.

- Average delivery time - Fig. 17 show the performance comparison of three approaches along with other three techniques in terms of average delivery time. From the projected graph, it can be inferred that when there is very high and evenly distributed traffic through out the boundary space, finding suitable route by assessing either local and global traffic situation have eventually less difference in their results. We find in extreme low density MADD-G have less delivery time than GPSR and GPCR. But in extreme high density of traffic GPSR and GPCR required less time due to following greedy techniques basing on local information. But in both situation MADD-L had less delivery time than RBVT-R. However, in all cases average delivery time of MADD-H is less than all other techniques.

Fig. 17. Average delivery time required in very low and very high traffic condition in three techniques: MADD-L, MADD-G, MADD-H, GPSR, GPCR and RBVT-R.

Fig. 18. Average delay in very low and very high traffic condition in three techniques: MADD-L, MADD-G, MADD-H, GPSR, GPCR and RBVT-R.
Average delay - While running the simulation in extreme low traffic density and evenly distributed traffic situation, it is observed that average delay time reduces significantly to 2.5%. On the contrary, it was almost 8% to 10% in low traffic with unevenly distributed traffic. Fig. 18 shows the relative values of all the techniques have hardly in differences which differs from previous results. This is due to the fact that in evenly distributed traffic situation local and global knowledge do not have significant differences especially in very high density of traffic.

E. Summary of Results

We have conducted experiment in two stages to observe both unevenly and evenly distribution of traffic environment. In unevenly traffic environment we started the simulation varying the weight factors of MADD-H function, $f_L$, to determine the best one. We incorporated this result to our detail experiment subsequently. To identify the efficiency of our algorithm we have also considered evenly distributed traffic to demonstrate the peak and off-peak traffic scenarios where we observed MADD-H still performs better. However, we summarize the results of unevenly distribution of traffic conditions which are as follows:

- The efficiency of MADD-H in contrast to MADD-G, MADD-L, GPSR, GPCR and RBVT-R in terms of average delivery time which are projected in Table VII.

<table>
<thead>
<tr>
<th>Density</th>
<th>MADD-G</th>
<th>MADD-L</th>
<th>GPSR</th>
<th>GPCR</th>
<th>RBVT-R</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low</td>
<td>2.04</td>
<td>1.50</td>
<td>2.13</td>
<td>2.07</td>
<td>1.17</td>
</tr>
<tr>
<td>Medium</td>
<td>2.50</td>
<td>1.44</td>
<td>2.72</td>
<td>2.56</td>
<td>1.57</td>
</tr>
<tr>
<td>High</td>
<td>3.01</td>
<td>2.18</td>
<td>3.37</td>
<td>3.20</td>
<td>2.32</td>
</tr>
</tbody>
</table>

- Again, we measure the efficiency of MADD-H in contrast to MADD-G, MADD-L, GPSR, GPCR and RBVT-R in terms of average delay are projected in Table VIII.

<table>
<thead>
<tr>
<th>Density</th>
<th>MADD-G</th>
<th>MADD-L</th>
<th>GPSR</th>
<th>GPCR</th>
<th>RBVT-R</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low</td>
<td>53</td>
<td>81</td>
<td>56</td>
<td>68</td>
<td>93</td>
</tr>
<tr>
<td>Medium</td>
<td>39</td>
<td>64</td>
<td>43</td>
<td>49</td>
<td>57</td>
</tr>
<tr>
<td>High</td>
<td>29</td>
<td>48</td>
<td>33</td>
<td>36</td>
<td>44</td>
</tr>
</tbody>
</table>
• In all cases MADD-H outperforms other approaches: MADD-L, MADD-G, GPSR, GPCR and RBVT-R.

VI. CONCLUSION

In this paper, we have introduced a novel methodology for multi hop data delivery, called the Mobility Aware Data Delivery (MADD). It addresses the problem of efficient handling of dynamically changing traffic pattern. We proposed three different approaches: global window based approach (MADD-G), local window based approach (MADD-L) and hybrid approach (MADD-H). MADD-G assess traffic situation within global window (approximated traffic) and then deliver data from a source node to a destination node. While MADD-L focus on immediate neighbours and progress on local window (real-time traffic). Lastly, MADD-H assess both global (approximate) and local (real-time) traffic thus blends two ideas of MADD-G and MADD-L. We have carried out extensive experiments to demonstrate the efficiency and effectiveness of MADD-H and MADD-L with baseline approach MADD-G and other contemporary schemes GPSR, GPCR and RBVT-R.

We have considered both uneven and even (uniform) distribution of traffic in the city area in different intensity of their density. We found the average delivery time can be reduced to 57% if we increase the density from low to medium and 82% from low to high density of traffic. The average delay can be further reduced to less than 5% from 13% to 16% when the density of traffic is increased from low to high. We observe that MADD-H reduced from 53% to 29% delay in contrast to MADD-G, from 81% to 48% delay in contrast to MADD-L, from 56% to 33% delay in contrast to GPSR, from 68% to 36% delay in contrast to GPCR and from 93% to 44% delay in contrast to RBVT-R with the increasing density of traffic. We have also considered even or uniform distribution of traffic with extreme low and extreme high density to portray the off-peak and peak hour traffic conditions where we found MADD-H though performs better both in terms of average delivery time and average delay time but have meagre differences comparing to other approaches. In each aforementioned cases, MADD-H outperforms MADD-L, MADD-G, GPSR, GPCR and RBVT-R. To summarize, MADD-L (real-time) approach is always much superior to MADD-G (approximate/predicted) approach. Furthermore, blending the two techniques for obtaining both predicted and real-time traffic in hybrid approach (MADD-H) can optimize the overall results.

ACKNOWLEDGMENT

The authors would like to express their sincere gratitude and appreciation to the people around who all have provided necessary assistance and make this work a successful one. This work is done...
Samsung Innovation Lab, Department of Computer Science and Engineering of Bangladesh University of Engineering and Technology (BUET).

REFERENCES


**Syed Muzakkir Ahmed** received his B.Sc. in Computer Science and Engineering from Military Institute of Science and Technology (MIST), Dhaka, Bangladesh in 2004 and M.Sc. in Computer Science and Engineering from Bangladesh University of Engineering and Technology (BUET), Dhaka, Bangladesh. He is affiliated with the Department of Computer Science and Engineering in MIST. His research interest includes wireless networks, vehicular networks and network security.

**Mohammed Eunus Ali** is currently an Associate Professor in the Department of Computer Science and Engineering (CSE) at Bangladesh University of Engineering and Technology (BUET), Dhaka. He received his PhD degree in Computer Science and Software Engineering from the University of Melbourne in 2010. He obtained his B.Sc. and M.Sc. degrees in Computer Science and Engineering from Bangladesh University of Engineering and Technology, Dhaka, Bangladesh in 1999 and 2002, respectively. He served as a Lecturer in Computer Science and Engineering at Bangladesh University of Engineering and Technology in 2000-2003, and then served as an Assistant Professor in the same department during 2003-2012. He also worked as a Research Fellow and Visiting Research Scholar in the University of Melbourne in 2010 and 2012, respectively. Dr. Eunus research falls in the intersection of data management and mobile computing. His research areas cover a wide range of topic in database systems and information management that include spatial and multimedia databases, large-scale distributed data management, and Big Data management.

**Dr. Tanzima Hashem** is currently an Assistant Professor at the Department of Computer Science and Engineering of Bangladesh University of Engineering and Technology (BUET). She received my PhD degree in Computer Science and Software Engineering from the University of Melbourne, Australia. She has also worked as a research fellow in the University of Melbourne, Australia in 2011. Her research interest falls into the intersection of spatial data management and mobile computing. Specifically, her research works focus on protecting user privacy while accessing location-based services and participating in social networks.