Journal of Network and Innovative Computing
ISSN 2160-2174 Volume 1 (2013) pp. 43- 54

Congestion Control and Revocation of Misbehaving Vehicles in VANET

Sulata Mitra[1], Trishita Ghosh[2]

[1]

Department of Computer Science and Technology
Bengal Engineering and Science University
Shibpur, West Bengal, India
sulata@cs.becs.ac.in

[2] Department of Information Technology
Gurunanak Institute of Technology
Sodepur, West Bengal, India
trish2921@gmail.com

***Abstract*: The wireless access in vehicular environment system is developed for enhancing the driving safety and comfort of automotive users. However, such system suffers from quality of service degradation for safety applications caused by the channel congestion in scenarios with high vehicle density. The present work is a congestion control mechanism in vehicular ad hoc network. It also revokes the misbehaving vehicles from the network. It supports vehicle to vehicle and vehicle to infrastructure communication of safe messages using control channel and unsafe messages using service channel. The proposed scheme reduces the channel congestion by allowing only the authentic vehicles to participate in vehicle to vehicle and vehicle to infrastructure communication. Each vehicle generates beacon message periodically for its neighbors. The size of the beacon message is controlled dynamically to minimize channel congestion. The duplicate message truncation algorithm at each vehicle also reduces channel congestion by discarding the duplicate safe messages. The proposed scheme also reduces the loss and delay in transmission of safe and unsafe messages by sharing the available bandwidth of the channels among vehicles efficiently. The unsafe messages at a vehicle may also be transmitted using control channel provided the control channel is free and service channel is overloaded which helps to reduce the loss of unsafe message at a vehicle which in turn reduces the congestion of a vehicle and also improves its quality of service. The performance of the proposed scheme is evaluated on the basis of average loss of unsafe message, average delay in safe and unsafe message, storage overhead per vehicle**.

***Keywords*:** V2V communication; V2I communication; Congestion control; Safe and unsafe message.

I. Introduction

Congestion control is an important research issue to ensure safe and reliable vehicle to vehicle (V2V) communication by using the limited resource available in vehicular ad-hoc network (VANET). Each vehicle in VANET is able to transmit its own message and can receive messages from other vehicles.

Several congestion control schemes have been proposed so far. Zhang et al. propose congestion control in wireless networks for vehicular safety applications [1]. The event driven congestion control is triggered reactively whenever a high priority safety message arrives to the system. It helps to guarantee quality of service (QoS) for safety applications. The measurement based congestion detection measures the channel usage and compares it with a defined threshold. But the effective transmission of the safety messages is not guaranteed as the authors did not consider the neighborhood context and the effective bandwidth sharing. Moreover the delay in event driven safety message is found experimentally as 50 msec. But for safety application such message need to disseminate among adjacent vehicles within 20 msec.

A scalable congestion control for event driven safety messages in VANET is proposed in [2]. The congestion is detected by scanning the message in the queue and by monitoring the channel communication based on predefined threshold. A priority based earliest deadline first algorithm is used to schedule the packets having the same high priority for communication using control channel. But they consider only the event driven safety messages having identical priorities.

A cooperative congestion control approach is proposed in [3]. The priority of the messages is determined dynamically depending upon their types, network context and the neighboring vehicles configuration. In [3] the cooperation in message transmission is achieved by dynamically distributing the available bandwidth among vehicles. A vehicle having the high priority message can use the available bandwidth only. If the two vehicles have to send two messages with the same priority, the available bandwidth is given to the first vehicle who notifies the priority of its message. Each vehicle specifies

the time of first notification in its beacon message. But in [3] the available bandwidth at each vehicle is independent on the congestion level of vehicles.

A congestion control approach is proposed in [4]. It considers dynamic priority based scheduling. The service messages are scheduled in control channel if it is free. But the authors have not suggested the message format. Moreover the performance analysis is not presented graphically in [4].

A cooperative scheme for service channel (SCH) reservation is proposed in [5]. According to Wireless Access Vehicular Environment standard, the safety messages are carried over a dedicated control channel (CCH) while non-safety messages are delivered over one of a set of available service channel (SCH) [5]. But in [5] there is a possibility of bandwidth wastage when the SCH is overloaded and CCH is idle.

An adaptive congestion control for dedicated short range communication vehicle networks is proposed in [6]. In this scheme each device periodically senses the channel based on the channel occupancy time. If the channel occupancy time measured at a vehicle in CCH exceeds the predefined threshold, all beacon messages will be blocked immediately.

A novel concept for utility based congestion control and packet forwarding in VANET is proposed in [7]. The congestion control algorithm uses an application specific utility function and encodes the quantitative utility information of each transmitted data packet in a transparent way for all users within a local environment. A decentralized algorithm then calculates the average utility value of each individual vehicle based on the utility of its data packets and assigns a share of the available data rate proportional to the relative priority. The evaluation of message priority based on utility and packet size reduces the performance of disseminating event driven safety messages.

A congestion control for safety messages in VANET is proposed in [8]. The algorithm operates only vehicles on the same lane. The vehicles forward the event driven safety messages after receiving it successfully from front vehicles. But in real scenarios an accident in a lane also affects other lanes.

A congestion control algorithm is proposed in [9, 10] to ensure high reliability and timely delivery of disseminating event-driven safety messages. It is the combination of event-driven detection and measurement-based detection. The measurement-based congestion detection monitors CCH. The CCH is congested if the number of packets in the control queue exceeds a defined threshold and the algorithm starts to discard the incoming beacon messages. The event-driven detection method monitors the event-driven safety message and decides to start the congestion control algorithm whenever event-driven safety message is detected or generated. But it considers only the event driven safety messages. Moreover a node may not have the up to date information about its neighbors due to the loss of beacon messages in case of CCH congestion.

The authors in [11] propose a new congestion control approach for beacon transmission. A distributed beacon frequency control for VANETs is proposed for the beacon safety message transmission. It mitigates the terrible channel condition in dense network by adaptively adjusting the transmission frequency for beacons according to the network context, while considering the accuracy of updating status information.

The objective of the proposed scheme is to minimize the channel congestion and to revoke misbehaving vehicles from VANET. The scheme in [12] is extended in the present work for further reduction of channel congestion. The VANET in the proposed scheme is a hierarchy having certifying authority (CA) at the root level, road side units (RSUs) at the intermediate level and vehicles at the leaf level. Each vehicle has an electronic license plate (ELP) in which the encrypted vehicle identification number (VIN) of the vehicle is embedded by the vehicle manufacturer. Each vehicle is equipped with global positioning system to know its current location.

The present work supports V2V communication of safe and unsafe messages among authentic vehicles. It also supports vehicle to infrastructure (V2I) communication of unsafe message among authentic vehicles and RSU. The priority of safe messages is assumed as higher than the priority of unsafe messages to disseminate the safe messages among vehicles without delay.

The ELP of a vehicle broadcasts (as per IEEE P1609 and IEEE 802.11p) the encrypted VIN after entering into the coverage area of a new RSU. The new RSU verifies the authentication of the vehicle. It assigns a digital signature to the vehicle as a valid key if the vehicle is authentic. Each authentic vehicle includes its digital signature in the message format which helps to prevent the unauthentic vehicle from participating in V2V and V2I communication. Each RSU revokes the misbehaving vehicles from its coverage area without which antisocial and criminal behavior jeopardizes the benefit of the system deployment.

A control queue (CQ) is maintained for keeping safe messages and a service queue (SQ) is maintained for keeping unsafe messages at each vehicle. The length of CQ and SQ at each vehicle is assumed as variable and it depends upon the number of safe and unsafe messages. The duplicate messages are discarded from CQ to minimize channel congestion. The congestion level of a vehicle is measured in terms of its priority. The priority of a vehicle is computed as a function of the number of high priority and low priority messages in its CQ and SQ respectively. Each vehicle sends variable length beacon message to its neighboring vehicles periodically and specifies its own priority in the beacon message. The effective bandwidth of each vehicle is computed dynamically depending upon its priority and the priority of its neighboring vehicles to minimize the average loss of safe and unsafe messages.

In the proposed scheme it is assumed that all the vehicles enter into VANET at the same instant of time. The channel time is divided into synchronization periods of 100 ms [13]. Each synchronization period consists of equal-length (τ) alternating CCH and SCH interval [13]. At the beginning of each interval a 4 ms long guard time (τ_g) is set to support radio, switching delay and timing inaccuracies. The CCH interval and SCH interval are shared among vehicles efficiently for the transmission of safe and unsafe messages respectively for reducing channel congestion. Both the CCH and SCH interval are divided into variable size slots. The number of slots in each interval depends upon the number of

vehicles in VANET. Each vehicle reserves a slot both in CCH and SCH interval depending upon its priority w.r.t. its neighboring vehicles. It also computes the size of the reserved slot both in CCH and SCH interval for the transmission of waiting messages in CQ and SQ respectively. Each vehicle also transfers messages from SQ to CQ in case the SCH is overloaded and CCH is free to reduce the loss of unsafe message which helps to maintain the quality of service of a vehicle at a satisfactory level.

The proposed scheme supports V2V and V2I communication of messages among authentic vehicles only. The dynamic selection of beacon message size and the removal of duplicate message help to reduce the channel congestion. Moreover the available bandwidth is shared dynamically among vehicles which help to utilize the available resource of VANET in an efficient way. It verifies the authentication of vehicles using VIN. The use of VIN for authentication of a vehicle is advantageous as it is impossible to transfer VIN among vehicles and to alter the information on it. Moreover VIN of a vehicle remains intact even in typical environmental condition. It contains information about the manufacturer of the vehicle and description of the vehicle. So other than identification and authentication VIN can also be used to know the manufacturing details and the details description of a vehicle which may require in case of accidents etc. The proposed scheme is also able to trace misbehaving vehicles and revokes them from the network in a timely fashion to prevent the misbehaving vehicles from causing more damage to the network.

The rest of the paper is organized as follows. The section II discusses the present scheme. The experimental results are elaborated in section III. Section IV concludes the paper.

II. Present Work

In this section the type of messages supported by the proposed scheme and their formats are considered for discussion. The function of CA and RSU are elaborated. The function of the i^{th} vehicle (N_i, $1 \le i \le N$, N-1 is the number of neighboring vehicles of N_i) is also discussed.

A. Message Format

The proposed scheme supports V2V communication of safe messages such as emergency message (Type_I message), warning message (Type_II message) and beacon message (Type_III message). It also supports V2V communication of unsafe message such as query for route to reach to a specific destination or query for nearest restaurant etc. (Type_IV message) and V2I communication of unsafe message such as query for traffic condition of a particular road (Type_V message).

The format of Type_I message is shown in Fig1(i). Such message is generated by an abnormal vehicle (AV). AV is a vehicle which behaves abnormally like declaration exceeds a certain threshold, dramatic change of moving direction, major mechanical failure etc. Here it is assumed that the computation power of AV is not affected and its speed is almost zero. It sends emergency message to all its neighboring vehicles. AV can calculate which vehicles are moving towards its location and how much time they need to reach this location from the beacon messages of its neighboring vehicles. The time out of emergency message is calculated by AV as (Distance of the nearest neighbor from AV)/(2*Speed of the nearest neighbor) so that the nearest neighbor of AV can receive this message in time. Such message needs at least two hop communications for informing the surrounding vehicles about the location of AV.

The format of Type_II message is shown in Fig1(ii). Such message is generated by a vehicle (Source_V) which detects traffic signal breakdown, jamming information etc. The Source_V sends this message to the vehicles behind it i.e. to the vehicles which are moving towards the jam location. The Source_V determines its distance from the nearest neighbor (Des_V) behind it and the speed of Des_V from its beacon message. The time out of warning message is calculated by Source_V as (Distance between Source_V and Des_V)/(2*Speed of Des_V) so that Des_V can receive this message in time. It is required to prevent the other vehicles from entering into this road for improving its traffic condition. So such message needs the communication up to the end of the road whose identification (Road_id) is specified in the message format.

The format of Type_III message is shown in Fig.1(iii). Such message is generated by all the vehicles periodically. The time out of such message is assumed as the reciprocal of beacon generation rate which is assumed as 5 to 10 Hz [10] otherwise the current beacon becomes obsolete by a freshly generated beacon.

The format of Type_IV message is shown in Fig1(iv). It is a query message which is generated by a vehicle (Source_V) to know the nearest restaurant (Destination_point) or the route to reach to a specific destination (Destination_point) etc. The Source_V sends this message to its neighboring vehicles and expecting its reply before reaching to the point (R_point) from where it has to enter into a new road so that the Source_V can move towards the Destination_point using proper route. The time out of this message is calculated by the Source_V as (Distance between the Source_V and R_point)/Speed of Source_V.

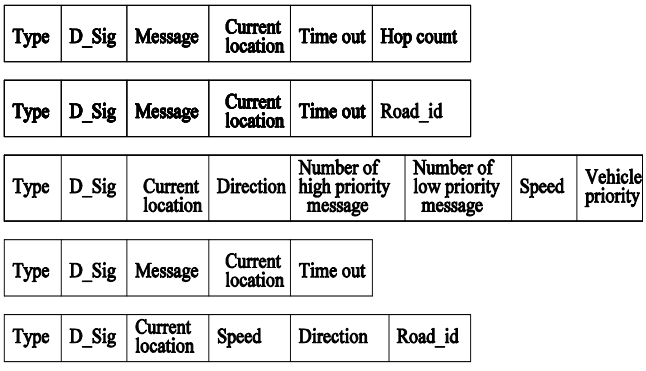

Fig.1 Format of (i) Type_I message (ii)Type_II message (iii)Type_III message (iv) Type_IV message (v) Type_V message

The format of Type_V message is shown in Fig1(v). It is a query message which is generated by a vehicle (Source_V) to know the traffic condition of a particular road (D_Road). The identification of D_Road is mentioned in the Road_id field of the message. The Source_V sends this message to its nearest RSU (N_RSU). N_RSU considers the time out of this message as the time during which the Source_V resides within its coverage area. It computes this time out by using the

current location, direction and speed of Source_V as mentioned in the message format. N_RSU processes the query by sending the query and its time out to the RSU associated with D_Road (D_Road_RSU). N_RSU delivers the answer of the query to the Source_V after receiving its reply from D_Road_RSU.

The format of message field in Type_I and Type_II message is shown in Fig.2.

EM_Event	D_Sig	Region

Fig. 2(i) Format of Message field in Type_I message

WM_Event	D_Sig	Region

Fig. 2(ii) Format of Message field in Type_II message

EM_EVENT field contains the event for which the emergency message is created. For example, in case of crash the EM_EVENT is "Pre-crash sensing", in case of major mechanical failure the EM_EVENT is "Major mechanical failure" etc.

WM_EVENT field contains the event for which the warning message is created. For example, in case of traffic jam the WM_EVENT is "Traffic jam", in case of traffic signal breakdown WM_EVENT is "Traffic signal breakdown" etc.

The EM_EVENT and WM_EVENT field will be same for every vehicle facing same emergency and warning event respectively. A few example of such phrase is shown in Table 1.

Situation/ Purpose	Phrase used in WM_Event/EM_Event
Dangerous road features	"Curve speed warning", "Low bridge warning", "Warning about violated traffic lights or stop signals"
Danger of collision	"Lane change warning", "Intersection collision warning", "Forward/rear collision warning", "Emergency electronic brake lights", "Rail collision warning", "Warning about pedestrian crossing"
Crash imminent	"Pre-crash sensing"
Incident occurred	"Post-crash sensing", "Traffic signal breakdown", "Traffic jam"

Table 1. Phrase of EM_EVENT and WM_EVENT"

The D_Sig field indicates the digital signature of the vehicle that is first facing the event and creating the message.

During simulation each road is divided into few regions so that each vehicle can determine the location of occurrence of an event in a road more accurately. Region field contains the number of the region where the event has taken place. The number of region in a road depends upon its length.

As the proposed scheme considers 5 different types of messages, the size of type field is 3 bits. The D_Sig field in the message format is used to identify the owner of the message and the size of this field is 160 bits (as discussed in section B.1). Each vehicle determines its location in terms of latitude and longitude. The size of current location field in the message format is assumed as 32 bits. As a vehicle can move in one of four directions (north, south, east, west), the size of direction field in the message format is 2 bits. The maximum speed of a vehicle is assumed as 120 km/hr and the size of speed field in the message format is 7 bits. The size of time out field in the message format is assumed as 32 bits. The size of message field in Type_I and Type_II type of message is $192+\log_2 R$ bits, where the value of R depends upon the number of regions in a road. The size of message field in Type_IV type of message is assumed as 32 bits. The size of vehicle priority field is assumed as 32 bits. In case of Type_I message the maximum hop count is 2 and so the size of hop count field in the Type_I message format is 2 bits. The proposed scheme considers 10 roads in the VANET environment during simulation. So the size of Road_id field in the message format is 4 bits. Hence the size of Type_I (Size_I) message is $421+\log_2 R$ bits, Type_II message (Size_II) is $423+\log_2 R$ bits, Type_III message (Size_III) is $236+\log_2$(Size of CQ + Size of SQ) bits, Type_IV message (Size_IV) is 259 bits and Type_V message (Size_V) is 210 bits.

B. Function of CA and RSU

The CA maintains a VIN database (VIN_CA) to store the VINs of the vehicles that are already manufactured. The VIN_CA is updated when a new vehicle is manufactured. The CA gathers knowledge about the available pattern of each character in VIN of the vehicles which are already manufactured by consulting with VIN_CA after updating it. Each RSU maintains a VIN database (VIN_RSU) to store the encrypted VIN and digital signature pair of all the authentic vehicles in VANET. Each RSU also maintains a revocation list to store the digital signature of the misbehaving vehicles.

Each RSU has an authentication module to verify the authentication of a vehicle within its coverage area and to assign a digital signature to each authentic vehicle. The authentication module is the combination of authentication function at RSU, verification function at CA, insertion function at RSU and signature assignment function at RSU. The message processing module at RSU receives messages from the vehicles which are within its coverage area, verifies them for validity and allows them to process if valid.

1) Function of authentication module at RSU:

The ELP of a vehicle sends its encrypted VIN after entering into the coverage area of new RSU (New_RSU). The authentication function at New_RSU adds the encrypted VIN in a VIN queue and searches the records in VIN database using VIN_SEARCH algorithm. If found the vehicle is authentic and so its authentication phase is over. The authentication function at New_RSU reads the digital signature from the record in VIN database corresponding to the encrypted VIN and triggers the signature assignment function by sending the digital signature. The signature assignment function at New_RSU assigns the digital signature to the vehicle.

Otherwise the authentication function at New_RSU initiates the authentication phase of the vehicle and calls the verification function [14] at CA by sending the encrypted VIN of the vehicle. The verification function at CA adds the

encrypted VIN in its VIN queue and decrypts the encrypted VIN using RSA algorithm [15]. The CA knows the possible pattern of each character in decrypted VIN [16]. It also has knowledge about the available pattern of each character in the decrypted VIN from VIN_CA. So it verifies each character in field_1, field_2 and field_3 of the decrypted VIN [17] for validity using VALID function. If the decrypted VIN is not valid the verification function at CA generates a security message for the police or checking personnel to make them aware about this vehicle.

Otherwise it generates the digital signature from the decrypted VIN of the vehicle using SHA-1 algorithm [15]. The size of digital signature is 160 bits [15]. It calls the insertion function at all RSUs under CA by sending the (encrypted VIN, digital signature) pair. The insertion function at each RSU inserts a record in the form (encrypted VIN, digital signature) in VIN database. During this span of time the vehicle may reside within the coverage area of New_RSU or may move to the coverage area of another RSU under CA. The RSU within which the vehicle is currently passing through is the Current_RSU. The insertion function at Current_RSU triggers the signature assignment function by sending the digital signature. The signature assignment function at Current_RSU assigns the digital signature to the vehicle. So even if the vehicle moves from New_RSU to another RSU during its authentication phase, it will receive its digital signature from the new RSU as all the RSUs under CA have a copy of the same digital signature. Therefore the authentication phase of a vehicle is independent on its velocity which is an obvious requirement in VANET due to its high mobility model.

Authentication function at New_RSU

```
{
 Receives encrypted VIN from vehicle
 Adds encrypted VIN in its VIN queue
 Searches VIN_RSU for the encrypted VIN using
 VIN_SEARCH algorithm
 If found
  {
   Reads digital signature from the record corresponding to
   the encrypted VIN of the vehicle from VIN_RSU
   Calls the signature assignment function by sending the
   digital signature of the vehicle
   Exit
  }
 Else
  {
   Calls the verification function at CA by sending the
   encrypted VIN
   Exit
  }
}
```

Verification function at CA

```
{
 Adds encrypted VIN in its VIN queue
 Decrypts the encrypted VIN
 Verifies each character of the decrypted VIN
  for validity using VALID function
  If decrypted VIN is valid
   {
    Generates digital signature from decrypted VIN
    Calls insertion function at all RSUs under it by
    sending encrypted VIN and digital signature pair
    Exit
   }
  Else
   {
    Generates security message
    Exit
   }
}
```

Insertion function at RSU

```
{
 Inserts encrypted VIN and digital signature pair in
 VIN database
 Insertion function at Current_RSU calls the signature
 assignment function by sending the digital signature
 Exit
}
```

Signature assignment function at RSU

```
{
  Assigns the digital signature to the vehicle
}
```

VIN_SEARCH algorithm at New_RSU

```
{
   Str_R ← Received VIN from vehicle
   Size ← Number of encrypted VIN in
   VIN_RSU
   If (Size=0)
    Str_R is not found in VIN_RSU
   Else
    {
     Length_R ← Length of Str_R
     D ← 1
L1:  Str_D ← D^th encrypted VIN in VIN_RSU
     Length_D ← Length of Str_D
     If (Length_R≠Length_D)
       Go to LOOP1
     Else
      {
       B ← 1
L2:    Str_R(B) ← B^th character of Str_R
       Str_D(B) ← B^th character of Str_D
       If (Str_R(B)≠Str_D(B))
         Go to LOOP1
       Else
        {
         B ← B+1
         If (B≤Length_R)
           Go to L2
         Else
           Str_R is found in VIN_RSU
        }
      }
     }
```

```
LOOP1: {
            D ← D+1
            If (D≤Size)
              Go to L1
            Else
              Str_R is not found in VIN_RSU
          }
```

VALID function at CA

```
k=1
LOOP:
{
 D_VIN_k: k^th character of decrypted VIN
 PP_k: Possible pattern of D_VIN_k as mentioned in [13]
 AV_k: Available pattern of D_VIN_k as gathered from
 VIN_CA
 if (D_VIN_k≠PP_k or D_VIN_k≠AV_k)
  {
    Decrypted VIN is not valid
    Exit
  }
if (D_VIN_k=PP_k and D_VIN_k=AV_k)
 {
   D_VIN_k is valid
   Go to L2
  }
}
L2: {
        k=k+1
        If (k>17)
        {
          Decrypted VIN is valid
          Exit
        }
       Else
         Go to LOOP
    }
```

2) Function of message processing module:

Each RSU maintains the traffic condition of the roads which are within its coverage area. It receives messages from vehicles within its coverage area. The function of this module for j^{th} message (M_j) is elaborated in this section. It verifies the validity of the digital signature in the D_Sig field of M_j (DS_{1j}) for all the five types of messages. It also verifies the validity of the digital signature in the message field of M_j (DS_{2j}) if M_j is Type_I or Type_II type. Moreover it consults with the traffic condition information available with it for verifying the validity of (EM_EVENT, Region) information in M_j if M_j is Type_I type and (WM_EVENT, Region) information in M_j if M_j is Type_II type.

```
ER: (EM_EVENT, Region) in Message field of M_j if M_j is
      Type_I type
WR: (WM_EVENT, Region) in Message field of M_j if M_j is
     Type_II type
If (M_j=Type_I type)
 {
  Searches VIN_RSU for DS_1j and for DS_2j
  If (DS_1j or DS_2j or both are invalid)
     {Inserts invalid digital signature(s) in the revocation list
      Go to L1}
  Else
{
     Verifies ER of M_j for validity
     If valid
       Allows processing of  M_j
     Else
       {Inserts DS_2j of M_j in the revocation list
        Go to L1}
     }
 }
If (M_j=Type_II  type)
 {
  Searches VIN_RSU for DS_1j and for DS_2j
  If (DS_1j OR DS_2j OR both are invalid)
     {Inserts invalid digital signature(s) in the revocation list
      Go to L1}
  Else
     {
      Verifies WR of M_j for validity
      If valid
        Allows processing of M_j
      Else
        {Inserts DS_2j of M_j in the revocation list
         Go to L1}
      }
 }

If ((M_j=Type_III type) OR (M_j = Type_IV type))
  {
   Searches VIN_RSU for DS_1j
   If found
     Allows processing of M_j
   Else
     {Inserts DS_1j in the revocation list
      Go to L1}
  }

If (M_j=Type_V type)
  {
   Searches VIN_RSU for DS_1j
   If found
     Processes M_j
   Else
     {Inserts DS_1j in the revocation list
      Go to L1}
  }

L1: Broadcasts the invalid digital signature(s) among
    vehicles within its coverage area
```

C. Function of N_i

In this section the function of N_i as a combination of message manager module (MMM_i), congestion control module (CCM_i) and bandwidth manager module (BWM_i) is elaborated.

1) Function of MMM_i:

MMM_i generates messages for its neighboring vehicles and receives messages from its neighboring vehicles. It generates variable length beacon message periodically. It stores the safe messages in CQ and unsafe messages in SQ at N_i. It discards the duplicate Type_I and Type_II type of messages from CQ at N_i using duplicate message truncation (DMT) algorithm. MMM_i computes the priority of its associated vehicle

dynamically. It stores its own beacon message and the beacon message of its neighbors in a neighbor table.

(a) Message generation:

MMM_i generates all 5 types of messages at N_i. The Direction field and Speed field are added in Type_III message only if N_i changes its direction and speed of movement. Accordingly Size_III varies dynamically which helps to minimize the channel load due to N_i. It places Type_I, Type_II and Type_III messages in CQ whereas Type_IV and Type_V messages in SQ.

(b) Message Reception:

MMM_i maintains an incoming queue (IQ) to store the received messages from neighboring vehicles of N_i. It transfers the safe messages from IQ to CQ and unsafe messages from IQ to SQ using monitor IQ function.

Monitor IQ function for M_j

M_j: receives j^{th} message in IQ
If (M_j=Type_I message)
{Extracts information from M_j
If (hop count=0)
Discards M_j
Else
Places M_j in CQ}
If (M_j=Type_II message)
{ROAD:= Road_id given in M_j format
Extracts information from M_j
If (N_i=last vehicle of Road)
Discards M_j
Else
Places M_j in CQ}
If (M_j= Type_III message)
Updates the neighbor table
If (M_j= Type_IV message)
Places M_j in SQ

(c) DMT algorithm:

This algorithm discards the duplicate Type_I and Type_II messages from CQ at N_i.

DMT algorithm for Type_I message:

{
M_j: Received message
$MESS_j$: Message field of M_j
E_j: EM_EVENT field in $MESS_j$
D_j: D_Sig field in $MESS_j$
R_j: Region field in $MESS_j$
NO_I: Number of Type_I message in CQ
i=1 //First Type_I message in CQ
LOOP: {
M_i: i^{th} Type_I message in CQ
$MESS_i$: Message field of M_i
E_i: EM_EVENT field in $MESS_i$
D_i: D_Sig field in $MESS_i$
R_i: Region field in $MESS_i$
If ($E_j \neq E_i$)
{Accepts M_j
Exit}
If (($E_j=E_i$) AND ($D_j=D_i$) AND ($R_j \neq R_i$))
{Accepts M_j
Exit}
If (($E_j=E_i$) AND ($D_j=D_i$) AND ($R_j=R_i$))
{Discards M_j
Exit}
If (($E_j=E_i$) AND ($D_j \neq D_i$) AND ($R_j=R_i$))
{Discards M_j
Exit}
If (($E_j=E_i$) AND ($D_j \neq D_i$) AND ($R_j \neq R_i$))
{Accepts M_j
Exit}
i=i+1
If (i≤NO_I)
Go to LOOP
Else
Exit
}
}

DMT algorithm for Type_II message:

{
M_j: Received message
$MESS_j$: Message field in M_j
W_j: WM_EVENT field in $MESS_j$
D_j: D_Sig field in $MESS_j$
R_j: Region field in $MESS_j$
NO_II: Number of Type_II message in CQ
i=1 //First Type_II message in CQ
LOOP: {
M_i: i^{th} Type_II message in CQ
$MESS_i$: Message field in M_i
W_i: WM_EVENT field in $MESS_i$
D_i: D_Sig field in $MESS_i$
R_i: Region field in $MESS_i$
If ($W_j \neq W_i$)
{Accepts M_j
Exit}
If (($W_j=W_i$) AND ($D_j=D_i$) AND ($R_j \neq R_i$))
{Accepts M_j
Exit}
If (($W_j=W_i$) AND ($D_j=D_i$) AND ($R_j=R_i$))
{Discards M_j
Exit}
If (($W_j=W_i$) AND ($D_j \neq D_i$) AND ($R_j=R_i$))
{Discards M_j
Exit}
If (($W_j=W_i$) AND ($D_j \neq D_i$) AND ($R_j \neq R_i$))
{Accepts M_j
Exit}
i=i+1
if (i≤NO_II)
Go to LOOP
Else
Exit
}
}

(d) Priority computation of N_i:

MMM_i counts the number of high priority messages in CQ (HP_i) and the number of low priority messages in SQ (LP_i) at N_i. It also computes the safe message priority of N_i (N_Safe_i)

as $(\sum_{r=1}^{HPi}(Size\_HP_r))/HP_i$, unsafe message priority of N_i (N_Unsafe_i) as $(\sum_{l=1}^{LPi}(Size\_LP_l))/LP_i$ and priority of N_i (P_i) as $(N\_Safe_i+N\_Unsafe_i)/2$, where $Size\_HP_r$ and $Size\_LP_l$ are the size of r^{th} high priority and l^{th} low priority message in CQ and SQ respectively at N_i.

(e) Maintenance of neighbor table:

The neighbor table (Table 2) contains N number of records. MMM_i updates this table after sending and receiving a beacon message.

D_Sig	Current location	Direction	Speed	Number of igh priority message	Number of low priority message	Vehicle priority

Table 2. Neighbor table

2) Functions of CCM_i:

CCM_i computes the dynamic priority of all the messages in CQ and SQ at N_i. It schedules the messages in CQ and SQ depending upon their dynamic priority. It reserves a slot both in CCH and SCH interval depending upon the priority of N_i w.r.t. its neighboring vehicles. It also computes dynamically the size of the reserved slot for the transmission of messages waiting in CQ and SQ at N_i respectively. CCM_i transfers messages from SQ to CQ in case the SCH is overloaded and CCH is free. CCM_i also computes the loss of safe and unsafe messages at N_i.

(a) Priority assignment:

A static priority is assigned to each message depending upon its type. The static priority of Type_I, Type_II, Type_III, Type_IV and Type_V messages are assumed as Prio_I, Prio_II, Prio_III, Prio_IV and Prio_V, where Prio_I>Prio_II>Prio_III>Prio_IV>Prio_V. It computes the dynamic priority of each message in CQ and SQ using the priority assignment function.

Priority assignment function for M_j

```
{
  Prio_j:= static priority of Mj
  Data_TR:= data transmission rate//assumed as 6
            Mb/sec [17]
  DPj:= dynamic priority of Mj
  TOj:= time out of Mj in sec
  Size_j:= size of Mj in bits
  TTj:= transmission time of Mj
  TTj:= Size_j/Data_TR
  Speed_S:= speed of Source_V
  If (Mj=Type_I message)
     DPj:=Prio_j/(TOj-TTj)
  If (Mj=Type_II message or Mj=Type_IV
     message or Mj=Type_V message)
     DPj:=(Prio_j*Speed_S)/(TOj-TTj)}
```

In case of Type_I message the speed of AV is assumed as zero. So the dynamic priority of Type_I message is independent on the speed of AV. In case of Type_II message if the speed of Source_V increases, Des_V may go out of the coverage area of Source_V. In case of Type_IV message if the speed of Source_V increases it reaches R_point quickly so it needs reply from its neighboring vehicles quickly. In case of Type_V message if the speed of Source_V increases it goes out from the coverage area of N_RSU quickly so it needs the reply from N_RSU quickly. Therefore the dynamic priority of Type_II, Type_IV and Type_V messages is directly proportional to the speed of Source_V.

(b) Message scheduling:

Initially CCM_i schedules the messages in CQ and SQ in the descending order of their dynamic priority. It consults with the neighbor table which is maintained by MMM_i for arranging all the N number of vehicles in the descending order of their priority in a list. CCM_i reserves the k^{th} slot of CCH and SCH interval for the transmission of its safe and unsafe messages if the position of N_i in the list is k. It computes the size of k^{th} slot in CCH interval (CCH_{k_i}) as $HP_i*(\tau-\tau_g)*\sum_{i=1}^{N}(HP_i)$ and in SCH interval (SCH_{k_i}) as $LP_i*(\tau-\tau_g)*\sum_{i=1}^{N}(LP_i)$. If CCM_i finds that its CQ is almost empty and a lot of messages are waiting in SQ, it triggers transfer algorithm for minimizing the loss of unsafe message at N_i.

(c) Transfer algorithm:

It uses SCH_OL function to verify whether SCH is overloaded, CCH_OL function to verify whether CCH is overloaded and SQ-CQ function to transfer messages from SQ to CQ if CCH is not overloaded but SCH is overloaded.

SCH_OL function:

```
{
  UM_T= total number of unsafe messages with N number of
vehicles as obtained from neighbor table
SCH_Interval= τ-τg
X=1  // first unsafe message in SQ
TT_X=transmission time of X
Size_X=size of X
LOOP:{
 TT_X=Size_X/Data_TR
 If (TT_X=SCH_Interval)
  {Xth unsafe message is transmitted successfully
   If (X<UM_T)
     {SCH is overloaded
      Call CCH_OL function}
   Else
      {SCH is not overloaded
       Exit}}
  If (TT_X< SCH_Interval)
   {Xth unsafe message is transmitted successfully
    SCH_Interval= SCH_Interval – TT_X
    X=X+1
    If (X>UM_T)
      {SCH is not overloaded
       Exit}
    Else
       Go to LOOP}}
 If(TT_X>SCH_Interval)
  {SCH is overloaded
   Call CCH_OL function}
}
```

CCH_OL function:

{SM_T=total number of safe message with N no of vehicles as obtained from neighbor table.
CCH_Interval=τ-τ_g
Y=1 //first safe message in CQ
TT_Y=transmission time of Y
Size_Y=Size of Y
LOOP1:
{
TT_Y=Size_Y/Data_TR
If (TT_Y=CCH_Interval)
{Y^{th} safe message is transmitted successfully
If (Y<SM_T)
{CCH is overloaded
Exit}
Else
Exit}
If (TT_Y<CCH_Interval)
{ Y^{th} safe message is transmitted successfully
CCH_Interval=CCH_Interval – TT_Y
Y=Y+1
If (Y>SM_T)
Call SQ-CQ function
Else
Go to LOOP1}}
If (TT_Y>CCH_Interval)
{CCH is overloaded
Exit}
}

SQ-CQ function:

CCM_i computes CCH_{k_i} as $LP_i*(\tau-\tau_g)*\sum_{i=1}^{N}(HP_i)$ for the transmission of its excess unsafe messages in SQ using CCH.

(d) Computation of loss of messages at N_i:

CCM_i monitors CQ and SQ to find the loss of safe and unsafe messages respectively due to time out. It increases $Loss\_CQ_i$ counter by 1 after discarding a safe message from CQ due to time out and $Loss\_SQ_i$ counter by 1 after discarding an unsafe message from SQ due to time out.

3) Function of BWM_i

The offered bandwidth (OF_BW) to VANET application per 10 MHz is equal to half of the total bandwidth to avoid the saturation of available bandwidth and to achieve reliable transmission of safe messages. BWM_i consults with the neighbor table which is maintained by MMM_i to know the priority of the neighboring vehicles of N_i and to compute the effective bandwidth (E_BW_i) of N_i as $P_i/\sum_{i=1}^{N}(P_i)*OF\_BW$ Hz.

III. Simulation

The simulation experiment is conducted by varying the number of neighboring vehicles from 0 to 50 and mean message size from 0 to 500 bytes. The number of roads is assumed as 10. The performance of the proposed congestion control mechanism is evaluated experimentally according to the metric of average loss of unsafe message, average delay in the transmission of safe message, average delay in the transmission of unsafe message and storage overhead per vehicle. The performance of the present scheme is also compared with [12].

Average loss of unsafe message

The average loss of unsafe message is computed as $(\sum_{i=1}^{N}(Loss\_SQ_i))/N$.

Fig.3 shows the plot of average loss of unsafe message vs. mean message size when N is equal to 30. Fig.4 shows the plot of average loss of unsafe message vs. the number of neighboring vehicles when mean message size is 200 bytes. It can be observed from Fig.3 and Fig.4 that the average loss of unsafe message increases with mean message size and the number of neighboring vehicles.

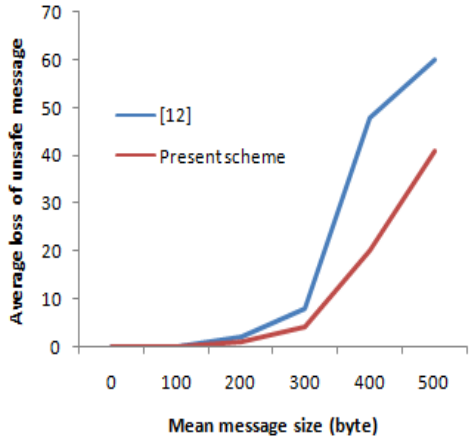

Fig.3 Average loss of unsafe message vs. Mean message size

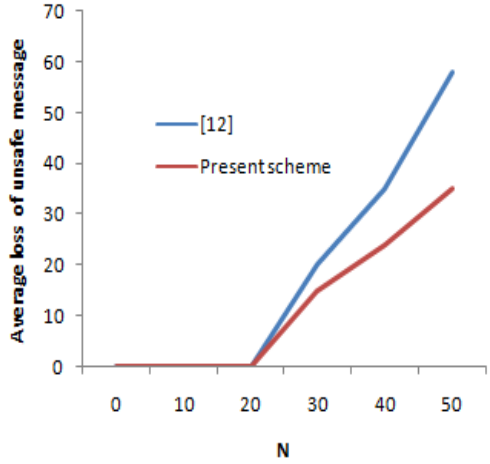

Fig.4 Average loss of unsafe message vs. Number of neighboring vehicles

Average waiting delay for safe and unsafe messages

The delay in the transmission of each message is the waiting time before its effective transmission from the appropriate queue. CCM_i determines the waiting time of each message in CQ and SQ at N_i. The waiting time of p^{th} message in a queue is the sum of the transmission time of the messages in front of it. The transmission time of a message is the ratio of its size and Data_TR.

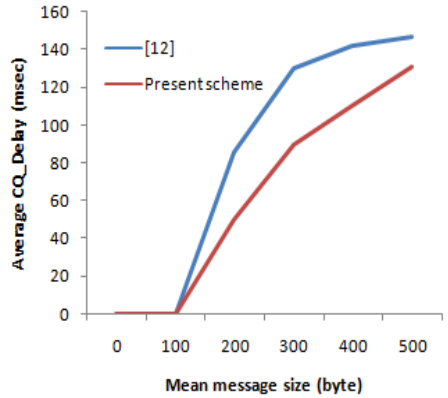

Fig.5 Average CQ_Delay vs. Mean message size

Let W_{ip_cq} be the waiting time of p^{th} message ($1 \le p \le HP_i$) in CQ and W_{iq_sq} be the waiting time of q^{th} message ($1 \le q \le LP_i$) in SQ at N_i. So the average delay in CQ (CQ_Delay) is $(\sum_{i=1}^{N} \sum_{p=1}^{HPi} (W_{ip\_cq}))/N$ seconds and in SQ (SQ_Delay) is $(\sum_{i=1}^{N} \sum_{q=1}^{LPi} (W_{iq\_sq}))/N$ seconds.

Fig.5 shows the plot of average CQ_Delay and Fig.6 shows the plot of average SQ_Delay vs. the mean message size when N is equal to 30. It can be observed from Fig.5 and Fig.6 that both CQ_Delay and SQ_Delay increase with mean message size.

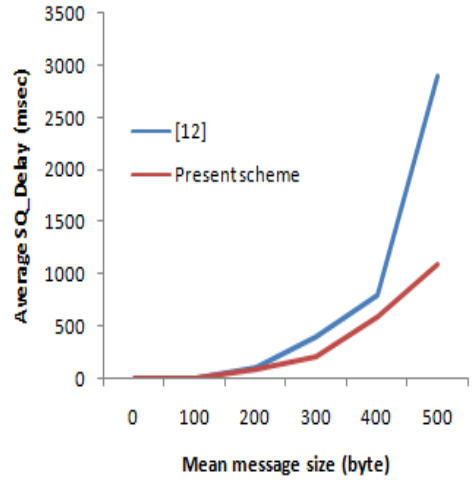

Fig.6 Average SQ_Delay vs. Mean message size

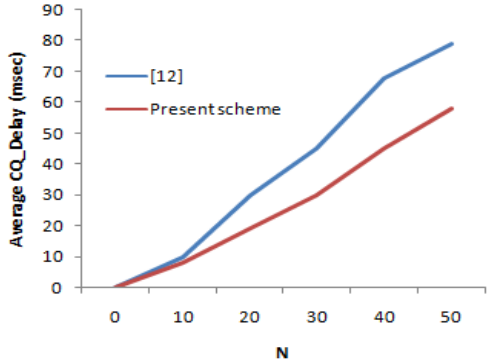

Fig.7 Average CQ_Delay vs. Number of neighboring vehicles

Fig.7 shows the plot of average CQ_Delay and Fig.8 shows the plot of average SQ_Delay vs. the number of neighboring vehicles when mean message size is equal to 200 bytes. It can be observed from Fig.7 and Fig.8 that both CQ_Delay and SQ_Delay increase with the number of neighboring vehicles.

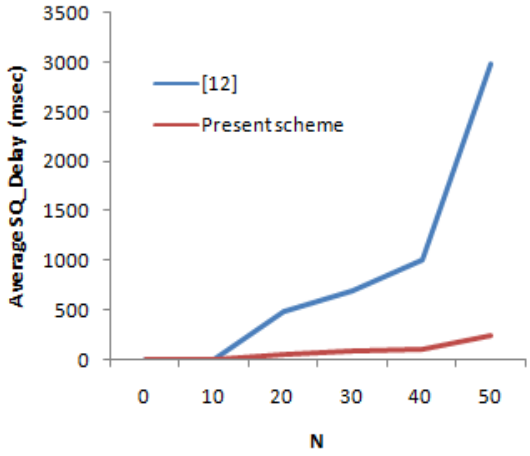

Fig.8 Average SQ_Delay vs. Number of neighboring vehicles

Storage overhead at N_i

The storage overhead at N_i (STO_OH_i) is the sum of overhead for maintaining CQ (STO_CQ_i), SQ (STO_SQ_i) and neighbor table (STO_NT_i). As MMM_i distributes the messages in IQ among CQ and SQ so STO_OH_i is assumed as independent on the size of IQ. The average message size in CQ (M_avg_{CQ}) is (Size_I+Size_II+Size_III)/3 bits and the average message size in SQ (M_avg_{SQ}) is (Size_IV+Size_V)/2 bits at N_i. So STO_CQ_i is $HP_i * M\_avg_{CQ}$ bits and STO_SQ_i is $LP_i * M\_avg_{SQ}$ bits. STO_NT_i is $N*(233+\log_2 HP_i+\log_2 LP_i)$ bits.

STO_OH_i in [12] is $N*(233+2\log_2 L)$ + L*Size_II + L*Size_IV bits, where L is the length of CQ and SQ. The identity attribute of the records in the neighbor table of [12] is replaced by the digital signature of the corresponding vehicle.

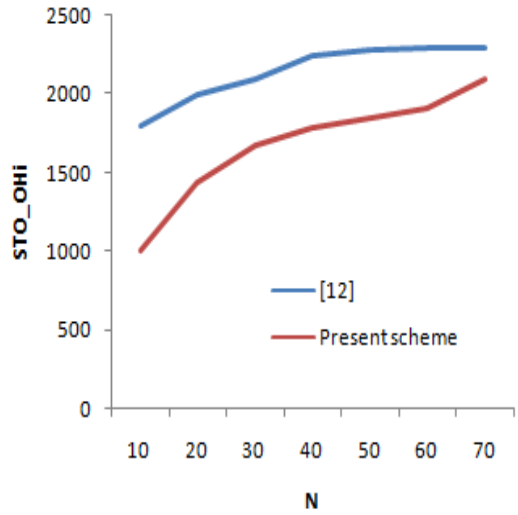

Fig.9 STO_OH_i vs. Number of neighboring vehicles

Fig.9 shows the plot of STO_OH_i vs. the number of neighboring vehicles. It can be observed from Fig.8 that STO_OH_i increases with the number of neighboring vehicles.

Discussion of results:

During simulation it has been observed that the increase in the number of neighboring vehicles reduces the effective bandwidth per vehicle and increases the message generation rate. As the effective bandwidth per vehicle reduces CCH remains busy most of the time. So the possibility of switching of unsafe messages from SQ to CQ reduces which in turn increases the average loss of unsafe message.

The increase in message generation rate increases the number of message both in CQ and SQ. As a result the waiting time of messages both in CQ and SQ increase which in turn increases the average delay in transmission of safe and unsafe messages.

The increase in mean message size increases its transmission time which in turn increases the waiting time of messages in CQ and SQ. As a result the average loss of unsafe message increases, average delay in transmission of safe and unsafe message also increases. Moreover the average CQ_Delay is lower than the average SQ_Delay due to the high priority of messages in CQ.

The present scheme considers variable length beacon message and removal of duplicate messages from each vehicle. Moreover it allows only the authentic vehicles to participate in V2V and V2I communication. It helps to reduce the number of messages at each vehicle which in turn reduces the effective channel load. So the average loss of unsafe message, average waiting delay and STO_OH_i are less in the present scheme than [12].

IV. Conclusion

The proposed work is a congestion control mechanism in VANET. It also identifies the misbehaving vehicles and revokes then from VANET. It considers both V2V and V2I communication of safe and unsafe messages among authentic vehicles. Each vehicle has a message manager module to manage the sending and receiving messages, a congestion control module to schedule the waiting messages in queue and a bandwidth manager module to calculate its effective bandwidth dynamically. The present work can be extended by evaluating the dynamic priority of a message as a function of network condition in terms of vehicle density, congestion level of the network etc.

References

[1] Y. Zang, L.Stibor, X.Cheng, H. -J.Reumerman, A.Paruzel and A. Barroso, “Congestion control in wireless networks for vehicular safety applications”, 8th European wireless conference, p.7, Paris, France, 2007.

[2] M.Y.B.Darus and K.A.Bakar, “Congestion control framework for disseminating safety messages in vehicular ad-hoc networks”, International journal of Digital content technology and its applications, vol.5, no.2, February 2011

[3] M.S. Bouassida and M.Showky, “A cooperative congestion control approach within VANETs: formal verification and performance evaluation”, Hindawi publishing corporation, Eurasip journal on wireless communications and networking, pp. 1-26, article no. 11, 2010.

[4] M.S.Bouassida and M.Shawky, “On the congestion control within VANET”, Wireless days, WD '08 1st IFIP Digital Object Identifier: 10.1109/WD.2008.4812915, pp. 1–5, 2008.

[5] C.Campolo, A. Cortese and A. Molinaro, “CRaSCH: A cooperative scheme for service channel reservation in 802.11p/WAVE vehicular ad hoc networks”, Proc. of international conference on ultra modern telecommunications & workshops, pp. 1-8, 2009.

[6] J.He, H-H. Chen, T.M.Chen and W.Cheng, “Adaptive congestion control for DSRC vehicle networks”, IEEE communication letters, vol.14, no.2, pp. 127-129, February 2010.

[7] L.Wischhof and H.Rohling, “Congestion control in vehicular ad hoc networks”, IEEE international conference on vehicular electronics and safety, Germany, pp.58-63, 2005.

[8] W.Zhang, A.Festag, R.Baldessari and L.Le, “Congestion control for safety messages in VANETs: concepts and framework”, Proceedings 8th conference on ITS telecommunications, pp.199-203, 2008.

[9] M. Y. Darus and K. A. Bakar, “Formal Verification of Congestion Control Algorithm in VANETs”, International Journal of Computer Network and Information Security, vol.4, pp.1-7, 2013.

[10] M. Y. Darus and K. A. Bakar, “Congestion Control Algorithm in VANETs”, World Applied Scisnces Journal, vol.21, no.7, pp.1057-1061, 2013.

[11] L.Humeng, Y. Xuemei, A. Li and W. Yuan, “Distributed Beacon Frequency Control Algorithm for VANETs (DBFC)”, International Conference on Intelligent Systems Design and Engineering Application, pp. 243-246, 2012.

[12] T. Ghosh and S.Mitra, Congestion control by dynamic sharing of bandwidth among vehicles in VANET, 12th international conference on intelligent systems design and applications, pp. 291-296., 2012.

[13] C.Campolo, A.Vinel, A.Molinaro and Y.Koucheryavy, “Modeling broadcasting in IEEE 802.11p//WAVE vehicular networks”, IEEE communications letters, vol.15, no.2, pp.199-201, February 2011.

[14] A. Mondal and S. Mitra, Identification, authentication and tracking algorithm for vehicles using VIN in centralized VANET. International conference on advances in communication, network, and computing, LNICST 108, pp. 115 – 120, 2012.

[15] A. Kahate, 2010. Cryptography and network security, 2ND Edition, TMH.

[16] http://www.angelfire.com/ca/TORONTO/VIN/VIS.html

[17] Towards effective vehicle identification, The NMVTRC’s strategic framework for improving the identification of vehicles and components, (2004).

Author Biographies

Sulata Mitra, received B.E. degree from Bengal Engineering College (India) in 1986 and PhD degree in Mobile Computing from Bengal Engineering College (D.U.), Shibpur (India) in 2005. She joined the Indian Institute of Technology, Kharagpur in 1989 as Senior Research Assistant and moved to the Regional Institute of Technology, Jamshedpur (India) in 1991 as Lecturer. Dr. Mitra has published 50 technical papers in journal and international conference proceedings. Her current research interest is QoS issues in 3G/4G cellular network, VANET, Multihomed mobile network. She is currently with the Computer Science and Technology Department of Bengal Engineering and Science University, Shibpur (India) as Associate Professor.

Trishita Ghosh, received B.E. degree from University Institute of Technology, Burdwan University in 2007 and M.Tech in 2009 from Calcutta University. She joined Gurunanak Institute of Technology, Sodepur as a Lecturer in 2010. Currently she is an Assistant Professor of department of Information Technology at Gurunanak Institute of Technology.