

Research Article

MCNC: Data Aggregation and Dissemination in Vehicular Ad hoc Networks Using Multicast Network Coding

Lingzhi Li,¹ Shukai Zhang,^{1,2} Yanqin Zhu,¹ and Zhe Yang¹

¹ School of Computer Science & Technology, Soochow University, Suzhou 215006, China

² State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210093, China

Correspondence should be addressed to Shukai Zhang; zhangsk2000@163.com

Received 12 April 2013; Accepted 4 September 2013

Academic Editor: Liusheng Huang

Copyright © 2013 Lingzhi Li et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Nowadays, more and more APs are deployed for the wireless access. The roadside APs are able to improve the bandwidth of VANETs. In this paper, we design a Multicast Network Coding (MCNC) method to aggregate and disseminate data in VANETs. It is appropriate for the areas where APs have been deployed but cannot cover the all range of vehicles movement. Data transmission is divided into 3 stages in MCNC. Network coding, multicast aggregating, and decoding are implemented in the 3 stages, respectively. Simulation results show that MCNC is able to cut down 2/3 overhead messages of Epidemic when the cover area of the APs is only 1/5. Its loss rate and delay are lower than the other methods. We also discuss its benefit for the deployment of VANETs.

1. Introduction

Vehicular networks are usually regarded as a class of ad-hoc networks, where mobile nodes are vehicles exchanging data with each other. Vehicle can communicate with nearby vehicles when their distance is less than the communication range. Data are delivered among vehicles via Vehicle-to-Vehicle (V2V) protocol. Vehicular Ad hoc Networks (VANETs) will diverse applications associated with traffic safety [1], traffic efficiency [2], entertainment, file sharing [3, 4], and so on [5]. However, vehicles move at different speeds and form dynamic networks over time. VANET changes so fast; it is often difficult to find a path between a pair of vehicles using ordinary ad hoc routing protocol. Epidemic is the high reliability protocol, but it will waste too much bandwidth and reduce the scalability of networks [6].

Deploying infrastructures is a method to improve the reliability of data delivery. Intelligent VANET is a way of integrating on multiple wirelesses such as IEEE 802.11, 3G cellular systems, and IEEE 802.16e in vehicular networks [7]. Vehicles communicate with roadside infrastructures via Vehicle-to-Infrastructure (V2I) protocol. The infrastructures are connected via the wired networks and are able to buffer more data than vehicles. The experiments in [8] show that the

deployment of infrastructures can bring up to the improvement of performance in delivery ratio. With the increasing use of smart phones, 3G cellular systems become a popular infrastructure, but the research in [9] suggests that cellular data bandwidth is likely to remain limited and expensive. With the recent development of the IEEE 802.11p [10] 5.9G DSRC/WAVE radios, Wi-Fi based Access Points (APs) on the side of road will be more as the infrastructures of VANETs.

Most applications of VANETs focus on planning vehicle paths, such as detecting remote road conditions [11]; sensing traffic lights [12]; avoiding rear collisions [10]. These applications require disseminating data to part of vehicles in some regions. The receivers are dynamic and cannot be identified by senders because of vehicles constant movement. It is impossible that infrastructures signal covers all of regions reached by vehicles. Multihop communications among vehicles are necessary to destinations or infrastructures. So routing packet remains to be a challenge in such network architecture which integrates V2V communication with V2I communication. However, vehicle movements are constrained to roads. Even though lots of vehicles are equipped with GPS receivers or smart phones to define their routes, their movements are probably constrained to preset paths. It is another characteristic of VANETs that it is possible to predict the vehicle movement [13].

In this paper, we present a Multicast Network Coding (MCNC) method to aggregate and disseminate data in VANETs. Data are divided into short fixed-size blocks in source vehicle and disseminated to the other vehicles. The vehicles nearby APs implement network coding and send the coding blocks to APs. The blocks are aggregated and recoded among APs multicast tree. Finally, the blocks are sent out in APs that the destinations are predicted to close. The experiments show that MCNC is able to reduce overhead messages and enhance the reliability of transmission. MCNC is designed according to the former characteristic of VANETs and appropriate for the areas that Wi-Fi based APs have been set up but cannot cover entirely.

The remainder of this paper is organized as follows. In the next section, we discuss the related work. Section 3 presents the description and implementation of the MCNC. The evaluation and analysis of the MCNC are showed in Section 4. The paper is concluded in Section 5.

2. Related Works

In recent years, many studies focus on deploying infrastructures in VANETs. It is proved [8] to bring up the improvement of data delivery. Cabernet [14] proposes one-hop Internet access schemes using open Wi-Fi based APs in VANETs. In [15], Bychkovsk et al. analyze the feasibility that vehicles can access open Wi-Fi based APs providing the wired network connections in VANETs. They define the prime method of data delivery from APs to vehicles. Throwboxes [16] as infrastructures can hold packets and later forward the packets to other nodes. It helps relay messages between mobile clients in a Delay Tolerant Network (DTN) and saves power in scheduling sleep. MAR [17] as an infrastructure is a commuter router that exploits wireless diversity to provide improved data for users. It can be placed in moving vehicles to enable high-speed data access. DP [18] on the roadside is proposed to address the data dissemination in VANETs. Data poured from the source are buffered and rebroadcast at the intersections. It focuses on data dissemination, and no buffer allocation scheme is proposed. MobTorrent [19] is a framework designed for vehicles accessing to roadside Wi-Fi APs. Vehicles inform the selected APs to fetch data. The scheduling algorithm in MobTorrent replicates the data on mobile helpers.

A few studies have been attracted on utilizing vehicle trajectory information for VANETs of deploying APs. More and more vehicles are equipped with satellite navigation systems. With increasing usage of smart phones like Apple iPhone or Android phones, navigation systems based on smart phones are popular. Vehicle trajectory information has become a valuable input for data forwarding. In [20], a protocol is proposed to enable efficient multihop routing capabilities. It fully supports two-way communications between mobile vehicles and APs. Vehicular mobility is predicted on the information offered by the navigation system in terms of final destination and path. TSF [21] considers a reliable, efficient APs-to-vehicles data delivery by minimizing the packet delivery delay. Data delivery is performed through

the computation of a target point based on the destination vehicle's trajectory information. Roadside APs can be selected as relays where destination vehicle is expected to pass by. Selecting AP optimally minimizes the packet delivery delay. TBD [22] utilizes vehicle trajectory information to improve communication delay and delivery probability for vehicles-to-APs destination communications. A delay model of packets routing along roads is set up. The path with minimum delay can be found with the help of the real-time traffic condition information.

There is also some research leveraging roadside parking to distribute data in VANETs. ParkCast [23, 24] assumes that the parked vehicles are grouped into a line cluster at roadside. It defines the communication between moving vehicles and parked ones. The wireless device on vehicles has a battery for supporting the communication in parking. APs are replaced by parked vehicles to delivery data in VANETs. PASS [2] is a parking-lot-assisted carpool method. It optimizes transport utilization by sharing ride among drivers. A user can get vehicles information from the drivers in PASS. The drivers decide whether to provide carpooling services or not. In fact, many V2V protocols are also able to be used as vehicles-to-APs communications. Epidemic Routing [25] is the first proposed to reliably deliver messages in intermittently connected networks. It is not optimized for the VANETs. PROPHET [26] was proposed to select the next hops using the history of encounters. Both of them are used as the typical protocols of VANETs and will be analyzed in our simulation.

Network coding has been proven to be a promising approach that can improve the reliability of communication. A few studies focus on using network coding in VANETs. CodeTorrent [27] is a file swarming protocol based on network coding for VANETs. It deals with typical mobile network issues such as dynamic topology and intermittent connectivity. CodeOn [28] is a push-based scheme for popular content distribution in VANETs. Data are actively broadcasted to vehicles from APs and further distributed among vehicles. It uses network coding to reduce the lossy of data. In [29], M. Sathiamoorthy et al. analyze the benefits of distributed storage using erasure codes for file sharing in VANETs. Distributed storage codes not only offer the reduction of bandwidth occupied by large file copy but also provide significant reduction of cost in VANETs. In [30], Palma et al. propose a data dissemination technique based on Fountain codes. Its goal is provide the real-time service in a lossy vehicular network. It achieves an improvement on arrival times of packets towards destination vehicles. The works adopt network coding to handle content distribution in VANETs and are referred for coding and transmission of this paper.

3. MCNC and Its Implementation

The process of information delivery is similar in the network architectures which integrate V2V with V2I communication. The sources firstly forward data to infrastructures in multihop communications among vehicles. Data are transmitted in wired networks formed by infrastructures and then are sent

from a few infrastructures to the destinations in multihop wireless communications. In our MCNC method, we define the 3 stages as ToInfra, BeInfra, and ToVehic.

In this section, we present, respectively, the method of data transmission on the 3 stages of MCNC. It firstly composes network coding and multicast to aggregate data in VANET.

3.1. ToInfra Stage. Data dissemination is unreliable when information is announced by the source far away from infrastructures. Before the vehicle is able to contact with infrastructures, the generic routing algorithm for VANETs is used among vehicles. We select the epidemic routing in this moment to describe our method. When vehicles sense a route to infrastructures, they will aggregate and send out data in the way of network coding. If data meet part of destinations in epidemic routing [26], they will be received directly by these nodes.

The communication between vehicles and infrastructures needs to be completed in a very short time when vehicles are moving fast. In order to transmit data integrally and realize the network coding operation, the source data are divided into a number of blocks that are the same as maximum length. Every data block contains a vehicle ID and a block ID. $B(p, q, t)$ is denoted as the block q sent by vehicular p . t is the time to the block timeout discarded. Every block has the live time to avoid heavy network load. The current value of t is taken to next hop to continue reducing over time. Block is discarded while $t = 0$.

The live time T_{\max} is the maximum values of t . t is counted down at T_{\max} while the block is generated from the source. T_{\max} can be computed by (1)

$$T_{\max} = \frac{w_0 * C}{\text{int}(\text{Flow}(t) + 1)}, \quad (1)$$

where C is the class of the packet QoS and w_0 is the constant adjusting T_{\max} to proper value. $\text{Int}()$ is the integral function. $\text{Flow}(t)$ is the current vehicle flow and given by

$$\text{Flow}(t) = \alpha * \text{Sum}(t) + (1 - \alpha) * \text{Flow}(t - 1). \quad (2)$$

$\text{Sum}(t)$ is the sum of vehicles that communicate with this vehicle in time t . The coefficient α is a constant value between 0 and 1, representing the degree of $\text{Flow}(t)$ decreases. The greater the value, the faster the older affects are depreciated.

Infrastructures send active messages periodically. l is the TTL of active message. $l = l - 1$ when the message is forwarded by a vehicle. Vehicles send the active message

immediately after receiving it and then do not send the same message. The active message will not be forwarded any more while $l = 0$. L_{\max} is the TTL of the active message sent by infrastructures. L_{\max} can be computed by (3) as follows:

$$L_{\max} = \text{int} \left(E \times \left(w_1 * \ln \left(1 + \frac{\sum_{i=1}^{\text{Sum}(t)} v_i}{\text{Sum}(t)} \right) + w_2 * \text{Sum}(t) + w_3 \right)^{-1} \right), \quad (3)$$

where E is the signal intensity sent by infrastructures. The higher the power of infrastructures, the more the TTL of active message will be. $\text{Sum}(t)$ is the sum of vehicles that communicate with this infrastructure in time t . v is the velocity of these vehicles. The more the vehicles are connected by the infrastructure and the faster the vehicles move, the less the TTL of active message will be. w_1 , w_2 , and w_3 are the weights of the former parameters, respectively. $L_{\max} = 1$ while the value obtained by (3) is less than 0. $L_{\max} = 15$ while the value obtained by (3) is more than 15. In other words, $1 \leq L_{\max} \leq 15$.

Linear network coding begins immediately when the vehicle receives the active message of infrastructures. The blocks $B(p, q, t)$ where $t > 0$ will be encoded and sent to infrastructures. In order to reduce the complexity of decoding operation, the dimension of encoding vector is taken a fixed value ω . A row vector contains ω consecutive blocks sent by the same vehicle. The blocks IDs in the vector i are from $(i - 1) * \omega + 1$ to $i * \omega$, where $i = 1, 2, 3, \dots$. If the blocks of ID within this range are not received, their values will be set to 0.

The blocks sent by the vehicle p are divided into block vectors in sequence of q . $B(p, q, t)$ should equal to the element j of the vector i , i and j are given by

$$\begin{aligned} i &= \text{int} \left(\frac{q}{\omega} \right) + 1, \\ j &= q - \omega * i. \end{aligned} \quad (4)$$

$x(p)_{ij}$ is denoted as the element j of block vector i sent by vehicle p . $x(p)_{ij}$ can be given by

$$x(p)_{ij} = B(p, (i - 1) * \omega + j, t). \quad (5)$$

$X(p)$ is the matrix consisting of all block vectors sent by vehicle p . $X(p)$ is denoted as

$$X(p) = \begin{bmatrix} X(p)_1 \\ X(p)_2 \\ \vdots \\ X(p)_v \end{bmatrix} = \begin{bmatrix} B(p, 1, t) & B(p, 2, t) & \cdots & B(p, \omega, t) \\ B(p, \omega + 1, t) & B(p, \omega + 2, t) & \cdots & B(p, 2\omega, t) \\ \vdots & \vdots & \vdots & \vdots \\ B(p, (v - 1) * \omega + 1, t) & B(p, (v - 1) * \omega + 1, t) & \cdots & B(p, v * \omega, t) \end{bmatrix}. \quad (6)$$

K is the local encoding matrix of the vehicle receiving the active message. All of elements in the matrix originate a

pseudorandom sequence produced according to the vehicle ID. We reform the Fibonacci Algorithm [31] to generate the

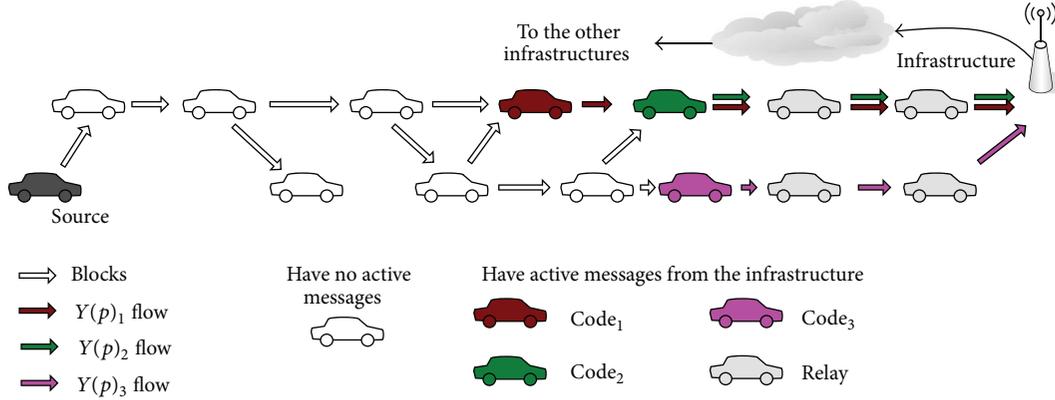


FIGURE 1: Data dissemination among vehicles on ToInfra stage.

sequence. $\text{Seq } k$ denotes the pseudorandom sequence and is given by

$$\begin{aligned} \text{Seq } k_a &= \frac{Sn_a}{\bar{\omega}}, \quad a = 1, 2, 3, \dots, \\ \bar{\omega} &= \max(\omega, v), \\ Sn_a &= (Sn_{a-2} + Sn_{a-1}) \bmod \bar{\omega}, \\ Sn_{-1} &= \text{VID}_{\text{source}} \bmod \bar{\omega}, \quad Sn_0 = \text{VID}_0 \bmod \bar{\omega}, \end{aligned} \quad (7)$$

where $\text{VID}_{\text{source}}$ is the unique identification of the source vehicle and VID_0 is the unique identification of the vehicle executing network coding. The element k_{ij} in the matrix K is gotten from $\text{Seq } k$ and given by

$$k_{ij} = \text{Seq } k_{i*\omega+j}. \quad (8)$$

K is the global coding matrix with $\lambda \times \omega$ dimension and can be denoted as follows:

$$\begin{aligned} K &= \begin{bmatrix} k_{11} & k_{12} & \dots & k_{1\omega} \\ k_{21} & k_{22} & \dots & k_{2\omega} \\ \vdots & \vdots & \vdots & \vdots \\ k_{\lambda 1} & k_{\lambda 2} & \dots & k_{\lambda\omega} \end{bmatrix} \\ &= \begin{bmatrix} \text{Seq } k_1 & \text{Seq } k_2 & \dots & \text{Seq } k_\omega \\ \text{Seq } k_{\omega+1} & \text{Seq } k_{\omega+2} & \dots & \text{Seq } k_{2*\omega} \\ \vdots & \vdots & \vdots & \vdots \\ \text{Seq } k_{(\lambda-1)*\omega+1} & \text{Seq } k_{(\lambda-1)*\omega+2} & \dots & \text{Seq } k_{\lambda*\omega} \end{bmatrix}. \end{aligned} \quad (9)$$

Implementing linear network coding on the blocks from the vehicle p is formalized as computing the product of two matrixes. It is denoted as

$$Y(p) = K \cdot X(p)^T. \quad (10)$$

The dimension of matrix $Y(p)$ is $\lambda \times v$, and its element $y(p)_{ij}$ can be computed by

$$\begin{aligned} y(p)_{ij} &= \sum_{m=1}^{\omega} k_{im} * x(p)_{jm} \\ &= \sum_{m=1}^{\omega} \text{Seq } k_{i*\omega+m} * B(p, (j-1) * \omega + m, t). \end{aligned} \quad (11)$$

The blocks sourced from the different vehicles have the different opportunity if they send the whole matrix $Y(p)$ successively. It is possible that the blocks sent later are not able to reach infrastructures. The movements of vehicles make the path unreachable. In order to make the equal opportunity for every vehicles transmitting information, the vehicles receiving active message only send an element of $Y(p)$, then send an element of $Y(p+1)$, and so on.

An infrastructure can receive multiple network codes that source from the same ω consecutive blocks but are computed by local encoding matrix of different vehicles (see Figure 1). Receiver can decode the ω blocks of row i completely when $Sc(p, i) \geq \omega$. $Sc(p, i)$ denotes the number of nonlinear codes received by the infrastructure and encoded by the i row blocks of the vehicle p . The element of $Y(p)$ in the same row is the nonlinear codes of the same ω blocks. The other vehicles also encode the ω blocks and send to the infrastructure. So the receiver maybe decodes the ω blocks completely when a vehicle only sends a few front elements of a row. The vehicle will send elements of coding matrix by columns. The infrastructure will send Break messages to the coding vehicles when $Sc(p, i) \geq \omega$. The vehicle breaks off network coding process when receiving Break messages.

When the vehicle on ToInfra stage receives a message, its process is formally shown in Figure 2.

Network coding algorithm on Figure 2 is described in Algorithm 1.

3.2. BeInfra Stage. Infrastructures are connected by wired media. They consist of wireless APs, network nodes, servers, and so on. It is able to guarantee the reliability and the real-time of data transmission among infrastructures.

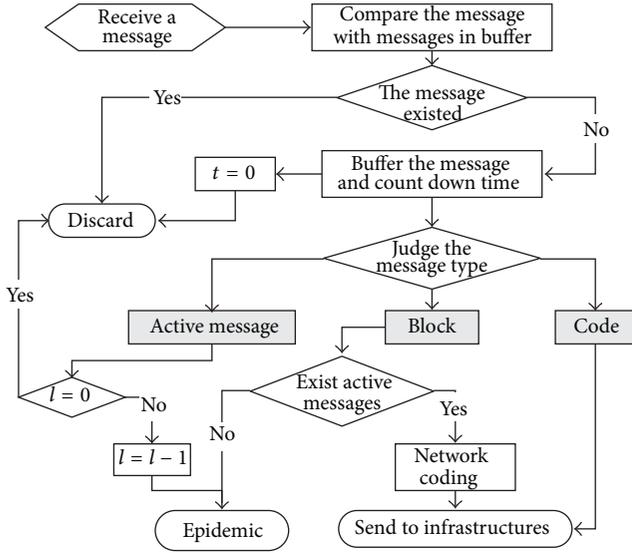


FIGURE 2: The process flowchart of vehicle on ToInfra stage.

The destinations of most blocks are not all of vehicles in VANET. We use multicast as the way of communication to aggregate data on BeInfra stage. Unicast and broadcast are regarded as two especial types of multicast in the method. Their group members are all or one node.

When wireless AP receives the coding blocks from vehicles, it will send these blocks to another node according to their destinations. The node is the root of multicast tree. The root aggregates and recodes these blocks according to their destinations and then sends them to egress APs following the tree. Creating tree for every multicast group will make more forward states on middle node and cause more delay in data transmission. We aggregate all of multicast groups in a few share trees to resolve the former problem on BeInfra stage.

3.2.1. Constructing Multicast Shared Tree. The multicast group means the set of the APs leading to members rather than the set of the members themselves. The domain of infrastructures contains at least one Network Manager Server (NMS). It controls the computation of all shared trees. NMS maintains a link-state database describing the topology of infrastructures.

There are one ingress router and multiple egress routers in a shared tree. The shared tree is rooted as a network node and leafing as egress APs. NMS selects routers as the root and leaves and computes the topology of shared tree. We partition all APs into subsets and cluster the members of all multicast group in the subsets in here then select the leaves and root of the shared tree. The method includes the detailed steps as the following.

(1) *Partitioning Aps.* All APs are partitioned according to the topology of infrastructure networks. Clustering algorithms are required in here. We use fuzzy clustering algorithm to

partition APs into subsets. First, r_{ij} denotes the relation strength between AP i and j and is given by

$$r_{ij} = \begin{cases} 1 & i = j; \\ \left[\min(D_{ij} * (H_{ij})^\gamma) \right]^{-1} & i \neq j. \end{cases} \quad (12)$$

γ is a constant to adjust the weight of D_{ij} and H_{ij} , and $\gamma \geq 0$. D_{ij} is the minimum delay (or the shortest path length) from AP i to j , and H_{ij} is the hop count of the path. The value of r_{ij} is concerned with the shortest distance between two APs. The less the distance is, the more the r_{ij} value is. The relational matrix R' is defined as the following equation:

$$R' = (r_{ij})_{n \times n}, \quad (13)$$

where n is the total number of APs in VANETs.

In order to prove a fuzzy equivalence relation, we must demonstrate that R' is reflexive symmetrical and transitive [32]. R' is reflexive because all its diagonal elements are equal to 1. Suppose all links are dull-duplex in here, and their shortest path sourced either extremities is the same between AP i and j , and so $r_{ij} = r_{ji}$, $R' = (R')^{-1}$, R' is symmetrical. All of the other elements are less than diagonal elements, and so the inner product of R' is computed by

$$(R')^2 = R' \circ R' = \left(\bigvee_{k=1}^n (r_{ik} \wedge r_{kj}) \right)_{n \times n} = (r_{ij})_{n \times n} = R'. \quad (14)$$

R' is transitive because $(R')^2 \subseteq R'$. Therefore R' is a fuzzy equivalence relation.

The partition of APs set can be got from λ -cut sets of the fuzzy equivalence relation R' . The cut set is denoted as $(R')^\lambda$, and the value of its elements r_{ij}^λ is obtained by the following equation:

$$r_{ij}^\lambda = \begin{cases} 0 & r_{ij} < \lambda \\ 1 & r_{ij} \geq \lambda. \end{cases} \quad (15)$$

The set of All LERs are partitioned into many subsets. The λ belongs to $[0, 1]$; the less the value of λ is, the more the number of subsets is.

(2) *Aggregating Members.* NMS gives an identifier to every sub-set in the infrastructures domain and finds members of every multicast group according to the route-state tables of APs. Multicast member is replaced with the ID of sub-set including the member, and the multicast group G (member₁, ..., member_N) is transferred to $G'(ID_1, \dots, ID_M)$, which is defined as Class Group. The same IDs are merged into one element in G' . All multicast groups are indexed in G' and saved as their registration in NMS.

NMS contrasts the G' in term of sub-set ID. The number of G' with the same ID is denoted as the compression parameter C . If C is more than a specific threshold α ($\alpha \geq 2$), those groups are aggregated to use a Class Group.

(3) *Selecting Leaves.* There is only one leaf for a shared tree in every sub-set. The leaf is an AP in or out the sub-set. NMS

```

 $S_{n-1} = \text{VID}_{\text{source}} \bmod \omega, S_{n_0} = \text{VID}_0 \bmod \omega$ 
For  $i = 1$  to  $\lambda$ 
  For  $j = 1$  to  $\omega$ 
     $S_n = (S_{n-1} + S_{n_0}) \bmod \omega;$ 
     $S_{n-1} = S_{n_0}; S_{n_0} = S_n;$ 
     $k[i, j] = S_n / \omega;$ 
  For  $j = 1$  to  $v$ 
    For every vehicle  $p$ 
      For  $i = 1$  to  $\lambda$ 
        sum = 0;
        For  $m = 1$  to  $\omega$ 
          sum = sum +  $k[i, m] * B[p, (j-1) * \omega + m];$ 
         $Y[p, i, j] = \text{sum};$ 
        Send  $Y[p, i, j]$  to next hop
      If receiving Break Message exit.

```

ALGORITHM 1: Formal description of network coding on ToInfra stage.

finds the multicast groups, which are aggregated to one Class Group G' . Then NMS divides all members of those multicast groups into Vertex set according to sub-set ID. The members in same G' and sub-set are placed a Vertex set. The Vertex set is linked into a sub-graph of the infrastructures domain. The nodes of the sub-graph consist of all APs in the Vertex set and the routers as intermediate nodes.

We select the median point of the sub-graph as the leaf of shared tunnel. The parameter D_i , which denotes the sum of distances from a node i in the sub-graph to all members of Vertex set, is computed by

$$D_i = \sum_{j=1}^m \left(d_{\min}(i, j) * \left(\sum_{k=1}^C b(j, k) \right) \right), \quad (16)$$

where $d_{\min}(i, j)$ is the length of shortest path between node i and member j and m is the number of members in the Vertex set. The parameter $t(j, k)$ is the bandwidth that multicast group k spares member j . The compression parameter C is the number of multicast groups aggregated into the shared tree. $b(j, k) = 0$ when multicast group k does not include member j . NMS selects the node with the minimum parameter D as the leaf of shared tree in the sub-set.

(4) *Selecting Multicast Root.* The root of shared tunnel is selected by the method similar to the selecting leaves. It can be any node in the infrastructures domain. The parameter P_i , which denotes the sum of distances from a node i to all sources of multicast group aggregated into the tree, is defined by the following equation:

$$P_i = \sum_{j=1}^C (d_{\min}(i, j) * T_j), \quad (17)$$

where $d_{\min}(i, j)$ is the length of shortest path between node i and source j and T_j is the bandwidth occupied by multicast flow from source j . NMS selects the node with the minimum

parameter P as the root of the shared tree. So the P of root P_r must satisfy the following condition:

$$P_r = \min_{1 \leq i \leq N} \left(\sum_{j=1}^C (d_{\min}(i, j) * T_j) \right). \quad (18)$$

N is the number of nodes in the infrastructures domain.

(5) *Computing Multicast Shared Tree.* NMS computes the shared tunnel tree for the Class Group which connects the selected root and leaves. It can use any existing multicast tree algorithm [22] (e.g. the Dijkstra's algorithm, Steiner tree-based heuristic routing algorithm with the objective of minimizing the cost on multi-QoS constraint). How the multicast tree is computed in the algorithms is outside the scope of this paper.

3.2.2. *Recoding on Root.* APs send coding blocks to the roots of multicast shared tree according to their destinations. The root discards the same coding blocks and send Break messages to the coding vehicles when $Sc(p, i) \geq \omega$. Then the root will recode to the coding blocks with the same destinations and sources. The recoding blocks are transferred to the APs along the multicast shared tree. The APs in the tree cover the area where destinations vehicles arrive at or will arrive at.

ω nonlinear codes of blocks in the same row are selected to recode with the global coding matrix, which is denoted as F . The matrix F is unique and constant to the infrastructures and need not be transferred on VANETs. All vehicles have the duplicate copy of F and decode the recoding blocks using F . All of elements in F originate a pseudorandom sequence. Seq f denotes the pseudorandom sequence and is given by

$$\text{Seq } f_a = \frac{(\arcsin Sm_a)}{\pi} + 0.5, \quad a = 1, 2, 3, \dots, \quad (19)$$

$$Sm_a = 1 - 2 * Sm_{a-1}^2, \quad Sm_{a-1} \in (-1, 0.5) \cup (0.5, 1).$$

Sm_a is an intermediate variable to produce the sequence. Sm_0 can be designated by the VANET administrator, and its

```

Sm = 0.3;
For i = 1 to ξ
  For j = 1 to ω
    Sm = 1 - 2 * Sm2;
    f[i, j] = arcsin(Sm)/π + 0.5;
  Select ω received rows from Y to Ys;
  For j = 1 to υ
    For every vehicle p
      For i = 1 to ξ
        sum = 0;
        For m = 1 to ω
          sum = sum + f[i, m] * Y[p, m, j]s;
        Z[p, i, j] = sum;
        Send Z[p, i, j] to leafs of multicast shared tree;
        If receiving ACK Messages of all leafs exit.

```

ALGORITHM 2: Formal description of recoding on root.

default value is equal to 0.3. The element f_{ij} in the matrix F is gotten from $\text{Seq}f$ and given by

$$f_{ij} = \text{Seq} f_{i*\omega+j}. \quad (20)$$

The dimension of matrix F is $\xi \times \omega$, and its row sum ξ can be computed by

$$\xi = \beta * \frac{\omega}{\text{Pr}(t)}, \quad \text{Pr}(t) \leq 1, \quad (21)$$

where $\text{Pr}(t)$ is the probability that blocks arrive at destination from the infrastructure in time t ; it can be designated by the VANET administrator according to the APs density. β is a guaranteed factor, where $\beta \geq 2$.

Implementing recoding on the coding blocks is formalized as computing the product of matrixes. It is denoted as

$$Z(p) = F \cdot Y(p)^s = F \cdot (K \cdot X(p)^T)^s. \quad (22)$$

$Y(p)^s$ is the ω nonlinear rows in matrix $Y(p)$, and they are received and selected by root. The dimension of matrix $Z(p)$ is $\xi \times \upsilon$, and its element $z(p)_{ij}$ can be computed by

$$z(p)_{ij} = \sum_{m=1}^{\omega} f_{im} * y(p)_{mj}^s = \sum_{m=1}^{\omega} \text{Seq} f_{i*\omega+m} * y(p)_{mj}^s. \quad (23)$$

The process of recoding on root is described in Algorithm 2.

3.3. ToVehic Stage. The leaves on multicast tree are the APs that may send blocks to destination vehicles within the designated time T_{\max} . Their communication ranges cover the location where vehicles are passing or will pass. The blocks are immediately sent out from APs after being recoded completely. Their live time $t = T_{\max}$. The function and computation of live time are the same as the ToInfra stage. AP stops sending the recoded blocks of the same row when it receives the ACK message from vehicles and then returns the message to the root of multicast tree. AP also stops sending

when the ξ recoded blocks are sent completely but does not return the ACK message.

The vehicles buffer and send the blocks which $t \neq 0$. The destination vehicle will return the ACK message when it receives the ω nonlinear recoded blocks of the same row and then begins decoding. The method of reducing packet includes the detailed steps as the following.

(1) *Computing Coefficient Matrix.* Coefficient matrix consists of the factors which the ω blocks multiply. It is denoted as G . $(Z(p)^s)^\omega$ is the received ω local encoding vectors and can be given by

$$(Z(p)^s)^\omega = G \cdot X(p) = (F^s)^\omega \cdot (K^s)^\omega \cdot X(p). \quad (24)$$

$(F^s)^\omega$ is the ω rows of local encoding matrix in root. $(K^s)^\omega$ is the ω rows of local encoding matrix in vehicles. The number of rows is sent with the blocks. Their elements are generated from the designated pseudorandom sequence. Destination vehicle can generate the sequence in the same rule. The element g_{ij} in G can be computed by

$$\begin{aligned} g_{ij} &= \sum_{m=1}^{\omega} k_{im}^s * f_{mj}^s \\ &= \sum_{m=1}^{\omega} \text{seq} k_{(i+a)*\omega+m} * \text{seq} f_{(m+b)*\omega+j}. \end{aligned} \quad (25)$$

a and b are the values that the received number of rows subtract the number of row in $(F^s)^\omega$ and $(K^s)^\omega$, respectively.

(2) *Decoding the Recoding Blocks.* The designated row of $(F^s)^\omega$ is constant to a VANET. The row of $(K^s)^\omega$ is variable because it changes with the vehicle IDs of source and coding. We do not decode the $Z(p)$ to $X(p)$ directly but decode two blocks successively, because the constant matrix $(F^s)^\omega$ is easy to preprocess. The recoding blocks can be decoded by the below equation:

$$(Y(p)^s)^\omega = ((F^s)^\omega)^{-1} \cdot (Z(p)^s)^\omega. \quad (26)$$

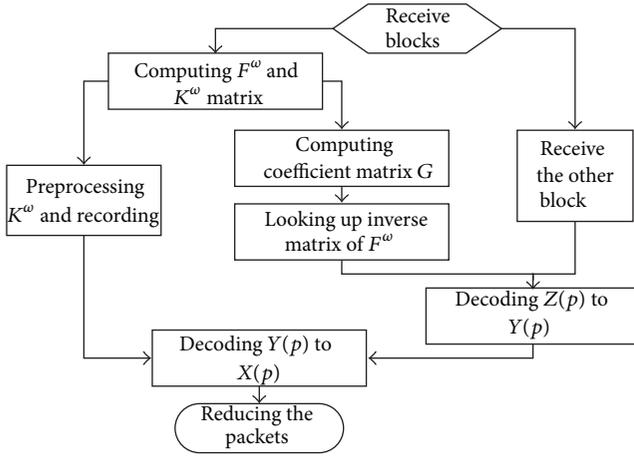


FIGURE 3: The process flowchart of destination vehicle on ToVehic stage.

Matrix inverse is not usually used in the decoding process because it is complex and can exhaust the computing resource. In our case, $(F^s)^\omega$ is the subset of constant matrix F . All subsets of F inverse can be precalculated in the mainframe computer. Their inverse matrixes are stored in vehicles as the table. The space occupied by the tables is $C_\xi^\omega * \omega^2$. This is storable to the vehicle. In this step, vehicles need not compute the inverse matrix, and they can find quickly the inverse matrix in the table according its rows sequence. So the decoding of the recoding blocks can be finished rapidly in (26).

(3) *Decoding the Coding Blocks.* Destination vehicles can decode $Y(p)$ to $X(p)$ in any decoding algorithm on the step. We improve the step in preprocessing $(K^s)^\omega$ matrix. To obtain $X(p)$ matrix, a variant of the Gaussian elimination is used. A typical Gaussian elimination or LU decomposition restricts us to wait until we decoded the recoding block and ω had the $\omega \times \omega$ coefficient matrix [33]. However, decoding operation is started early in our case, and $(K^s)^\omega$ is preprocessed as the first block arrives.

$Y(p)$ is confirmed of the sequences of the rows and the IDs of coding vehicles on the root. The first block can carry this information to the destination vehicle. Destination computes the $(K^s)^\omega$ matrix in advance. It preprocesses the $(K^s)^\omega$ in Gaussian elimination while receiving $Z(p)$ and records the processing steps. These steps contain the processing order of rows and the multiplying coefficients of every row. After $Z(p)$ is decoded to $Y(p)$, the vehicle decodes $Y(p)$ in the steps of preprocessing $(K^s)^\omega$. Processing objects are only the elements of $Y(p)$, and decoding $Y(p)$ is speed-up.

(4) *Reducing the Packets.* Destination composes the blocks of $X(p)$ into the packets. The packets are reduced completely.

The steps are formally shown in Figure 3.

4. Evaluation

In this section, we generate the movement of vehicles on the ONE simulator [34] and evaluate the performance of

our MCNC method in contrast with Epidemic, PROPHET routing, and generic Wi-Fi.

4.1. *Simulation Setup.* The ONE simulator uses the Helsinki city map and generates node movement and routing [35]. We extract the area with the range of $4000 \text{ m} \times 3000 \text{ m}$ from the map and configure two types of network nodes. Vehicle node moves on the map at the speed of 10 to 50 km/h and waits time of 10 to 120 seconds to stop communication after arriving to the destination. The vehicles choose random destinations on the map and move there following the shortest path. AP node is static and communicates with other APs in wired links. The radio range is configured as 200 m, and the MAC protocol is IEEE 802.11.

Epidemic replicates messages to all encountered nodes that do not have them yet. PROPHET only sends messages to the nodes that have the highest chances to deliver the messages to the destination. They are two typical routing protocols for ad hoc networks and have been implemented by the users of the ONE simulator as add-ons. In generic Wi-Fi, vehicles data must be delivered by APs in the radio range, and the routing capability of vehicles is disabled.

Two scenarios are generated in the evaluation. A sparse network with 30 vehicles simulates the rural area. A dense network with 150 vehicles simulates the urban area.

The other parameters of MCNC method are listed as follows. The TTL of the active message $L_{\max} = 2$ or 5. The live time of blocks among vehicles $T_{\max} = 300$ seconds. The dimension of encoding vector $\omega = 10$. The number of APs is variable.

4.2. *Performance Analysis.* The measured parameters on the simulator are the block loss rate, transmission delay, and overhead messages. Block loss rate [13] is the ratio of lost blocks to the sending blocks from sources. Transmission delay is the time of a block delivered from the source to all destinations. Overhead ratio [13] is the effective blocks as the percentage of all forwarding blocks among vehicles. They synthetically reflect the performance of MCNC. The simulation lasts for 30 minutes. The 10 sources in 2 scenarios are selected to generate flows for the evaluation. The average values of the 3 parameters are presented as the basis for analyzing their performance.

Figure 4 shows the average blocks loss rate in the sparse scenario. It is easy to see that as the number of APs increases, blocks loss rate decreases in the five data plots. When the number of APs is 80, the radios of APs cover the whole area and blocks loss rate is 0 for all the method. Epidemic has the less loss rate than the others, since it sends copies to all encountered vehicles and transmits successively in the greatest probability. Generic Wi-Fi is the most loss rate and cannot be used while APs only cover part of area. PROPHET also cannot delivery many blocks to the destinations in the sparse scenario. MCNC is close to Epidemic in blocks loss rate. When the cover area of APs is 1/5 (the number of APs is 16 in Figure 4), the blocks loss rate of MCNC is less than 0.2, and it can reach the target of transmission reliability. L_{\max} has some influence on the parameter. The values of the parameter

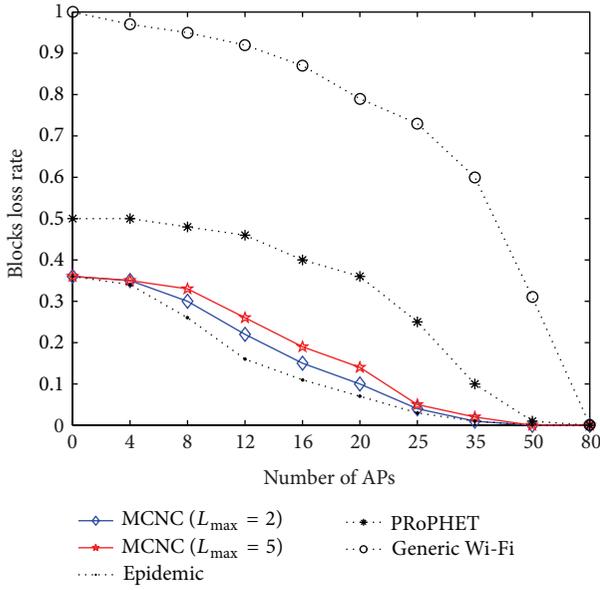


FIGURE 4: Blocks loss rate in the sparse scenario.

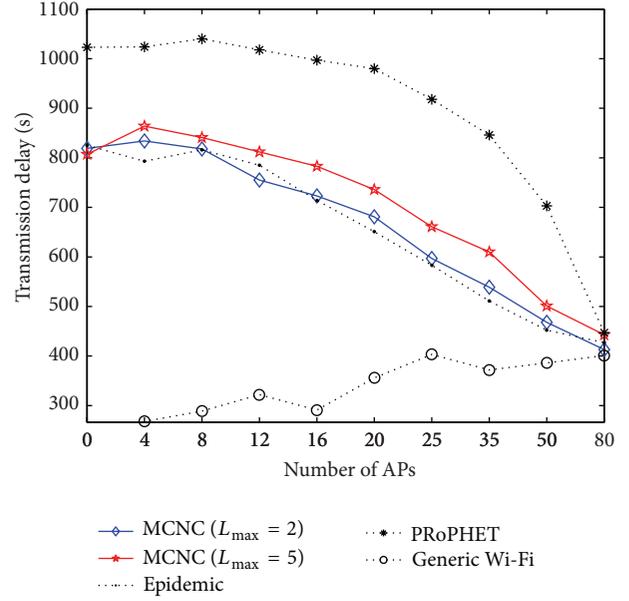


FIGURE 6: Transmission delay in the sparse scenario.

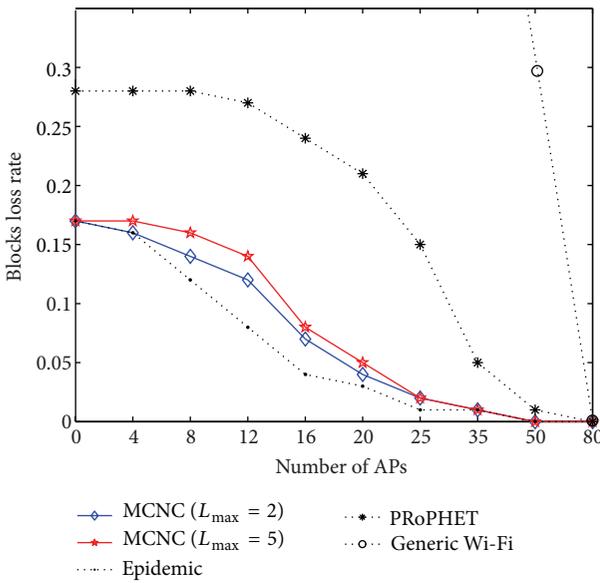


FIGURE 5: Blocks loss rate in the dense scenario.

will increase weakly while L_{max} is changed from 2 to 5. With the increase of hops after network coding and aggregating, the block loss rate only increases 0.04 when the number of APs is 16.

Figure 5 is similar to Figure 4. Generic Wi-Fi is unchangeable in different scenarios. Figure 5 only shows a part plot of generic Wi-Fi. The other methods all decrease their blocks loss rate in dense scenario. MCNC in the scenario is closer to Epidemic than the sparse scenario. When the cover area of APs is 1/5, the blocks loss rate of MCNC decreases to less than 0.08. The changes of L_{max} do not have strong influence on the transmission reliability of MCNC.

Figure 6 shows the average delay of blocks that are transmitted successively to destinations in the sparse scenario. Generic Wi-Fi only deliver few blocks to destinations while the number of APs is less than 20. Its delay changes randomly and does not associate with the number of APs and vehicles. The other methods all decrease the transmission delay with the increase of APs. PRoPHET only takes one of encountering vehicles as the next hop. Although the vehicle has the highest chance to destinations, it maybe moves more time to send block. So the delay of PRoPHET is more than the other methods. MCNC is also near to Epidemic in delay. Their delays are same in no AP setting. MCNC use APs to delivery blocks, but increase in little delay. When the cover area of APs is 1/5, the delay of MCNC is more 8% than Epidemic, and it can reach the target of transmission real-time. The changes of L_{max} have some influence on the delay of MCNC. The delay increases less than 10% when L_{max} is changed from 2 to 5.

The delays of four methods in dense scenario are shown in Figure 7, which is similar to Figure 6. They all decrease their delays in the dense scenario. Their delays are closer than the sparse scenario. When the number of APs is 16, the delay of MCNC is more than 6% Epidemic, PRoPHET is more than 22% Epidemic, and the delay increases less than 3% with the change of L_{max} from 2 to 5.

Figure 8 shows the average overhead ratios, which are the same in all test scenarios. Generic Wi-Fi does not generate overhead messages on any conditions, and its values are always least in four methods. Epidemic generates most overhead messages, which rise with the increase of APs, since AP also sends Epidemic message to the other vehicles. For this reason, PRoPHET also generates more overhead messages while setting more APs. MCNC has the same overhead ratio with Epidemic in no AP setting. With the increase of the number of APs, the overhead messages begin to decrease rapidly in MCNC. The rate of decrease is diminished after

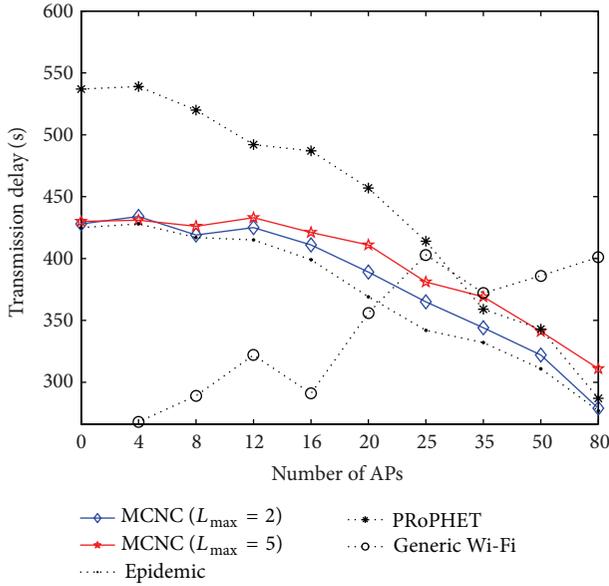


FIGURE 7: Transmission delay in the dense scenario.

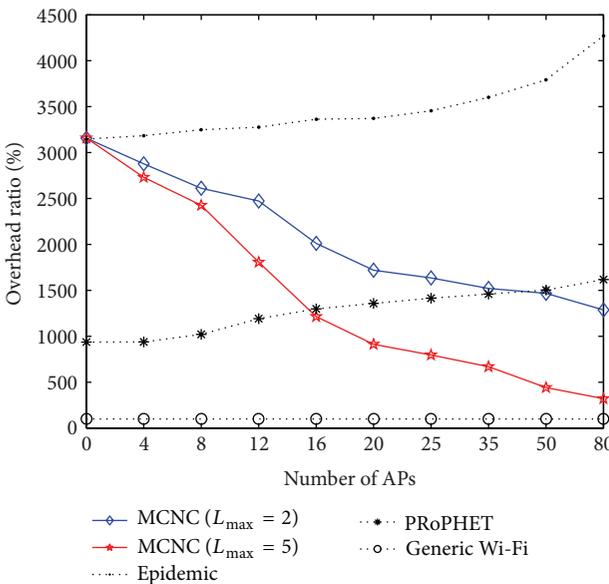


FIGURE 8: Overhead ratio of the four methods.

the cover area of the APs is more than 1/4 (the number of APs is 16). MCNC reduces the overhead message after network coding and aggregates the flows in multicast share trees on infrastructures. So its aggregating ability is obvious.

The changes of L_{\max} have huge influence on the overhead ratio of MCNC. The greater L_{\max} is, the less the overhead ratio is and the more the aggregating ability is. With L_{\max} changing from 2 to 5, the overhead ratio cuts down more than 1/2 after the cover area of the APs is more than 1/4. If L_{\max} and the number of APs satisfy the requirements, blocks can be coded and sent to APs after leaving the sources. The overhead ratio of MCNC ($L_{\max} = 5$) is 1200% when the cover area of the APs is 1/5. This value is the 1/3 of Epidemic and even less than

PRoPHET. So the 2/3 overhead messages are cut down, and the scalability of VANETs is improved.

Given all that, MCNC is able to achieve lower overhead ratios than both Epidemic and PRoPHET. Its blocks loss ratio and delay are tiny higher than Epidemic but lower than the other methods. It aggregates greater part of overhead messages using multicast and network coding. Installing APs in part of area can improve the performance of MCNC in any scenario. MCNC is appropriate for the areas that APs have been deployed but cannot cover entirely.

5. Conclusion

VANET can use APs as the infrastructures to improve its performance. Our works aggregate messages in multicast trees on infrastructures networks mainly consisted of APs and implement network coding on the vehicles that the hop to APs is less than L_{\max} . We divide data transmission into 3 stages: ToInfra, Belnfra, and ToVehic. Network coding, multicast aggregating, and decoding are implemented in the 3 stages, respectively. Data dissemination uses the generic routing algorithm for VANETs before network coding.

The experiments show that our MCNC is able to cut down 2/3 overhead messages of Epidemic when the cover area of the APs is 1/5. The costs of aggregations are a tiny improvement of delay. The blocks loss ratio and transmission delay of MCNC are also lower than the other methods. The performance of MCNC is similar in sparse scenario and dense scenario. In many places, Wi-Fi based APs can be deployed, but their radios cannot cover the whole area of vehicles movement, and MCNC will exert the promoting influence on the development of VANETs.

Acknowledgments

The work is supported in part by the National Science Foundation of China under Grant nos. 61202378, 61070169, and 61070170, University Science Research Project of Jiangsu Province under Grant no. 11KJB520017, the Natural Science Foundation of Jiangsu under Grant no. BK2011376, and Suzhou Application Foundation Research Project under Grant nos. SYG201238 and SYG201118.

References

- [1] F. Farnoud and S. Valaee, "Reliable broadcast of safety messages in vehicular Ad hoc networks," in *Proceedings of the IEEE INFOCOM 2009*, pp. 226–234, Rio de Janeiro, Brazil, April 2009.
- [2] J. Zhu, Y. Feng, and B. Liu, "PASS: parking-lot-assisted carpool over vehicular Ad hoc networks," *International Journal of Distributed Sensor Networks*, vol. 2013, Article ID 491756, 9 pages, 2013.
- [3] K. C. Lee, S. H. Lee, R. Cheung, U. Lee, and M. Gerla, "First experience with CarTorrent in a real vehicular Ad hoc network testbed," in *Proceedings of the Mobile Networking for Vehicular Environments (MOVE '07)*, pp. 109–114, Anchorage, Alaska, USA, May 2007.

- [4] U. Lee, J. Lee, J. Park, and M. Gerla, "FleaNet: a virtual market place on vehicular networks," *IEEE Transactions on Vehicular Technology*, vol. 59, no. 1, pp. 344–355, 2010.
- [5] G. Karagiannis, O. Altintas, E. Ekici et al., "Vehicular networking: a survey and tutorial on requirements, architectures, challenges, standards and solutions," *IEEE Communications Surveys and Tutorials*, vol. 13, no. 4, pp. 584–616, 2011.
- [6] M. J. Khabbaz, C. M. Assi, and W. F. Fawaz, "Disruption-tolerant networking: a comprehensive survey on recent developments and persisting challenges," *IEEE Communications Surveys and Tutorials*, vol. 14, no. 2, pp. 607–640, 2012.
- [7] S. Olariu and M. C. Weigle, *Vehicular Networks: From Theory to Practice*, CRC Press/Taylor & Francis, 2009.
- [8] Y. Wu, Y. Zhu, and B. Li, "Infrastructure-assisted routing in vehicular networks," in *Proceedings of the IEEE INFOCOM 2012*, pp. 1485–1493, Orlando, Fla, USA, March 2012.
- [9] J. Jaehoon, G. Shuo, G. Yu, H. Tian, and D. Du, "TBD: trajectory-based data forwarding for light-traffic vehicular networks," in *Proceedings of the 29th IEEE International Conference on Distributed Computing Systems Workshops (ICDCS '09)*, pp. 231–238, Montreal, Canada, June 2009.
- [10] D. Jiang and L. Delgrossi, "IEEE 802.11p: towards an international standard for wireless access in vehicular environments," in *Proceedings of the IEEE Vehicular Technology Conference-Spring (VTC '08)*, pp. 2036–2040, Singapore, May 2008.
- [11] J. Eriksson, L. Girod, B. Hull, R. Newton, S. Madden, and H. Balakrishnan, "The pothole patrol: using a mobile sensor network for road surface monitoring," in *Proceedings of the 6th International Conference on Mobile Systems, Applications, and Services*, pp. 29–39, June 2008.
- [12] Y. M. Zhu, X. M. Liu, M. L. Li, and Q. Zhang, "POVA: traffic light sensing with probe vehicles," in *Proceedings of the INFOCOM 2012*, pp. 2661–2665, 2012.
- [13] A. Palma, P. R. Pereira, and A. Casaca, "Multicast routing protocol for vehicular delay-tolerant networks," in *Proceedings of the IEEE 8th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob '12)*, pp. 753–760, Barcelona, Spain, October 2012.
- [14] J. Eriksson, H. Balakrishnan, and S. Madden, "Cabernet: vehicular content delivery using WiFi," in *Proceedings of the 14th Annual International Conference on Mobile Computing and Networking (MobiCom '08)*, pp. 199–210, September 2008.
- [15] V. Bychkovsky, B. Hull, A. Miu, H. Balakrishnan, and S. Madden, "A measurement study of vehicular internet access using in situ Wi-Fi networks," in *Proceedings of the 12th Annual International Conference on Mobile Computing and Networking (MOBICOM '06)*, pp. 50–61, September 2006.
- [16] N. Banerjee, M. D. Corner, and B. N. Levine, "An energy-efficient architecture for DTN throwboxes," in *Proceedings of the 26th IEEE International Conference on Computer Communications (INFOCOM '07)*, pp. 776–784, Anchorage, Alaska, USA, May 2007.
- [17] P. Rodriguez, R. Chakravorty, J. Chesterfield, I. Pratt, and S. Banerjee, "MAR: a commuter router infrastructure for the mobile internet," pp. 217–230, 2004.
- [18] J. Zhao, Y. Zhang, and G. Cao, "Data pouring and buffering on the road: a new data dissemination paradigm for vehicular Ad hoc networks," *IEEE Transactions on Vehicular Technology*, vol. 56, no. 6, part 1, pp. 3266–3277, 2007.
- [19] B. B. Chen and M. C. Chan, "MobTorrent: a framework for mobile internet access from vehicles," in *Proceedings of the IEEE INFOCOM 2009*, pp. 1404–1412, Rio de Janeiro, Brazil, April 2009.
- [20] I. Leontiadis, P. Costa, and C. Mascolo, "Extending access point connectivity through opportunistic routing in vehicular networks," in *Proceedings of the IEEE INFOCOM 2010*, pp. 1–5, San Diego, Calif, USA, March 2010.
- [21] J. Jeong, S. Guo, Y. Gu, T. He, and D. Du, "Trajectory-based statistical forwarding for multihop infrastructure-to-vehicle data delivery," *IEEE Transactions on Mobile Computing*, vol. 11, no. 10, pp. 1523–1537, 2012.
- [22] J. Jeong, S. Guo, Y. Gu, T. He, and D. H. C. Du, "Trajectory-based data forwarding for light-traffic vehicular Ad hoc networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, no. 5, pp. 743–757, 2011.
- [23] N. Liu, M. Liu, G. Chen, and J. Cao, "The sharing at roadside: vehicular content distribution using parked vehicles," in *Proceedings of the IEEE INFOCOM 2012*, pp. 2641–2645, Orlando, Fla, USA, March 2012.
- [24] N. Liu, M. Liu, W. Lou, G. Chen, and J. Cao, "PVA in VANETs: stopped cars are not silent," in *Proceedings of the IEEE INFOCOM 2011*, pp. 431–435, Shanghai, China, April 2011.
- [25] W. Vogels, R. Renesse, and K. Birman, "The power of epidemics: robust communication for large-scale distributed systems," *ACM SIGCOMM Computer Communication Review*, vol. 33, no. 1, pp. 131–135, 2003.
- [26] A. Lindgren, A. Doria, and O. Schelen, "Probabilistic routing in intermittently connected networks," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 7, no. 3, pp. 19–20, 2003.
- [27] U. Lee, J. Park, J. Yeh, G. Pau, and M. Gerla, "CodeTorrent: content distribution using network coding in VANET," in *Proceedings of the 1st International Workshop on Decentralized Resource Sharing in Mobile Computing and Networking (MobiShare '06)*, pp. 1–5, ACM, September 2006.
- [28] M. Li, Z. Yang, and W. Lou, "CodeOn: cooperative popular content distribution for vehicular networks using symbol level network coding," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 1, pp. 223–235, 2011.
- [29] M. Sathiamoorthy, A. G. Dimakis, B. Krishnamachari, and F. Bai, "Distributed storage codes reduce latency in vehicular networks," in *Proceedings of the IEEE INFOCOM 2012*, pp. 2646–2650, Orlando, Fla, USA, March 2012.
- [30] V. Palma, E. Mammi, A. M. Vegni, and A. Neri, "A fountain codes-based data dissemination technique in vehicular Ad-hoc networks," in *Proceedings of the 11th International Conference on ITS Telecommunications (ITST '11)*, pp. 750–755, August 2011.
- [31] S. Primak, V. Lyandres, O. Kaufman, and M. Klinger, "On the generation of correlated time series with a given probability density function," *Signal Processing*, vol. 72, no. 2, pp. 61–68, 1999.
- [32] Z. Pawlak, "Rough sets," *International Journal of Computer and Information Science*, vol. 11, no. 5, pp. 341–356, 1982.
- [33] T. Ho, M. Médard, R. Koetter et al., "A random linear network coding approach to multicast," *IEEE Transactions on Information Theory*, vol. 52, no. 10, pp. 4413–4430, 2006.
- [34] The ONE, <http://www.netlab.tkk.fi/tutkimus/dtn/theone/>.
- [35] A. Keränen, T. Kärkkäinen, and J. Ott, "Simulating mobility and DTNs with the ONE," *Journal of Communications*, vol. 5, no. 2, pp. 92–105, 2010.

