# OpenL3: Embedding Diverse Network Services Into MANETs Using Multidimensional Identifier

Jiangyu Lan, Shuai Gao, *Member, IEEE*, Weiting Zhang, *Member, IEEE*, Xindi Hou, Minghui Xi, Yuming Zhang, *Member, IEEE*, Bo Lei, Hongke Zhang, *Fellow, IEEE*, and Xuemin Shen, *Fellow, IEEE*

***Abstract*—Practical applications in mobile ad-hoc networks (MANETs) require the support of diverse network services, e.g., host-centric, content-centric, and location-centric routing and forwarding services. However, existing solutions are typically designed over a single network service rather than integrated ones. To embed diverse network services into MANETs, the major challenge is enabling interoperability among various network-layer (L3) protocols without suffering complexity and scalability issues. In this article, we propose OpenL3, a programmable L3 approach to support the coexistence of diverse network services in MANETs. Specifically, OpenL3 first abstracts key attributes from network entities, such as content, locations, or groups of devices. These attributes are embedded into a network address, named multidimensional identifier (MID), to control the routing and forwarding processes. Then, a distributed MID mapping system is established to facilitate efficient MID registration and query. Based on the MID, a programmable routing and forwarding scheme is proposed, which incorporates a lightweight packet processing design using a P4 programmable data plane to enable interoperability among various L3 protocols. A cluster of SDN-based control plane devices collaboratively distribute flow rules to manage data plane behavior. Furthermore, a prototype system is built to implement and evaluate the proposed solutions. Experimental results show that OpenL3 outperforms the existing solutions in terms of end-to-end latency and network throughput while being deployable in MANETs without modifications to network protocols or sockets.**

***Index Terms*—Mobile ad-hoc networks (MANETs), network identifiers, network services, P4, SDN.**

## I. Introduction

MOBILE ad-hoc networks (MANETs) consist of numerous mobile nodes and provide communication in situations lacking fixed infrastructure [1]. The advancement of MANETs has enabled emerging applications, such as data collection [2] and emergency response [3], driving high-communication demands. Consequently, enhancing the performance of network services, including routing and forwarding services, has become a research hotspot for these applications in MANETs [4].

Existing MANET routing and forwarding solutions typically rely on the host-centric mechanism, which assigns network identifiers, such as IP addresses, to all hosts or nodes [5]. However, these solutions have been proven inefficient for the applications above [6], [7], [8]. For data collection, source nodes are required to retrieve data from the target nodes identified by the destination address, but intermittent connectivity in MANETs typically leaves the target nodes unreachable. For emergency response, source nodes need to multicast toward nodes in a specific geographic area. Yet, the inherent mobility of nodes in MANETs leads to frequent changes in the network topology, necessitating regular updates to multicast group members, which results in significant overhead.

To address these issues, researchers have proposed various nonhost-centric solutions. For example, Dowling et al. [9] utilized an NDN-based [10] content-centric mechanism, employing content names as network identifiers for network-wide data retrieval instead of targeting specific nodes. Sandonis et al. [11] implemented a location-centric mechanism, using locations as network identifiers to facilitate direct communication among nodes in specific geographic areas without maintaining a multicast group. These solutions optimize the routing and forwarding process by customizing network identifiers for specific applications, improving data retrieval efficiency, and reducing multicast overhead. Consequently, a well-designed network identifier, along with corresponding routing and forwarding solutions, can effectively enable MANETs to support specific applications.

However, in real-world scenarios, multiple applications often run simultaneously [12], [13]. In disaster relief scenarios, for example, mobile nodes play critical roles in applications, such as disaster-site data collection, emergency task offloading, and regional strategy distribution, as shown in Fig. 1. Each application has distinct routing and forwarding requirements: data collection involves content retrieval routing, task offloading requires device-to-device unicast routing for efficient task processing, and strategy distribution depends on multicast routing to target specific geographic areas. Managing these diverse demands poses significant challenges for MANETs, leading to two major issues.

Received 31 October 2024; revised 30 December 2024 and 12 January 2025; accepted 28 January 2025. Date of publication 4 February 2025; date of current version 23 May 2025. This work was supported in part by the Talent Fund of Beijing Jiaotong University under Grant 2024XKRC063; in part by the National Natural Science Foundation of China under Grant 62394324 and Grant 61972026; and in part by the Beijing Natural Science Foundation under Grant 4242008. *(Corresponding author: Shuai Gao.)*

Jiangyu Lan, Shuai Gao, Weiting Zhang, Xindi Hou, Minghui Xi, Yuming Zhang, and Hongke Zhang are with the School of Electronic and Information Engineering, Beijing Jiaotong University, Beijing 100044, China (e-mail: lanjiangyu@bjtu.edu.cn; shgao@bjtu.edu.cn; wtzhang@bjtu.edu.cn; 19111019@bjtu.edu.cn; 21120147@bjtu.edu.cn; ymzhang3@bjtu.edu.cn; hkzhang@bjtu.edu.cn).

Bo Lei is with the China Telecom Research Institute, Beijing 102209, China (e-mail: leibo@chinatelecom.cn).

Xuemin Shen is with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada (e-mail: sshen@uwaterloo.ca).

Digital Object Identifier 10.1109/JIOT.2025.3538167

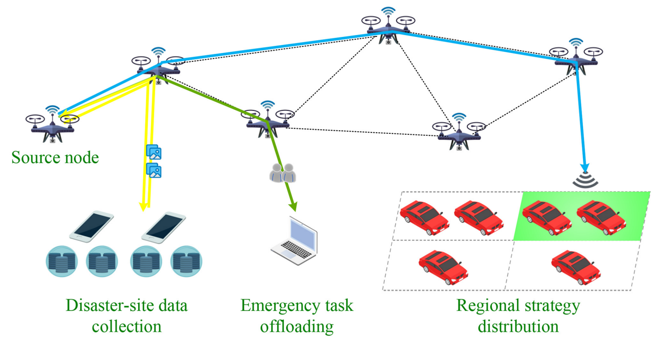



Fig. 1. Multiple applications in disaster relief scenarios.

1) *Integrated Network Identifier Design:* Current network identifiers are designed for specific applications but lack an integrated design for diverse network entities, which encompass any identifiable component within the network, including physical devices, nonphysical content, and locations. Creating identifiers that can describe any combination of these entities while remaining flexible and unique is challenging. Additionally, an efficient registration and query service for these entities is crucial for effective communication. However, this is complicated by the diversity of network entities and the mobility inherent in MANETs.
2) *Diverse Network Service Coexistence:* Achieving interoperability among multiple network-layer (L3) protocols for diverse routing and forwarding services coexistence is challenging. Traditional overlay solutions result in high-deployment costs and limited scalability [14], [15], [16], [17]. Recently, P4-based SDN [18], [19] has emerged as a solution, but it is primarily designed for wired networks with multiple interfaces and robust processing capabilities. In contrast, MANET nodes are resource-constrained and lack various interfaces, making it challenging to apply P4-based SDN directly.

In this article, we propose OpenL3, a programmable L3 approach to address the above challenges. OpenL3 abstracts key attributes from network entities and embeds these attributes into an integrated network identifier, termed multidimensional identifier (MID), to control the forwarding and routing processes. Then, a distributed MID mapping system is established to provide efficient MID registration and query services. Based on the MID, we propose a programmable routing and forwarding scheme. This scheme incorporates a lightweight packet processing design using a P4 programmable data plane to enable interoperability among OpenL3 protocols. Additionally, multiple control-plane devices collaboratively distribute flow rules within a cluster to manage data plane behavior. Specifically, the main contributions of this article are summarized as follows.

1) To ensure uniqueness and flexibility, we designed selection criteria of key attributes and established a unified format for the MID to accommodate various applications. Additionally, we propose an MID generation scheme to illustrate the process of converting entities into MID.
2) To enable efficient MID registration and query services, a distributed MID mapping system is established. Then, a mapping information synchronization scheme using P4-based SDN is proposed to ensure data consistency with low overhead.
3) We propose a programmable routing and forwarding scheme. Specifically, we introduce a MID forwarding scheme to support various forwarding services, implemented via a lightweight P4 pipeline. Additionally, a MID routing scheme is proposed to manage the data plane for diverse routing services.
4) A prototype system is implemented to evaluate OpenL3. Experimental results demonstrate that OpenL3 outperforms the existing solutions in both end-to-end latency and network throughput. Moreover, OpenL3 can be deployed in current MANETs, requiring no modifications to existing network protocols or sockets.

The remainder of this article is organized as follows. Related works are reviewed in Section II. Section III proposes the architecture of OpenL3. Then, we provide the design details of OpenL3 in Section IV. Section V designs the OpenL3 prototype system and evaluates its performance. Finally, we present the conclusions and future works in Section VI.

## II. Related Work

Mobile devices can interconnect to form a reliable network known as MANETs, which collectively enhance coverage, provide flexibility in deployment, and ensure robustness in communication [20]. MANETs are categorized by device type into flying ad-hoc networks (FANETs) [21], vehicle ad-hoc networks (VANETs) [22], etc. Various emerging applications make traditional host-centric routing and forwarding schemes inefficient for their communication needs [23]. Consequently, customized routing and forwarding methods have been proposed for these applications.

This section provides a brief overview of the solutions for diverse routing and forwarding solutions in MANETs, including host-centric, nonhost-centric, and integrated solutions. Our objective is to present a comprehensive understanding of the existing literature on this topic.

### A. Host-Centric Solutions

Host-centric solutions associate service information with host addresses, storing it on a server for query-based retrieval, similar to DNS.

Hormati et al. [24] proposed a disaster response system where MANET devices upload service information to a server for service discovery. This method employs the DNS-based service discovery method to represent service information as DNS records and employs a cluster-based approach to form service directories dynamically. However, it does not consider the corresponding routing and forwarding scheme of service packets in MANETs.

Fioreze and Heijenk [25] introduced a geographic addressing method for VANETs, querying a server for the IPv6 addresses of all vehicles in the target area and unicast sending IPv6 data packets via RSUs. However, this method relies on server queries, making it unsuitable for environments lacking network infrastructure. Additionally, the DNS-like querying approach often introduces significant delays.

### B. Nonhost-Centric Solutions

Nonhost-centric solutions propose customized network identifiers with corresponding routing and forwarding mechanisms. Researchers aim to design entirely new solutions, overcoming the constraints of traditional approaches [26]. These solutions are classified into content-centric, location-centric, and other categories.

Content-centric solutions use content names as network identifiers, with information-centric networking (ICN) [27] being a notable example. ICN enhances communication by naming content, which leads to more efficient data transmission and distribution. Named data networking (NDN) [28] is a representative ICN architecture, which maintains three tables, such as content store (CS), pending interest table (PIT), and forwarding information base (FIB), supporting data sharing and collaboration in edge computing environments.

Location-centric solutions designate geographical regions as network identifiers [29]. Hou et al. [8] constructed network identifiers with geographic semantics, enabling multicast communication for all devices within a geographic area at L3 based on the Earth Mover's Distance model and IPv6 address format.

In addition, researchers have proposed routing and forwarding solutions for special applications. Cheng et al. [30] proposed a tactic-centric approach to address information-sharing requirements in military MANETs, creating routing and forwarding mechanisms tailored to specific tasks. This approach limits most collaboration and network traffic to groups of users with shared interests, resulting in minimal communication overhead when performing specific tasks.

While these studies provide network service solutions for specific applications in MANETs, they do not address the need for diverse network services concurrently.

### C. Integrated Solutions

In practical applications, conducting diverse network services simultaneously poses significant challenges due to the intrinsic contradictions in the logic of different routing and forwarding services. For instance, host-centric routing and forwarding services are push-based, whereas content-centric is pull-based [5]. Therefore, integrated solutions are essential to harmonize these varying routing and forwarding logics. Researchers have primarily focused on two main lines of research to address this issue.

*1) Overlay Approach:* Traditional overlay solutions deploy one network protocol stack over another to achieve compatibility. For instance, Luo et al. [15] proposed an NDN-IP compatibility strategy that captures the IP datagrams using a TUN device and employs a translator for bidirectional conversion between IP datagrams and NDN packets. Shannigrahi et al. [16] introduced the tunneling protocol IPoC, which forwards IP traffic from terminal devices by encapsulating it into NDN interest messages for transmission into the NDN network, effectively implementing IP-over-ICN. Thomas et al. [17] advocated for an IP-ICN hybrid network architecture in MANETs, deploying TCP/IP and NDN protocol stacks separately on terminal devices to mitigate issues arising from the exclusive use of IP or ICN.

While these overlay approaches enhance functionality by integrating different network protocols, they face challenges, such as complex operations, high-deployment costs, and compatibility issues due to conflicting protocols, which can degrade overall network performance.

*2) P4-Based SDN:* Recent advancements in P4-based SDN technology enable flexible deployment of diverse network services through programmable operations in control and data planes. For example, Guo et al. [31] demonstrated the coexistence of NDN and TCP/IP on programmable switches, proposing an NDN routing mechanism based on this integration. Similarly, Hou et al. [8] proposed a location-centric network solution for edge devices using P4-based SDN to create a flexible data plane for customized packet processing.

Despite showcasing the capabilities of P4-based SDN, these studies highlight that the technology is primarily suited for wired networks due to its high-computational demands and multiple interfaces. In contrast, MANET nodes generally possess limited computing power and a single network interface, necessitating designs specifically tailored to their constraints.

## III. OpenL3 Architecture

### A. Requirements Analysis

In this section, we outline the essential requirements for embedding diverse network services into MANETs as follows.

1) *Integrated Network Identifier:* An integrated network identifier is crucial for managing routing and forwarding processes in MANETs. This identifier is required to represent various network entities, including nodes, content, and locations, individually or in combination.
2) *Security Support:* To ensure security, MANET nodes should be capable of authenticating the identities of their counterparts. This prevents impersonation attacks to guarantee the legitimacy of communication participants.
3) *Efficient Identifier Discovery:* To efficiently map diverse network entities to their corresponding identifiers, a design of a robust registration and query framework for identifier management is necessary.
4) *Network Services Coexistence:* A scalable and flexible framework is needed to overcome the conflicting logic of diverse routing and forwarding services and accommodate future additions.
5) *Easy to Implementation:* MANET nodes exhibit significant heterogeneity in hardware capabilities. Customizing the network protocol stack for each node is impractical, making it essential to develop a general-purpose implementation scheme.
6) *Compatibility Mechanism:* MANET nodes should be capable of communicating with Wide Area Networks (WANs). A compatibility mechanism must be devised to enable seamless transmission across WANs.

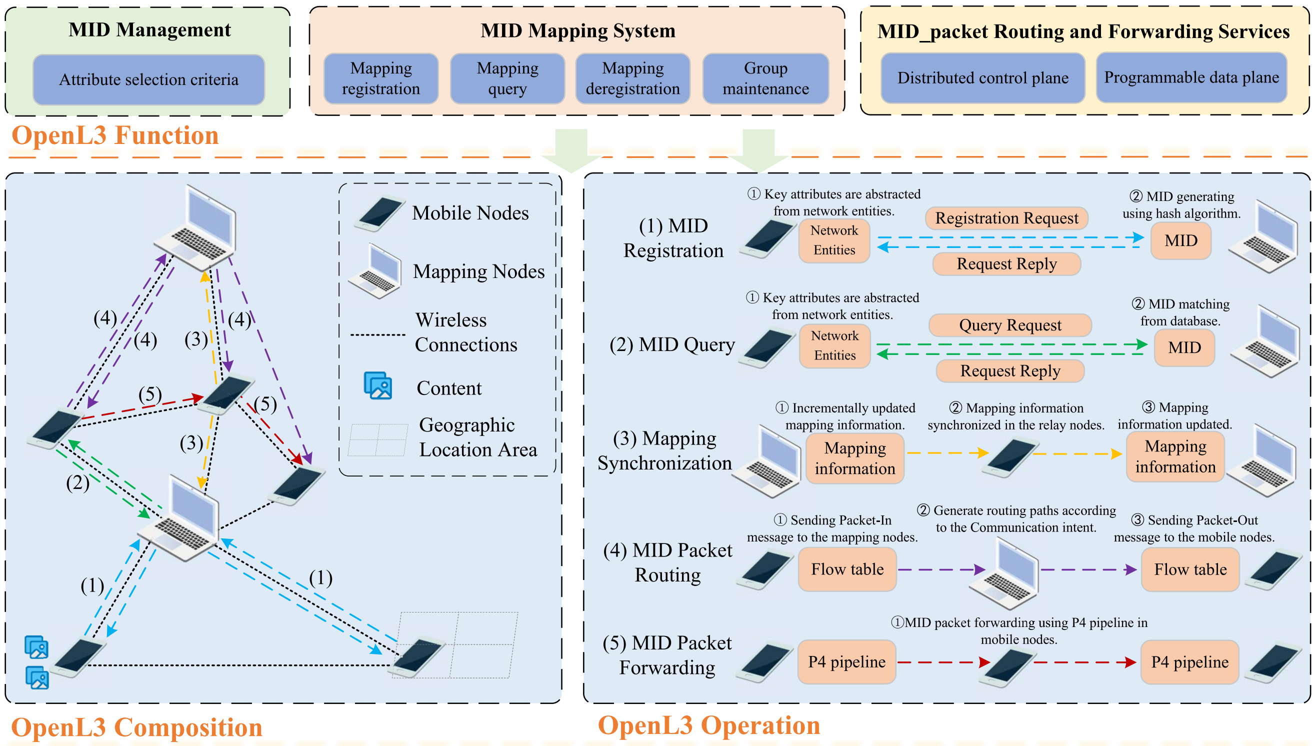



Fig. 2. Architectural overview of OpenL3.

### *B. Architectural Overview of OpenL3*

Based on the requirements analysis, we propose the architectural overview of OpenL3, as illustrated in Fig. 2.

*1) OpenL3 Function:* OpenL3 embeds diverse network services into MANETs through three core functions: 1) MID management; 2) MID mapping system; and 3) MID_packet routing and forwarding services. MID Management defines the criteria for selecting key attributes of network entities to generate unique identifiers. MID Mapping System facilitates efficient MID registration and query services by mapping network entities to their respective MIDs. MID_packet Routing and Forwarding Services enable the coexistence of diverse network services by treating each service as an application and dynamically selecting the appropriate service for MID_packets.

*2) OpenL3 Composition:* In MANETs, mobile nodes form a dynamic wireless network and host various network entities, such as content and locations. OpenL3 selects several high-performance nodes as mapping nodes based on factors like relative location, computing power, and residual energy. These mapping nodes perform mapping operations while handling data packet forwarding, similar to regular nodes.

*3) OpenL3 Operation:* OpenL3 embeds diverse network services into MANETs via the following operational steps.

1) *MID Registration:* When a new network entity is generated on a node, the node extracts key attributes from the entity and sends this information to the mapping node. The mapping node then generates a MID using a hash algorithm and returns it to the originating node.
2) *MID Query:* During communication, nodes query the mapping system by sending a request with one or more attributes. The mapping node resolves the query and returns the corresponding MID.
3) *Mapping Synchronization:* To ensure the consistency of MID mapping information, mapping nodes periodically synchronize updates with other mapping nodes.
4) *MID_packet routing:* When encountering a MID_packet without a matching flow entry, nodes send a Packet-In message to the control plane. Then, the control plane generates an appropriate routing path and sends it to nodes to update the flow table.
5) *MID_packet Forwarding:* OpenL3 dynamically selects the appropriate forwarding service for each MID_Packet at each hop using the programmable data plane.

### *C. Main Components of OpenL3*

To achieve the OpenL3 architecture design outlined above, we provide a brief overview of its main components. Detailed design specifics are elaborated in Section IV.

1) *MID Design and Generation Scheme:* OpenL3 defines the selection criteria for key attributes of network entities and establishes a unified identifier format for the MID. This scheme is utilized to convert any combination of network entities into an MID.
2) *MID Mapping System:* OpenL3 uses the MID mapping system to resolve and store the mapping relationship between network entities and MIDs. The mapping system includes registration, query, deregistration, and group maintenance operations, supported by a mapping and storage module.

3) *Distributed Control Plane:* A distributed control plane is proposed to integrate multiple routing protocols. It enables seamless switching among diverse routing services, optimal route computation for MID_packets, and facilitates flow table distribution to the data plane.
4) *Lightweight Data Plane:* The lightweight data plane is established using P4, enables flexible forwarding of MID_packets in MANET nodes. These nodes function as programmable switches, parsing MID_packet and completing the forwarding process.

## IV. OpenL3 Designs

### *A. MID Design*

Existing network identifiers are typically designed for specific applications, such as IP addresses for host-centric applications and NDN addresses for content-centric applications. However, there is a lack of an integrated design to support diverse applications. Therefore, a novel network identifier is required to describe diverse network entities, such as nodes, contents, and locations. In this section, we outline the MID identifier design, generation scheme, and security analysis

*1) MID Identifier Design:* Network identities can be categorized as hierarchical or flat [32]. Hierarchical identifiers, such as content names in NDN, comprise multiple semantic components. They are human-readable and support aggregation but are susceptible to forgery and tend to have variable lengths, which can grow significantly when describing diverse services. In contrast, flat identifiers, such as IP addresses, consist of nonsemantic strings. These identifiers are not human-readable but offer stronger security and typically have a fixed length, such as the 128-bit structure of IPv6.

The design of MID is required to satisfy several requirements: describing diverse network entities, ensuring security, and maintaining IP compatibility. To address these needs and consider the "Zooko's triangle problem" [33], MID adopts a flat identifier design, prioritizing decentralization and security over human readability. Specifically, MID is designed as a 128-bit flat identifier within an IPv6-like namespace. This structure prevents forgery through nonsemantic encoding while supporting diverse network applications without the overhead of excessively long identifiers.

To distinguish between network entities, MID is divided into two types: 1) device unified identifier (DUID) for device identities and 2) service unified identifier (SUID) for services, such as content, location, and groups. Both DUID and SUID are generated using the hash algorithm from multidimensional attribute description (MAD), a semantically meaningful vector array generated with key attributes of network entities.

Specifically, DUID is used for host-centric services, its $\text{MAD}_D$ includes attributes, such as manufacturer, device_type, device_version, and device_owner. SUID is designed to describe diverse virtual network entities in MANETs, which are listed as follows.

1) *Content:* $\text{SUID}_C$ is used for content retrieval applications (e.g., videos, images). $\text{MAD}_C$ attributes include publisher, content_name, content_type, and content_version.

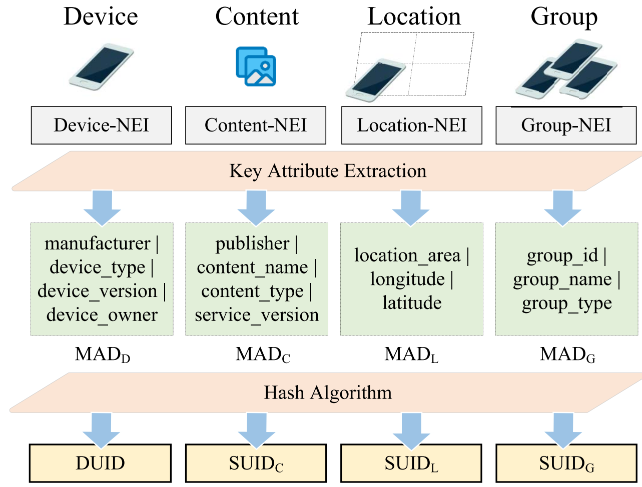



Fig. 3. MID generation scheme.

2) *Location:* $\text{SUID}_L$ is used for communication within specific geographic areas. $\text{MAD}_L$ attributes include location_area, longitude, and latitude.
3) *Group:* $\text{SUID}_G$ is used for group-based communication involving multiple entities (e.g., a node group or shared content). $\text{MAD}_G$ attributes includes group_id, group_name, and group_type.

Note that, MID enables flexible descriptions of network entities, supporting composite queries for diverse applications. For instance, to communicate with devices meeting specific criteria, such as ["Location: Area X," "Device Type: UAV," "Content: Surveillance Video"], MID allows interaction with all devices matching these attributes.

*2) MID Generation Scheme:* To efficiently generate MID, we developed a structured three-stage process involving network entity information (NEI), MAD, and the final MID, which is shown in Fig. 3. NEI represents raw information about diverse network entities in MANETs. Key attributes are extracted from NEI to form MAD, which is then processed using a hash algorithm to generate the MID.

The MID generation process is designed for both single-entity and multientity applications, following separate workflows as detailed below. To help explain, the complete workflow is illustrated in Algorithm 1.

*Step 1—Initialization (Lines 1 and 2):* At the beginning of the scheme, we format the MNEI and extract key attributes. For a set of selected network entities, we construct a vector MNEI

$$\begin{aligned} \text{MNEI} &= [\text{NEI}_1, \ldots, \text{NEI}_i, \ldots, \text{NEI}_k] \\ \text{s.t.} \quad & i \in N, 1 \leq i \leq k \end{aligned} \tag{1}$$

where $\text{NEI}_k$ represents the $k$th network entity. For each $NEI$, key attributes are identified to form MAD. The workflows for single-service ($k = 1$) and multiservice ($k > 1$) are described below.

*Step 2.1—Single-Service Workflow (Lines 3–10):*

1) *Attribute Extraction:* Key fields are extracted from the NEI, represented as $AD_i$, i.e.,

**Algorithm 1** MID Generation Scheme

```
Input:
    MNEI and n;
Output:
    MID;
1: Construct MNEI.
2: Determine the number k of NEI in MNEI.
3: if k = 1 then
4:    //Single-service.
5:    MAD=[];
6:    for i in n do
7:       AD_i ← extract key attributes of NEI;
8:       Append AD_i to MAD;
9:    end for
10:   MID ← HASH(MAD);
11: else
12:   //Multi-service.
13:   MAD=[];
14:   for j in k do
15:      MAD_g=[];
16:      for i in n do
17:         AD_i ← extract key attributes of MNEI(j);
18:         Append AD_i to MAD_j;
19:      end for
20:      Append MAD_j to MAD_g;
21:   end for
22:   MID ← HASH(MAD_g);
23: end if
24: return MID
```

$$AD_i = \text{NEI}[s_i : e_i] \quad \text{s.t.} \quad i \in N, 1 \le i \le n \tag{2}$$

where $s_i$ and $e_i$ denote the start and end indices of the $i$th substring.

2) *MAD Formation:* These fields are combined to form the MAD vector

$$\text{MAD} = [AD_1, \ldots, AD_i, \ldots, AD_n] \quad \text{s.t.} \quad i \in N, 1 \le i \le n \tag{3}$$

where $AD_i$ represents the $i$th attribute in MAD.

3) *MID Generation:* The MID is generated by applying a hash function (e.g., MD5 or SHA-2) to MAD, i.e.,

$$\text{MID} = F(\text{MAD}). \tag{4}$$

*Step 2.2—Multiservice Workflow (Lines 11–23):*

1) *MAD Matrix Construction:* For each NEI, key attributes are extracted and organized into a $k \times n$ matrix, i.e.,

$$\begin{cases} \text{MAD}_1 = [AD_{11}, \ldots, AD_{1i}, \ldots, AD_{1n}] \\ \text{MAD}_j = \left[AD_{j1}, \ldots, AD_{ji}, \ldots, AD_{jn}\right] \\ \text{MAD}_k = [AD_{k1}, \ldots, AD_{ki}, \ldots, AD_{kn}] \end{cases} \quad \text{s.t.} \quad i, j \in N, 1 \le i \le n, 1 \le j \le k. \tag{5}$$

2) *MAD Aggregation:* The individual MAD vectors are combined into a single $\text{MAD}_g$ matrix, i.e.,

$$\text{MAD}_g = \left[\text{MAD}_1, \ldots, \text{MAD}_j, \ldots, \text{MAD}_k\right]. \tag{6}$$

3) *MID Generation:* The aggregated $\text{MAD}_g$ is processed using the hash function to produce the final MID

$$\text{MID} = F\left(\text{MAD}_g\right). \tag{7}$$

By leveraging the process of extracting key attributes, forming MAD, and applying a hash function, this scheme ensures the scalability and efficiency of OpenL3.

*3) MID Security Analysis:* In MANETs, the lack of centralized authentication authority exposes the network to impersonation attacks, where attackers can forge the node identities to intercept or modify data packets [34]. Due to its uniqueness and decentralization, MID supports self-authentication using identity-based encryption (IBE) [35], enabling mutual authentication via digital signatures and challenge-response protocol without third-party authorities.

In the authentication process between node $A$ and node $B$, node $A$ has a public key $PK_A$, a private key $SK_A$, system public parameters Params, and a public key calculation function Fun(). Node $B$ initiates by sending a random challenge $n$ to node $A$. Node $A$ then encrypts $n$ with $SK_A$ and replies with the encrypted message $E_{SK_A}(n)$. Upon receiving the response, node $B$ reconstructs $A$'s public key from its identifier $\text{DUID}_A$

$$PK_A = \text{Fun}(\text{Params}, \text{DUID}_A) \tag{8}$$

and decrypts the received message using $PK_A$ to obtain challenge $n$

$$n = D_{PK_A}\left(E_{SK_A}(n)\right). \tag{9}$$

If the decrypted value matches the original challenge, the identity of node $A$ is successfully verified. This approach can also enable authentication between virtual network entities and nodes using SUID for the security of diverse network services.

## B. MID Mapping System

To enable OpenL3 in MANETs, establishing efficient mappings between network entities and MIDs is crucial. This ensures that nodes quickly discover and use these mappings. While previous studies have proposed service discovery methods to facilitate information exchange about available services [36], they primarily focus on architectural design and overlook implementation details from a network perspective.

In this section, we introduce the MID Mapping System to handle the registration and querying of mappings between network entities and MIDs. Additionally, we propose a low-cost synchronization scheme for mapping information using P4 to minimize communication overhead.

*1) Mapping System Architecture:* In OpenL3, we select $k$ mobile nodes with strong computational performance and sufficient residual energy as mapping nodes. The selection method has been widely discussed in previous studies [37], [38], [39]. Mapping nodes handle the registration and management of network entity mappings. Each mapping node hosts a lightweight database to store MAD-MID mappings, ensuring efficient lookups and updates. To maintain consistency, the $k$ mapping nodes periodically synchronize their databases.

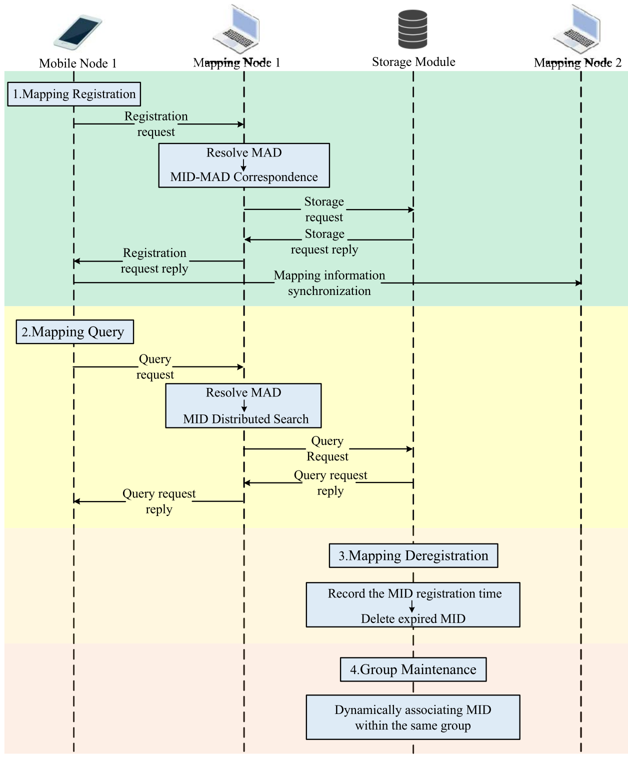



Fig. 4. Workflow of MID mapping system.

*2) Mapping System Operations:* The MID mapping system supports four core operations: 1) mapping registration; 2) mapping query; 3) mapping deregistration; and 4) group maintenance. To enable these functionalities, the system consists of two primary modules: 1) the mapping module, which handles the mapping between MAD and MID and 2) the storage module, which stores MID-MAD pairs. Fig. 4 illustrates the system workflow.

1) *Mapping Registration:* When a new network entity (e.g., a video) is created, the originating node initiates a registration process with the mapping nodes. The node generates a MAD from the NEI using the MID generation algorithm and caches the MAD locally. It then sends a registration request containing the MAD to the mapping nodes, which generate the SUID and send a confirmation reply.
2) *Mapping Query:* If a node needs to communicate but lacks the destination SUID, it sends a query to the mapping nodes. Using the MAD in the query, the mapping nodes retrieve the corresponding SUID from their database and return it to the requesting node. This enables the node to obtain the required service using the SUID.
3) *Mapping Deregistration:* To conserve resources, nodes must remove outdated or unnecessary mapping data. When a node is decommissioned or service information becomes obsolete, the mapping system performs deregistration. The expiration of mapped data depends on factors, such as service type and application scenario.
4) *Group Maintenance:* The system supports multicast services by maintaining group mappings. For example, nodes can multicast to device groups (e.g., smartphones) or content groups (e.g., surveillance videos). These groups can be predefined or dynamically registered during tasks. Once established, the storage module associates MIDs of similar categories using MAD information.

*3) Mapping Synchronization Scheme:* We employ a distributed mapping node architecture to significantly reduce communication overhead compared to a fully distributed architecture [37]. However, this method still requires periodic broadcast data synchronization among multiple mapping nodes, adding to the communication overhead.

To address this challenge, we propose a mapping information synchronization strategy using the programmable data plane. This strategy allows the addition and modification of packet headers to control the forwarding process. When synchronization packets (sync_packet) from mapping nodes are forwarded, the relay nodes use the P4 pipeline to forward sync_packets that meet the specified requirements. This approach aims to reduce redundant packets in the network without modifying the protocol stack.

*Step 1—Distance-Based Filtering:* Define $d$ as the distance between the current node and its nearest mapping node, and $d_{\text{prev}}$ as the distance between the previous hop node and its nearest mapping node. When the relay nodes forward the sync_packet, if $d > d_{\text{prev}}$, they drop the sync_packet. The distance $d$ can be calculated using the free space propagation model approach as

$$d = \frac{\lambda}{4\pi} \cdot \sqrt{\frac{P_t}{P_r}} \tag{10}$$

where $P_r$ is the received power, $P_t$ is the transmitted power, $\lambda$ is the wavelength of the signal. To obtain the necessary information for this calculation, nodes can leverage existing physical layer protocols to measure received power $P_r$ through signal strength indicators and can derive transmitted power $P_t$ based on their transmission settings.

*Step 2—Timestamp-Based Filtering:* Each sync_packet is assigned a creation timestamp $T_{\text{msg}}$ at the mapping node. Relay nodes maintain a timestamp list $L$

$$\begin{aligned} L = (AD_1, T_1), (AD_i, T_i), \ldots, (AD_n, T_n) \\ \text{s.t.} \quad i \in N, 1 \le i \le n \end{aligned} \tag{11}$$

where $AD_i$ is the source address and $T_i$ is the creation time of the processed sync_packet. When a sync_packet arrives at the relay nodes, they extract $T_{\text{msg}}$ from the sync_packet. If $(AD_{\text{msg}}, T_{\text{msg}}) \in L$ and $T_i \ge T_{\text{msg}}$, the sync_packet is considered an expired version and is dropped.

The complete mapping synchronization scheme is detailed in Algorithm 2.

### C. MID_packet Routing and Forwarding Scheme

Routing and forwarding schemes often exhibit distinct and conflicting logic. For instance, location-centric routing typically involves multicasting to nodes within a specific area, whereas content-centric routing relies on a request-reply

**Algorithm 2** Information Synchronization

```
Extract T_msg, and d_prev from sync_packet;
Calculate d using Eq.10;
if d > d_prev then
    Drop the sync_packet
else
    if (AD_msg, T_msg) ∈ L and T_i ≥ T_msg then
        Drop the sync_packet
    else
        Update (AD_msg, T_msg) in L
        Update t_u in sync_packet to current time t_current
        Update d_prev in sync_packet to d
        Forward the sync_packet
    end if
end if
```

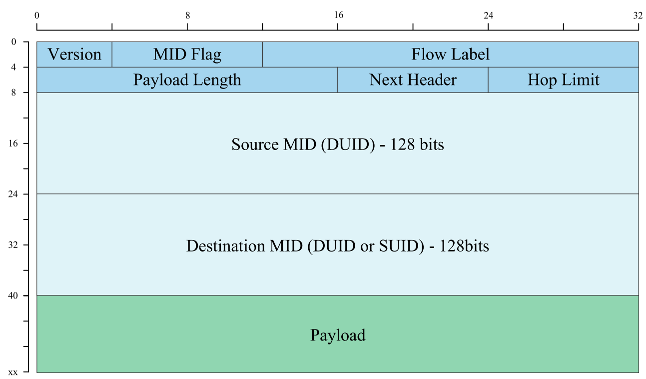



Fig. 5. MID_packet format.

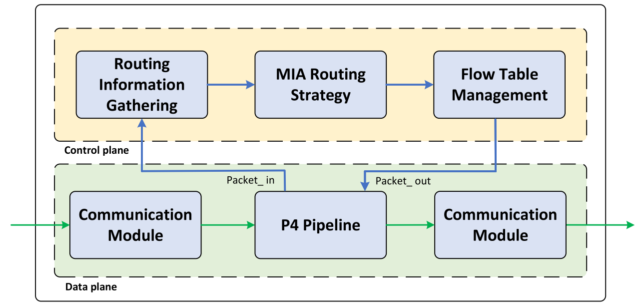



Fig. 6. Distributed control plane design.

mechanism for content retrieval. These differences make integration challenging. Existing overlay-based approaches [14], [15], [16], [17] face deployment difficulties and scalability limitations.

In this section, we propose the MID_packet routing and forwarding scheme using P4-based SDN for these challenges.

*1) MID_Packet Design:* To enhance the forwarding capabilities, critical information is encapsulated in the packet header, enabling flexible behaviors during packet forwarding. For compatibility with existing protocol stacks and to simplify deployment, MID_packets are designed based on the IPv6 format, as shown in Fig. 5. Standard IPv6 fields, such as *Version*, *Flow Label*, *Payload Length*, *Next Header*, and *Hop Limit*, are retained. Source and destination addresses also comply with IPv6 standards, with the *Source MID* representing the originating DUID and the *Destination MID* specifying the target DUID or SUID.

The *IPv6 Traffic Class* is replaced with the 8-bit *MID_Flag*, which indicates the required routing and forwarding service, such as host-centric, content-centric, or location-centric, including unicast and multicast.

When initiating communication, mobile nodes encapsulate data into MID_packets. During packet forwarding, relay nodes extract the routing and forwarding requirements from the MID_packet header, selecting the appropriate behavior.

Note that, for content-centric services, a request-reply mechanism is adopted: the source node sends an interest packet, and the target node responds with a data packet. To improve forwarding efficiency, we implement a PIT table inspired by NDN. The PIT tracks outstanding interests, facilitating efficient routing decisions. Therefore, differentiating between interest and data packets is critical in the MID_packet design.

*2) MID_Packet Routing Scheme:* Existing routing methods in MANETs are typically designed for specific routing service types, such as OLSR [40] for host-centric and NLSR [41] for content-centric. These protocols collect particular information, such as host IDs or content names, to construct routing paths. However, their single-service focus limits adaptability to diverse application requirements. Additionally, deploying multiple routing protocols within the same network introduces significant complexity, especially when managing interactions and transitions across different protocol stacks.

The software-defined Internet of Things (SD-IoT) paradigm offers a promising solution for managing diverse network services through centralized, software-driven control planes. A QoS-aware SD-IoT method described in [42] improves service traffic management and balances server loads. Similarly, an approach proposed in [43] leverages SDN to dynamically allocate virtual resources and optimize network loads based on multimedia content priorities. These approaches demonstrate how SDN supports diverse services in wireless networks.

Inspired by SD-IoT, we design an SDN-based distributed control plane to manage existing routing protocols in MANETs. In this architecture, the mapping node functions as the distributed SDN controller, collecting network topology information and establishing host-centric, content-centric, and location-centric topologies based on service registration data. For each application, the corresponding routing algorithm is selected to generate flow table entries, which are then distributed to the data plane. This design ensures compatibility with existing hardware and protocol stacks, minimizing deployment complexity while maintaining scalability. The distributed control plane design is shown in Fig. 6, and the complete process of the MID_packet routing scheme is outlined in Algorithm 3.

*Route Information Gathering:* Nodes periodically transmit hello messages to gather network information, including details about neighboring nodes and link status. This data is used to construct and update the network topology, represented as $G = (V, E)$, where $V$ denotes the set of nodes and $E$

**Algorithm 3** MID_Packet Routing Scheme

**Input:**
$G = (V, E)$, $L$, $C$;
$DUID_{src}$, $SUID_{dst}$, $DUID_{dst}$;
**Output:**
Flow table $F$;
Determine routing type $T_r$ from MID_Flag
Find $N_{src}$ from $DUID_{src}$
**if** $T = Host$ **then**
  Find $N_{dst}$ from $DUID_{dst}$
**else if** $T = Content$ **then**
  Find $N_{dst}$ from content $C$ and $SUID_{dst}$
**else if** $T = Location$ **then**
  Find $N_{dst}$ from location $L$ and $SUID_{dst}$
**end if**
Determine the number $N$ of $N_{dst}$
**if** $N_{dst} == 1$ **then**
  $\mathcal{R} \leftarrow Routing(N_{src}, N_{dst}, G)$ using Dijkstra
**else**
  $\mathcal{R} \leftarrow Routing(N_{src}, N_{dst}, G)$ using SPT
**end if**
Initialize empty flow table $F$
**for** each path $R$ in $\mathcal{R}$ **do**
  **for** each hop $i$ in $R$ **do**
    $F_i \leftarrow$ (match: $P$, action: forward to next hop)
  **end for**
**end for**
**return** Flow table $F$

represents the links between them. To support content-centric and location-centric communication, the collected location information ($L$) and content name information ($C$) from each node are linked to the host address, resulting in a unified network topology that integrates content and location attributes.

*MID_packet Routing Strategy:* The routing strategy leverages flexible control policies and rules to designate paths for data packets, allowing seamless adaptation to varying application requirements. The proposed strategy integrates existing routing algorithms and supports dynamic switching among them, as described in Algorithm 4. The routing type $T_r$ is determined by the MID_Flag. Then, using the topology $G$, the corresponding routing algorithm generates routing paths, denoted as $\mathcal{R}$.

*Flow Table Management:* A corresponding flow table $F$ is generated, dynamically scheduled, and deployed to the data plane by the distributed control plane, facilitating the matching, forwarding, and control of MID_packets.

*3) MID_Packet Forwarding Scheme:* We design a data plane processing logic for MID_packet forwarding using P4-based SDN to support the coexistence of diverse forwarding logic. P4 allows flexible programmable data processing through match-action operations in flow tables [44]. This design addresses the challenges of supporting diverse forwarding logic and implementing P4-based SDN on MANET nodes.

The MID_packet forwarding scheme, outlined in Algorithm 4, starts by examining the MID_Flag field to select

**Algorithm 4** MID Forwarding Scheme

Determine routing type $T_f$ from MID_Flag
**if** the packet is not MID_Flag **then**
  drop the MID_Packet;
  **return**
**end if**
**if** $T_f$ == "content.interest" **then**
  **if** PIT entry exists MID_Packet **then**
    drop the MID_Packet;
    **return**
  **end if**
**else if** $T_f$ == "content.data" **then**
  **if** PIT entry exists MID_Packet **then**
    $P_{out} \leftarrow F.port$;
    Copy MID_Packet to $P_{out}$
  **end if**
**end if**
**if** $T_f$ == "unicast" **then**
  $P_{out} \leftarrow F.port$;
  Unicast the MID_Packet to $P_{out}$;
**else if** $T_f$ == "multicast" **then**
  $P_{group} \leftarrow F.group$;
  Multicast the MID_Packet to $P_{group}$;
**end if**
**if** forwardToWAN(MID_Packet) **then**
  Encapsulated header onto MID_Packet;
**end if**

TABLE I
EXAMPLE MATCH-ACTION TABLE OF OPENL3

| Table name | Key | Action |
|---|---|---|
| MID_flag | hdr.mid.flag | get_value() |
| | Other values | drop() |
| MID_fwd | hdr.mid.fwd: 0001 | set_egress_port() |
| | hdr.mid.fwd: 0010 | P4_broadcast() |
| | hdr.mid.fwd: 0100 | copy_to_port() |
| | hdr.mid.fwd: 1000 | drop() |
| Enable_ip | hdr.ip.fwd | hdr.ip.SetValid() |
| | Other values | NoAction() |

the appropriate forwarding service. If the MID_Flag does not comply with the expected format, the packet is discarded.

If no matching flow entry exists, a Packet-In message is sent to the control plane to fetch the required flow entry. Once the flow entry is matched, the forwarding type $T_f$ is determined. For content-centric interest packets, the PIT is checked; if an entry exists, the packet is discarded. For data packets, if a corresponding interest packet is found, the MID_packet is forwarded to the relevant output ports $P_{out}$. For unicast packets, the forwarding port is selected, and for multicast, forwarding is based on the multicast group.

Additionally, we propose a compatibility solution for communication between OpenL3 and TCP/IP: if the MID_packet needs to be transmitted to a WAN, an IPv6 header is encapsulated onto the MID_packet.

We design three match-action tables to implement the MID_packet forwarding scheme, as shown in Table I. The

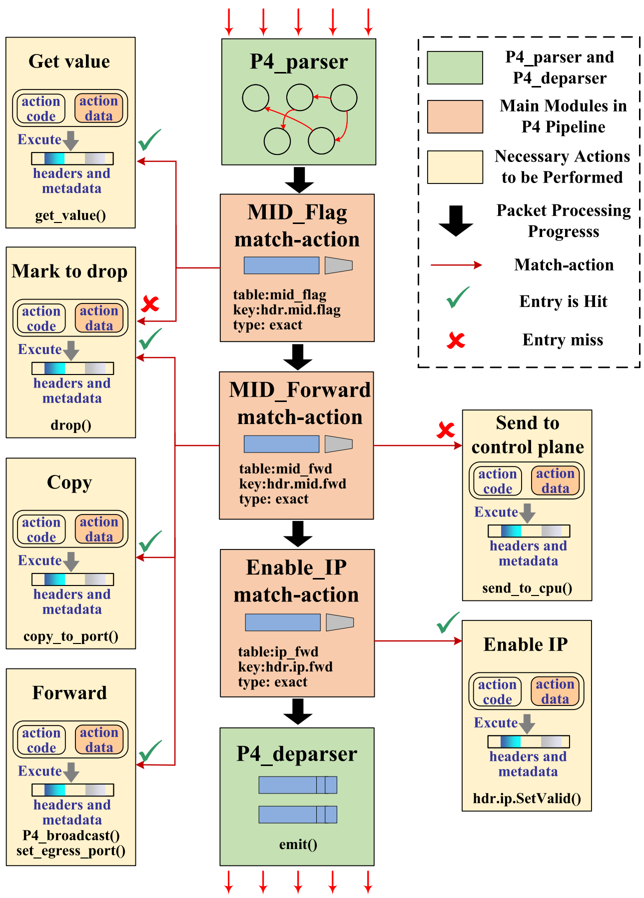



Fig. 7. P4 pipeline design of OpenL3.

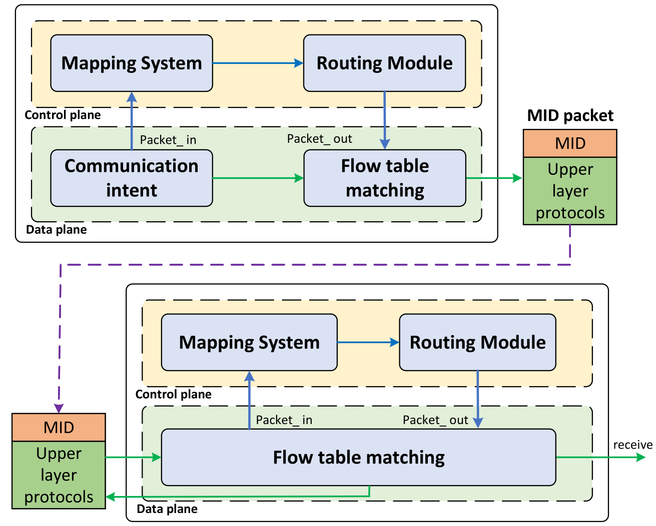



Fig. 8. Workflow of OpenL3.

MID_flag gets the MID Flag values from the MID header. According to the MID_fwd, packets can be discarded, copied to specified ports, or forwarded. The Enable_IP table encapsulate MID_packets with an IP header. We designed a P4 pipeline for the OpenL3 data plane to facilitate MID_packets parsing and reconstruction, illustrated in Fig. 7. The primary components in the P4 pipeline are as follows.

*P4_parser:* This module extracts necessary fields and metadata from the incoming MID_packet.

*MID_flag Match-Action Module:* This module determine the forwarding service of the MID_packet. It matches the MID_Flag from the packet with hdr.mid.flag. If no match is found, the drop() action is triggered.

*MID_Forward Match-Action Module:* This module handles the forwarding of MID_packets within MANETs. It matches against hdr.mid.fwd and supports three actions.

1) *Drop Packet:* In content-centric forwarding, if the PIT contains the interest packet, we use drop() to prevent duplicate transmissions.
2) *Copy Packet:* In content-centric forwarding, if the PIT contains the interest packet, copy_to_port() copies the packet to the appropriate output port.
3) *Forward:* The set_egress_port() action selects an output port for unicast forwarding, while P4_broadcast() is used for multicast forwarding to multiple output ports.

*Enable_IP Match-Action Module:* This module encapsulates the MID_packet with an IP header by matching hdr.ip.fwd and executing the hdr.ip.SetValid() action, enabling compatibility with WANs and ensuring seamless communication with TCP/IP.

*P4_deparser:* This module reconstructs the fields and metadata into the final MID_packet.

The MID header processing relies on simple match-action operations, avoiding complex logic processing or packet recirculation. This lightweight data plane design is well-suited to the resource-constrained MANET nodes, enabling diverse forwarding logic while maintaining low-resource usage.

### D. OpenL3 Workflow

To provide a comprehensive overview, we summarize the workflow of OpenL3 in this section.

The whole workflow of OpenL3 is illustrated in Fig. 8. When a MANET node initiates communication, it queries the mapping nodes to retrieve the relevant MID based on the application type. The routing module then computes a forwarding path, sending the data packet to the next hop. Upon receiving a MID_packet, the node performs flow table matching to determine the appropriate forwarding behavior, guided by the MID_Flag. For content-centric forwarding, the node matches the content based on the SUID and returns the requested content to the source node. Specifically, OpenL3 supports diverse network services as follows.

*Host-Centric Services:* Nodes are identified using DUID, similar to IP addresses in TCP/IP. A source node queries the mapping system to obtain the DUID of the target node, enabling data packets to be routed to their destination.

*Content-Centric Services:* This communication employs a request-reply mechanism. Nodes first identify the desired content name and query the mapping server to retrieve $SUID_c$. Interest packets are sent to the destination node, which responds with the corresponding data packets. To improve efficiency, a PIT table stores unresolved interest packets to prevent duplicate requests and enable multiplexing.

TABLE II
SIMULATION PARAMETERS

| Parameters | Settings |
|---|---|
| Number of nodes | 20-100 |
| Movement speed of nodes | 1-5 m/s |
| MAC protocol | IEEE 802.11 |
| Evaluation area size | $100 \times 100\ m^2$ |
| Max communication range of nodes | 30 m |
| Mobile Model | Random Way Point |
| Evaluation duration | 1 minute |
| Evaluation times | 10 |

*Location-Centric Services:* For communications targeting a geographic region, the sender specifies the desired area and queries the mapping system to obtain $\text{SUID}_L$. Nodes within this area are identified, and the sender multicasts the MID_packets to them using location-based routing.

*Multicast Services:* OpenL3 supports dynamic multicast services using group maintenance operations in the MID mapping system. Groups can be predefined or created dynamically based on application requirements. Group memberships are updated following MID registration. Unlike traditional multicast methods, OpenL3 allows temporary group formation tailored to application needs, reducing the overhead for preestablished group management.

Beyond the services mentioned above, the programmability of OpenL3 ensures scalability beyond the described services. Its flexible design allows the modification and deployment of new workflows efficiently, ensuring adaptation to diverse applications in MANET environments.

## V. PERFORMANCE EVALUATION

In this section, we first introduce the experimental setup, including the simulation and prototype system. Then, we experimentally evaluate the performance of OpenL3.

### A. Experimental Setup

*1) Simulation Evaluation:* To evaluate the performance of OpenL3 across varying numbers of network nodes and different selections of mapping nodes, we create a simulation environment. The simulation setup included different node counts to represent a range of MANET scales, from small to large networks. Nodes are randomly deployed within a predefined area to simulate different network topologies.

*Simulation Parameters:* The initial positions of the nodes are uniformly distributed in a rectangular area of $100 \times 100$ m$^2$, and then they move in the Random Waypoint model (RWP). The stop time in the RWP model is set to 0. The speeds of the nodes vary from 1 to 5 m/s. The specific prototype system implementation parameters are set in Table II.

*2) Prototype System Implementation:* To evaluate the performance of OpenL3 in real-world conditions, we implemented a prototype system using Raspberry Pi devices.

*Hardware Environment:* The prototype system is built on Raspberry Pi, featuring the BCM2711B0 chipset with a 1.5GHz quad-core CPU and 8GB of memory. Each Raspberry Pi is equipped with a WLAN card to enhance wireless communication and a GPS module for geolocation (latitude and longitude). The hardware setup is illustrated in Fig. 9.

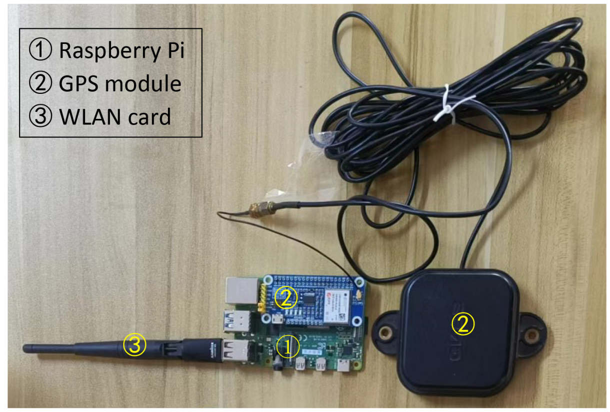



Fig. 9. Hardware of the prototype system.

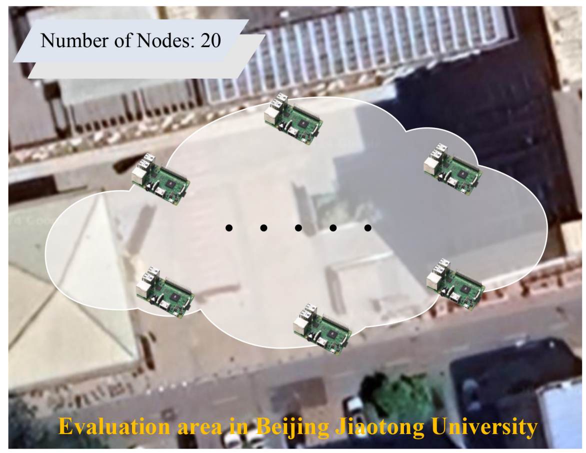



Fig. 10. Evaluation area.

*Software Environment:* The system uses the ONOS controller, P4 runtime environment, and MySQL database. The ONOS controller manages the distributed control plane, interfacing with the data plane via P4Runtime for flow table installation. A MySQL database stores mapping entries essential for system operation. P4Pi is installed on each Raspberry Pi to create a P4 environment, with packet forwarding managed using the bmv2 program [18].

*Implementation Parameters:* The open area within the Beijing Jiaotong University campus was chosen for evaluation. Multiple network nodes were randomly deployed in a $40 \times 20$ m$^2$ area, as illustrated in Fig. 10. Networks comprising 20 nodes were selected, with node positions randomly altered for each experiment. Nodes, carried by the experimenter, moved randomly at 1 m/s in the RWP model. The specific prototype system implementation parameters are set in Table III.

### B. Evaluation of OpenL3 Architecture

To evaluate the scalability and robustness of OpenL3 in larger networks, we analyzed its performance under varying network sizes and mapping node ratios (MNR). These experiments assessed two key scalability metrics.

TABLE III
PROTOTYPE SYSTEM PARAMETERS

| Parameters | Settings |
|---|---|
| Number of nodes | 5, 10, 20 |
| Movement speed of Nodes | 1 m/s |
| MAC protocol | IEEE 802.11 |
| Evaluation area size | $40 \times 20\ m^2$ |
| Mobile Model | Random Way Point |
| Evaluation duration | 1 minute |
| Evaluation times | 10 |

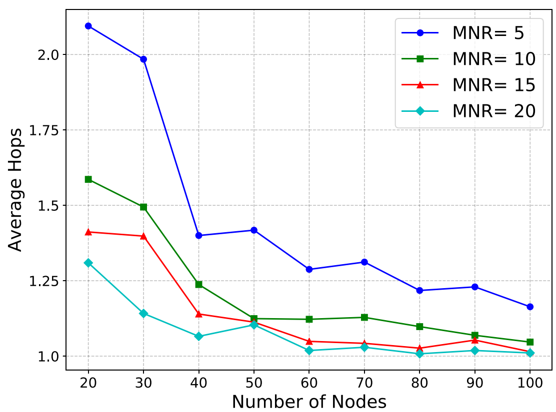



Fig. 11. Average hops of mobile nodes to their nearest mapping nodes.

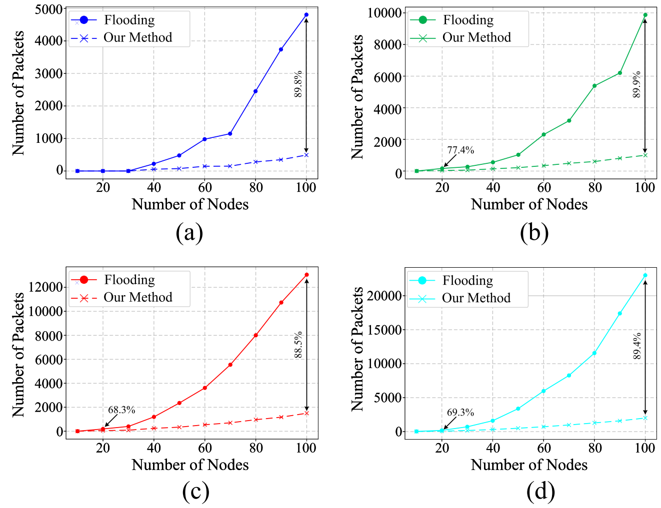



Fig. 12. Number of synchronous packets of the mapping nodes. (a) MNR = 5. (b) MNR = 10. (c) MNR = 15. (d) MNR = 20.

1) *Mapping Hops:* Reflecting routing efficiency, this metric evaluates how the number of mapping hops changes as the network grows.
2) *Synchronization Overhead:* Measuring communication efficiency, this metric quantifies the cost of synchronizing mapping information as the network scales.

We conducted simulations in networks with 20 to 100 nodes, covering various node densities and MNR values ranging from 5 to 20. Each experiment was repeated 10 times to ensure reliability.

Fig. 11 demonstrates that as the network size increases, the average number of mapping hops decreases and converges to approximately one hop, particularly at higher MNR values. This indicates that OpenL3 can maintain low-latency mapping operations even as the network scales.

Additionally, Fig. 12 shows that while synchronization overhead increases with the number of nodes, our scheme mitigates this growth. For example, in a 100-node network, our method reduces synchronization overhead by 88.9% to 89.9%, depending on the MNR. These results confirm that OpenL3 remains efficient in large-scale MANETs, balancing mapping efficiency and communication overhead.

This analysis demonstrates that OpenL3 effectively scales with network size and adapts to varying MNR values, achieving low-latency mapping operations and efficient synchronization. Its inherent robustness further supports deployment in dynamic and heterogeneous MANET environments, making it a promising solution for large-scale networks.

## C. Evaluation of MID Mapping System

To evaluate the performance of the mapping system, we assessed the response time for mapping operations, including registration and queries. Response time refers to the duration from when a node initiates a mapping request to when it receives a response. It covers both processing latency in the mapping nodes and wireless transmission time. This metric measures the efficiency of the MID mapping system.

For this evaluation, we used a MANET with 20 mobile nodes and set the MNR to 20%. Mobile nodes sent mapping registration and query requests to their nearest mapping nodes and received responses. Using JMeter on network nodes, we conducted 10 to 100 iterations of stress evaluations to simulate concurrent mappings in varying-scale MANET scenarios.

Fig. 13 presents the latency of mapping registration and queries under different concurrency levels using a box plot to depict data characteristics. Results indicate that mapping registration latency is 40 ms, while mapping query latency is 15 ms. As concurrency increases, mapping response latency slightly increases, whereas query latency remains constant. Some outliers were observed, likely due to wireless channel interference and system resource contention, but they did not significantly impact overall performance.

The MID mapping system demonstrates low-response latency and effectively supports mapping operations under normal concurrency levels.

## D. Evaluation of MID_Packet Routing and Forwarding

We evaluated the routing and forwarding performance of MID_packets using the following metrics.

1) *Average Routing Latency:* Measures the ability of the control plane to compute routes and install flow table entries when switching among diverse services.
2) *Average Throughput:* Assesses the efficiency of MID_packet forwarding, given the additional data plane processing steps compared to IP.

To evaluate routing latency, we sent MID_packets to trigger control plane decisions. The latency was measured as the time taken from receiving a `packet_in` message to executing a `packet_out` message. Results were compared with NDN/IP [15] and IPoC [16].

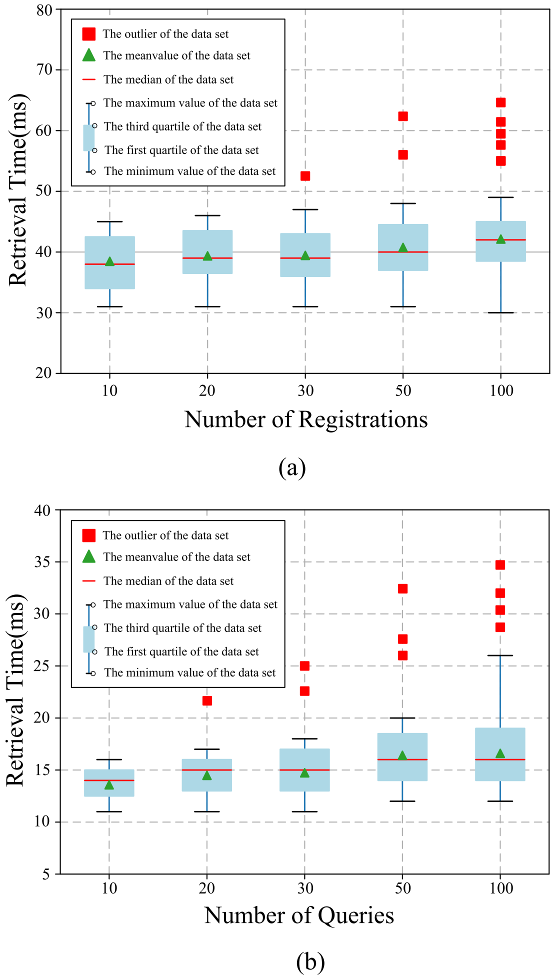



Fig. 13. Latency of the MID mapping system under different levels of concurrency. (a) Registration. (b) Query.

As shown in Fig. 14, OpenL3 achieves lower routing decision latency compared to NDN/IP and IPoC. OpenL3 benefits from a unified service format, eliminating the need for protocol stack adjustments during service switching. In contrast, NDN/IP introduces higher latency due to its multilayer protocol design based on the IP stack. IPoC, which maps NDN network functions onto IP tunnels, introduces additional overhead for tunnel management and mapping.

To evaluate throughput, we injected MID traffic into a BMV2 switch at a rate of 300 Mb/s, varying packet sizes from 50 to 300 bytes in increments of 10 bytes. The average throughput was recorded over 1 min for each packet size.

Fig. 15 shows that while MID_packet forwarding involves slightly higher processing costs than IP due to additional match-action flow table operations, the performance gap diminishes as packet size increases. At a packet size of approximately 170 bytes, MID_packet forwarding achieves line-rate performance. Under high-packet loads, MID and IP demonstrate comparable forwarding efficiency. These results confirm that OpenL3 supports diverse services with minimal additional forwarding latency.

## *E. Evaluation of Diverse Network Services*

OpenL3 supports diverse network services in MANETs. Its performance can be evaluated using the following metrics.

1) *End-to-End Latency:* The time taken for a data packet to travel from source to destination, reflects the responsiveness of the network.
2) *Throughput:* The amount of data successfully transmitted per unit of time, indicates the efficiency of data transfer.

Experiments were conducted in MANETs with 5 to 20 mobile nodes and an MNR of 20%. These metrics were assessed for three service types, representing real-world disaster relief scenarios.

1) *Host-Centric Service* simulates task offloading in scenarios, such as post-disaster computing, where computation tasks are sent to nearby mobile nodes for processing. We evaluated the latency and throughput of unicast (node-to-node) and multicast (node-to-multiple-nodes) communication.
2) *Content-Centric Service* represents data collection scenarios, such as gathering sensor readings or images from affected areas for situational analysis. The evaluation measures latency and throughput for both single-content retrieval (e.g., one image or report) and multiple-content retrieval (e.g., bulk sensor data or reports).
3) *Location-Centric Service* focuses on strategy distribution, such as broadcasting operational plans or updates to all devices in a specific geographic region. The assessment includes latency and throughput for communications targeting nodes within designated areas.

To the best of our knowledge, there is no integrated approach natively supporting these diverse network services. For comparison, we selected TCP/IP and NDN, which are representative host-centric and content-centric architectures, respectively. While TCP/IP can support content-centric and location-centric services via application-layer designs [24], [25], and NDN can accommodate host-centric service using additional operations [45]), OpenL3 natively integrates these capabilities with minimal complexity.

*1) Host-Centric Service:* We evaluated host-centric unicast services using MANETs with 5 to 20 nodes and multicast services with 2 to 8 nodes within a 20-node MANET. For host-centric unicast service [Fig. 16(a) and (b)], OpenL3 achieves comparable latency and throughput to IP, while NDN suffers significantly higher latency and lower throughput due to its additional communication process for host-centric service [45] and reliance on CS and PIT operations. Latency for OpenL3 and IP grows modestly with network size, while NDN experiences significant latency increases. In addition, for multicast service [Fig. 17(a) and (b)], OpenL3 maintains stable latency as the number of nodes increases, whereas IP and NDN experience significant latency growth. For 8 multicast nodes, OpenL3 reduces latency by 61.4% compared to IP and 86.2% compared to NDN. Then, OpenL3 achieves substantially higher throughput with more multicast nodes. This is thanks to its on-demand group formation and efficient routing, unlike traditional multicast strategies requiring predefined groups.

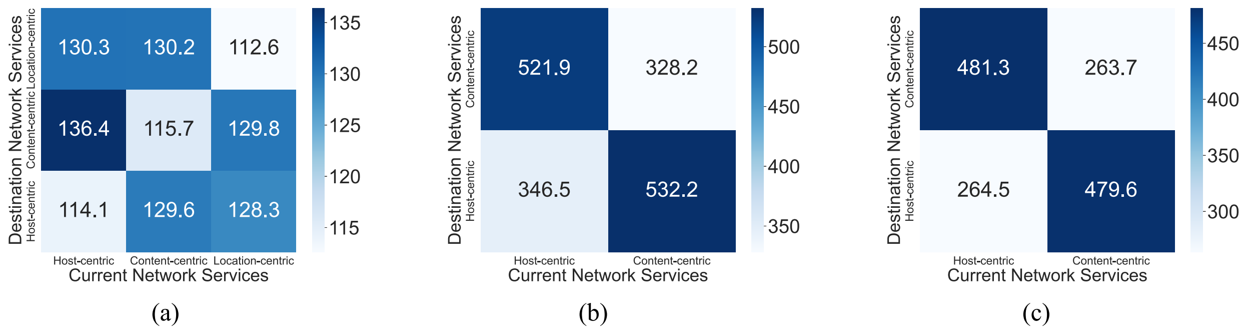



Fig. 14. Control plane route switching latency for OpenL3. (a) OpenL3. (b) NDN/IP. (c) IPoC.

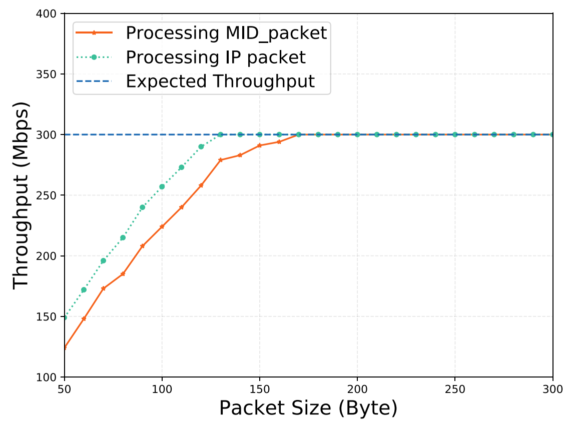



Fig. 15. Throughput of the BMv2 switch for processing MID_packets of different sizes.

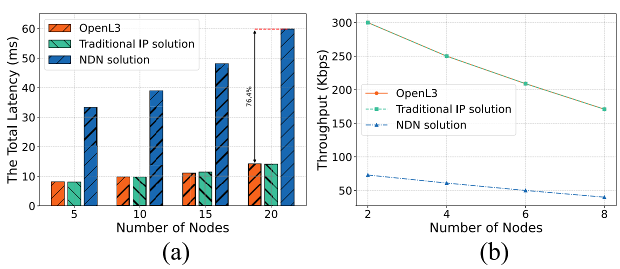



Fig. 16. Communication performance comparison between OpenL3 and existing solutions in host-centric unicast service. (a) Latency. (b) Throughput.

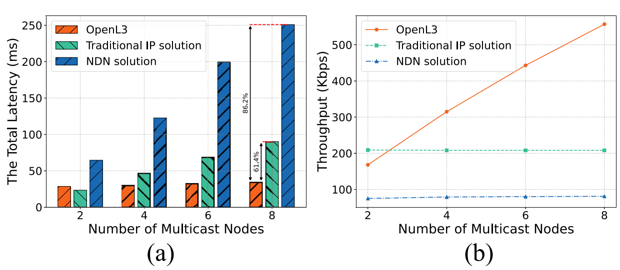



Fig. 17. Communication performance comparison between OpenL3 and existing solutions in host-centric multicast service. (a) Latency. (b) Throughput.

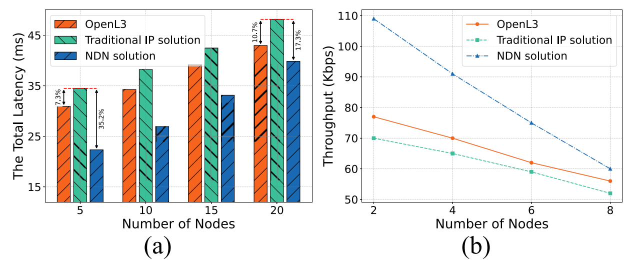



Fig. 18. Communication performance comparison between OpenL3 and existing solutions in content-centric unicast service. (a) Latency. (b) Throughput.

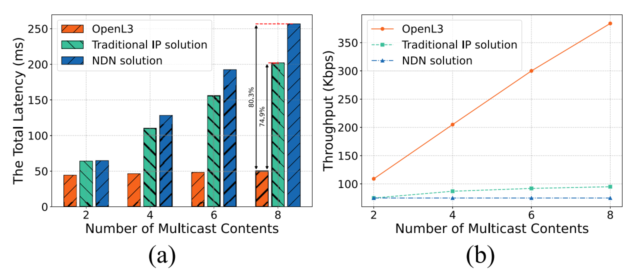
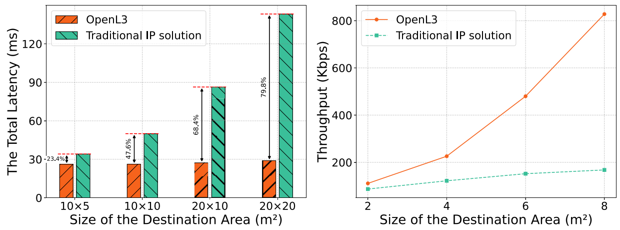



Fig. 19. Communication performance comparison between OpenL3 and existing solutions in content-centric multicast service. (a) Latency. (b) Throughput.

*2) Content-Centric Service:* We deployed 100 content items across 20 nodes and evaluated single-content retrieval (5 to 20 nodes) and multiple-content retrieval (2 to 8 items). For content-centric unicast service [Fig. 18(a) and (b)], NDN achieves the lowest latency and highest throughput due to its content-based routing. However, NDN loses its performance advantage as the network grows, while OpenL3 and IP maintain stable performance. For multiple-content retrieval [Fig. 19(a)–(c)], OpenL3 demonstrates the lowest latency, with reductions of 74.9% and 80.3% compared to IP and NDN, respectively, for 8 content items. OpenL3 achieves significantly higher throughput due to its multicast-on-demand capability, enabling efficient batch content retrieval. In contrast, IP and NDN struggle to support multiple simultaneous content requests.

*3) Location-Centric Service:* We randomly deploy 20 mobile nodes in a 40 $\times$ 20 $m^2$ area, varying the target communication area size. As shown in Fig. 20(a) and (b), OpenL3 achieves significantly lower latency and higher throughput than IP. With increasing area size, OpenL3 reduces latency by 23.4%, 47.6%, 68.4%, and 79.8%, while throughput increases substantially. IP relies on querying geographic and IPv6 addresses from centralized servers [25], leading




(a) (b)

Fig. 20. Communication performance comparison between OpenL3 and existing solutions in location-centric service. (a) Latency. (b) Throughput.

TABLE IV
COMPARISON OF OPENL3 WITH EXISTING SOLUTIONS

| Contents | TCP/IP | NDN | OpenL3 |
|---|---|---|---|
| End-to-end latency (ms) | 61.42 | 96.28 | 30.28 |
| Network throughput (Kbps) | 146.98 | 76.58 | 414.85 |
| Scalability | Low | High | High |
| Resource consumption | Low | High | Low |
| Flexibility | Medium | Low | High |
| Security | Low | High | High |
| Ease of Implementation | Easy | Hard | Easy |

to significant delays as node numbers grow. In contrast, OpenL3 directly routes packets to target regions without querying individual devices, maintaining low latency and high throughput as the area size increases.

### F. Comparison of OpenL3 with Existing Solutions

Table IV summarizes the comparative analysis of OpenL3, TCP/IP, and NDN in terms of latency, scalability, resource consumption, security, flexibility, and ease of implementation.

*1) Latency and Throughput:* Latency and throughput performance measures the efficiency of routing and forwarding. In a real-world environment, we measured the average end-to-end latency and throughput of 20 MANET nodes in 1 min. Each node sends service requests with an 80% probability per second (based on campus network traffic patterns). The results showed that OpenL3 achieved 30.68 ms, significantly outperforming TCP/IP (61.42 ms) and NDN (96.28 ms). In addition, the throughput of OpenL3 is 414.85 Kb/s, which is better than TCP/IP (146.98 Kb/s) and NDN (76.58 Kb/s) respectively. In conclusion, OpenL3 performs well both in latency and throughput.

*2) Scalability:* The scalability reflects the adaptability of the schemes to large-scale MANETs. TCP/IP requires centralized service registration and query, and frequent maintenance of group relationships, which makes it difficult to maintain high efficiency in large-scale MANETs. NDN enhances scalability by decoupling communication from endpoints but suffers from growing PIT and FIB table overhead in dense networks. OpenL3 addresses these limitations with a lightweight programmable data plane and hierarchical MID-based addressing, enabling efficient operation in dynamic, large-scale MANETs.

*3) Resource Consumption:* Resource consumption includes computation, memory, and bandwidth usage. TCP/IP minimizes the data transmission use but requires high computation and memory for service registration and updates. NDN incurs greater resource demands as its PIT and FIB tables grow. OpenL3 reduces computational and memory requirements with simple match-action forwarding and hierarchical MIDs, making it more resource-efficient in constrained environments.

*4) Security:* Security evaluates the ability of schemes to mitigate potential attacks. TCP/IP associates IP addresses with user identity, location, and content, making it vulnerable to spoofing, privacy breaches, and other attacks. NDN employs content-based encryption and signature mechanisms to prevent forgery and ensure data integrity. OpenL3 enhances security by using location-agnostic identifiers and distributed self-certification. It facilitates mutual authentication between devices without relying on third-party entities, strengthening its defense against network threats.

*5) Flexibility:* Flexibility refers to the ability to adapt to diverse network services. TCP/IP achieves flexibility by supporting application-layer protocols, which often results in increased latency and preconfiguration requirements. NDN is optimized for content-centric services, but it struggles to accommodate other services. OpenL3 introduces MID to represent diverse network entities and employs P4-based SDN to enable seamless switching among the network services.

*6) Ease of Implementation:* Ease of implementation assesses the practical feasibility of deploying a network architecture. TCP/IP, as the current standard, is readily supported by most devices. NDN requires specialized routers and protocol stack modifications, complicating deployment. OpenL3 bridges this gap by leveraging P4-based SDN, which operates on existing hardware and protocol stacks without requiring specialized devices. This approach simplifies deployment, making OpenL3 more practical than NDN.

## VI. CONCLUSION

In this article, we have presented OpenL3, a novel approach for enabling diverse network services in MANETs. OpenL3 incorporates MID as the network address to control the routing and forwarding processes. Then, we have proposed a programmable routing and forwarding scheme leveraging P4-based SDN, facilitating interoperability among various L3 protocols. A prototype system has been built to validate the proposed solution. Experimental results have demonstrated the effectiveness of OpenL3 in end-to-end latency and network throughput compared to existing solutions. OpenL3 differs from traditional solutions, with the ability to adapt to diverse network services without modifications to the network protocol stack. In addition to MANETs, the OpenL3 design principle that integrates diverse routing and forwarding can be extended to other network architectures. For future work, we will incorporate emerging network services and security mechanisms to ensure robust and secure communication in MANETs.

## REFERENCES

[1] W.-K. Jia and C.-M. Yu, "WALS: A lightweight stateless label switching framework based on prime theorem for MANETs," *IEEE Internet Things J.*, vol. 10, no. 20, pp. 17905–17919, Oct. 2023.

[2] G. Liu, H. Dong, Z. Yan, X. Zhou, and S. Shimizu, "B4SDC: A blockchain system for security data collection in MANETs," *IEEE Trans. Big Data*, vol. 8, no. 3, pp. 739–752, Jun. 2022.

[3] S. H. Alsamhi et al., "UAV computing-assisted search and rescue mission framework for disaster and harsh environment mitigation," *DRONES*, vol. 6, no. 7, p. 154, 2022.

[4] W. Zhang, D. Yang, C. Zhang, Q. Ye, H. Zhang, and X. Shen, "(Com)2Net: A novel communication and computation integrated network architecture," *IEEE Netw.*, vol. 38, no. 2, pp. 35–44, Mar. 2024.

[5] X. Qiu, S. Zhang, Z. Wang, and H. Luo, "Integrated host- and content-centric routing for efficient and scalable networking of UAV swarm," *IEEE Trans. Mobile Comput.*, vol. 23, no. 4, pp. 2927–2942, Apr. 2024.

[6] X. Shen et al., "AI-assisted network-slicing based next-generation wireless networks," *IEEE Open J. Veh. Technol.*, vol. 1, pp. 45–66, Jan. 2020.

[7] X. Wang and Y. Lu, "Efficient forwarding and data acquisition in NDN-based MANET," *IEEE Trans. Mobile Comput.*, vol. 21, no. 2, pp. 530–539, Feb. 2020.

[8] X. Hou, S. Gao, N. Liu, F. Yao, H. Zhang, and S. K. Das, "L3Geocast: Enabling P4-based customizable network-layer geocast at the network edge," *IEEE Trans. Mobile Comput.*, vol. 23, no. 8, pp. 8323–8340, Aug. 2024.

[9] A. Dowling, Y. Liu, L. Huie, and K. Chen, "Building an information-centric and LoRa-based sensing platform for IoT," in *Proc. IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, 2021, pp. 1–6.

[10] L. Zhang et al., "Named data networking," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 3, pp. 66–73, 2014.

[11] V. Sandonis, I. Soto, M. Calderon, and M. Urueña, "Vehicle to Internet communications using the ETSI ITS GeoNetworking protocol," *Trans. Emerg. Telecommun. Technol.*, vol. 27, no. 3, pp. 373–391, 2016.

[12] M. A. Khan et al., "Swarm of UAVs for network management in 6G: A technical review," *IEEE Trans. Netw. Service Manag.*, vol. 20, no. 1, pp. 741–761, Mar. 2023.

[13] W. Zhang, N. Tang, D. Yang, R. Guo, H. Zhang, and X. Shen, "Det(Com)2: Deterministic communication and computation integration toward AIGC services," *IEEE Wireless Commun.*, vol. 31, no. 3, pp. 32–41, Jun. 2024.

[14] M. Conti, A. Gangwal, M. Hassan, C. Lal, and E. Losiouk, "The road ahead for networking: A survey on ICN-IP coexistence solutions," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 3, pp. 2104–2129, 3rd Quart., 2020.

[15] S. Luo, S. Zhong, and K. Lei, "IP/NDN: A multi-level translation and migration mechanism," in *Proc. IEEE/IFIP Netw. Oper. Manag. Symp. (NOMS)*, 2018, pp. 1–5.

[16] S. Shannigrahi, C. Fan, and G. White, "Bridging the ICN deployment gap with IPOC: An IP-over-ICN protocol for 5G networks," in *Proc. Workshop Netw. Emerg. Appl. Technol.*, 2018, pp. 1–7.

[17] Y. Thomas, N. Fotiou, S. Toumpis, and G. C. Polyzos, "Improving mobile ad hoc networks using hybrid IP-information centric networking," *Comput. Commun.*, vol. 156, pp. 25–34, Apr. 2020.

[18] S. Kaur, K. Kumar, and N. Aggarwal, "A review on P4-programmable data planes: Architecture, research efforts, and future directions," *Comput. Commun.*, vol. 170, pp. 109–129, Mar. 2021.

[19] M. Bin-Yahya, O. Alhussein, and X. Shen, "Securing software-defined WSNs communication via trust management," *IEEE Internet Things J.*, vol. 9, no. 22, pp. 22230–22245, Nov. 2022.

[20] H. Wang, H. Zhao, J. Zhang, D. Ma, J. Li, and J. Wei, "Survey on unmanned aerial vehicle networks: A cyber physical system perspective," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 2, pp. 1027–1070, 2nd Quart., 2020.

[21] F. Pasandideh, J. P. J. da Costa, R. Kunst, N. Islam, W. Hardjawana, and E. P. de Freitas, "A review of flying ad hoc networks: Key characteristics, applications, and wireless technologies," *Remote Sens.*, vol. 14, no. 18, p. 4459, 2022.

[22] A. Mchergui, T. Moulahi, and S. Zeadally, "Survey on artificial intelligence (AI) techniques for vehicular ad-hoc networks (VANETs)," *Veh. Commun.*, vol. 34, Apr. 2022, Art. no. 100403.

[23] W. Zhang, Y. He, T. Zhang, C. Ying, and J. Kang, "Intelligent resource adaptation for diversified service requirements in industrial IoT," *IEEE Trans. Cogn. Commun. Netw.*, early access, Nov. 19, 2024, doi: 10.1109/TCCN.2024.3502493.

[24] M. Hormati, F. Belqasmi, R. Glitho, and F. Khendek, "A DNS protocol—Based service discovery architecture for disaster response systems," in *Proc. IEEE Symp. Comput. Commun. (ISCC)*, 2013, pp. 366–371.

[25] T. Fioreze and G. Heijenk, "Extending the domain name system (DNS) to provide geographical addressing towards vehicular ad-hoc networks (VANETs)," in *Proc. IEEE Veh. Netw. Conf. (VNC)*, 2011, pp. 70–77.

[26] A. K. Niari, R. Berangi, and M. Fathy, "ECCN: An extended CCN architecture to improve data access in vehicular content-centric network," *J. Supercomput.*, vol. 74, pp. 205–221, Jan. 2018.

[27] B. Nour et al., "A survey of Internet of Things communication using ICN: A use case perspective," *Comput. Commun.*, vols. 142–143, pp. 95–123, Jun. 2019.

[28] A. Tariq, R. A. Rehman, and B.-S. Kim, "Forwarding strategies in NDN-based wireless networks: A survey," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 1, pp. 68–95, 1st Quart., 2020.

[29] B. Meijerink, M. Baratchi, and G. Heijenk, "Design & analysis of a distributed routing algorithm towards internet-wide geocast," *Comput. Commun.*, vol. 146, pp. 201–218, Oct. 2019.

[30] B.-N. Cheng, G. Kuperman, P. Deutsch, L. Mercer, and A. Narula-Tam, "Group-centric networking: Addressing information sharing requirements at the tactical edge," *IEEE Commun. Mag.*, vol. 54, no. 10, pp. 145–151, Oct. 2016.

[31] X. Guo, N. Liu, X. Hou, S. Gao, and H. Zhou, "An efficient NDN routing mechanism design in P4 environment," in *Proc. 2nd Inf. Commun. Technol. Conf. (ICTC)*, 2021, pp. 28–33.

[32] Y. Ren, R. Xie, F. R. Yu, T. Huang, and Y. Liu, "Potential identity resolution systems for the industrial Internet of Things: A survey," *IEEE Commun. Surveys Tuts.*, vol. 23, no. 1, pp. 391–430, 1st Quart., 2021.

[33] A. Mühle, A. Grüner, T. Gayvoronskaya, and C. Meinel, "A survey on essential components of a self-sovereign identity," *Comput. Sci. Rev.*, vol. 30, pp. 80–86, Nov. 2018.

[34] M. A. Al-Shareeda, M. Anbar, I. H. Hasbullah, and S. Manickam, "Survey of authentication and privacy schemes in vehicular ad hoc networks," *IEEE Sensors J.*, vol. 21, no. 2, pp. 2422–2433, Jan. 2021.

[35] H. Deng et al., "Identity-based encryption transformation for flexible sharing of encrypted data in public cloud," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 3168–3180, 2020.

[36] M. Achir, A. Abdelli, L. Mokdad, and J. Benothman, "Service discovery and selection in IoT: A survey and a taxonomy," *J. Netw. Comput. Appl.*, vol. 200, Apr. 2022, Art. no. 103331.

[37] A. Heidari and N. J. Navimipour, "Service discovery mechanisms in cloud computing: A comprehensive and systematic literature review," *Kybernetes*, vol. 51, no. 3, pp. 952–981, 2022.

[38] V. Raychoudhury, J. Cao, W. Wu, Y. Lai, C. Chen, and J. Ma, "K-directory community: Reliable service discovery in MANET," *Pervasive Mobile Comput.*, vol. 7, no. 1, pp. 140–158, 2011.

[39] G. B. Brahim, N. Mohammad, W. El-Hajj, G. Parr, and B. Scotney, "Performance evaluation and comparison study of adaptive MANET service location and discovery protocols for highly dynamic environments," *EURASIP J. Wireless Commun. Netw.*, vol. 2022, pp. 1–21, Dec. 2022.

[40] M. Kadadha and H. Otrok, "A blockchain-enabled relay selection for QoS-OLSR in urban VANET: A Stackelberg game model," *Ad Hoc Netw.*, vol. 117, Jun. 2021, Art. no. 102502.

[41] A. M. Hoque, S. O. Amin, A. Alyyan, B. Zhang, L. Zhang, and L. Wang, "NLSR: Named-data link state routing protocol," in *Proc. 3rd ACM SIGCOMM Workshop Inf.-Centric Netw.*, 2013, pp. 15–20.

[42] A. Montazerolghaem and M. H. Yaghmaee, "Load-balanced and QoS-aware software-defined Internet of Things," *IEEE Internet Things J.*, vol. 7, no. 4, pp. 3323–3337, Apr. 2020.

[43] A. Montazerolghaem, "Software-defined Internet of Multimedia Things: Energy-efficient and load-balanced resource management," *IEEE Internet Things J.*, vol. 9, no. 3, pp. 2432–2442, Feb. 2022.

[44] X. Hou et al., "L3DML: Facilitating geo-distributed machine learning in network layer," *IEEE Trans. Netw. Service Manag.*, early access, Nov. 29, 2024, doi: 10.1109/TNSM.2024.3509031.

[45] V. Jacobson et al., "VoCCN: Voice-over content-centric networks," in *Proc. Workshop Re-Archit. Internet*, 2009, pp. 1–6.


**Jiangyu Lan** received the B.S. and M.S. degrees in information and communication engineering from Beijing Information Science and Technology University, Beijing, China, in 2018 and 2021, respectively. He is currently pursuing the Ph.D. degree with the National Engineering Research Center for Advanced Network Technologies, Beijing Jiaotong University, Beijing.

His research interests include MANETs, future network addressing and routing, and programmable data plane.

**Shuai Gao** (Member, IEEE) received the B.S., M.S., and Ph.D. degrees in communication and information systems from Beijing Jiaotong University, Beijing, China, in 2001, 2004, and 2010, respectively. From 2015 to 2016, he was a Visiting Scholar with The University of Arizona, Tucson, AZ, USA. He is currently a Full Professor with the School of Electronic and Information Engineering, Beijing Jiaotong University. His research interests are in the areas of Internet architecture, sensor networks, and mobile Internet.

**Weiting Zhang** (Member, IEEE) received the Ph.D. degree in communication and information systems from Beijing Jiaotong University, Beijing, China, in 2021.

From November 2019 to November 2020, he was a visiting Ph.D. student with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada. Starting from December 2021, he works as an Associate Professor with the School of Electronic and Information Engineering, Beijing Jiaotong University. His research interests include industrial Internet of Things, deterministic networks, edge intelligence, and machine learning for network optimization.

**Xindi Hou** received the M.S. degree in information and communication engineering from Northeast Electric Power University, Jilin, China, in 2018. He is currently pursuing the Ph.D. degree with the National Engineering Research Center for Advanced Network Technologies, Beijing Jiaotong University, Beijing, China.

His research interests include clean-slate networks, future network addressing and routing, compute-aware networks, and programmable data plane.

**Minghui Xi** is currently pursuing the postgraduate degree with the National Engineering Research Center for Advanced Network Technologies, Beijing Jiaotong University, Beijing, China.

His current research interests include network security and P4 programmable data plane.

**Yuming Zhang** (Member, IEEE) received the Ph.D. degree in telecommunications and information system from the National Engineering Laboratory for Next Generation Internet Interconnection Devices, Beijing Jiaotong University, Beijing, China, in 2021.

He severed as a Postdoctoral Fellow with the School of Computer Science, Beijing University of Posts and Telecommunications, Beijing, from 2021 to 2024. He is currently a Lecturer with the School of Electronic and Information Engineer, Beijing Jiaotong University. His current research interests include drone network, mobile edge computing, queue theory, and machine learning.

**Bo Lei** received the master's degree in telecommunication engineering from the Beijing University of Posts and Telecommunications, Beijing, China, in 2006.

He is currently the Deputy Director of Network Technology Research Department, China Telecom Co. Ltd. Research Institute, Beijing, China, and the Professor Level Senior Engineer. He currently works as a Visiting Research Fellow of Computer Network Information Center, Chinese Academy of Sciences, Beijing; a Part-Time Professor of Beijing University of Posts and Telecommunications, Beijing; the Vice-Chairman of CCSA TC617 "Edge Computing Industry Development and Standards Promotion Committee"; and the Deputy Leader of Computing Power Network Special Work Group in CCSA TC614 "Network 5.0 Technical Standards Promotion Committee." His research interests include cloud-network convergence, computing power network, and future network technology.

**Hongke Zhang** (Fellow, IEEE) received the M.S. and Ph.D. degrees in electrical and communication systems from the University of Electronic Science and Technology of China, Chengdu, China, in 1988 and 1992, respectively.

He is currently a Professor with the School of Electronic and Information Engineering, Beijing Jiaotong University, Beijing, China, and the Director of the National Engineering Research Center for Advanced Network Technologies, Beijing. He has authored more than ten books and is the holder of more than 70 patents. His research has resulted in many papers, books, patents, systems, and equipment in the areas of communications and computer networks.

**Xuemin (Sherman) Shen** (Fellow, IEEE) received the Ph.D. degree in electrical engineering from Rutgers University, New Brunswick, NJ, USA, in 1990.

He is a University Professor with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada. His research focuses on network resource management, wireless network security, Internet of Things, 5G and beyond, and vehicular ad hoc and sensor networks.

Dr. Shen received the R.A. Fessenden Award in 2019 from IEEE, Canada, the Award of Merit from the Federation of Chinese Canadian Professionals (Ontario) in 2019, the James Evans Avant Garde Award in 2018 from the IEEE Vehicular Technology Society, the Joseph LoCicero Award in 2015 and Education Award in 2017 from the IEEE Communications Society, and the Technical Recognition Award from Wireless Communications Technical Committee (2019) and AHSN Technical Committee (2013). He has also received the Excellent Graduate Supervision Award in 2006 from the University of Waterloo and the Premier's Research Excellence Award in 2003 from the Province of Ontario, Canada. He served as the Technical Program Committee Chair/Co-Chair for IEEE Globecom'16, IEEE Infocom'14, IEEE VTC'10 Fall, IEEE Globecom'07, and the Chair for the IEEE Communications Society Technical Committee on Wireless Communications. He is the President Elect of the IEEE Communications Society. He was the Vice President for Technical and Educational Activities, the Vice President for Publications, the Member-at-Large on the Board of Governors, the Chair of the Distinguished Lecturer Selection Committee, and the member of IEEE Fellow Selection Committee of the ComSoc. He served as the Editor-in-Chief for the IEEE INTERNET OF THINGS JOURNAL, IEEE NETWORKS, and *Peer-to-Peer Networking and Applications*. He is a Registered Professional Engineer of Ontario, Canada, an Engineering Institute of Canada Fellow, a Canadian Academy of Engineering Fellow, a Royal Society of Canada Fellow, a Chinese Academy of Engineering Foreign Member, and a Distinguished Lecturer of the IEEE Vehicular Technology Society and Communications Society.