The following publication K. F. Chu, E. R. Magsino, I. W. -H. Ho and C. -K. Chau, "Index Coding of Point Cloud-Based Road Map Data for Autonomous Driving," 2017 IEEE 85th Vehicular Technology Conference (VTC Spring), Sydney, NSW, Australia, 2017 is available at https://doi.org/10.1109/VTCSpring.2017.8108280.

Index Coding of Point Cloud-based Road Map Data for Autonomous Driving

Kai Fung Chu*, Elmer R. Magsino*, Ivan Wang-Hei Ho* and Chi-Kin Chau†
*Department of Electronic and Information Engineering, The Hong Kong Polytechnic University
†Masdar Institute of Science and Technology
{kai-fung.chu, elmer.magsino}@connect.polyu.hk, ivanwh.ho@polyu.edu.hk, ckchau@masdar.ac.ae

ABSTRACT

Information exchange in a vehicular network between autonomous vehicles and the roadside infrastructure is important for improving road safety. These autonomous vehicles, equipped with a sensor suite, are capable of obtaining road map data that can be used to inform other vehicles and update the central road map repository through roadside units. The roadside infrastructure nodes act as local databases for distributing regional 3D road map data in form of point clouds to autonomous vehicles passing by. Since the vehicles might have various side information regarding the road network and traffic condition, minimizing the required number of transmissions to satisfy the demand of participating vehicles through network coding is an interesting research problem in road map data dissemination.

In this paper, we propose the Road Map Data Encoding and Dissemination System (REDS) and evaluate its performance in a four-way junction scenario. It is based on index coding for broadcasting road map data from a centrally-managed roadside node to vehicles. REDS uses the data availability and demand knowledge for encoding and transmitting 3D point cloud road map data from different road segments. The data availability information helps prevent the transmission of duplicated road map data and provides the sets of side information in the index coding problem, while the data demand information further defines the message transmission priority based on the data demand of different road segments. Simulation results indicate that REDS reduces the average number of transmissions and transmitted point cloud data size by around 30% when the data availability probability is about 0.5 under random mobility in all simulated scenarios when compared to the traditional broadcasting approach.

Keywords

Vehicular Ad-hoc Network, Index Coding, Autonomous Vehicle, 3D Point Cloud

1. INTRODUCTION

Autonomous vehicles are expected to accomplish an advanced variety of tasks such as perception, localization and navigation to travel safely and efficiently from a source location to its destination without the need of any human assistance.

In 2011, Google introduced a driverless car that can navigate on city streets offering complicated scenarios. It used a $70,000 3D-lidar for generating and displaying the environment in 3D view [1]. However, 3D data require lots of memory for storage and transmission. For example, the Microsoft Kinect generates about 331MB data per second for a 3D representation of its environment [2], needless to say LIDAR sensors that generate 3D point cloud data up to Gb/sec per node. Sharing such huge amount of data with other vehicles and the roadside infrastructure requires a huge bandwidth in the communications network.

In order to overcome the bandwidth drawback and facilitate road map data exchange while improving the driving experience, data compression, communications and cooperation among vehicles and roadside nodes need to be exploited to extend the sights of autonomous vehicles.

In [3], a content-centric communications system designed for sensor data sharing in autonomous driving, called CarSpeak, was presented. With CarSpeak, cars access the sensor data of other cars just like how it accesses its own sensors. Another possible way of how information sharing can be done between vehicles is through a cloud-assisted system such as Carcel [4]. Since autonomous vehicles cannot fully obtain all the needed information due to its limitations or blind spots in the environment, they can use the cloud to obtain road map data from other vehicles and the roadside infrastructure. In addition, content distribution in vehicular ad hoc networks (VANETs) can be enhanced by network coding. Preliminary [5] and follow-up [6] works discuss how network coding configurations such as resource constraints affect content distribution performance.

On the other hand, there is a large body of related work on cooperative localization and mapping. For example, research in [7], [8] studied real-time cooperative navigation and planning, mapping, exploration and object recognition in a 500m × 500m environment using 20 autonomous robots. Another cooperative Kalman filter-based localization and mapping scheme is reported in [9]. This was implemented by using two mobile robots situated in an uneven and unstructured environment.

In this paper, we present a novel Road Map Data Encoding and Dissemination System (REDS) for broadcasting various road map data on an n-way junction using the concept of index coding. Specifically, we consider the road map data exchange between roadside units (RSU) and the nearby vehicles in an n-way junction scenario. RSU can serve as local databases to gather and distribute road map data from and to all passing vehicles respectively. Index coding is performed at RSU to aggregate compressed road map data before broadcasting. The traditional broadcasting scheme broadcasts each message one by one, regardless whether the incoming vehicle already has that piece of road map data or not. However, if certain vehicles already have partial knowledge regarding the environment of the n-way junction, then, transmitting the encoded pieces of road map data will reduce the overall number of transmissions on average [10], [11]. Generally in an n-way junction, there are various number of vehicles, each possibly carrying map data regarding some road segments. Thus, from the point of view of the RSU, there are many possible encoded message combinations for transmission that can provide the least number of transmissions for the limited bandwidth condition. REDS determines which encoded message to be broadcasted according to the message priority based on the overall data demand and availability in the network.

This paper is organized as follows: Section 2 briefly discusses the point cloud data compression and index coding concepts used in this work. Section 3 presents the Road Map Data Encoding and Dissemination System (REDS) while Section 4 outlines the simulation setup and presents and discusses the important simulation findings. Finally, this paper is concluded in Section 5.

2. POINT CLOUD DATA & INDEX CODING

In this section, a background discussion of point cloud data, octree compression and index coding are presented.

2.1 3D Point Cloud Data

3D point cloud data (PCD) depict the objects in the environment as a collection of 3D points in the Cartesian coordinate system, (x_i, y_i, z_i), where i is the index of the point located in the 3D object [12]. Instead of having 2D pixels to represent the surrounding, *voxel* is used for object representation that can enhance object detection performance. Fig. 1 shows an example of the 3D PCD representation of the road environment captured by LIDAR mounted on the roof of a vehicle.

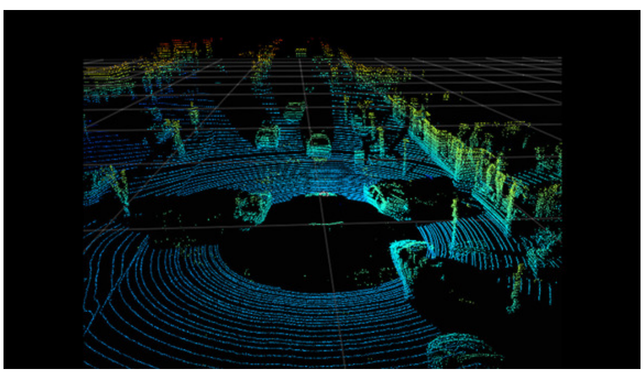

Figure 1: An example of 3D point cloud data collected from LIDAR mounted on a vehicle [13].

Before transmitting the PCD in the air, Octree compression is employed to reduce the data load. The structure of an Octree is shown in Fig. 2. It is a tree data structure suitably used for sparse 3D data where each node represents a cube or volume element (*voxel*). From the root, it is iteratively divided into eight children until a certain depth or level L is achieved [14] or if there are no more 3D PCD to be partitioned. If a cube or voxel contains a point or a set of points, it will be labeled "1", otherwise "0". A node labeled "1" can be further decomposed into eight more child nodes while there is no need to further expand a node labeled "0". Accordingly, the larger the height of the tree, the higher the resolution of the 3D object. The point cloud bitstream for transmission can be readily obtained by traversing the tree in a top-down and breadth-first manner.

There is a large number of works on point cloud compression. For example, Schnabel and Klein [15] introduced a progressive lossless compression method on point clouds represented by octree decomposition to achieve high compression rates. Works by Huang, et. al. [16], [17] focused on a generic scheme for encoding point clouds for compressing 3D objects with different attributes and topology.

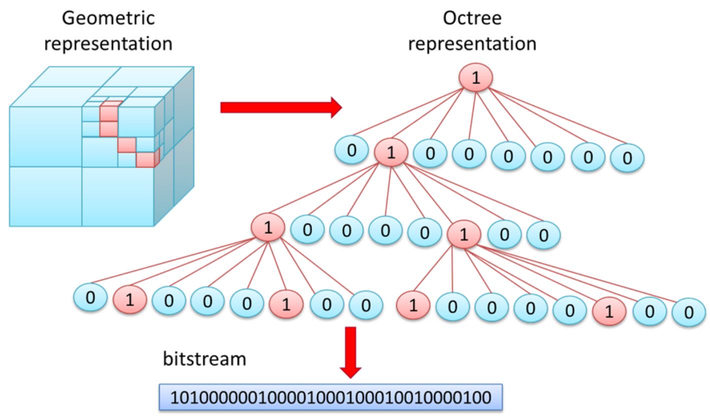

Figure 2: Octree representation of 3D point clouds.

2.2 Index Coding for Map Data Delivery

Consider the scenario given in Fig. 3 where two cars are approaching a traffic signal in an intersection. Each car is assumed to have obtained its respective static and dynamic road map data denoted by {A,a} and {B,b} respectively. Both cars are not within each other's transmission range, thus, direct data exchange is not possible. We also assume that the base station connected to the central server and roadside units (traffic light) perceives static map data A and B as global knowledge.

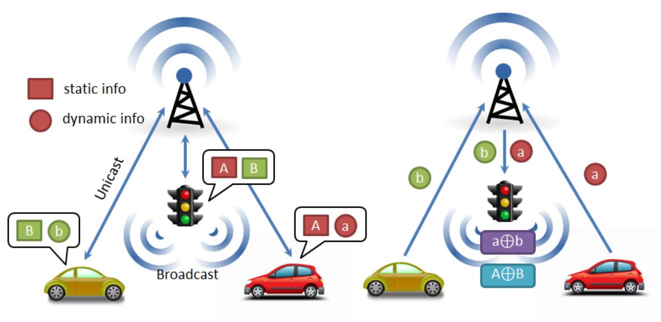

Figure 3: An illustration of how index coding works in practice with static and dynamic data.

Traditionally, in order for each car to have the road map data of the other road segment, two transmissions are needed from the RSU to the two cars. With network coding, the number of transmissions is reduced to one through broadcasting the encoded message A $\oplus$ B as illustrated in Fig. 3

(right). It was shown in [18] that network and index coding problems are equivalent in the general setting and that the index coding problem is a *simple* and *representative* network coding case.

For dynamic data regarding moving objects that the central server is not aware of, vehicles that serve as pervasive sensors can update the server through cellular network, and the same network coded broadcast can be done at the RSU. After receiving the broadcast message, for example, the left car performs: b ⊕ (a ⊕ b) to retrieve its desired message a.

3. ROAD MAP DATA ENCODING AND DISSEMINATION SYSTEM

In this section, we discuss how the Road Map Data Encoding and Dissemination System (REDS) performs the information exchange between vehicles and the infrastructure for reducing the number of transmissions needed in road map data dissemination.

3.1 Road Junction Scenario & System Model

We expand the scenario in Fig. 3 to a four-way junction as shown in Fig. 4. Each road segment is assumed to have a vehicle v_i, i = 1, 2, 3 or 4 moving towards the junction. A *message* is defined as an individual map data of a road segment or an encoded/combined map data of multiple road segments. The set of map data for all road segments is

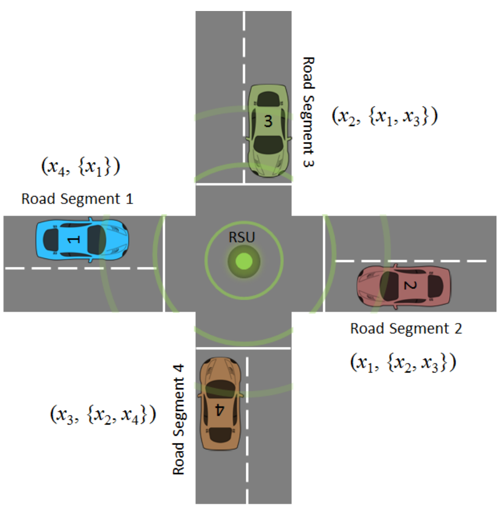

Figure 4: Index coding for a four-way junction scenario.

defined as $X = \{x_1, x_2, x_3, x_4\}$. Each vehicle $v_i = (x_i, S_i)$, where $x_i \in X$ denotes the desired message (map data for road segment i) of vehicle v_i and $S_i \subseteq X$ is the set of side information that vehicle v_i has.

Consider vehicle v_4 in Fig. 4. Vehicle v_4 desires the road map data for road segment 3, x_3 and currently has road map data on road segments 2 and 4, x_2 and x_4. Using the index coding technique, the following coding and decoding steps take place:

1. RSU broadcasts in time slot 1 $e_1 = x_1 \oplus x_2 \oplus x_3$. Vehicle v_4 will get $r_1 = (x_1 \oplus x_2 \oplus x_3) \oplus x_2 = x_1 \oplus x_3)$.
2. RSU broadcasts in time slot 2 $e_2 = x_1 \oplus x_4$. Vehicle v_4 will get $r_2 = (x_1 \oplus x_4) \oplus x_4 = x_1$.
3. From x_1 and r_1, Vehicle v_4 can get its desired message x_3. Note that other vehicles also obtained their desired messages through this process.

In this example, the number of transmissions is reduced from four to two via the index coding technique to satisfy the demand of all the participating vehicles. The RSU needs to determine which message to broadcast such that the minimum number of transmissions can be achieved. In a four-way junction, there are 15 possible message combinations that can be broadcasted by the RSU. In general, there are 2^n-1 possible messages for an n-way junction.

In this paper, we assume the most general case that every vehicle has a probability of going to any road segments (except the one it is from) when it arrives at the road junction, and each of them has a probability of carrying road map data of any road segments. These are stored respectively in the data demand and data availability databases (which will be introduced in the next subsection) for making coding deicisions, and other specific cases of data demand and availability can be considered based on this framework. Given this, REDS addresses how to assign broadcasting priority to different messages such that the overall offered load to the network for satisfying the participating vehicles can be significantly reduced.

3.2 Road Map Data Transmission Schemes

In this section, we consider three index coding based transmission schemes. The first one is a simple scheme that transmits encoded messages in a uniformly random manner. The second scheme aims to avoid transmission of duplicated messages, and the last scheme takes data demand and data availability into account.

3.2.1 Random Transmission Scheme

We define $\mathbf{P}$ as the message selection priority vector set that records the priority index of every message.

$$\mathbf{P} = \{P_1, P_2, \ldots, P_n\} \tag{1}$$

For an n-way junction, $\mathbf{P}$ is a collection of vectors of n column-vector blocks. The vector P_i contains the priority values of encoded messages of map data from i road segments. For example, P_3 contains the priority values for all the possible combinations of encoded map data from three road segments. We define that messages in P_3 have message length equal to three.

Therefore, P_i for i = 1, 2, ..., n is defined as follows:

$$P_i = \begin{bmatrix} p(m_{i1}) \\ p(m_{i2}) \\ \vdots \\ p(m_{iC(n,i)}) \end{bmatrix} \tag{2}$$

where

$$C(n,i) = \frac{n!}{i!(n-i)!} \tag{3}$$

and $p(m_{ij})$ is the priority index for transmitting message m_{ij} for j = 1, 2, ..., $C(n,i)$. Message m_{ij} is the j-th encoded message of map data from i road segments. We define the XOR operator (Φ) for multiple variables as:

$$\Phi_{i=1}^{n} x_i := x_1 \oplus x_2 \oplus \cdots \oplus x_n \tag{4}$$

To determine all the possible message combinations, m_{ij}, for each P_i block, Algorithm 1 is employed.

Algorithm 1 Identifying the corresponding encoded messages m_{ij} in each vector P_i

```
Initialize n, i, X={x_1, x_2, x_3, ..., x_n}
Initialize mx=[1, 2, ..., i]
                                  ▷ mx is an array of message indexes
m_i1={x_1, x_2, ..., x_i}
for j = 2 to C(n, i) do
    Find the largest β satisfying mx(β) ≠ (n − i + β)
    ▷ β is an array index of mx
    mx(β) = mx(β)+1
    for ctr = β+1 to i do
        mx(ctr) = mx(ctr − 1) + 1
    end for
    m_ij ← Use Eq. 4 on X following mx
end for
```

For an n-way junction, $\mathbf{P}$ is given below according to Algorithm 1.

$$\mathbf{P} = \left\{ \begin{bmatrix} p(x_1) \\ p(x_2) \\ \vdots \\ p(x_n) \end{bmatrix}, \begin{bmatrix} p(x_1 \oplus x_2) \\ p(x_1 \oplus x_3) \\ \vdots \\ p(x_{n-1} \oplus x_n) \end{bmatrix}, \ldots, p(\Phi_{i=1}^{n} x_i) \right\} \tag{5}$$

For the random transmission scheme (RTS), the priority of all messages is equal, i.e. all elements of $\mathbf{P} = \frac{1}{2^n-1}$ for the n-way junction case. Re-transmission of duplicated messages is not prevented in this scheme.

3.2.2 Condensed Transmission Scheme

For the condensed transmission scheme (CTS), the priority assignment of sending a message to approaching vehicles follows that of RTS, i.e., uniformly random. However, a condition must be met in CTS before a message is being re-transmitted. Also, transmission of redundant messages is avoided in this scheme.

The following rules are applied to avoid the transmission of redundant messages.

1. Re-transmission of broadcasted message is not allowed until a certain number of participating vehicles within the junction has been satisfied. We define that a vehicle is satisfied if it has obtained more than 80% of its desired road map data.
2. Redundant messages will not be transmitted. There are two possible cases:
   (a) Shorter messages will not be transmitted if it can be decoded from a longer message previously transmitted. For example, if $x_1 \oplus x_2$ and x_2 were previously transmitted, x_1 will not be transmitted as it can be decoded from the previously received messages.
   (b) Longer messages will not be transmitted if it can be obtained from shorter messages previously transmitted. For example, if x_1 and x_2 were previously transmitted, $x_1 \oplus x_2$ will not be transmitted.

3.2.3 Data Demand & Availability Transmission Scheme

For the data demand and availability transmission scheme (DDATS), priority assignment of sending a message is real-time according to the data demand and data availability databases. The databases can be established through prediction based on historical traffic data or beacon exchange between vehicles and RSUs. The Data Demand database, DD_{db}, stores the probability of a certain vehicle going from one road segment to another while the Data Availability database, DA_{db}, records whether a particular piece of map data is carried by a vehicle or not.

(6) illustrates the DD_{db} database for an n-way junction. Note that the main diagonal is zero signifying that a vehicle is not making a U-turn.

$$DD_{db} = \begin{bmatrix} 0 & \delta_{12} & \cdots & \delta_{1(n-1)} & \delta_{1n} \\ \delta_{21} & 0 & \cdots & \delta_{2(n-1)} & \delta_{2n} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \delta_{(n-1)1} & \delta_{(n-1)2} & \cdots & 0 & \delta_{(n-1)n} \\ \delta_{n1} & \delta_{n2} & \cdots & \delta_{n(n-1)} & 0 \end{bmatrix} \tag{6}$$

where δ_{vk} defines the probability of vehicle v going to road segment k. $\delta_k = \sum_{v=1}^{n} \delta_{vk}$ is the sum of all vehicles' δ_{vk} values for selecting road segment k. v = 1, 2, ..., n and k = 1, 2, ..., n.

Similarly, the DA_{db} database for an n-way junction is given in (7). When a vehicle v carries map data of a road segment k, x_k, its corresponding α_{vk}=1, else α_{vk}=−1.

$$DA_{db} = \begin{bmatrix} \alpha_{11} & \alpha_{12} & \cdots & \alpha_{1(n-1)} & \alpha_{1n} \\ \alpha_{21} & \alpha_{22} & \cdots & \alpha_{2(n-1)} & \alpha_{2n} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \alpha_{n1} & \alpha_{n2} & \cdots & \alpha_{n(n-1)} & \alpha_{nn} \end{bmatrix} \tag{7}$$

$\alpha_k = \sum_{v=1}^{n} \alpha_{vk}$ is the sum of all vehicles' α_{vk} for road segment k. This signifies the availability of a certain piece of road map data among all passing vehicles. If α_k is positive, it means that the majority of the passing vehicles is carrying the map data of road segment k. If α_k is negative, it implies that majority of the passing vehicles does not carry the map data of road segment k. Finally, if α_k is equal to zero, it suggests that the number of cars with and without road segment k's map data is equal. By default, if the RSU does not receive anything from a vehicle regarding its data availability, its α_{vk} is automatically set to −1.

We then define the data demand and data availability priority vector sets, $\mathbf{D}$ and $\mathbf{A}$, respectively, which are derived from the databases. Note that the structure of $\mathbf{D}$ and $\mathbf{A}$ are the same as that of $\mathbf{P}$, as the indexes in $\mathbf{D}$ and $\mathbf{A}$ will be summarized for the overall priority indexes in $\mathbf{P}$ finally.

$$\mathbf{D} = \{D_1, D_2, \ldots, D_n\} \tag{8}$$

where

$$D_i = \begin{bmatrix} d_{i1} \\ d_{i2} \\ \vdots \\ d_{iC(n,i)} \end{bmatrix} \tag{9}$$

d_{ij} is the data demand index for message m_{ij}, which denotes the likeliness of vehicles proceeding to a certain subset of i road segments.

For an n-way road junction, each D_i block in the vector set $\mathbf{D}$ is as follows.

$$\mathbf{D} = \left\{ \begin{bmatrix} \delta_1 \\ \delta_2 \\ \vdots \\ \delta_n \end{bmatrix}, \begin{bmatrix} \sqrt{\delta_1 * \delta_2} \\ \sqrt{\delta_1 * \delta_3} \\ \vdots \\ \sqrt{\delta_{n-1} * \delta_n} \end{bmatrix}, \ldots, \sqrt[n]{\prod_{i=1}^{n} \delta_i} \right\} \tag{10}$$

Similarly, the data availability priority vector set is defined as:

$$\mathbf{A} = \{A_1, A_2, \ldots, A_n\} \tag{11}$$

where

$$A_i = \begin{bmatrix} \psi_{i1}\text{sgn}(a_{i1}) \sqrt[i]{|a_{i1}|} \\ \psi_{i2}\text{sgn}(a_{i2}) \sqrt[i]{|a_{i2}|} \\ \vdots \\ \psi_{iC(n,i)}\text{sgn}(a_{iC(n,i)}) \sqrt[i]{|a_{iC(n,i)}|} \end{bmatrix} \tag{12}$$

and

$$\psi_{ij} = \begin{cases} +1 & \text{if the number of } (\alpha_k < 0) \text{ is odd and } \geq 3 \\ -1 & \text{otherwise} \end{cases}$$

$\psi_{ij}\text{sgn}(a_{ij}) \sqrt[i]{|a_{ij}|}$ is the data availability index of message m_{ij} (where a_{ij} is the product of α_k term(s) as shown in (13) and ψ_{ij} is for sign adjustment so that more demanded messages will be reflected with positive values), it denotes how unlikely an encoded message is being carried by the participating vehicles. Hence, the value of the index will be higher if any of the following happens:

1. For $i = 1$ (or messages of length 1), most vehicles do not have that piece of message.
2. For $i > 1$ (or message of length greater than 1), only one message component is missing from that piece of encoded message of longer length.

For an n-way junction, $\mathbf{A}$ is represented as follows.

$$\mathbf{A} = \left\{ \begin{matrix} \begin{bmatrix} -\text{sgn}(\alpha_1)|\alpha_1| \\ -\text{sgn}(\alpha_2)|\alpha_2| \\ \vdots \\ -\text{sgn}(\alpha_n)|\alpha_n| \end{bmatrix}, \begin{bmatrix} -\text{sgn}(\alpha_1\alpha_2)\sqrt{|\alpha_1\alpha_2|} \\ -\text{sgn}(\alpha_1\alpha_3)\sqrt{|\alpha_1\alpha_3|} \\ \vdots \\ -\text{sgn}(\alpha_{n-1}\alpha_n)\sqrt{|\alpha_{n-1}\alpha_n|} \end{bmatrix}, \\ \ldots, \psi_1 \text{sgn}(\prod_{i=1}^{n} \alpha_i) \sqrt[n]{|\prod_{i=1}^{n} \alpha_i|} \end{matrix} \right\} \tag{13}$$

After obtaining all the coefficients in $\mathbf{D}$ and $\mathbf{A}$, the priority index for every message $p(m_{ij})$, and the overall message selection priority vector set $\mathbf{P}$ can be obtained.

$$p(m_{ij}) = f(d_{ij}) f(\psi_{ij}\text{sgn}(a_{ij}) \sqrt[i]{|a_{ij}|}) \tag{14}$$

where

$$f(u) = \frac{u - \min(u)}{\max(u) - \min(u)} + 1 \tag{15}$$

is a scaling function with output value between 1 and 2.

Algorithm 2 illustrates how to determine the message to be transmitted based on the priority indexes in $\mathbf{P}$. In general, the message with the maximum priority index is transmitted in every time slot. After transmission, its priority will be reset until another round of transmission.

In DDATS, a message with larger index value in $\mathbf{D}$ and $\mathbf{A}$ is more likely to be transmitted. If in case two or more messages come up with the same priority, one of them will be chosen to transmit randomly. In addition, the rules stated in CTS are also applied in DDATS to avoid transmission of redundant messages.

Algorithm 2 Message selection in DDATS

```
Determine length(P)
while length(P) ≠ 0 do
    Determine index/indexes of the maximum element(s) of P, index_n
    if length(index_n) > 1 then
        Select a message index from index_n to be sent based on a uniform random number
    end if
    Send message with index_n
    Set P(index_n) = 0
    length(P) = length(P) − 1
end while
```

4. NUMERICAL RESULTS AND DISCUSSION

In this section, we present extensive simulation results from different scenarios in a four-way junction, i.e., n=4. In general, we come up with four different scenarios through varying the data demand and data availability patterns.

1. In DDATS, we considered three mobility models for vehicles to select which road segment to proceed.
   (a) Manhattan Mobility model (MM) [19] with uniform path selection rules for all vehicles;
   (b) Non-uniformly Random Mobility model (RMM);
   (c) Deterministic Mobility model (DMM)
      i. Case 1: Majority of the vehicles wants to proceed to a particular road segment
      ii. Case 2: Half of the vehicles wants to proceed to a particular road segment.
2. For all transmission schemes, each vehicle is assigned with a Data Availability Probability (DAP) for determining the probability of carrying a particular piece of road map data. The DAP matrix, $\mathbf{\Xi}$, that summarizes the data availability of vehicles is as follows.

$$\mathbf{\Xi} = \begin{bmatrix} \xi_{11} & \xi_{12} & \xi_{13} & \xi_{14} \\ \xi_{21} & \xi_{22} & \xi_{23} & \xi_{24} \\ \xi_{31} & \xi_{32} & \xi_{33} & \xi_{34} \\ \xi_{41} & \xi_{42} & \xi_{43} & \xi_{44} \end{bmatrix} \tag{16}$$

where ξ_{vk} denotes the probability that $\alpha_{vk} = 1$ or vehicle v carries map data of road segment k.

By considering different patterns in the DAP, the four scenarios investigated are given below.

1. Scenario 1: The probability values in (16) are all randomly generated.
2. Scenario 2: $\xi_{vk} = \frac{1}{4}$, for $v = 3$ and 4. Otherwise, the probability is randomly generated. This represents the case that certain vehicles got a fixed probability of having all the road map data.
3. Scenario 3: $\xi_{vk} = \frac{1}{4}$, for $k = 3$ and 4. Otherwise, the probability is randomly generated. This represents the case that data for certain road segments is known by all vehicles with fixed probability.
4. Scenario 4: $\xi_{vk} = 1$, for $v = k$ and the probability is randomly generated for $v \neq k$. This represents the case

that vehicles are equipped with sensors so that they acquire data for the road segments they have traveled.

The simulation terminates when all the participating vehicles are satisfied. We define here that a vehicle is satisfied if 80% of its data demand is achieved. E.g., if there is a 50% chance that a vehicle demands for the data of road segment 1, 35% for that of road segment 2, and 15% for that of road segment 3, then the vehicle is satisfied even if it receives data regarding road segments 1 and 2 only, since 50% + 35% > 80%. In order to make the simulation more realistic, several sets of 3D point cloud data are collected within the campus (see Fig. 5) to evaluate the overall performance of point cloud coding. These point cloud data are processed as discussed in Section 2.1. Their attributes (e.g., number of points, data size, etc.) are summarized in Table 1.

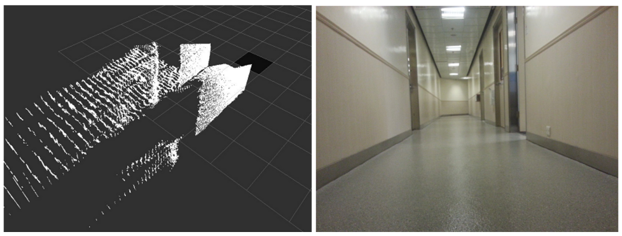

Figure 5: 3D PCD representation (left) and the corresponding photo (right) of a corridor in campus.

There are 100,000 runs for every simulated scenario. We compute the average number of transmissions for each map data. This, together with the data size of each encoded map data, gives the average transmitted data size. The simplest transmission scheme that guarantees the satisfaction of all vehicles is to broadcast the data for the four road segments one by one. This method corresponds to a total transmitted data size of 168.24KB (= 39.75 + 40.42 + 37.4 + 50.67). We set this as our benchmark for comparison and evaluation of the three transmission schemes presented.

Table 1: Attributes of point cloud data used in the simulation.

Map Data	Number of Points	Original Size (KB)	Compressed size (KB)	Compression Ratio (%)
x_1	88340	1380	39.75	34.72
x_2	88761	1387	40.42	34.31
x_3	85200	1331	37.4	35.59
x_4	109041	1704	50.67	33.63
$x_1 \oplus x_2$	96867	1514	46.99	32.22
$x_1 \oplus x_3$	97681	1526	46.02	33.16
$x_1 \oplus x_4$	128098	2002	59.93	33.41
$x_2 \oplus x_3$	25900	405	17.19	23.56
$x_2 \oplus x_4$	47106	736	28.48	25.84
$x_3 \oplus x_4$	49627	775	28.51	27.18
$x_1 \oplus x_2 \oplus x_3$	91475	1429	46.69	30.61
$x_1 \oplus x_2 \oplus x_4$	111389	1740	57.26	30.39
$x_1 \oplus x_3 \oplus x_4$	108704	1699	55.37	30.68
$x_2 \oplus x_3 \oplus x_4$	103249	1613	50.97	31.65
$x_1 \oplus x_2 \oplus x_3 \oplus x_4$	100614	1572	53.2	29.55

Fig. 6 compares the performance of the transmission schemes against the benchmark value in Scenario 1. For all three schemes, we can see that when the passing vehicles have more map data beforehand, less amount of data

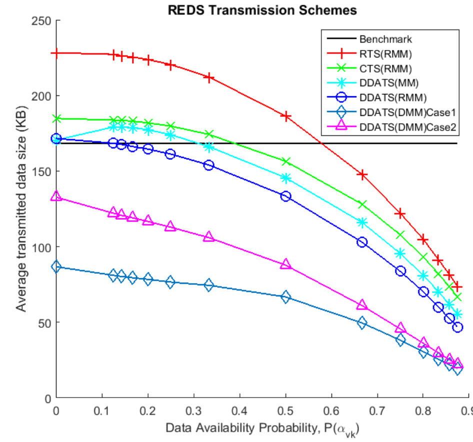

Figure 6: Average transmitted data size against data availability probability for the three transmission schemes using the three mobility models in Scenario 1.

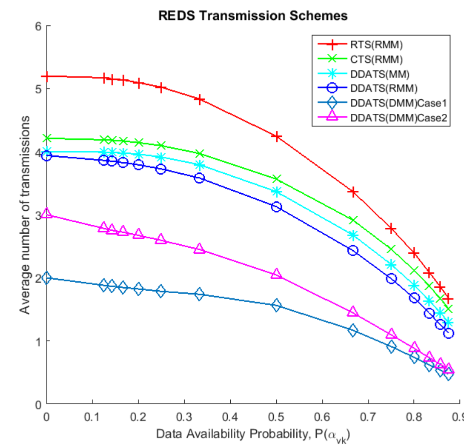

Figure 7: Average number of transmissions against data availability probability for the three transmission schemes using the three mobility models in Scenario 1.

is required to be broadcasted to satisfy the vehicles. This translates to fewer number of transmissions, as shown in Fig. 7. However, if passing vehicles carry too few road map data, (e.g., DAP < 0.3) both RTS and CTS send more data than the benchmark. On the other hand, DDATS peforms better than the benchmark even if the vehicles carry very little prior map data (e.g., for DAP ≈ 0.2).

Fig. 6 also shows the effect of different data demand/mobility patterns on the performance of DDATS. For this investigation, we use the Manhattan Model (MM) and compare it to the non-uniformly random mobility (RMM) and deterministic mobility (DMM) models. In the MM model, the vehicles' probabilities of going straight is 0.5 and turning left or right is 0.25. Given these, δ_k's for all road segments equal one. In RMM, the data demand of every vehicle for every road segment is randomly generated, hence, the δ_k's are not always equal to one and certain road segment can have a higher data demand over others. For the two cases considered in DMM, we have $\delta_1 = 3, \delta_2 = 1$ and $\delta_3 = \delta_4 = 0$ for Case 1 and $\delta_1 = 2$, $\delta_2 = \delta_3 = 1$ and $\delta_4 = 0$ for Case 2.

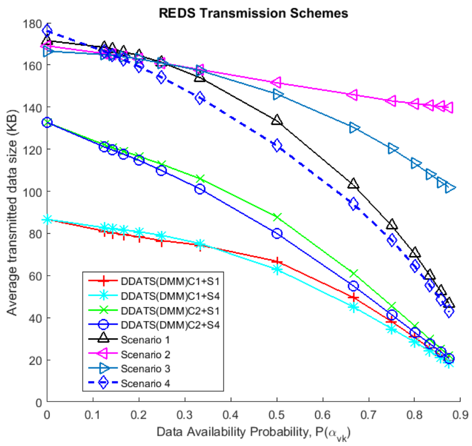

Figure 8: Performance of DDATS(RMM) and DDATS(DMM) under various Scenarios.

We simulate Scenarios 2 to 4 using DDATS(RMM). The results are shown in Fig. 8. The same trend is still exhibited by DDATS(RMM) for all simulated scenarios. At very low values of DAP (DAP < 0.15), DDATS(RMM) under Scenario 3 performs the best since certain map data is known by all vehicles in this scenario. We can also see that Scenario 4: vehicles with sensors performs the best with the least average number of transmissions and transmitted data size, which demonstrates the suitability of the index coding scheme for autonomous vehicles with heterogeneous sensors.

The reduction in transmitted data size in Scenario 2 is not as significant as that in Scenario 3. This suggests that DDATS performs better when certain map data are known by all vehicles rather than having certain vehicles carrying certain amount of the global map data initially.

Finally, we simulate the effects of deterministic mobility (e.g., the exact directions of the vehicles are known to the RSUs through beacon exchange) with random road map data and with sensors. The results are also shown in Fig. 8. It again demonstrates the importance of communications between the vehicles and RSUs (so that map data for the driving paths can be downloaded in advance) as well as the sensing capabilities of vehicles in the index-coded road map data dissemination problem.

5. CONCLUSION

In this work, a Road Map Data Encoding and Dissemination System (REDS) has been proposed and evaluated under various scenarios. REDS, based on index coding, jointly considers the availability of side information at vehicles as well as their data demand. This reduces the network load, and improves the overall data dissemination efficiency. Knowledge of the data demand helps suppress the transmission of low-demanded data while data availability information helps limit the transmission of duplicated or unnecessary messages. Real-world 3D point cloud data compressed according to the octree data structure are employed in the simulation study. Our simulation results indicate that the proposed system can achieve a much lower average number of transmissions and required bandwidth (around 30% reduction with a data availability probability of 0.5) compared to the traditional broadcasting approach.

Our future work will focus on expanding the scenario to multiple n-way junctions to determine the scalability of the system. Empirical mobility traces will also be included to determine the data demand and availability in a city-wide scenario for a more realistic evaluation.

6. REFERENCES

[1] A. Hard, "Pioneer is using laserdisc tech to help self-driving cars see," Sep 2015, retrieved July 10, 2016. [Online]. Available: http://www.digitaltrends.com/cars/pioneer-lidar-for-self-driving-cars-laserdisc-tech-news/

[2] R. A. El-laithy, J. Huang, and M. Yeh, "Study on the use of microsoft kinect for robotics applications," in *IEEE/ION PLANS*, 2012.

[3] S. Kumar, L. Shi, N. Ahmed, S. Gil, D. Katabi, and D. Rus, "Carspeak: a content-centric network for autonomous driving," in *ACM SIGCOMM*, 2012.

[4] S. Kumar, S. Gollakota, and D. Katabi, "A cloud-assisted design for autonomous driving," in *MCC Workshop*, 2012.

[5] U. Lee, J.-S. Park, J. Yeh, G. Pau, and M. Gerla, "Code torrent: content distribution using network coding in vanet," in *Mobishare*, 2006.

[6] S.-H. Lee, U. Lee, K.-W. Lee, and M. Gerla, "Content distribution in vanets using network coding: the effect of disk i/o and processing o/h," in *IEEE SECON*, 2008.

[7] A. Boeing, T. Braunl, A. M. R. Reid, and K. Vinsen, "Cooperative multi-robot navigation and mapping of unknown terrain," in *IEEE RAM*, 2011.

[8] R. Reid, A. Cann, C. Meiklejohn, L. Poli, A. Boeing, and T. Braunl, "Cooperative multi-robot navigation, exploration, mapping and object detection with ros," in *IEEE Intelligent Vehicles Symposium (IV)*, 2013.

[9] R. Madhavan, K. Fregene, and L. E. Parker, "Distributed cooperative outdoor multirobot localization and mapping," *Autonomous Robots*, vol. 17, no. 1, pp. 23–39, 2004.

[10] S. El Rouayheb, A. Sprintson, and C. Georghiades, "On the index coding problem and its relation to network coding and matroid theory," *IEEE Transactions on Information Theory*, vol. 56, no. 7, pp. 3187–3195, 2010.

[11] Z. Bar-Yossef, Y. Birk, T. Jayram, and T. Kol, "Index coding with side information," *IEEE Transactions on Information Theory*, vol. 57, no. 3, pp. 1479–1494, 2011.

[12] R. B. Rusu, Z. C. Marton, N. Blodow, M. Dolha, and M. Beetz, "Towards 3d point cloud based object maps for household environments," *Robotics and Autonomous Systems*, vol. 56, no. 11, pp. 927–941, 2008.

[13] M. Williams, "Driverless cars yield to reality: It's a long road ahead," Jul 2013, retrieved Aug 21, 2016. [Online]. Available: http://www.pcworld.com/article/2043878/driverless-cars-yield-to-reality-its-a-long-road-ahead.html

[14] J. Kammerl, N. Blodow, R. B. Rusu, S. Gedikli, M. Beetz, and E. Steinbach, "Real-time compression of point cloud streams," in *IEEE ICRA*. IEEE, 2012, pp. 778–785.

[15] R. Schnabel and R. Klein, "Octree-based point-cloud compression," *SPBG*, vol. 6, pp. 111–120, 2006.

[16] Y. Huang, J. Peng, C.-C. Kuo, and M. Gopi, "Octree-based progressive geometry coding of point clouds," *SPBG*, vol. 6, pp. 103–110, 2006.

[17] Y. Huang, J. Peng, C. C. Kuo, and M. Gopi, "A generic scheme for progressive point cloud coding," *IEEE Transactions on Visualization and Computer Graphics*, vol. 14, no. 2, pp. 440–453, 2008.

[18] M. Effros, S. E. Rouayaheb, and M. Langberg, "An equivalence between network coding and index coding," *IEEE Transactions on Information Theory*, vol. 61, no. 5, pp. 2478–2487, 2015.

[19] F. Bai, N. Sadagopan, and A. Helmy, "Important: A framework to systematically analyze the impact of mobility on performance of routing protocols for adhoc networks," in *IEEE INFOCOM*, 2003.