

Research Article

Ad Hoc Communication Topology Construction for Delay-Sensitive Internet of Vehicle Applications

Joong-Lyul Lee ¹, Jaejin Hwang,² Hyesung Park,³ and Donghyun Kim⁴

¹Department of Mathematics and Computer Science, University of North Carolina at Pembroke, Pembroke, NC 28372, USA

²LG Hausys, Atlanta, GA 30339, USA

³Department of Computer Science, Georgia Gwinnett College, Lawrenceville, GA 30043, USA

⁴Department of Computer Science, Georgia State University, Atlanta, GA 30302, USA

Correspondence should be addressed to Joong-Lyul Lee; joonglyul.lee@uncp.edu

Received 29 July 2021; Revised 26 January 2022; Accepted 1 February 2022; Published 13 April 2022

Academic Editor: Changqing Luo

Copyright © 2022 Joong-Lyul Lee et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Advances in new technologies are changing the Internet of Things (IoTs) from vehicular ad hoc networks (VANETs) to Internet of Vehicles (IoVs). This IoV needs to construct a network topology in order to efficiently send and receive data to Cloud or Fog nodes. In such a topology construction, there are various methods of configuring a topology considering network interference and configuring a topology for energy saving, such as a mobile sensor network. However, there has been no research on the topology construction considering the rapidly high movement of mobile node, characteristic of IoV. In this article, we proposed the k -Lifetime Tree Topology Construction (k -LTTC) algorithm during emergency situations for IoV with highly dynamic characteristics. By applying the time domain to the topology construction, it is guaranteed to maintain the topology for a specific period of time (k -Life Time). The first algorithm is a centralized algorithm to construct k -lifetime tree topology, and the second algorithm is a decentralized algorithm to construct k -lifetime tree topology. We describe the time complexity and message complexity for the mathematical analysis of our proposed algorithms. Furthermore, we provided the simulation results to verify our proposed method.

1. Introduction

The vehicular ad hoc networks (VANETs) have been received a lot of attention in research in both academia and industry. Currently, with the development of Internet of Things (IoTs) technology and the increase in the number of IoTs, these two technologies of IoT and VANETs are converging [1–5]. IoVs improve the efficiency and safety of modern transportation systems. Traffic congestion detection, road condition monitoring (e.g., icy road surface), and pothole detection are a few examples demonstrating the significance of IoV and its applications [6–12]. In a typical setting, each IoV node is equipped with an On-Board Unit (OBU), which is an integrated electric device with various modules such as GPS and a radio transceiver. Thanks to the advances in the relevant technologies, the cost to build an IoV node is getting lower; therefore, more and more people believe the benefits of IoV will come to our real life in the very near future [13–16].

The communication technologies adopted by IoV so far fall into one of the following five categories; (a) Vehicle to Roadside (V2R), (b) Vehicle to Infrastructure (V2I), (c) Vehicle to Vehicle (V2V), (d) Vehicle to Sensors (V2S), or (e) Vehicle to Personal device (V2P). Each vehicle has an OBU that is capable of being connected with the Road Side Unit (RSU) infrastructure (which is connected to the Internet and the rest of the IoV infrastructure) using cellular or WLAN communication technology. In the literature, such a communication strategy is frequently referred to as V2I communication. While this approach is highly applicable to some instances, it is not adaptable for the areas where a proper infrastructure (e.g., 3G, 4G, and 5G cellular towers, IEEE 802.11p WAVE) is not in place for some reasons, e.g., very high deployment cost and extreme geographic limitation. The third approach can be used for situations where the first two approaches cannot be adopted at all. V2V is a well-established IoV networking concept for such a

completely infrastructure-less environment. V2S connects with sensor nodes to gather environment sensing information from each sensor node and V2P connects with personal devices to provide network service or personal device services (e.g., Google Android Auto and Apple CarPlay). In this IoV communication type, the hybrid approach is useful if the area of interest is partially covered by the service infrastructure, and the combination of V2V and V2I is used to provide the connectivity among IoV nodes and the IoV infrastructure. Regardless of the communication network implementation strategies, in IoV, each vehicle should be able to communicate with the other vehicles and the rest of IoV infrastructure directly (via single-hop communication) and/or indirectly (via multihop communication) (see Figure 1).

The hybrid communication framework is very useful in many real-world IoV application scenarios as every IoV node is not necessarily able to directly communicate with the main IoV infrastructure throughout an RSU. In this framework, the overall stability of IoV is highly dependent on the stability of the multihop routing path from an IoV node to an RSU over a number of intermediate IoV nodes. However, the identification of such a routing path is extremely difficult due to the highly dynamic nature of IoV. In addition, unless carefully chosen, such a path may suffer from unacceptable latency, which might be a crucial issue for many time-critical IoV applications. This article introduces a new strategy to construct a tree topology in emergency responding IoV applications, in which the root of the tree is a special emergency responding vehicle with an extremely long-distance communication capability and thus is directly connected to the rest of IoV infrastructure. We assume that the IoV nodes are highly mobile, but at the same time, they provide their short-term travel plans voluntarily to assist the formation of the tree topology to cope with the emergent situation. In fact, such a tree topology structure could be useful for data collection for IoV nodes about emergent information and the broadcasting of important notifications about the emergency situation to the IoV nodes, thus having great real-world significance. The main contributions of this article can be summarized as follows:

- (1) For the first time in the literature, we investigate the problem of constructing a tree topology with bounded latency in a highly dynamic IoV environment. In detail, given a positive integer k and a set of IoV nodes, whose future travel trajectories are known in advance, this article investigates the problem of constructing a tree topology, which is rooted at an IoV node (i.e., a police patrol car) and connects all the other IoV nodes such that the height of the tree is bounded by the given constant k . We formally define the problem of our interest as the k -Lifetime Tree Topology Construction (k -LTTC) problem, whose goal is to construct a k -lifetime tree topology, a bounded height rooted-tree topology over IoV nodes that can last for coming k -time slots.

- (2) We propose two heuristic algorithms with polynomial running time for the k -LTTC problem. The first algorithm is a centralized algorithm for the k -LTTC problem by exploiting the concept of the connected dominating set based on the assumptions that Intelligent Transportation Systems (ITS), which is operating the IoV infrastructure and attempts to construct the topology and knows the detailed information of every participating vehicle (e.g., node location and future travel trajectories). The second algorithm is a distributed version for the k -LTTC problem. Therefore, it is more applicable to real-world situations because sometimes it does not exist the central node that has the entire information of the network.

The rest of this article is structured as follows: Section 2 describes the related work. In Section 3, we present the network model and problem formulation. Section 4 describes the result and discussion. Finally, Section 5 concludes this article.

2. Related Work

Recently, several efforts have been made to study the data aggregation and topology control schemes for conventional VANET. In [17], Ibrahim and Weigle proposed Cluster-based Accurate Syntactic Compression of Aggregated Data in VANETs (CASCADE) data aggregation technique. In [18], Nzouonta et al. proposed two road-based routing protocols using vehicular traffic routing protocols: a reactive protocol RBVT-R and a proactive protocol RBVT-P. Using simulations, the article showed that the RBVT protocols outperform existing ones in terms of the average delivery ratio and average delay. In [6], Yu et al. proposed an adaptive forwarding delay control scheme for VANET data aggregation based on a distributed learning algorithm. The authors provided simulation results to demonstrate the effectiveness of the proposed scheme. In [19], Zhu et al. studied a delay-constrained data aggregation scheme for VANETs. With the analysis based on real traces, the authors proposed a centralized and distributed aTree. The centralized and distributed aTree first constructs a data aggregation tree based on the shortest path tree and then assigns a waiting time budget to each node by the centralized or distributed way. In [20], Yuan et al. proposed an infrastructure-free data aggregation algorithm by restricting forwarders. The authors found that throughput decrement can be caused by DCF collisions in VANETs and showed that the proposed algorithm improves throughput with beacon-ignorance in high-density vehicle situations.

In [21, 22], Ferreira and Xuan et al. proposed and formally analysed the least cost path using the extended evolving graphs in Fixed Schedule Dynamic Networks (FSDNs). In [23], Bernd Thallner et al. proposed a distributed fault-tolerant algorithm for constructing a k -regular, k -connected topology for fault-tolerant communication in wireless ad hoc networks. The proposed algorithm adapts to highly dynamic

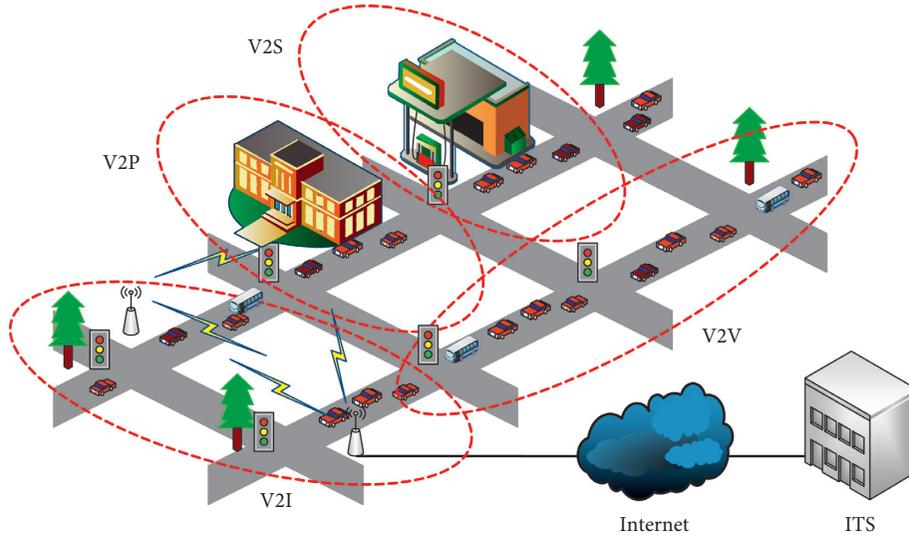


FIGURE 1: The types of vehicular communication in the Internet of Vehicles (IoV).

wireless networks as it is guaranteed to converge and exhibits good performance. In [24], Su et al. proposed a scheme to perform rerouting before route breaks predicting mobility of node using GPS devices in wireless networks. In [25], Zhigang et al. proposed a new clustering technique for large multihop vehicular ad hoc networks. A cluster is controlled by the priorities related to vehicles traffic information. It has high priority as a vehicle moves longer distances. In [26], Srivastava et al. proposed distributed algorithm for constructing a rooted spanning tree in a dynamic network and for finding a k -core of spanning tree. In [27], Althaus et al. proposed an algorithm for the minimum cost k -hop spanning tree problem in the given complete graph on n nodes with metric edge costs. In [28], Ghosh et al. proposed a bicriteria formulation to the optimal routing tree construction problem in order to consider bounded node degree and minimum radius. This result showed the trade-off in maximizing the data collection rate and minimizing the maximum packet delay. In the previous related work, the authors proposed an algorithm for topology construction in a dynamic environment and k -hop spanning tree. Our work is different from the previously existing work considering both the time domain and transmission delay in highly dynamic IoV environments.

3. Network Model and Problem Formulation

In this section, we describe the network model, notation, and problem formulation.

3.1. Notation and Problem Statement

Definition 1 (graph by time space). Given a graph $G = (V_{(n,t)}, E_{(i,j,t)})$ by time-spaces, each IoV node has location $V(n,t) = (x_p, y_p)$ in and each edge $E_{(i,j,t)} = (V_{(i,t)}, V_{(j,t)})$ in the time-space.

Definition 2 (radius of node v). The radius R_v of node v is defined as the maximum communication range of each IoV node v , i.e., $R_v = \max\{d_{cr}(v): v \in V\}$.

Definition 3 (moving distance of node v). The moving distance D_v of node v is defined as the maximum moving distance of each IoV node v , i.e., $D_v = \max\{d_{mv}(v): v \in V\}$.

Definition 4 (distance metric space). Consider the distance metrics space (V, δ) , where V is the set of IoV nodes with their locations at each time slot. For each pair of vehicular nodes $u, v \in V$, $\delta(u, v)$ is the Euclidean distance between vehicular node u and v :

- (1) $\delta(u, v) > 0$
- (2) $\delta(u, v) = 0$ if $u = v$
- (3) $\delta(u, v) = \delta(v, u)$ for all $u, v \in V$
- (4) $\delta(u, w) + \delta(w, v) \geq \delta(v, u)$ for all $u, v, w \in V$

Definition 5 ($\delta(i, j)$ of node i, j). $\delta(i, j)$ of node i, j is defined as $\delta(i, j) = |V_i - V_j| = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2}$.

Definition 6 (edge weight $w(i, j)$ of node i, j). The edge weight of node i, j is defined as network priority for Quality of Service (QoS) in the network. The high priority has a lower value. The edge weight is set to 1 as an initial value by default. This will be used to regulate overall network traffic by prioritizing specific high-priority applications.

Definition 7 (list of total edge weight $W(i, j)$ of node i, j). The list of edge weight of node i, j is defined as $G_{List} = \text{Sort}\{W(i, j), W(k, l), \dots\}$.

Definition 8 (location prediction of node v). The location prediction of IoV node v is defined as the next location in each time t moving each IoV node v by time t , speed s , and direction d , i.e., $x_{\text{moving}} = s \times \cos d$, $y_{\text{moving}} = s \times \sin d$,

- The location of $v(i, t_1) = \{x_{i0} + x_{\text{moving}}, y_{i0} + y_{\text{moving}}\}$
- The location of $v(i, t_2) = \{x_{i1} + x_{\text{moving}}, y_{i1} + y_{\text{moving}}\}$
- The location of $v(i, t_k) = \{x_{ik-1} + x_{\text{moving}}, y_{ik-1} + y_{\text{moving}}\}$

Problem 1. Given a set of vehicular set $V = \{v_1, v_2, v_3, \dots, v_n\}$ and constant $k \geq 2$, we want to construct k -lifetime tree topology in order to guarantee data aggregation within and during k -lifetime while maintaining topology during k -lifetime slots in a highly dynamic topology scenario due to the high mobility nature of IoV node.

In this problem, we consider the design guidelines to an effective topology control algorithm as in [29]. Broadcasting by flooding is a common communication method in many network applications. However, this communication method causes a broadcast storm problem resulting in serious redundant rebroadcasts, contention, and collision [30]. Moreover, for long network connectivity, in other words, long network lifetime with reducing the energy consumption with the use of possible minimal power, WSNs consider the transmission range [31]. In a different way, IoVs also need to consider the transmission range but not for the energy consumption and network connectivity like in the WSNs, however, to avoid packet retransmission incurring by packet collisions caused by channel interference.

It is desirable to manage the network connectivity and avoid packet retransmission in highly dynamic mobility. Therefore, we need to construct a k -lifetime tree topology considering the delay time and dynamic environment with two different transmission ranges in the centralized and decentralized way. In general, there is no central authority in a wireless multihop network. In this case, we use a decentralized algorithm to construct a k -lifetime tree topology. If we have a centralized authority that collects local information and manages the network, we will use a centralized algorithm to construct a k -lifetime tree topology.

3.2. A Centralized Algorithm on k -Lifetime Tree Topology Construction for IoVs. In this section, we describe a centralized algorithm for the k -LTTC problem for IoVs. A centralized algorithm has three phases: Phase 1, partition into small clusters by Expected Cluster Size (ECS) β and greedy algorithm; Phase 2, connectivity calculation by an $\delta(u, v)$ between each vehicle during k -lifetime applying time domain; Phase 3, the k -LTTC algorithm to construct a k -lifetime tree topology by the connected dominating set. In Phase 1, we apply a partition by adapting a greedy algorithm and cluster size. In Figure 2, we establish clustering from the large-scale network region into the small-scale network for network efficiency, bandwidth efficiency, and calculation efficiency [32, 33]. We designed geographic clustering algorithm (GCA) to make a partition into several clusters.

An ECS β is the size to construct a k -hop tree by the minimum value of transmission range or moving distance (i.e., $\alpha^* = \min\{R_v, D_v\}$) during k -time slots.

$$\text{ECS} = \beta \leq 2 \times k \times \alpha^*, \quad (1)$$

where $\alpha^* = \min\{R_v, D_v\}$ and R_v is the radius of node v and D_v is the moving distance of node v . We will create the clusters by the GCA and if some clusters are greater than the ECS β , we will set up a partition cluster into a small cluster again using the GCA (Algorithm 1).

Theorem 1. ECS β has at most one cluster at $2 \times k \times \alpha^*$ in the given complete graph.

Proof. In the given complete graph, we can formulate the following equations. If $\alpha^* = R_v$, $2 \times k \times R_v \leq 2 \times k = k$ hop tree and if $\alpha^* = D_v$, $2 \times k \times D_v \leq 2 \times k \times R_v \leq 2 \times k = k$ hop tree. \square

Theorem 2. $\text{OPT} \leq \text{Min}(\text{TotalSize}/(2 \times k \times \alpha^*))$.

Proof. Each cluster has the same size with $2 \times k \times \alpha^*$. The best cluster is the one that has the biggest Euclidean distance that has the same size with $2 \times k \times \alpha^*$.

In Phase 2, we can check network connectivity by calculating a Euclidean distance between each vehicle predicting the mobility of vehicle during certain k -lifetime, as shown in Figure 3. In [24], the authors' model depicts the duration of time in network connectivity between two nodes using the motion parameters of two neighbour nodes (such as speed, moving direction, and radio propagation range). Using a different approach with [24], we construct a model to check network connectivity by applying the time domain in this problem as we assume that we can predict the mobility of vehicle nodes by Definition 8. Calculating the connected edges in this k -space-time domain in the following equation, we can find connected edges in a unified time domain, as shown in Figure 3.

$$W_{(u,v)} = \frac{\sum_{l=0}^k w_{t_l} \delta(u, v)_{t_l} \prod_{l=0}^k \varepsilon(u, v)_{t_l}}{k}, \quad (2)$$

where w_{t_l} is the network priority weight at time l , $\delta(u, v)_{t_l}$ is the Euclidean distance between the node u and the node v at time l , and $\varepsilon(u, v)_{t_l}$ is the network connection weight between the node u and the node v at time l .

$$\varepsilon(u, v) = \begin{cases} 1, & \text{if } \delta(u, v) \leq R_v, \\ 0, & \text{otherwise.} \end{cases} \quad (3)$$

We calculated a total weight $W_{(i,j)}$ by multiplying the distance weight, the network priority weight, and the network connection weight between each vehicle at each time slot. We do not consider the out-of-bounds distance of communication range between each vehicle. In Figure 3, node 2 and node 7 have a connected edge with two nodes existing in the communication range at time slot 1 and slot 2. The two nodes do not have a connected edge as they exist over the communication range at time slot 3. Therefore, this edge does not include an integrated topology as the total weight $W_{(u,v)}$ on a time union graph is 0 (Algorithm 2).

Phase 3. The k -LTTC algorithm. In this phase, we construct k -lifetime tree topology to gather the data within k -lifetime slots. We assume that it will take 1-unit time to transfer the data per one hop. Therefore, we want to construct a k -hop spanning tree to transfer data to root node from leaf nodes during k -lifetime slots. The hop-constrained spanning tree problem is well known to be a NP-hard problem [34]. Accordingly, we propose the greedy algorithm to find the k -hop spanning tree (Algorithm 3). In this

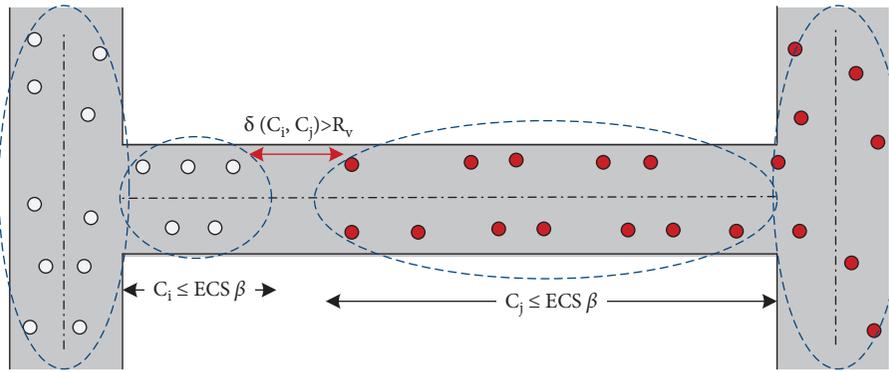


FIGURE 2: Partition in the overall region by ECS β .

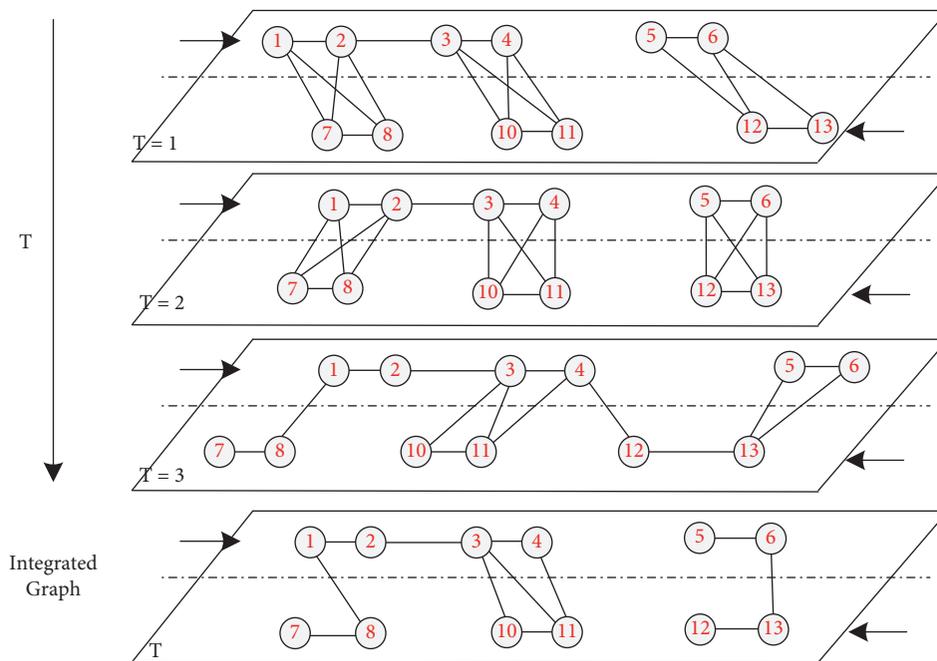


FIGURE 3: An integrated topology snapshots over the time domain.

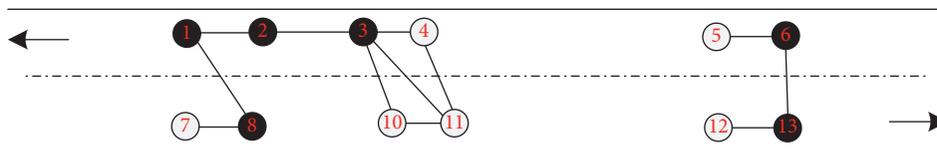


FIGURE 4: Connected dominating set in the unified graph.

algorithm, we find the dominating set converting our problem into set cover problem. Then, we find the connected dominating set using the shortest path between dominating set. If this connected path is greater than $2k$, we chose to divide the subgraph. If this connected graph is less than or equal to $2k$, we choose the root node as in equation (3). Therefore, it has the connected dominating set using the greedy algorithm in [35]. We describe a greedy algorithm to

choose a dominating set in the given set G converting from the dominating set to the set cover problem. After calculating the average distance by k -lifetime slots, we find the connected dominating set, as shown in Figure 4. In the given connected dominating set, if the number of connected dominating sets is an odd number, we will add 1 to the number of connected paths (γ). Then, we can choose a root node as the centre node, as shown in Figure 5. After checking

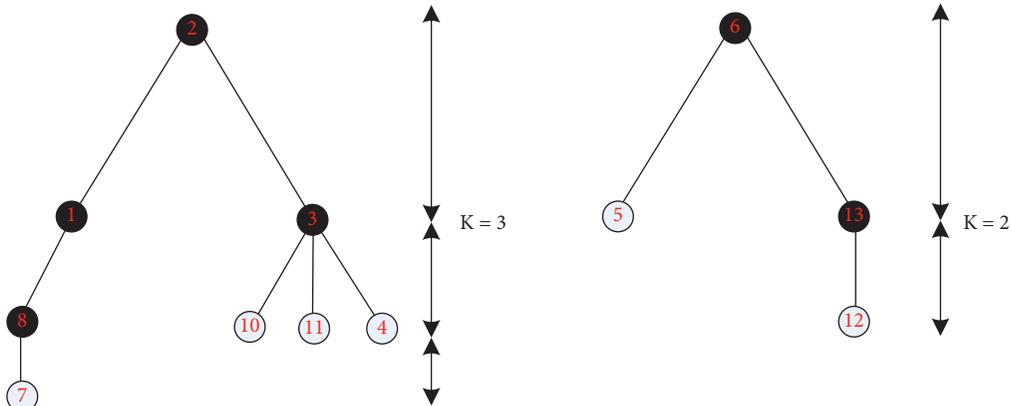
Input: A set of vehicle V^* , time constant k
Output: Cluster set C^*

- (1) $\alpha^* = \min\{R_v, D_v\}$;
- (2) $\beta = 2 \times k \times \alpha^*$;
- (3) $G_{List} = \{\delta(i, j), \delta(k, l), \dots\}$;
- (4) **for** $i = 1$ to G_{List} length **do**
- (5) $\delta(i, j) = \text{getElement}(G_{List})$;
- (6) **if** $\delta(i, j) \geq R_v$ **then**
- (7) $G_{List} = G_{List} - \{\delta(i, j)\}$
- (8) **end if**
- (9) **end for**
- (10) $EC_{List} = \text{make the beginning Clusters by disconnected edge}$;
- (11) **for** $i = 1$ to EC_{List} length **do**
- (12) $C_i = \text{getClusterElement}(EC_{List})$;
- (13) $TECS = \text{TotalSize}\{D_s, v_i(x, y) - D_e, v_i(x, y)\}$
- (14) **if** $TECS \leq \beta$ **then**
- (15) $C^* = C^* \cup \{C_i\}$
- (16) **else**
- (17) make sub-Clusters dividing by ECS β ;
- (18) $GCA(V^*, k)$;
- (19) **end if**
- (20) **end for**
- (21) Return cluster set $C^* = \{C_1, C_2, \dots, C_n\}$

ALGORITHM 1: Geographic clustering algorithm (GCA) for the k -LTTC problem.

Input: clustering set C^* , time constant k
Output: List of edge weight $W_{(i,j)}$

- (1) $C^* = \{C_1, C_2, \dots, C_n\}$;
- (2) $G_{List} = \text{Sort}\{\delta(i, j), \delta(k, l), \dots\}$;
- (3) **while** $C^* \neq \phi C$ **do**
- (4) **if** $\delta(i, j) \geq \text{CommunicationRange}$ **then**
- (5) remove Edge;
- (6) **end if**
- (7) $W_{(i,j)} = \sum_{l=0}^k W_{t_l} \delta(i, j)_{t_l} \prod_{l=0}^k \varepsilon(u, v)_{t_l} / k$;
- (8) Calculate edge weights considering time slots k at each Cluster;
- (9) **end while**
- (10) Return List of edge weight $W_{(i,j)}$;

ALGORITHM 2: A weight calculation for the k -LTTC problem.FIGURE 5: k -lifetime tree in IoVs.

```

Input: Cluster  $C_i$ , time constant  $k$ 
Output:  $k$ -hop bounded tree  $T_m$ 
(1) make subset  $S^*$  in the  $C_i$ ;
(2) while ( $S_j \in S^* \neq \phi$ ) do
(3)   Choose max number of set  $S_j$  in the subset  $S^*$ ;
(4)   set  $D^* \leftarrow D^* \cup j$ ;
(5)   set  $S^* \leftarrow S^* - j$ ;
(6) end while
(7) find connected path between  $D^*$  using shortest path;
(8)  $\gamma \leftarrow$  size of connected path;
(9) if  $\gamma > 2k$  then
(10)   divide connected path;
(11) end if
(12) if  $\gamma$  is odd number then
(13)    $v \leftarrow (\gamma + 1)/2$ ;
(14) else
(15)    $v \leftarrow \gamma/2$ ;
(16) end if
(17)   => construct the bounded  $k$ -hop tree removing cycling edge
(18) while ( $v \in G \cdot v \neq \phi$ ) do
(19)   for each adjacent vertex to ( $v$ ) to ( $u$ ) do
(20)     if  $v$  has no cycling then
(21)        $T_m = T_m \cup \{(v, u)\}$ 
(22)     end if
(23)   end for
(24)    $G \cdot v \leftarrow G \cdot v - v$ 
(25)    $v \leftarrow$  the min number of  $u$ 
(26) end while
(27)
(28) Return  $k$ -hop bounded tree  $T_m$ ;

```

ALGORITHM 3: A centralized algorithm for the k -LTTC problem.

whether a k -hop bounded graph can construct in the given graph, we choose the root node from centre node in the given connected dominating sets.

$$\text{root node} = \begin{cases} \frac{\gamma + 1}{2}, & \text{if } \gamma = \text{odd number,} \\ \frac{\gamma}{2}, & \text{if } \gamma = \text{even number,} \end{cases} \quad (4)$$

where γ is the number of connected path. Finally, we construct a k -hop spanning tree using a connected dominating set with a removing cycle (Algorithm 3). We find the bounded k -hop tree.

Theorem 3. Time complexity of the proposed algorithm is $O(VE)$.

Proof. Initially, a centralized algorithm makes subsets in a cluster. It executes as many as the number of subsets. It executes as many times as each vertex and as many as the number of neighbouring edges of each vertex. Therefore, the time complexity of the proposed algorithm is $O(VE)$. \square

3.3. A Decentralized Algorithm on k -Lifetime Tree Topology Construction for IoVs. In this section, we describe the distributed algorithm for the k -LTTC problem in highly dynamic networks. Unlike the centralized algorithm, the central node does not need to have the entire information of the networks in a decentralized method. This distributed algorithm is a more realistic algorithm than the centralized method. Our proposed algorithm is a modified version by adding time domain in order to consider the mobility of the IoV node in the existing distributed algorithm [36]. Our algorithm 4 is as follows.

In the initialization step, every one of the nodes is state 0. The random number generator generates a random number in each vehicular node at the same time duration by the synchronized time clock. Some nodes are promoted as an initiator if the generated random number is over the threshold (ρ).

In the construction step, each initiator node sends an INVITE message to neighbour nodes, when the neighbour nodes receive the INVITE message, including location information, time slot k , and level l . For M {Timestamp, $L(x_t, y_t)$, Constant k , Level l }, the neighbour node will begin calculating the distance between the

```

Input: A set of vehicle  $V^*$ , time constant  $k$ 
Output:  $k$ -hop topology tree  $T$ 
(1) Set  $rand_{num}$ ;
(2) Set  $token_{level} = 0$ ;
(3) Set  $vehicle_{location} = \{(Xt_i, Yt_i), \dots, (Xt_k, Yt_k)\}$ ;
(4) if  $rand_{num} \geq \rho$  then
(5)   Set INITIATOR;
(6)   Broadcast INVITE Message;
(7) end if
(8) if Receive “Token” then
(9)    $W_{(i,j)} = \sum_{l=0}^k \delta(i, j)_{t_l} \prod_{l=0}^k \varepsilon(u, v)_{t_l} / k$ ;
(10)  if  $Vehicle_{level} == Zero_{level}$  then
(11)    if  $W_{(i,j)} \leq CR$  and  $W_{(i,j)} \neq 0$  then
(12)      Change  $Vehicle_{level}$  from current state;
(13)      Reply JOIN Message;
(14)      Broadcast INVITE Message;
(15)    end if
(16)  else
(17)    if  $Vehicle_{level} \geq Token_{level}$  then
(18)      Discard Token;
(19)    else
(20)      Change  $Vehicle_{level}$  from Token;
(21)    if  $W_{(i,j)} \leq CR$  and  $W_{(i,j)} \neq 0$  then
(22)      Change  $Vehicle_{level}$  from current state;
(23)      Reply JOIN Message;
(24)      Broadcast INVITE Message;
(25)    end if
(26)  end if
(27) end if
(28) else if  $Time \geq \tau$  then
(29)   Set INITIATOR;
(30)   Broadcast INVITE Message;
(31) end if

```

ALGORITHM 4: A decentralized algorithm for the k -LTTC problem.

location of receiver node $L(x_t, y_t)$ and the location of the initiator or sender $L(x_t, y_t)$ and the time slot k . If the Euclidean distance ($\delta(i, j)$) between receiver node and neighbour node exists within the communication range during k -lifetime slot units, the neighbour node will join the initiator or sender node that sends the JOIN message. As the neighbour node increases the level by adding a child node, these neighbour nodes will rebroadcast an INVITE message to their neighbour nodes. If the node does not receive any INVITE message after some time (τ) duration, the node starts working as an INITIATOR. Hence, this node also sends an INVITE message to the neighbour nodes.

In the maintenance and conflict step, one node can receive an INVITE message from two neighbour nodes, as shown in Figure 6(a). In this case, the node will join a high-level node group. If any node group is the same, it will make a decision to join the group based on an earlier timestamp. The initiator node can receive an INVITE message from another initiator, as shown in Figure 6(b). As the initiator

compares the sending time, the initiator with late sending time will join the other groups and this initiator will send a discard message to the neighbour nodes. Initiator node can receive an INVITE message from child node, as shown in Figure 6(c).

In this case, if the initiator does not have any child node, the initiator will join the group, which has sent the INVITE message. After the k -LTTC, if a node receives an INVITE message from neighbour node, the node will make the decision in the following. If the node sends its data to the parent node and the level of the received node is higher than the parent node, the node will work to join from the sender node even though the node still has remaining time (k -lifetime).

Theorem 4. *Message complexity of the proposed algorithm is $2k(n-1)$.*

Proof. Each initiator node sends an invite request message to neighbour nodes. This communication counted

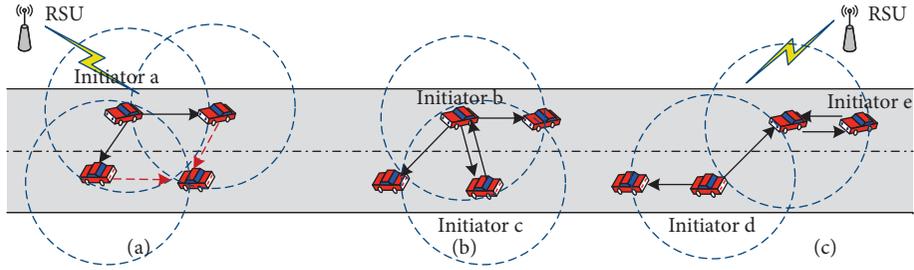


FIGURE 6: A decentralized algorithm for the k -LTTC problem.

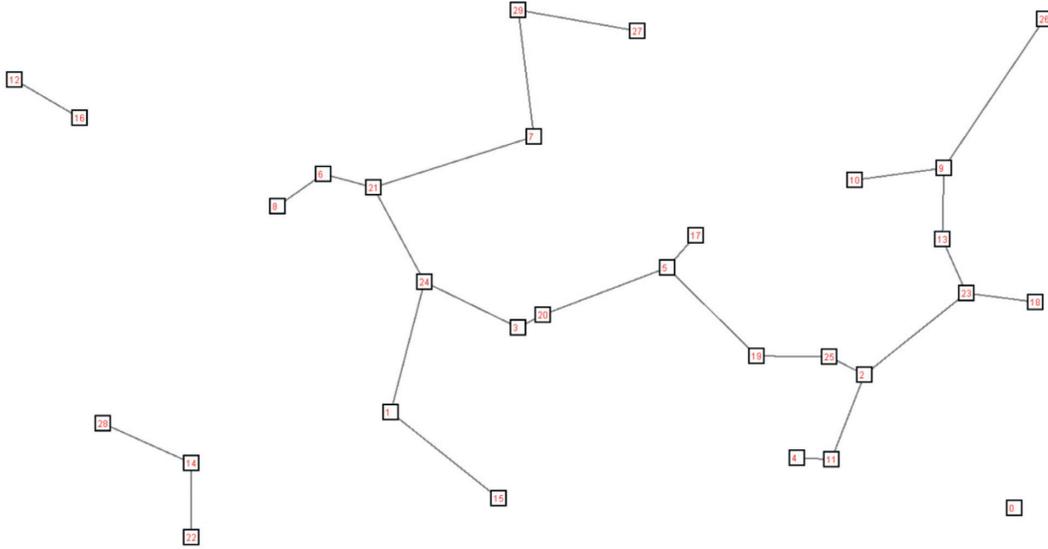


FIGURE 7: Spanning tree (30 nodes).

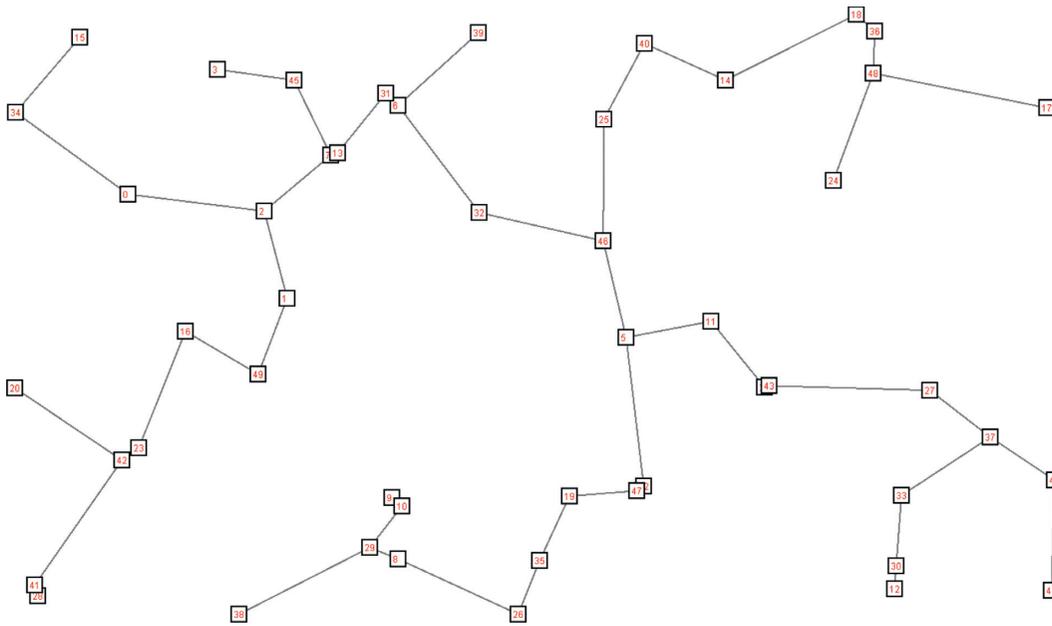


FIGURE 8: Spanning tree (50 nodes).

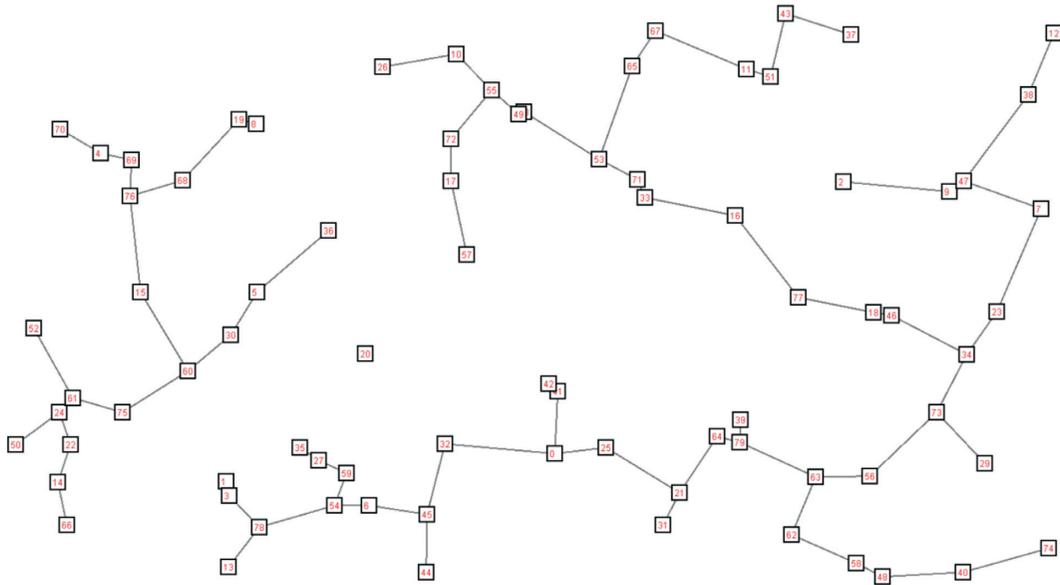


FIGURE 9: Spanning tree (80 nodes).

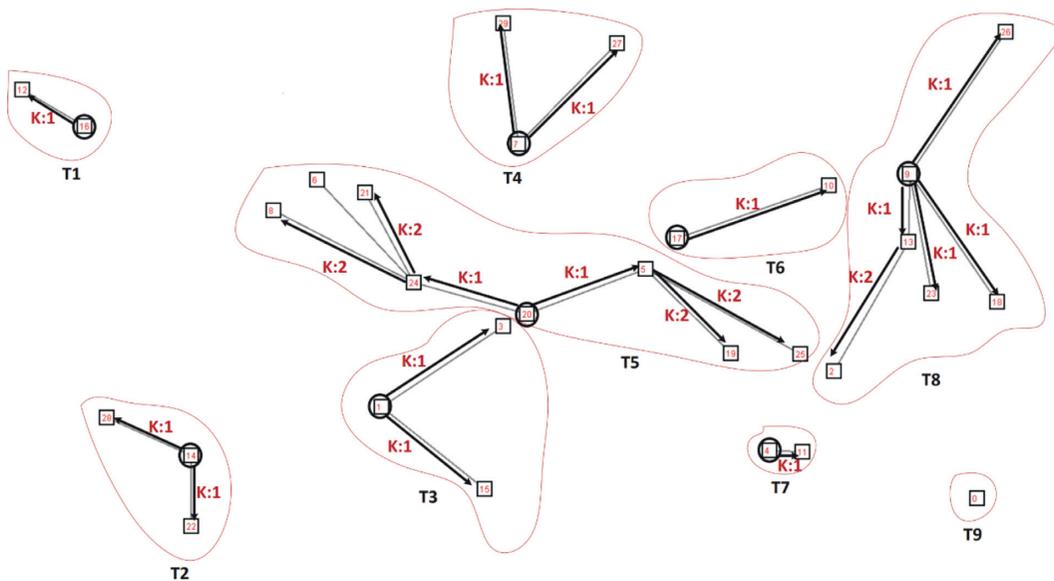


FIGURE 10: k -lifetime tree ($k=2$) (30 nodes).

$(n-1)$ messages. Neighbour nodes send an invite reply message to the initiator node. This communication counted $(n-1)$ messages. It repeats k times for k -lifetime tree construction. Therefore, the total communication message is $2k(n-1)$. \square

4. Simulation Results and Discussion

In this section, we present the simulation result to evaluate our algorithm. The simulation was implemented using Java language. This simulation was performed on an Intel

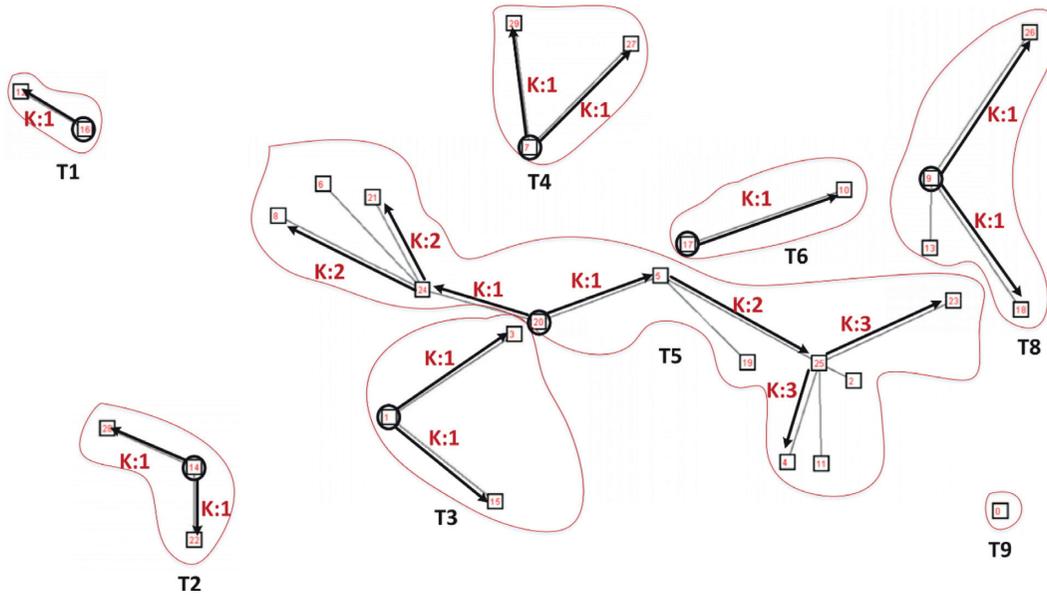


FIGURE 11: k -lifetime tree ($k = 3$) (30 nodes).

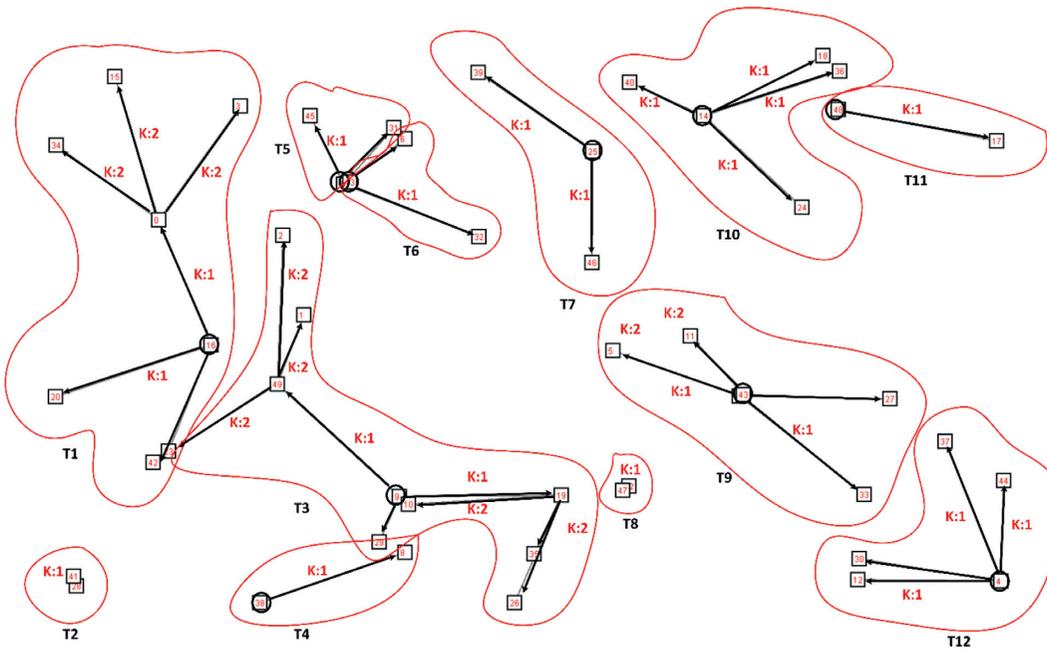


FIGURE 12: k -lifetime tree ($k = 2$) (50 nodes).

i5 core CPU, 8 GB RAM, and Windows 10 environment. The running time of the simulation is less than 1 second, as it took an average of 600 milliseconds to get results. By GCA and weight calculation algorithm, we have constructed a complete graph. Based on the complete graph,

we constructed a k -lifetime tree topology by the centralized algorithm. We compare the minimum spanning tree algorithm and k -LTTC algorithm in this simulation. Figures 7–9 represent a network topology constructed by a minimum spanning tree algorithm at 30 nodes, 50 nodes,

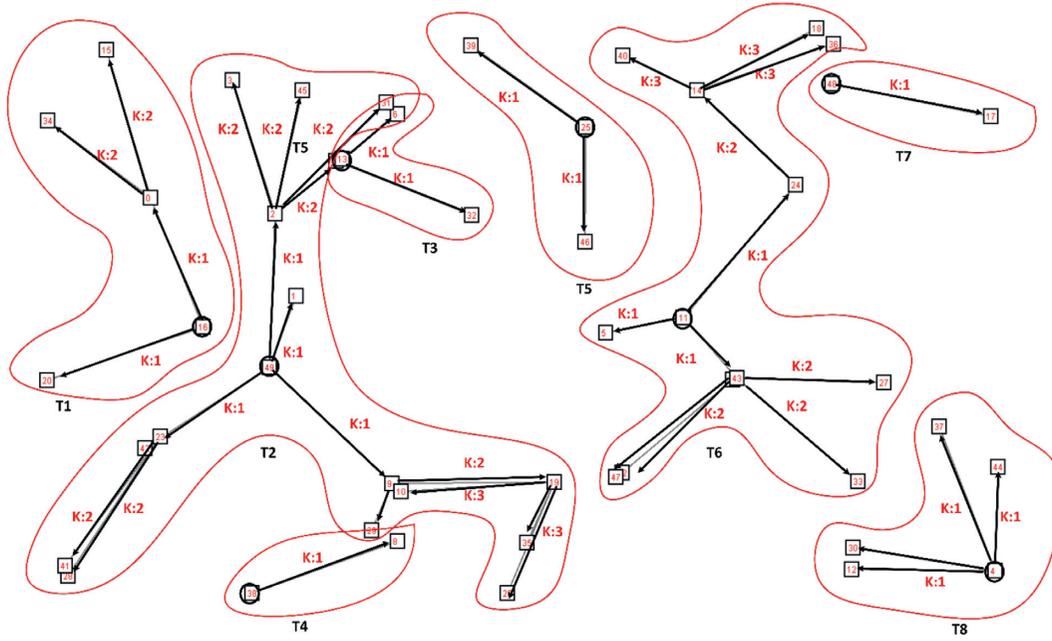


FIGURE 13: k -lifetime tree ($k=3$) (50 nodes).

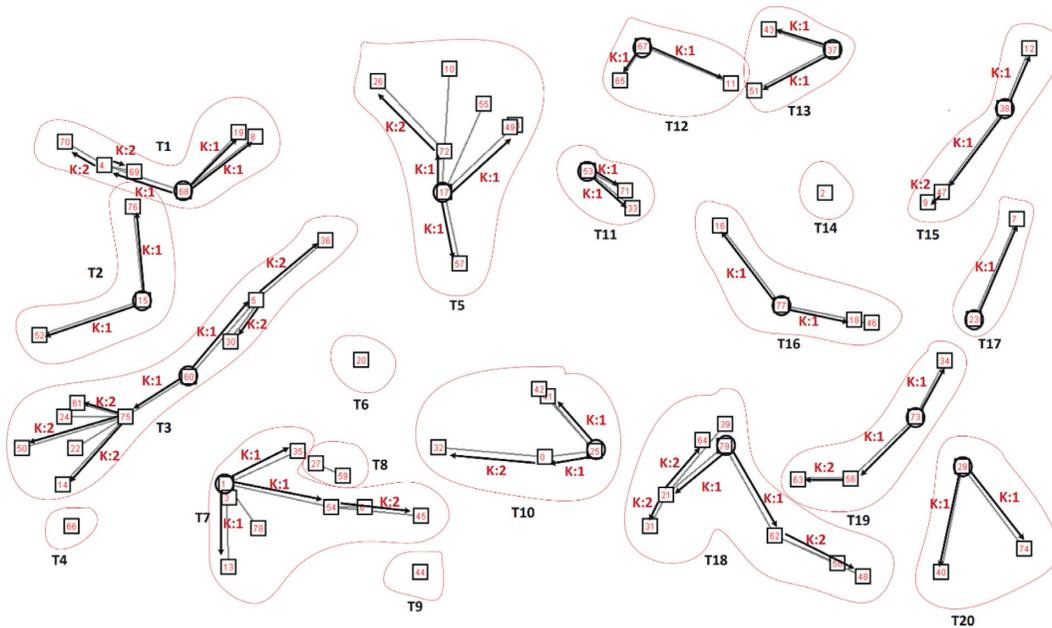


FIGURE 14: k -lifetime tree ($k=2$) (80 nodes).

and 80 nodes with a connection within the communication range (250). In this simulation, we can see that topology is constructed by not hop limitation but

communication range. Figures 10–15 represent the network topology constructed by the proposed k -LTTC algorithm with a connection within the communication

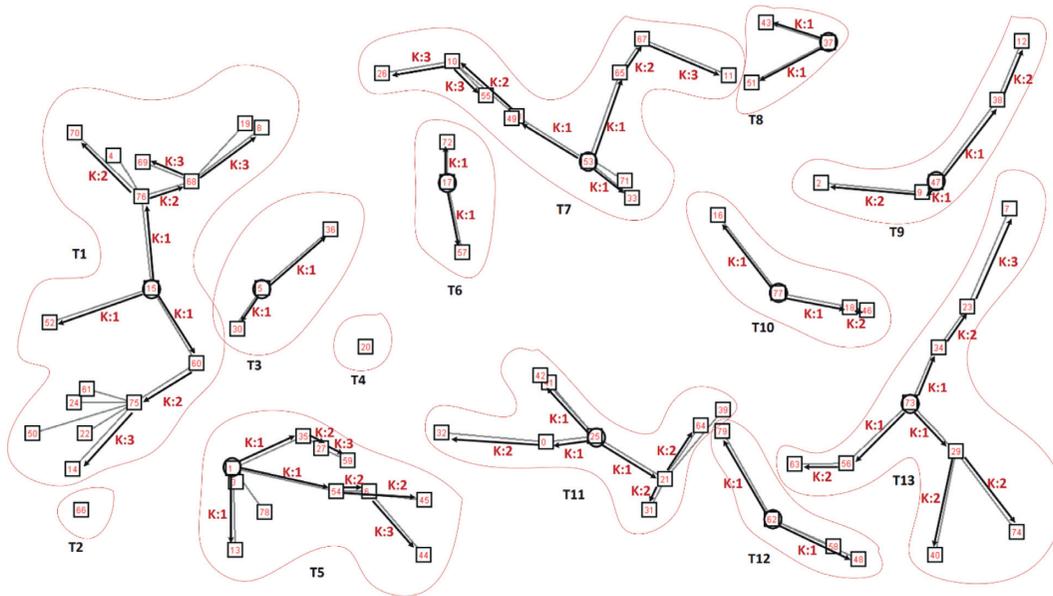


FIGURE 15: k -lifetime tree ($k = 3$) (80 nodes).

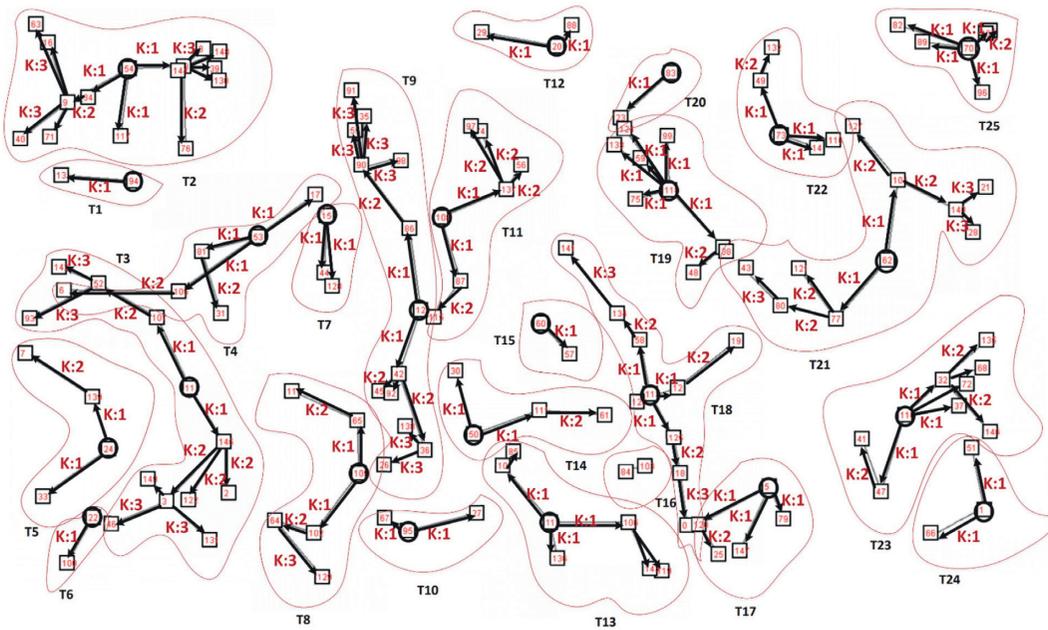


FIGURE 16: k -lifetime tree ($k = 3$) (150 nodes).

range (250). In Figures 10, 12, and 14, the connected edge is not over 4 connections due to the 2-hop limitation from root node. In other words, every constructed tree has at most 4-hops. In Figures 11, 13, 15, and 16, the connection edge is not over 6

connections by the 3-hop limitation. Generating 150 nodes randomly, Figures 16 and 17 represent network topologies constructed by the minimum spanning tree algorithm and the proposed k -LTTC algorithm with a connection within

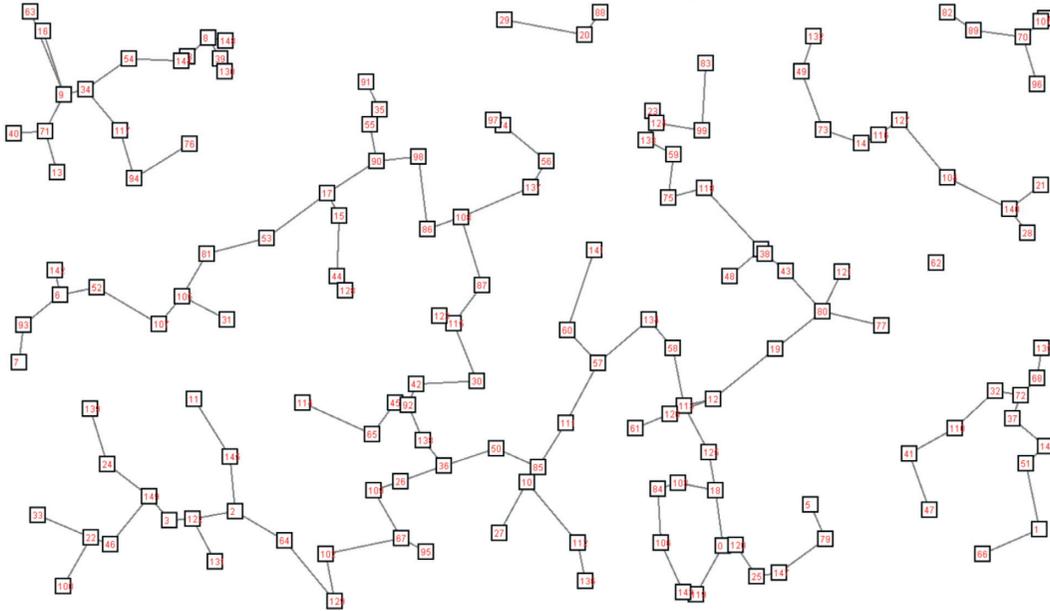


FIGURE 17: Spanning tree (150 nodes).

communication range (120). In this simulation, we can see that the limited topology is constructed by the hop limitation algorithm (Algorithms 3).

5. Conclusions

In this article, we propose the k -LTTC algorithm to consider highly dynamic networks in the characteristic of IoVs. As IoV is one category of MANET, topology control is important in order to send and receive data efficiently. To the best of our knowledge, this is the first work to adopt time slots as a mechanism to consider IoVs with high mobility. As we construct k -lifetime tree topology, this algorithm guarantees to transfer the data within k -lifetime slots and maintain topology during k -lifetime. We have proposed two different algorithms in a centralized and decentralized topology construction way. It has been shown that the proposed central algorithm has $O(VE)$ time complexity and the proposed distributed algorithm has $2k(n-1)$ message complexity. The proposed problem is the same as the minimum cost k -hop spanning tree problem (kHMST) [27, 37]. In [27], Althaus et al. proposed approximating kMST algorithm with $O(n^2)$ and Hassin et al. proposed TVC algorithm with $O(mn^2)$ in a subproblem. Our algorithm has better time complexity than the previously proposed algorithm. Furthermore, through simulation, we have verified the correctness of our proposed algorithms.

Data Availability

No data were used to support this study.

Disclosure

This paper is an extended version of the authors' paper, presented at IEEE INFOCOM 2018-IEEE Conference on

Computer Communications Workshops (INFOCOM WKSHPS) [38].

Conflicts of Interest

The authors declare that they have no conflicts of interest regarding the publication of this article.

References

- [1] F. Yang, S. Wang, J. Li, Z. Liu, and Q. Sun, "An overview of internet of vehicles," *China communications*, vol. 11, no. 10, pp. 1–15, 2014.
- [2] M. Gerla, E. K. Lee, G. Pau, and U. Lee, "Internet of vehicles From intelligent grid to autonomous cars and vehicular clouds," in *Proceedings of the 2014 IEEE world forum on internet of things (WF-IoT)*, pp. 241–246, IEEE, Seoul, Korea, March 2014.
- [3] J. Contreras-Castillo, S. Zeadally, and J. A. Guerrero-Ibañez, "Internet of vehicles. architecture, protocols, and security," *IEEE Internet of Things Journal*, vol. 5, pp. 3701–3709, 2017.
- [4] O. Kaiwartya, A. H. Abdullah, Y. Cao et al., "Internet of vehicles. Motivation, layered architecture, network model, challenges, and future aspects," *IEEE Access*, vol. 4, pp. 5356–5373, 2016.
- [5] E. K. Lee, M. Gerla, G. Pau, U. Lee, and J. H. Lim, "Internet of Vehicles. From intelligent grid to autonomous cars and vehicular fogs," *International Journal of Distributed Sensor Networks*, vol. 12, Article ID 1550147716665500, 2016.
- [6] B. Bo Yu, C. Z. Cheng-Zhong Xu, and M. Minyi Guo, "Adaptive forwarding delay control for VANET data aggregation," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 1, pp. 11–18, 2012.
- [7] G. V. Chandrasekaran, *The Networking Platform for Future Vehicular Applications*, Department of Computer Science, Rutgers University, New Brunswick, NJ, USA, 2008.
- [8] K. Paridel, J. Balen, Y. Berbers, and G. V. V. ID. Martinovic, "A delay tolerant data dissemination architecture for VANETS

- using V2V and V2I communication,” in *Proceedings of the Second International Conference on Mobile Services, Resources, and Users*, MOBILITY 2012, Venice, Italy, October 2012.
- [9] J. L. Lee, D. Kim, L. Fan, and H. J. Chang, “Barrier-coverage for city block monitoring in bandwidth sensitive vehicular adhoc networks,” in *Proceedings of the 2014 10th International Conference on Mobile Ad-hoc and Sensor Networks (MSN)*, pp. 80–87, IEEE, Maui, HI, USA, December 2014.
 - [10] G. Karagiannis, O. Altintas, E. Ekici et al., “Vehicular networking: a survey and tutorial on requirements, architectures, challenges, standards and solutions,” *IEEE Communications Surveys & Tutorials*, IEEE, vol. 13, no. 4, pp. 584–616, 2011.
 - [11] A. D. Joseph, A. R. Beresford, J. Bacon et al., “Intelligent transportation systems,” *IEEE Pervasive Computing*, vol. 5, no. 4, pp. 63–67, 2006.
 - [12] G. Dimitrakopoulos and P. Demestichas, “Intelligent transportation systems,” *IEEE Vehicular Technology Magazine*, vol. 5, no. 1, pp. 77–84, 2010.
 - [13] Z. Cai, X. Zheng, and J. Yu, “A differential-private framework for urban traffic flows estimation via taxi companies,” *IEEE Transactions on Industrial Informatics*, vol. 15, no. 12, pp. 6492–6499, 2019.
 - [14] J. Wang, Z. Cai, and J. Yu, “Achieving personalized \$k\$-Anonymity-Based content privacy for autonomous vehicles in CPS,” *IEEE Transactions on Industrial Informatics*, vol. 16, no. 6, pp. 4242–4251, 2020.
 - [15] Y. Huang, X. Guan, Z. Cai, and T. Ohtsuki, “Multicast capacity analysis for social-proximity urban bus-assisted VANETs,” in *Proceedings of the 2013 IEEE International Conference on Communications (ICC)*, pp. 6138–6142, Budapest, Hungary, June 2013.
 - [16] Z. Xiong, Z. Cai, Q. Han, A. Alrawais, and W. Li, “Adgan. protect your location privacy in camera data of auto-driving vehicles,” *IEEE Transactions on Industrial Informatics*, vol. 17, pp. 6200–6210, 2020.
 - [17] K. Ibrahim and M. C. Weigle, “Optimizing CASCADE data aggregation for VANETs. Mobile ad hoc and sensor systems,” in *Proceedings of the 5th IEEE International Conference on 2008 MASS 2008*, pp. 724–729, IEEE, Louis, Missouri, USA, April 2008.
 - [18] J. Nzouonta, N. Rajgure, G. Guiling Wang, and C. Borcea, “VANET routing on city roads using real-time vehicular traffic information,” *IEEE Transactions on Vehicular Technology*, vol. 58, no. 7, pp. 3609–3626, 2009.
 - [19] Y. Zhu, Q. Zhao, and Q. Zhang, “Delay-constrained data aggregation in VANETs,” *IEEE Transactions on Vehicular Technology*, vol. 64, no. 5, pp. 2097–2107, 2015.
 - [20] Y. Yuan, J. Luo, W. Yan, T. Zhao, and S. Lu, “DA2RF. a data aggregation algorithm by restricting forwarders for VANETs,” in *Proceedings of the 2014 International Conference on Computing, Networking and Communications (ICNC)*, pp. 393–397, IEEE, Honolulu, HI, USA, February 2014.
 - [21] B. B. Xuan, A. Ferreira, and A. Jarry, “Computing shortest, fastest, and foremost journeys in dynamic networks,” *International Journal of Foundations of Computer Science*, vol. 14, no. 02, pp. 267–285, 2003.
 - [22] A. Ferreira, “Building a reference combinatorial model for MANETs,” *IEEE Network*, vol. 18, no. 5, pp. 24–29, 2004.
 - [23] H. M. Bernd Thallner and H. Moser, “Topology control for fault-tolerant communication in highly dynamic wireless networks,” in *Proceedings of the Third International Workshop on Intelligent Solutions in Embedded Systems, 2005*, pp. 89–100, Hamburg, Germany, May 2005.
 - [24] W. Su, S.-J. Lee, and M. Gerla, “Mobility prediction in wireless networks,” in *Proceedings of the MILCOM 2000. 21st Century Military Communications Conference Proceedings*, vol. 1, pp. 491–495, Los Angeles, CA, USA, October 2000.
 - [25] Z. Zhigang Wang, L. Lichuan Liu, M. MengChu Zhou, and N. Ansari, “A position-based clustering technique for ad hoc intervehicle communication. Systems, man, and cybernetics, Part C. Applications and reviews,” *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 38, no. 2, pp. 201–208, 2008.
 - [26] S. Srivastava and R. K. Ghosh, “Distributed algorithms for finding and maintaining a k -tree core in a dynamic network,” *Information Processing Letters*, vol. 88, no. 4, pp. 187–194, 2003.
 - [27] E. Althaus, S. Funke, S. Har-Peled, J. Könemann, E. A. Ramos, and M. Skutella, “Approximating k -hop minimum-spanning trees,” *Operations Research Letters*, vol. 33, no. 2, pp. 115–120, 2005.
 - [28] A. Ghosh, O. D. Incel, V. S. A. Kumar, and B. Krishnamachari, “Bounded-degree minimum-radius spanning trees for fast data collection in sensor networks,” in *Proceedings of the INFOCOM IEEE Conference on Computer Communications Workshops, 2010*, pp. 1–2, San Diego, CA, USA, March 2010.
 - [29] N. Li, J. C. Hou, and L. Sha, “Design and analysis of an MST-based topology control algorithm,” vol. 3, pp. 1702–1712, in *Proceedings of the INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications*, vol. 3, IEEE Societies, San Francisco, CA, USA, March 2003.
 - [30] Y.-C. Tseng, S.-Y. Ni, Y.-S. Chen, and J.-P. Sheu, “The broadcast storm problem in a mobile ad hoc network,” *Wireless Networks*, vol. 8, no. 2/3, pp. 153–167, 2002.
 - [31] Z. Cai and Q. Chen, “Latency-and-Coverage aware data aggregation scheduling for multihop battery-free wireless networks,” *IEEE Transactions on Wireless Communications*, vol. 20, no. 3, pp. 1770–1784, 2021.
 - [32] K. Kaixin Xu, X. Xiaoyan Hong, and M. Gerla, “An ad hoc network with mobile backbones,” *Communications in Proceedings of the IEEE International Conference on ICC 2002*, vol. 5, pp. 3138–3143, New York, NY, USA, April 2002.
 - [33] S. Banerjee and S. Khuller, “A clustering scheme for hierarchical control in multi-hop wireless networks,” vol. 2, pp. 1028–1037, in *Proceedings of the IEEE INFOCOM 2001 Twentieth annual joint conference of the IEEE computer and communications societies*, vol. 2, pp. 1028–1037, IEEE, Anchorage, AK, USA, April 2001.
 - [34] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman & Co., New York, NY, USA, 1979.
 - [35] S. Guha and S. Khuller, “Approximation algorithms for connected dominating sets,” *Algorithmica*, vol. 20, no. 4, pp. 374–387, 1998.
 - [36] P.-J. Wan, K. M. Alzoubi, and O. Frieder, “Distributed construction of connected dominating set in wireless ad hoc networks,” *Mobile Networks and Applications*, vol. 9, no. 2, pp. 141–149, 2004.
 - [37] R. Hassin and A. Levin, “Minimum spanning tree with hop restrictions,” *Journal of Algorithms*, vol. 48, no. 1, pp. 220–238, 2003.
 - [38] J.-L. Lee, J. Hwang, H. Park, and D. Kim, “On latency-aware tree topology construction for emergency responding VANET applications,” in *Proceedings of the IEEE INFOCOM 2018-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pp. 57–63, Honolulu, HI, USA, April 2018.