This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2021.3059696, IEEE Transactions on Network and Service Management

DND: Driver Node Detection for Control Message Diffusion in Smart Transportations

Peizhuang Cong*, Yuchao Zhang*,‡, *Member, IEEE*,
Wendong Wang*, *Member, IEEE*, Ning Zhang§, *Member, IEEE*
*Beijing University of Posts and Telecommunications, §University of Windsor

***Abstract*—Along with the development of IoT and mobile edge computing in recent years, smart transportation holds great potential to improve road safety and efficiency. The network that carries smart transportation service is highly dynamic. Controllability has long been recognized as one of the fundamental properties of such temporal networks, which can provide valuable insights for the construction of new infrastructures, and thus is in urgent need to be explored. In this paper, under the smart transportation scenario, we first disclose the controllability problem in Internet of Vehicles (IoV), and then design DND (Driver Node Detection) algorithm based on Kalman's controllability rank condition to analyze the controllability and control message diffusion in such a dynamic temporal network. Moreover, we use the control message diffusion efficiency as a metric to assist in selecting suitable driver nodes. At last, we conduct a series of experiments to analyze the controllability of the IoV network, and the results show the effects of vehicle density, speed, coverage radius on network controllability, and the efficiency of the control message diffusion algorithm and its feedback effect on driver nodes selection. These insights are critical for varieties of applications in the future smart transportation.**

***Index Terms*—IoT, Dynamic Network, Internet of Vehicles, Network Controllability, Driver Nodes**

I. Introduction

In recent years, with the rapid development of Internet of Things (IoT) technology, massive smart devices can form a self-organizing network and access to the Internet, which lead to the formation of characteristic dedicated networks, such as the interconnection of vehicles and road side units (RSUs) forms the Internet of Vehicles (IoV). The communication of vehicles to vehicles (V2V), vehicles to road (V2R), vehicles to the Internet (V2I) and vehicles to human (V2H) adopts different mechanisms, e.g., dedicated short-range communications (DSRC) and cellular networking which result in the heterogeneity of IoV. The mobility of vehicles causes the connection time-varying, which makes the network topology dynamic. In the IoV, the network topology, infrastructure resources and business requirements are dynamic, which makes it extremely difficult to manage. As a matter of fact, IoV with dynamic network characteristics can be regarded as a complex dynamic network [1]–[7]. The controllability of the dynamic network was first proposed in modern control theory which means the system can transfer its state to any specified state within finite time according to a series of input control vectors [8]–[10]. Similarly, the controllability of the IoV refers to all vehicles and RSUs reach an expected state, such as the connection state or movement velocity, etc.

Software-Defined Network (SDN) is an emerging concept in network that permits the network administrator to manage the network abstraction through the lower-level functions. The distinctive features of SDN enables centralized control and global view of the network, and various control policies can be formulated via control plane. The combination of SDN and vehicular ad hoc network (SDN-VANET) for IoV can solve many existing problems, e.g., real-time recalculation and reallocation of network resources based on the network state, which will fully improve network resources utilization and achieve differentiated services to a great extent. The working model of SDN-VANET usually contains three phases, first phase is to collect network state information, such as the network topology state, the bandwidth of each link, the traffic situation, etc. The second phase is to analyze and process the collected data in software defined (SD)-controllers. The third phase is to diffuse the control message to each node after calculation and meet the controllability of the entire network ultimately.

At present, many scholars have made efforts in SDN-VANET. Chuan Lin et al. introduced the fog computing into the SDN-enabled vehicular network architecture [11], Muhammad Sohail et al. maintained social connections among vehicles via SDN cloud [12], and Jie Zhang et al. introduced SDN to provide supports for the centralized network and vehicle information management [13]. Most of the current works are focused on phases 1 and 2, while phase 3 on controllability and control message diffusion is not well studied. However, the process of diffusion of control messages to all nodes is particularly important, which will affect the efficiency, timeliness, and security of the network. Ineffective phase 3 will severely harm the works of phases 1 and 2, and affect the overall network controllability.

In this work, we design an algorithm based on the Kalman's controllability rank condition, which is equivalent to the Popov- Belevitch-Hautus (PBH) rank condition, to select suitable driver nodes to diffuse control messages to all nodes of the network quickly and efficiently. These driver nodes correspond to elements of the control matrix in the controllable network. The control signal is sent to driver nodes through RSUs or generated by driver nodes to make the network transform into a specified state. The DND algorithm designates the minimum driver nodes under the condition of controllability

‡Corresponding author: Yuchao Zhang (yczhang@bupt.edu.cn)

and takes the number of hops to the driver node as the priority to minimize forwarding nodes number in diffusion process. Furthermore, after considering driver nodes calculation characteristic and control message diffusion process, we optimize node indexing to minimize the message packet forwarded hops which will reduce the control time.

The remainder of this paper is organized as follows. Related works are reviewed in Section II. Section III presents the background and the problem definition. Section IV describes some variables and specific formulas of our modeling. Section V presents the results and analysis. Finally, the conclusions and future work are provided in Section VI.

II. Related Work

A. SDN-VANET

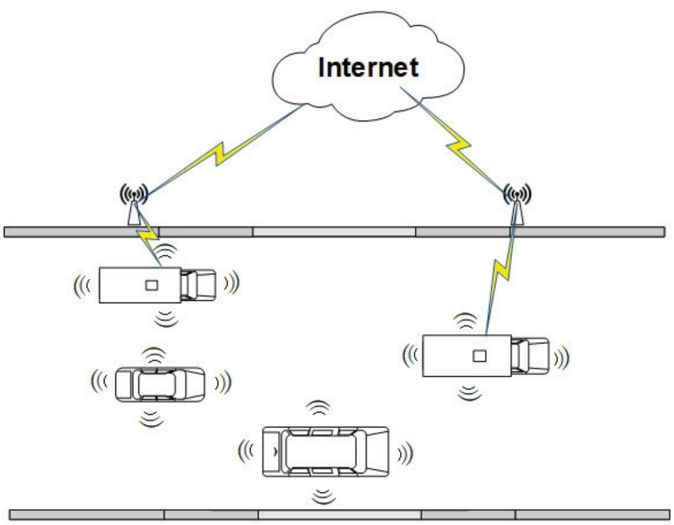


Fig. 1. Illustration of V2V and V2R communication.

The IoV develops rapidly with the progress of communication technology (e.g., Bluetooth, WIFI, microwave, infrared technology, short-range communication, cellular technology), and it can provide multimedia data sharing services, such as uploading/downloading of photos and audio/video, navigation, information collection, or real-time emergency warning [14]–[20]. As shown in the Fig.1, vehicles can share data or access the Internet through RSUs. Vehicle Ad Hoc Network refers to an open mobile Ad hoc network composed of communication between vehicles, fixed access points, and pedestrians in a traffic environment, whose goal is to build a self-organized, convenient, low-cost, and open communication network on the road. The flexibility and programmability of SDN can help meet the performance and management requirements of VANET, so SDN-VANET develops.

The characteristics of SDN-VANET make network resource management become a very difficult problem [21]. By designing an SDN-based network model, adding layers to the network, or changing the centralized computing model to distributed computing in mobile edge computing servers, intelligent flow control and effective resource management can be realized [11], [22]–[24]. There is no doubt that security is extremely important on the IoV, Weng et al. used broadcast encryption to control network resources dynamically and flexibly, rather than altert source codes of the controller or update permission lists with various degrees of granularity [25]. Sahil et al. used the elliptic curve cryptographic based authentication protocol to reduce computational complexity and identify potential intrusions in the network [26]. And Gao et al. combined blockchain technology to propose a trust-based model that curbs malicious activities in the network, and can help to relieve the pressure off the controller due to the ubiquitous processing that occurs [27]. The demand for delay or high bandwidth of the application in the Internet of vehicles leads to offload some tasks to the edge server. Modeling under the SDN controller optimizes the task offload according to the traffic classification and the load-balancing scheme can achieve superior performance on processing delay reduction by utilizing the edge servers' computation resources more efficiently [13], [28], [29].

However, the above work is focused on aforementioned phases 1 and 2. The phase that involves in the rapid distribution of control messages to achieve the controllability of the Internet of Vehicles are relatively lacking and should also be studied.

B. Data transmission in VANET

An efficient data transmission mechanism is critical in the network, many scholars and institutions have a long history of research and outstanding improvement in this area. For example, the original flooding algorithm is mainly used in route diffusion in traditional wired networks. The basic principle is that a router updates the packet sent from a neighbor router, and then forwards it to all neighbors except the sender. This guarantees the packet will reach the destination eventually as long as the destination is reachable.

In Ad hoc, Mehran et al. proposed virtual ad hoc routing protocol to introduce better security, lower routing overhead, and higher scalability [30]. And there are also lots of works on information dissemination or flooding algorithms in VANET [31], [32]. Priyashraba et al. developed a new protocol based on the characteristics of the scene, which obtains vehicle information over multiple hops at long distances, uses a causal position-based indirect communication method to allow RSUs to disseminate messages with an uncoupled communication, and leverages message redundancy to recover some lost messages during the transmission [33], [34]. It is necessary to use a suitable flooding algorithm to alleviate the broadcast storm problem, which there are currently algorithms based on probability, prediction, or concurrent transmission to achieve performance improvements concerning the achieved reachability, end-to-end delay, and the number of rebroadcasts, improving the overall efficiency of the network [35]–[37]. Florian Klingler et al. used a bloom filter to maintain 2-hop neighborship information and proposed a novel 2-hop broadcast algorithm, which significantly reduces the length of the beacon messages and keeps channel load and collision probability substantially lower than in the naive scheme [38].

C. Dynamic network

The network is composed of nodes and relations between all nodes. The rapid development of the Internet makes human beings enter the network era, such as all kinds of social networks [39]–[43]. The development of Industrial Internet forms such as large power grid and the Internet of Things which connects the whole world into a huge network. At present, network research has infiltrated into mathematics, life science, information science and many other fields, its fundamental goal is to find effective means to control network behavior and make it serve human beings. Lombardi first applied the classical control theory to the analysis of network control [44]. And points out here, in network science, if there is a directed edge between the node N1 and the node N2 in the directed network, the node N2 is usually said to be controlled by the node N1. In this paper, the network is abstracted into a linear time-invariant system, and the Kalman controllability criterion in the control theory is introduced into the research. The sufficient and necessary condition for the network to achieve complete controllability is that the controllability matrix reaches full rank, so the controllability problem of the network is transformed into the problem of calculating the rank of the matrix.

Liu et al. theoretically proved that the minimum driver nodes needed to control the entire network depends on the maximum matching in the network, in which all unmatching nodes are driver nodes of the network [45]. Thus, the problem of network structure controllability is transformed into a classical graph theory matching problem which reduces the time complexity of this kind of problem. However, this method can only be applied to directed networks or networks with unknown edge weights. Based on Popov-Belevitch-Hautus criterion, Zhengzhong et al. proved that the minimum number of control nodes required by the network is equal to the maximum geometric multiplicity of all eigenvalues of the network matrix, as for undirected networks, the minimum control nodes is the maximum algebraic multiplicity of all eigenvalues [46]. It reduces the computational complexity of related problems greatly.

Cornelius et al. presented a theoretical approach to describing the controllability of networks and proposed a way to change the state of some nodes to stabilize the network [47]–[50]. Xiao et al. designed a data downloading/uploading queuing mechanism to cooperate data sharing among peers to overcome the data dissymmetry in dynamic network context [51]. All papers and methods mentioned above have not solved the problem of controllability generated in dynamic networks well until now.

III. Problem definition

In modern cybernetics, controllability is used to describe linear systems. A linear system is controllable if its internal state can change to an expected state based on external inputs. Based on this theory, the dynamic vehicle network can be regarded as a kind of linear system, and it can reach the expected state, such as specified velocity or location, according to diffused control messages that originate from some key nodes we call driver nodes. There is a minimal drivers nodes to achieve network controllability, and we try to find them to make a highly dynamic network reach an expected state.

To find driver nodes and diffuse control messages quickly, we take vehicles of specified density travel along a section of road at a certain speed as an example. Each vehicle with the communication capability to exchange the required data with other vehicles within the communication radius[1]. If each vehicle is regarded as a node, and there is an edge between the two vehicles that can exchange data with each other, then at a certain point in time, all nodes and their connections can be abstracted into an undirected graph as shown in Fig.2[2].

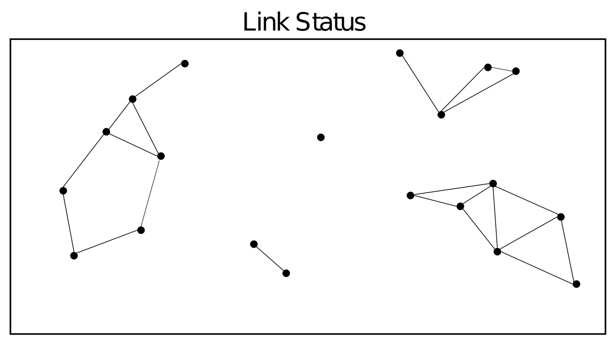


Fig. 2. The abstract graph of vehicle network.

The continuous movement of vehicles makes connection relations between vehicles dynamic, so it is necessary to calculate the connection relationship information in each small time period. The time interval for recalculating the connection relations is called refresh time. The refresh time is usually related to the velocity of vehicles. In order to ensure the controllability of the network, the greater the vehicle speed, the higher the refresh frequency. We set a so-called control time, and it means that the network of all vehicles in the entire area needs to reach the expected state during this time period. The control time includes at least one refresh time, and often more than one. During the control time, all vehicles connection status in the entire area will be updated after each refresh. We need not only the current connection state, but also the previous connection state. Finally, we obtain the required connection state within the control time and can abstract it as an undirected graph.

According to the undirected graph generated by these parameters, the required driver nodes can be calculated under

[1]Regularly, these data are peers' public states' information, such as velocity, acceleration, and direction, or control message generated by driver nodes, which don't involve private data.

[2]The connections of cars are not a state that needs to be maintained, but the ability to send/receive the diffused control messages to/from 'connected' neighbor cars.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2021.3059696, IEEE Transactions on Network and Service Management

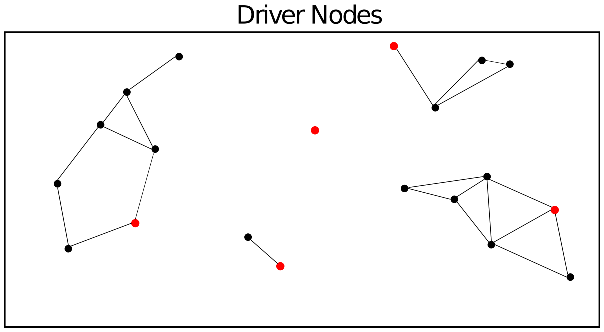


Fig. 3. The abstract graph of vehicle network with driver nodes.

the premise of the controllability of entire network. As shown in Fig.3, where the red point is the driver node. This will be explained in detail in Section IV.

In the area of interest, the density, velocity and radius of communication of the vehicle, the refresh time and control time which we set will affect the state of the entire network and the minimal number of driver nodes. We want to find the relationship between the number of minimum driver nodes and these parameters mentioned above.

As briefly mentioned here, drive node problem is different from placement problem and they can act as controller signaling commands [52]. For example, it could speed up traffic by sending driving signals via driver nodes along congested roads. As for the specific content of signals, driving rules will be the next work for further study.

IV. DND: Driver Node Algorithm

A. Driver node definitions

Driver nodes are the sources to diffuse control messages that received from external or generated by driver nodes internally to whole network. Then, a network with n nodes can be described as the following equation [53]:

$$x(t) = Ax + Bu \tag{1}$$

the state $x(t)$ should be a vector including some attributes of all vehicles in time t, such as position, speed, and so on. To simplify the theoretical proof in this section, we use a vector[n, 1]($[x_1, ..., x_n]$) to represent the state vector of all vehicles. $A[n, n]$ denotes the network's coupling matrix, in which a_{ij} denotes the weight of directed link from node j to i (for undirected networks, $a_{ij} = a_{ji}$), u is the controller vector with $u = (u_1, u_2, ..., u_m)^T$, and $B[n, m]$ is the control matrix. Any state of $x(0)$ can be adjusted to the initial state $x(o)$, which is $x(o) = [0, 0, ..., 0]$, that is:

$$x(o) = A^t x(0) + A^{t-1}Bu(0) + ... + ABu(t-2) + Bu(t-1) \tag{2}$$

If $x(0)$ can be adjusted to $x(o)$, then there must exist $u(0), u(1)..., u(t-1)$ makes the Eq.(3) true.

$$-A^t x(o) = A^{t-1}Bu(0) + ... + ABu(t-2) + Bu(t-1) \tag{3}$$

The Eq.(3) can be rewritten as Eq.(4).

$$-A^t x(o) = (A^{t-1}B \quad \dots \quad AB \quad B) \begin{pmatrix} u(0) \\ \vdots \\ u(t-2) \\ u(t-1) \end{pmatrix} \tag{4}$$

According to Cayley-Hamilton theorem and its inference, the condition for the solution of Eq.(4) is:

$$rank(A^{n-1}B \ ... \ AB \ B) = n \tag{5}$$

Similarly, Kalman rank condition is equivalent to the Popov-Belevitch-Hautus (PBH) rank condition, $rank(cI_N - A, B) = N$ is satisfied for any complex number c. And full control can be ensured iff any eigenvalue λ belonging to matrix A satisfies equation. Next, based on PBH, we prove the solution with the minimum number of driver nodes [46].

Using the nonsingular transformation $y = P^{-1}x$ and $Q = P^{-1}B$, Eq.(1) can be rewritten as Jordan form as $y = Jy + Qu$, where J is the Jordan matrix.

For an arbitrary Jordan matrix, $J = diag(J(\lambda_1), ..., J(\lambda_l))$, we have

$$\lambda_i I_N - J = diag(\lambda_i I_1 - J(\lambda_1), ..., \lambda_i I_l - J(\lambda_l)) \tag{6}$$

where the unit matrix $I_i(i = 1, 2, ..., l)$ is of the same order as $J(\lambda_i)$. If $\lambda_i \neq \lambda_j$, $\lambda_i I_j - J(\lambda_j)$ is a nonsingular matrix. In this case, rank deficiency can only appear in $\lambda_i I_i - J(\lambda_i)$. Using the fact $J(\lambda_i = diag(j_1, ..., j_{\mu(\lambda_i)}))$, where $\mu(\lambda_i)$ is the geometric multiplicity of λ_i and is equal to the number of basic Jordan block in $J(\lambda_i)$, we can conclude that $\lambda_i I_n - J$ has $\mu(\lambda_i)$ zero columns and $N - \mu(\lambda_i)$ independent columns. Thus $rank(\lambda_i I_n - J) = N - \mu(\lambda_i)$.

According to the exact controllability theory, driver node is determined by the maximum geometric multiplicity $\mu(\lambda^M)$ of the eigenvalue λ^M. Thus, the control matrix B to ensure full control should satisfy the condition by substituting λ^M for the complex number c, as:

$$rank(\lambda^M I_n - A, B) = n \tag{7}$$

The value of Eq.(7) is contributed by the number of linearly independent rows. In this regard, we implement elementary column transformation on the matrix $\lambda^M I_n - A$, which yields a set of linearly dependent rows that violate the full rank condition. The controllers located via B should be imposed on the identified rows to eliminate all linear correlations to ensure condition. The nodes corresponding to the linearly dependent rows are the driver nodes.

B. Driver nodes detection

1) Parameters And Variable Declarations: In this section, we formally define the driver nodes problem in vehicle network which aims to obtain the minimum number of driver nodes of the network. Table I defines the notations to be used in the problem definition.

TABLE I
NOTATION DEFINITIONS

Notation	Meaning
X	the length of the calculated range
Y	the width of the calculated range
N	the number of nodes in the calculated range
N_D	number of control nodes in the calculated range
P	device density in every Y*Y area of calculated range
R	communication radius of the device
(x_i, y_i)	coordinate value of the device
V_ix	the speed of device i in the X direction
V_iy	the speed of device i in the Y direction
T	control time
t	refresh time
A	0-1 adjacency matrix of the graph, $A_{ij} = 1$ means that devices i and j are connected by edges,and $A_{ij} = 0$ means that there is no edge connecting between devices i and j.

2) Modeling: Set a coordinate system for calculating range:XoY.

Vehicle nodes are randomly generated in the whole region according to the normal distribution probability of $X \sim N(P, 1)$, where the coordinates of the nodes are:

$$x_i = random[0, X) \tag{8}$$

$$y_i = random[0, Y) \tag{9}$$

Coordinate of nodes are updated in a unit interval as follows:

$$(x_i, y_i) = \begin{cases} x_i + V_i x, y_i + V_i y, & x_i + V_i x \le X, y_i + V_i y \le Y \\ X, y_i + V_i y, & x_i + V_i x > X, y_i + V_i y \le Y \\ x_i + V_i x, Y, & x_i + V_i x \le X, y_i + V_i y > Y \\ X, Y, & x_i + V_i x > X, y_i + V_i y > Y \end{cases} \tag{10}$$

Suppose the road is along the X-axis direction because the vehicle should adjust the direction when driving to the edge of the road (Y-axis direction) to prevent the collision, that is, reverse the velocity of the Y-axis direction. To simulate the entering and exiting areas of the vehicle while ensuring that the total number of observed vehicles remains unchanged, we stipulate that when the vehicle reaches the boundary of the x-axis, the velocity in the X-axis direction is reversed. So the node velocity is updated:

$$V_i x = \begin{cases} -V_i x, & x_i = X \ or \ x_i = 0 \\ V_i x, & 0 < x_i < X \end{cases} \tag{11}$$

$$V_i y = \begin{cases} -V_i y, & y_i = Y \ or \ y_i = 0 \\ V_i y, & 0 < y_i < Y \end{cases} \tag{12}$$

According to all nodes and their connection relationships, the adjacency matrix A can be obtained. And all element values of A comply with Eq.(13).

$$a_{ij} = \begin{cases} 1, & \| n_i, n_j \|^2 \le R \\ 0, & \| n_i, n_j \|^2 > R \end{cases} \tag{13}$$

The adjacency matrix after update is as follows:

$$a_{ij}^{t+1} = \begin{cases} 0, & a_{ij}^t = a_{ij} = 0 \\ 1, & a_{ij}^t \ne 0 \ or \ a_{ij} \ne 0 \end{cases} \tag{14}$$

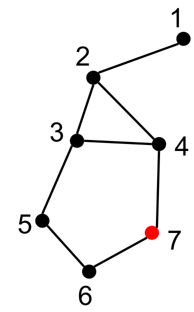


Fig. 4. Driver nodes of network

where $a_{ij}^t \in A^t$, which is the adjacency matrix calculated according to the nodes' state at current moment, and $a_{ij}^{t+1} \in A^{t+1}$,which is the state adjacency matrix at the previous moment.

The number of driver nodes is [46]:

$$N_D = \mu(\lambda^M) \tag{15}$$

It should be noted that A must be connected. If the graph in the current state is not a connected graph, it needs to calculate its connected subgraph graph separately, and add the results of each connected subgraph finally.

3) Abstraction of Topology: In this subsection, we introduce the process of calculating the minimum required driver nodes for the network based on adjacency matrix:

- *A=get_matrix(network_topology):* According to Eq.(13), the adjacency matrix A of $network\_topology$ can be calculated.
- λ*=get_eigenvectors(A):* After matrix operation, the eigenvalue vector of A is $\lambda = [\lambda_1, \lambda_2, ..., \lambda_n]^T$.
- λ^M*=get_max_geometric_multiplicity(*λ*):* Assuming that λ^M is eigenvalue corresponding to the maximum geometric multiplicity.
- Dn*=*$\mu(\lambda^M)$*:* The number of driver nodes is the multiplicity of λ_M, and which nodes are the driver nodes can be determined as follows.
- A' *= canonical_formed(*$A-\lambda^M I_n$*):* The matrix $A-\lambda^M I_n$ as Eq.(16), where I_n is the identity matrix. We perform the elementary column transformation on $A - \lambda^M I_n$ and to get the column canonical form, A'.

$$A-\lambda^M I_n = \begin{bmatrix} a_{11} - \lambda_i & \cdots & & & a_{1n} \\ & \ddots & & & \\ \vdots & & a_{ij} - \lambda_i & & \vdots \\ & & & \ddots & \\ a_{n1} & \cdots & & & a_{nn} - \lambda_i \end{bmatrix} \tag{16}$$

- DNs *= get_linearly_dependent_rows(*A'*):* All nodes that corresponding to the linearly dependent rows are the driver nodes of the network.

Given a topology with certain index as shown in the Fig. 4., after the above calculation, it can be concluded that node 7 (the red one) is the driver node.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2021.3059696, IEEE Transactions on Network and Service Management

C. *Diffusion of control message*

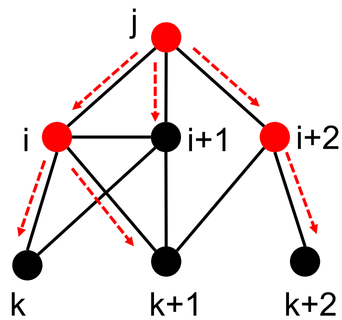


Fig. 5. Diffusion example.

1) DND Diffusion Algorithm: The driver node is the special node in the dynamic network, which can be used to access the edge server or the Internet. These nodes, as key nodes and initiators of instructions for controlling the state of the network, need to transmit control messages to all nodes of entire network. The most basic algorithm in a wireless network is all nodes will forward the packet directly after receiving a new control message, or discard the packet without processing when receiving a duplicate control message. This method is feasible in networks where degrees of nodes are generally small. However, when the degrees of connections are large, which is a common phenomenon in current scenarios, many redundant control message packets will be forwarded [54]. Therefore, it undoubtedly will increase the computational burden of all nodes and the transmission delay, which degrade the transmission performance of the entire network.

The topological structure of the entire network has been obtained when calculating driver nodes. Based on this feature, we can calculate the nodes that do not need to forward data packets according to the connection state under the premise of covering the entire network. And redundant control message packets are minimized to improve the diffusion efficiency of control messages in the current scenario. Taking a network that requires one driver node as an example, the hop count of all nodes from the driver node is taken as the level of each node. Besides, different priorities are set for nodes in the same level. By default, the node with a smaller index value in the adjacency matrix has a higher priority. Control messages are transmitted in a hierarchical order like a stream. After a node receives a new control message if the node has sub-level nodes and these nodes do not fully belong to the same level with high-priority nodes, then the packet will be forwarded. Otherwise, the packet will be dropped. The pseudo code for control message diffusion is Algorithm 1.

As the topology is shown in Fig. 5, supposing node j is the driver node, from which the control message will be transmitted out. According to the minimum hops rule, all i-series nodes' level is 1, and all k-series nodes' is 2. The direction of transmission of valid (not discarded) control messages is shown by the arrow. For example, the valid control message

Algorithm 1 Diffusion of control message

Taking $node_i$ **as example:**

```
while True do
    packet = receive()
    if is_repetitive(packet) then
        discard(packet)
    else
        if is_covered(node_i.children()) == False then
            forward(packet)
        end if
        process(packet)
    end if
end while
```

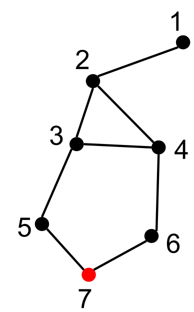


Fig. 6. New index value of Fig. 4.

received by node i+1 is sent by j. Because it is a wireless network, each node will inevitably receive invalid packets from neighboring nodes. For instance, node i+1 will also receive packets forwarded by node i, but this invalid message will be discarded. After receiving the control message packet, node i and i+2 will forward it. This is because i has the highest priority in this layer, and i+2 sub-level node (k+2) is not covered. And i+1's sub-level nodes are completely covered by node i, so i+1 node do not forward messages. Besides, k-series nodes have no sub-level nodes, so they do not need to perform forwarding operations. In the basic diffusion algorithm, all nodes need to forward related packets what received . Though, according to this strategy, only three nodes need to forward messages in this architecture.

When a network requires multiple driver nodes, each node needs to accumulate all control signals according to the characteristics of driver nodes. That is, for multi-driver nodes, it can be calculated independently for each driver node based on above steps.

2) Driver Nodes Optimization: According to the above calculation process, we can know that the number of minimum driver nodes of a network is a certain value, but which nodes are selected as the driver nodes is flexible and selectable. In the process of getting column canonical form of a matrix, the calculation process determines that the node with the larger index value will be selected as the driver node among the equivalent nodes. However, it is clear that for a graph, all nodes can be indexed according to certain rules, and different indexing rules can generate different adjacency matrices for the same graph, which also affect subsequent calculations and which nodes will be selected as driver nodes.

As the index value of each node in Fig. 4, the driver node is node 7. Supposing the index value shown in Fig. 6 is used, then the driver node is 7 as shown the red node. However, the specific nodes corresponding to these two nodes 7 in the graph are entirely different, and the latter node 7 corresponds to the former node 6.

When diffusing control messages to the entire network, the transmission speed and efficiency of different driver nodes are different. The more forwarding hops a message needs to pass, the more processing time and transmission time it takes. Obviously, the closer the node is to the topology boundary, that is, the larger the maximum hops from the driver node in the entire network nodes, the longer time it will take, and vice versa. For message diffusion efficiency, it is necessary to avoid selecting nodes close to the network topology boundary as driver nodes as much as possible. In view of this feature, the selection process of driver nodes can be optimized.

$$argmin_{i \in ADNs}[\max\left(hops(j,\ i), j \in All\right] \tag{17}$$

The most ideal state is to adjust the driver node to obtain the minimum value of the maximum number of hops from each node to the driver node as Eq.(17), in which the alternative driver nodes ($ADNs$) means a set of equivalent driver nodes, the function $hops(j,i)$ represents the shortest hop between node i and j and All represents all nodes of this network. It requires a lot of computing resources and time, so it is unrealistic to calculate the optimal value in the current scenario. Two natural ways to index the nodes from one side to the other directly or in a completely random way, which, especially the former, will increase the possibility that the driver node close to network topology boundary. The node with the larger index value will be selected mathematically, so we use the S2C (from Surrounding to Center) indexing rule, strictly speaking, which is not a solution algorithm but a strategy, to make the central nodes indexed larger value than the surroundings. The benefit of this indexing rule is more apparent in the highly connected network. For ease of understanding, we take the network with three nodes as an example and the maximum number of hops under the two indexing rules (S2C and S2S, from one Side to the other Side) as shown in the Fig. 7. The S2S needs two hops to diffuse, and the S2C only needs one hop.

The node processing abilities will mainly affect the stage of control message calculation. If it is taken into consideration when selecting driver nodes, then we propose a mechanism to combine S2C indexing rule. When computing is the bottleneck, high processing abilities nodes are critical, so index nodes in order of processing ability or specify large index values for high-performance nodes directly. In other cases, high processing abilities nodes can be tilted based on the S2C indexing rule. Since it is difficult to quantify the impact of

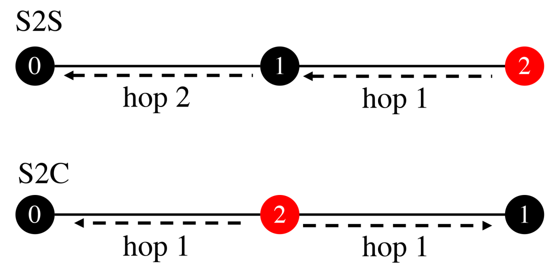


Fig. 7. Example for indexing rule.

computing power on calculation delay and transmission delay, we just provide a simple strategy to deal with the different processing abilities of each node.

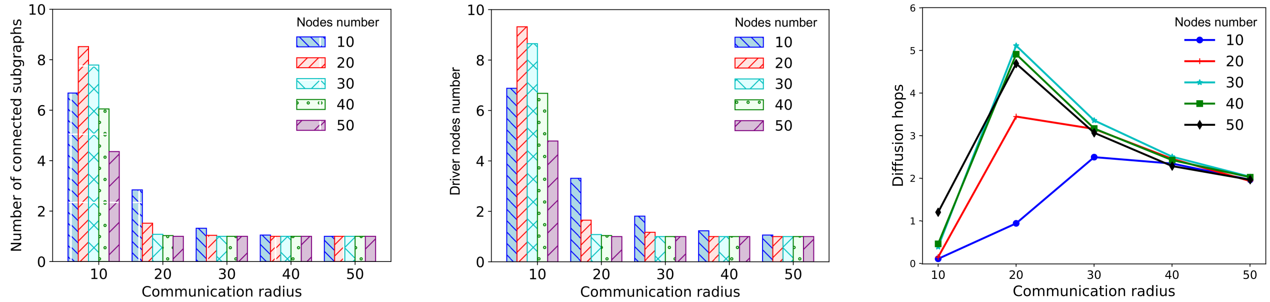
The demand for smart transportation and autonomous driving has prompted the development of vehicle sensors, communication equipment, and roadside units, which in turn makes the deployment of DND feasible. For example, on-sale vehicles or test models with automatic driving functions have computing capabilities and can perform information interaction or networking with roadside units and peers through sensors and communication devices. On this basis, the deployment of our proposed method, DND, can assist autonomous driving technology to complete more systematic decision-making and so on.

V. EVALUATION

By randomly generating nodes on a fixed area to simulate vehicles or other mobile devices, experimental analysis is performed from three perspectives, DND comprehensive performance, the relationship between the number of driver nodes and the node density, velocity, communication radius, control time, etc., and the improvement of DN Optimization algorithm. Since the focus of each experiment is different, we set different area parameters and node generation rules respectively. At present, it is difficult to deploy DND in a practical scenario or obtain fully matching data, so we generated the simulation data based on the public data of AutoNavi Map [55].

A. Comprehensive analysis

Assuming a fixed speed and control time, by adjusting the number of nodes and the communication radius, we observe the number of hops required for control messages to diffuse from driver nodes to the entire network, which corresponds to the diffusion time. The observation area is set to be 30 m*100 m, the number of nodes in the area is from 10 to 50, and the communication radius of each node is from 10 m to 50 m. As mentioned above in the paper, the limitation of finding driver nodes via the adjacency matrix is that the graph is connected. If there are several disconnected subgraphs in, then each subgraph needs to be calculated independently. Under the parameters set above, the number of subgraphs is shown in Fig.8(a). When the number of nodes is fixed, the connection of each node will increase with the increase of the



(a) Number of subgraphs (b) Number of driver nodes (c) Diffusion hops

Fig. 8. DND comprehensive analysis

communication radius, and then the number of subgraphs will also decrease. When the communication radius is fixed, it is intuitively assumed that the number of subgraphs will decrease as the nodes number increases. However, the number will increase first and then decrease in reality. This is because when the number of nodes in the increases, due to the limitation of the communication radius, many isolated subgraphs will appear, which will also cause the driver node to show an approximate rule, as shown in Fig.8(b).

All of the above will affect the diffusion of control messages. We stipulate that it is 1 hop to forward from one node to another node, and the message of the driver node itself does not need to be forwarded, which is 0 hop. When the communication radius is small, many isolated nodes will be the driver nodes in the single-node network, so the number of diffusion hops is small; as the communication radius increases, the connected subgraph will merge into a new subgraph, but due to the connection degree is small, so the diffusion hops show an increasing trend. As the communication radius continues to increase, the connectivity of the entire network increases and the diffusion hops will show a downward trend, as shown in Fig.8(c).

B. Driver nodes number analysis

In this part, the experiment sets a scene of a road with Y meters wide and X meters long. According to the width of the road, the entire road is divided into X/Y segments. Nodes in each segment are generated according to the density and obey the Poisson distribution. In each segment the position (x_i, y_i) of each node is completely random, and with velocity $V_i x$ in the X direction and velocity $V_i y$ in the Y direction according to the velocity. And note that the value of the velocity is $Vx \pm 2$ or $Vy \pm 2$ actually. Both the communication radius and the control time are set according to different values within a practical reasonable range. When updating the position of each node according to its velocity, it is set that node moves to the opposite direction at the next moment when it reaches the boundary of the region, so as to ensure the number of nodes of the entire region do not change.

According to the model, we compared experiments' results by assigning different values to parameters of node velocity, communication radius, density and control time. In order to eliminate the randomness of the results caused by randomly generated nodes, the same random nodes are used in calculations of different parameter assignment when the density is fixed. And the results were calculated 100 times for each assignment and get the average value as the final result.

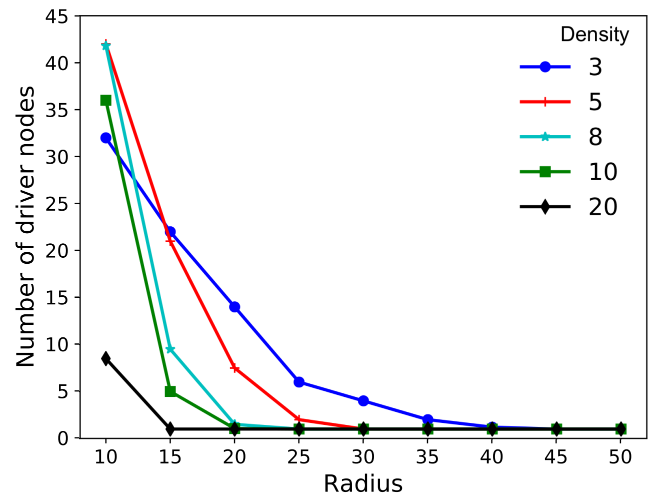


Fig. 9. Relationship between number of driver nodes and communication radius.

1) Communication Radius: In practical, the communication radius is based on signal module hardware or communication protocols (e.g. DSRC, LTE-V-Direct, etc.) rather than artificially set. And it will also be affected by the environment, such as the attenuation caused by the occlusion of vehicles, so we set radius as a variable parameter. When only considering the state at a certain time, that is, do not consider the impact of refresh and node movement, the relationship between the number of driver nodes and node communication radius is shown in Fig. 9.

The abscissa is the value of communication radius in m and the ordinate represents the number of required driver nodes. The density is in units of $90m^2$. Five curves with different colors represent different node densities. It is necessary to explain that at the radius of 10, the density of 3, the blue curve, requires less driver node than the density of 5, the red

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2021.3059696, IEEE Transactions on Network and Service Management

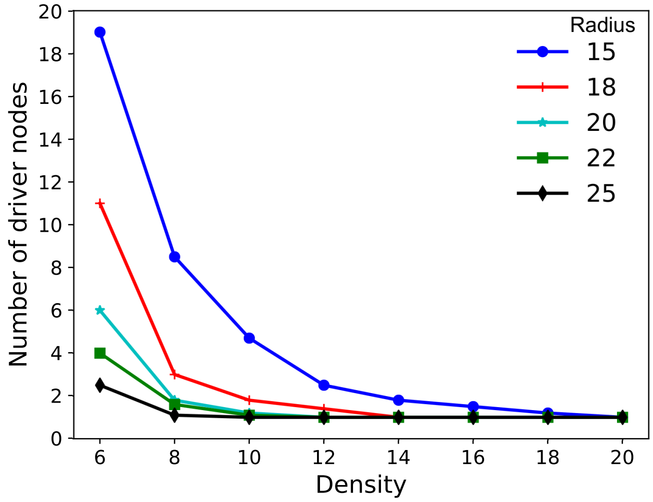


Fig. 10. Relationship between number of driver nodes and density.

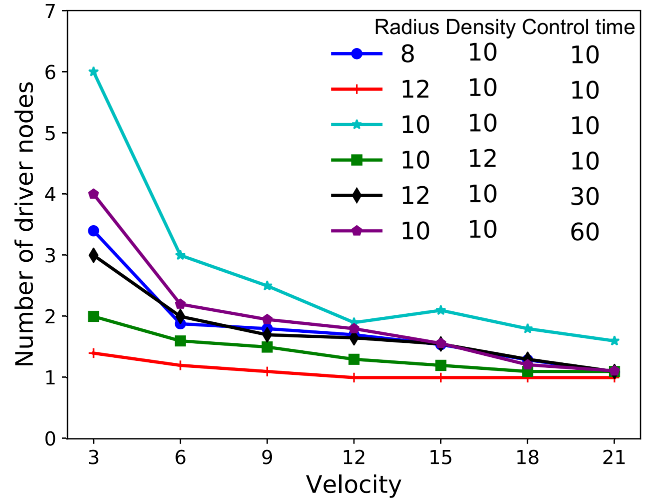


Fig. 11. Relationship between number of driver nodes and velocity.

curve or 8, the yellow curve. This is because the total number of driver nodes at density of 3 is much smaller than in other cases, however, the proportion of driver nodes of density of 3 is actually much higher than other cases.

It can be observed from the graph that with the increment of communication radius, the number of driver nodes needed in the whole network decreases and tends to the minimum of 1. The reason for the overall downward trend is that as the communication radius increases, each node can establish connection relations with more nodes, and the number of edges in the whole network increases. Under different density conditions, it is that increase speed of connections number is different, so the pace of declines is different, moreover, the higher the density, the higher the rate of increase of the number of connections.

2) Nodes Density: Because the observation range size is fixed in experiment, the node density also could reflect the network scale. Similarly, all conditions are consistent with those mentioned above. When only considering the static state at certain time point, the relationship between the number of driver nodes and the density of nodes is shown in Fig. 10. Five curves with different colors represent different communication radius of nodes. The number of driver nodes decreases as the density increases and tends to a minimum of 1 finally.

Moreover, under different communication radius conditions, the pace of declines of number of driver nodes is different with the increase of node density, and the larger the communication radius is, the faster it decreases to the minimum value. This is because, when increasing the same node density, the number of new connections with others of each node is proportional to the square of the radius, so the larger the communication radius, the faster it approaches the minimum value.

3) Nodes Velocity: For the node velocity as a control variable parameter, other variables are combined in different cases. The result of number of driver nodes is shown in Fig. 11.

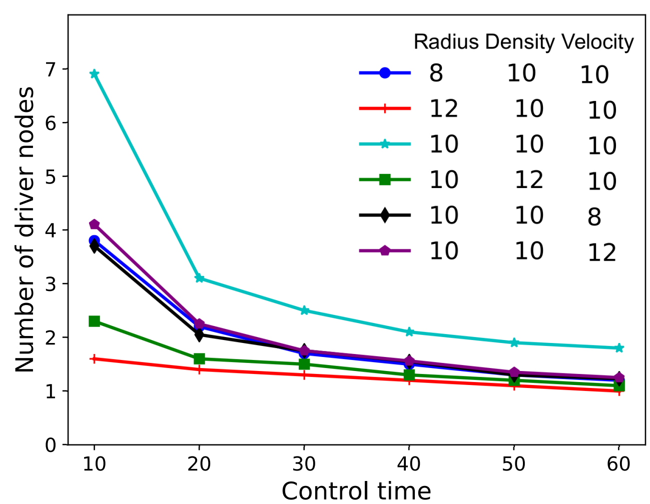


Fig. 12. Relationship between number of driver nodes and control time.

Six curves with different colors correspond to different control time, node communication radius or node density. According to the overall trend of the six color curves, as the node velocity increases , the number of required driver nodes in the network shows a downward trend, ultimately, it tends to the minimum value of 1. This is because with the increment of velocity, refresh frequency will increase, then the connection degree of the whole network will increase in the same control time, longer control time is even more obvious. And why does each curve have a different rate of decline we've explained above.

4) Control Time: For the control time in s as a control variable parameter, other variables compose different situations. The results of number of driver nodes as shown in Fig. 12. Six curves with different colors correspond to different node density, node communication radius and node velocity. With the control time increases, the total times of refresh

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2021.3059696, IEEE Transactions on Network and Service Management

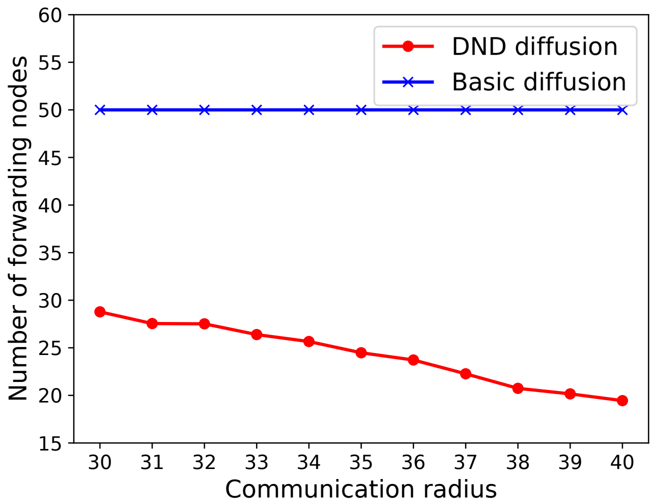


Fig. 13. DND diffusion algorithm.

will increase that means the connection degree of the whole network will increase, which leads to the number of required driver nodes in the network shows a downward trend, and tends to the minimum of 1 finally.

The driver nodes required in the latter two experiments are much less than those in the former two. This is because the influence of velocity and control time on dynamic network were considered, resulting in the connection degree of the whole graph is much higher than former two experiments.

Although the experimental data are simulated, but they are generated as close to the actual situation as possible. If there is a chance to obtain the actual road condition data through the relevant agencies, the real results can be obtained. However, the regular relationship between the parameters obtained through the experiment is universal.

C. Diffusion of control message analysis

In this part, we define a 100m * 100m area and generate 50 nodes at random positions in this area. By adjusting the communication radius to generate networks with different connectivity, the adjustment range is from 30m to 40m. The efficiency of the control message diffusion algorithm of DND adapted in the network with driver nodes is experimented, and the selection of driver nodes in the previous step is optimized according to the DND diffusion algorithm. Taking diffusion efficiency as the optimization evaluation standard, the experimental results confirm that the optimized driver node selection strategy has a certain improvement in the overall efficiency of the network. Moreover, to avoid randomness and bring unnecessary impact on the experimental results, we first generate 100 sets of random nodes data and average the results of these sets under different radius.

1) DND Diffusion Algorithm: The larger the communication radius of the nodes, the more distant vehicles can be connected, so the overall connection degree of each node is greater, and vice versa. In the current network topology, the adjacency matrix is generated by randomly indexing all the nodes, thereby calculating the driver node, which is the origin node of the control message. Under this kind of node density and communication radius, all the networks in this group of experiments only need one driver node, which is suitable for experimental comparison of diffusion algorithm efficiency. In contrast, the basic algorithm is that all nodes must forward the effective diffusion message, that is, the diffusion efficiency is to handle all nodes.

Experimental results are shown in the Fig. 13, the value of the basic algorithm is constant, and the efficiency of the DND diffusion algorithm increases with increasing radius. When the radius is 30m, 29 nodes need to forward packets. As the radius increases, the relevant value decreases. When the radius reaches 40m, the number of nodes that need to forward packets is 20, only accounting for 40% of all nodes.

Through the DND diffusion algorithm, the number of nodes that forward packets can be reduced, thereby reducing the computational burden of each node and the number of redundant packets in the network, avoiding unnecessary waste of network equipment as much as possible, and improving the overall efficiency of the network system.

2) DNs Optimization: We judge the efficiency of the entire system in a time when the control message diffuses to the whole network. In a real scenario, what differences in computing power and other network hardware devices of each device may result in different processing times for a packet, but each additional packet forwarding will increase the overall transmission time. According to this feature, to concentrate on the points that need to be compared, we ignore the different effects that different hardware devices may cause and simplify the calculation of efficiency to the number of forwarding corresponding to the control message diffused to the entire network. Since the forwarding operation of each device is relatively independent, that is, the whole forwarding processes are executed in parallel like branches of a river, so the maximum forwarding times in all nodes demand determines the final diffusion time.

When the index operation traverses from one side of the area to the other side in a certain direction or completely randomly, the driver node is likely to be at the boundary of the network topology, the and there is no doubt that the value of the maximum hops of the network is increased, thereby rising transmission time, which ultimately increases the diffusion time. Regarding the index operation, we adopt the index from the surrounding to the center which will limit the position of the driver node to the topological center with a high expectation, which can diminish the value of the maximum hops and promote the diffusion efficiency.

About the test parameters, 50 nodes are randomly generated in a fixed area (100m * 100m), the communication radius of each node is equal whose value is set in groups of different values of 30m to 40m to enhance the reliability of the experimental results. Index traversal of all nodes in three fashions, S2S, completely random way and S2C, apply the above DND algorithm to find driver nodes, and then calculate the value

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2021.3059696, IEEE Transactions on Network and Service Management

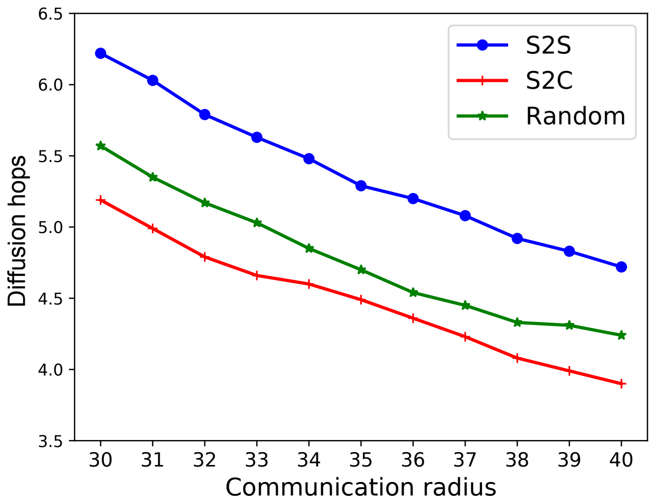


Fig. 14. Comparison of indexing rules.

of the maximum number of hops, which can represent the diffusion time. To eliminate the unnecessary impact of the randomness of all nodes and augment the accuracy of the experiment, 100 different sets of random nodes are calculated in each case, and their average is adopted as the final result.

The experimental results are shown in Fig. 14. With the current parameter settings, the maximum hop count of the S2S or random index rule is obviously larger than the S2C index rule. It should be noted here, due to the uncertainty of the nodes quantity, the random method requires additional calculation to satisfy the randomness than the other two rules, while the S2C rule does not cause additional calculate consumption than S2S. When the communication radius is 30m, the number of the average maximum hops required for the diffusion of the S2S rule is 6.22, the completely random way's is 5.57, and the S2C rule's is 5.19. As the communication radius increases, this value shows a downward trend, but S2C maintains smaller than S2S and random way. The performance is improved by 1.2x and 1.1x than S2S and random way when the communication radius is 40m. Therefore, the 'from the surrounding to the center' index rule can enhance the transmission efficiency in the process of control message diffusion.

VI. Conclusion

The rapid movement of nodes in dynamic network results in the rapid change of network topology, for example, the smart transportation networks, and such network is thus facing the controllability challenge. In this paper, we first formulate such scenario into dynamic temporal network, and then design the DND (Driver Node Detection) algorithm to calculate the minimum number of driver nodes that are needed to make the whole network controllable. We conduct series of experiments and then disclose the relationship between driver node number and different network settings. As supplements to analyze the controllability of time-varying dynamic networks and explore the application of driver nodes, we adapted the DND control message diffusion algorithm for networks, which reduced the number of nodes that need to forward packets during diffusion and redundant control message packets in the network. And we use the diffusion algorithm as feedback to optimize the driver node selection, which reduced the control message maximum forward times over the entire network. These both developed the transmission efficiency to the network. These experiments and results mentioned above can not only give reference to network controllability, but also save the deployment cost in smart transportations. In the future, we'll further analyze the deployment problem of the selected driver nodes in more complex scenarios.

VII. Acknowledgment

The work was supported in part by the National Natural Science Foundation of China (NSFC) Youth Science Foundation under Grant 61802024, Key Special Projects of Beijing Natural Science Foundation under Grant M21030, the Fundamental Research Funds for The Central Universities under Grant 2482020RC36, and the National Key R&D Program of China under Grant 2019YFB1802603. Wendong Wang's work was supported by the National Natural Science Foundation of China (NSFC) under Grant 62072047.

References

[1] A. Casteigts, P. Flocchini, W. Quattrociocchi, and N. Santoro, "Time-varying graphs and dynamic networks," *International Journal of Parallel Emergent & Distributed Systems*, vol. 27, no. 5, pp. 387–408, 2012.

[2] Z. Xiao, C. Moore, and M. E. J. Newman, "Random graph models for dynamic networks," *European Physical Journal B*, vol. 90, no. 10, p. 200, 2016.

[3] Y. Hao, Y. Miao, L. Hu, M. S. Hossain, G. Muhammad, and S. U. Amin, "Smart-edge-cocaco: Ai-enabled smart edge with joint computation, caching, and communication in heterogeneous iot," *IEEE Network*, vol. 33, no. 2, pp. 58–64, 2019.

[4] A. Baltag, Z. Christoff, R. K. Rendsvig, and S. Smets, "Dynamic epistemic logics of diffusion and prediction in social networks," *Studia Logica*, vol. 107, no. 3, pp. 489–531, 2019.

[5] Y. Huang, H. Xi, S. C. Shah, and F. Ye, "The evaluation and application of node influence in dynamic networks based on evolving communities," *Transactions on Emerging Telecommunications Technologies*, vol. 30, no. 9, p. e3556, 2019.

[6] N. Hafiene, W. Karoui, and L. Ben Romdhane, "Influential nodes detection in dynamic social networks," in *Business Information Systems*, W. Abramowicz and R. Corchuelo, Eds. Cham: Springer International Publishing, 2019, pp. 62–73.

[7] J. Cheng, G. Yuan, M. Zhou, S. Gao, C. Liu, H. Duan, and Q. Zeng, "Accessibility analysis and modeling for iov in an urban scene," *IEEE Transactions on Vehicular Technology*, 2020.

[8] A. Yassine, S. Singh, M. S. Hossain, and G. Muhammad, "Iot big data analytics for smart homes with fog and cloud computing," *Future Generation Computer Systems*, vol. 91, pp. 563–573, 2019.

[9] S. Ghoorchian and S. Maghsudi, "Multi-armed bandit for energy-efficient and delay-sensitive edge computing in dynamic networks with uncertainty," *CoRR*, vol. abs/1904.06258, 2019.

[10] P. Cong, Y. Zhang, W. Wang, and B. Bai, "Dnd: The controllability of dynamic temporal network in smart transportations," in *2019 IEEE Globecom Workshops (GC Wkshps)*. IEEE, 2019, pp. 1–6.

[11] C. Lin, G. Han, X. Qi, M. Guizani, and L. Shu, "A distributed mobile fog computing scheme for mobile delay-sensitive applications in sdn-enabled vehicular networks," *IEEE Transactions on Vehicular Technology*, 2020.

[12] M. Sohail, L. Wang, R. Ali, S. Rahim, and J. Yao, "Efficient data handover and intelligent information assessment in software-defined vehicular social networks," *IET Intelligent Transport Systems*, vol. 13, no. 12, pp. 1814–1821, 2019.

[13] J. Zhang, H. Guo, J. Liu, and Y. Zhang, "Task offloading in vehicular edge computing networks: A load-balancing solution," *IEEE Transactions on Vehicular Technology*, 2019.

[14] O. Kaiwartya, A. H. Abdullah, Y. Cao, A. Altameem, and X. Liu, "Internet of vehicles: Motivation, layered architecture network model challenges and future aspects," *IEEE Access*, vol. 4, no. pp, pp. 5356–5373, 2017.

[15] K. M. Alam, M. Saini, and A. E. Saddik, "Toward social internet of vehicles: Concept, architecture, and applications," *IEEE Access*, vol. 3, pp. 343–357, 2015.

[16] M. Liu, J. Yang, and G. Gui, "Dsf-noma: Uav-assisted emergency communication technology in a heterogeneous internet of things," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 5508–5519, 2019.

[17] X. Wang, S. Han, L. Yang, T. Yao, and L. Li, "Parallel internet of vehicles: Acp-based system architecture and behavioral modeling," *IEEE Internet of Things Journal*, 2020.

[18] C. Yu, B. Lin, P. Guo, W. Zhang, S. Li, and R. He, "Deployment and dimensioning of fog computing-based internet of vehicle infrastructure for autonomous driving," *IEEE Internet of Things Journal*, vol. 6, no. 1, pp. 149–160, 2018.

[19] W. Feng, J. Wang, Y. Chen, X. Wang, N. Ge, and J. Lu, "Uav-aided mimo communications for 5g internet of things," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1731–1740, 2018.

[20] S. Zhang, J. Liu, and W. Sun, "Stochastic geometric analysis of multiple unmanned aerial vehicle-assisted communications over internet of things," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 5446–5460, 2019.

[21] B. Baron, P. Spathis, H. Rivano, M. D. de Amorim, Y. Viniotis, and M. H. Ammar, "Centrally controlled mass data offloading using vehicular traffic," *IEEE Transactions on Network and Service Management*, vol. 14, no. 2, pp. 401–415, 2017.

[22] O. Sadio, I. Ngom, and C. Lishou, "Design and prototyping of a software defined vehicular networking," *IEEE Transactions on Vehicular Technology*, 2019.

[23] J. Guo, B. Song, S. Chen, F. R. Yu, X. Du, and M. Guizani, "Context-aware object detection for vehicular networks based on edge-cloud cooperation," *IEEE Internet of Things Journal*, 2019.

[24] H. Peng, Q. Ye, and X. S. Shen, "Sdn-based resource management for autonomous vehicular networks: A multi-access edge computing approach," *IEEE Wireless Communications*, vol. 26, no. 4, pp. 156–162, 2019.

[25] J.-S. Weng, J. Weng, Y. Zhang, W. Luo, and W. Lan, "Benbi: Scalable and dynamic access control on the northbound interface of sdn-based vanet," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 1, pp. 822–831, 2018.

[26] S. Garg, K. Kaur, G. Kaddoum, S. H. Ahmed, and D. N. K. Jayakody, "Sdn-based secure and privacy-preserving scheme for vehicular networks: a 5g perspective," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 9, pp. 8421–8434, 2019.

[27] J. Gao, K. O.-B. O. Agyekum, E. B. Sifah, K. N. Acheampong, Q. Xia, X. Du, M. Guizani, and H. Xia, "A blockchain-sdn enabled internet of vehicles environment for fog computing and 5g networks," *IEEE Internet of Things Journal*, 2019.

[28] G. Raja, A. Ganapathisubramaniyan, S. Anbalagan, S. B. M. Baskaran, K. Raja, and A. K. Bashir, "Intelligent reward based data offloading in next generation vehicular networks," *IEEE Internet of Things Journal*, 2020.

[29] G. S. Park and H. Song, "Video quality-aware traffic offloading system for video streaming services over 5g networks with dual connectivity," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 6, pp. 5928–5943, 2019.

[30] M. Abolhasan, M. Abdollahi, W. Ni, A. Jamalipour, N. Shariati, and J. Lipman, "A routing framework for offloading traffic from cellular networks to sdn-based multi-hop device-to-device networks," *IEEE Transactions on Network and Service Management*, vol. 15, no. 4, pp. 1516–1531, 2018.

[31] T. H. Luan and X. S. Shen, "A queuing based model for analyzing multihop performance in vanet," in *2018 IEEE International Conference on Communication Systems (ICCS)*. IEEE, 2018, pp. 186–191.

[32] N. Cheng, F. Lyu, J. Chen, W. Xu, H. Zhou, S. Zhang, and X. S. Shen, "Big data driven vehicular networks," *IEEE Network*, vol. 32, no. 6, pp. 160–167, 2018.

[33] P. Misra and V. Kulathumani, "Roadnet: A multi-resolution transmission strategy for long range information diffusion in vanets," in *2019 IEEE 90th Vehicular Technology Conference (VTC2019-Fall)*. IEEE, 2019, pp. 1–6.

[34] G. Evropeytsev, S. E. P. Hernández, J. R. P. Cruz, L. M. R. Henríquez, and E. L. Domínguez, "A scalable indirect position-based causal diffusion protocol for vehicular networks," *IEEE Access*, vol. 7, pp. 14 767–14 778, 2019.

[35] X. M. Zhang, L. Yan, H. Zhang, and D. K. Sung, "A concurrent transmission based broadcast scheme for urban vanets," *IEEE Transactions on Mobile Computing*, vol. 18, no. 1, pp. 1–12, 2018.

[36] X. M. Zhang, L. Yan, K. H. Chen, and D. K. Sung, "Fast, efficient broadcast schemes based on the prediction of dynamics in vehicular ad hoc networks," *IEEE Transactions on Intelligent Transportation Systems*, 2019.

[37] T. Saeed, Y. Mylonas, A. Pitsillides, V. Papadopoulou, and M. Lestas, "Modeling probabilistic flooding in vanets for optimal rebroadcast probabilities," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 2, pp. 556–570, 2018.

[38] F. Klingler, R. Cohen, C. Sommer, and F. Dressler, "Bloom hopping: Bloom filter based 2-hop neighbor management in vanets," *IEEE Transactions on Mobile Computing*, vol. 18, no. 3, pp. 534–545, 2018.

[39] A. Rahim, T. Qiu, Z. Ning, J. Wang, N. Ullah, A. Tolba, and F. Xia, "Social acquaintance based routing in vehicular social networks," *Future Generation Computer Systems*, vol. 93, pp. 751–760, 2019.

[40] R. Urena, G. Kou, Y. Dong, F. Chiclana, and E. Herrera-Viedma, "A review on trust propagation and opinion dynamics in social networks and group decision making frameworks," *Information Sciences*, vol. 478, pp. 461–475, 2019.

[41] F. Zhang, J. Thiyagalingam, T. Kirubarajan, and S. Xu, "Speed-adaptive multi-copy routing for vehicular delay tolerant networks," *Future Generation Computer Systems*, vol. 94, pp. 392–407, 2019.

[42] D. Liu, J. Wang, Y. Xu, Y. Xu, Y. Yang, and Q. Wu, "Opportunistic mobility utilization in flying ad-hoc networks: A dynamic matching approach," *IEEE Communications Letters*, vol. 23, no. 4, pp. 728–731, 2019.

[43] D. Liu, J. Wang, Y. Xu, Y. Zhang, Q. Wu, and A. Anpalagan, "Opportunistic data ferrying in uav-assisted d2d networks: A dynamic hierarchical game," in *ICC 2019-2019 IEEE International Conference on Communications (ICC)*. IEEE, 2019, pp. 1–6.

[44] A. Lombardi and M. Hörnquist, "Controllability analysis of networks," *Phys.rev.e*, vol. 75, no. 5 Pt 2, p. 056110, 2007.

[45] Y.-Y. Liu, J.-J. Slotine, and A.-L. Barabási, "Controllability of complex networks," *nature*, vol. 473, no. 7346, p. 167, 2011.

[46] Z. Yuan, C. Zhao, Z. Di, W. X. Wang, and Y. C. Lai, "Exact controllability of complex networks." *Nature Communications*, vol. 4, no. 2447, p. 2447, 2013.

[47] S. P. Cornelius, W. L. Kath, and A. E. Motter, "Realistic control of network dynamics," *Nature Communications*, vol. 4, no. 3, p. 1942, 2013.

[48] S. Francesco, D. B. Mario, G. Franco, and C. Guanrong, "Controllability of complex networks via pinning," *Physical Review E Statistical Nonlinear & Soft Matter Physics*, vol. 75, no. 2, p. 046103, 2007.

[49] F. Pasqualetti, S. Zampieri, and F. Bullo, "Controllability metrics, limitations and algorithms for complex networks," *IEEE Transactions on Control of Network Systems*, vol. 1, no. 1, pp. 40–52, 2014.

[50] W. X. Wang, X. Ni, Y. C. Lai, and C. Grebogi, "Optimizing controllability of complex networks by minimum structural perturbations," *Physical Review E Statistical Nonlinear & Soft Matter Physics*, vol. 85, no. 2 Pt 2, p. 026115, 2012.

[51] W. Xiao, X. Zhu, W. Bao, L. Liu, and J. Yao, "Cooperative data sharing for mobile cloudlets under heterogeneous environments," *IEEE Transactions on Network and Service Management*, vol. 16, no. 2, pp. 430–444, 2019.

[52] L. Lv, Y. Zhang, Y. Li, K. Xu, D. Wang, W. Wang, M. Li, X. Cao, and Q. Liang, "Communication-aware container placement and reassignment in large-scale internet data centers," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 3, pp. 540–555, 2019.

[53] W. L. Brogan, *Modern control theory*. Pearson education india, 1991.

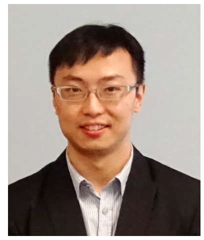
[54] S. Panichpapiboon and W. Pattara-Atikom, "A review of information dissemination protocols for vehicular ad hoc networks," *IEEE Communications Surveys & Tutorials*, vol. 14, no. 3, pp. 784–798, 2011.

[55] A. Map, "Traffic health list of major cities in china." [Online]. Available: https://report.amap.com/diagnosis/index.do

Peizhuang Cong is currently a Ph.D. candidate in State Key Laboratory of Network and Switching Technology, Beijing University of Posts and Telecommunications, Beijing, China. His current research interests include the next generation network architecture, data-driven networks and mobile Internet.

Yuchao Zhang received her Ph.D. degree from Computer Science Department at Tsinghua University in 2017. Before that, she received the B.S. degree in computer science and technology from Jilin University in 2012. Her research interests include large scale datacenter networks, content delivery networks, data-driven networks and edge computing. She is currently with the Beijing University of Posts and Telecommunications as an associate professor.

Wendong Wang (M'05) received his B.E. and M.E. degrees both from the Beijing University of Posts and Telecommunications, China, in 1985 and 1991, respectively, where he is currently a Full Professor in State Key Laboratory of Networking and Switching Technology. He has published over 200 of papers in various journals and conference proceedings. His current research interests are the next generation network architecture, network resources management and QoS, and mobile Internet. He is a member of IEEE.



Ning Zhang (M'15-SM'18) received the Ph.D degree from University of Waterloo, Canada, in 2015. After that, he was a postdoc research fellow at University of Waterloo and University of Toronto, Canada, respectively. He is an Associate Professor at University of Windsor, Canada. He serves as an Associate Editor of IEEE Internet of Things Journal, IEEE Transactions on Cognitive Communications and Networking, IEEE Access, and IET Communications, and Vehicular Communications (Elsevier); and a Guest Editor of several international journals, such as IEEE Wireless Communications, IEEE Transactions on Industrial Informatics, and IEEE Transactions on Cognitive Communications and Networking. He also serves/served as a track chair for several international conferences and a co-chair for several international workshops. He received the Best Paper Awards from IEEE Globecom in 2014, IEEE WCSP in 2015, and Journal of Communications and Information Networks in 2018, IEEE ICC in 2019, IEEE Technical Committee on Transmission Access and Optical Systems in 2019, and IEEE ICCC in 2019, respectively. He has been a senior member of IEEE since 2018.