ISSN 2250-3153

Enhanced Routing in Delay Tolerant Enabled Vehicular Ad Hoc Networks

Arun Kumar

R&D, Syscom Corporation Ltd, Noida, Uttar Pradesh, India

Abstract- Vehicular ad hoc networks (VANETs) are special case of mobile ad hoc networks (MANETs) where non real-time applications are appreciated as compared to real time because of very frequent network partitions caused by high node mobility. Delay tolerant network (DTN) follows the approach of store and forward i.e. message ferrying. In this paper the selection of ferry to forward the data is based on fastest ferry and dynamic speed of ferries. The selection of fastest ferry in variable blocks makes it adaptive to accommodate high speed vehicles which provide maximum packet delivery ratio. The superiority of proposed algorithm is validated by the results obtained by network simulator 2 (ns-2).

Index Terms- Delay tolerant network (DTN), Ferry, Vehicular ad hoc networks (VANETs).

I. Introduction

Ad hoc networks are characterized by infrastructure free, wirelessly connected and distributed systems with no central administration for controlling different operations in network. Every node has capabilities of a router which helps in providing multi hop communication among nodes which don't have direct link. There are constraints in ad hoc networks i.e. limited battery backup, radio range, and heterogeneity of devices [4] [14]. Mobile ad hoc network (MANET) is a sub class of ad hoc networks with mobile nodes. Vehicular ad hoc networks (VANETs) are special class of MANETs which are characterized as, distributed, self-organized networks formed by moving vehicular nodes with no central administration. VANETs are different from MANETs because of high mobility and predictable mobility patterns [13]. Vehicular ad hoc network is becoming a popular area of research where researchers from all over the world are devoting their extensive time to provide safe (providing required information in advance about dangerous situations like accident) and enjoyable (real time game/file sharing between two user in different vehicles) drive over crowded roads. The roads are very crowded now a day due to abrupt increase in number of vehicles in last few years. On congested roads, vehicles speed is low due to large number of vehicles which causes interference in communication among vehicles. Intelligent transportation systems (ITS) [7] is a group which looks after safety of vehicles on roads and traffic management in USA. All the intelligent vehicles can follow the schedule provided by ITS system, which allows vehicles to reach their destination timely. Government security agencies and vehicle owners can track the vehicles if required. ITS uses infrastructure-to-vehicle (I2V) and vehicle-to-vehicle (V2V) communication system. This system also helps in the situations, when an accident occurs on the road, and the vehicles coming in the direction of accidental place should be aware of incident so that vehicles can choose alternate path to avoid congestion on the road.

On highways, driving is not safe because speed limits are hardly followed by drivers. Communication in VANETs is more challenging due to infrastructure free environment and frequent network partitioning caused by high speed of vehicles as compared to MANETs. Fortunately, in VANTEs the nature of motion can be predicted by the roads and digital city maps [8]. The aim of this research is to provide an enhanced routing technique for delay tolerant vehicular ad hoc networks. Delay tolerant networks (DTN) are those networks which do not require immediate data delivery and can wait for a specific time period before the delivery of data. DTN uses the concept of store and forward [11]. The DTN can be considered as overlay network. DTN network uses bundle protocol over IP network [10]. Bundle protocol wraps up data of applications and transfers it as a bundle to lower layers of overlay network. There may be multiple copies of a bundle simultaneously in a DTN network because of store and forward strategy. Bundle can be fragmented by overlay networks if required during transmission [16]. DTN vehicular ad hoc network uses store and forward strategy because of frequent network partitioning due to high speed of vehicles.

II. Related Work

Due to very dynamic nature of VANETs, it is obvious to have frequent network partitioning and link failures. During the design of routing algorithm for VANETs these issues must be kept in mind. VANETs has three modes: (1) Infrastructure, (2) Ad Hoc, (3) Hybrid.

In infrastructure mode, each vehicle can communicate to one another through infrastructure unit only. In ad hoc mode, all vehicles can communicate to each other directly without the need of infrastructure units. Hybrid mode is the combination of infrastructure and ad hoc mode in which vehicles can communicate to each other directly or through infrastructure units as shown in figure 1. The basic protocols used in VANETs are AODV [2], DSR [6], DSDV [3], OLSR [15], and LAR [17] etc. Recent developments in the field of VANETs are more focused towards real time communication and delay tolerant applications.

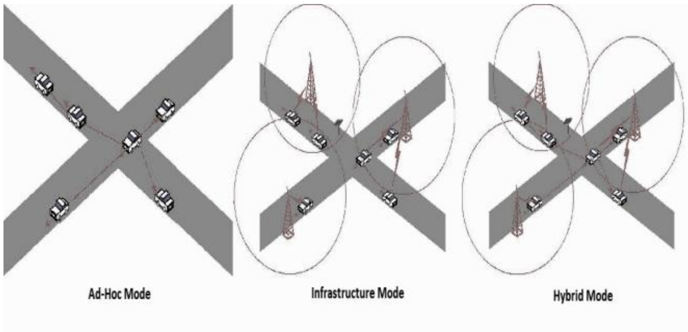

Figure 1: Different modes of VANETs

K. Prasanth et el [9], proposed an algorithm for minimizing end to end delay in VANET by using edge node based greedy routing (EGBR). EBGR can send packets in unicast or broadcast mode. EBGR has three functional units: (1) Neighbor Node Identification (NNI), (2) Node Direction Identification (NDI), and Edge Node Selection (ENS). NNI collects information of all nodes within the range of current node. NDI identifies the direction of movement of nodes moving in direction of destination. ESN selects the edge node within transmission range to forward particular packets. The selection of next node is based on characteristics of nodes and destination of packet. The greedy approach offers minimum delay as compared to simple selection mechanism.

Repository Based Forwarding Protocol (RFP) [1] is another approach in which data is stored at fixed nodes, known as repositories. If a node is moving towards destination, data stored in repository is transferred to that node. This node can further store data in another repository if destination is not reached. If destination node is found, it transfers the data directly. Repositories are installed at most visited places.

Zhi Li et al [18] describe the practical location based routing protocol for VANETs in real scenario. It uses a location predictor tool which can predict the location of a vehicle by using previous history of location and velocity. After getting predicted location, the current node uses greedy forwarding approach to minimize delay. If the mobility is random, the next node also uses predictor tool to get updated location and it further uses greedy forwarding.

Danlei Yu et al [5] proposed fastest ferry based routing algorithm in DNT-enabled VANETs (FFRDV) on highways. FFRDV divides roads into logical blocks and initial ferry chooses the next ferry at the starting of each block by matching its speed with other ferries in block and forwards the bundle to fastest ferry. If initial ferry doesn't find any other ferry with higher speed, it carries the bundle till next logical block, this process continues till the destination node is reached.

The problem in FFRDV algorithm arises when the speed limit on highways is not followed by drivers, and speed difference between current ferry (CF) and designated ferry (DF) is large enough to cause the quick disconnection. When the CF transmits the bundle by analyzing speed of ferry in block, as connection can not be maintained for long time due to high speed of DF, the bundle gets dropped. This makes FFRDV inefficient in high speed scenarios. This paper presents an enhanced ERDV (**E**nhanced **R**outing in **D**elay-tolerant enabled **V**ANETs) algorithm which can handle the high speed vehicles and utilize them to provide maximum packet delivery ratio and ensure minimum end to end delay.

III. Proposed Scheme

In the proposed scheme (ERDV), we are considering that each vehicle is equipped with Global Positioning System (GPS) and is able to get the information about its current location. The geographic information is used to generate logical blocks which divide the road into sections. These blocks are of variable size based on speed of vehicle. The size of block is inversely proportional to speed of vehicle. Each vehicle has its own logical block based on its velocity. Every vehicle broadcasts HELLO message every time it enters its own block. Each HELLO message has the information about, speed and direction of vehicle which has generated it. If there is a vehicle with high speed, it will broadcast HELLO message very frequently because increase in the speed of vehicle will decreases its block size. This helps in detecting the high speed vehicle quickly, rather than detecting in next block as in case of same block size [5]. In ERDV we have taken three steps of four cars with different speeds. To distinguish among cars of different speeds, these are numbered. The car with highest speed is numbered as car1 and slowest as car4. Similarly blocks of each car are also numbered with number of car. Each car follows the respective lane (numbered with car number). As there are four different cars, there will be four different blocks numbered as 1 to 4 for respective car. Car1 broadcasts HELLO messages while entering block 1, car2 while entering block2 and so on.

In first step, whenever an event (accident, query related to destination or road map etc.) occurs, any nearby ferry becomes CF and carries the message. As discussed in figure 2, car4 becomes CF and carries the message. Blocks have been shown as marked line in figure 2.

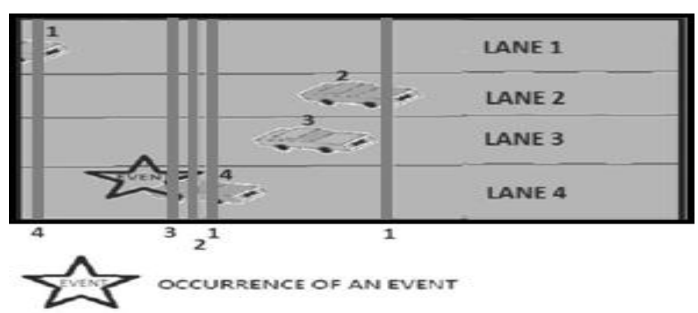

Figure 2: Ferry becomes current ferry by sensing an event.

This CF (car4) will carry the data (in case of light load there are 100 packets to be sent as a bundle and 500 packets in case of heavy load) until it finds a ferry with greater speed. The blocks for specific car are numbered with respective numbers of cars. In second step, every vehicle sends HELLO packet at the starting of its block as shown in figure 3. Here car1 and car3 are entering their respective blocks. In such case CF will receive two HELLO packets broadcasted by car1 and car3. HELLO packet contains the information about velocity and direction of vehicle.

Figure 3: Ferries send HELLO messages at the starting of their blocks.

In third step, CF compares its velocity with velocity in received HELLO packet and decides to send bundle. CF selects a ferry between cars if the velocity in any of received HELLO packets is greater than the velocity of CF and the direction is same as of CF. Here CF (car4) will compare the velocity of car1 and car3 and selects the ferry with highest speed among them which is car1 in this case. The ferry with greater velocity is called designated ferry (DF) and all ferries with lower speed than DF but greater than CF are called candidate ferry (CdF). In given scenario, DF is car1 and CdF is car3. After selecting DF (car1) the CF (car4) will send the data to DF as shown in figure 4.

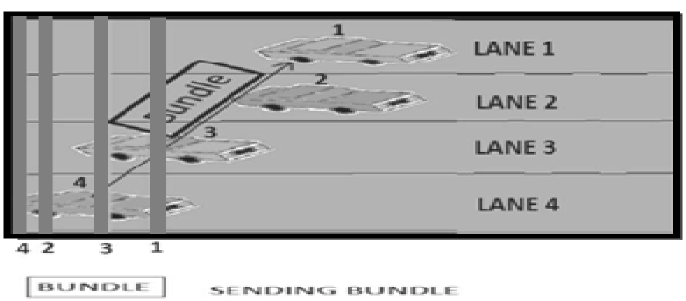

Figure 4: Sending Bundle to Designated Ferry.

A ferry is called CF till it is carrying bundle. When the bundle is transmitted and acknowledged the CF discards stored bundles and becomes normal ferry. In this scenario, there is no boundary of blocks for receiving HELLO packets, but the block boundaries are strictly followed while broadcasting HELLO messages.

The above proposed steps can be implemented using the following algorithm:

The algorithm has four steps:

3.1. Logical blick generation.
3.2. HELLO message broadcasting.
3.3. Selection of DF
3.4. Message transfer

The common terms used in algorithm are:

- BS and blocksize are float type variables which holds the value of block size.
- velocity is the current velocity of vehicle.
- blocksize is a float type variable which holds the calculated value of block size.

3.1. Logical block generation

In this mode, every vehicle generates the block size depending upon its current speed. These blocks are variable depending upon the velocity. At the starting of each block, HELLO messages are broadcasted by every vehicle in its respective block. The block size is inversely proportional to velocity. Block is calculated in this mode with the help of method shown below:

```
BS=BLOCK_SIZE (velocity);

// function for calculating block size.
float BLOCK_SIZE (int velocity)
{
   float blocksize;
   int k;              // in our case k=6.
   blocksize = (k/velocity)*1000;  //meter
   return blocksize;
}
```

"k" is the constant of relationship between velocity and distance, for example to make a block size of 100 meters for 60 km/h speed k will be 6.

This step updates the block size whenever there is a change in speed of vehicle.

3.2. HELLO message broadcasting

In this step each ferry broadcasts HELLO message at the starting of each variable block (which is calculated in step 1). HELLO message contains the information about vehicles current velocity and direction of vehicle.

TIME_HELLO_BROADCAST is float type variable which holds the value of time to broadcast the HELLO packet.

```
// broadcasting the HELLO message.
BROADCAST (BS, velocity);
void BROADCAST (float blocksize, float velocity)
{
   float TIME_HELLO_BROADCAST;
   TIME_HELLO_BROADCAST= (blocksize/velocity)*3.6;
  //seconds
   while (true)
   {
      broadcast_HELLO ();
      WAIT (TIME_HELLO_BROADCAST);
   }
}
```

3.3. Selection of DF

As soon as CF receives the broadcast HELLO message it starts comparing its speed with speed in received HELLO massages from different ferries.

- newpacket is an integer variable which holds the value of number of HELLO packets received.
- CF_SPEED is the speed of current ferry.
- HELLO_SPEED[i] is the speed in ith HELLO packet.
- N is total number of ferries.

```
// store all ferries with greater speed as CdF
int i=1, CdF[N];

while(newpacket!=0)
{
   if( CF_SPEED< HELLO_SPEED[i])
   {
      CdF[N-i]=HELLO_SPEED[i];
   }
   i--;
}

// selecting fastest ferry among CdFs
float MAX=CdF[1]

for(int j=2; j<=newpacket; j++)
{
   if(CdF[i]>MAX)
   {
      MAX=CdF[i];
   }
}
```

3.4. Message transfer

In message transfer mode, ferry sends the bundle to selected DF. If the selected DF rejects to receive the bundle, CF does not need to repeat the process for selecting fastest ferry. In that case CF selects fastest ferry from CdF list. If CdF list is empty, CF needs to repeat the process for selecting DF.

```
// sending bundles to selected DF.
for(N>0)
{
   if(DF -> ACK)
   {
      sendbundle ();
   }
   else if (CdF[0]!=NULL)
   {
      DF=CdF[N];
      sendbundle();
      N=N-1;
   }
   else
   {
      exit();
   }
}
```

The best feature of this algorithm is that it provides minimum end-to-end delay by providing extra time to send bundle rather than selecting ferry. In case of large speed difference between CF and DF, there is very less time to send bundle. The problem in FFRDV is that it checks for fastest ferry in the starting of fixed sized blocks, in mean time the fastest ferry covers large distance and the precious time is lost. But in our proposed solution the fastest ferry broadcast HELLO messages very frequently as its speed increases. This allows CF to detect the fastest ferry as soon as HELLO packets are received and sends data immediately to selected fastest ferry.

IV. Performance Evaluation

For the performance evaluation of ERDV algorithm ns-2 [12] is used. The simulation topography is set to 2500 × 350 meters area. The 2.4 GHz IEEE 802.11 model is used as radio model. Packet size is set to 512kB and transmission range to 250 meters. Traffic is generated for two different scenarios,

Light load ⟶ 100 packets
Heavy load ⟶ 500 packets

The simulation is carried out for different speed differences i.e. speed difference between CF and DF. The results are compared with FFRDV [5] and Distance Aware Epidemic Routing (DAER) [13] with flooding only scheme. In FFRDV algorithm the fastest ferry is selected within the blocks, and ferry sends bundles to DF if fastest ferry is found in the block. In DAER the message is forwarded greedily based on distance of next node from destination node. In ERDV algorithm CF forwards bundle to DF, based on its speed and direction. These schemes are tested for packet delivery ratio and end to end delay vs. speed difference between CF and DF.

a) Packet delivery ratio

Packet delivery ratio is the ratio of "number of packets received at destination end (DF in current case)" to "total number of packets sent by sender (CF in current case)".

Packet delivery ratio =
(Total number of packets received / Total number of packets sent);

In figure 5, the packet delivery ratio is plotted against speed difference between CF and DF for light traffic conditions. As DAER is used with flooding only scheme, its delivery ratio is up to 92% for speed difference of 20 km/h. But as the speed difference increases the delivery ratio decreases due to its fixed flooding time period. When the speed difference between CF and DF reaches 100 km/h the packet delivery ratio decreases up to 11%. At the speed difference of 120 or above the packet delivery ratio is less than 1%. This is because of less connection duration, due to high speed difference. DAER with flooding only scheme also have overheads of duplication of packets in network which causes congestion in network. FFRDV on the other hand provides delivery ratio is up to 91% for speed difference of 20km/h and, but as speed difference increases its delivery ratio decreases because of its fixed block size scheme. For speed difference of 100 km/h the delivery ratio reduces up to 22%. At the speed difference of 120 or above the packet delivery ratio is less than 5%. This is because of less time available to send the bundle, due to fixed block size. ERDV performs well up to large speed difference. Its delivery ratio is up to 95% when speed difference is 20km/h. When the speed difference between CF and DF reaches 100 km/h the packet delivery ratio remains up to 52%. At the speed difference of 120 the packet delivery ratio is up to 30%. This shows that the ERDV algorithm performs well even when the speed difference between CF and DF is high.

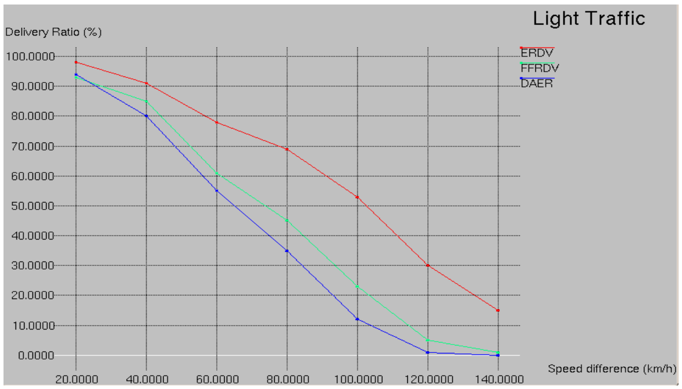

Figure 5: packet delivery ratio (%) vs. speed difference (Km/h) under light traffic load

Figure 6 shows the packet delivery ratio vs. speed difference for heavy traffic load. For heavy traffic load, the delivery ratio of DAER with flooding only scheme degraded because of lesser connectivity time as compared to time required to transfer data.

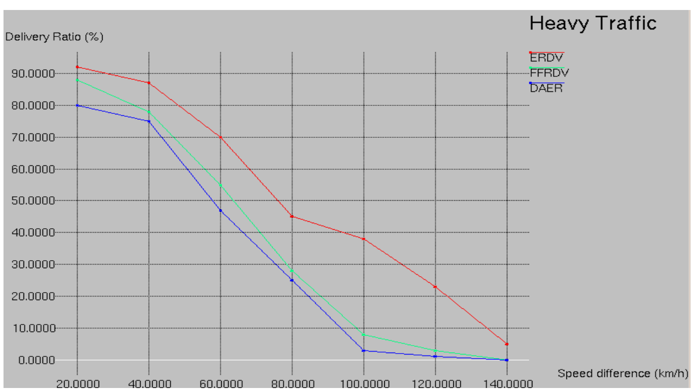

Figure 6: packet delivery ratio (%) vs. speed difference (Km/h) under heavy traffic load

At the speed difference of 120 the packet delivery ratio is up to 22%. Both the results show that the ERDV algorithm performs well even when the speed difference between CF and DF is high for both light traffic and heavy traffic conditions.

b) End to end delay

End to end delay is the time taken by a packet to reach the destination from sender,

$$D_{end\text{-}to\text{-}end} = \sum_{i=1}^{n} [D_{queue}(i) + D_{trans}(i) + D_{prop}(i) + D_{proc}(i)]$$

- $D_{end\text{-}to\text{-}end}$: total end to end delay
- **n** is total number of vehicles involved in carrying the bundle end to end.
- **D_{queue}(queuing delay):** is the amount of time packet stays in queue. In case of DTNs it is also called storage time (duration of time for which data is being stored by vehicles).
- **D_{trans} (transmission delay):** is the amount of time required to push all of the bits of a packet into the medium. In other words, this is the delay caused by the data-rate of the link.
- **D_{prop} (propagation delay):** is the amount of time taken by a packet to travel from the sender to the receiver over a medium. It is the ratio of link length and propagation speed over the specific medium.
- **D_{proc} (processing delay):** is the time it takes ferries to process the packet header.

Figure 7 shows increase in end to end delay as the speed in DAER flooding only scheme decreases due to availability of high speed ferry. At the speed difference of 20km/h, DAER with flooding only provides end to end delay of ~169 seconds.

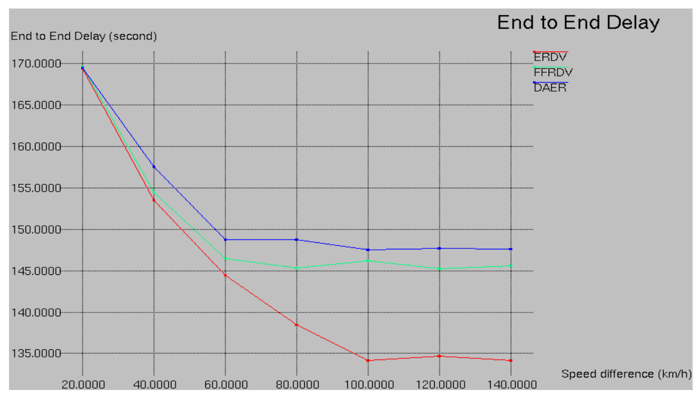

Figure 7: end to end delay (seconds) vs. speed difference (Km/h)

The end to end delay at the speed of 60km/h is up to 148 seconds and it remains up to 142 seconds for further increase in speed difference. This happens because DAER is not able to detect high speed cars quickly. The speeds of these high speed cars are not utilized by DAER flooding only mechanism.

FFRDV on the other hand provides end to end delay of 169 seconds at the speed difference of 20km/h and 145 seconds at speed difference of 60km/h. The end to end delay remains up to 145 seconds for further increase in speed difference. ERDV algorithm also provides the end to end delay of ~169 seconds at the speed difference of 20km/h. At speed difference of 60km/h the ERDV algorithm provides delay of less than 145 seconds. At the speed difference of 100km/h the delay is up to 135 seconds. This is because of availability of extra time to send bundle to DF. The above results show that as the speed difference increases the performance of FFRDV and DAER decreases but the performance of ERDV performs well up to large range, due to its dynamic HELLO broadcasting mechanism.

V. Conclusion and Future Work

The fastest ferry based algorithm for DTN-enabled VANETs is proposed in this paper and the performance is compared with other DTN routing algorithms. The selection of ferry is dynamic, every ferry broadcasts HELLO message in its logical blocks which are dynamic in nature. The proposed algorithm performs well and provides good packet delivery ratio, as speed difference increases, as compared to "FFRDV" and "DAER with flooding only" algorithms under light traffic load as well as heavy traffic load. Results obtained from ns-2 shows that proposed algorithm is superior to both "FFRDV" and "DAER with flooding only scheme" in case of high speed difference between CF and DF as the delivery ratio is higher than other two schemes. In future, this algorithm needs to be tested for end-to-end delay and delivery ratio, for different number of nodes and traffic conditions.

References

[1] Abdolbast Greede. (2009). RFP: Repository Based Forwarding Protocol for Opportunistic Networks. *Proc. of NGMAST*, 2009, (pp. 329-334).

[2] C. E. Perkins, E. Belding-Royer, Santa Barbara, S. Das. (2003). Ad hoc On-Demand Distance Vector (AODV) Routing. Experimental. RFC: 3561.

[3] C. E. Perkins and P. Bhagwat. (1994). High Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers. *Proceedings of ACM SIG-COMM'94, (*pp. 234-244).

[4] Daming Feng. (2009). An Improved AODV Routing Protocol Based on Remaining Power and Frame. *Proc. of ICECT'09, (*pp. 117-121).

[5] Danlei Yu and Young-Bae Ko. (2008). FFRDV: Fastest-Ferry Routing in DTN-enabled Vehicular Ad Hoc Networks. *Proc. ICACT'09*, (pp. 1410-1414).

[6] D. Johnson. (2007). The Dynamic Source for Mobile Ad Hoc Networks for IPv4. Experimental. RFC: 4728.

[7] Intelligent Transportation System, http://www.its.dot.gov/research/v2i.htm last accessed on 10 February 2012.

[8] Jose Santa. (2009). Experimental Analysis of Multi-hop Routing in Vehicular Ad-hoc Networks. *Proc. of TRIDENTCOM'*09, (pp. 1-8).

[9] K.Prasanth. (2009). Minimizing End-to-End Delay in Vehicular Ad Hoc Network using Edge Node Based Greedy Routing. *Proc. of ICAC'*09, (pp. 135-140).

[10] K. Scott. (2007). Bundle Protocol Specification. Experimental. RFC: 5050.

[11] L. Franck. (2007). Using Delay Tolerant Networks for Car2Car Communication. *Proc. of ISIE2007, (*pp. 2573 – 2578).

[12] NS-2 available at, http://nsnam.isi.edu/nsnam/index.php/User_Information last accessed on 20 February 2012.

[13] P. Luo, H. Huang. (2008). Performance Evaluation of Vehicular DTN Routing under Realistic Mobility Models. *Proc. of Wireless Communications and Networking Conference, 2008 IEEE Communication Society*, (pp. 2206 – 2211).

[14] S. Harous. (2008). An Energy Aware Multi-Path Routing Algorithm For Mobile Ad Hoc Networks. *International Journal of Business Data Communications and Networking, IGI publishing,* 4(2), 58-75.

[15] T. Clausen, Ed. P. Jacquet, Ed. (2003). Optimized Link State Routing Protocol (OLSR). RFC: 3626.

[16] V. Cerf, S. Burleigh, A. Hooke, and L. Torgerson. (2007). Delay-Tolerant Networking Architecture. RFC: 4838.

[17] Young-Bae Ko and Nitin H. Vaidya. (2000). Location-Aided Routing (LAR) in mobile ad hoc networks. *Journal Wireless Networks, Kluwer Academic Publishers Hingham,* 6(4), 307–321.

[18] Zhi Li. (2009). Practical Location-based Routing in Vehicular Ad Hoc Networks. *Proc. of Mobile Ad-hoc and Sensor Systems, 2009, IEEE 6th International Conference,* (pp. 900-905).

Authors

First Author – Arun Kumar, M.Tech (Wireless Communication & Networks), B.Tech (Electronics & Instrumentation), Software Engineer at Syscom Corporation Ltd, email: arunkumar7620@gmail.com, arunkumar.gbu@gmail.com.