CNN: A Cluster-based Named Data Routing for Vehicular Networks

SAEID POURROOSTAEI ARDAKANI1, CHIEW FOONG KWONG2, PUSHPENDU KAR1, QIANYU LIU3 AND LINCAN LI2

1School of Computer Science, University of Nottingham Ningbo China Ningbo 315100, China
2Department of Electrical and Electronic Engineering, University of Nottingham Ningbo China Ningbo 315100, China
3International Doctoral Innovation Centre, University of Nottingham Ningbo China Ningbo 315100, China

Corresponding author: Saeid Pourroostaei Ardakani (e-mail: saeid.ardakani@nottingham.edu.cn).

ABSTRACT VANET routing aims to interconnect vehicular nodes via wireless links to transmit network packets. The goal of VANET routing is to reduce the communication cost, decrease the latency and increase the interoperability of the network. In this paper, a cluster-based routing protocol for VANETs called CNN is proposed. It takes advantage of the Hamming distance technique to partition a vehicular network into information-centric clusters based on the mobility of vehicular nodes. However, the proposed approach uses a named data network technique to forward network transmissions according to a hybrid communication model of Dedicated Short Range Communication and Mobile Agent. The latter focuses on reactive intra-cluster link establishment, while the latter proactively forwards inter-cluster transmissions. A simulation measures the performance of the proposed approach and compares the results with two well-known VANET routing protocols: AODV and A-STAR. According to the simulation results, CNN outperforms the benchmarks in average end-to-end delay, path length, data delivery ratio, and total transmitted traffic, especially when the network is dense and the nodes are highly mobile.

I. INTRODUCTION

Recent advances in Vehicular Ad-hoc Network (VANET) address several new applications such as autonomous vehicle navigation, traffic monitoring, and fleet management [1]. Autonomous driving vehicles (e.g., driverless cars or AGVs) collect and analyze ambient data such as environmental conditions, traffic status, road safety, and location information for navigation. They may need to communicate and share their data to avoid accidents, obstacles, and traffic, leading to increased public safety.

VANET includes vehicle nodes that are connected via cellular (e.g., 5G), WiFi, Bluetooth, and Zigbee links. The nodes are typically equipped with sensors/cameras to collect environmental data and use a wide range of on-board transceivers (e.g., simple RFID or satellites) to communicate with Road-Side Unites (RSUs) or with each other [2].

VANET is similar to mobile Ad-hoc networks (MANETs) in terms of routing, as both are self-configuring [3] and [4]. However, additional routing issues need to be addressed for the following reasons [5], [6] and [7]: (1) The speed of VANET nodes is comparatively higher than MANET. (2) Data transmissions are more time sensitive in VANETs than in MANET. Delayed and stale transmissions in VANETs can lead to transport and security risks. (3) In VANETs, there is a predictable mobility pattern as nodes usually move along predefined roads. However, the mobility of MANET is usually random. (4) VANET nodes usually do not suffer from energy constraints as they are continuously powered, whereas MANET nodes suffer from high energy constraints. (5) VANET changes network topology comparatively faster than MANET due to fast mobility. (6) VANET increases the transmitted network traffic compared to MANET due to frequent topology changes. (7) VANETs are large, dense and unbounded while MANETs are usually limited and sparsely distributed.

VANET communication models are classified into two categories: Address-Centric (AC) and Information-Centric (IC) [8], [9]. The former uses the address of the node (e.g., ID) to establish communication paths, while the latter establishes a data link. Named Data Networking (NDN) is an architecture of IC routing [10]. It uses naming of data to establish communication links. An NDN node establishes a route if it has data sets of the same interest or knows a path to the requested content stores. As Figure 1 shows, NDN forwards data interest packets (DI) to establish an IC connection and acquire data content. DI packets update the Pending Interest...
Tables (PIT) at each receiving (intermediate) node if they are new and do not correspond to any PIT record. They are forwarded through the Forwarding Interest Base (FIB) until the content store (i.e., the destination node) is reached [11]. However, a DI packet is discarded if PIT matches a link record. Although NDN routing can provide benefits—especially a loop-free route—it increases network traffic and resource consumption when the network topology changes rapidly [12], [13]. To solve this problem, [14] proposes a distributed adaptive caching strategy over a hierarchical NDN backbone to reduce the amount of transmitted traffic for large and congested vehicular networks.

This article proposes a routing algorithm called Cluster-based Named data routing for vehicular networks (CNN) to address the existing drawbacks of V Anet routing. V Anet routing suffers from frequent changes in network topology. This leads to increased network resource consumption, message collisions, communication delays, and increased network traffic, especially when the network is large and dense and the vehicular nodes are moving fast. To address these drawbacks, CNN combines Dedicated Short Range Communication (DSRC) and Mobile Agent (MAs) routing models and takes advantage of the NDN technique to establish IC links over which network traffic is routed. CNN uses a lightweight and decentralized clustering technique, Hamming distance [15], to support on-demand, context-aware ad hoc network partitioning according to node mobility. It reduces the network transmission since the network clustering technique restricts the network communication according to the convergence principle. However, the cluster heads perform in-network data aggregation to avoid transmission of redundant packets. The DSRC communication establishes connections within the cluster, while the MAs are responsible for the routes between the clusters. The contributions of CNN are described below:

- **CNN takes advantage of the IC connection setup, which results in a higher data delivery ratio.**
- **CNN uses the NDN technique to reduce the data traffic transmitted over the network, resulting in a lower message transmission error rate and greater conservation of network resources, especially bandwidth.**
- **The proposed approach takes advantage of (NDN-enabled) MAs to minimize path length, avoid looping, and reduce communication delay.**

This paper is organized as follows: Section II reviews the routing techniques of V Anets and highlights the existing drawbacks and advances in this research area. Section III presents the remarks, features and techniques of CNN and explains how CNN addresses the existing drawbacks of V Anet routing. Section IV presents the experimental research plan and describes a simulation model to evaluate CNN performance. Section V discusses the evaluation results of the proposed protocol using four key metrics: average end-to-end delay, path length, data delivery ratio, and total transmitted traffic. The performance of CNN is compared with two well-known benchmarks: A-STAR [16] and AODV [17]. Section VI summarizes the contributions of CNN and outlines the main points of this research that should be addressed in further work.

II. RELATED WORKS

This section briefly discusses the V Anet routing categories and highlights their design principles, superiorities, and functionalities. According to Figure 2, V Anet routing protocols are classified based on schemes, network architectures, transmission strategies, and routing information features [18], [19].

![Figure 1. NDN diagram](image1)

![Figure 2. VAnet routing classification](image2)

A. ROUTING SCHEME

There are two V Anet routing schemes: Client/server and mobile agent (MA) [20]. Client/server forms single-hop or multi-hop DSRC ties to forward network traffic through [4]. Optimized Link State Routing (OLSR) [21] is a client/server routing protocol that supports DSRC and forwards network traffic through multi-hop links using a multipoint relaying
technique. Path maintenance, load balancing and scalability are the key drawbacks of client/server routing.

MA routing forwards single/multiple Mobile Agent(s)\(^1\) are deployed in a network. A MA moves autonomously and performs computational tasks. There are two techniques to enable MA mobility: mobile code and remote objects [22]. The former deals with code migration, while the latter enables remote access to data. Agent-based Event-driven Route Discovery Protocol (AERDP) [23] proposes a MA routing through which multiple MAs track mobile objects in a partitioned network.

CNN combines client/server and MA -routing to eliminate the existing drawbacks of both routing schemes. MA -routing suffers from implementation complexity, while client/server routing must consider path maintenance and load balancing. CNN forwards MAs over an RSU backbone to support inter-cluster communication (i.e., multi-hop) and uses client/server only for local transmissions (i.e., single-hop).

B. ROUTING ARCHITECTURE

VANET routing deals with flat and hierarchical path planning according to the network architecture. In the former case, vehicular nodes play the same role, while in the latter case they may have different roles. In flat routing, no special communication infrastructure is built and maintained. It reduces the consumption of network resources. However, flat path planning can eliminate bottlenecks and increase network traffic.

Hierarchical routing builds communication infrastructure such as cluster, tree and chain to support in-network aggregation. It reduces network transmission and resource consumption. Cluster-based routing divides the network into a number of groups that are similar in terms of location, speed, mobility, or communication characteristics of the nodes and connects them through inter/intra-cluster links. Each cluster is managed by a single or multiple cluster heads (CHs). Cluster-based routing offers several advantages, such as in-network packet aggregation, higher message delivery ratio, routing delay reduction, and fair channel assignment [24], [25]. However, a major drawback of cluster-based routing is the cost of maintaining the infrastructure. Landmark ad-hoc routing (LANMAR) [26] forms clusters (i.e., zones) according to the mobility pattern of nodes - mainly direction. In LANMAR, a CH (i.e., landmark) is responsible for communication within and between clusters for each zone.

CNN forms a hierarchical communication infrastructure using a lightweight clustering technique to address the drawbacks of flat routing. It uses low-cost techniques (e.g., Hamming distance) to partition the network and support inter-network aggregation.

C. ROUTING TRANSMISSION STRATEGY

There are three transmission strategies in VANETs: broadcast, multicast, and unicast [9]. Unrestricted broadcast (e.g., flooding) increases the number of transmissions because it forwards network traffic without restriction. Multicast routing avoids flooding but forwards network transmissions to specific destination nodes. Compared to unrestricted broadcast routing, it reduces the traffic transmitted over the network. Location-Aided Routing (LAR) [27] is a limited broadcast routing that forwards network packets to the nodes that are located in a specific area. Unicast routing avoids scheduling multiple paths. It aims to establish a single connection to each target node and forward network traffic through.

VANETs usually use broadcast communication because they have no energy constraints. However, packet broadcast is easy to implement and can cover both unicast (e.g., a single target) and multicast (e.g., several targets) communications. However, it increases overhearing as every node has to repeatedly listen to the wireless channel and check the received network packets.

CNN uses the NDN technique to establish context-aware communication, which reduces the overhead of overhearing, limits broadcasting, and reduces the number of transmissions.

D. TOPOLOGY-BASED ROUTING

Topology-based routing forwards network traffic according to two models: proactive and reactive. Proactive routing (or table-driven routing) aims to collect routing information and store it in tables. It reduces routing delay compared to reactive path planning [20]. However, updating routing tables increases routing cost, network resource consumption and transmitted traffic when nodes are moving fast. Destination-sequenced Distance-Vector (DSDV) [28] is a well-known proactive routing protocol that keeps the path length (hop count) and updates the routing tables in two forms: incremental and full-dump. In the former, only the changed records in the table are changed, while in the latter, the entire table is updated.

Reactive routing builds a path on demand. In fact, in reactive routing, a node forwards a route request when it needs to establish a communication link. This reduces the routing cost and the consumption of network resources. However, reactive routing increases the delay compared to proactive routing. Ad-hoc On-demand Distance Vector (AODV) [17] is a reactive routing protocol that forwards route requests either unicast or multicast as needed. Route requests are recorded and forwarded accordingly through intermediate nodes until destinations are reached.

Hybrid routing combines the key aspects of proactive and reactive routing to reduce routing costs and delays in VANETs. There are two techniques for hybrid routing: proactive zones and network backbone [20]. In the first case, network zones (e.g., clusters) support network communication, while in the second case, a stable network backbone forwards network traffic. CNN takes advantage of a network backbone to support hybrid routing.

\(^1\)MA can be either in the form of software (e.g. code migration) or hardware (e.g. drones). In this research, software MAs.
E. GEOGRAPHIC-BASED ROUTING

Geographic-based (location-based) routing uses location information to forward the network traffic. This results in a reduction in routing traffic transmission and delay. The geographical information is usually obtained from the Global Position System (GPS). As a result, location-based routing usually suffers from GPS availability and accessibility.

Anchor-based Street and Traffic-Aware Routing (A-STAR) [16] establishes the shortest path with maximum connectivity in VANETs. It finds the shortest link between the source node and the destination node using Dijkstra route selection algorithm. For this, A-STAR uses a GPS, to calculate and evaluate the path length of all available routes, and then selects the shortest one if it has the maximum connectivity.

CNN avoids the continuous GPS information collection due to the cost and constraints. It uses RSUs to collect mobility data (e.g., roads, lanes, and directions) on demand and at minimal cost.

III. THE PROPOSED PROTOCOL

In this section, we propose a cluster-based routing protocol called CNN that uses NDN to forward network traffic in VANET. It supports IC communication to reduce network traffic, communication delay and network resource consumption. For this purpose, CNN partitions a vehicular network according to the mobility of nodes (e.g., region and direction) using Hamming distance technique. DSRC communication supports intra-cluster connections and in-network packet aggregation, while MAs forward inter-cluster packets over an RSU backbone. Figure 3 shows a diagram summarizing the CNN methodology including steps, objectives, and techniques.

A. THE NETWORK MODEL

CNN’s network model consists of vehicular nodes and Road-Side Units (RSUs) interconnected by wireless links. The RSUs are typically equipped with roadside sensors/cameras for data collection and wireless transceiver modules for communication. They provide a robust communication infrastructure capable of supporting node connections. Nevertheless, RSUs can provide location information, navigate vehicle nodes, transmit network traffic (e.g., MAs), and report ambient data.

Figure 4 shows a CNN application in which vehicle nodes move through dedicated (urban) streets. They are grouped according to their mobility (e.g., location). Each group is managed by a cluster head (CH) whose task is to perform intra-network data aggregation, intra-cluster communication, and inter-cluster data dissemination (e.g., MA).

| Acronym Definition | Acronym Definition |
|--------------------|--------------------|
| RSU                | Road Side Unit     |
| MA                 | Mobile Agent       |
| DSRC               | Dedicated Short Range Communication |
| LC                 | Location code      |
| CE                 | Cluster Flag       |
| Iter(t)            | Message response timer |
| CR                 | Content Store      |
| CH                 | Cluster-head       |
| CM                 | Cluster-member     |
| RF                 | Routing Table      |
| VH                 | Target Cluster-head |
| PIT               | Pending Interest Tables |
| PIB                | Forwarding Information Base |
| RMA               | Data interest query |
| TMA               | Data interest reply |
| NDN                | Named Data Networking |
| HELLO             | Hello message      |
| ETU               | Average end-to-end delay |
| DOD               | Data Delivery Ratio |
| TtT               | Total transmitted Traffic |

TABLE 1. A Glossary for CNN

B. NETWORK CLUSTERING

CNN uses a distributed clustering algorithm to partition the network according to the mobility of the nodes (i.e., movement region, road ID, and direction). Clustering aims to reduce network traffic as it supports packet aggregation at each CH. However, it reduces packet congestion and collisions, especially when the wireless communication channel is restricted [29]. Network clustering decreases transmitted traffic and consequently minimizes the consumption of network resources, especially bandwidth, power, and memory. Moreover, clustering supports contention-free MAC (Medium Access Control) protocols and provides benefits such as reduced overhearing and idle listening [30].

1) Network Clustering

CNN forms the clusters in a self-configuring manner. According to Algorithm 2, nodes are clustered if they are in the same region and move through the same road ID and lane.
number. For this, Hamming distance technique [15] is used to find similarities in mobility. Here, the number of flipped bits is counted to understand the distance (dissimilarity) of node mobilities. As Algorithm 1 shows, Hamming distance counts the number of different (flipped) bits in the location codes (LCs) containing the address and road/lane number of the node’s moving region. The vehicular nodes receive/update LCs as soon as they find a connection to RSUs. For example, the LC of node B contains {road 1 & lane 1} according to Figure 4. The Hamming distance returns zero if the nodes are on the same road and moving in the same direction, while it shows a non-zero value if the mobility patterns (i.e., region and road/lane numbers) are different. For example, nodes B and C have similar mobility in Figure 4 as both have the same LC {road 1 & lane 1}, but node B {road1 & lane 1} and node X {road 4 & lane 2} have different mobility due to different LCs. Under CNN, nodes are grouped as clusters if they are directly connected and have zero Hamming distance.

Hello (H) messages are simple and lightweight packets used to form clusters. They are commonly used to set up ad hoc networks, join a communication infrastructure, and/or reconnect [20], [31], [32]. In this research, H messages consist of two fields: Data and LC. The former consists of data parameter naming (e.g., data interest), routing information, and clustering data (e.g., data type), while the latter contains information about node location and mobility, especially region ID and road/lane number. Each H message is assigned with a timestamp (tsmp) to avoid looping and support data freshness. Thus, an H message is immediately discarded when tsmp has expired. CHs and the nodes with false cluster flags (CFs) are the only nodes allowed to send an H message. CHs forward H messages to invite next-hop nodes to join their clusters, while false CF nodes use them to join a cluster. They broadcast H messages as soon as an RSU link is available. CF flag is true if the node belongs to a cluster, while it is false if there is no cluster association (e.g., a new node). A node whose CF is false presents itself as CH if no H response is received until the timer Timer(t) expires. Timer(t) is used to avoid starvation and a long wait for joining a cluster. [33] recommends a two-second timer on a single-hop path until the response to an H message is received.

Algorithm 1: Hamming Distance Algorithm

```
/*Calculate Hamming distance between node 1 and node 2. */
Hamming function HAM(LC1 ∧ LC2)
/*starts from the first bit until end.*/
while LC1(i) ≠ LC2(i) do
    if LC1(i) ≠’0’ do
        distance++;
        end
    end
    next bit.*/
i++;
end
return distance;
```

Algorithm 2: Clustering Algorithm

```
/*A CH is leaving.*/
if (my.ID == my.CH) then
    Routing table(RT) is ranked according to connectivity.*/
    SRT ← Sort.RT(Count);
/*Top record is selected as CH */
if (SRT.top() == null) then
    NewCH ← Reset Timer();
    Reply(Call-On-Duty) to NewCH;
    Broadcast ("id is the new CH, NOW", NewCH);
else
    /"No one around, forms/joins a cluster."*/
    my.CH ← null;
    Broadcast ("Leave");
end
```

Algorithm 3: Cluster Leaving Algorithm

```
/*Location is updated.*/
my.LC ← GetLocation (RSU);
/*Not belong to a cluster.*/
while (my.CF == false) do
    if (H is received while waiting.*)
        Message freshness check.*
        if (tsmp == true) then
            Calculate Hamming Distance.*
            HD ← HAM(my.LC ∧ H.LC);
            Same mobility pattern.*
        else
            H is received while waiting.
        end
        if (H.ID == 0) then
            RT.update(H);
            CF ← true;
            join cluster.*
            if (H.DATA (CH) == null) then
                my.CH = H.DATA(CH);
            else
                old H message.*
                my.CH = my.ID;
                CF ← false;
                Reset Timer();
                Broadcast (H);
            end
        end
        if (timer(t) expired.*)
            No one around, forms/joins a cluster.*
            my.CH ← null;
            Broadcast ("Leave");
        end
```

2) Leaving Cluster

The topology of the VANET changes rapidly due to the mobility of the nodes. This means that VANET clusters are frequently updated as vehicular nodes lose links within clusters due to changes in direction or region. Therefore, nodes set CF to false before leaving a cluster. The leaving node is detected as disconnected until it joins a new cluster or forms a new cluster. As Algorithm 3 shows, leaving nodes may need to take different actions depending on their role (CH or CM):

- A CH should find a replacement to keep the cluster...
interconnected as much as possible. To this end, CH orders the routing table entries according to the degree of connectivity. Therefore, a node with maximum connectivity is selected as the next CH candidate if the ranked list is not empty. The candidate CH rebuilds the cluster to cover the maximum number of CMs in a self-configuring and decentralized manner. This prevents a cluster from breaking into isolated nodes when a CH is eliminated. However, small clusters are formed if the candidate CH has a low connectivity degree. Section III-B3 addresses how CNN combines small clusters into larger clusters.

- A simple CM leaves a cluster by broadcasting a leave message to inform the CH and neighboring CMs to update their routing tables. The leaving CM resets its CH’s ID to null and CF to false and then leaves the cluster.

3) Cluster Combination

Frequent cluster updates have to do with the instability of the network topology and bring disadvantages - mainly the consumption of network resources. To solve this problem, CNN combines clusters with the goal of reducing the number of (isolated) nodes/clusters. As Algorithm 4 shows, neighboring clusters are combined into one large cluster if they share the same mobility (e.g., the same LC). This leads to a reduced number of clusters and consequently to increased in-network data aggregation and reduced network transmissions. To this end, CHs that are directly linked and have a Hamming distance of zero participate in a cluster combination. In this way, each CH advertises its connectivity degree using an H message as soon as an RSU link is available. The receiving node (i.e., CHs or nodes with false CF) joins the advertising cluster and updates its neighbors (e.g., CMs) if it has a lower connectivity degree, while it replies to the message and asks the sender node to join if it has a higher connectivity degree.

```javascript
/* Location is updated.*/
my.LC ← GetLocation(RSU);

/*CH advertises itself using Hello (H) message.*/
Broadcast (H);

/* Calculate Hamming Distance.*/
HD ← HAM(my.LC ∧ H.LC);

/* Same mobility pattern.*/
if (HD == 0) then
  /* a bigger cluster is found.*/
  if (H.DATA(CF) == true) ∧ (H.DATA(count) ≥ my.RT(count)) then
    /* join cluster.*/
    my.CH = H.DATA(CH);
    RT.update(my.CH);
    Broadcast {(my.CH)} ;
  end
  else
    /* invite H’s sender to join.*/
    Reply (my.ID) to H’s sender;
  end
else
end

Algorithm 4: Clustering Combination Algorithm
```

C. CNN ROUTING

In this section, we present a CNN routing approach in which network packets are classified into two classes: Data Interest Query (DIQ) and Data Interest Reply (DIR). DIQs are forwarded from source nodes to request VANET routing information such as road safety, traffic conditions, and/or parking availability, while DIRs are routed back from targets (i.e., CSs) to source nodes to provide the requested data.

DIQ and DIR are structured in four fields: Source Mac Address, Data Interest/Response Content, Relay Mac Address, and Packet Timer. MAC addresses are unique IDs to identify the nodes. Data interest/reply allows nodes to check or update the message content according to IC routing and FIB strategy. Relays are the last interconnected nodes on each path. They are used for immediate response and establish a return path. This allows nodes to maintain a return link with minimal cost.

A timer is assigned to each network packet to provide data freshness and avoid looping. CNN assumes that the nodes are synchronized and the source nodes reset the timer before each transmission. According to [3], the allowable interval for lost messages is one second for IEEE 802.11. Thus, the timer is set to one second to support data and avoid loss of data packets. This means that a packet is still fresh and should be forwarded for further processing if it is delivered within one second. However, in real applications, data packets may have different priorities that provide a different interval for packet loss. Some messages (e.g., accident alerts) have a high priority and should be delivered with minimal latency, while others have a low priority (e.g., parking availability). CNN assumes that all network packets have the same priority and uses a packet loss time (one second). However, Section VI addresses a future work to extend CNN where routing messages have different priorities.

1) Intra-cluster Routing (NDN)

As Algorithm 5 shows, source nodes take advantage of NDN routing to forward DIQs. To this end, each CH checks PIT, to find a record that matches the DIQ. It updates PIT, discards DIQ, and forwards back a DIR if the DIQ record matches PIT. Otherwise, it sends a MA, which moves through the RSU infrastructure to find the requested CS in another cluster. Expired DIQs (or DIRs) are immediately discarded to support data freshness and to avoid a loop.

DIRs are forwarded from CSs (e.g., target nodes) to source nodes, using Algorithm 6 takes two approaches: DSRC and MA. DSRC is used for intra-cluster forwarding DIR, where a node checks PIT for a matching record when it receives DIR. The node drops the DIR packet if there is a PIT record. Otherwise, it sends a MA, which moves through the RSU infrastructure to find the requested CS in another cluster. Expired DIQs (or DIRs) are immediately discarded to support data freshness and to avoid a loop.

Intra-cluster communication and data dissemination (e.g., DIR and DIQ routing) are supported by MAs. A MA consists of

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/
Algorithm 5: NDN enabled DIQ Dissemination Algorithm

```
/* A DIQ is received; timer(T) is not expired yet. */
while ((DIQ != null) ∧ (T ≥ 0)) do
    /* Content store is found. */
    if (CS ∈ DIQ) then
        /* Data query is replied. */
        printf("DI query is successfully received.");
        Forward DIR\{my.ID, Reply, DIQ.source, T.reset()\};
    end
    /* Otherwise, find a PIT record. */
    else
        /* A PIT found. */
        if (DIQ ∈ PIT) then
            PIT.update(DIQ);
            Discard DIQ;
        end
        /* No PIT found. */
        else
            PIT.insert(DIQ);
            FIB.update();
            Forward DIQ\{my.ID, T.reset()\};
        end
end
```

Algorithm 6: NDN enabled DIR Dissemination Algorithm

```
/* A DIR is received; timer(T) is not expired yet. */
while ((DIR != null) ∧ (T ≥ 0)) do
    /* DIR matches data query. */
    if (my.ID == my.CH) then
        /* A MA is forwarded. */
        MA ← \{(MA.ID, my.ID, next.RSU, T.reset()), DIQ, MA.Code, itinerary(EMPTY)\};
        PIT.insert(DIQ);
        FIB.update();
        Forward DIQ\{my.ID, T.reset()\};
    else
        PIT.delete(DIQ);
        /* DIQ source is not directly interconnected. */
        if (DIR ∈ PIT) then
            /* Immediate reply, source is still interconnected. */
            if (DIQ.source ∈ RT) then
                PIT is updated by removing the replied DiQ.*/
                PIT.delete(DIQ);
                /* Reply is forwarded to DIQ.source. */
                Forward DIR\{DIQ.source, Reply, my.ID, T.reset()\};
            end
            /* DIQ.source is not directly interconnected. */
            if (my.ID == my.CH) then
                /* MA is forwarded. */
                MA ← \{(MA.ID, my.ID, next.RSU, T.reset()), DIR, MA.Code, itinerary(EMPTY)\};
            end
            /* This is a CH. */
            if (DIQ.source ∈ CH) then
                /* MA is forwarded. */
                MA ← \{(MA.ID, my.ID, next.RSU, T.reset()), DIR, MA.Code, itinerary(EMPTY)\};
            end
            /* This is a CM. */
            else
                /* Forward DIQ\{my.ID, T.reset()\} is forwarded to CH; */
                Discard DIQ;
            end
            /* No PIT found. */
        else
            Discard DIR;
        end
    end
end
```

of four main components: identity, data, code, and path. Identity contains the basic information of MA such as MA, ID, dispatcher and time. MA, data addresses the DIQ or DIR parameters, while code contains MA migration algorithm and routing information. Path records the MA’s traversed path list of visited nodes throughout a journey. As Algorithm 7 shows, CHs dispatch MAs when inter-cluster communication is required to forward DIQ/DIR. A CH receives and aggregates data requests from CMs and then sends MAs throughout the network to reach the target nodes.

CHs forward MAs to the available RSUs according to an NDN routing. The RSU checks its routing table and forwards the MA if it matches the MA’s DI and the target cluster is still interconnected. Otherwise, the RSU finds out the cluster’s movement pattern (i.e., direction), computes the LC of target CH, and forwards the MA. However, the RSU clones the MA into multiple copies and forwards it to the next RSUs if finds no DI. Although cloning MA is avoided in resource-constrained networks (e.g., wireless sensor networks) [6], it has the potential to increase the parallelism of data dissemination when nodes are equipped with continuous power resources (e.g., VANET). The same algorithm forwards back the DIR MA to the source cluster once the target is captured. CNN monitors MA itineraries to avoid a loop if a particular RSU is visited more than once. MA loops can cause network congestion, packet collisions, and consume network resources.

IV. EXPERIMENT METHODOLOGY

Simulations are used to test and evaluate the performance of CNN. The reason is that real vehicular networks are very expensive in terms of communication infrastructure, navigation platform and route planning. Moreover, real VANETs are capable of handling public safety risks such as accidents and road congestion. Accordingly, simulations are often used to test VANET routing protocols.

OMNET++ [34] is a discrete event simulator that has an open source framework called INET [35] to support node mobility, MAC protocol and communication model. It provides the implementation of well-known VANET routing protocols (e.g., AODV). Therefore, OMNET++ was chosen to implement, test and evaluate CNN. Nevertheless, a Metropolitan Grid (M-Grid) Manhattan model [36] is used to simulate a grid-based road infrastructure through which vehicular nodes move.

As Figure 5 shows, three scenarios are studied in the simulation experiments: sparse, medium, and dense networks with 50, 250, and 500 nodes, respectively. In this way, the scalability and performance of CNN are studied as a function of the change in network density. Each vehicle is assigned a unique ID and is randomly localized in a (12.5km × 12.5km) field. It moves through the predefined roads provided by the M-Grid mobility pattern. According to Figure 6, the grid infrastructure consists of 500m squares (e.g., city buildings) and 100m two-lane roads. The vehicle nodes are assigned a constant speed for each simulation.
/*A MA is received.*/
while (MA.ID \notin RT) do
  RT.update(MA);
  /*RSU is on the same path.*/
  if (RSU.ID == MA.(RSU)) then
    /*Find a matched record.*/
    RT.update(MA);
    /*Target CH is found.*/
    TCH ← RSU.RT(DIQ/R);
    /*Interconnected CH.*/
    if (RT{TCH.flag} == true) then
      MA ← {(MA.ID, DispatcherID, LC, RSU.ID, TCH, T.reset()), Data(DIQ/R), Code, Itinerary.update());
      Forward MA (TCH);
      /*Not interconnected.*/
    else
      RSU.ID ← RSU.TCH.LC;
      MA ← {(MA.ID, DispatcherID, LC, RSU.ID, T.reset()), Data(DIQ/R), Code, Itinerary.update());
      Forward MA (RSU.ID);
    end
    /*No matching record.*/
  else
    /*Check for Loops.*/
    if (RSU \in MA.Itinerary()) then
      /*MA copies are forwarded.*/
      for (∀ RSU \in (RT.RSU.flag == true)) do
        MA ← {(MA.ID, DispatcherID, LC, RSU.ID, T.reset()), Data(DIQ/R), Code, Itinerary.update());
        Forward MA (RSU.ID);
      end
      /*Loop found!*/
    else
      Discard MA;
    end
    /*Wrong RSU is reached.*/
  else
    Discard MA;
  end
end

Algorithm 7: MA Migration Algorithm

scenario: slow (5 m/s), moderate (15 m/s) and fast (30 m/s). The RSUs are localized at the middle of intersections to form the communication backbone.

A Carrier Sense Multiple Access (CSMA/CA) MAC protocol is used in the simulation experiments to enable wireless channel sharing and avoid collisions. In addition, a line-of-sight (LOS) wireless signal propagation model is used for all simulation experiments. This assumes no wireless interference from physical obstacles (e.g., trees and buildings) and ambient factors (e.g., noise and weather). For each experiment, source and target nodes are randomly selected to simulate data packet dissemination.

The statistical power analysis technique [37] is used to calculate the required number of experiments. This technique uses the standard deviation of the population for a subset of experiments/samples to calculate the number of repetitions according to an acceptable confidence level. For this purpose, the simulation results of 10 repetitions (the subset of experiments) are collected and analyzed. According to the subset results, at least 150 repetitions (sample size) are recommended to achieve a confidence level of 90% for all experiments. Table 2 summarizes the simulation parameters.

### TABLE 2. Simulation parameters

| Parameter | Range | Parameter | Range |
|-----------|-------|-----------|-------|
| Simulation time | 3600 s | Message intervals | 200 s |
| Repetition | 150 | Ambience noise and Obstacles | disabled |
| Communication Range | 200 m | Distribution Model | random |
| RSUs | 600m (grid) | Area Size | 12.5 km x 12.5 km |
| MAC Protocol | CSMA/CA | Communication Protocol | IEEE 802.11 |
| Mobility Model | M-Grid | Road Selection | minimum traffic |
| Vehicle Count | 50, 250, 500 | Speed | 5, 15, 30 m/s |

Four metrics are measured to evaluate CNN performance: average end-to-end delay, route length, data delivery ratio, and total traffic transmitted. These metrics are commonly used to evaluate the performance of VANET routing protocols in the literature [9], [38], [39], [13].

1) **Average End-to-End Delay (ETE):** It measures the delay since a packet leaves a node until it reaches the target. ETE is affected by the number of intermediate hop count, node mobility, packet transmission latency, bandwidth availability, and network traffic. Data accuracy and freshness are achieved when ETE is minimized.
2) Route Length (RL): this is the number of intermediate hop count from a source to a target node (e.g., CS). RL is collected to study the routing protocol’s ability to avoid random/blind path selection and establish paths with minimum hop count paths. A reduced RL will result in a lower ETE.

3) Data Delivery Ratio (DDR): counts the number of successful replied DIQs. A data query is replied if the routing approach successfully establishes a path between a source and target. The DDR can be affected by the MAC protocol, network traffic, available bandwidth, and routing algorithm design.

4) Total transmitted Traffic (TtT): it represents the total number of network messages (e.g., control and data packets) transmitted during the routing process. An increase in TtT results in increased consumption of network resources - mainly bandwidth, increased packet congestion and reduced DDR.

V. RESULTS AND DISCUSSIONS

The performance of the CNN is measured and compared with two benchmarks (A-STAR [16] and AODV [17]) using four metrics: ETE, RL, DDR, and TtT. A-STAR was developed specifically for VANETs in urban areas with M-grids, while AODV was implemented in the real world and has a verified implementation model on OMNET++ (INET) [40].

A. AVERAGE END-TO-END DELAY

VANET routing aims to reduce average End-to-End Delay (ETE) by reducing the traffic transmitted over the network and the number of path hop count. It provides data freshness benefits to increase public safety in VANETs.

According to the Figure 7, CNN outperforms AODV especially when the speed of the node increases. This is because a hybrid routing scheme is used to forward network traffic. CNN forwards network transmissions (e.g., MAs) over the RSU infrastructure according to a proactive routing. This results in ETE reduction compared to reactive routing (e.g., AODV), which needs to collect routing information on demand for each transmission. As the Figure shows, the delay in AODV depends on the number and speed of intermediate nodes involved in routing. This is due to the routing delays caused by frequent changes in the network topology. AODV has to repeatedly collect routing information that is constantly changing due to the mobility of the nodes.

CNN outperforms A-STAR especially when the network is dense and the topology changes frequently. This is because CNN uses the clustering technique. As mentioned earlier, clustering has the capacity to offer benefits -mainly in-network packet aggregation. It results in reduced network transmission which reduces wireless channel access and packet transmission delays. Therefore, the ETE in cluster-based VANETs (e.g., CNN) is lower compared to flat A-STAR.

CNN underperforms A-STAR, when nodes move slowly. A-STAR establishes the shortest path for forwarding network messages between source and target nodes. It reduces the path length and hence reduces the ETE. The path stays interconnected longer time if the nodes move slowly and the network topology changes infrequently. In fact, slow changes in network topology lead to minimal route computation and hence lower ETE. However, A-STAR needs to update and rebuild the shortest paths frequently when the nodes move fast and the network topology changes rapidly. This increases the ETE of A-STAR compared to CNN.

B. ROUTE LENGTH

According to Figure 8, CNN minimizes RL compared to benchmarks when the deployed network is dense and the topology changes rapidly. The proposed protocol uses clustering to support in-network aggregation, reducing the number of route hops. In this way, source nodes avoid establishing flat routes to forward network transmissions to the targets. However, they take advantage of cluster-based routing to forward messages such as DIQ hierarchically. In this way, CHs collect and aggregate network messages (e.g., DIQ) to reduce the number of hops traversed between source and target nodes. Moreover, CNN reduces RL since it uses MA to support loop-free path planning. However, A-STAR uses Dijkstra algorithm to select the shortest path. This reduces RL compared to CNN and AODV when the network topology changes slowly and route re-establishment is minimized.

As the results show, AODV increases the RL compared to A-STAR and CNN. This is because AODV builds the network paths heuristically. Therefore, path hop-count is greatly increased in dense networks, especially when the topology changes rapidly.
C. DATA DELIVERY RATIO

According to Figure 9, CNN increases the DDR compared to the benchmarks when the deployed network is dense and/or the node speed increases. This is due to the benefits of using RSU infrastructure to transmit network traffic in the proposed protocol. CNN reduces the number of message failures as it forwards messages over stable links. It also uses clustering to limit concurrent access to wireless channels. As a result, the CMs do not have to forward the data packets directly to target nodes; instead, they transmit the data packets to the CHs for in-network aggregation. This minimizes network transmission, wireless channel access failure, and network packet congestion in the network compared to the benchmarks.

DDR is reduced in AODV when the network topology changes rapidly. This is due to the fast mobility of nodes and frequent disconnections. In fact, frequent communication outages reduce packet delivery in AODV.

D. TOTAL TRANSMITTED TRAFFIC

As Figure 10 shows, CNN outperforms AODV and A-STAR, when the speed of mobility increases. CNN uses RSU infrastructure for inter-cluster communication. It is a stable infrastructure that does not require control message to be transmitted to re-establish/update links. Also, CNN uses clustering to reduce the number of messages within a cluster. Clustering supports in-network aggregation and avoids pointless/redundant data transmissions. In addition, CNN forwards the network packets using NDN routing. Thus, intermediate nodes avoid retransmission of redundant and meaningless data interest packets. This results in lower TtT. However, AODV frequently forwards control messages to create reactive routes. This drastically increases the TtT when nodes are moving fast and network routes are frequently updated/re-established. Nevertheless, AODV increases the TtT in dense networks. This is due to the increased number of intermediate nodes which broadcast network packets (e.g., control messages).

A-STAR has better performance compared to CNN when nodes are moving slowly. This is because A-STAR avoids clustering. This leads to reduction in network transmission as compared to CNN which transmits multiple control messages to form clusters. However, the TtT increases in A-STAR, when the speed of the nodes increases. This is due to the increased number of network messages that are transmitted frequently to update the failed routes.

VI. CONCLUSION AND FURTHER WORK

CNN proposes a (hybrid) cluster-based routing approach for VANETs. It partitions the network according to the movement region of the vehicular nodes, road ID, and lane number (direction) using Hamming distance technique. However, CNN uses NDN routing to reduce network transmissions (TtT) and save network resources, mainly bandwidth. It uses DSRC and MA communication techniques to forward network traffic throughout the network. MAs are used for inter-cluster links over an RSU infrastructure, while DSRC provides intra-cluster communications. CNN outperforms the benchmarks (i.e., AODV and A-STAR) for ETE, RL, DDR, and TtT when the deployed network is dense and nodes are moving fast.

The performance of CNN still needs to be improved by multi-hop clustering. CNN partitions the network using...
Hamming distance technique which forms single-hop clusters. However, multi-hop clustering can provide additional benefits as the clusters are formed with a larger number of CMs and a smaller number of CHs. This results in lower routing costs, especially for inter-cluster communication.

CNN should support IEEE 802.11p communication protocol in the future. The OMNET++ simulation model deals with a IEEE 802.11 wireless communication for mobile ad hoc networks. However, this model needs to be updated as IEEE 802.11p that supports VANETs and improves the reliability of the results.

CNN must use a network synchronization protocol to forward network traffic. Synchronization is challenging in mobile ad hoc networks (e.g., VANETs), especially in terms of resource consumption and communication overhead. CNN needs to synchronize nodes with minimum cost. It can take advantage of partial synchronization approaches that can synchronize only a subset of the network that participates in NDN routing.

Further research is needed to perform additional simulation scenarios according to the non-line-of-sight (NLOS) communication model. CNN uses the line-of-sight (LOS) model, which assumes no wireless interference from obstacles and ambient noise. However, CNN has yet to investigate the impact of obstacles (e.g., surrounding buildings) on the quality of wireless communications in VANETs.

REFERENCES

[1] L. A. Maglaras, A. H. Al-Bayatti, Y. He, I. Wagner, and H. Janicke, “Social internet of vehicles for smart cities,” Journal of Sensor and Actuator Networks, vol. 5, no. 1, 2016. [Online]. Available: https://www.mdpi.com/2224-2708/5/1/3

[2] A. Colakovic and M. Hadzialic, “Internet of things (IoT): A review of enabling technologies, challenges, and open research issues,” Computer Networks, vol. 144, pp. 17–39, 2018.

[3] A. Srivastava and B. Chaurasia, “Survey of routing protocol used in vehicular ad hoc networks,” International Journal of Current Engineering and Technology, vol. 7, no. 3, pp. 1051–1062, 2017.

[4] S. Al-Sultan, M. M.A.-Al-Doori, A. H.Al-Bayatti, and H. Zedan, “A comprehensive survey on vehicular ad hoc network,” Network and Computer Applications, vol. 37, pp. 380–392, 2014.

[5] P. Bedi and V. Jindal, “Use of big data technology in vehicular ad-hoc networks,” International Conference on Advances in Computing, Communications and Informatics (ICACCI), New Delhi, India, 01 December 2014.

[6] S. P. Ardakani, J. Padget, and M. D. Vos, “A mobile agent routing protocol for data aggregation in wireless sensor networks,” International Journal of Wireless Information Networks, vol. 24, no. 1, pp. 27–41, 2017.

[7] S. P. Ardakani, “Acr: A cluster-based routing protocol for vanet,” International Journal of Wireless & Mobile Networks, vol. 10, no. 2, pp. 39–53, 2018.

[8] X. Liu, Z. Li, P. Yang, and Y. Dong, “Information-centric mobile ad hoc networks and content routing: A survey,” Ad Hoc Networks, vol. 58, pp. 255–268, 2017.

[9] L. R. H. Guo, H. Huang, R. Shi, and X. Qiu, “Mupf: Multiple unicast path forwarding in content-centric vanets,” Ad, vol. 81, pp. 211–225, 2018.

[10] K. Ahed, M. Benamar, A. A. Lahcen, and R. E. Ouaazzani, “Forwarding strategies in vehicular named data networks: A survey,” Journal of King Saud University - Computer and Information Sciences, jul 2020.

[11] M. Amadeo, C. Cam polo, and A. Molinaro, “Forwarding strategies in named data wireless adhoc networks: Design and evaluation,” Journal of Network and Computer Applications, vol. 50, pp. 148–158, 2015.

[12] D. O. Mau, Y. Zhang, T. Taleb, and M. Chen, “Vehicular inter-networking via named data - an opnet simulation study,” Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, LNCS, vol. 137, pp. 116–125, 05 2014.

[13] S. P. Ardakani, “MINDS: Mobile agent itinerary planning using named data networking in wireless sensor networks,” Journal of Sensor and Actuator Networks, vol. 10, no. 2, p. 28, apr 2021.

[14] N. Yang, K. Chen, and Y. Liu, “Towards efficient NDN framework for connected vehicle applications,” IEEE Access, vol. 8, pp. 60 850–60 866, 2020.

[15] Z. Xiong, A. D. Liveris, and S. Cheng, “Distributed source coding for sensor networks,” IEEE SIGNAL PROCESSING MAGAZINE, vol. 21, no. 5, pp. 80–94, 2004.

[16] G. Liu, B. sung Lee, B. chong Seet, C. heng Foh, and K. kee Lee, “A routing strategy for metropolis vehicular communications,” International Conference on Information Networking (ICOIN), Busan, Korea, February 19-20, 2004.

[17] C. E. Perkins and E. M. Royer, “Ad-hoc on-demand distance vector routing,” Second IEEE Workshop on Mobile Computer Systems and Applications (WMCSA ’99), New Orleans, Louisiana, USA, February 25-26, 1999.

[18] M. Altayeb and I. Mahgoub, “A survey of vehicular ad hoc networks routing protocols,” International Journal of Innovation and Applied Studies, vol. 3, no. 3, pp. 829–846, 2013.

[19] J. Wang, X. Xiao, and P. Lu, “A survey of vehicular ad hoc network routing protocols,” Journal of Electrical and Electronic Engineering, vol. 7, no. 2, pp. 46–50, 2019.

[20] S. P. Ardakani, “Wireless sensor network routing protocols for data aggregation,” Ph.D. dissertation, Computer Science Department, University of Bath, 2014.

[21] P. Jacquet, P. Muhlethaler, T. Clausen, A. Laouiti, and A. Qayyum, “Optimized link state routing protocol for ad-hoc networks,” IEEE INMIC 2003, Lahore, Pakistan, pp. 62-68, 2003.

[22] E. P. de Freitas, “Cooperative context-aware setup and performance of surveillance missions using static and mobile wireless sensor networks,” Ph.D. dissertation, Halmstad University, 2011.

[23] P. SETHI, D. JUNEJA, and N. CHAUHAN., “A mobile agent-based event driven route discovery protocol in wireless sensor network: Aerdp,” International Journal of Engineering Science and Technology (IJEST), vol. 3, no. 12, pp. 8422–29, 2011.

[24] A. Abuashour and M. Kadoch, “Performance improvement of cluster-based routing protocol in vanet,” IEEE Access, vol. 5, pp. 15 354 – 15 371, 2017.
[25] M. M. Nasr, A. M. S. Abdelgader, Z.-G. Wang, and L.-F. Shen, “VANET clustering based routing protocol suitable for deserts,” Sensors, vol. 16, no. 4, pp. 478–501, 2016.

[26] G. Pei, M. Gerla, and X. Hong, “Lanman: Landmark routing for large scale wireless ad hoc networks with group mobility,” in First Annual Workshop on Mobile and Ad Hoc Networking and Computing (MobiHOC), 2000, 11th August, Boston, USA, 2000, pp. 11–18.

[27] Y.-B. Ko and N. H. Vaidya, “Location-aided routing (lar) in mobile ad hoc networks,” Wireless Networks, pp. 307–321, vol. 6(4), 2000.

[28] C. E. Perkins and P. Bhagwat, “Highly dynamic destination-sequenced distance-vector routing (dsdv) for mobile computers,” SIGCOMM, vol. 24, no. 4, pp. 234–244, 1994.

[29] C. Cooper, D. Franklin, M. Ros, F. Safaei, and M. Abolhasan, “A comparative survey of vanet clustering techniques,” IEEE Communications Surveys and Tutorials, vol. 19, no. 1, pp. 657 – 681, 2016.

[30] Y. Qian, K. Lu, and N. Moayeri, “A secure vanet mac protocol for dsrc applications,” IEEE Conference on Global Telecommunications, IEEE GLOBECOM’8, New Orleans, LA, USA, 30 November-4 December., pp. 1–5, 2008.

[31] S. S. A. Basurra, “Collision guided routing for ad-hoc mobile wireless networks,” Ph.D. dissertation, Computer Science Department, University of Bath, UK, 2012.

[32] S. P. Ardakani, J. Padget, and M. D. Ves, “CBA: A cluster-based client/server data aggregation routing protocol,” Ad Hoc Networks, vol. 50, pp. 68–87, nov 2016.

[33] I. D. Chakeres and E. M. Belding-Royer, “The utility of hello messages for determining link connectivity,” The 5th International Symposium on Wireless Personal Multimedia Communications, Honolulu, HI, USA, 27-30 October, 2002.

[34] OMNET+++, “Omnet++ simulator,” 2019, http://www.omnetpp.org/, Retrieved (March 2019).

[35] INET, “Inet framework,” 2019, https://inet.omnetpp.org/introduction.html, Retrieved (March 2019).

[36] F. J. Martinez, J.-C. Cano, C. T. Calafate, and P. Manzoni, “Citymob: A mobility model pattern generator for vanets,” in IEEE International Conference on Communications Workshops, Beijing, China, Jun. 2008.

[37] E. Serqeant, “Sample size to estimate a single mean with specified precision,” 2014, http://epitools.ausvet.com.au/content.php?page=1Mean, Retrieved (March 2019).

[38] S. Mohapatra and P. Kanungo, “Performance analysis of aodv, dsr, olsr and dsdv routing protocols using ns2 simulator,” Procedia Engineering, vol. 30, pp. 69–76, 2012.

[39] M. Chen, D. O. Mau, Y. Zhang, T. Taleb, and V. C. M. Leung, “Vendnet: Vehicular named data network,” Vehicular Communications, vol. 1, pp. 208–213, 2014.

[40] OMNET+++, “Aodv in omnet++,” 2020, https://doc.omnetpp.org/inet/api-current/neddoc/inet.routing.aodv.Aodv.html, Retrieved (June 2020).

CHIEW FOONG KWONG received his B.Eng. (Hons) in Electrical and Electronic Engineering from the University of Hertfordshire, UK, and received his M.Sc. in Personal, Mobile and Satellite Communications from the University of Bradford, UK, in 2001 and 2004, respectively. He received his Ph.D. in wireless communications from Multimedia University, Malaysia in 2015. He is currently an Associate Professor at the University of Nottingham Ningbo China. His research interest is in heterogeneous networks, mobility management for LTE, 5G and V2X. He has published a number of papers in journals and conferences in this area.

PUSHPENDU KAR is an Assistant Professor with the School of Computer Science, University of Nottingham Ningbo China. Prior to this, he was a Research Fellow with the Department of ICT and Natural Sciences, Norwegian University of Science and Technology, Norway. He was also a Post-doctoral Research Fellow with the Department of Electrical and Computer Engineering, National University of Singapore, and the Energy Research Institute, Nanyang Technological University, Singapore. His research interests include Mobile Ad Hoc Networks, Wireless Sensor Networks, Internet of Things, and Content-Centric Networking. He is a recipient of the Erasmus Mundus Postdoctoral Fellowship of the European Commission, the European Research Consortium for Informatics and Mathematics Alain Bensousan Postdoctoral Fellowship of the European Union, and the Science and Engineering Research Board Overseas Postdoctoral Fellowship of the Department of Science and Technology, Government of India. He received 2020 IEEE Systems Journal best paper award. He is an IEEE Senior Member.

QIANYU LIU received the MSc degree in electronic communications and computer engineering from the University of Nottingham Ningbo China in 2017. He is currently pursuing his PhD degree at the University of Nottingham Ningbo China. His research interests include mobility management in 5G, handover management, and self-organisation networks.

LINCAN LI received the M.S.degree in Electronic Communications and Computer Engineering from the University of Nottingham Ningbo China in 2017. He is currently a PhD student in the faculty of Science Engineering in the University of Nottingham Ningbo China. His research area is about the mobile edge cache.

SAEID POURROOSTAEI ARDAKANI received the Ph.D. degree in computer science from the University of Bath, U.K., in 2015. He currently works with the University of Nottingham Ningbo China (UNNC), as an Assistant Professor in computer science. His research interests include design and implementation of various aspects of networking, including simulation, resource management, synchronization, and routing. He is also interested in big data analysis, blockchain, and the Internet of Things (IoT) applications.