

RESEARCH ARTICLE

BusCache: V2V-Based Infrastructure-Free Content Dissemination System for the Internet of Vehicles

ABDENACER NAOURI¹, NABIL ABDELKADER NOURI²,
AMAR KHELLOUFI¹, ABDELKARIM BEN SADA¹, SALIM NAOURI¹,
HUANSHENG NING¹, (Senior Member, IEEE),
AND SAHRAOUI DHELMIM³

¹School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing 100083, China

²Department of Mathematics and Computer Science, University of Djelfa, Djelfa 17000, Algeria

³School of Computer Science, University College Dublin, Dublin 4, D04 V1W8 Ireland

Corresponding author: Sahraoui Dhelim (sahraoui.dhelim@ucd.ie)

ABSTRACT Internet of Vehicles (IoV) offers many services aiming to enhance the safety and comfort of drivers and passengers such as accident alarms, congestion avoidance, and multimedia, entertainment applications. Cooperatively sharing and retrieving large-scale files in IoV is a challenging task due to the extremely volatile nature of IoV. Therefore, it is paramount to develop a vehicle-to-vehicle (V2V) content delivery mechanism that can adapt to IoV communication requirements. One of the challenging tasks in this regard is to locate content pieces and gather information about peers. Previous proposed systems broadcast beacon messages to the whole network to locate certain content pieces, which consume the bandwidth and limit the network resources. To overcome these issues, in this paper, we propose BusCache, a traffic-aware content delivery system for IoV. BusCache uses buses as trackers for the content distribution on the overlay network, due to their availability and unique characteristics (predefined trajectories and time). Instead of flooding the entire network, a request to the bus is enough to determine the piece's location. We first propose a clustering strategy to control the data followed between peers without relying on infrastructure, which reduces the end-to-end delay network between the receiving peer and the tracked node. Then we introduce a peer selection mechanism that selects the most appropriate peers that will deliver the desired content by taking into account their velocity, destination, and other traffic-related factors. Finally, we proposed a policy-based content selection algorithm that identifies and prioritizes the scarce content that is present in just a few vehicles. Simulation results show that BusCache can reduce content lookup time and content dissemination compared to state-of-the-art approaches.

INDEX TERMS Peer selection algorithm, content dissemination, V2V dissemination, Internet of Vehicles (IoV), quality of experience (QoE).

I. INTRODUCTION

Vehicular networks are considered the main pillar of the Internet of Vehicles (IoV), which offers many services: improving the safety of drivers and passengers by offering traffic information such as road traffic, accidents events, location, and even meteorological information, and also

The associate editor coordinating the review of this manuscript and approving it for publication was M. Anwar Hossain¹.

improve passenger comfort by providing mobile internet access, chat, online gaming and more. Moreover, improves the overall efficiency of the road network, by reducing travel time and congestion [1], [2]. Many applications oriented to IoV include using vehicles as a mobile sink, where sensors are integrated into a vehicle that can detect and save events in advance, the vehicles can subsequently exchange this data between each other or with infrastructure to process them [3], [4]. However, a vehicle can receive the same

information from different sources. Therefore, the use of architectures and protocols will be necessary to reduce the rate of data exchange and avoid data redundancy [5]. The recent advances in IoV made the vehicles not just traveling medium, vehicles able to communicate with each other using wireless dedicated short-range communication (DSRC). Vehicles can communicate with infrastructure via vehicle-to-infrastructure (V2I), and with each other via vehicle-to-vehicle (V2V) directly or via multi-hop links, Intermediate nodes between source and target node can hold and forward data based on the network status, using the carry-and-forward mechanism [6], [7]. Due the network instability, controlling data traffic between nodes is considered as big challenge where necessary to define efficient mechanisms to deliver data on time by reducing the data delivery delay [8].

Conventional P2P content-sharing systems adopt fully distributed and largely unstructured overlays, where they can perform complex queries more efficiently than structured overlays and it is not subject to constraints which makes it easy to handle easily network churn, Also easy to build and scale up [9]. Most of P2P content-sharing systems target wired networks, and it is difficult to adapt these systems to IoV networks without modification, due to IoV characteristics, which feature node speed, unpredictable routes, and high dynamics topologies [10]. Using structured protocols for sharing content in IoV is impractical, especially when traffic is unstable due to the high network churn, which leads to an increase the maintenance cost and fault lookups. IoV high mobility introduces frequent lookup failures. Due to IoV network instability, unstructured protocols can be more suitable for IoV. Therefore, In this paper, we propose BusCache, a traffic-aware content delivery system for IoV. BusCache leverages buses as trackers for content distribution on the overlay network. The scenario depicted in the Figure 1 illustrates a densely populated urban area with heavy traffic congestion, where numerous buses are operating on the road network. In this setting, cars rely on these buses as a means to retrieve data, possibly through a vehicular ad hoc network (VANET) or a similar communication infrastructure. The scenario aims to examine how vehicles utilize buses as intermediaries for data retrieval, highlighting the dynamics of information exchange in urban environments characterized by high vehicle density and traffic congestion.

BusCache mainly tries to compensate the trackers in wired networks by presenting a set of algorithms based on certain advantages of public transport nodes' "Buses" characteristics in IoV, such as predefined trajectory and schedule time as stable speed and direction predictability to maintain the constancy of the overlay and reduce the content dissemination time between vehicles. A request to the bus is enough to determine the piece's location, without needing to flood the entire network. Our contributions can be summarized as follows:

- We propose a clustering strategy to control the data flow between IoV nodes without relying on infrastructure, which reduces the end-to-end delay network between the

receiving peer and the tracked node. We leverage buses as trackers for providing better data content exchange and reducing content dissemination time

- We develop a peer selection mechanism that selects the most appropriate peers that will deliver the desired content by taking into account their velocity, destination, and other traffic-related factors.
- We proposed a policy-based content selection algorithm that identifies and prioritizes the scarce content that is present in just a few vehicles.

The rest of the paper is organized as follows: In Section II, we reviewed the literature on content dissemination in the context of IoV and social vehicular networks. In Section III, we presented the assumption and main components of BusCache architecture. Section IV details the system modeling of the proposed system. While in Section VI, we presented the experimental evaluation and discussed the obtained results. Finally, we concluded and gave some future work in Section VII.



FIGURE 1. Interdependence between cars and buses in congested environments.

II. RELATED WORKS

Peer-to-peer network working on control and load balancing the data between peers in the overlay, aiming to reduce the lookups delay and provide local content without relying on infrastructure, different protocols and strategies proposed in internet network categorized into structured and unstructured protocols [11].

In unstructured protocols, where the overlay links between peers are established arbitrarily, in such kind of networks, if a peer looks for a specific piece of data in the network, must flood the network by sending a query message to all peers to find as many peers as possible that share the target data, with unstructured networks there is no guarantee the queries sent by peers to be resolved, due the different repatriation of pieces between peers [12]. Certain content may be accessible when it is available across multiple peers, while other content may remain inaccessible due to its absence. When data is shared by only a few peers, additionally, there is no correlation between the peers and the content they manage. Moreover, flooding the network by sending queries increases the traffic in the network. In structured protocols, the overlay links between peers are established based on distributed hash table DHT, Each peer in structured P2P networks is responsible for specific data, hash function is defined as a key used to assign data to each peer, where the peer can locate the content at which peer the data associate by following protocol strategy on limited time. Even structured P2P networks overcome

issues related to unstructured networks, such as reducing the amount of traffic and bandwidth consumption. [13], [14].

Elsayed et al. [5] proposed Prediction-Assisted Cooperative Content Discovery (PACD) scheme. PACD uses the fixed and mobile nature of parked and moving vehicles, respectively, to promulgate cached content into the network using bloom filters. PACD uses such information to offer a cooperative cache discovery to locate nearby replicas to the requester by dynamically predicting the location of caching vehicles. The Any Relation Clustering Algorithm (ARCA) is used to cluster trips based on their route similarity through the XXDice similarity coefficient. Each cluster is then trained using the Mixture Transition Distribution-Probit (MTD-Probit) model to predict the rest of the vehicle's trajectory. Wang et al. [15] introduced Bus-Based Content Offloading (BBCO), an algorithm that is designed to enhance the volume of content transferred from buses to sedans, while also ensuring fairness in offloading opportunities for each sedan. BBCO algorithm forecasts the number of buses a sedan will encounter along its future route and estimates the transmission rate. By scheduling content offloading for sedans on a slot-by-slot basis, the BBCO algorithm maximizes the volume of offloaded content and ensures equitable opportunities for offloading. Zhang et al. [16] Explored the context of advertising dissemination within vehicular environments, where buses function as originating nodes for broadcasting advertisements and private vehicles relay the received advertisements via vehicle-to-vehicle communication. They introduced a trajectory-based advertising distribution approach named Bus-Ads, employing coalition games to direct the sharing process among private vehicles. Bus-Ads facilitates advertisement transmission by buses and private vehicles, while effectively mitigating significant interference. Xue et al. [17] introduced a two-layer distributed content caching strategy for VANETs, leveraging caches in both vehicles and roadside units (RSUs). Their approach addresses the content caching challenge by aiming to minimize total transmission delay and cost, formulated as a nonlinear integer programming (NLIP) problem. They present an Alternate Dynamic Programming Search (ADPS) algorithm to tackle this problem, which breaks it down into three subproblems and employs dynamic programming (DP) to solve each one independently. Additionally, to manage the complexity of large-scale scenarios, they propose a Cooperation-Based Greedy (CBG) algorithm as an efficient solution. Chaib et al. [18] introduce a novel routing approach named Bus-based Routing Technique (BRT) that leverages the predictable movement patterns of buses to estimate the required transmission time (temporal distance) for data delivery to Road-Side-Units (RSUs) via a dedicated bus-based network backbone. BRT consists of two main phases: Learning Phase: This phase is performed once to enable buses to construct routing table entries and anticipate delays for routing data packets via buses. Data Delivery Phase: Utilizing the pre-learned temporal distances, this phase routes data packets through the bus backbone towards

an RSU (backbone mode). BRT employs other vehicle types to enhance data packet routing and includes a maintenance procedure to handle unexpected situations such as a missing next-hop bus, ensuring uninterrupted data packet routing within the network. However, the above-mentioned protocols considered the content to be cached in a single source, which makes it difficult to download. Unlike BusCache which leverages a peer-to-peer paradigm that makes it possible to effectively obtain the content from multiple peers, which makes it resilient to node disconnections and communication interactions.

III. GENERAL OVERVIEW

In this section, we will outline BusCache basic architecture. Content dissemination in high dynamics topology is a challenging task. BusCache exploits the advantage of public transport buses to use them as trackers for content in their covered area, whereas buses are used as a trading mechanism, if a vehicle is interested in specific content, it must share its resources with other content-consuming vehicles. Vehicles can start exchanging content with each other in a P2P fashion using a blind flooding mechanism to look for specific content, but this strategy could overload and consume the network resources, also add more delay, so using buses as trackers for target data will reduce the consumption of bandwidth and delay content dissemination process. BusCache distinguishes a two-layer architecture of nodes (Buses, content consumer vehicles), where Buses reference to the upper layer (which consists of a set of buses) and the low layer consists of content consumer vehicles. BusCache aims to realize a fully distributed system to share or download data between ordinary vehicles on P2P communications without infrastructure intervention. When a vehicle registers in a bus, the bus forwards its id and its content to other peers in the vicinity that are registered with that bus. To start content dissemination, the content consumer vehicle needs to select the close vehicle based on different metrics such as(upload/download data rate, and close peers on term distance "hop count"). Vehicles can leave/join the network initiated by the bus. To overcome this issue, it's necessary to update the register information of the participating vehicles periodically to maintain the consistency of the network. Buses can share the same segment or appear on the same junction, in such a scenario, buses exchange their tracking information with each other and aggregate it locally, which will expand the view of the interested vehicles by increasing their knowledge about content distribution, with keeping download data via information provided by a new bus in their direction.

The high-tier nodes (buses) during service hours broadcast beacon messages in their covered area, whereas ordinary vehicles of low-tier receive several beacons from different high-tier nodes. When an interested vehicle receives an active beacon from buses, it should be registered among the adequate buses. If a vehicle loses connection with its currently registered bus, it needs to switch its registration to another

bus to keep tracking the data in the P2P overlay network. Choosing a bus to register can be based on different metrics as the closest bus to the peer or the bus that has the most corresponding trajectory to the interested peer, so such a decision will relate to the road nature, whereas in urban areas the interested peer will register with bus has similar trajectory to the bus and in highway areas, the registration will base on distance from the tracked node.

Registration among the tracker nodes (buses) serves a critical function in locating shared content for other consumer vehicles nearby. Without this registration process, vehicles would have to flood the network each time they attempt to locate desired data, resulting in excessive network traffic and potential delays. This not only wastes resources but also increases the delay. Although the registration process is important, there are some issues remain to be solved. For instance, if there are multiple buses near a vehicle, which one should be selected, and what metric should be based on to make such a decision? If there is no bus nearby, what a vehicle should do? In addition, which peer should be downloaded from it and which pieces should be downloaded first; we will discuss these issues in section IV.

IV. SYSTEM MODELING

1) MODEL DESCRIPTION

Traffic roads are generally composed of frequented public transport and a high number of common vehicles, public transport vehicles contribute to increasing the connectivity between vehicles, and based on their predefined trajectory and fixed time, we use them to download content in a P2P fashion, we exploit those characteristics to propose our model where: Let $B = b_0, b_1, b_2 \dots b_n$ represent a set of buses traveled in metropolitan areas, with predefined trajectory, every bus has road trajectory traversed among his service, and $T = t_0, t_1, t_2 \dots t_n$ represent the trajectories of B respectively. Every trajectory if defined by a set of segment S , for example, predefined segments of trajectory b_0 expressed by $s_0 = s_0, s_1, s_2 \dots s_m$, Accompanied by the scheduled time of the bus in a specific area. Buses can share the same road segment. And let $V = v_0, v_1, v_2 \dots v_n$ denote content consumer vehicles that will cooperate in a distributed manner to disseminate the content. Content consumer vehicles register with the bus in their vicinity by providing $(V_{id}, SP, Dr, hops, LP, BW, Live)$, where V_{id} : represent the id of the vehicles, SP represent the speed, Dr is vehicle direction, $hops$: reference to number of hops between peer and bus, LB : represent the list of pieces that vehicle have and $live$: reference to the presence of the vehicle, BW : is the bandwidth.

A. VEHICLE REGISTRATION

Content dissemination in IoV networks relies on nodes being within the same communication range. As nodes come into close proximity within this range, data exchange commences. However, connectivity is lost once nodes move beyond this communication range. In BusCache nodes can exchange

content with each other within the same communication range or outside it using multi-hop communication, the connectivity between nodes outside the communication range is evaluated by measuring the message delivery delay and network density, if the message delivery delay exceeds a given threshold T , the nodes consider as disconnected. Content consumer vehicles can register among the bus via one hop “share the same communication range” or via multi-hop. The buses serve as the backbone of the exchange process between peers, forwarding information to subscribed vehicles in the P2P overlay. This information pertains to other vehicles in the vicinity. Therefore, it is necessary for the buses, acting as ‘trackers,’ to continuously update information about all participating vehicles in the network. This requires defining a mechanism to maintain references to all peers on time. Messages transferred between vehicles and buses are utilized to evaluate connectivity. Each vehicle periodically sends a report about its state to the tracked node. Additionally, buses check their peers by sending ping messages. If the delivery delay of a ping message exceeds a predefined threshold (T), the vehicle is considered disconnected. Otherwise, the tracked node updates its reference information about participating peers.

Furthermore, the message delivery delay in IoV has two phases (carry, and forward). The carry phase is the time that a node carries until forward message to the next hop when be in its communication range, and the concern-forwarding phase is the transmit time when nodes are close enough to each other, the message delivery delay can be expressed in terms of forwarding delay D_f and carrying delay D_c . When the content consumer vehicles “peers” within the same communication range R with the tracked node Bus, the delay delivery message is negligible, our concern is about distant vehicles, where vehicles outside the range communication of the tracked node. Hence, to evaluate the delivery delay of remote vehicles, we need to estimate the D_f and D_c along the delivery path from the source node to the destination node.

To ensure network consistency and accurate data look-ups among peers, peers periodically transmit relevant information about their status (including velocity, distance, and remaining pieces) to the tracked node. However, this solution is impractical in IoV networks due to their characteristics. It would consume network resources, increase delay, and make maintaining stable routing paths difficult. To address these challenges, we propose a clustered structure that divides the network into sub-networks, enhancing stability. As an example, we consider an urban traffic scenario illustrated in Figure 2. In this scenario, edges represent junction points $\langle A, B, C, D, E, F, G, H, I \rangle$, with each segment accommodating traffic, including buses (representing tracked nodes), content consumer vehicles (peers), and content provider vehicles (seeds). Clusters are formed by content consumer vehicles within each segment based on the similarity of buses’ trajectories, creating a vehicular overlay network. Each cluster is overseen by a super node (super peer)

responsible for data retrieval and updates to/from the bus. Buses follow predefined trajectories for a defined duration, encountering intersection points and sharing road segments. For instance, segments $\langle B, E \rangle$ and $\langle F, I \rangle$ are shared by buses (1, 2, 3), allowing buses to appear simultaneously in the same segment. Additionally, at junction points $\langle B \rangle < 1.2 \rangle$, $\langle F \rangle < 1.3 \rangle$, $\langle H \rangle < 3.2 \rangle$, buses may intersect.

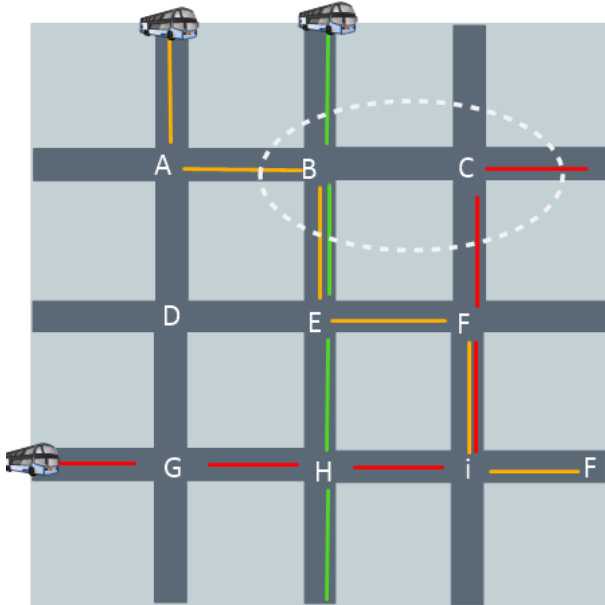


FIGURE 2. Urban road networks.

B. PEER SELECTION

Instead of transmitting messages separately between the content consumer and the bus on the same trajectory, or vice versa, which consumes excessive bandwidth and network resources, the cluster structure features aim to reduce end-to-end network delay and minimize bandwidth usage. Within this structure, a super-peer is selected from among the peers to manage data transfer between the source and destination nodes. The super peer periodically sends status updates about content providers (CPs) directly to the tracked node if it is within communication range, or to the nearest super peer otherwise. This process is depicted in Figure 3. BusCache utilizes trajectory metrics to establish more stable clusters, ensuring their longevity.

C. CLUSTER CREATION

Based on the tracked node's overview of its P2P overlay network, clusters are established using status information retrieved from this node. This data, which includes speed, location, and direction, serves as the basis for creating clusters. Once a peer connects with others in its peer list to initiate trading processes, periodic exchanges of status information occur among peers to maintain cluster stability. Peers with similar movement patterns are selected to join the same cluster. In the P2P overlay, the criteria for selecting a

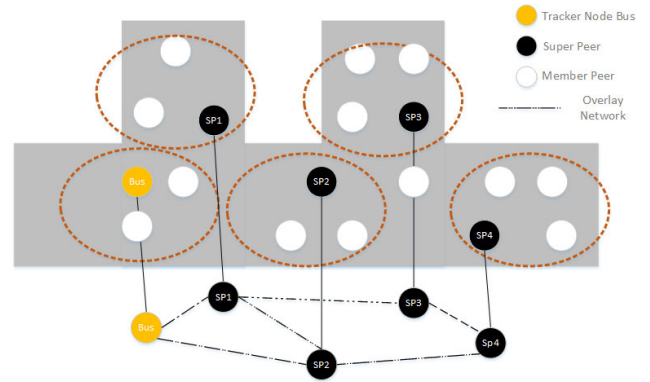


FIGURE 3. Peer clustering process.

super-peer should be based on the movement pattern of the tracked node, specifically “the bus.”

BusCache uses a utility function algorithm to control the selecting process of a cluster head among a cluster, this strategy is based on different metrics (velocity, distance, connectivity), the peer with the closet average of position and velocity to other peers along the connectivity level, it's suitable to act as super peer. Each peer in the P2P overlay receives status information about peers in its communication range, which can evaluate the utility of each peer, the peer with high utility is selected as a super peer. The probability that peers follow the same direction in a straight line is bigger than the probability at intersections.

We take into consideration the impact of vehicle trajectories by leveraging tracked information. This allows us to divide the cluster formation process into two cases: before and after intersections, where vehicles share the same road segment for a duration T . We then connect the selection process of the super-peer to the respective road segment, as depicted in Figure 3.

Various metrics are taken into account, including velocity, distance, connectivity, and trajectory. Each peer calculates a weight value WP_i based on network status information and shares it with other peers through broadcasting. The peer with the highest WP_i is chosen as the super peer.

$$WP_i = PCL_i + PAV_i + PAD_i + PTR_i \quad (1)$$

where PCL_i is the peer's connectivity level, PAV_i is the average velocity, PAD_i is the average distance and PTR_i is the trajectory similarity.

The peer connectivity level is computed as shown in (2)

$$PCL_i = \sum_j^p PS(i, j, t). \quad (2)$$

$$PS(i, j, t) = \begin{cases} 1 & \text{if } j \text{ connected to } i \\ 0 & \text{else} \end{cases} \quad (3)$$

where i is the target peer and j is the neighbor peer, $PS(i, j, t)$ is equal to 1, if j is connected to target i at time T , else is equal to 0, where there is no connectivity.

Peer average velocity is computed as shown in (4). Periodically, each peer will have information about all peers within its vicinity and hence will calculate its average velocity difference PAV from all other vehicles as:

$$PAV = \sum_{j=1}^P \frac{|V_i - V_j|}{|NP|} \quad (4)$$

where j is the neighbor of peer i under the same communication range, V_i and V_j represent the velocity respectively, NP represents the number of peers in the same communication range.

The peer average distance is computed as shown in (6). Peers in the overlay network check their mobility information and obtain their location information every duration time T. To measure the distance, we calculate the distance between all peers that are directly connected to the target node I. We denote the positions of peers i and j as D_{x_i} and D_{y_i} , respectively. Consequently, the distance between a peer j that is directly connected to a vehicle i is defined as:

$$D_{ij} = \sqrt{(D_{x_{ij}})^2 + (D_{y_{ij}})^2} \quad (5)$$

$$PAD = \sum_{j=1}^P \frac{\sqrt{(D_{x_i})^2 + (D_{y_j})^2}}{|NP|} \quad (6)$$

where $D_{x_{i,j}} = x_j - x_i$, $D_{y_{i,j}} = y_j - y_i$ and PAD represents the average distance.

Peer trajectory is computed as shown in (7), based on the predicted trajectory for the tracked nodes, we define function PTR to determine, the number of segment S_i of peer trajectory that intersects with segments traversed by the tracked node.

$$PTR = \sum_{j=1}^P |S(i, J_{tr}, t)| \quad (7)$$

where i represents the interested peer, J_{tr} represents the tracked node. The tracked node can be the bus if i and j are in the same communication range, or a super peer otherwise. Peers with trajectories most similar to their tracked node are the most suitable for super-peer selection.

$$S(i, J_{tr}, t) = \begin{cases} |T_{vi} \cap TB_{vi}| & \text{if } t \text{ connected to } J_{tr} \\ 0 & \text{else} \end{cases} \quad (8)$$

D. INTERSECTION PHASE

If a set of peers moves away from of tracked node “intersection case” in that case the cluster formation will rely on their trajectory in the reconnection phase. Many peers with similar trajectories will start the cluster formation and the suitable peer selected as a super-peer, hence based on the peer’s knowledge maps and bus schedules, peers keep exchanging data with each other until the next registration. because peers have knowledge about the bus scheduling and lines and that will help to estimate the reconnection phase time TR to the next tracked node, the SP of the free cluster has a report about its members which helps to locate data and

start the trading process with the new peers among the new tracked node. Due to the unpredictable behavior of the driver, the selection head process runs every period T_{ch}

E. CLUSTER CONNECTIVITY

Due to the high network topology change and partition, it is challenging to guarantee connectivity between vehicles, in BusCache the connectivity metric is selected by computing the delivery delay between the interested vehicle and its tracked node.

$$C(i, J_{tr}, T_{thr}) = \begin{cases} 1 & \text{if } DD < T_{thr} \\ 0 & \text{otherwise} \end{cases} \quad (9)$$

$C(i, j, T_{thr})$ indicate the connectivity of the tracked node with the peer, where i indicate the interested peer, J_{tr} is the tracked node, and T_{thr} is a given threshold. DD is the delivery delay of message from i to J_{tr} , where

$$DD = Delay_F + Delay_C + Delay_A \quad (10)$$

If the delivery delay DD is less than T_{thr} then we can consider they are connected. $Delay_A$ is the transmission time between node i and its next tracked node J_{tr} . Since the delivery, the delay is composed of two phases: carrying and forwarding, If the super peer i with the same segment $S_{i,j}$ with its next tracked node, the delivery delay in $S_{i,j}$, can expressed as:

$$DD_{ij} = \frac{L_f}{R} + \frac{L_c}{v_{ij}} \quad (11)$$

where

- $\frac{L_f}{R}$: Represents the delay due to forwarding (L_f) divided by the transmission rate (R). This term indicates the time taken for the message to be forwarded.
- $\frac{L_c}{v_{ij}}$: Represents the delay due to carrying (L_c) divided by the velocity (v_{ij}) of the vehicle. This term indicates the time taken for the message to be physically carried by the vehicle.

The term L_f in the equation is calculated as follows:

$$L_f = (1 - e)p_{ij} * LS_{ij} \quad (12)$$

where

- e represents the probability of message loss during forwarding. This factor accounts for the likelihood that a message may be lost or dropped during transmission. It’s typically a value between 0 and 1, where 0 means no loss and 1 means complete loss.
- p_{ij} represents the length of the segment $S_{i,j}$. It indicates the physical distance between the current position of the sender (super peer i) and the destination (next tracked node J_{tr}).
- LS_{ij} represents the load on the segment $S_{i,j}$. It indicates the current traffic or load on the segment, which could affect the speed of message transmission.

If the super peer i is not in the same segment $S_{i,j}$ with its next tracked node, the total delivery delay can be expressed

as:

$$TD = \sum_b^a DD_{ij} \quad (13)$$

This equation calculates the total delivery delay (TD) when the super peer i is not in the same segment $S_{i,j}$ with its next tracked node j_{tr} . It sums up the individual delivery delays (DD_{ij}) for each segment $S_{i,j}$ along the route from i to j_{tr} .

Algorithm 1 Delivery Timing Control

```

1: while  $DD < Threshold$  do
2:   if  $Peer_i \in C_i$  then
3:     Send_SR( $TRNi$ )
4:   else
5:     if There is no  $V_i$  then
6:       if  $CurrentTime - GenerationTime > Msg\_LifeTime$  then
7:         Drop Msg
8:       else
9:         Each 10 s
10:        if There is  $V_i$  then
11:          Forward_MSG
12:        else
13:          Carry_MSG
14:        end if
15:      end if
16:    end if
17:  end if
18: end while

```

V. PEER-TO-PEER OVERLAY

The P2P data delivery policy is the main operation that determines the performance of the data exchange in IoV, in order to increase the efficiency of P2P services over the Vehicular Ad hoc Network. In this section, we propose an efficient P2P path selection based on the instance network information collected from other peers.

A. PEERS SELECTION

After the content consumer vehicle receives the peer list from its tracked node, it sends a request to the P2P overlay of its peer list to start trading pieces with them, the P2P overlay is presented with a set of peers, which share the same content. There may be multiple paths to each of these peers, resulting in a set of paths with different costs that the content consumer vehicles have to choose from. The optimal path data delivery indicates how many pieces to download from each peer, and which path should be used to reduce the download time, It is important to make optimal decisions about which paths to use for their exchange data, especially in the challenge of a fast-moving IoV environment. We should take into consideration that if a peer receives, several download requests at the same time, that peer can be chock or unchock corresponding peers based on their

capacities, where necessity determines a strategy to decide which request to handle first.

1) PATH SELECTION

The content consumer vehicle starts by sending a download request message concerning their interest to its overlay peer list, every peer who knows the requested pieces will reply by sending a confirmation message. The content consumer vehicle stores all paths acquired by the searching process according to their arrival times. A content consumer can travel through paths based on their delivery time, But since IoV nodes are always on the move and experience high topology changes, this approach may not be feasible. So we define triplet to determine the optimal paths for each link connectivity, delay the time needed for delivery pieces by path and bandwidth is the upload/download rate of each peer.

Algorithm 2 Select Optimal Path to Dedicated Peers

```

1: A peer  $i$  receive status information about is a peer in the network
2: for each received RRDSVMSG do do
3:    $Peer\_List[i] \leftarrow \langle ID, PeerSatus \rangle$ 
4:    $Path[i] \leftarrow \langle P \rangle$ 
5: end for
6: for each Peer in  $Peer\_List$  do do
7:   if  $(ECT_i > CT)$  then
8:      $Optimal\_Peers[i] \leftarrow \langle ID \rangle$ 
9:      $Path[i] \leftarrow \langle ID \rangle$ 
10:  else
11:     $Temporary\_Peers[i] \leftarrow \langle ID \rangle$ 
12:  end if
13: end for
14: Sort peers based on their capacities
15:  $SortMAX(Optimal\_peers)$ 
16: for each Peer  $j$  in  $Optimal\_peers$  do do
17:   With Given path and peers capacities
18:   for each Peer  $j$  in  $Optimal\_Path$  do do
19:     Download Pieces  $X_i$  from Path  $j$ 
20:   end for
21: end for

```

2) CONTENT PARTITION

We assume that content files are partitioned into pieces and each vehicle of the P2P overlay has parts of the file or the whole file or recently connected vehicle to the P2P overlay does not have any pieces to share, and we have N request download associated to one file. When a peer requests more than one file, then it should initiate more than one request. Each file associated with Request i is split into P_i pieces, to download the whole file, the peer needs to get all P_i pieces from different peers, peer sets T_i a maximum time needed to download a file within. Each peer associated with a path j has P_j pieces of the file if $P_j = P_i$ that indicates it has the whole file to share, if $0 < P_j < P_i$, that means the peer has just a part of it. Each peer has an upload and download

rate. Our problem concentrates in determining which N paths should be used to start the download, and how many pieces are needed to download it from each path to minimize the download time subject to bandwidth constraints, with known bandwidth rate and times, we formulate a linear mixed integer problem, we define variable $x_{i,j}$, where reference to the number of pieces should download it using path j , We define the objective function and the constraint in (14), where $X_{i,j}$ is the number of pieces multiplied by the download time per piece for each path selected $d_{i,j}$

$$\sum_{i=0}^{m-1} \sum_{j=0}^{n-1} x_{i,j} d_{i,j} \quad (14)$$

This equation aims to minimize the total download time by determining the optimal allocation of pieces across paths for all files, considering the download time per piece for each path.

This optimization aimed at minimizing download time while satisfying bandwidth constraints in a P2P overlay network, such that BW_i is not exceeded as shown in (15-18).

$$\sum_{j=0}^{n-1} x_{i,j} c_{ij} \leq BW_i \quad \forall i \quad (15)$$

$$\sum_{j=0}^{n-1} x_{i,j} = P_i \quad \forall i \quad (16)$$

$$0 \leq x_{ij} \leq p_j \quad \forall i, j \quad (17)$$

$$x_{i,j} \in Z \quad (18)$$

The constraint 16 ensures that each peer downloads the total number of pieces needed for each file i . It sums over all paths j for each file i , setting it equal to the total number of pieces P_i required for the file. Constraints (17-18) specify that the number of pieces to download ($x_{i,j}$) should be non-negative and less than or equal to the total number of pieces P_j . Additionally, the variable $x_{i,j}$ is an integer (denoted by Z).

B. DISCOVERY PHASE

The content consumer vehicle starts the initiation phase by sending a request message (Dsvmsg) to all vehicles within its peer list (PL). The structure of Dsvmsg message is shown in Table 1, whenever the content consumer vehicle sends a new Dsvmsg, the message identifier DsvID is incremented by one. The peer ID and the DsvID identify the discovery message. Identifiers determine which peer sends the messages and indicate if the message is new or already a replay has been sent according to the message. PL and WPL indicate the list of pieces that the peer is ready to share and the pieces that the peer requested.

TABLE 1. Discovery message structure.

DsvID	peerID	PL	WPL	Timer	loc	speed
-------	--------	----	-----	-------	-----	-------

Algorithm 3 Executed by Peer i ‘‘Upon Receive Msg’’

- 1: Create a New DSVMSG
- 2: PeerList[I] \leftarrow <List of neighbor peers>
- 3: DSVMSG[ID] \leftarrow Vid
- 4: DSVMSG[Speed] \leftarrow Vsi
- 5: DSVMSG[PL] \leftarrow pieces list downloaded
- 6: DSVMSG[WPL] \leftarrow pieces list Missed
- 7: Peeri Send DSVMSG to its PeerList[I]
- 8: Interested Vi ‘‘peers’’ Replay to DSVMSG and forward its status information
- 9: **for** each Segement Sij **do**
- 10: SendDSVMSG(Vi, S, D, PL, H, BW)
- 11: **end for**
- 12: Upon receiving a request message DSVMSG from peeri within time T
- 13: **if** node Vi receives MSGDSV from node Vj Within Time T **then**
- 14: Send status information accord to the peer J request, including id, speed, direction, PieceList info, Bandwith rates, Hops count
- 15: ForwardRDSVMSG<J,D, S,LB, WLP, BW, H>
- 16: **end if**

A Timer T is initiated by the content consumer, which indicates the stability of the peers, during this time, if the requested peers do not replay, the peer is considered disconnected. Every peer, upon receiving the DSVmsg, verified if it has the requested pieces corresponding to the discovery message, and estimated the communication time ECT based on their speeds and positions with the interested vehicles. Concerned peers replay by sending a unicast response discovery message to the content consumer, with information shown in Figure 4. Otherwise, the peers do not replay to the discovery messages meaning that peers are not anymore in the P2P overlay, so they are deleted from the content consumer’s peer list. DWR and UPR indicate the download and upload rate of each peer.

C. ESTIMATION COMMUNICATION TIME

In peer-peer VANets systems, communication time is considered an important issue, where the objective is to transfer whole files between peers, we aim to share all file pieces among the P2P overlay, here the estimated communication time and the transmission duration intervals contribute to selecting adequate peers via optimal paths to reduce the download time. Therefore, the Time field set in the Discovery message serves to estimate the communication duration between the interested vehicle and its peers. If the TD exceeds the ECT , the optimal paths of the peer list are updated according to the vanet conditions. To estimate the communication time along a path, we take a sequential approach. Where we need to find the link time expiration between the interested vehicle and the concerned peer via the intermediate nodes, and we compare it to the duration

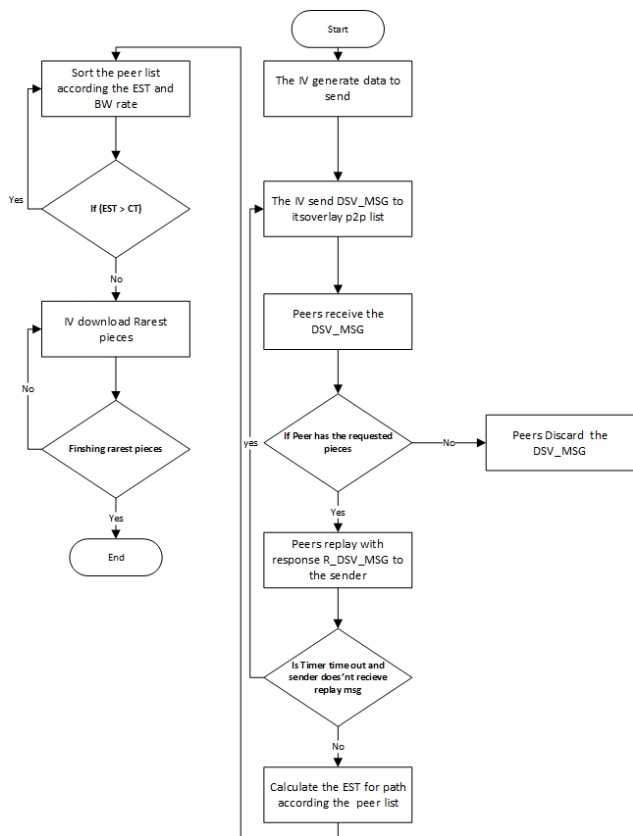


FIGURE 4. Peer discovery flowchart.

transmission pieces DTs , if the link communication time among the path via intermediate nodes exceeds the TD , the path will use to start pieces exchange.

Otherwise, the TD will reconfigured to accord the duration of the communication link. Figure 5 explains the selecting peer’s process according to the DT and ECT . After the interested vehicle or peer selects the optimal paths, based on the given bandwidth rates and delays, we calculate the ECT for each path. In our example IV represented with the red circle, the IV have different paths to different peers belonging to its peer list, based on the network information, peers selected via the path, arcs from the source node IV to its peers define path metrics, every path associate with $\langle hcount/delay, \langle upload, download rates \rangle$, among each path several pieces defined to start download from selected peer, peers with the high upload and download rates considered as the backbone of the exchange process, because they effectively reduce download time.

Every path is associated with several pieces to download, in our example P1, P3 the red peers appear in Figure 5.b, have the highest bandwidth rate, Figure 5.c show the distribution of pieces via different paths, TD and ECT calculated for each path to determine the link lifetime during the transmission process.

To estimate the communication lifetime between peers, we calculate for each intermediate node i in path j the

“ ECT_i ”, where every node calculates ECT_i with its next node in the path. ECT is the sum of all ECT_i among the path j .

$$ECT = \sum_1^k ECT_i \tag{19}$$

To calculate the ECT , we suppose that motion parameters for nodes are known. Let N_i, N_j represent two vehicles “peers” with transmission range R , and each peer has its proper coordinate $(x_i, y_i), (x_j, y_j)$ with velocity V_i, V_j and with defined directions DR_i, DR_j . After time T peers N_i, N_j their coordinate updated to $(x'1, y'1)$ for $N1$ and $(x2', y2')$ for $N2$. $D1, D2$ indicate the traversed distance after time T . For link stability the distance d must less than range of communication R , as shown in Figure 6.

$$d = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2} \tag{20}$$

the distance D between N_i and N_j is given by:

$$D_x = [x_j - x_i] + [v_j \cos \theta_j - v_i \cos \theta_i] t \tag{21}$$

$$D_y = [y_j - y_i] + [v_j \sin \theta_j - v_i \sin \theta_i] t \tag{22}$$

N_i, N_j in the same communication if the distance difference is less than R

$$d = \sqrt{D_x^2 + D_y^2} < r \tag{23}$$

$$t < = \frac{-\langle ab + cd \rangle \pm \sqrt{\langle a^2 + c^2 \rangle r^2 - \langle ad - bc \rangle^2}}{\langle a^2 + c^2 \rangle} \tag{24}$$

where

$$v_j \cos \theta_j - v_i \cos \theta_j = a, v_j \sin \theta_j - v_i \sin \theta_j = c \tag{25}$$

$$x_j - x_i = b, y_j - y_i = d \tag{26}$$

D. UNCHOKING/CHOKING PEER ALGORITHM

The P2P has two main algorithms P2P system is based on, the unchoking/choking algorithm and the pieces’ selection algorithm, the unchoking algorithm is about which peers select to upload to them, choking algorithm is temporary refuse upload to certain peers, and for pieces selection is reference to which pieces to request for download.

Firstly, we will concentrate on the unchoking/choking algorithms after we will discuss the pieces selection algorithm. The big challenge here it’s how to select which peers to choke and which peers to unchoke under VANets condition.

In our proposal, selecting peers to unchoke/choke is based on their percent download rate. Figure 7 shows different nodes (peers, seeds), where each node is associated with a value indicating its average download rate. Peers with high download averages (HDA) typically have high bandwidth rates and exchange data with each other and also with seeds. We classify peers according to their HDA, with seeds being in the first level and closely associated with peers in subsequent levels. This approach, known as ‘tree-based unchoking,’

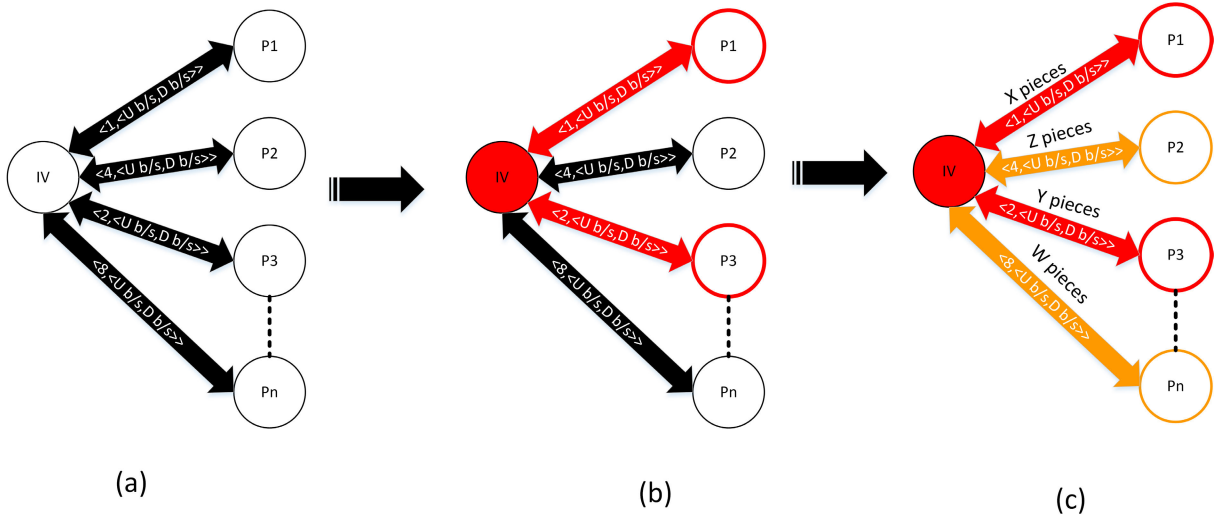


FIGURE 5. Pieces distribution among selected peers.

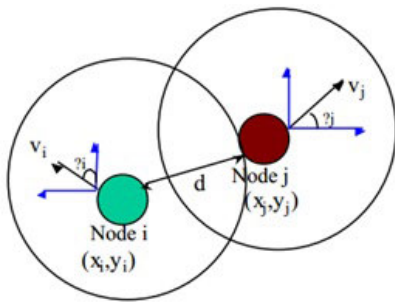


FIGURE 6. Transmission between two nodes.

allows for rapid duplication of seeds throughout the network while simultaneously maintaining load balancing among the peers.

As shown in Figure 7, the first level contains a seed ‘S’ that has the entire file, while the second level contains peers that have a part of the file with high average download rates (HAD). Over time (after a period of time T), the download average for each peer increases due to the trading process, resulting in more seeds, as depicted in Figure 7b. In this strategy, seeds unchoke peers with high HDA. For example, in Figure 7a, seed S1 unchokes peers P1, P2, P3, and P4, while peer P2 unchokes peers P4 and P5. This strategy classifies the peers into levels or groups, and each peer must demonstrate its qualifications to move to the highest level by efficiently contributing to the exchange process.

Choke/unchoke algorithms and piece selection algorithms are atomic strategies, where the main goal of piece selection is to distribute content equitably between participating peers to guarantee relationships between peers (i.e., peers interested in each other). Before delving further into our proposed algorithm, we highlight some points. In the BitTorrent

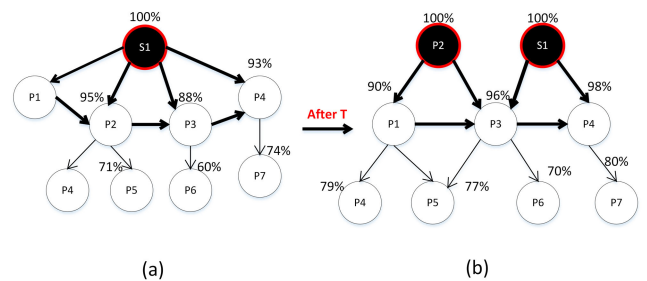


FIGURE 7. HDA tree-based peer unchok.

protocol, seeds and peers unchoke peers with high upload rates, ignoring the physical positions of peers. This approach does not consider the peer’s position in the network, whether it is near or far. This behavior introduces extra traffic to the network and increases download time. Due to the high mobility of nodes in VANets, nodes farther away from the source node are less reliable than nodes closer to the source node. Therefore, we modified the choking/unchoke algorithm to overcome these issues and adapt to VANet networks.

We classify nodes into different categories (seed, peer, new peer), with each node defining a value throughput DA that permits it to select which peers to unchoke and which peers to choke. As we know, seeds are the most requested nodes in the overlay. Seeds define their throughput DA according to the download average of each requested node and sort them based on their upload rate. Applying this strategy aims to balance the load in the network, where peers with low throughput can exchange with each other, and the same for peers with high throughput. Additionally, due to network churn where seeds leave the network, this strategy permits duplicating seeds across the network, thereby reducing download time and increasing download speed.

Algorithm 4 OnSend

A peer i sends a download request to its peers in the network based on their download values
 $Queries_{List}[i] \leftarrow \langle Peers \rangle$
 Firstly All peers are choiced and classified into tree structure
 $List_C P \leftarrow Queries_{List}$
 $List_{UNP} \leftarrow Empty$
 Pointers to determine which peers to unchoke
 $P, Q : \leftarrow null$
 $p \leftarrow Root(T)$
while ($P! = NULL$) **do**
 $Q \leftarrow p$
 if $Value(Peer_i) \geq Value(P)$ **then**
 $P \leftarrow right[P]$
 else
 $P \leftarrow Left[P]$
 end if
end while
for $i = 1$ to $Q.arrayLength$ **do**
 $List_{UNP} \leftarrow values[Q]$
end for
while ($L \geq j$) **do**
 unchoke 3 peers with the highest capacities and a random peer from the rest.
 for each T **do**
 Unchoke Peers with the highest capacities
 UnchokePeer($List_{UNP}[j]$)
 end for
end while
 Optimistic random Chock
for each T=10s **do**
 Unchoke Peers with Lowest capacities
 UnchokePeer($List_{UNP}[j]$)
end for

Suppose that we have an interested peer with N requests and their value/score (DA) received from different peers. Each request corresponds to a peer set $S = \{Q_1, \dots, Q_N\}$. Let K be the download average of all elements in S . We will group the peers based on their DA values to determine which peers to choke and unchoke. Each peer sorts its requested peer's list according to their DA from the lowest to the highest DA. We define a function F that will split the list of peers into two parts at each level until size L , where $L \geq 4$. If $L < 3$, the fastest peers and 1 random peer are unchoked. As shown in Figure 8, a list of queries received by certain peers LR_i , where each peer is defined by its download score, peers are sorted based on that score. Suppose that P_6 and P_{17} receive different queries with respective values (66, 39). We aim to classify the peers into similar groups to balance the network and avoid problems such as the rarest pieces.

In our example, we have 2 peers, P_6 and P_{17} , receiving the same download requests. It is necessary to determine which peers must choke and which to unchoke. Here, LR_i is split

into 2 each time until $L \geq 4$, where $L = Length(LR_i)$. $P_6(60)$ compares its download value. If within $LR_i < 41, 44 >$, then it compares with $< 71, 78 >$ after with $< 61, 65 >$. P_6 unchokes peers with $< 65, 65, 71 >$ and randomly chokes one of the remaining two $< 65, 65 >$. The same strategy is applied to $P_{17}(39)$ where it chooses $< 41, 44 >$, $< 18, 20 >$, $< 32, 32 >$. In case reference values are equal, the peer chooses the rightmost. Here, $P_{17}(39)$ unchokes $< 41, 33, 33 >$ and randomly unchokes $< 32, 32 >$. This process is further explained in Algorithm 5.

Algorithm 5 Algorithm for Tree-Based Choke/unChoke

-
- 1: **OnSend:**
 - 2: **Step 1:** All peers in the list of interested vehicles are initially choked and classified into interested and not interested peers.
 - 3: **Step 2:** The upload capacity of interested peers is shared with 4 peers: three peers are favored to start downloading, and optimistic choke is applied to the last peer.
 - 4: **Repeat:**
 - 5: **Step 3:** Every T seconds (Regular unchoke interval).
 - 6: **Step 3.1:** If the interested vehicle (IV) is in the downloading state, all peers in its interested list are divided into two groups based on their maximum or minimum DA values.
 - 7: Peers in each group are ordered based on their upload rate relative to the upload rate of the IV.
 - 8: The three fastest peers in the maximum group are unchoked.
 - 9: **Step 3.2:** If the IV is in the seed state, peers with high HAD (going to finish downloading the entire file) and high download rate are favored to be unchoked.
 - 10: **Step 4:** Every $3T$ seconds (Optimistic unchoke interval), one interested peer is randomly unchoked.
 - 11: The optimistic unchoke aims to provide the first pieces to new peers for trading with others and to evaluate the upload capacity.
-

E. PIECES SELECTION ALGORITHM

PSA is considered the backbone of the trading process. It is necessary to define an intelligent mechanism for piece selection to avoid situations where pieces are missed in the P2P overlay network. To overcome this issue, replicating pieces over different peers at the earliest opportunity will improve download speed and increase the probability of successful downloads. In PSA, we will define two main policies that contribute to determining the download time: Rare pieces and Random Pieces.

1) RANDOM PIECES SELECTION

At the beginning of the downloading process, when newly joined peers are eager to start exchanging content but lack pieces to trade with other peers, difficulties, and delays in downloading may arise. In such cases, obtaining the initial

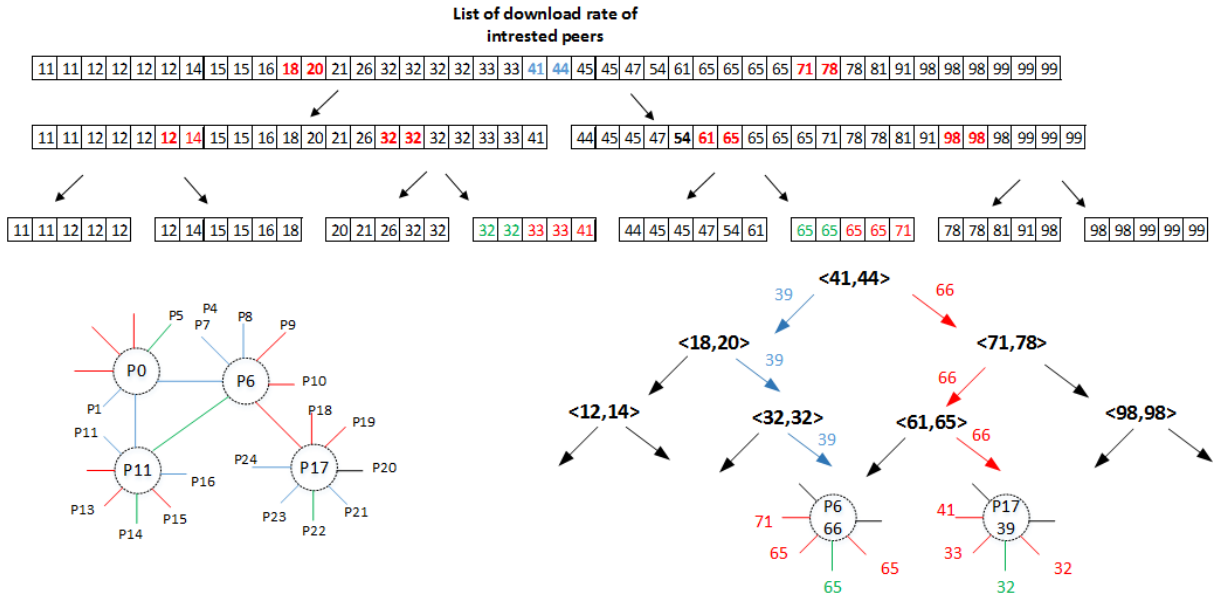


FIGURE 8. Peers check/uncheck download.

pieces as quickly as possible is essential. Selecting pieces randomly in the initial stage can be considered an efficient solution to increase download speed and reduce download time. Once a peer acquires the first pieces to trade, it can then prioritize which pieces to download first based on its needs and preferences.

2) RAREST PIECES SELECTION

Rarest pieces can be a serious problem when a seed leaves the network, so this strategy aimed to make peers interested in each other or in other words to guarantee that all peers can start an exchange with each other without exception, Every interested peer knows about pieces that exist in its peer list, this is the information used to select rarest pieces, this strategy show very high impact for sharing the content. In the beginning of download, the seed will be a bottleneck since is the only one that has all the pieces, IV can fetch the rarest pieces from the source peer “seed”. In this strategy, most of the peers looking for rare pieces, by getting those pieces for the new peers will increase the chances of exchanging data with other peers.

The piece selection strategy algorithm of our protocol employs a hybrid mechanism where both random and rarest pieces are downloaded simultaneously from various other peers. Initially, priority is given to downloading from peers with high upload rates, particularly for acquiring the rarest pieces, while random pieces are obtained from normal peers. Our algorithm consists of two main strategies: Rare piece selection and Random piece selection. Further details are provided in Algorithm 6.

Our algorithm is based on two main strategies: peer selection and piece selection. Initially, peers in the overlay

Algorithm 6 HPS Hybrid Pieces Selection Algorithm

```

1: while !FileDownload do
2:   if (Target_peer.status == 'New') then
3:     for each i=1 To N do
4:       Sort peers based on their hpos distance
5:       HP <> ← SortPeers(HAD, Hops)
6:       LP <> ← SortPeers(LAD, Hops)
7:     end for
8:     for each i=1 To 3 do
9:       Exchange LRarest[Xi] with HP[i]
10:    end for
11:    Exchange LRandom[Xi] with LP[i]
12:  end if
13:  for each i=1 To 3 do
14:    Exchange LRandom[Xi] with HP[i]
15:  end for
16:  Exchange LRarest[Xi] with LP[i]
17: end while

```

are classified into groups based on their DA (Download Average), distinguishing between those with high and low upload/download rates. The interested peer selects its target peers with DA values similar to its own to initiate the exchange operation. Additionally, the interested peer prioritizes selecting peers that are closer in terms of distance (measured in hops count) and based on their upload rate. This means that peers in close proximity are favored over distant peers due to the stability of the link.

Figure 9. show the selecting process among the participating peers, where Peer A with two categories, < B, E, F, G > peers with high upload rate and < C, D, H, I > peers with

Algorithm 7 Vehicular Overlay Network Initialization

```

1:  $S \leftarrow List(\text{of all Buses on the segment vicinity})$ 
2:  $TRB[] \leftarrow BusTrajectory[0..n]$ 
3:  $TRVi[] \leftarrow vehicleTrajectory[0..n]$ 
4: BUS:
   {Buses Send beacon messages during its service}
   {interested  $V_i$  "peers" register among adequate Bus}
5: while !Destination do
6:   for Each bus on  $S$  do
7:     Bus broadcast registration beacon message
8:   end for
   Peers:
   {Peers register among close buses with trajectory similarities}
9:   for  $i = 0$  to  $S.length$  do
10:     $TargetBus \leftarrow Max(TRB \text{intresection with } TRVi)$ 
11:     $TmpBus \leftarrow Min(TRB \text{intresection with } TRVi)$ 
12:   end for
13: end while

```

low upload rate, Peer chose 3 peers with high upload rate and a peer with low upload rate "optimistic unchock", in our example A chose peers B, E, F, C, Peer A chose the target peers B, E, F according to their position and upload rate, and chose peer C for optimistic unchock to avoid isolate new empty peers. In first place peer A starts by downloading the rarest peers among its peer list via paths with HUR and at the same time downloads random pieces from paths with low upload rate. In case when the joined peer to the P2P overlay does not have a piece to trade, conversely the path with HUR will associate to download random pieces and low paths for rarest pieces for some time T.



- 1- Green peer is the requested peer.
- 2- Red peers indicate peers with high upload rate.
- 3- Orange peers indicate peers with low upload rate.

FIGURE 9. Peers selection.

VI. EVALUATION AND RESULT

In this section, we measure the performance of our proposed cooperative download scheme on P2P networks and the

accuracy of the theoretical analysis through simulation conducted by SUMO and OMNET++ tools to evaluate our proposed protocol performance in a real environment. We study the amount of data that a vehicle can download in a drive-through, with different scenarios (without using tracker nodes "flooding" CarTorrent, with tracker node, with tracked node and clustering strategy). In addition, comparisons between our protocol and existing cooperative downloading protocols such as "Car-torrent". In addition, the download time occupied by vehicles to download all the data is subject to the bandwidth constraint. Finally, we evaluate the throughput of our protocol when multiple files are requested. Moreover, Comparisons between our protocol and a simple sharing protocol are presented. We simulate our protocol in an urban scenario where we use a vehicular network consisting of 50 to 100 ordinary vehicles randomly distributed along a predefined segment and a set of buses travels along predefined trajectories at a constant speed and scheduled times. Ordinary vehicles and buses are configured with 802.11n standard MAC protocol that has a transmission range of approximately 250 to 300 meters and a transmission rate of 27M bits/sec. The major parameters for simulation are shown in Table 2.

TABLE 2. Simulation parameters.

Symbol	Description	Value
L	Length of the segment	1 km
S_b	Bus speed	35-40 km/h
R	V2V communication range	250-300 m
S_v	Vehicle speed	20-60 km/h
F_s	File volume planned for download	2000-5000 Kb
N_v	Number of ordinary vehicles	50
N_b	Number of buses	6

In our experiment, we aimed to study the average download time in VANETs without relying on infrastructure. The experiment involved buses traveling along predefined segments in urban zones. The objective is to provide download information to interested vehicles, enabling them to look up data over the overlay network without flooding the network. This approach aimed to reduce lookup delay, minimize bandwidth consumption, and decrease download time. The proposed Bus Torrent mainly consists of four phases.

- **Registration phase:** Each interested vehicle registers among buses encountered in its trajectory.
- **Clustering phase:** Each interested vehicle estimates its similarity with others by computing its weight to form groups. Each group includes a super-peer responsible for managing the data flow to/from the next super-peer.
- **Download phase:** Interested vehicles or peers establish connections with each other to determine the optimal paths and formulate a Selection Function to decide which peers to choke and which to unchoke based on their capacities.

- **Pieces Hybrid Selection phase:** Rarest and random pieces are selected to download data according to the peer status.

Simulation results demonstrate that the proposed algorithm is likely to maximize the volume of downloaded data and achieve minimum download time. The performance metrics used for evaluating our protocol, Bus Torrent, are described below:

- 1) **Bandwidth Consumption/Throughput:** This metric refers to the amount of data packets exchanged between different peers, usually measured in bits per second (bit/s) or sometimes in data packets per second (p/s).
- 2) **Packet Delivery Ratio (PDR):** This is the average number of packets received at each receiver node.

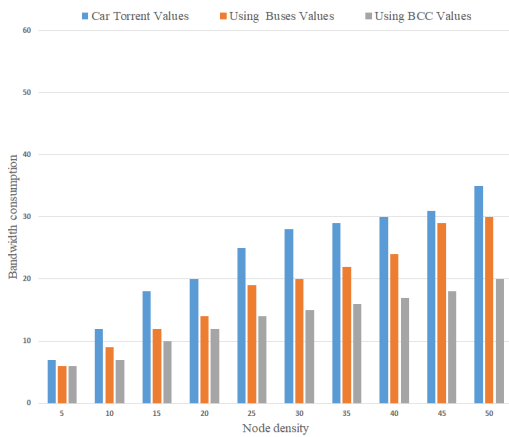


FIGURE 10. Bandwidth consumption comparison.

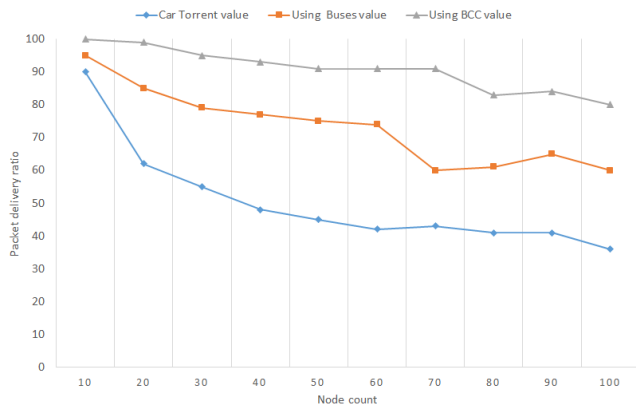


FIGURE 11. Packet delivery ratio comparison.

Figure 10 shows the bandwidth consumption per vehicle for three cases (Flooding network “Car torrent,” Less tracker, fully tracked). We notice that the average bandwidth consumption per vehicle increases as the number of vehicles increases. This is because the number of neighbors of each vehicle increases as the total number of vehicles increases. Hence, the vehicle will send and receive more control packets, such as lookup queries. Additionally, the difference between

the bandwidth consumption of a vehicle in the three cases increases as the number of vehicles increases due to the same fact mentioned before: the round trip of a query in ABSRP is much longer than that in ROADS.

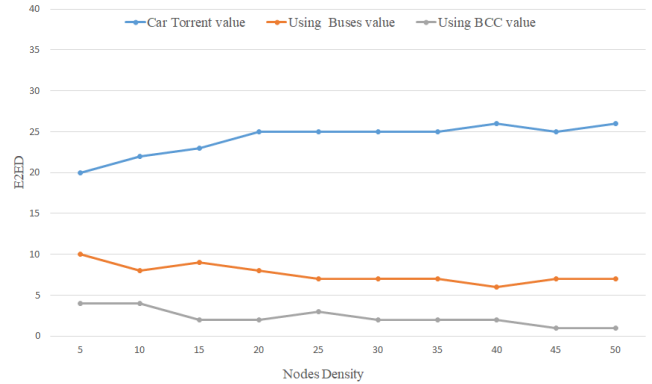


FIGURE 12. End-to-end delay comparison.

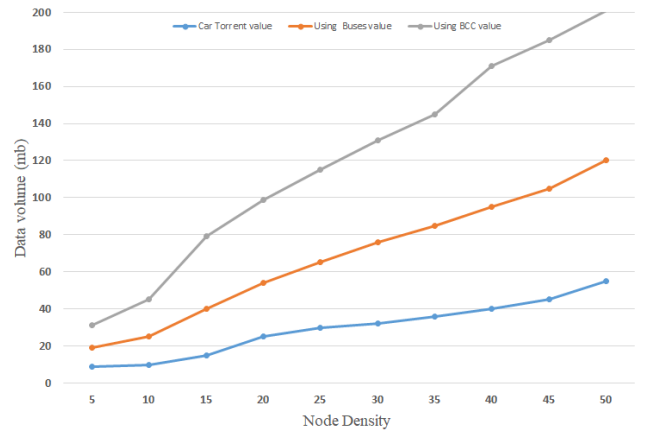


FIGURE 13. Download capacity comparison.

Figure 11 shows the Packet delivery ratio(PDR)of all three approaches concerning the number of vehicles. It is observed that PDR decreases with increasing node density. This is because when the node density is high, the interference among nodes increases, leading to packet collisions and loss. This interference can be caused by overlapping transmissions or limited available bandwidth. As a result, As the number of vehicles increases the PDR decreases due to higher packet loss rates. Upon analysis, it is evident that the Bus Cluster Approach (BCC) exhibits superior performance in maintaining a higher PDR, particularly during the initial phase, compared to both CarTorrent and Bus-based approaches. This advantage can be primarily attributed to the data requesting and retrieving process employed by BCC, which efficiently manages network resources and minimizes delays. In contrast, CarTorrent and Bus-based approaches resort to flooding the network randomly, leading to excessive consumption of network resources and prolonged query resolution times without ensuring satisfactory outcomes.

Thus, the BCC approach emerges as a more effective solution for sustaining a better Packet Delivery Ratio overall.

Figure 12 depicts the end-to-end delay for the three approaches as the number of vehicles increases. Notably, CarTorrent demonstrates an increase in end-to-end delay over time. This increase in end-to-end delay with CarTorrent can be attributed to its flooding data transmission strategy, which likely leads to high network traffic. Conversely, both the Bus-based and Bus cluster-based (BCC) approaches exhibit a slight reduction in end-to-end delay with the growing number of vehicles, indicating improved efficiency in data transmission. This reduction in end-to-end delay with Bus-based and BCC approaches can be attributed to their distinct data retrieval strategies. In the Bus-based approach, vehicles request the bus to locate data, which can lead to a bottleneck as the bus handles multiple data requests. However, with the BCC approach, vehicles proactively approach their corresponding super peers within the cluster, reducing reliance on a centralized entity and distributing data retrieval tasks more efficiently among vehicles. This decentralized approach likely contributes to the observed reduction in end-to-end delay as the number of vehicles increases.

Figure 13, which displays the data volume downloaded for the three approaches, the significant increase in data volume observed with the Bus Cluster Approach (BCC) this can be explained by its inherent scalability and robustness of locating data within another peer on clusters. The Bus Cluster Approach likely employs clustering techniques mechanisms, enabling it to handle larger volumes of data with greater efficiency compared to CarTorrent and the other approaches. Additionally, the relatively lower data volumes for CarTorrent and the Bus-based may be attributed to their less efficient data management strategies, such as flooding the network.

VII. CONCLUSION

In this paper, we proposed BusCache, a traffic-aware content delivery system for IoV. BusCache uses buses as trackers for the content distribution on the overlay network, due to their availability and unique characteristics such as the predefined trajectories and time constrain. Simulation results show that BusCache can reduce content lookup time and content dissemination compared to state-of-the-art approaches. Our proposed method proved to be effective in controlling the data flow between vehicles without rely on external infrastructure. The proposed peer selection mechanism selects the most appropriate peers that will deliver the desired content by taking into account their velocity, destination and other traffic-related factors.

REFERENCES

- [1] B. Ji, X. Zhang, S. Mumtaz, C. Han, C. Li, H. Wen, and D. Wang, "Survey on the Internet of Vehicles: Network architectures and applications," *IEEE Commun. Standards Mag.*, vol. 4, no. 1, pp. 34–41, Mar. 2020.
- [2] W. Zhang, N. Aung, S. Dhelim, and Y. Ai, "DIFTOS: A distributed infrastructure-free traffic optimization system based on vehicular ad hoc networks for urban environments," *Sensors*, vol. 18, no. 8, p. 2567, Aug. 2018. [Online]. Available: <http://www.mdpi.com/1424-8220/18/8/2567>

- [3] I. Ud Din, B. Ahmad, A. Almogren, H. Almajed, I. Mohiuddin, and J. J. P. C. Rodrigues, "Left-right-front caching strategy for vehicular networks in ICN-based Internet of Things," *IEEE Access*, vol. 9, pp. 595–605, 2021.
- [4] D. Lin, J. Kang, A. Squicciarini, Y. Wu, S. Gurung, and O. Tonguz, "MoZo: A moving zone based routing protocol using pure V2V communication in VANETs," *IEEE Trans. Mobile Comput.*, vol. 16, no. 5, pp. 1357–1370, May 2017.
- [5] S. A. Elsayed, S. Abdelhamid, and H. S. Hassanein, "Prediction-based cooperative cache discovery in VANETs for social networking," *Comput. Commun.*, vol. 214, pp. 184–200, Jan. 2024.
- [6] N. Aung, S. Dhelim, L. Chen, A. Lakas, W. Zhang, H. Ning, S. Chaib, and M. T. Kechadi, "VeSoNet: Traffic-aware content caching for vehicular social networks using deep reinforcement learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 8, pp. 8638–8649, Aug. 2023.
- [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 Trans. Veh. Technol.*, vol. 69, no. 4, pp. 4246–4256, Apr. 2020.
- [8] T. Thirugnanam and M. R. Ghalib, "A reward based connectivity-aware IoV neighbor selection for improving reliability in healthcare information exchange," *Peer-to-Peer Netw. Appl.*, vol. 13, no. 6, pp. 2112–2122, Nov. 2020.
- [9] T. Zaarour and E. Curry, "SemanticPeer: A distributional semantic peer-to-peer lookup protocol for large content spaces at Internet-scale," *Future Gener. Comput. Syst.*, vol. 132, pp. 239–253, Jul. 2022.
- [10] A. Hammoud, H. Sami, A. Mourad, H. Otrok, R. Mizouni, and J. Bentahar, "AI, blockchain, and vehicular edge computing for smart and secure IoV: Challenges and directions," *IEEE Internet Things Mag.*, vol. 3, no. 2, pp. 68–73, Jun. 2020.
- [11] A. I. Ameur, A. Lakas, M. B. Yagoubi, and O. S. Oubbati, "Peer-to-peer overlay techniques for vehicular ad hoc networks: Survey and challenges," *Veh. Commun.*, vol. 34, Apr. 2022, Art. no. 100455.
- [12] E. Khatibi and M. Sharifi, "Resource discovery mechanisms in pure unstructured peer-to-peer systems: A comprehensive survey," *Peer-to-Peer Netw. Appl.*, vol. 14, no. 2, pp. 729–746, Mar. 2021.
- [13] N. Aung, W. Zhang, S. Dhelim, and Y. Ai, "T-Coin: Dynamic traffic congestion pricing system for the Internet of Vehicles in smart cities," *Information*, vol. 11, no. 3, p. 149, Mar. 2020. [Online]. Available: <https://www.mdpi.com/2078-2489/11/3/149>
- [14] N. Aung, W. Zhang, K. Sultan, S. Dhelim, and Y. Ai, "Dynamic traffic congestion pricing and electric vehicle charging management system for the Internet of Vehicles in smart cities," *Digit. Commun. Netw.*, vol. 7, no. 4, pp. 492–504, Nov. 2021. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S2352864821000080>
- [15] Z. Wang, Z. Zhong, M. Ni, M. Hu, and C.-Y. Chang, "Bus-based content offloading for vehicular networks," *J. Commun. Netw.*, vol. 19, no. 3, pp. 250–258, 2017.
- [16] L. Zhang, S. Jia, Z. Liu, Y. Wang, and Y. Liu, "Bus-Ads: Bus trajectory-based advertisement distribution in VANETs using coalition formation games," *IEEE Syst. J.*, vol. 11, no. 3, pp. 1259–1268, Sep. 2017.
- [17] Z. Xue, Y. Liu, G. Han, F. Ayaz, Z. Sheng, and Y. Wang, "Two-layer distributed content caching for infotainment applications in VANETs," *IEEE Internet Things J.*, vol. 9, no. 3, pp. 1696–1711, Feb. 2022.
- [18] N. Chaib, O. S. Oubbati, M. L. Bensaad, A. Lakas, P. Lorenz, and A. Jamalipour, "BRT: Bus-based routing technique in urban vehicular networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 11, pp. 4550–4562, Nov. 2020.



ABDENACER NAOURI received the B.S. degree in computer science from the University of Djelfa, Algeria, in 2011, and the M.Sc. degree in networking and distributed systems from the University of Laghouat, Laghouat, Algeria, in 2016. He is currently pursuing the Ph.D. degree with the University of Science and Technology Beijing, Beijing, China. His current research interests include cloud computing, smart communication, machine learning, the Internet of Vehicles, and the Internet of Things.



NABIL ABDELKADER NOURI received the B.S. degree in computer sciences from the University of Laghouat, Algeria, in 2003, and the magister degree in networking and distributed systems from the University of Bejaia, Algeria, in 2007. He is currently an Associate Professor with the University of Djelfa, Algeria. His current research interests include wireless networking design, the Internet of Things, performance evaluation, fog computing, and optimization.



SALIM NAOURI received the B.S. degree in computer science from the University of Djelfa, Algeria, in 2014, and the M.Sc. degree in embedded system engineering from the University of Oran 1 Ahmed Ben Bella, Oran, Algeria, in 2017. His current research interests include cloud computing, smart communication, machine learning, the Internet of Vehicles, and the Internet of Things.



AMAR KHELLOUFI received the B.S. degree (Hons.) in computer science from the Faculty of Sciences and Technology, Ziane Achour University of Djelfa, Djelfa, Algeria, in 2012, and the M.S. degree in distributed information systems from the Faculty of Sciences, University of Boumerdès, Boumerdès, Algeria, in 2014. He is currently pursuing the Ph.D. degree with the School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing, China. His current research interests include the Internet of Things, blockchain applications, edge computing, and distributed systems.



HUANSHENG NING (Senior Member, IEEE) received the B.S. degree from Anhui University, in 1996, and the Ph.D. degree from Beihang University, in 2001. He is currently a Professor and the Vice Dean of the School of Computer and Communication Engineering, University of Science and Technology Beijing, China. His current research interests include the Internet of Things and general cyberspace. He has presided many research projects, including Natural Science Foundation of China, National High Technology Research and Development Program of China (863 Project). He has published more than 150 journals/conference papers, and authored five books. He has been serving as an Associate Editor for IEEE SYSTEMS JOURNAL, since 2013, IEEE INTERNET OF THINGS JOURNAL, from 2014 to 2018, and a Steering Committee Member for IEEE INTERNET OF THINGS JOURNAL, since 2016.



ABDELKARIM BEN SADA received the B.Sc. degree in computer science from the University of Djelfa, Algeria, in 2014, and the M.Sc. degree majoring in networking and distributed systems from the University of Laghouat, Algeria, in 2016. He is currently pursuing the Ph.D. degree with the University of Science and Technology Beijing, China. His research interests include computer vision, machine learning, and the Internet of Things.



SAHRAOUI DHELIM received the master's degree in networking and distributed systems from the University of Laghouat, Algeria, in 2014, and the Ph.D. degree in computer science and technology from the University of Science and Technology Beijing, China, in 2020. He is currently a Senior Postdoctoral Researcher with University College Dublin, Ireland. He was a Visiting Researcher with Ulster University, U.K., from 2020 to 2021. He serves as a Guest Editor for several reputable journals, including *Electronics* journal and *Applied Science* journal. His research interests include social computing, smart agriculture, deep-learning, recommendation systems, and intelligent transportation systems.

...