

## Research Article

# A Novel Fog Computing Based Architecture to Improve the Performance in Content Delivery Networks

Fatimah Alghamdi , Saoucene Mahfoudh , and Ahmed Barnawi 

Faculty of Computing and Information Technology, King Abdulaziz University (KAU), Jeddah, Saudi Arabia

Correspondence should be addressed to Fatimah Alghamdi; [fmalghamdi2@kau.edu.sa](mailto:fmalghamdi2@kau.edu.sa)

Received 18 May 2018; Revised 15 October 2018; Accepted 6 November 2018; Published 23 January 2019

Academic Editor: Miguel Garcia-Pineda

Copyright © 2019 Fatimah Alghamdi et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Along with the continuing evolution of the Internet and its applications, Content Delivery Networks (CDNs) have become a hot topic with both opportunities and challenges. CDNs were mainly proposed to solve content availability and download time issues by delivering content through edge cache servers deployed around the world. In our previous work, we presented a novel CDN architecture based on a Fog computing environment as a promising solution for real-time applications. In such architecture, we proposed to use a name-based routing protocol following the Information Centric Networking (ICN) approach, with a popularity-based caching strategy to guarantee overall delivery performance. To validate our design principle, we have implemented the proposed Fog-based CDN architecture with its major protocol components and evaluated its performance, as shown through this article. On the one hand, we have extended the Optimized Link-State Routing (OLSR) protocol to be content aware (CA-OLSR), i.e., so that it uses content names as routing labels. Then, we have integrated CA-OLSR with the popularity-based caching strategy, which caches only the most popular content (MPC). On the other hand, we have considered two similar architectures for conducting performance comparative studies. The first is pure Fog-based CDN implemented by the original OLSR (IP-based routing) protocol along with the default caching strategy. The second is a classical cloud-based CDN implemented by the original OLSR. Through extensive simulation experiments, we have shown that our Fog-based CDN architecture outperforms the other compared architectures. CA-OLSR achieves the highest packet delivery ratio (PDR) and the lowest delay for all simulated numbers of connected users. Furthermore, the MPC caching strategy shows higher cache hit rates with fewer numbers of caching operations compared to the existing default caching strategy, which caches all the pass-by content.

## 1. Introduction

Content retrieval has become the primary usage of the Internet. To solve content availability and download time issues, Content Delivery/Distribution Networks (CDNs) [1, 2] have evolved as virtual overlay networks built on top of existing network infrastructures. With CDNs, content is distributed to cache servers located close to users, instead of to single, remote servers. CDNs have become the main part of Internet architecture, since they help improve the quality of Internet services (QoS) as well as the quality of users' experience (QoE) [2]. Globally, CDNs are expected to carry 70% of all Internet traffic by 2021, and most of them will be carrying video traffic [3]. Although CDNs succeed in delivering content with high availability and performance, they cannot properly handle the recent, heavy workload

on edge servers [2, 4]. Accordingly, real-time and latency-sensitive applications, like video streaming, are delivered out of place, and the user experience is affected. To overcome this issue, we have proposed [5] a Fog-based CDN architecture, in which Fog nodes are introduced at the edges of CDN servers without disrupting the conventional CDN infrastructure. In our proposed model, Fog nodes are suggested to communicate with each other by the Information Centric Networking (ICN), name-based routing approach, and cache only the most popular content (MPC). Fog computing and ICN are two dominant technologies discussed in the future Internet research context.

This paper aims to prove the effectiveness of our Fog-based CDN model while analyzing its performance. To achieve this, we propose a new ICN name-based routing protocol by extending the Optimized Link-State Routing

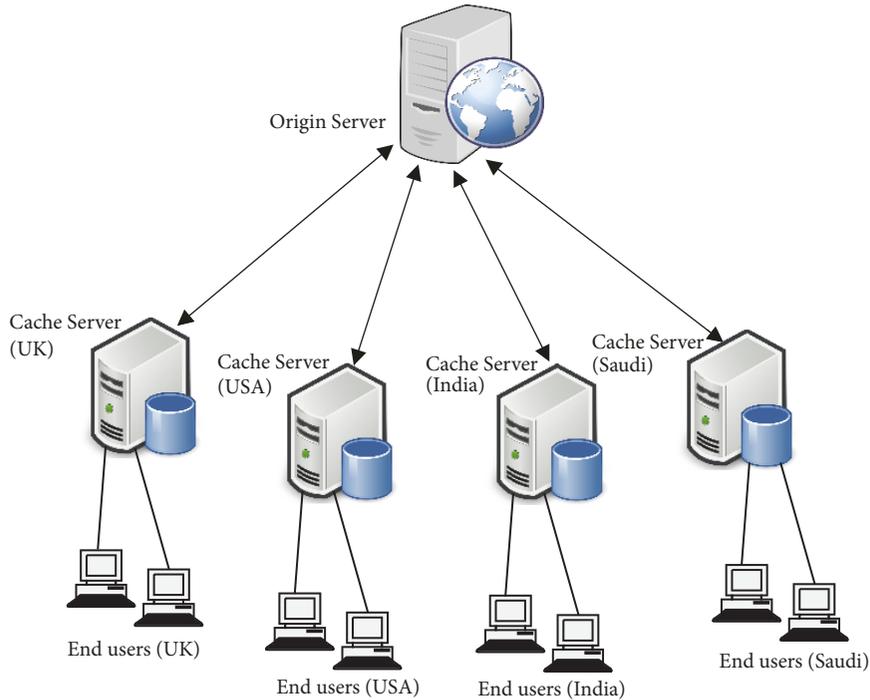


FIGURE 1: Abstract architecture of CDN.

(OLSR) protocol to be content aware, briefly named CA-OLSR. Moreover, we incorporate the following caching strategy: Most Popular Content (MPC) caching into CA-OLSR. The performance of the proposed model is evaluated using a network simulator tool and is compared with two similar architectures. The first is another Fog-based CDN model, in which Fog nodes are introduced as native cache resources, while the request routing is performed by the original OLSR (IP-based routing). The second is the classical cloud-based CDN. Comparing Fog-based CDN with classical cloud CDN can show us the impact of Fog on CDN performance.

The main contributions of this work are as follows:

- (i) Design a novel CDN architecture based on the Fog computing environment, in which Fog nodes are closer to users and can provide them with the most desired content.
- (ii) Exploit ICN, name-based routing mechanisms in the Fog network by implementing a new, content aware routing protocol based on the OLSR ad hoc routing protocol, using the network simulator tool NS2.
- (iii) Evaluate the performance of the proposed CDN compared with two similar approaches. The results show that our Fog-based CDN achieves significant performance gains and outperforms the compared architectures.

The rest of the paper is organized as follows. In Section 2, we provide an overview of the three Internet-based infrastructures that have been combined in our architecture: CDN, ICN, and Fog computing. Section 3 presents our Fog-based CDN architecture with its content delivery process and MPC

caching strategy, while Section 4 focuses on the proposed CA-OLSR routing protocol design. Our simulation approach and results are discussed in Section 5. Finally, Section 6 concludes the paper and provides the direction of future work.

## 2. Internet Content Delivery: State of the Art

This section reviews the existing content delivery platforms, including CDN, ICN, and Fog computing, and provides a detailed analysis of their approaches.

*2.1. Content Delivery Network (CDN).* Content delivery networks, or content distribution networks, (CDNs) are defined in the Internet Engineering Task Force (IETF) RFC 3466 [6] as a type of content network, which emerged centered around “content”. CDNs can be seen as virtual overlay networks built on top of generic IP to solve performance problems related to network congestion and to improve web content accessibility in a cost-effective way [2, 7]. CDNs consist of several cache servers, also called surrogates, containing copies of web content and distributed around the world in order to satisfy user requests by utilizing the most appropriate server, rather than remote origin servers, as shown in Figure 1.

Therefore, CDNs benefit not only the end users, but also the content providers and the Internet service providers (ISPs) who deploy CDN servers in their networks [7]. The end user can perceive higher QoS, in terms of download time and bandwidth, resulting in improved user experience (QoE). The content provider can offer larger volumes of reliable services, and the ISP can benefit from reducing the traffic transmitted to its origin server (backbone).

TABLE I: Comparison of information-centric and host-centric networking.

	Information-centric networking	Host-centric networking
<b>Naming</b>	Content name is persistent and unique.	Host name can easily break and is not aware of the existence of replica content.
<b>Security Model</b>	Secures the content itself, which allows for stronger, more flexible, and customizable content protection.	Secures the host-to-host communication channels.
<b>Mobility of Nodes</b>	Does not affect connection management.	Makes the connection hard to manage.
<b>Failure Tolerance</b>	High.	Low.
<b>Scalability of Content Distribution</b>	High (no need to modify any domain or architecture).	Limited (depends on overlay dedicated systems).

However, because of the continuous increase in Internet traffic, CDNs cannot continue consistent, high quality content delivery due to overloading on their edge servers [2, 4]. Over the last decade, CDN architecture has seen rapid evolution to solve the scalability issue of CDN edge servers and optimize delivery quality as well as user experience. Limited to the length of the article, we only present some examples from the existing work in this context.

A hierarchical architecture with cooperative caching [9] and one with application level multicast [10] were proposed for delivering on-demand and live multimedia content, respectively. Such architecture scales well for increasing traffic, but CDN servers are expensive to deploy and maintain. To save infrastructure cost, CDNs were integrated with infrastructure-less content delivery technology, which is Peer to Peer (P2P) [4, 8]. Such hybrid architecture combined the success and reliability of CDNs with the scalability and cost effectiveness of P2P.

Recently, leveraging cloud computing resources in CDNs has gained extensive attention [9]. Such cloud CDN models can provide high-performance delivery for wider ranges of applications without costly infrastructure. Furthermore, they can be established on either of the previously mentioned approaches, hierarchal CDN and hybrid CDN/P2P. In [10], a hierarchal cloud CDN was proposed, combining multicloud providers, while, in [11], P2P communication was incorporated in cloud CDN edge servers to improve the response time of video streaming services. Accordingly, cloud CDNs are considered the most valuable and cost-effective alternatives to traditional CDNs [9], especially for high bandwidth demanding applications. Although, since such cloud CDNs do not exploit the full advantages of cloud computing [12], they can be defined as cloud infrastructure-assisted CDN models.

In our work, we have considered this issue and proposed to move away from the centralized cloud to its edge extensions (Fog computing) to offer cloud rich services at the edges.

**2.2. Information Centric Networking (ICN).** Information Centric Networking (ICN) [13] is a new networking paradigm proposed to provide highly scalable and efficient content distribution. ICN, like CDNs, has emerged centered around “content”. The difference is that ICN provides content delivery functions within the communication infrastructure [14],

more specifically, in its network layer. It leverages router buffers as in-network caching and performs the requested routing as a native network operation based on the content name [13]. In contrast, CDNs provide content delivery functions through overlay networks built over the traditional Internet [7], which has a host-centric infrastructure not aware of content. Caching in CDNs is provided as an application-layer service, and request routing is mostly performed by a third party, the Domain Name System (DNS) infrastructure [12]. The ICN approach was mainly proposed to switch from the traditional IP network, and it is much more adapted to current Internet usage [15] (i.e., users care only about the content, or service they want, and not about the machine that hosts it).

Using the ICN approach not only improves content distribution, but also has many motivating advantages [13, 15] against host-centric networks, as listed in Table 1.

A fundamental ICN function is routing content requests toward a particular node that has the requested content. To achieve this goal, different routing approaches have been proposed. They can be classified as name resolution approaches and name-based routing approaches [13]. In this paper, we consider the name-based routing approach in which content names are included in the router’s Forwarding Information Base (FIB) table. This approach eliminates the delay caused by the resolution process [16] and simplifies content delivery. Moreover, it selects the serving node based on network-related information, considering the server load and its proximity to users [14, 15]. Therefore, it improves the reliability of content access and uses network resources more efficiently compared to CDNs that work over the top and are not aware of the underlying network status.

ICN name-based routing approaches can be designed within overlay networks, as in TRIAD [17], or within clean slate networks, as in Content Centric Networks (CCNs) [18].

Although ICN outperforms CDNs in terms of routing efficiency, it is still lacking scalability to be deployed on a widespread range [13, 15]. Furthermore, the application-awareness of CDNs is more efficient than in-network caching and is required for improving content delivery performance [19, 20].

In this work, we aim to get the best of both worlds by exploiting ICN name-based routing in the second level of the proposed architecture, which is the Fog network, while keeping access to conventional CDNs at the first

TABLE 2: Comparison of cloud computing and Fog computing.

	Cloud	Fog
<b>Deployment</b>	Centralized.	Distributed in regional areas; related to specific locations.
<b>Distance to Users</b>	Faraway. Users connect to it through IP networks.	In the proximity. Users often connect through single hop connections.
<b>Target User</b>	General internet users.	Mainly mobile users.
<b>Hardware</b>	Scalable storage space and computing power.	Limited storage, computing power, and wireless interface.
<b>Latency</b>	High.	Low.
<b>Service Type</b>	Global information collected worldwide.	Limited localized information services related to specific deployment locations.

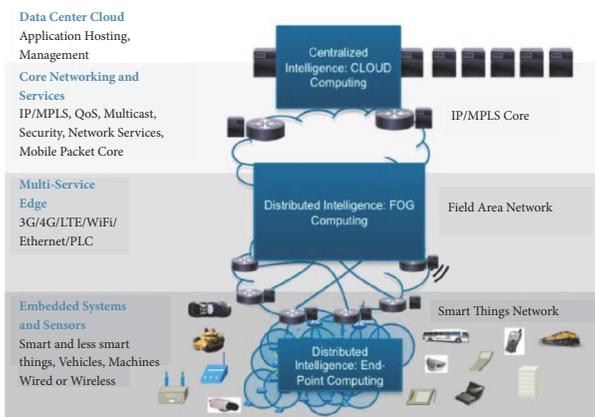


FIGURE 2: Abstract architecture of Fog computing [21].

level. Because the clean slate approach requires significant and costly modifications, we choose to follow the overlay approach adopted over the existing Internet infrastructure.

**2.3. Fog Computing.** Fog computing technology was invented recently by Cisco as a promising computing platform to support future Internet of Things (IoT) applications, which are mostly critical (i.e., latency-sensitive) applications [21]. Fog computing extends cloud computing to the edge of the network to eliminate the delay caused by transferring data to the remote cloud [22]. Furthermore, it has many characteristics for supporting future IoT applications, to name but a few: dense geographical distribution, proximity to end users, mobility support, and real-time interaction. Fog computing architecture shows intermediate layers composed of Fog nodes located between traditional cloud data centers and IoT end devices, as depicted in Figure 2.

While Fog nodes provide localized services deployed in different locations, the cloud provides global services and acts as a central controller for those distributed Fog nodes. In addition, the cloud is like a central information repository from which the Fog nodes get the requested information for their own caches to serve subsequent requests locally [23]. Once an end device connects to a Fog node, the Fog node can serve it either directly or with assistance from the cloud.

Thus, there is an essential interaction between Fog nodes and the cloud [21], and many applications require both Fog localization and cloud globalization.

Table 2 summarizes the differences between cloud computing and Fog computing according to [23, 25].

Fog computing has a generic computing paradigm aiming to bring cloud services (computing, storing, and networking) closer to physical IoT devices through wired or wireless technologies [21]. In addition, Fog nodes can be wired devices, such as edge routers and switches; wireless devices, such as access points and cellular base stations; or mobile devices, such as laptops and smartphones [23, 26?]. In our model, we have addressed Fog usage as a content delivery technique, more specifically, to provide caching and routing services. The detailed architecture that we have considered is described in the next section.

To the best of our knowledge, no previous studies have shown the impact of integrating Fog computing with CDN systems [?].

### 3. Proposed Fog-Based CDN Architecture

Our review concludes that CDNs and Fog are Internet-based infrastructures, and both have a similar style consisting of origin servers surrounded by sets of surrogates at the edges of networks. While CDN surrogates are pure cache servers deployed on widespread ranges, Fog nodes are intelligent, small, cloud units providing computing, storage, and networking services in localized sites. Thus, introducing Fog computing on CDN systems to provide additional levels of content delivery has high potential for solving the scalability issues of CDN edge servers. It furthermore optimizes the delivery of modern Internet applications. Additionally, ISPs can benefit from our proposed architecture since Fog nodes can reduce traffic transmitted on the links that connect their network with the Internet backbone and other ISP networks. The most promising advantage of our proposed Fog-based architecture resides in supporting the emerging 5G wireless technology to meet the requirements of latency-sensitive, IoT applications/services.

Without disrupting the existing CDN infrastructure, we have proposed deploying Fog computing layers between the cloud-based CDN layers and the end user layers. Our

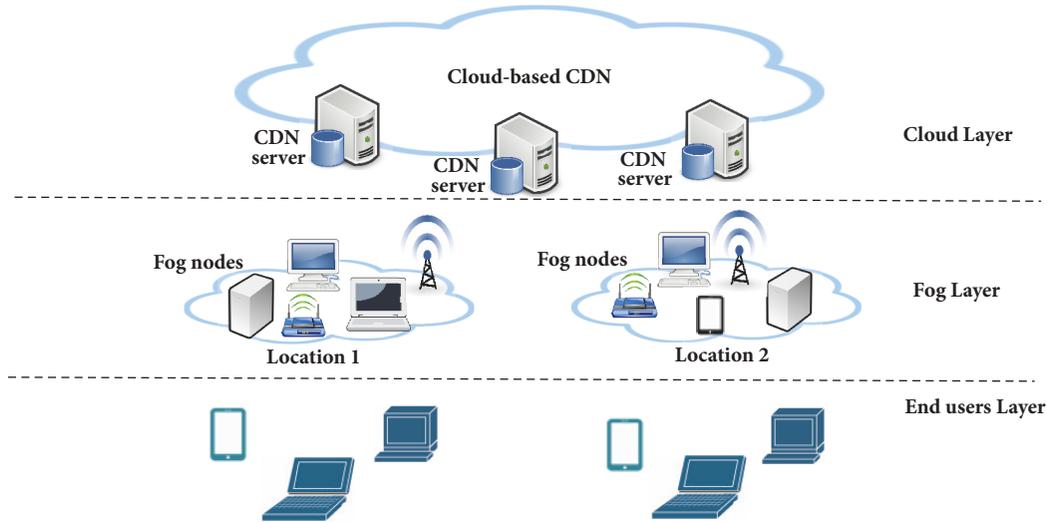


FIGURE 3: Abstract architecture of Fog-based CDNs.

architecture of Fog-based CDNs can be further abstracted to a three-level, hierarchical model, as shown in Figure 3. At the cloud layer, the cloud-based CDN system is deployed, and the content is disseminated on the CDN servers located inside the cloud. At the Fog layer, the Fog nodes are redeployed, forming more edge networks, which are geographically closer to the users than CDN servers. The last level represents the end users' devices utilizing Fog nodes to connect to the Internet.

The communication model considered in the Fog network is as follows. Each Fog node is implemented with a wireless interface, and the user's device can directly connect to it through single-hop wireless connection. The user's device uses the nearby Fog node to connect to the Internet. The communication between Fog nodes that exist in the same location is handled through a mobile ad hoc network (MANET) routing protocol, as suggested by [25].

We have proposed to implement ICN networking methodology into the Fog network, aiming to achieve a highly efficient content delivery at the edges of CDNs. The proposed architecture should contribute toward improving CDN performance by utilizing Fog-based architecture, where Fog nodes act as ICN nodes. Unlike pure ICN architectures that require changing basic network operations and likely changing hardware/software configurations, our proposed architecture brings about ICN benefits toward the access network in the most cost-effective way; it deals with plain IP packets, and the next hop in each node is determined by consulting IP Forwarding Information Base (FIB) entries. This flexible strategy for deploying ICN functionalities at the edge of the network is certainly beneficial from operational and economical points of view in comparison to the pure ICN network.

More specifically, we consider that each Fog node has a cache store and routing table, and the delivery process is realized via three major phases:

- (i) Request routing, using the name-based routing approach.

- (ii) Content routing back to the requester, using the conventional (IP) address-based routing approach.

- (iii) Caching the popular pass-by content.

For name-based routing, Fog nodes that exist in the same location exchange information about how to reach given content (represented by its name) and then set up the FIB tables with content names as destinations. For the Fog nodes to maintain the same and correct view of others, the FIB tables containing the routing and the cached content information are periodically exchanged.

When a user wants to find specific content, the network sends a request containing the content name to the nearest Fog node. Once the Fog node receives the request, it looks for the content name in its own cached records. If the content is available in the local cache store, the Fog node sends the content to the requester using the conventional (IP) address-based routing approach. Otherwise, the Fog node fetches the missing content from the Fog node that has it cached. More specifically, it looks through its FIB routing table to determine which Fog node hosts the requested content, and then it forwards the request toward the selected node, using the conventional forwarding (IP-based) mechanism. If multiple Fog nodes host the requested content, the nearest node (in terms of hop count) will be selected. That is, Fog nodes support cooperative caching and can work independently from CDN servers, except when the requested content is not available in the Fog nodes. In this case, the request is forwarded to the CDN servers.

As content is routed back toward the requester, it will be cached by the edge Fog node in order to satisfy subsequent requests from the local cache. Caching all the pass-by contents involves many challenges. On the one hand, it causes overload on the memory, bandwidth, and processing. On the other hand, it may lead to removing popular content to make space for unpopular content, decreasing caching performance and user experience. To improve caching efficiency, we have proposed to use a popularity-based caching strategy

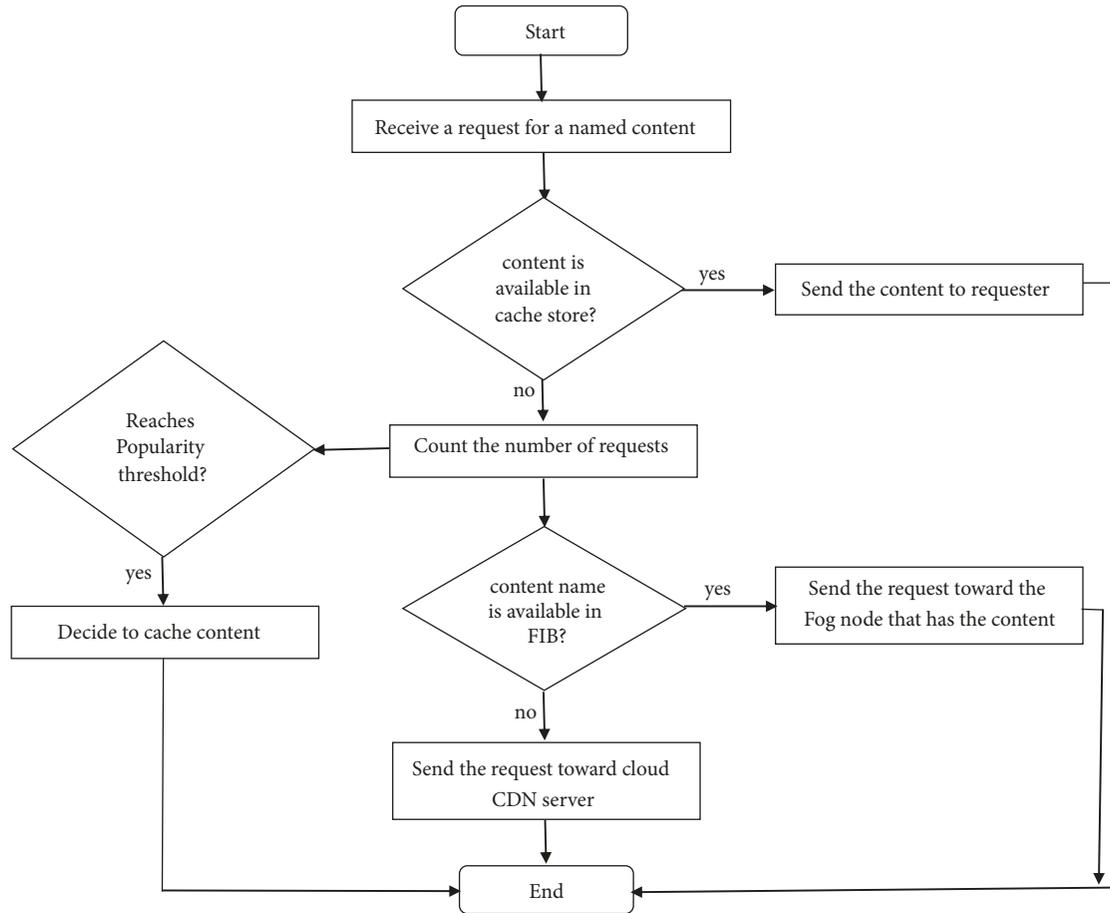


FIGURE 4: Fog content delivery flowchart.

allowing the Fog nodes to cache only the most popular content (MPC) in their deployment locations. The design detail is given in the next subsection.

**3.1. Most Popular Content (MPC) Caching Strategy.** Within a unit of time, each Fog node locally counts the number of requests for content that is not found in its own cache. It maintains a popularity table to store the content name along with its popularity count (i.e., number of requests). Once a local popularity count for specific content reaches a popularity threshold, the content is tagged as popular and cached by the node.

Caching strategies and replacement policies are closely related and are required to manage the node's cache [16]. A Least Recently Used (LRU) policy is one of the most common existing replacement policies, and we have proposed extending it in our architecture. Fog nodes maintain a popularity count of each piece of cached content. Once the buffer of the Fog node is full, the Fog node will decide to remove the least popular content from the LRU list to make space for the new content.

Accordingly, once a user requests cached content, the user will receive the content directly from the relevant Fog node, rather than the CDN server or the original distant cloud server, unless the cached content has been replaced.

Like the ICN name-based routing mechanism, caching strategy is involved in the routing phase. The next section will discuss, in detail, the design of our content aware routing protocol, CA-OLSR. The flowchart of the proposed content delivery process, as performed by a Fog node, is presented in Figure 4.

#### 4. CA-OLSR Routing Protocol Design

In the previous section, we proposed a Fog network configuration acting as a MANET. Many routing protocols were proposed for this MANET [24]. Among them, we have selected the OLSR [?] protocol since it is a table-driven protocol, and it has very satisfying performances due to its multipoint relaying (MPR) strategy, which is what a Fog content delivery network requires.

We have extended OLSR to be content aware CA-OLSR. While OLSR routes data packet based on the destination address specified in the packet, CA-OLSR differentiates between the request data packet (Request Packet) and the reply data packet (Content Packet) with two different routing strategies. Request Packet is being routed to requested content based on its ID while Content Packet is routed back to the requester based on its address.

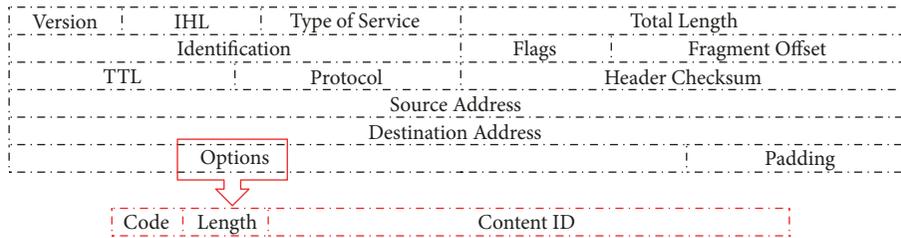


FIGURE 5: IP packet header used with CA-OLSR.

TABLE 3: CA-OLSR routing table.

Destination Address	Cached content IDs	Next Hop Address	Distance
---------------------	--------------------	------------------	----------

Similar to ICN, our proposed system deals with content as blocks of packets, but the packets in our system are regular data packets, i.e., IP packets. As IP packets do not contain a field of content ID, we have specified it using the Option field. The format of the packet header used with CA-OLSR is presented in Figure 5.

CA-OLSR performs the main functionalities of OLSR, with some modifications related to content consideration as follows.

**4.1. Neighbor Detection.** Each node periodically broadcasts, to all one-hop neighbors, its Hello messages containing the IDs of the content available in its cache store and its neighbors information, including the IDs of the content available in their cache stores. Once a node receives a Hello message, it can detect its one-hop and two-hop neighbors with their cached content IDs, and, correspondingly, the neighbor tables will be built. In addition, a node can select its MPRs in the same way as the original OLSR.

**4.2. Topology Discovery.** MPRs periodically generate Traffic Control (TC) messages, containing information about network topology and the content cached in network nodes. TC messages are flooded by MPRs to all nodes in the network. TC messages, more specifically, advertise the addresses of MPR selector nodes with their cached content IDs. Once a node receives a TC message, it can discover the network topology with the content locations and build its topology table.

**4.3. Routing Table Calculation.** Like OLSR, the node builds its routing table based on the information contained in its neighbor tables and topology table so the shortest route to each destination is given. In CA-OLSR, we have extended recorded route entries to contain the information of the destinations with their cached content IDs, as shown in Table 3.

Once the node receives a Request Packet for forwarding, it looks for the requested content ID, specified in the packet, in its routing table to find the destination that has the content. If multiple destinations have the requested content, the node selects the closest one. That means the node finds the optimal

route (in terms of hop counts) to the requested content. Content Packets are routed in the same way as the original OLSR (i.e., based on the packet destination address).

**4.4. Applying CA-OLSR in Fog-Based CDN Architecture.** We have extended CA-OLSR to identify the hierarchical architecture of Fog-based CDNs, so it can route unsatisfied requests to cloud CDN servers. In addition, we incorporate it with the MPC caching strategy explained in the previous section. Thus, our architecture involves the caching strategy in the routing phase, like the ICN name-based routing mechanism.

To clarify the workflow scenario of the proposed Fog-based CDNs (including routing and caching strategies) and the communication flow between system items (user/Fog node/cloud CDNs), we present different use cases as follows. In the beginning, the user connects to the nearest Fog node and sends a Request Packet containing the ID of the desired content. Then, one has the following:

**Use Case 1.** A Fog node receives a request for content available in its cache store. The node sends the content directly back to the requester.

**Use Case 2.** A Fog node receives a request for content that is available in another Fog node's cache store. The node looks for the content ID on its routing table and routes the Request Packet to the nearest node containing the content. The destination node sends back the content to the requester through its connected Fog node.

**Use Case 3.** A Fog node receives a request for content that is not available in the Fog network. The content ID could not be found in the node's cache store nor routing table, so the node forwards the Request Packet to the cloud CDN server. The cloud CDN sends the content back to the requester through its connected Fog node.

**Use Case 4.** Within a unit of time, a Fog node receives multiple requests for a particular content unavailable in its cache store. The number of incoming requests reaches the popularity threshold, and the Fog node decides to cache this popular content on its store so that it can satisfy requests for that content. Note that, if the buffer of the Fog node is full, the Fog node will decide to remove the least popular content, as explained in Section 3.

TABLE 4: Simulation environment parameters.

<b>Network Type</b>	Static
<b>Packet Size</b>	512 bytes
<b>Duration</b>	1000s
<b>Connection Type</b>	CBR/UDP
<b>Simulation Area</b>	500m * 500m
<b>Transmission Range</b>	250m
<b>Total Content in CDN</b>	1000
<b>Replacement Policy</b>	LRU (Least Recently Used) + Least Popular
<b>Popularity Model</b>	Zipf's distribution with exponent of 1

## 5. Performance Evaluation

In this study, we have used the simulation-based performance evaluation approach to prove the effectiveness of our model. Moreover, we have conducted a comparative performance study considering the following architectures:

- (i) Our Fog-based CDN uses CA-OLSR (name-based routing) along with an MPC caching strategy implemented in Fog nodes. We denote this approach by **CA-OLSR**.
- (ii) Pure Fog-based CDN uses the original OLSR (IP-based routing) along with a default caching strategy implemented in Fog nodes. We denote this approach by **NCA-OLSR** (Non-Content Aware OLSR).
- (iii) Classical cloud-based CDN uses the original OLSR routing. We denote this approach by **Cloud**.

Using the network simulator tool NS2 [?], we have implemented each architecture by considering it as an interdomain network.

In our experiments, we have assumed that the popularity of each content follows Zipf's Law [27] to ensure that we have a high popularity for some content. Regarding the other parameters introduced by the MPC caching strategy, cache size, and popularity threshold, we have selected their values as explained below. The cache size of the Fog node is assigned to be 1% to 2% of the CDN cache size to avoid load at the node, since the Fog node has limited storage capacity. Popularity threshold values are selected according to extensive simulation processes in which Zipf's low exponent, Fog cache size, and total content are fixed, and the resulting cache hit rates are varied.

The environment parameters considered in our simulation are reported in Table 4.

Content delivery performance is essentially affected by routing protocol and caching strategy. Hence, our experiments are done to evaluate both as follows.

*5.1. Network Performance Analysis.* To evaluate the routing protocol, we have studied the impact of the number of connected users and the number of Fog nodes on the network performance in terms of data transfer delay and packet

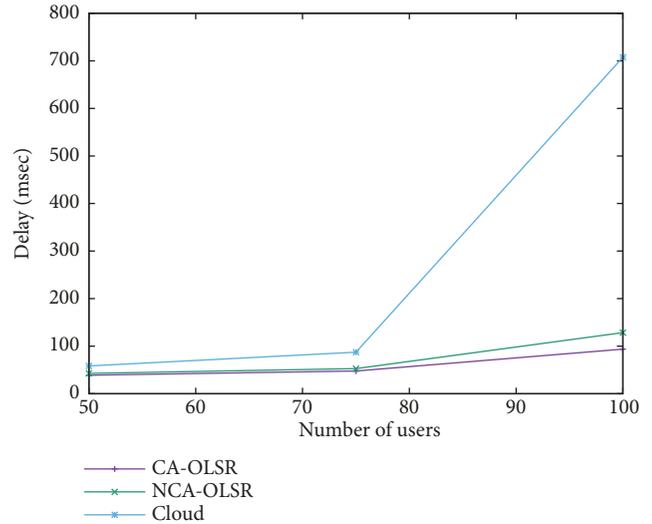


FIGURE 6: Data transfer delay by varying number of users.

delivery ratio (PDR). We have considered the delay from sending a Request Packet to receiving a Content Packet. PDR is the ratio of the received data size to the sent data size. Through the following routing evaluation experiments, we keep the caching parameters in terms of popularity threshold and cache size fixed to five and ten, respectively.

*5.1.1. Impact of Number of Connected Users.* In this experimental setup, the number of users varied from 50 to 100, with an increment of 25 users, whereas the number of Fog nodes was fixed at 15 nodes. Other parameters were kept as previously mentioned in Table 4.

Figure 6 shows the data transfer delay by varying the number of connected users. As expected, when the number of users increased, the traffic load increased, causing data transmission delay. This delay is shown clearly in cases of the cloud CDN approach, since the user request is always satisfied from the distant cloud CDN server. The delay in the Fog CDN approaches, CA-OLSR and NCA-OLSR, is lower than that in cloud CDN. This is because the users can receive the content from the distributed nearby Fog nodes in a shorter time than they can receive it from single, distant cloud CDN servers. CA-OLSR achieves the lowest delay compared to NCA-OLSR and cloud CDN approaches. This is because the CA-OLSR approach allows fetching of the missed content from other Fog nodes due to its content aware routing protocol. Furthermore, its caching strategy guarantees that most popular content is available in the Fog nodes, which reduces the need for transferring requests to the distant cloud CDN servers. As shown, the CA-OLSR approach reduces the delay about 28% and 87% compared to NCA-OLSR and cloud CDN approaches, respectively. In addition, the delay with CA-OLSR increases slowly when the load in the network increases, which guarantees very high QoS for real-time applications. Figure 7 shows packet delivery ratio (PDR) by varying number of connected users. Increased traffic load lowers PDR using the cloud CDN

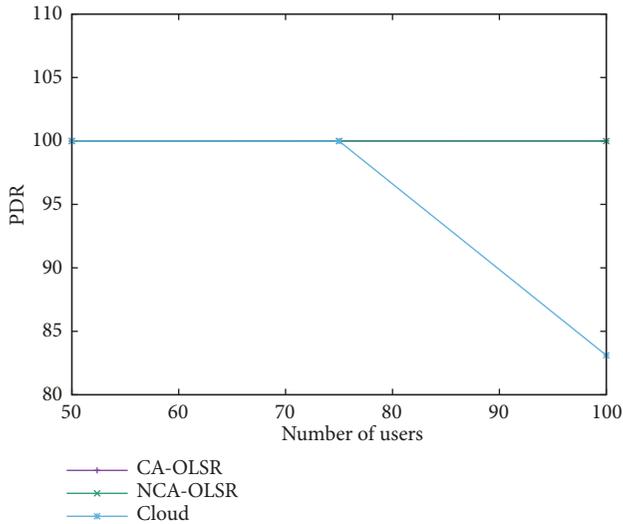


FIGURE 7: Packet delivery ratio by varying number of users.

approach, but it does not affect the PDR performance of CA-OLSR and NCA-OLSR approaches. Hence, both Fog CDN approaches guarantee users will receive most of the sent data, even with high traffic loads. This results from satisfying user requests from nearby Fog nodes instead of the distant cloud CDN server. Therefore, Fog CDN approaches, CA-OLSR and NCA-OLSR, outperform cloud CDN approaches in terms of PDR. This result shows the improvement that Fog computing can introduce to CDN performance. The improved PDR (almost 100%) shows promising results, especially when QoS is concerned, such as in e-health care or any emergency application. As QoS is improved, QoE will improve as well.

**5.1.2. Impact of Number of Fog Nodes.** In this experiment, the number of Fog nodes varied from ten to 20 with an increment of five nodes, whereas the number of users was fixed at 50. Other parameters were kept as previously mentioned in Table 4. Although the cloud CDN approach is not affected by this factor (i.e., the number of Fog nodes), we included it for the purpose of comparison. Concerning the cloud performance results, we consider 20 hops between the users and cloud CDN servers. Figure 8 shows that Fog CDN approaches outperform cloud CDN approaches in terms of data transfer delay, as the content is delivered from the edge Fog nodes rather than from distant cloud CDN servers. Based on this experiment, the benefit of edge computing is evident since it has reduced the content delivery delay significantly. In Fog CDN approaches, CA-OLSR and NCA-OLSR, Fog nodes can satisfy user requests from their local caches if the content is available within the caches. When the requested content is missing from the local cache, CA-OLSR and NCA-OLSR act in different ways. In the NCA-OLSR approach, the missing content is fetched from the distant cloud CDN server, even if the content is available in nearby Fog nodes. However, the CA-OLSR approach fetches the missing content from the nearest Fog node that has the requested content cached. For this reason, our CA-OLSR approach outperforms the NCA-OLSR approach in terms of delay. This result is

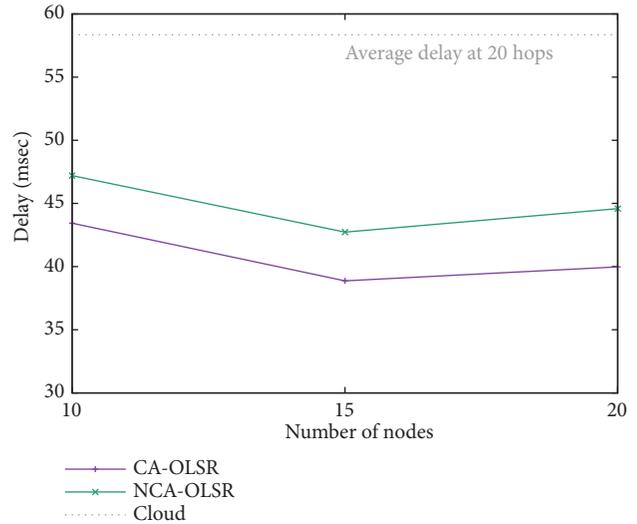


FIGURE 8: Data transfer delay by varying number of Fog nodes.

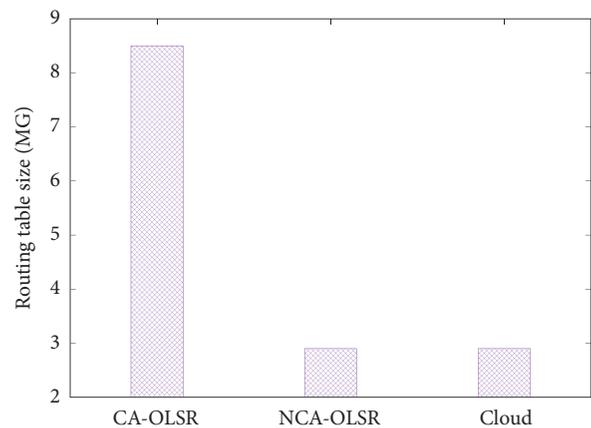


FIGURE 9: Routing Overhead in terms of size of routing table.

very interesting for future Internet applications that require low delay, such as audio, video streaming, or any real-time application.

Despite the advantages of CA-OLSR, it involves numerous overheads related to name-based routing. More specifically, creating and updating CA-OLSR routing tables consume more computational resources in terms of memory, bandwidth, and processing because the routing tables include content names. Figure 9 shows that the routing tables of CA-OLSR (using the same circumstance of 20 nodes and 50 users) are the largest compared to those of the NCA-OLSR and cloud CDN approaches.

Even though CA-OLSR, NCA-OLSR, and the cloud CDN approach exchange the same number of Routing Packets to build routing tables, as depicted in Figure 10, the size of these packets is higher in CA-OLSR because they include content IDs in Hello and TC messages, which, in turn, lead to more communication overhead. This overhead depends on the content ID format and the number of content pieces stored in each node. We reduced this overhead in our approach by

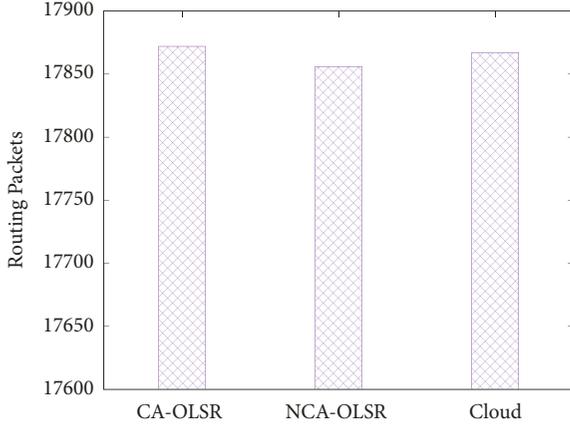


FIGURE 10: Routing Overhead in terms of number of Routing Packets.

using a popularity-based caching strategy in which only the MPC is cached in the Fog node.

**5.2. Caching Performance Analysis.** To evaluate the caching strategy of Fog CDN approaches, we have studied the impact of popularity threshold and Fog node cache size on caching performance in terms of the following measures:

- (i) **Local cache hit rate (LCHR):** the ratio of requests that are satisfied by the local caches of attached Fog nodes (HL) versus the total number of requests (R).

$$LCHR = \frac{HL}{R} \quad (1)$$

- (ii) **Cache hit rate (CHR):** the ratio of requests that are satisfied by all nodes in Fog networks (H) versus the total number of requests (R).

$$CHR = \frac{H}{R} \quad (2)$$

- (iii) **Number of caching operations:** computed by the number of cached elements.

Note that the cloud CDN approach is not included in this analysis. Through the following caching evaluation experiments, we keep the network parameters, in terms of number of nodes and number of users, fixed at 15 nodes and 50 users.

**5.2.1. Impact of Popularity Threshold.** In this experiment, the popularity threshold value varied as one, two, five, and ten, whereas the cache store size was fixed at ten. Other parameters were kept as previously mentioned in Table 4.

As shown in Figure 11, the CA-OLSR approach, with different popularity threshold values, outperformed the NCA-OLSR approach because the latter caches content without considering popularity. The CA-OLSR approach, with a popularity threshold of five, shows the highest local cache hit rate. When the popularity threshold increases from one to

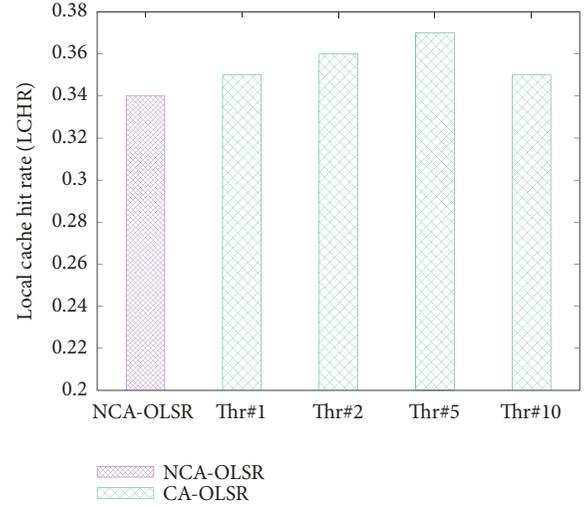


FIGURE 11: Local cache hit rate by varying popularity threshold.

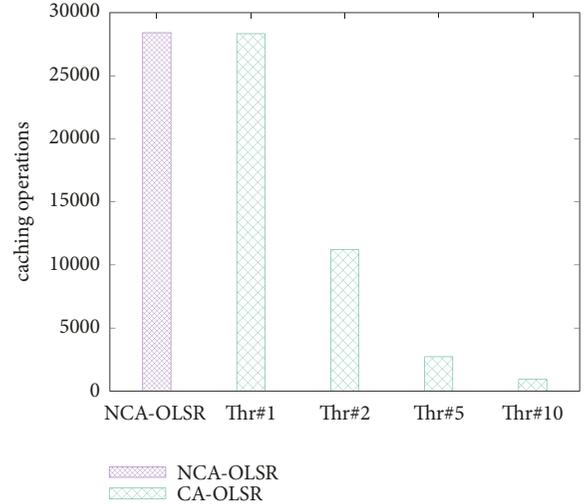


FIGURE 12: Caching operations by varying popularity threshold.

five, the local cache hit rate also increases. However, when the popularity threshold is raised to ten, the local cache hit rate decreases. A high local cache hit rate is important for multimedia applications that consume large amounts of bandwidth. As such multimedia can be delivered from nearby Fog nodes, the network bandwidth can be saved, and the QoS will satisfy users. As shown in Figure 12, both the NCA-OLSR approach and the CA-OLSR approach with popularity thresholds of one have the highest number of caching operations. A popularity threshold set to five lowers caching operations by 90% from the threshold of one, although it achieves the highest local cache hit rate, as shown in Figure 11. As the popularity threshold value increases, the number of caching operations is reduced, saving memory, processing, and bandwidth resources.

**5.2.2. Impact of Fog Cache Size.** In this experiment setup, the cache store size of Fog nodes varied at ten, 15, and 20, whereas

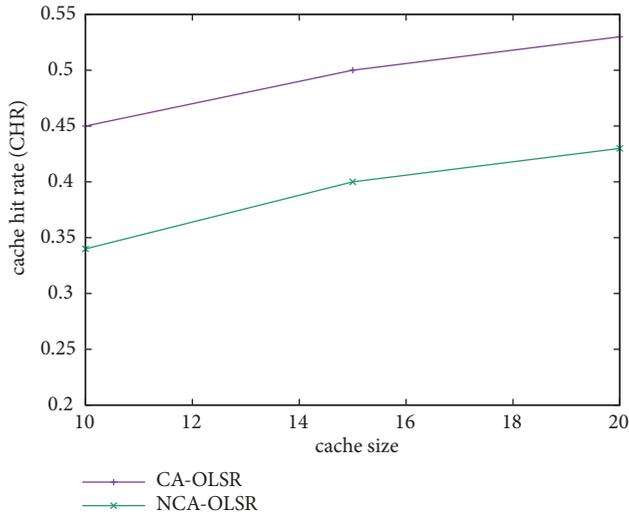


FIGURE 13: Cache hit rate by varying cache size.

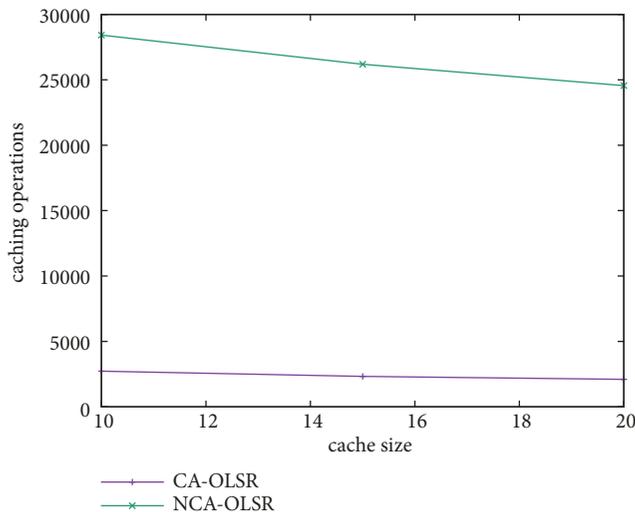


FIGURE 14: Caching operations by varying cache size.

the popularity threshold was fixed at five. Other parameters were kept as previously mentioned in Table 4.

Clearly, as the size of cache store increased, the Fog nodes made more content available in their caches, and then most of the requests could be satisfied from the Fog network, increasing the cache hit rate. Figure 13 shows that the NCA-OLSR approach has a lower cache hit rate compared to the CA-OLSR approach. This is because (1) it lacks the content aware routing mechanism, and (2) it caches content without considering its popularity. Consequently, the node in NCA-OLSR often cannot satisfy the most-requested content, neither from its local cache nor from other Fog nodes, which in turn leads to low cache hit rates compared to CA-OLSR.

Figure 14 shows the number of caching operations by varying the cache size of Fog nodes. When the cache size is small, it will be filled faster. Thus, the node continuously replaces existing content to cache missing content. According to our replacement policy, the least popular content from

the LRU list is replaced. On the other hand, large Fog nodes make larger amounts of content available in their local caches and perform fewer replacements and caching operations. This justification applies to NCA-OLSR more than to CA-OLSR because the CA-OLSR approach assigns a popularity threshold to cache content, and the node cache store is continuously filled by only the most popular content. Therefore, the CA-OLSR approach can satisfy user requests with fewer operations compared to NCA-OLSR. Moreover, caching only the most popular content maintains similar numbers of caching operations, whether the cache store size is large or small.

## 6. Conclusion

In this paper, we presented a novel content delivery network (CDN) based on the Fog computing environment and the Information Centric Networking (ICN) approach. Fog nodes are introduced at the edge of CDNs to reduce overload on CDN servers. Fog nodes in each location are assumed to provide local content delivery services using the network caching feature and the ICN name-based routing approach. As Fog nodes have limited storage and processing capabilities, we used a popularity-based caching strategy in which only the most popular content (MPC) would be cached to avoid the load at nodes without decreasing content availability.

To meet our goal, we modified the Optimized Link-State Routing (OLSR) protocol to be content aware (CA-OLSR). Moreover, we integrated CA-OLSR with the MPC caching strategy. To validate our design principle, we implemented our proposed architecture and major protocol components in a network simulator tool: NS2. Then, we evaluated its performance against two similar architectures. The first was pure Fog-based CDN implemented by the conventional, IP-based OLSR routing protocol along with the default caching strategy. The second was classical cloud-based CDN, also implemented by the conventional IP-based OLSR routing protocol.

Through extensive simulation experiments, we showed that our Fog-based CDN architecture outperforms the other compared architectures. CA-OLSR delivers content with the lowest delay and highest packet delivery ratio (PDR) for all the simulated numbers of connected users. Moreover, its MPC caching strategy showed high cache hit rates with low caching operations. Therefore, we have shown that this architecture is a promising solution for delivering real-time or latency-sensitive applications. In addition, the inherent characteristics of Fog computing can offer high QoS for future Internet applications that will satisfy users QoE.

Our architecture design assumed that the Fog networks are MANET. We also carried out our performance analysis in a Mesh network without considering node mobility. In future work, we will expand our experiments to analyze mobile content delivery.

Finally, although the localized delivery service provided by our architecture reduces delivery latency and inter-ISP network traffic, it involves numerous overheads related to combining caching and routing at a single entity. On the

one hand, caching content at a network entity will consume its memory, bandwidth, and processing resources. This limitation is handled in our architecture by using an MPC caching strategy that caches less content, saving network resources. On the other hand, including content names in the Forwarding Information Base (FIB) table causes a network overhead, resulting from table updating traffic. As future work, we will consider new parameters to minimize the FIB table size.

## Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

This work is funded by the King Abdul-Aziz City for Science and Technology (KACST), Kingdom of Saudi Arabia [Grant no. PS-36-344].

## References

- [1] M. Pathan, R. Buyya, and A. Vakali, "Content delivery networks: State of the art, insights, and imperatives," in *Content Delivery Networks*, pp. 3–32, Springer, 2008.
- [2] G. Pallis and A. Vakali, "Insight and perspectives for content delivery networks," *Communications of the ACM*, vol. 49, no. 1, pp. 101–106, 2006.
- [3] "Cisco Visual Networking Index: Forecast and Methodology, 2016–2021," Zettabyte Era, 2017.
- [4] H. Jiang, J. Li, Z. Li, and J. Liu, "Efficient hierarchical content distribution using P2P technology," in *Proceedings of the 2008 16th International Conference on Networks, ICON 2008*, India, December 2008.
- [5] F. Alghamdi, A. Barnawi, and S. Mahfoudh, "Fog-Based CDN Architecture Using ICN Approach for Efficient Large-Scale Content Distribution," in *Advances in Information and Communication Networks*, vol. 887 of *Advances in Intelligent Systems and Computing*, pp. 685–696, Springer International Publishing, 2019.
- [6] M. Day, B. Cain, G. Tomlinson, and P. Rzewski, "A Model for Content Internetworking (CDI)," RFC Editor RFC3466, 2003, <https://tools.ietf.org/html/rfc3466>.
- [7] H. Yin, X. Liu, G. Min, and C. Lin, "Content delivery networks: A bridge between emerging applications and future IP networks," *IEEE Network*, vol. 24, no. 4, pp. 52–56, 2010.
- [8] F. Bronzino, R. Gaeta, M. Grangetto, and G. Pau, "An adaptive hybrid CDN/P2P solution for Content Delivery Networks," in *Proceedings of the 2012 IEEE Visual Communications and Image Processing, VCIP 2012*, USA, November 2012.
- [9] M. Wang, P. P. Jayaraman, R. Ranjan et al., "An overview of cloud based content delivery networks: research dimensions and state-of-the-art," in *Transactions on large-scale data- and knowledge-centered systems*, vol. 9070 of *Lecture Notes in Comput. Sci.*, pp. 131–158, Springer, Heidelberg, Germany, 2015.
- [10] C. Papagianni, A. Leivadreas, and S. Papavassiliou, "A cloud-oriented content delivery network paradigm: modeling and assessment," *IEEE Transactions on Dependable and Secure Computing*, vol. 10, no. 5, pp. 287–300, 2013.
- [11] X. Guan and B.-Y. Choi, "Push or pull? Toward optimal content delivery using cloud storage," *Journal of Network and Computer Applications*, vol. 40, no. 1, pp. 234–243, 2014.
- [12] M. Pathan, R. K. Sitaraman, and D. Robinson, Eds., *Advanced content delivery, streaming, and cloud services*, John Wiley & Sons, Inc., Hoboken, NJ, USA, 2014.
- [13] B. Ahlgren, C. Dannewitz, C. Imbrenda, D. Kutscher, and B. Ohlman, "A survey of information-centric networking," *IEEE Communications Magazine*, vol. 50, no. 7, pp. 26–36, 2012.
- [14] G. Carofiglio, G. Morabito, L. Muscariello, I. Solis, and M. Varvello, "From content delivery today to information centric networking," *Computer Networks*, vol. 57, no. 16, pp. 3116–3127, 2013.
- [15] A. Detti, M. Pomposini, N. Blefari-Melazzi, and S. Salsano, "Supporting the Web with an information centric network that routes by name," *Computer Networks*, vol. 56, no. 17, pp. 3705–3722, 2012.
- [16] F. Zhang, "Comparing alternative approaches for mobile content delivery in information-centric networking," in *Proceedings of the 16th IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks, WoWMoM 2015*, USA, June 2015.
- [17] D. R. Cheriton and M. Gritter, *TRIAD: A Scalable Deployable NAT-based Internet Architecture*, 2000.
- [18] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, "Networking named content," in *Proceedings of the 5th ACM Conference on Emerging Networking Experiments and Technologies (CoNEXT '09)*, pp. 1–12, December 2009.
- [19] J. Chen, H. Xu, S. Penugonde, Y. Zhang, and D. Raychaudhuri, "Exploiting ICN for efficient content dissemination in CDNs," in *Proceedings of the 4th IEEE Workshop on Hot Topics in Web Systems and Technologies, HotWeb 2016*, pp. 14–19, USA, October 2016.
- [20] W. You, B. Mathieu, and G. Simon, "How to make content-centric networks interwork with CDN networks," in *Proceedings of the 2013 4th International Conference on the Network of the Future, NoF 2013*, Republic of Korea, October 2013.
- [21] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things," in *Proceedings of the 1st ACM Mobile Cloud Computing Workshop, MCC 2012*, pp. 13–15, Finland, August 2012.
- [22] S. Yi, Z. Hao, Z. Qin, and Q. Li, "Fog computing: Platform and applications," in *Proceedings of the 3rd Workshop on Hot Topics in Web Systems and Technologies, HotWeb 2015*, pp. 73–78, USA, November 2015.
- [23] T. H. Luan, L. Gao, Z. Li, Y. Xiang, and L. Sun, "Fog computing: focusing on mobile users at the edge," *Networking and Internet Architecture*, 2015, <https://arxiv.org/abs/1502.01815>.
- [24] M. Abolhasan, T. Wysocki, and E. Dutkiewicz, "A review of routing protocols for mobile ad hoc networks," *Ad Hoc Networks*, vol. 2, no. 1, pp. 1–22, 2004.
- [25] L. M. Vaquero and L. Rodero-Merino, "Finding your way in the fog: Towards a comprehensive definition of fog computing," *Computer Communication Review*, vol. 44, no. 5, pp. 27–32, 2014.

- [26] I. Stojmenovic, "Fog computing: A cloud to the ground support for smart things and machine-to-machine networks," in *Proceedings of the 2014 Australasian Telecommunication Networks and Applications Conference, ATNAC 2014*, pp. 117–122, Australia, November 2014.
- [27] L. Breslau and P. Cao, "On the Implications of Zipfs," in *Proceedings of the 3rd International WWW Caching Workshop*, p. 11, 1998.



**Hindawi**

Submit your manuscripts at  
[www.hindawi.com](http://www.hindawi.com)

