Computer Networks 219 (2022) 109433

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

Forwarding and caching in video streaming over ICSDN: A clean-slate publish-subscribe approach

M. Wasim Abbas Ashraf [a], Chuanhe Huang [a,*], Arif Raza [b], Kashif Sharif [c], Md Monjurul Karim [d], Shidong Huang [a]

[a] School of Computer Science, Wuhan University, 430072 Wuhan, China
[b] College of Computer Science and Software Engineering Shenzhen University Shenzhen, China
[c] School of Computer Science and Technology, Beijing Institute of Technology, 100081 Beijing, China
[d] Research Institute of Trustworthy Autonomous Systems, Southern University of Science and Technology, Shenzhen, China

ARTICLE INFO

Keywords:
Software-defined networking
Information-centric networking
Pub-Sub
Controller
Forwarding
Caching

ABSTRACT

Nowadays, Internet usage has become prevalent, primarily because of high-quality heterogeneous multimedia content expectations from the subscriber (consumer), which puts tremendous pressure on the publisher (producer) in the networks. Information-Centric Networking (ICN) is a future internet architecture that optimizes data resources through content-based forwarding and caching, making it well-suited for multimedia content and video streaming (VS) scenarios. However, real-time data delivery is challenging in the current ICN-based publish–subscribe (pub-sub) mechanism, which pushes the existing pub-sub studies to prioritize more on the forwarding information base (FIB) rather than the pending interest table (PIT). This leads to issues such as inefficient caching and forwarding mechanisms, high overhead, and communication costs. To address these challenges, in this paper, we present a novel forwarding and caching solution named VS-ICSDN, integrating the combined principles of ICN-based pub-sub scheme and software-defined networking (SDN) in order to utilize the network resources more efficiently. We design a clean-slate caching strategy and name-based forwarding method to support both on-path and off-path caching on ICN nodes to coordinate flow entries among the SDN controller and clean-slate ICN nodes to maximize PIT utilization. In addition, the framework allows the content to be stored and searched in chunks with a single request to access the desired content, reducing the communication overhead and significantly improving overall performance. A simulation-based testbed and experimental result analysis validate our proposed work's effectiveness in ensuring efficient network resource usage with low communication overhead and computational cost compared to other baseline methods.

1. Introduction

Video streaming services are the core of internet traffic and have become very popular in this digital world [1,2]. Internet applications from popular online services, e.g., Facebook, Youtube, Twitter, Youku, and live news channels, demand efficient content distribution mechanisms [3]. Therefore, the video steaming topic is popular in the research community regarding the Quality of Experience (QoE), future transmission bandwidth demand, and reducing delay in the whole network [4]. However, the current internet architecture is inefficient in providing the desired streaming quality and maintaining the network's position due to unsupported content recognition. Therefore, a future network paradigm named Information-Centric Network (ICN) appears with the new communication pattern pub-sub to prioritize content instead of a location that balances the desired data quality with the best service possible to improve the overall network performance [5]. Moreover, ICN brings great attention to video transmission [6] in which the subscriber subscribes to the number of interest packets as a video segment. Each transmission mode plays an important role in the ICN architecture, depending on the network condition. Besides, each node has its local cache that distributes the content with other nodes to overcome load balancing issues. The ICN delivers the data segment based on the hop [7]. A subscriber node is not recognized by its potential publisher. However, the ICN improves the video quality and somewhat mitigates challenges. Therefore, a suitable approach is still required to enhance video streaming performance [8]. On the other hand, high-demand media has been shifted to the internet, which demands Quality of Service (QoS) and QoE. The adaptive bitrate (ABR) bitrate algorithm

---

* Corresponding author.
E-mail addresses: wasimabbas@whu.edu.cn (M.W.A. Ashraf), huangch@whu.edu.cn (C. Huang), arifrazakh@gmail.com (A. Raza), kashif.sharif@ieee.org (K. Sharif), karim@mail.sustech.edu.cn (M.M. Karim), optelemag@whu.edu.cn (S. Huang).

https://doi.org/10.1016/j.comnet.2022.109433
Received 5 July 2022; Received in revised form 26 September 2022; Accepted 24 October 2022

was introduced [9] to deal with high-quality video streaming issues. Similarly, ABR and Dynamic Adaptive Streaming (DAS) have been proposed to improve the efficiency of video streaming [10].

Recently, the combined Information-centric and Software-defined networking (ICSDN) has become a promising concept to address the growing multimedia demands by utilizing in-network caching, and named-based content delivery [11]. SDN keeps the information about network topology [12,13] and provides a global view of the network through network function virtualization (NFV) [14] with a simplified and scalable centralized network control that allows automated task initiation, efficient off-path caching, robust, and data transmissions. Moreover, the SDN controller keeps the information bout ICN and provides in-network caching and multi-casting through disseminating the data by utilizing the content's ID compared to traditional Internet Protocol (IP) that mainly focuses on the location instead of content [15]. Therefore, the combination of ICN and SDN magnifies both technology's advantages. However, there are many challenges due to their differences in how they are implemented as sole technology [16]. For example, unlike SDN, which is based on IP address, ICN routes information using named objects instead of IP addresses. Besides, ICN adopts in-network caching, which is difficult to implement in the SDN that requires each network switch to be identified.

To tackle this issue, [17] discusses different methods to cache the content from the nearest node to improve the on-path caching performance but encounter problems that affect network performance. Firstly, the delay becomes a concern if the requested video content is close to the server node. Secondly, ICN nodes require additional computational overhead due to updating the header of content and data. On the other hand, off-path caching [18] aims to mitigate the on-path caching issues through an efficient caching scheme to improve the cache hit ratio and reduce delay and content overhead. The research community proposed many off-path caching methods and algorithms but did not meet the desired results. The existing strategies get the desired data by requesting individual requests for each video's content, resulting in increased content overhead and latency. In this regard, we introduce the ICN theory into SDN to help overcome these issues with content-based host-centric implementation. According to ICN's principle, various new methods can propose and implement in SDN [16]. Therefore, we propose the VS-ICSDN approach for video streaming in ICN-based pub/sub to overcome these issues.

Accordingly, this paper focuses on the practical realization of caching in the ICN-based pub/sub communication patterns, considering all its implementation challenges. Moreover, the core contributions of this paper are listed as follows:

- We formulate latency and cache hit problems for video streaming in an ICN-based pub/sub scenario. We propose a named-based forwarding and an on-off-path caching scheme in a clean-slate software-defined information-centric environment.
- We redesign the SDN controller according to the data-centric principle to synchronize with cached information of subscriber and publisher node and provide content at the minimum cost. Regarding data flow, we emphasize the PIT instead of FIB to improve network speed and reduce round-trip time and bandwidth usage.
- A content forwarding algorithm is designed to deal with pub/sub request packets. The SDN controller distributes the flow information of desired data in chunks to initiate scalable data transmission between publisher and subscriber.
- Finally, we provide ndnSIM-based extensive simulations and results assessments to justify the efficiency of our proposed model against baseline methods in terms of the cache hit ratio, latency, and overhead.

The rest of this paper is organized as follows. We discuss the related work in Section 2. In Section 3 the system model, communication model, and problem formulation are discussed. We presents clean-slate OpenFlow approach in Section 4. The immense analysis and VS-ICSDN performance are demonstrated in Section 5, and finally, we conclude the work in Section 6.

2. Related works

Many studies presented work to improve network communication costs, reduce delays, and provide reliable high-video streaming services based on different cache policies, i.e., FIFO, LFU, and LRU [19]. Most of these methods adopted ideas that cached the popular content on individual nodes but suffered some performance issues due to only focusing on cache behavior for an individual node [20–25]. The author introduced a distributed function based on scalable video coding (SVC) to improve video delivery performance [26]. A rate-based adaptive strategy is introduced in [27]. This idea continually inquiries the available bandwidth that varies the requested bitrate. A load balancing adaptive video scheme is discussed [28]. The author presented the cooperative caching [29] scheme using scheduling in which partial sensor node sleep with its ranging and others active to serve the request. A caching scheme is presented based on data lifetime, and the request rate [30] to reduce the delay in the network. Another cooperative cache study is proposed in [31,32] for a dynamic network. However, the result shows that the proposed scheme is good instead of mentioned existing work. Meanwhile many researchers [33–36] have considered the caching decision a typical NP-hard problem that placed the content in different networks. the authors introduced various way using polynomial time and greedy algorithm to solve this problem. The push-based caching scheme is proposed to embed the data or add a payload in the interest packet and data generated when the payload request is sent. Datacenter (DC) [37] plays an important role in SDN-based networks, so the author presents the over-subscription in DC to reduce the project's overall cost. The over-subscription term has defined the bandwidth ratio between the specific node to the final host node in which 1:1 indicates that all nodes communicate to other nodes using their total bandwidth. The demand for high-quality video content has increased rapidly so, many researchers paid attention for the growth in global video traffic [38–42]. The ICN approach brings an efficient method for distributing video streaming. Many studies discussed the proactive caching and routing methods in [43–46] to address the latency and caching popular data issues. To reduce the controller overhead and hop count, the router publicizes its neighboring exact prefix' availability. This proposed study is applied in the old version ndnSim-1.0, which has many deficiencies like no concept of flow installation, cache, and content refresh issues compared to the current ndnSIM version [47]. A transcoding scheme is suggested in [48] to store the highest content quality and forward it to others to improve the network performance.

Many forwarding methods are introduced in [53–55], which discusses that the control and the data planes are not logically separated. Additionally, controlling all traffic via the centralized controller decreased the whole system's efficiency. Similarly, the authors proposed the routing scheme in [56] and lookup method in [57] to enhance content access time and routing scalability but faced higher latency than the default ICN routing method. A probabilistic caching and forwarding scheme [49] is introduced and compared the proposed method with [50,51] to enhance the caching and forwarding performance in terms of a cache hit, latency, and interest overhead. However, the proposed method covers only a small area and still faces redundancy issues when data is retrieved from the producer node. Therefore, according to the above discussion, an efficient caching and forwarding scheme is required to mitigate performance bottlenecks. We observe that the combined functionalities of ICN and SDN play a key role in solving and improving the caching and routing performance. Furthermore, the key summary of comparison approaches is discussed in Table 1.

3. System model

In this section, we discuss the network model, communication model, problem statements and formulation.

Table 1
Comparison of existing approaches.

Ref	Cache behavior	Caching strategy	Replacement strategy	Forwarding strategy	Data retrieval	Softwarization	Remarks
[49]	Off-path	LCE	LFU	Default	Chunks	✓	- Cover-small area; - Content redundancy, - Less cache utilization.
[50]	On-path	Popularity	LRU	CPT	Packets	✗	- Ignore local node popularity, - Less caching when increase network size.
[51]	On-path	HPC	LRU	Default	Chunks	✗	- Lower caching rate(w.r.t subscribers); - No content distinction, - Increases cache weight, - Increase content delay.
[52]	Not discussed	LRU	Popularity	Bloom filter	Packets	✓	- Extra storage space in need for additional component; - May failure to detect interest duplication, - Not discussed method, if PIT overflow, - Required extra computing cost resulting increase overhead.
This work	On and Off path	CACC	Popularity and time-aware	PIT	Chunks	✓	- Support on-off-path caching ; - Improve cache hit rate, - Reduced interest overhead and Latency, - Covered Small and Large-scale area.

3.1. Network model

In this subsection, S_n is an SDN controller, and multiple ICN nodes M are assumed to optimize NDN's caching approach. We consider an undirected graph $G = \{V, E\}$, where V represents the node's set $V = \{v_1, v_2, \dots, v_m\}$, $m \in M$ and all network's nodes are connected by L links and the links set $E = \{e_1, e_2, \dots, e_l\}$, where $l \in L$ and $e_l = l_{x,y}, (x, y) \in M$ indicated the link between v_x and v_y. The publisher node P_{nd} generates the video content a to the controller node S_n for registration and the subscriber node C_{nd} sends the desired content a to the S_n to find the original content a from nearest node via flow information. The PIT entry updates after receiving the chunks of data.

$$1 \le x, y \le M, x \ne y. \tag{1}$$

According to (1) the connection link is verified with $l_{x,y} \in \{0, 1\}$. Moreover, We denote the numbers of video content by n_r where $|n_r|$ is the total of the subscriber's request. Each request is represented with $n_r(\mathrm{a}) \in N_r$. The definitions of key notations are summarized in Table 2.

3.2. Communication model

We design a named flow-based communication model in our proposed work, shown in Fig. 1. Generally, we divide it into four parts: (i) SDN controller, (ii) multiple ICN-based nodes, (iii) users (subscriber), and (iv) content provider (publisher). The publisher P_{nd} shares the content information with controller S_n for content registration, and the subscriber C_{nd} generates the content request for desired data. After requesting the content, the S_n finds the shortest path between the content requester node and the content provider node. Moreover, S_n sends the content flow information with chunk size to the PIT, and time-interval I_t is set for each entry in the PIT table before creating. After getting the desired content, the ICN node M sends the caching information to the controller. The desired content's entry is removed from the PIT table when the content is transferred from the content provider node (nearest node or publisher node). According to our method, a single content delivers numerous data chunks, improving the network performance using PIT rather than FIB, enhancing cache performance, and reducing latency. Beside, the controller is divided into three main components southbound, northbound, and East–west interfaces. The information-centric OpenFlow switch is enabled in the southbound interface and acts as a forwarding plane that contains ICN-node (switch based on ICN), subscriber, and publisher nodes. The northbound act as a management plan to provide the global view and interface of the network. The East–West bound interface (EWBI) is responsible for communicating with the other apps in the distributed controller. The controller installs flow when a request from the subscriber reaches the controller and finds the shortest path. Moreover, the details of the controller are described in Fig. 2, and implementation is discussed in Fig. 1.

3.3. Problem statements and formulation

This section discusses the forwarding and caching problems in ICN off-path and on-path caching. The traditional forwarding methods face latency issues due to traffic overloading in which each content request is generated for each data resulting in increased latency. The second problem is cache utilization due to on-path caching. For example if the desired content is available close to the server node resulting less cache utilization and increase delay in the network. Therefore, we propose the VS-ICSDN approach for video streaming in ICN to overcome these issues. We formulate the latency and overhead problems. We aim to minimize latency, overhead and maximize the cache hit ratio. So, we present two formulations off-path caching in a static and dynamic environment.

3.3.1. Off-path in static environment

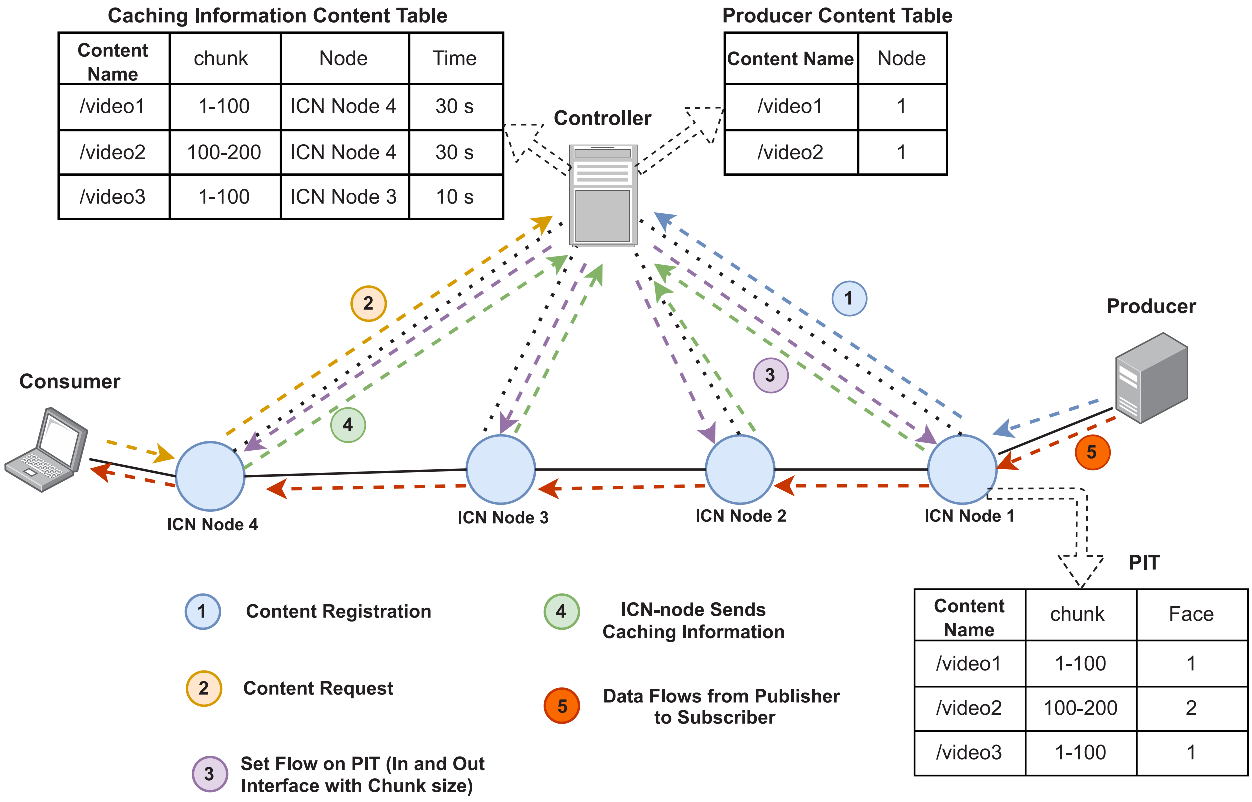
In this environment, we can get the reduced latency, represented as L_r. The requester can get their request from the nearest node or content server as follows:

$$Nr(m) = \sum_{a \in A} T_{nr}(a) \cdot C_s(a), \tag{2}$$

where Nr_m is the numbers of request from node m, $T_{nr}(a) \in |n_r|$ indicates the total numbers of request for content a, and $C_s(a)$ represent the cache state based on binary value, where content a is cached if $C_s(a) = 1$; otherwise $C_s(a) = 0$. We get the reduced latency rate L_r by

$$L_r = \sum_{m \in M} Nr(m) \cdot R(m), \tag{3}$$

where L_r is the total reduced latency of desired content between two nodes $(m, v \in M)$ and $R(m)$ indicates the reduced content delay at node m. The distance of the desired content $dis = 0$ indicates that the content is cached locally and there is no delay. Suppose the requested content is available on multiple nodes, then the content is cached from the nearest node and cached partially from different nodes as per chunks of data



Caching Information Content Table

Content Name	chunk	Node	Time
/video1	1-100	ICN Node 4	30 s
/video2	100-200	ICN Node 4	30 s
/video3	1-100	ICN Node 3	10 s

Producer Content Table

Content Name	Node
/video1	1
/video2	1

PIT

Content Name	chunk	Face
/video1	1-100	1
/video2	100-200	2
/video3	1-100	1

Fig. 1. System and communication model.

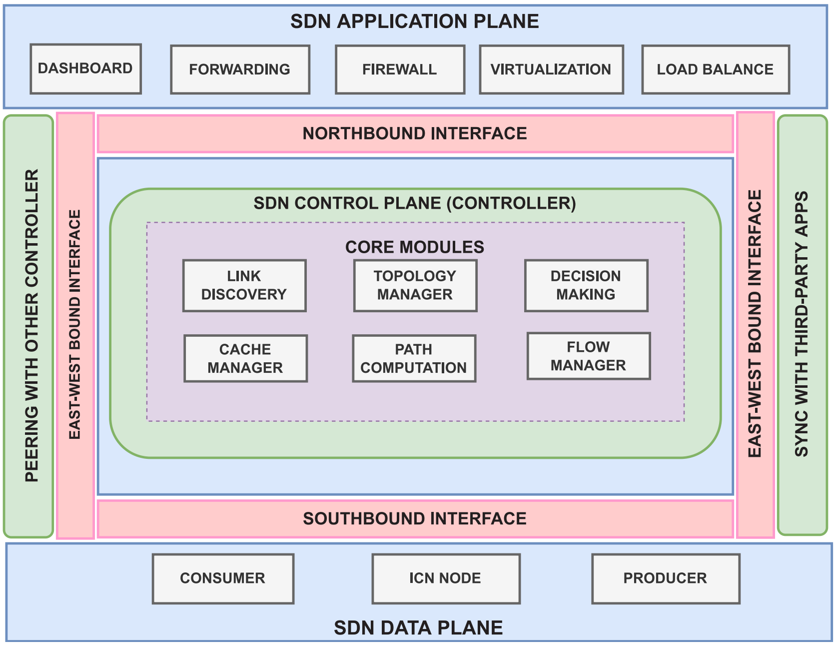


Fig. 2. Overall VS-ICSDN controller plane.

policy. The content will be partially cached from different nodes if the distance and latency rate is the same. We merge (2) and (3) into (4) as

$$L_r = \sum_{m\in M}\sum_{v\in M/m}\sum_{a\in A} a^a_{mv}\cdot T_{nr}(a).C_s(a)\cdot D_{mv}(a), \tag{4}$$

Here, a^a_{mv} indicates the value to cached the data from multiple nodes and $D_{mv}(a)$ represents the delay of video content for a. If we assume that the desired content is cached from a single node, then we formulate as

$$max\ L_r, \tag{5}$$

$$s.t. \sum_{a\in A} C_z(a)\cdot C_s(a) \le Z_a, \tag{6}$$

$$a^a_{mv}(I_t) \in \{0,1\}, \tag{7}$$

Table 2
Key notations.

Notation	Description
M	Numbers of ICN nodes
S_n	Controller node
C_l	Links cost
R_l	Forwarding links availability
$R(m)$	Reduced content delay at node (m)
a	Set of content and $\|a\|$ is total number of sets
Z_a	Cache size
C_z	Content Size
n_r	Set of request for content, and $\|n_r\|$ total number of request.
C_s	Current cache state
I_t	Time interval
τ	Ending process time
L_r	Reduced latency
C_{nd}	Subscriber node
P_{nd}	Publisher node
Ch_t	Cache table
E_t	Content entry table
Top_t	Topology table
$D_{mv}(a)$	Delay of video content for a

$$\sum_{v\in M/m} a^a_{mv} = 1. \tag{8}$$

According to (6), the content size must be less than node's cache size. Constraint (7) describes that the node v caches the content from node m by a given specific time. As per (8), a single node is guaranteed to send the desired content to the subscriber rather than multiple nodes.

3.3.2. Off-path in dynamic environment

We set the time-interval I_t to get the desired performance in the dynamic environment to get the position for new arrival content with $t-1$. So, the (6) will be reformulated as

$$\sum_{a\in A} C_z(a)\cdot C_s(a,I_t) \le Z_a. \tag{9}$$

Eq. (2) is further explicated as

$$Nr(m,I_t) = \sum_{I_t\in T}\sum_{a\in A} T_{nr}(a,I_t)\cdot C_s(a,I_t), \tag{10}$$

where $\sum_{I_t\in T}$ is the interval-time that sets the specific time to 0.5. We use (2) to obtain the reduced latency by

$$L_r(It) = \sum_{m\in M} Nr(m,I_t)\cdot D_{mv}(a,I_t), \tag{11}$$

where we obtain (12) by using (4).

$$L_r(It) = \sum_{I_t\in T}\sum_{m\in M}\sum_{v\in M/m}\sum_{a\in A} a^a_{m,v}(I_t)\cdot T_{nr}(a,I_t)\cdot C_s(a,I_t)\cdot D_{mv}(a,I_t). \tag{12}$$

Finally, the problem is formulated as.

$$max\ L_r \tag{13}$$

$$s.t. \sum_{a\in A} C_z\cdot C_s(a,I_t) \le Z_a, \tag{14}$$

$$a^a_{mv}(I_t) \in \{0,1\}, \tag{15}$$

$$\sum_{v\in M/m} a^a_{mv} = 1. \tag{16}$$

Here, $a^a_{mv}(I_t)$ represents that the desired content is retrieved between setting I_t, otherwise miss the content.

4. Proposed clean-slate flow-based approach

In this section, we discuss the proposed clean-slate flow-based approach that is divided into four sections: (i) Interest and data structure, (ii), Content registration and management, (iii) Clean-slate caching scheme, (iv) Flow-based forwarding scheme.

4.1. Interest and data structure

The default NDN's structure [58] of interest and data packet is modified to carry the flow information in the proposed system. The default interest packet is divided into five parts: (1) content-prefix, (2) sequence-no, (3) ID-next-hope, (4) flow-code, and (5) caching status. Similarly, the "ack" acknowledgment field is added to the default data packet in the current ICN structure.

4.2. Content registration and management

The controller registers the content from the intermediate node and content generator. Moreover, to ensure content caching, we designed Algorithm 1 for content registration and caching. The controller node S_n extracts the information of the prefix. After getting prefix information, two main events occur to find the content in the entry table E_t or cache table Ch_t. First, the S_n looks at the content in the cache table and finds the concerned ICN node m with its ID, then finds the shortest path via Dijkstra Algorithm to fulfill the subscriber's request. Second, if the subscriber's request exists in the E_t but not in the Ch_t then the S_n finds the nearest P_{nd} with the shortest-path to fulfill the subscriber's request. See more detail in the Algorithm 1.

4.3. Clean-slate caching scheme

We divided this section into controller-aware content caching and content replacement decisions.

4.3.1. Controller-aware content caching

In our proposed scheme controller-aware content caching (CACC), we assume that a cache size Z_a and the process $W_{n,a}$, where $W_n = \{W_{n1},\ldots.,W_{nn}\}$ is a process with content a and time interval $[I_t]$, where $W_{n,a}[I_t] = 1$. The content either requested by the subscriber or not by following:

$$W_n(I_t) := \sum_{a=1}^{n} W_{n,a}(I_t). \tag{17}$$

The unique contents are requested between $[-I_t, 0]$. Let us take the minimum interval $(-\tau_n, 0)$ in which Z_a unique contents are assigned as follow: where τ indicates the ending time of the process.

$$\tau_n = inf[(I_t) : W_n(I_t) \le Z_a]. \tag{18}$$

We get the same process of requesting content if we change the time in reverse order, where τ_n is the ending time for $W_n(I_t)$. The default cache of NDN is LRU [43] in which the less frequent content is dropped first when the new content is arrived to add in the full cache. Thus, the most frequent content remains in the existing cache. We can calculate the cache hit probability of the default NDN cache by

$$CH^L_n = \mathbb{P}^0_n[W_{n,u^0_n}(\tau_n) = 1]. \tag{19}$$

where CH^L_n presents the numbers of cache hit for content a, L indicate default cache and $\mathbb{P}^0_n$ is a Palm probability that is associated with process $n \in W_n$ and $u^0_n \in A$ [59]. We can get cache hit probability for requested content a by

$$CH^L_{n,a} = \mathbb{P}^0_{n,a}[W_{n,a}(\tau_n) = 1], \tag{20}$$

$$CH^L_n = \sum_{a=1}^{n} Pr_{n,a}[C^L_{a,b}]. \tag{21}$$

Algorithm 1: Content caching and registration process.

```
Procedure Packet FORWARD with information-path
    forall Incoming $n_r$_a do
        Read TLV (a, Prefix, $m$);
        while $m(x) = P_{nd}$ do
            Ch$_t$ ← Add.Entry (Prefix, $P_{nd}$, $I_t$);
            $P_{nd}$ ← ACK send;
        end
        while $m(x) = I_{nd}$ do
            E$_t$ ← updated cache and add entry Top_t ←
             Graph Updated (gr);
        end
        while $m(x) = C_{nd}$ do
            Ch$_t$ ← Search (a, Prefix, $P_{nd}$);
            if Search = Successful then
                Find-shortest-path ← Dijikstra (gr,x);
                $C_{nd}$ ← Send-(ACK) Generated $n_r$_a (a, $m$,
                 Prefix, P);
                $m$ and $P_{nd}$ ← Forward $n_r$_a;
            else if $a \neq$ desired content then
                PIT ← Add.Entry (a, Prefix, $C_{nd}$);
                $C_{nd}$ ← Send-(NACK);
            end
        end
    end
end
```

Here, Pr indicates probability. Finally, for the Controller-aware cache, a I_t is added with content for the scope of the content. When the content time expires, that content will be evicted from the cache. In this work, we design the cache eviction policy based on controller synchronization time-aware content cache. In our proposed scheme, we set the time value I_t with the most popular content. When the most popular content is identified, then the I_t is counted; when the I_t expires, that content will be evicted. We can get content popularity by

$$CH_n^{cl}(a) = \mathbb{P}_n^0[W_{n,u_n^0}(a) = 1], \tag{22}$$

where CH_n^{cl} illustrates the clean-slate cache hit and finally concludes by (22)

$$CH_n^{cl}(I_t) = \sum_{a=1}^{n} Pr_{n,a}[C_{n,a}^{T_l}(I_t)]. \tag{23}$$

Furthermore, we deal with the ergodic content process and independent content by

$$Z_a(I_t) := \mathbb{E}\Big(W_n(I_t)\Big), \tag{24}$$

the numbers of requested contents are satisfied with I_t, where W_a is mentioned (1), $\mathbb{E}$ represent the values of the expected outcomes, and we can calculate the hit probability of the controller time-awarded cache as

$$CH_{n,a}^{cl}(I_{t_a}), \forall a = 1, \dots \dots\dots, a. \tag{25}$$

and the following is to deal with the Poisson requests,

$$Z_a = \sum_{a=1}^{n} \left[1 - \mathrm{e}^{-\lambda_{n,a} I_{t_n}}\right]. \tag{26}$$

4.3.2. Cache replacement decision

We design Algorithm 2 for content replacement in VS-ICSDN. The content provider nodes maintain the CS table for a given timer. We design the synchronization function for getting cached content information from the providing node with a controller for content registration. The I_t is set with cached content, and the content is removed when I_t reaches 0. Furthermore, the decision of content replacement is based on the number of data chunk sizes. Moreover, we deal with the most popular content to cached. We replaced the content in data chunks, increasing cache locality and reducing latency and computation cost.

Algorithm 2: Content replacement decision.

```
Function Procedure content replacement
    $Initialize$ :
    $a$ : $Desired\ content$
    $m$ : $ICN\ node$
    $I_t$ : $Time\ interval$
    for each $m$ has content $a$ do
        if $m$ =Node-id and $I_t \neq = 0$ then
            start schedule ← (a, $m$, $I_t$) ;
            break;
        end
        if $m$ =Node-id and $I_t = 0$ then
            Deleted content from $CS_t$;
            start replacement function ← popular(a); //
             Deal with most popular content
        end
    end
```

4.4. Flow-based forwarding scheme

Our proposed method aims to reduce latency and maximize the cache hit ratio. In this regard, we design the flow-based forwarding strategy in which we choose the shortest path of caching node based on links cost. Beside, we implement the content forwarding strategy on PIT using Algorithm 3 and discussed in Fig. 3. To populate the PIT entry, the S_n sends the content packet with flow information in chunks. Then ICN node gets the flow information and installs the PIT's entries on its table with prefix and face-id. After that, it waits for data from either ICN node m or the subscriber node C_{nd}. Moreover, the S_n guide instructions to C_{nd} for content caching. Furthermore, we set the fixed entry expiration time to generate data from the caching node to the subscriber node. Suppose CD_m is the content demand flow in m node and $CD_m = \{f_1, f_2, \dots \dots f_m\}$ is the traffic flow demand for $m \in m$. Then the content distribution Δ_m can be modeled as:

$$\{f^{p1}{}_{\mathrm{w1}}, f^{\mathrm{p2}}{}_{\mathrm{w2}}, \dots, f^{\mathrm{pn}}{}_{\mathrm{wn}}\} = \Delta_k \times CD_m \quad \forall_m \in M, \tag{27}$$

where $f^{p1}{}_{\mathrm{w}}$ represent the rate flow of w which is sent by node m via path p. Furthermore, to assure the content flow demand CD that should be less than link capacity C_l within the given time t.

$$\sum_{p \in f(l)} f^p \leq C_l \quad \forall l \in L. \tag{28}$$

In (28), we can get the traffic flow content for each node in the network. Moreover, we designed a new pattern for interest, in which one interest

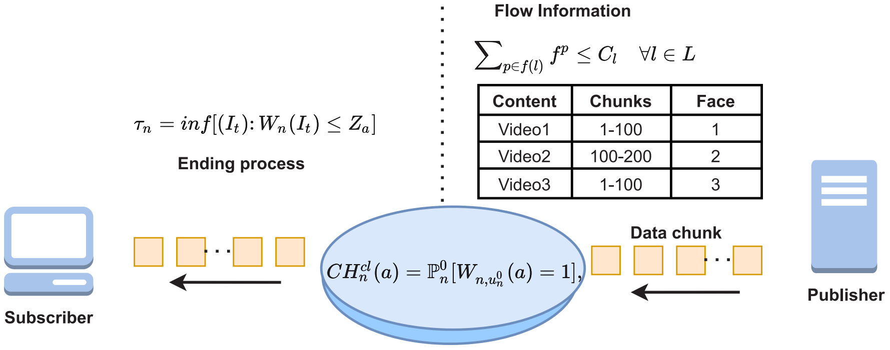


Fig. 3. PIT entry process.

is sent to controller node S_n for desired content instead of each interest for each data. The S_n generates flow information to other nodes, and the chunks of data are delivered to the Pending Interest Table (PIT) discussed in Algorithm 2 and 3. Thus one interest is enough to deliver the numbers of data chunks, and we enhanced the utilization of PIT entry.

Algorithm 3: Content forwarding and lookup.

```
Function FACE SELECTION and UPDATE(PIT_Entry)
    forall Incoming $n_r$_a do
        Read Packet-Structure ($m$, Prefix, P);
        if Entry = New then
            PIT ← Insert ($m$, Prefix, In.Face);
            $S_n$ ← Forward-(ACK);
            else if Entry ≠ New then
                PIT ← Updated Entry;
                $S_n$ ← Forward-(NACK);
            end
        end
    end
    while Forwarding Interest_Pkt do
        $S_n$ ← Forward-Interfaces (In or Out);
        $S_n$ ← Forward content-caching_Information;
    end
    while Receiving Data do
        Data (a, Sig); Verify data
        PIT ← Search (a, interfaces );
        Next-node ($m$ or $C_{nd}$) ← Forward to requester;
    end
```

5. Performance evaluation

In this section, we divided the performance evaluation into two scenarios. We consider the small-scale area in scenario-1, in which we compare the proposed VS-ICSDN with the baselines strategies SRP [49], DPCP [50], and HCP [51]. Scenario-2 is designed for large-scale networks to check our proposed method's efficiency and compare it with the existing approach FCR-NS [52]. We used different performance metrics such as cache hit ratio in percentage %, latency millisecond (ms), and overhead in megabytes (MB). The extensive analysis is examined under different network scenarios with network size, cache size, and payload size.

Table 3
Simulation parameters.

Parameter description		Value
Scenario -1	Interest rate	10/s
	Subscribers	10–50
	ICN nodes	2
	Publisher	5% of total nodes
	Cache replacement	Proposed scheme
	Data payload	128–512 (Bytes)
	Caching capacity	50–150 (MB)
	Topology	Hierarchical (2-AS)
	Simulation time	100-(s)
Scenario -2	Interest rate	10/s
	Subscribers	150–1000
	ICN nodes	5% of total nodes
	Publisher	15% of total nodes
	Cache replacement	Proposed scheme
	Data payload	256 (Bytes)
	Caching capacity	50–100 (MB)
	Topology	Hierarchical (3-AS)
	Simulation time	180 (s)
	Name prefix	Same as [52]

5.1. Simulation setup and implementation

We use ndnSIM-2.x [60] (based on NS-3) to execute simulation-based experiments to evaluate the feasibility and technical soundness of our solution. We design customized applications (subscriber, publisher, off-switch, and controller) and change the default interest and data packet according to our proposed scheme. Besides, we employ the NDN default topology generator to design the network topology hierarchically. This study uses two scenarios for small and large-scale environments. Scenario 1 consists of two Autonomous Systems (ASs) [49] that connect to the controller, and each AS maintains 25 ICN nodes and 5% publisher of total nodes in the network. Scenario −2 consists of three ASs, each consisting of 5% of ICN-nodes, 150–1000 subscriber nodes, and 15% of publisher nodes in the entire network. Moreover, the publisher node sends the information of newly generated content to the controller. Meanwhile, the controller installs flows and sets the PIT entry on each ICN-node to reduce the overhead and latency, respectively. The controller communicates with the bottom-level nodes to get the caching content information for future requests. Furthermore, 4 shows our designed topology, and the details of our simulation parameters are shown in Table 3.

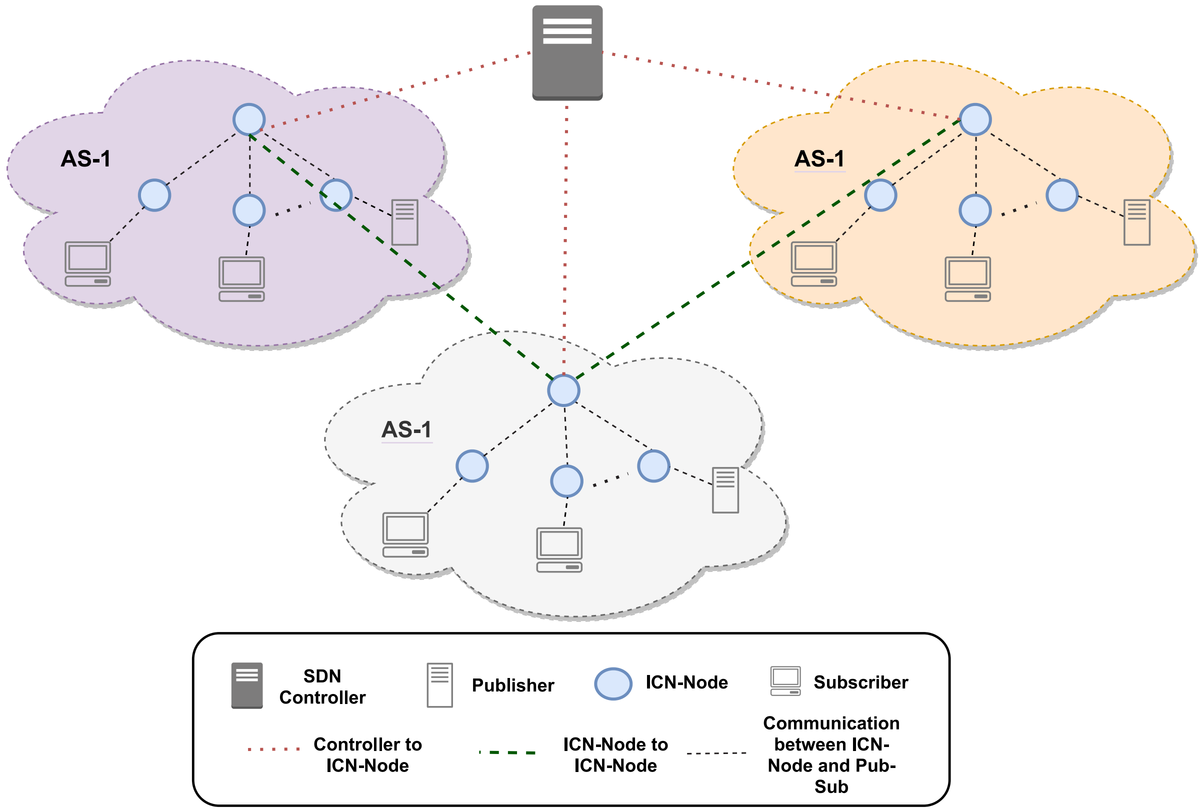
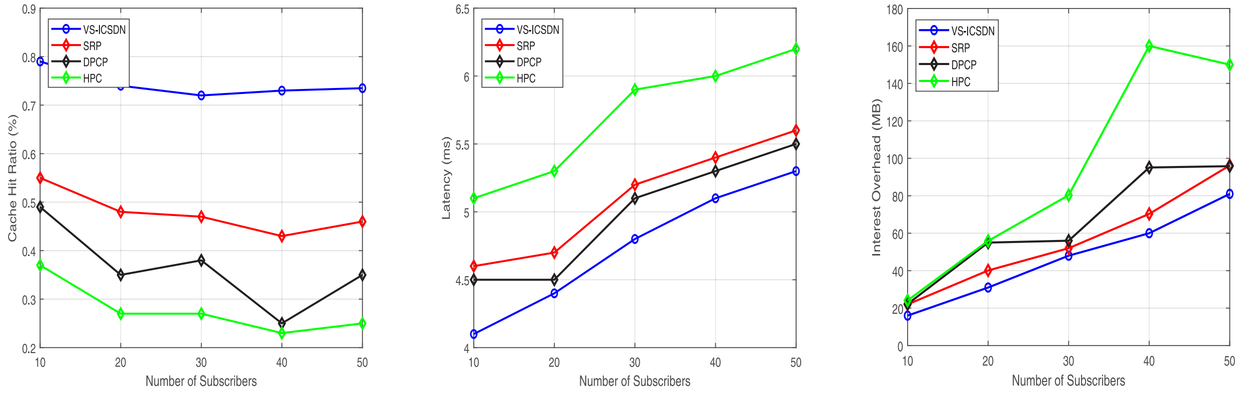


Fig. 4. Simulation scenario.



(a) (b) (c)

Fig. 5. Performance comparison against increasing number of subscribers: (a) cache hit ratio, (b) latency, and (c) interest overhead.

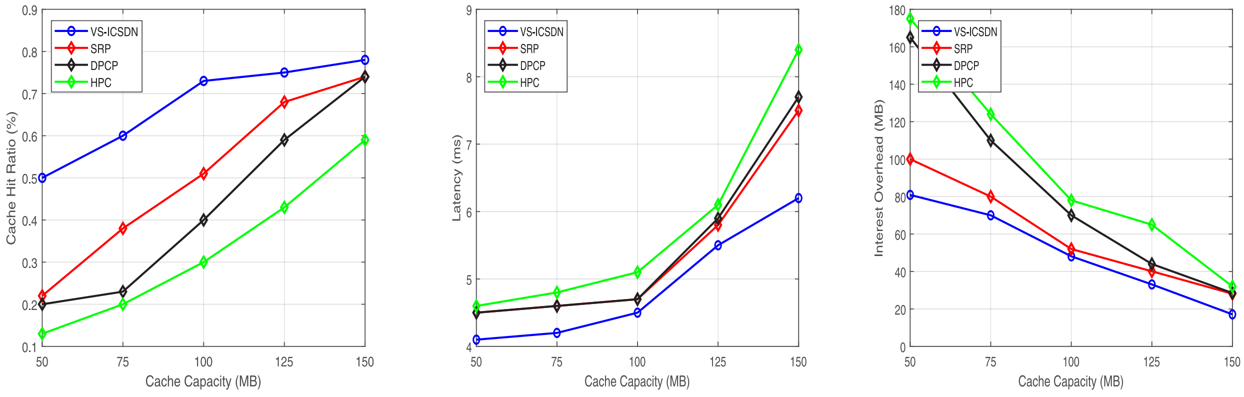
5.2. Experimental results based on scenario -1

5.2.1. Effect of network size

In this experiment, we implement our proposed strategy on the small-scale network. We use 256 payload size in bytes, the same as used in baseline strategies [49]. To assess the performance of our proposed study with state-of-the-art strategies, we change the number of subscribers from 10 to 50, the interest rate is 10 per second, and simulation time is 100 s. Moreover, the details of parameters are discussed in Table 3. The overall effect of network size is shown in Fig. 5. Our scheme is better than other methods and maintains the performance even if we increase the number of subscribers.

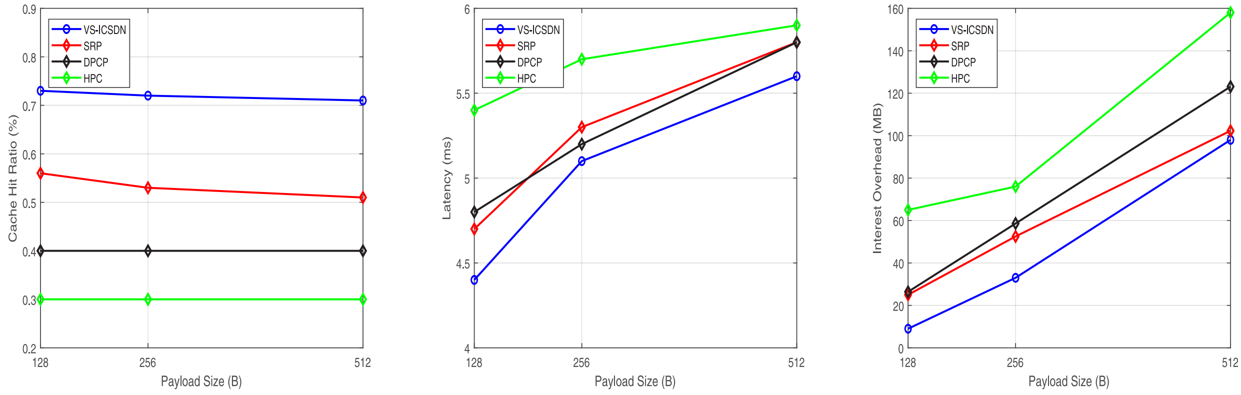
Fig. 5(a) shows the cache hit performance of our proposed scheme with other baseline methods. To verify the impact of network size, we gradually increase the number of subscribers. As a result, we notice that VS-ICSDN's performance improves due to the improvement in the PIT performance with maximum utilization of the PIT entries.

The utilization of PIT overcomes the latency issue and increases the overall performance of the whole network. Fig. 5(b) shows the latency performance of the VS-ICSDN scheme with other state-of-art strategies. The results show that some desired contents are not cached from nearby nodes due to the high volume of requests generated by the highly dynamic environment and diversity of content. However, VS-ICSDN achieves performance with low latency than other methods. Fig. 5(c) shows that the overhead rate of the baseline strategies is very high compared to our proposed scheme. Although we increase the number of subscribers to affect the network size for the proposed scheme, the VS-ICSDN still maintains the performance position due to a higher cache hit ratio.



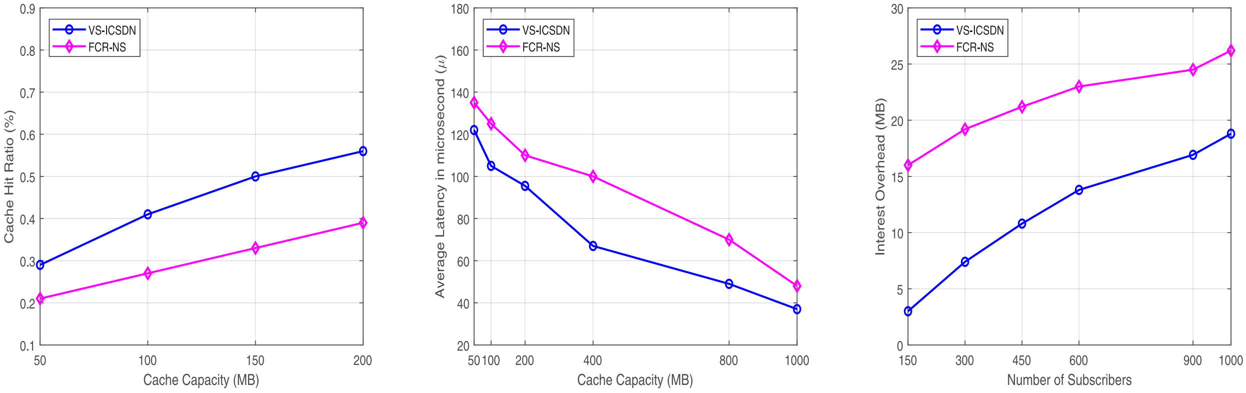
(a) (b) (c)

Fig. 6. Performance comparison against cache capacity: (a) cache hit ratio, (b) latency, and (c) Interest overhead.



(a) (b) (c)

Fig. 7. Performance comparison against different payload sizes: (a) cache hit ratio, (b) latency, and (c) Interest overhead.



(a) (b) (c)

Fig. 8. Performance comparison (scenario-2): (a) cache hit ratio, (b) latency, and (c) Interest overhead.

5.2.2. *Effect of cache capacity*

In this subsection, we used different cache capacities from 50 MB to 150 MB to evaluate the performance of other methods. In Fig. 6(a), the VS-ICSDN cache hit ratio performance is analyzed against various cache capacities with other existing strategies. The performance of SRP is much more reasonable than others, but the VS-ICSDN is superior in the cache hit performance due to better management of the cache. Moreover, the cache hit ratio reduces with the decrease of cache capacity. The latency performance with other existing methods is demonstrated in Fig. 6(b). Due to limited cache capacity, we noticed that the latency increases when the nearby node does not cache the content with the given time slot. However, the efficiency of VS-ICSDN is better than other existing studies. Fig. 6(c) shows that the overhead rate increases with the change in the cache capacity. Our proposed VS-ICSDN scheme gets lower overhead due to a good forwarding scheme with a single interest that gets many chunks of data compared to baseline strategies.

5.2.3. *Effect of payload size*

To check the effect of payload size on our proposed VS-ICSDN scheme with other existing strategies, we apply different payload sizes from 128 to 512. Our proposed scheme is superior to other baseline schemes, as depicted in Fig. 7(a). The cache hit ratio performance is analyzed against various payload sizes with other strategies. The nearest node achieves the desired content due to a proposed caching and forwarding strategy. Fig. 7(b) illustrates the latency performance with existing strategies. When the payload size is increased, the latency gradually increases with payload sizes. We observe that the other existing works manage the latency better than HPC; it could not maintain the latency due to poor caching and content forwarding schemes. However, the proposed strategy maintains the position of latency performance due to good management of caching data resulting in low latency. We increase the payload size to verify the overhead performance. Fig. 7(c) shows that the proposed VS-ICSDN method is better than other methods due to reduced round-trip time and getting the data in chunks.

5.3. *Experimental results based on scenario -2*

In order to check the scalability of our proposed method under large traffic, we consider scenario 2. In this evaluation, we use 150–1000 subscribers and compare our work with FCR-NS [52]. Moreover, Table 3 and Section 5.1 discuss the details of simulation parameters and settings.

Fig. 8(a) represents the cache hit ratio concerning the change in cache size. We use cache sizes 50 to 1000 (MB), 150 subscribers, and a simulation time of 180 s, the same as mentioned in the existing approach FCRN-NS. The result shows that our proposed strategy achieves performance with a maximum cache hit ratio compared to the existing study due to a controller-aware caching scheme based on popular content. Fig. 8(b) shows the latency performance with the baseline approach, and we notice that the performance of VS-ICSDN is better due to the PIT approach rather than FIB, and our proposed controller aware caching scheme in which the controller knows the nearest publisher to provides the subscriber content with minimum hops. Fig. 8(c) represents the rate of interest overhead. In this experiment, we use 150 to 1000 subscribers, cache size 800 (MB), and the details of other parameters are described in Table 3. The results show that our proposed scheme VS-ICSDN maintains the position against the FCR-NS approach and minimizes the interest overhead due to getting the data in chunks and maximum utilization of PIT entry.

6. Conclusion

This paper presents a caching and forwarding scheme for video streaming applications over the ICN-based pub/sub communication model. A softwarized controller is initialized in a clean-slate information-centric network. By adopting the single-request-multiple-flows and on-off-path caching principle, we facilitate PITs instead of FIBs to minimize the round-trip time and bandwidth while improving the solution's scalability. Experimental results from different cache capacity, network, and payload sizes show that the cache hit ratio, latency, overhead and other parameters favor the proposed solution compared to state-of-the-art methods. In the future, we plan to improve the existing scheme through distributed SDN controllers to take advantage of caching, computing, and networking capabilities of both ICN-based edge and cloud nodes. Besides, we plan to integrate machine learning approaches like deep reinforcement learning into the controller to improve the data-centric synchronization among the participating nodes.

CRediT authorship contribution statement

M. Wasim Abbas Ashraf: Conceptualization, Acquisition of data, Analysis and/or interpretation of data, Writing – original draft, Revising the manuscript critically for important intellectual content. **Chuanhe Huang:** Conceptualization, Acquisition of data, Writing – original draft. **Arif Raza:** Acquisition of data, Analysis and/or interpretation of data, Writing – original draft, Revising the manuscript critically for important intellectual content. **Kashif Sharif:** Acquisition of data, Revising the manuscript critically for important intellectual content. **Md Monjurul Karim:** Analysis and/or interpretation of data, Revising the manuscript critically for important intellectual content. **Shidong Huang:** Analysis and/or interpretation of data, Revising the manuscript critically for important intellectual content.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgment

All authors approved the final version of the manuscript. This work was supported by the National Natural Science Foundation of China (No. 61772385).

References

[1] I. Dasgupta, S. Shannigrahi, M. Zink, A hybrid NDN-IP architecture for live video streaming: A QoE analysis, in: 2021 IEEE International Symposium on Multimedia (ISM), IEEE, 2021, pp. 148–157.

[2] D.R. Chowdhury, S. Nandi, D. Goswami, Video streaming over IoV using IP multicast, J. Netw. Comput. Appl. 197 (2022) 103259, http://dx.doi.org/10.1016/j.jnca.2021.103259.

[3] I.U. Din, S. Hassan, M.K. Khan, M. Guizani, O. Ghazali, A. Habbal, Caching in information-centric networking: Strategies, challenges, and future research directions, IEEE Commun. Surv. Tutor. 20 (2) (2018) 1443–1474, http://dx.doi.org/10.1109/COMST.2017.2787609.

[4] B. Rainer, D. Posch, H. Hellwagner, Investigating the performance of pull-based dynamic adaptive streaming in NDN, IEEE J. Sel. Areas Commun. 34 (8) (2016) 2130–2140, http://dx.doi.org/10.1109/JSAC.2016.2577365.

[5] I. Livaja, K. Pripuzic, S. Sovilj, M. Vukovic, A distributed geospatial publish/subscribe system on Apache Spark, Future Gener. Comput. Syst. 132 (2022) 282–298, http://dx.doi.org/10.1016/j.future.2022.02.013.

[6] L. Han, J. Zhao, X. Bao, G. Liu, Y. Liu, T. Maksymyuk, An adaptive video transmission mechanism over MEC-based content-centric networks, Wirel. Commun. Mob. Comput. 2021 (2021) 9968550:1–9968550:10, http://dx.doi.org/10.1155/2021/9968550.

[7] Y. Ye, B. Lee, Y. Qiao, Hop-by-hop congestion measurement and practical active queue management in NDN, in: IEEE Global Communications Conference, GLOBECOM 2020, Virtual Event, Taiwan, December 7-11, 2020, IEEE, 2020, pp. 1–6, http://dx.doi.org/10.1109/GLOBECOM42002.2020.9322585.

[8] I. Dasgupta, S. Shannigrahi, M. Zink, A hybrid NDN-IP architecture for live video streaming: From host-based to content-based delivery to improve QoE, Int. J. Semant. Comput. (2022) 1–25.

[9] F. Wu, W. Yang, J. Ren, F. Lyu, X. Ding, Y. Zhang, Adaptive video streaming using dynamic NDN multicast in WLAN, in: 39th IEEE Conference on Computer Communications, INFOCOM Workshops 2020, Toronto, on, Canada, July 6-9, 2020, IEEE, 2020, pp. 97–102, http://dx.doi.org/10.1109/INFOCOMWKSHPS50562.2020.9162662.

[10] S. Yang, J. Hu, K. Jiang, H. Xiao, M. Wang, Hybrid-360: An adaptive bitrate algorithm for tile-based 360 video streaming, Trans. Emerg. Telecommun. Technol. 33 (4) (2022) http://dx.doi.org/10.1002/ett.4430.

[11] J.O. Murillo, A.F. Skarmeta, SDN enabled information centric networking (ICN) as a service prefetching mechanism for HyperText transfer protocol (HTTP) based services. The scalable video streaming case, 2020, CoRR, abs/2005.07392 arXiv:2005.07392 URL https://arxiv.org/abs/2005.07392.

[12] L. Mamushiane, J. Mwangama, A.A. Lysko, Given a SDN topology, how many controllers are needed and where should they go? in: 2018 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN), 2018, pp. 1–6, http://dx.doi.org/10.1109/NFV-SDN.2018.8725710.

[13] D. Hasan, M. Othman, Efficient topology discovery in software defined networks: Revisited, Procedia Comput. Sci. 116 (2017) 539–547, http://dx.doi.org/10.1016/j.procs.2017.10.051.

[14] S.R. Zahedi, S. Jamali, P. Bayat, A power-efficient and performance-aware online virtual network function placement in SDN/NFV-enabled networks, Comput. Netw. 205 (2022) 108753, http://dx.doi.org/10.1016/j.comnet.2021.108753.

[15] S. Zhang, X. Wang, J. Lv, M. Huang, Routing and content delivery for in-network caching enabled IP network, Multimedia Tools Appl. 81 (1) (2022) 715–735.

[16] H.L. Mai, M. Aouadj, G. Doyen, W. Mallouli, E.M. de Oca, O. Festor, Toward content-oriented orchestration: SDN and NFV as enabling technologies for NDN, in: J. Betser, C.J. Fung, A. Clemm, J. François, S. Ata (Eds.), IFIP/IEEE International Symposium on Integrated Network Management, IM 2019, Washington, DC, USA, April 09-11, 2019, IFIP, 2019, pp. 594–598, URL https://ieeexplore.ieee.org/document/8717804.

[17] I. Abdullahi, A.S.M. Arif, S. Hassan, Survey on caching approaches in information centric networking, J. Netw. Comput. Appl. 56 (2015) 48–59, http://dx.doi.org/10.1016/j.jnca.2015.06.011.

[18] X. Hu, S. Zheng, G. Zhang, L. Zhao, G. Cheng, J. Gong, R. Li, An on-demand off-path cache exploration based multipath forwarding strategy, Comput. Netw. 166 (2020) http://dx.doi.org/10.1016/j.comnet.2019.107032.

[19] E.T.d. Silva, J.M.H.d. Macedo, A.L.D. Costa, NDN content store and caching policies: Performance evaluation, Computers 11 (3) (2022) 37.

[20] Y. Zhang, X. Tan, W. Li, PPC: Popularity prediction caching in ICN, IEEE Commun. Lett. 22 (1) (2018) 5–8, http://dx.doi.org/10.1109/LCOMM.2017.2731312.

[21] S. Khan, F. Luo, Z. Zhang, M.A. Rahim, M. Ahmad, K. Wu, Survey on issues and recent advances in vehicular public-key infrastructure (VPKI), IEEE Commun. Surv. Tutor. 24 (3) (2022) 1574–1601, http://dx.doi.org/10.1109/COMST.2022.3178081.

[22] Y. Li, T. Zhang, X. Xu, Z. Zeng, Y. Liu, Content popularity and node level matched based probability caching for content centric networks, in: 2016 IEEE/CIC International Conference on Communications in China (ICCC), 2016, pp. 1–6, http://dx.doi.org/10.1109/ICCChina.2016.7636877.

[23] M. Garetto, E. Leonardi, G. Neglia, Content placement in networks of similarity caches, Comput. Netw. 201 (2021) 108570, http://dx.doi.org/10.1016/j.comnet.2021.108570.

[24] Y. Liu, T. Zhi, H. Zhou, H. Xi, PBRS: A content popularity and betweenness based cache replacement scheme in ICN-IoT, J. Internet Technol. 22 (7) (2021) 1495–1508.

[25] C. Bernardini, T. Silverston, O. Festor, MPC: Popularity-based caching strategy for content centric networks, in: 2013 IEEE International Conference on Communications (ICC), 2013, pp. 3619–3623, http://dx.doi.org/10.1109/ICC.2013.6655114.

[26] H. Okada, M. Yoshimi, C. Wu, T. Yoshinaga, A failsoft scheme for mobile live streaming by scalable video coding, IEICE Trans. Inf. Syst. 104-D (12) (2021) 2121–2130, http://dx.doi.org/10.1587/transinf.2021pap0015.

[27] N. Kan, J. Zou, C. Li, W. Dai, H. Xiong, RAPT360: Reinforcement learning-based rate adaptation for 360-Degree video streaming with adaptive prediction and tiling, IEEE Trans. Circuits Syst. Video Technol. 32 (3) (2022) 1607–1623, http://dx.doi.org/10.1109/TCSVT.2021.3076585.

[28] H.A. Nekrasov, A.Y. Romanov, Distributed load balancing for the adaptive video streaming using CDN with ring system of server consolidation by circulant topology, J. Phys. Conf. Ser. 1210 (2019) 012105, http://dx.doi.org/10.1088/1742-6596/1210/1/012105.

[29] N. Dimokas, D. Katsaros, Detecting energy-efficient central nodes for cooperative caching in wireless sensor networks, in: L. Barolli, F. Xhafa, M. Takizawa, T. Enokido, H. Hsu (Eds.), 27th IEEE International Conference on Advanced Information Networking and Applications, AINA 2013, Barcelona, Spain, March 25-28, 2013, IEEE Computer Society, 2013, pp. 484–491, http://dx.doi.org/10.1109/AINA.2013.120.

[30] Z. Zhang, C.-H. Lung, I. Lambadaris, M. St-Hilaire, IoT data lifetime-based cooperative caching scheme for ICN-IoT networks, in: 2018 IEEE International Conference on Communications (ICC), 2018, pp. 1–7, http://dx.doi.org/10.1109/ICC.2018.8422100.

[31] M.W. Ashraf, C. Huang, A.R. Khuhawar, S. Huang, Y. Yin, D.-F. Wu, Dynamic cooperative cache management scheme based on social and popular data in vehicular named data network, Wirel. Commun. Mob. Comput. 2022 (2022) 1–11, http://dx.doi.org/10.1155/2022/8374181.

[32] D. Wu, B. Liu, Q. Yang, R. Wang, Social-aware cooperative caching mechanism in mobile social networks, J. Netw. Comput. Appl. 149 (2020) http://dx.doi.org/10.1016/j.jnca.2019.102457.

[33] M. Brehob, S. Wagner, E. Torng, R.J. Enbody, Optimal replacement is NP-hard for nonstandard caches, IEEE Trans. Comput. 53 (1) (2004) 73–76, http://dx.doi.org/10.1109/TC.2004.1255792.

[34] B.K. Maddali, P. Lorenz, Cognitive radio enabled cache map-and-route using context mapping and decision making approach in software defined networks, IEEE Trans. Veh. Technol. 68 (6) (2019) 5849–5858, http://dx.doi.org/10.1109/TVT.2019.2903222.

[35] Y. Hayamizu, A. Shibuya, M. Yamamoto, The combination effect of cache decision and off-path cache routing in content oriented networks, IEICE Trans. Commun. 102-B (5) (2019) 1010–1018, http://dx.doi.org/10.1587/transcom.2018EBP3170.

[36] W. Zhu, X. Zeng, Decision tree-based adaptive reconfigurable cache scheme, Algorithms 14 (6) (2021) 176, http://dx.doi.org/10.3390/a14060176.

[37] N. Zhou, J.Y.B. Lee, CAPTON - centralized all-path transmission for over-subscribed datacenter networks, IEEE Access 10 (2022) 8208–8218, http://dx.doi.org/10.1109/ACCESS.2022.3143160.

[38] M. Abiri, M. Mehrjoo, M. Rezaei, Scalable video traffic offloading for streaming services in 5G HetNets, Multimedia Tools Appl. 81 (9) (2022) 12325–12347, http://dx.doi.org/10.1007/s11042-021-11312-1.

[39] H. Yoon, J. Han, Content-based video retrieval with prototypes of deep features, IEEE Access 10 (2022) 30730–30742, http://dx.doi.org/10.1109/ACCESS.2022.3160214.

[40] L. Gobatto, M. Saquetti, C.M. Diniz, B. Zatt, W. Cordeiro, J.R. Azambuja, Improving content-aware video streaming in congested networks with in-network computing, 2022, CoRR, abs/2202.04703 arXiv:2202.04703 URL https://arxiv.org/abs/2202.04703.

[41] B. Alaya, L. Sellami, Multilayer video encoding for QoS managing of video streaming in VANET environment, ACM Trans. Multimedia Comput. Commun. Appl. 18 (3) (2022) 82:1–82:19, http://dx.doi.org/10.1145/3491433.

[42] N. Jiang, M.C. Vuran, S. Wei, L. Xu, Qos-aware network energy optimization for danmu video streaming in WiFi networks, in: 29th IEEE/ACM International Symposium on Quality of Service, IWQOS 2021, Tokyo, Japan, June 25-28, 2021, IEEE, 2021, pp. 1–10, http://dx.doi.org/10.1109/IWQOS52092.2021.9521338.

[43] Y. Wang, V. Friderikos, Energy-efficient proactive caching with multipath routing, 2021, CoRR, abs/2104.13493 arXiv:2104.13493 URL https://arxiv.org/abs/2104.13493.

[44] J.N.R. Araujo, L.S. Batista, C.C. Monteiro, Improving proactive routing with a multicriteria and adaptive framework in ad-hoc wireless networks, Wirel. Netw. 26 (6) (2020) 4595–4614, http://dx.doi.org/10.1007/s11276-020-02366-4.

[45] G. Zhang, C. Wu, Q. Gao, Exploiting layer and spatial correlations to enhance SVC and tile based 360-degree video streaming, Comput. Netw. 191 (2021) 107985, http://dx.doi.org/10.1016/j.comnet.2021.107985.

[46] A. Malik, B. Aziz, M. Adda, C. Ke, Smart routing: Towards proactive fault-handling in software-defined networks, 2019, CoRR, arXiv:1904.00717 arXiv:1904.00717 URL http://arxiv.org/abs/1904.00717.

[47] S. Mastorakis, A. Afanasyev, L. Zhang, On the evolution of ndnSIM: an open-source simulator for NDN experimentation, ACM Comput. Commun. Rev. (2017).

[48] H. Hu, Y. Jin, Y. Wen, C. Westphal, Orchestrating caching, transcoding and request routing for adaptive video streaming over ICN, ACM Trans. Multimedia Comput. Commun. Appl. 15 (1) (2019) 24:1–24:23, http://dx.doi.org/10.1145/3289184.

[49] S. Asif Iqbal, Asaduzzaman, M. Hoque, A source-driven probabilistic forwarding and caching strategy in NDN and SDN-based NDN, Int. J. Commun. Syst. 35 (2022) http://dx.doi.org/10.1002/dac.5093.

[50] M. Yu, R. Li, Dynamic popularity-based caching permission strategy for named data networking, in: W. Shen, J. Luo, J.A. Barthès, F. Dong, J. Zhang, H. Zhu (Eds.), 22nd IEEE International Conference on Computer Supported Cooperative Work in Design, CSCWD 2018, Nanjing, China, May 9-11, 2018, IEEE, 2018, pp. 576–581, http://dx.doi.org/10.1109/CSCWD.2018.8465140.

[51] Y. Wang, M. Xu, Z. Feng, Hop-based probabilistic caching for information-centric networks, in: 2013 IEEE Global Communications Conference, GLOBECOM 2013, Atlanta, GA, USA, December 9-13, 2013, IEEE, 2013, pp. 2102–2107, http://dx.doi.org/10.1109/GLOCOM.2013.6831385.

[52] A. Kalghoum, L.A. Saïdane, FCR-NS: a novel caching and forwarding strategy for named data networking based on software defined networking, Clust. Comput. 22 (3) (2019) 981–994, http://dx.doi.org/10.1007/s10586-018-02887-w.

[53] K. Delvadia, N. Dutta, R. Jadeja, CCJRF-ICN: A novel mechanism for coadjuvant caching joint request forwarding in information centric networks, IEEE Access 9 (2021) 84134–84155, http://dx.doi.org/10.1109/ACCESS.2021.3087558.

[54] M. Zhang, J. Luo, L. Zhang, X. Yu, T. Xu, K. Lei, Comparative analysis of probabilistic forwarding strategies in ICN for edge computing, Peer-to-Peer Netw. Appl. 14 (6) (2021) 4014–4030, http://dx.doi.org/10.1007/s12083-021-01219-x.

[55] K.A. Raza, A. Asheralieva, M.M. Karim, K. Sharif, M. Gheisari, S. Khan, A novel forwarding and caching scheme for information-centric software-defined networks, in: International Symposium on Networks, Computers and Communications, ISNCC 2021, Dubai, United Arab Emirates, October 31 - November 2, 2021, IEEE, 2021, pp. 1–8, http://dx.doi.org/10.1109/ISNCC52172.2021.9615667.

[56] H. Jung, S. Kim, A multiple hash routing scheme for fast data retrieval in ICN, in: International Conference on Information and Communication Technology Convergence, ICTC 2018, Jeju Island, Korea (South), October 17-19, 2018, IEEE, 2018, pp. 1562–1565, http://dx.doi.org/10.1109/ICTC.2018.8539379.

[57] M.W. Ashraf, C. Huang, S. Khalid, A. Rana, M. Ahmad, U. Raza, Dynamic naming scheme and lookup method based on trie for vehicular named data network, Wirel. Commun. Mob. Comput. 2022 (2022) 1–13, http://dx.doi.org/10.1155/2022/6539532.

[58] N.S. Foundation, NDN Packet Format Specification v0.3, 2022, URL https://named-data.net/doc/NDN-packet-spec/current.

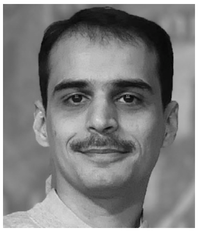
[59] P. Brémaud, Palm probability, 2020, pp. 639–675, http://dx.doi.org/10.1007/978-3-030-40183-2_17.

[60] S. Mastorakis, A. Afanasyev, L. Zhang, On the evolution of ndnSIM: an open-source simulator for NDN experimentation, Comput. Commun. Rev. 47 (3) (2017) 19–33, http://dx.doi.org/10.1145/3138808.3138812.

M. Wasim Abbas Ashraf received the master's degree in computer science from the University of Agriculture Faisalabad, Pakistan, in 2016. He is currently pursuing the Ph.D. degree with the School of Com puter Science at Wuhan University, China. His research interests include, Information-Centric Networks, caching, vehicular networks and Federated and Transfer Learning.

Chuanhe Huang received his B.Sc., M.Sc., and Ph.D. degrees, all in Computer Science, from Wuhan University, Wuhan, China, in 1985, 1988, and 2002, respectively. He is currently a Professor at the School of Computer Science, Wuhan University. His research interests include computer networks, VANETs, Internet of Things, and Distributed Computing.

Khuhawar Arif Raza received his MS degree in computer science form Beijing Institute of Technology (BIT), China in 2020. He served as research assistant in Southern University of Science and Technology (SUSTech). He is currently pursuing Ph.D. degree at College of Computer Science and Software Engineering Shenzhen University Shenzhen, China. His research interests include ICN , Intelligent resource optimization at edge, Transfer and Federated learning.



Kashif Sharif [M'08] received his M.S. degree in information technology in 2004 from National University of Sciences and Technology, Pakistan, and Ph.D. degree in computing and informatics from University of North Carolina at Charlotte, NC, USA in 2012. He is currently an Associate Professor (Research) at Beijing Institute of Technology, Beijing, China. His research interests include data centric networks, blockchain and distributed ledger technologies, wireless and sensor networks, software-defined networks, and 5G vehicular and UAV networks. He currently serves as associate editor for IEEE Access, and regularly serves on program committees of IEEE and ACM conferences.

Md Monjurul Karim received his M. Eng and B. Eng in Computer Science from Northwestern Polytechnical University, Xian, China. He is currently working as a research and teaching assistant at Southern University of Science and Technology, Shenzhen, China. His research interests include Software-Defined Networking, Information-Centric Networking, Named Data Networks, and Next-Generation Networking.

Shidong Huang received the M.S. degree in 2013 from East China University of Science and Technology, Shanghai, China. He is currently pursuing the Ph.D. degree with the School of Computer Science, Wuhan University, Wuhan, China. His research interests include C-V2X communications, mmWave ad hoc networks and network coding.