IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 17 March 2023, accepted 12 April 2023, date of publication 17 April 2023, date of current version 9 May 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3267737

SURVEY

A Review on Software Defined Content Delivery Network: A Novel Combination of CDN and SDN

HUIXIANG YANG[1], HANLIN PAN[2], AND LIN MA[3]
[1]School of Mechatronic Engineering, Changchun University of Technology, Changchun 130012, China
[2]College of Business, City University of Hong Kong, Hong Kong, China
[3]College of Computer Science and Technology, Jilin University, Changchun 130012, China

Corresponding author: Lin Ma (malin20@mails.jlu.edu.cn)

This work was supported by the Scientific and Technological Developing Scheme of Jilin Province of China under Grant 20200404201YY.

ABSTRACT With the rapid growth of multimedia services, the burden of infrastructure is becoming heavier, which may not guarantee a high-quality experience for users. Whether the multimedia category is video on demand (VOD), live broadcast, or social media is a hot issue. Traditionally, content delivery network (CDN) service providers deploy servers as close to clients as possible to reduce latency. CDN is an overlay network mainly responsible for routing requests, distribution, delivery, and audit. As a supplement to computing and storage capacity, CDN service providers have begun migrating some of their services to the cloud to focus on the delivery process. At the same time, the collaboration between CDN service providers promotes scalability to meet the growing number of requests. However, the underlying physical layer needs to be improved. The cost of the traditional underlying network is very high and cannot be easily expanded. The concept of software definition network (SDN) is proposed to solve these problems. SDN functions are divided into control layer and data layer. In the SDN, the global view is achieved to receive information by centralization, and then the deployment is dynamically adjusted to control the network. In this paper, we summarize the advantage structure and problems of a newly developed system SDCDN (software defined content distributed network). We introduce similar research progress. Then we summarize the current situation of SDN and CDN, compare and analyze their advantages and disadvantages, and finally put forward the directions of open issues and the future research.

INDEX TERMS Content delivery network (CDN), multimedia service, network architecture, routine, software defined network (SDN).

I. INTRODUCTION

With the goal [1], [2], [3] of providing fast, reliable, and available distribution services, content delivery network (CDN) has developed rapidly from the beginning of 2000. Thanks to the architecture of the CDN in Fig.1, it is a network of servers that are distributed globally to deliver content efficiently to users. The primary goal of CDN is to reduce the latency and improve the user experience by delivering content from the server closest to the user. When a user clicks on a content on the app, the app requests the IP address for the URL from the local DNS (Domain Name System) system. The local DNS system directs the domain name resolution to a CDN-private DNS server, which returns the IP address of the CDN's global load balancing system to the user. The user then sends a request for the content URL to the CDN's load balancing device. Based on the user's IP address and requested content URL, the load balancing device selects a cache server located in the user's region. The device provides the user with the IP address of the selected cache server and directs them to send the request to that server. The selected cache server responds by delivering the requested content to the user's device. If the cache server does not have the requested content, it requests the content from the origin server, then returns the content to the cache server. After that, the server delivers it to the user and determines whether to cache the content based on the user's caching policy. This process allows CDN providers to deliver content more quickly and efficiently to users by

---

The associate editor coordinating the review of this manuscript and approving it for publication was Gangyi Jiang.

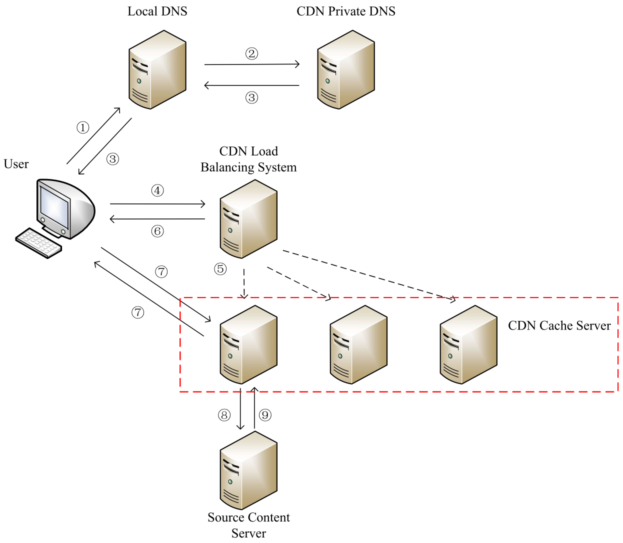


FIGURE 1. **Workflow of CDN.**

storing copies of popular content on servers which are closer to the users. CDN uses a technique called load balancing to distribute the traffic among the servers in the network. This ensures that the load is distributed evenly among the servers, and there is no single point of failure. CDN also uses various optimization techniques to improve the performance of the website. For example, it uses techniques such as minification, compression, and caching to reduce the size of the website's content and improve the page load time.

There are several network architectures dedicated to content distribution. Information-centric networking (ICN) is a networking paradigm that emphasizes content rather than its location. In ICN, data is addressed by content name rather than by host or location. ICN aims to enable efficient and secure content distribution by leveraging in-network caching and multicast. ICN is designed to handle the growing demand for content distribution and to address the limitations of traditional IP-based networking. Named Data Networking (NDN) is an architecture built on top of ICN principles. NDN is a data-centric approach to networking that aims to replace the current host-to-host communication model with a content-centric model. In NDN, every piece of data has a unique name, and routers use this name to route requests and responses. NDN is designed to provide more efficient and secure content distribution by leveraging in-network caching and multicast. Overlay networking refers to a network topology where a virtual network is built on top of an existing physical network. The virtual network can be used to provide additional functionality, such as security or quality of service (QoS), that is not available in the underlying physical network. Overlay networks can be created using various techniques, such as VPNs or software-defined networking (SDN). Peer-to-Peer (P2P) Networking is a type of network where nodes in the network communicate with each other directly, without the need for a central server. In a P2P network, every node can act as both a client and a server. P2P networks can be used for content distribution, file sharing, and other applications. One advantage of P2P networks is their resilience compared to traditional client-server networks since there is no single point of failure. In summary, CDN, ICN, and NDN are all content-centric networking architectures that focus on efficient and secure content distribution. Overlay networking and P2P networking are more general network topologies that can be used for a variety of applications.

CDN has various applications in improving website performance: 1. Improved website performance. CDN improves website performance by caching content closer to the end-users, reducing latency and improving page load times; 2. Content delivery for streaming services. CDN is widely used for content delivery for streaming services such as video and audio streaming. CDN ensures that the content is delivered to the user in real-time with minimal buffering; 3. Software distribution. CDN is used for software distribution by hosting and delivering software updates, patches, and other software-related content; 4. E-commerce: CDN can improve the performance of e-commerce websites by caching product images and other product-related content; 5. Cloud computing: CDN is used for cloud computing by hosting and delivering cloud-based content such as data, applications, and APIs.

However, there are still some challenges that CDN faces. Firstly, the cost of CDN's construction is still high. Setting up and maintaining a CDN can be expensive, especially for smaller websites; secondly, it is difficult to guarantee content consistency. CDN caching can result in outdated or inconsistent content being delivered to users; thirdly, CDN can increase the attack surface of the website, and vulnerabilities in the CDN network can result in security breaches; fourthly, the configuration of CDN is complex. CDN requires proper configuration and management to ensure that the content is delivered efficiently and consistently. In order to solve these problems, the concept of software definition network (SDN) is proposed. SDN is a network architecture that separates the network control plane from the data plane, enabling administrators to dynamically manage and optimize network resources. Here is a brief description of the SDN architecture, working, and advantages through Fig.3.

The SDN architecture consists of three main components: the control layer, data layer, and application layer. Control layer contains the SDN controller, which communicates with the network devices and manages the network's routing policies and configurations. The controller also receives information about the network's topology and traffic patterns from the data layer. The data layer includes the network devices that forward data traffic based on the decisions made by the SDN controller. These devices can be traditional switches and routers or more specialized devices designed for SDN environments. The application layer includes various network services and applications that can run on top of the SDN controller. Examples include network monitoring, load balancing, and security applications. In an SDN network, the

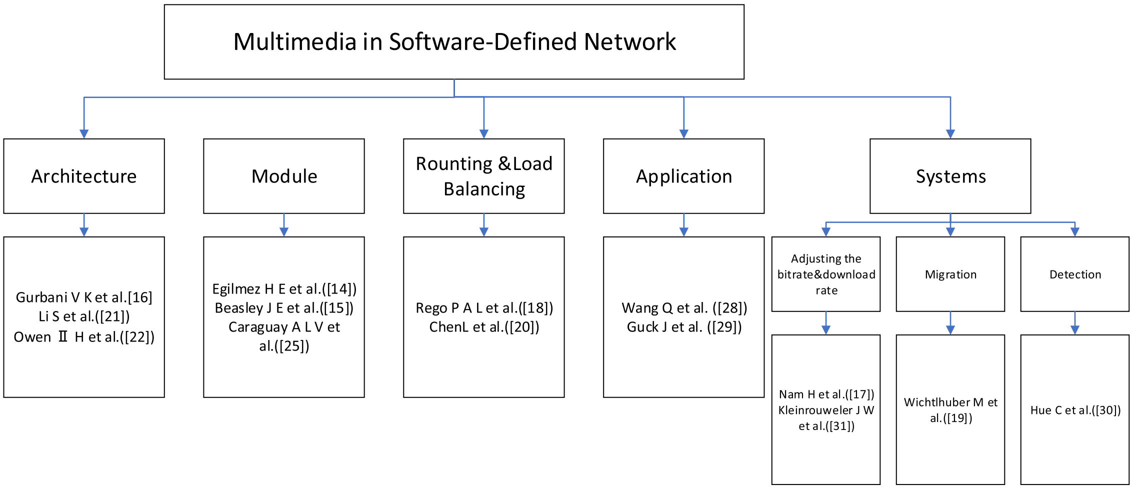


FIGURE 2. Classification of the related work for multimedia in software-defined network.

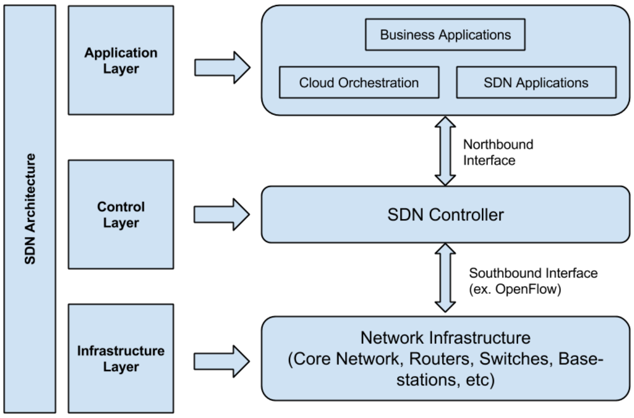


FIGURE 3. Architecture of SDN.

control layer and data layer are separated, and the control layer is managed by a centralized controller. The controller communicates with the network devices, such as switches and routers, through a protocol such as OpenFlow. The controller defines policies and rules for traffic management, and the network devices implement these policies by forwarding traffic according to the rules defined by the controller.

SDN has some advantages as following: simplified network management. SDN simplifies network management by providing a centralized view of the network and enabling administrators to define policies and rules that can be easily implemented by the network devices; improved network agility. SDN enables network resources to be dynamically allocated and optimized, improving network agility and flexibility; enhanced security. SDN provides a centralized view of the network, enabling administrators to monitor and control traffic more effectively, thereby improving network security.

Considering an example of SDN used by a large e-commerce website. The website experiences a sudden surge in traffic during a holiday season, and the network is unable to handle the increased traffic load, resulting in slow page load times and a poor user experience. To address this issue, the network administrator uses a SDN controller to dynamically allocate network resources and optimize traffic flow. The controller analyzes traffic patterns and identifies bottlenecks, and then dynamically allocates resources to handle the increased traffic load.

By adapting a variety of new platforms and technologies, and putting some functions to the newly introduced modules, such as transcoding and computing by using the large-scale computing processing capacity of the cloud platform, accurate control and flexible feedback of the underlying network data by using SDN, CDN can meet the current development needs and facilitate the solution in combination with the underlying network status and other factors. The software defined content distribution network is a new distribution network architecture constructed by coupling CDN and SDN. This architecture integrates storage, computing resources and networks to improve the service performance of end-to-end applications [124], [125].

Since only a few literatures have mentioned the structure of coupling SDN and CDN, we have summarized the existing SDCDN methods and sorted out the general structure. We have analyzed and summarized this structure in many aspects: the research status and problems. Our mainly purpose is mainly improving the quality of multimedia services by considering the situations from the two aspects to improve the QoS/QoE. The information could be shared more convenient so that to make the adjustment suitable. From the view of the CDN, the routing information we get is more complete than requiring indirectly from the side of the clients or servers. From the perspective of the SDN, the clients' experience or the servers' state could be known to design a better path for the delivery. Although it will take new problems, the two advantages added to one architecture is meaningful for the latest trend. Through this paper, we will introduce the researches and evolution of the SDN and CDN, then we will introduce the current situation and future direction of the SD-CDN architecture by 8 categories: scenarios, architecture, routine deployment, interoperability between the SDN and CDN, the applications of machine learning algorithms in SDN-based CDN, standardization, and cross-cutting approaches. Over this process, more than 140 related papers are reviewed.

Many excellent surveys [102], [103], [104], [114] have aimed at the depth of one domain such as routing, deployment and architecture. In the case, our research is in a relatively new field, so except to talk about the related work as the evolutionary, the directions for the exploration from various angles are the focus in this paper. Thus, the framework will be more practical by using the plans through the exploration. Then, we clarify the related work as the categories with time elapsing in Section II, and conclude in Fig.2. Section III summarizes the general architecture and the procedure of the work. Combined with the current development respectively, we analyze the problems to give directions of the future researches. In section IV, we introduce the potential research directions to optimize the SDN-based CDN. Section V concludes the paper and discusses our future work.

II. RESEARCHES AND EVOLUTION

A. CONTENT DELIVERY NETWORK

There have been many changes in the development of CDN: from a single CDN to cooperation among CDNs; From the basic CDN to the new mode of developing P2P to the content distribution network of mixed structure. Now, when we mention CDN, we have defaulted to include P2P. In the current research, the new architecture based on CDN is still being explored.

First of all, regarding the definition and architecture of inter CDN, IETF first proposes the concept of CDI [80](content delivery inter). Turrini [81] describes, deploys and evaluates the architecture of CDI, allowing each CD to expand the number of potential users in CDI, so that users can obtain content more quickly. Generally, the system is constructed based on star topology and in a semi centralized manner, that is, there is an authoritative CDN in a certain number of CDNs that is responsible for registration, request transfer, etc. in this set, and the protocol in the CDI structure is called rieps (routing IEP for star topology). Sarma and Setua [116] proposes a hybrid cdni on the cdni architecture. Compared with a single cdni, its search time and content transmission time are shorter. At the same time, the classic problems in the content distribution network, such as request routing and content distribution, are redesigned, analyzed and tested under the new architecture. Buyya and Pathan [82] proposes a peer-to-peer interconnected CDN system based on VO (Virtual Organization), which is called CSDN system. There are individuals and groups sharing resources in a VO. The components of the VO network include the service registry function and the VO scheduler responsible for the coordination function. Participants register through the service registration module and access resources according to the established rules.

As for each individual CDN in the inter CDN, the independent and universal content distribution network architecture has always been the focus of academic research. After eight years of research on the content distribution network architecture, Ganjam and Siddiqui [58] divides the control layer of CDN into two parts, one is the decision-making layer and the other is the model layer. The lower layer is responsible for collecting local real-time data and making some decisions, and then uploading the data to the upper layer to make near optimal global decisions from a global perspective, and then act on the local part to adjust. That is, from the fine-grained decision algorithm to the coarse-grained calculation, and then to the fine-grained level. This not only ensures the strategy for individual diversity, but also deploys the overall situation. Alasaad and Gopalakrishnan [59] creates a helper role to assist the content distribution network in order to alleviate the computing capacity of the edge server, reduce the purchase and maintenance costs in the content distribution network, as well as the bandwidth and energy structure of the ISP. When the ability of the server to process the request is insufficient to meet the user's needs and the QoS index is affected, the edge server delivers the content to the helper, and the helper provides the content delivery for the user. Ma and Wang [60] push the devices that cache content closer to the user, change the basic CDN architecture, redesign the workflow and protocol between the CDN edge server and the more "close" smart router, including how to schedule content and how to cache. Meanwhile, the paper also discusses the relationship between intelligent routing and video on demand rate, the response speed of replicas, and the indicators that affect content caching.

Deep into the characteristic of roles in CDN, the content is owned by Content Provider; the content needs the delivery service provided by the CDN service provider. Delivery service contains not only the servers but also the network layer. The CDN service providers pay fees for the underlying to ISP such as China Mobile, and China Unicom. The service object is the client. Wichtlhuber and Reinecke [19] analyze the advantages of CDN service providers and ISP. It has the detailed topology information and the direct information for the state of networks as an ISP, while the CDN service providers possess the state of servers and the strategies for redirecting. The cooperation between CDN service providers and ISPs is a better solution for flash crowds and congestion problems. Meanwhile, the paper also refers to the latest research and gave an improved plan for migrating the redirected requests, and this resulted in the seamless procedure of redirecting to the suitable servers in CDN.

However, with the development of the Internet, in the face of these rapidly changing business needs, CDN has encountered a new challenge: its response speed seems inadequate. In addition, it also faces the problems of complex operation and maintenance management and low resource utilization. The design idea of separating SDN control from forwarding and the open, flexible and programmable features provide a new way to solve the problems faced by the development of CDN. Simply put, CDN solves the problem of slow user access response due to small bandwidth of backbone network, large user access and uneven distribution of network points by caching content near the edge of the network and dispatching user requests to the optimal server through load balancing technology. Faced with rapidly changing Internet applications and business needs, CDN supports fast deployment and high scalability of business.

One possible solution is to move toward pipelining. The core idea of pipelined CDN is to reduce the CDN from the application layer of IP to the transport layer, which is equivalent to the content transport layer overlaid on the IP network. At present, China Telecom [127] CDN 3.0 has achieved loose coupling between business system and CDN, standardized interface, and initially reached the goal of pipelining. However, it still faces some technical challenges and problems: 1. Unable to quickly respond to the new business requirements, such as content encoding format, transmission and service protocols, terminal types, and business requirements differentiation are constantly changing. Existing architectures and business processes make CDNs unable to respond to these changes in demand. 2. Operations and management of complex CDNs mostly use proprietary hardware devices, or use generic hardware but require dedicated installations. Faced with large-scale applications, complex business logic and difficult operation and maintenance management, CDN operators have high complexity and high operating costs. 3. Low utilization rate of resources in idle time. Device processing power is usually designed to meet peak performance requirements, and these redundant capabilities cannot be opened when idle. Peak average utilization and resilience of CDN nodes are lower than those indicated in hardware resource sharing mode.

B. SOFTWARE DEFINED NETWORK

At the beginning of multimedia popularity, the purpose of service is only hop-by-hop delivery through the traditional networks, which is the best-effort mechanism and does not require quality. Due to the large amount of traffic and the guarantee for quality, the emergency of the CDN is widely used, which is an overlay network. The reason for bringing SDN into the field is also striving for the better quality of service. Egilmez and Dane [14] modify the algorithms in the controller module of the SDN so that the delay would decrease, which was an important metric in the multimedia field. The modules in the controller layer of the SDN could be subdivided into six categories, which are the topology management, routing management, flow management, routing calculation, emergency control, and flow regulation. The routing management is responsible for storing the information about the link capacity, the state of the link, and so on. This mainly aids in the module of routing calculation. The common routing algorithm is the constrained shortest path algorithm [15].

Then, the development is not limited to modifying the existing module. The architecture of the SDN will be reconsidered. The classic architectures which are specific to QoS are IntServ [23] and DiffServ [24]. The choice of path is in line with the routing protocol in the IntServ. It is easy to make the decision, but the disadvantage is that the result is not optimal, and the dropping packets may occur in the process. The DiffServ improves the shortage of scalability in the IntServ, nor the quality of end-to-end service. Owens II and Durresi [22] design an architecture based on SDN, and analyze the protocol, applications, then evaluate the proposed architecture called VSDN compared with IntServ and DiffServ. The VSDN takes the actions to offset the MPLS, which allows the dynamic configuration for routing path, and plays a role as a packet shaper by setting a traffic specification module on the edge switch.

Although the field of multimedia has the universal property, the different applications of multimedia have the fine-sorted characteristic. The live streaming has higher demand on delay, and others may focus more on the resolution. Paraguay and Fernández [25] propose three flow models for different multimedia services. The QoS parameters module is designed for distinguishing the applications, and the parameters contain the type of data structure, priority, QoS level, and devices. Then, the parameters will be sent to the module of routing management. The routing management module could be subdivided into three parts that are device management, topology management, and flow management. The device management confirms and tracks the location for devices based on the MAC address and IP address, the topology management deal with the information about the capacity of the link and switch, and the flow management issues the flow tables to the switches according to the information of path and priority. The research has developed from only modifying the modules to adding the modules to improve the performance.

Generally speaking, the architecture of SDN mainly consists of three layers: application layer, control layer and infrastructure layer. Usually, the functions of the application layer are divided into three main parts: collecting data, processing data, and issuing decisions. Then the application layer connects to the control layer through the north interface:The control layer uploads data to the application layer for analysis and decision making. The application layer sends decisions to the control layer. The control layer sends data flow to the infrastructure layer according to the decision making, and performs corresponding operations on the infrastructure layer. The control layer controls the infrastructure layer in a logically centralized way. It is the hub connecting the application layer and the control layer. It can obtain the global information of the lower network, and is responsible for maintaining the network topology, state information and so on. On the control layer, the controller is the main component of the control layer. Controllers are connected through east-west interfaces inside the control layer. The control layer receives network data, device status and other information obtained from the data forwarding layer, and sends instructions as data streams to the infrastructure layer through the south interface. The infrastructure layer does not have control logic. After receiving the rules for the control layer transfers, information is transferred through the data flow. The architecture of SDN can be seen in Fig.3.

Next, the interaction between SDN and CDN is closer, and the information will be shared to make a better decision for routing strategies and so on. Gurbani and Scharf [16] introduce ALTO into the architecture based on SDN. The definition of RFC7285 explains the service of supporting the optimal plan for the flow traffic in the application layer and provides more information about the network, such as maximal bandwidth, minimal crossing domain, and cost. Li and Doh [21] promote the architecture for the mixture of SDN and ALTO. The application scene is specific to CDN. So the process is that the ALTO servers receive the information from SDN, then send it with self-information to the redirected cluster of CDN servers. The cluster would choose a suitable edge server to deliver the content to clients. Meanwhile, the security problem was also considered.

Through the precise positioning of ISP subnets, links, and nodes by monitoring of SDN, the limited value of the buffer would be calculated, Nam and Kim [17]recommend the optimal download rate dynamically combined with metrics of QoE in case of congestion. This characteristic of SDN also will be a benefit in load balancing due to its global view. Rego and Bonfim [18] supply the function of load balancing in the Openflow by defining the threshold, which decides when to add or remove the servers in the cloud to achieve the balance.

The routing algorithms consider both improving the QoE and guaranteeing the load balancing as mentioned above. Chen and Qiu [20] propose two routing algorithms: bandwidth-aware video content selection and the trigger of the threshold for load balancing into the algorithm. When the overload occurs, the content would be replicated to another server in accordance with its capacity.

Although the field of SDN for video is mainly limited in the range of routing and load balancing, the scope has been extended into dynamic adaptive streaming HTTP. Hue and Chen [30] utilize the dynamic programming of SDN. The dash player initialize and chose the bitrate lower than the aiming to deliver through the adaptive algorithms. Once the value of the buffer triggers the threshold, the bitrate would turn into the aiming until stable. Then, the service manager could change the bitrate to call other modules again.

C. CONCLUSION OF CURRENT SITUATION

By integrating network, storage and computing resources, the software defined content distribution network makes up for the incomplete and inaccurate acquisition of the underlying network information by the CDN. At the same time, the underlying network resources can be dynamically deployed through programming to assume the tasks and functions of indirectly acquiring network information by the CDN. This not only reduces the load of the CDN, but also strengthens the control of the underlying network. At the same time, sdcdn transfers some computing functions to the cloud platform for processing, so as to focus more on strengthening the core responsibilities of decision-making such as server deployment, cache allocation, redirection routing and pricing in the software defined content distribution network. In combination with its own ability to collect application layer information in the distribution network, sdcdn provides good services to users.

In general, most research in this field pointed to how to improve the current state but seldom talked about when to change it. Uzakgider and Cetinkaya [26] mainly concentrate on this based on the Markov decision process and information of network provided by SDN and Liu and Ding [105] propose to solve this using a constrained reinforcement learning-based resource allocation algorithm. Under the circumstance of maintaining the quality of service and decreasing the switching routing path, the timing is calculated by the learning-based algorithms. The rewards function in the algorithm returned results to adjust the timing.

Although the topics about how to deal with the different kinds of video streaming have been studied, how to differentiate whether it is a video is still unsolved. McKeown [5] propose an inspection-free traffic-aware technology compared to DPI and inspected the matching level with classic video pattern through the growth rate of download and the total traffic. Then, the cycling of the ‘on-off’ pattern is checked. The two parameters together concluded whether the pattern was.

Due to the benefit of SDN and CDN, it is used in various application scenes. Wang and Xu [28] optimize the kernel forwarding tables to achieve the seamless upload and download of videos on the campus. The design for gateways and video servers is categorized into three parts to satisfy the seamless technology, which were front-end connecting and choosing, back-end forwarding, and the bridge of OpenVswitch. Guck and Reisslein [29] add queue links model to assign the resource, the most critical issue is to improve the control loop through the network calculus and a delay-constrained least-cost routing algorithm to guarantee the hard real-time of QoS in the industrialization.

Wu and Hou [44] divide the workflow for transmission of multimedia into six-step steps, which involve compression, controlling QoS in the application layer, continuously distributed service, flow servers, synchronization, and protocols. Specific to CDN, the service is provided among flow servers to guarantee QoS and even QoE, which include storage servers, index servers, and edge serves. The technologies contain filtering, broadcast, replica, and fault tolerance to provide complete distribution.

III. A GENERAL REFERENCE ARCHITECTURE

With the consideration of the architecture for SDN and CDN, the design for SDCDN is in the light of functions to divide into five layers, from the top to bottom which includes

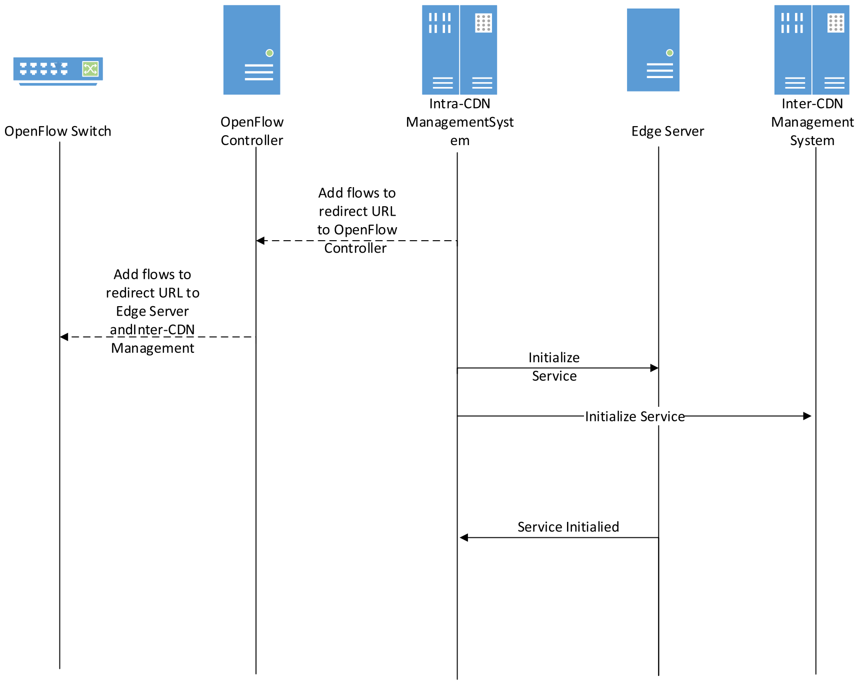


FIGURE 4. A workflow process of the SDCDN:initialization.

Inter-CDN management layer, Intra-CDN management layer, SDN controller layer, the layer for CDN edge servers and data layer. The first layer is charged for cooperation including the disposition of revenue, coordination and protocols. The second decides how to choose the edge servers, placement, and cache whose jobs is the same to the independent CDN management. Then, the communication between CDN and SDN depends on the northern API of SDN which connects with the upper application CDN, so that to the share information and make the decisions to direct the data layer.

Initialization, the inner adjustment starts to deploy first in the SDCDN in Fig.4. The register module records the cooperative providers and marks the policy between them in the Inter-CDN management. The Intra-CDN management decide how to choose the edge server through the preference, cost, performance, popularity and so on, the emergency like congestion and flash crowds is considered into it due to it occurs more often, so the dynamic adjustment needs urgently. And it also decide whether to need the other providers' help. Then, it issues the order to the SDN controller through the northern API, so that to give a direction about the forwarding tables. After that, the SDN controller communicates with the data layer through the southern API in accordance with Openflow protocol.

The more detailed description in Fig.5. When the requests sponsored by the clients, the switches modify the head of the packet to direct it to the Intra-CDN cluster. The server for redirecting will find a suitable cache server for the client to get the content, the suitable cache server is chosen through the information of Intra-CDN and SDN, then how to transfer the packets of requests depends on the data layer. So the Intra-CDN layer communicates with the SDN controller to inform the decision, the switches forward the packets to the destination according to the decision. The packets arrived the cache server, the necessary check will be performed to look for whether a needed cache exits. If it is in the cache server, then the content will be delivered to the client. Or else, it will notify the origin server to pass the content to this server, then the server could deliver the content to the client. Kleinrouweler and Cabrero [31] illustrate the shortage of this way, so the client could receive the content from the origin server directly.

Then, the next step is to consider that how to evaluate and choose the indexes in SDCDN, the aspect for how to achieve the standard from different angles as mentioned before. We will add the control logic and strategies into the framework SDCDN, then we take suitable indexes to measure the performance. Some researches has studied the characteristics of these indexes. Maia and Yehia [63] summarize the evaluation into three categories which were subjective, objective and hybrid. The subjective evaluation contain single stimulus, double stimulus, the objective evaluation contain full-reference metric, reduced-reference metric and so on, the hybrid contained RNN, regression mode, and etc. Through the MSE and PCC, it proves its precision. Juluri and Tamarapalli [64] classify four levels to evaluate QoE including system level, environment level, user level and content level according to the delivery technologies. Bouraqia and Sabir [106] sum up five methods to measure the QoE. The mainly technologies involve real-time video streaming,

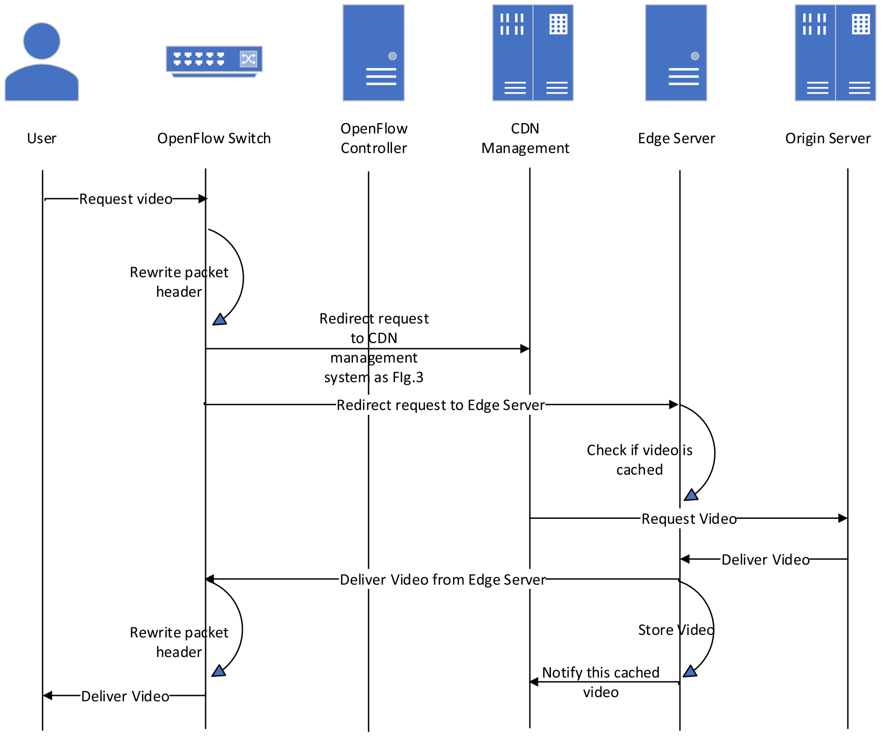


FIGURE 5. A workflow process of the SDCDN:steady process.

real-time messaging protocol streaming, progressive download streaming, and HTTP-based adaptive bitrate streaming. The indexes of QoE in the system level were just like dropping rate, delay, the indexes in the environment level were location, in the user level contain preference and history logs and in the content level were bit rate and visual quality. How to measure these is also introduced in this paper. Through the information which is measured, the related work applied for the realistic scenes to adjust the state of network dynamically.

We have witnessed the latest researches in the cooperation field of multimedia and software defined network. To the best of our knowledge, this work is the first attempt to systematically introduce the complete framework for SDCDN.

Due to the benefits involved much, we not only connect the network layer to the application layer, but also consider the cooperation between CDN service providers. There exits the exploration for cooperation. Courcoubetis and Gyarmati [89] maximize the benefits of ISP and CSP, and trade off the transit cost savings and monetization of QoE from cost, profits and churn to talk about. Then, it proposes a plan for paying and supplying clients with bandwidth. Pathan and Buyya [90] explain the cooperation between CDNs to solve the congestion and flash crowds, and build the utility model to expose the usage benefits. Then, it designs an algorithm to choose the least load of servers and proves it through the response time, fault rate and so on. Liu and Yang [107] propose an cooperation mechanism to solve the loading balance issue in mobile end.

Through the trace-driven or data-driven approaches in the system presented in Section III-F the decision is derived from three parts. Firstly, it considers the circumstance of its own service provider. Secondly, it contains the results of the negotiation from cooperated service providers. Finally, it takes the network situations as the reference. Then the decision will be issued from the management module. The strategies will transform to the readable orders in the part discussed in Section III-D and the routing in Section III-B, and the deployment in Section III-C. Then, the equipment of the network layer will do the actions as the strategies.

Of course, the framework is also inspired by the Section III-B, and takes advantage of the CDN and SDN. In the meantime, the functions such as Section III-C could be easily implemented without huge change. As the basis of the framework, the related topics continue to explore, and then categorized into different aspects, where we will introduce the current situation and the future directions.

A. ARCHITECTURE

With the current requirements of the environment, the architecture has evolved gradually, as shown in Table 1. From the bottom layer, scalability becomes a problem due to the centralization in SDN. The researchers have studied the distributed model of the controller for the large-scale scene. Lange and Gebert [41] design an optimal deployment for controllers considering the delay, fault tolerance, and load balancing, then solve the problems through heuristic

TABLE 1. Summary of the architecture domain.

Paper	Cooperation	Structure	Addition	Objective
Ganjam A et al. ([58])	No	centralized	Fine-granularity, Decision layer && Model layer	Diversity, global view
Alasaad A et al. ([59])	No	centralized	The role of helper	Decrease the cost for buying and maintaining
Ma M et al. ([60])	No	centralized	Smart routers	Decrease the response time
Turrini E ([81])	Yes	semi-centralized distributed	The complete process	Lower delay service with higher quality
Buyya R et al. ([82])	Yes	semi-centralized distributed	The complete process(protocol, mechanism…)	Better cooperation service
Mowla N I et al ([95])	Yes	centralized	Improved the CDNI	Improve the security of the network

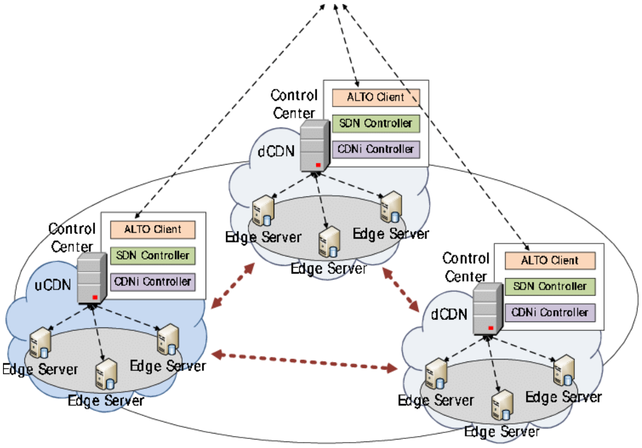


FIGURE 6. Architecture of CDNI.

algorithms. And Li and Guan [108] design a parameter optimization model based on heuristic algorithms.

The upper layer always has more variations than the network layer. Although the upper layer does not exit the bottleneck, cooperation is the trend for scalability. And The Inter-CDN layer in SDCDN is an example for this. IEIF defined the conception of CDNI (content delivery networks interconnection) [80] as shown in Fig.6 CDNI (Content Delivery Networks Interconnection) is an IETF standard that defines a series of interfaces. This standard enables upstream CDNs (uCDNs) to delegate requests to downstream CDNs (dCDNs). Open Caching is a special use case of CDNI. Content providers or commercial CDN servers are generally used as uCDNs, while a caching layer in an ISP is used as a dCDN. CDNI needs to define how to access the content required in the content distribution process such as source, authentication rules, and transmission through configuration file metadata. In Open Caching, metadata is included to define the request routing method, the capacity in the transmission process, the cache capacity and other information. In a content distribution system, efficient CDN metadata exchange between different parts is the key to achieve efficient interoperability.

Turrini [81] analyzes the architecture of CDNI from its outline, implementation, and evaluation. The advantages of cooperation satisfy more needs from clients, and the outstanding is more closet servers to choose from so that it could provide lower delay service with higher quality. In general, the architecture of CDNI takes a semi-centralized distributed manner based on a star topology, and the protocol is called RIEPS (Routing IEP for Star topology). Namely, there is a role of traditional CDN among the cooperative parts to do the register, passing, and so on. Mowla and Doh [83] improve the CDNI security by using a model that all possible forms of DDoS attack can be classified and defended. This model can significantly improve the accuracy of detecting the DDoS by using SVM and logic regression. Thus the security of the CDNI can be enhanced.

As for another architecture for cooperation CDNs, Buyya and Pathan [82] propose a Virtual Organization peering CDN called CSDN. CSDN guarantees the quality of service based on SLA negotiation, and solves the problem of logic separation. A virtual organization (VO) model is proposed to form csdn, which not only shares Web servers in its own network, but also shares with other csdns. In order to encourage different CDN providers to carry out continuous resource sharing and peer-to-peer arrangements at the global level,they use the market-based model in resource allocation and management. This inspiration comes from their successful use of autonomous resource management, especially in the global network. Fig.7 shows the architecture of such a system, and the following sections describe its main elements. The module for service registry and coordination part exits in the VO, and the VO contains the protocol and sharing information for individuals and groups.

If we consider the single CDN, the architecture is changing and explored by many researchers now, from our familiar hybrid CDN and P2P to the one we will discuss. Ganjam and Siddiqui [58] separate the CDN into two parts. One is the decision layer; the other is a model layer. The local information is provided for the decision layer to conclude a real-time result, then uploaded to the model layer to make an optimal global decision, then the decision will act on the local. This ensures the diversity of the policy for individuals and gives a global view. Alasaad and Gopalakrishnan [59] relieve the pressure of edge servers in CDN so to decrease the cost of buying and maintaining, and then they created a role of helper to aid the basic architecture when the number of edge servers is not enough to guarantee the QoS, the helper will be the direct transmitter. Ma and Wang [60] add the smart routers into the basic architecture of CDN, and this will push the content closer to the clients. This research introduces the protocols, workflow, scheduling and cache strategies for the smart router, then evaluates the response time and explains the reasons for influencing the cache.

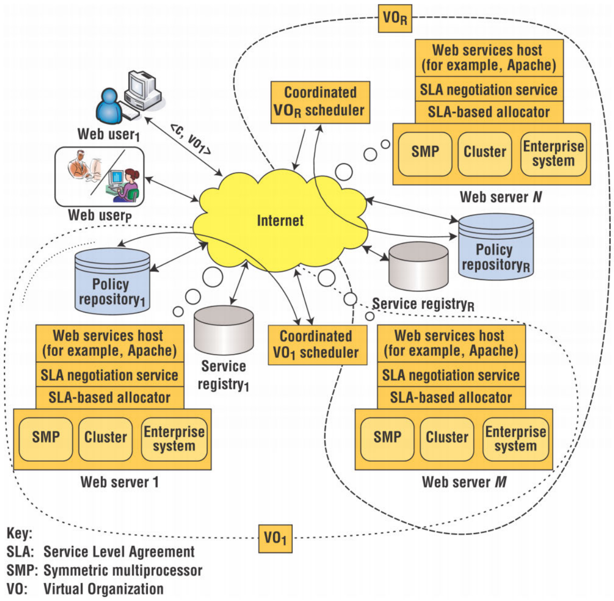


FIGURE 7. Architecture of CSDN.

The framework SDCDN considers the cooperation between CDNs, and the scalability of the software-defined network. It supports the extension and needs no modification for the general framework. Xinli and Cheng [99] address the advantages of combining CDN with SDN. Ibrar et al. [137] propose a sdn-based cdn architecture aiming to decouple the control plane from the forwarding plane for flexibility and programmability in using the centralized controller. Tran et al. [19] propose a architecture that promote the collaboration between ISP and CDN via SDN.

Generally speaking, SDCDN structures can be divided into two categories: centralized structures and distributed structures. With the increasing traffic, distributed architecture is becoming more and more important. A feasible research direction is the combination of cdn and blockchain:First of all, the application scenarios of blockchain are generally limited to multi-layer and multi range fields, which is consistent with the three-tlayer architecture of content distribution network and the characteristics of large coverage. Secondly, blockchain technology can fundamentally solve the coupling problem between CDN architecture systems and business systems, as well as between platforms at all levels. Finally, the characteristics of blockchain decentralization and openness provide a theoretical basis for building a blockchain based CDN platform.

B. ROUTINE

The purpose of routing is to respond quickly so that to optimize the forwarding path. Due to the characteristic of SDN, the routing path is programmed to change dynamically. Thus, the parameters to adjust the path appear more. This is also the combination problem to give an optimization plan.

The current work in Fig.8 starts with traffic engineering and optimization algorithms such as load balancing. First, specific to SDN, Al-Fares and Radhakrishnan [35] propose a flow scheduling system as Hedera to deal with the scheduling for multi-level switches and compute the non-conflict paths for the flow. Ishimori and Farias [36] transform the FIFO for dealing with packets to the plan for multi-packets based on a Linux kernel called QoSFlow. Selin and Tekalp [37] propose a p2p-CDN service model to save energy by the multi-access edge data center. It finds the minimal power among clusters through monitoring, then assigns the work to link and switch dynamically to satisfy the changing traffic. Sarkar and Rakesh [109] and Zhang and Ma [110] design different algorithms based on various machine learning methods to form a hybrid model to solve the load scheduling issues.

For the cross-domain problem, the problem is classified into IXP [74], [75], [76]. Joshi and Kataoka [79] propose an end-to-end QoS multi-domain mechanism where only binary decisions are reported instead of complete ones to ensure privacy. The underlying architecture consists of distributed SDN controllers, and it introduces two plans for the controller layer to ensure scalability and security.

Load balancing is not a problem to be ignored in many fields. Shi and Li [32] develop a load balancing strategy based on the Openflow protocol. It can classify elephant flows accurately and then execute different schemes to reach the goal of network capacity's high usage rate. Handel and Seetharaman [33] propose a load balancing system Plug-n-Serve based on Openflow. The system adds and removes the link dynamically by tracking the state to achieve a lower response time by the reassignment. Medhi and Saikia [34] design a model FOIL (Fault-tolerant Open-Flow multi-controller model with ICMP-based lightweight inter-controller) to enhance the scalability and fault tolerance.

Generally, the routing problems could be divided into two, which are algorithms and mechanisms. The basic algorism is like Round-Robin [47]. Irengbam T et al. apply it for server load balancing and implement it with a POX controller and an OpenFlow switch. Suh and Koo [48] chose the replica that minimizes the control traffic. Sumiati and Trisnawan [49] select the edge servers based on the hash function, which also guarantee the load balancing in the system. Liu and Shi [50] design an algorithm to achieve a near-optimal latency to redirect the requests closet to the clients. It took latency as the index to provide the service for the clients. Ardaiz and Freitag [51] extract the service time from the log and chose it as the index to redirect the edge servers. Kumari and Kang [52] introduce the concept of hybrid algorithms and the indexes to choose the edge servers which combine RTT, dropping packets, load balancing, and so on. The mechanisms are often used in CDN, which contain Global Server load balancing, DNS-based request routing [53], HTTP direction [54], URL writing [55], Anycasting [56], and CDN peering [33] [57], and so on. Besides the two parts for routing in CDN, Benchaita and Ghamri-Doudane [70] modify the routing table to add user location and content type items. After the information is received, which include average round trip time, servers' load, cache hit, and so on, the update process would be performed that this information calculated

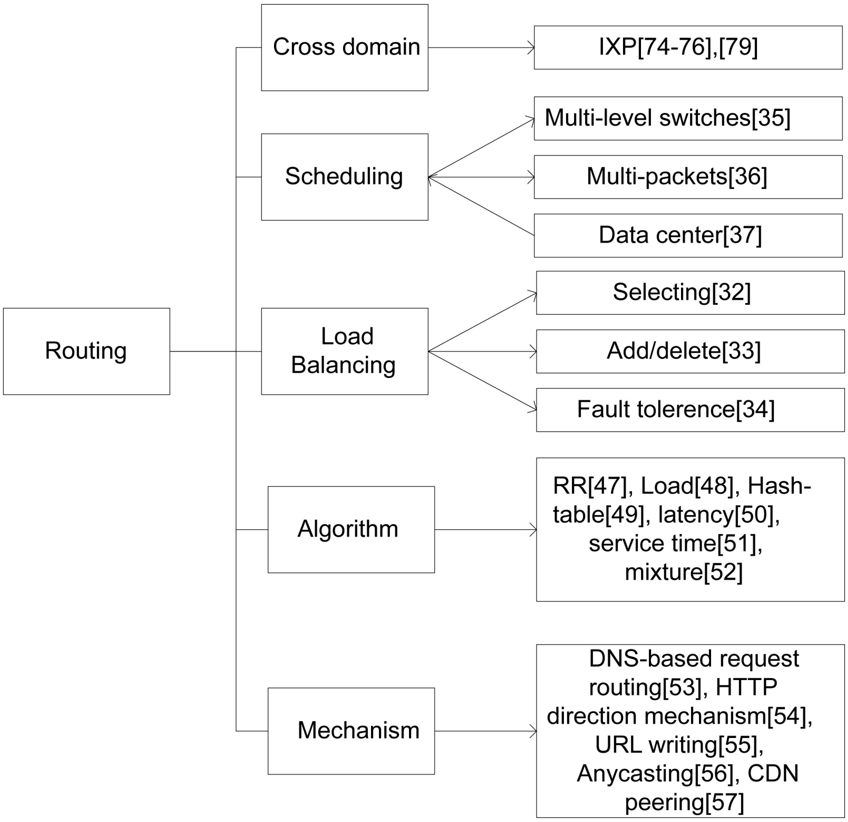


FIGURE 8. **Summary of the routing domain.**

the results. In the last, the switches forward as the new routing table. Khare and Zhang [71] trade off the ISP cost and application performance and took the network delay, servers‘ load, and system throughput as the indexes of request routing. Compared to the nearest-available routing, the two online and offline algorithms prove them available and respond to emergencies effectively.

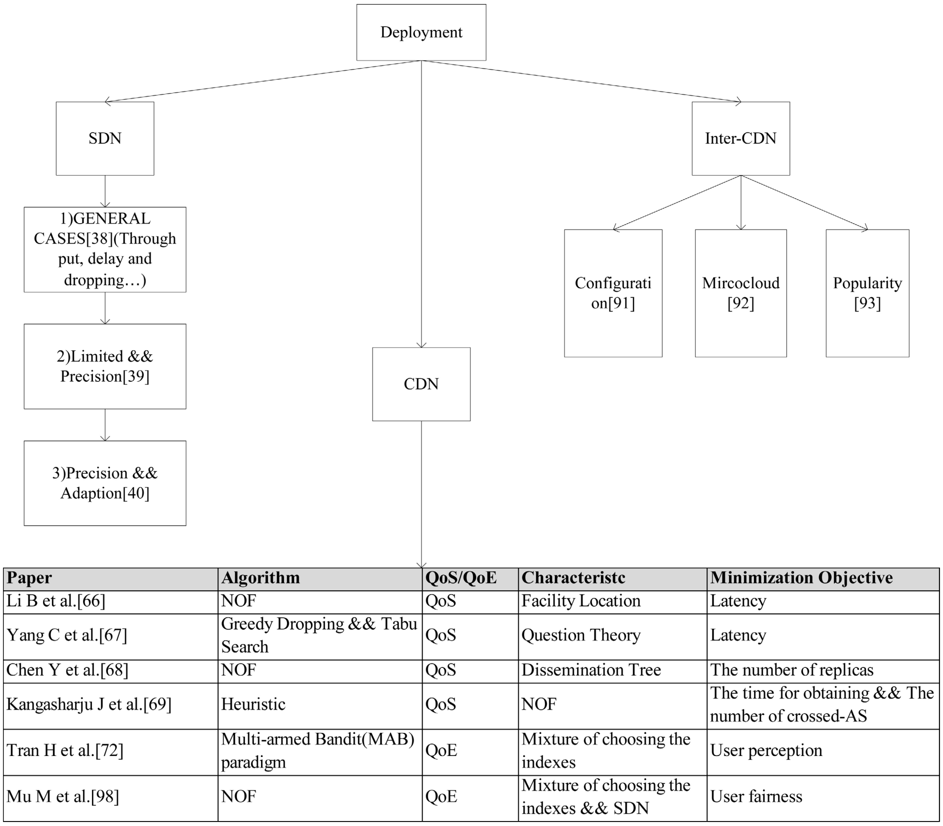
Next, we consider the cooperation in the CDNs. Of course, the algorithms and mechanisms of the single CDN could be referred to. But the challenges in the Inter-CDN still contain the request-routing problem, the placement problem, and the accounting problem. Fabra and Cánovas [84] propose CDN brokering to solve the request routing dynamically between CDNs. The CDN brokering system redesign the Intelligent Domain Name Server module, which redirect the requests to the edge server by the load-sensitive regulation. Still, the shortage of it is not applied to CDNI. Pathan and Broberg [85] conclude that choosing the own edge servers was in the first place when the own edge servers could satisfy the clients‘ demand. Rodrigues and Fernandes [86] talk about two request routing mechanisms which were interactive and recursive. Amini and Shaikh [87] propose a peering CDN system for multi-providers and a peer selection algorithm based on cost-optimized. Compared to some heuristic algorithms, it is proved better considering latency and limited capacity.

Pathan and Buyya [88] bring in the Hotspot detections module based on the architecture CSDN to deal with the overload problem through the detection module. Then, it designs autonomic resource negotiation to ease the amount of requests. Due to its single angle, the view is incomplete. From the CDN service providers, the network data is often measured indirectly, such as response time, and other specific information of the network layer is difficult to obtain in time. Thus, the information is inaccurate for the decision. And because of this, the decisions still have more space to improve. The SDN providers’ relationship between network information and applications is not so close, so the routing strategies could not match the different circumstances perfectly. Even the state of routing links is well enough, but for the specific live streaming, the circumstance is that the rouging path needs to adjust. The tight coupling framework solves this problem effectively.

As the routine and deployment issue are always closely connected, we will discuss the detail of future direction in the next subsection.

C. DEPLOYMENT

This part decides the deployment of servers, the location of caches, and the content in each server, which is shown in Fig.9. The global monitoring and measurement of SDN are essential for the timely assignment together with CDN. Van Adrichem and Doerr [38] propose the OpenNetMon for measuring throughput, delay, and dropping of each flow to justify whether it satisfies the standard quality. Then, it decreases or



Paper	Algorithm	QoS/QoE	Characteristc	Minimization Objective
Li B et al.[66]	NOF	QoS	Facility Location	Latency
Yang C et al.[67]	Greedy Dropping && Tabu Search	QoS	Question Theory	Latency
Chen Y et al.[68]	NOF	QoS	Dissemination Tree	The number of replicas
Kangasharju J et al.[69]	Heuristic	QoS	NOF	The time for obtaining && The number of crossed-AS
Tran H et al.[72]	Multi-armed Bandit(MAB) paradigm	QoE	Mixture of choosing the indexes	User perception
Mu M et al.[98]	NOF	QoE	Mixture of choosing the indexes && SDN	User fairness

FIGURE 9. Summary of the deployment domain.

adds the forwarding rate through the votes to reduce the load of the network and devices. Tootoonchian et al. [39] design a traffic matrix estimation system called OpenTM that solves the traffic matrix estimation problem by using the information of switches. The algorithm in the system is aimed at which devices should be asked to require information so that the cost was lower and the precision was higher. Chowdhury and Bari [40] propose an SDN/NFV based framework to provide distributed on-demand deployment of network functions, network slicing with service assurance, flexible network function arrangement, and optimized workload distribution.

We summarize the deployment of servers and the location of caches as the placement problem. Due to its long history, we take the classic examples to give the research directions. Li and Golin [66] chose the placement for agent servers of the web from the potential locations, and the traffic assume to be one available. Due to the more fixed location of servers, the trend of placement turns to do the research for the placement of replicas. Yang and Huang [67] model the CDN process through the queuing system and independent Poisson distribution, then they design two heuristic algorithms, Greedy Dropping and Tabu Search, to decrease the delay of requesting. Chen and Katz [68] utilize the distribution of the dissemination tree to place the replica. Under the limited capacity of servers, it guarantees the minimal number of replicas with QoS. Kangasharju and Roberts [69] propose a heuristic algorithm to decrease the time for obtaining the content, which reduces the number of crossed-AS. Tran and Hoceini [72] consider three angles in CDN together: the network conditions, the states of servers, and the score by users in choosing the edge servers. Mu and Broadbent [98] propose a user-fairness model based on SDN and CDN, then evaluate three indexes, including video quality, switching impact, and cost efficiency from three aspects which were HAS stream metadata, current client link capacity, and network resource to provision. The three indexes were taken comprehensively to conclude a result to assign the resource to achieve the user-fairness.

When the cases involve the Inter-CDN, the indexes choosing and the incentives are critical. Natalino and de Sousa [91] focus on the efficiency of configuration based on K-best CPP with minimal distance (K-CPP-minD) and then detected critical links to judge the enforceability of the solution. Takamasa and Pannu [92] try to solve the data loss issue by a micro cloud-based way that uploaded the content to different member devices. Yan and Chen [93] built a model of Inter-CDN based on the deep reinforcement learning and propose a replication algorithm based on popularity distribution.

Recently, as the video traffic has been increased significantly, information-centric networking(ICN) has emerged and some new deployment approaches have been proposed: X.Han and Chen [128] propose a new architecture to deploy the server with minimum cost, where caching and routing are separated to enhance the feasibility of the ICN. Kim and Kim [129] propose a solution that uses SVD and QR-factorization that turn the traffic matrix of the network to lower-rank one to reduce the computation time, traffic overhead,and energy consumption. Badshah et al. [130] also propose a method that deploys multiple cache servers based on joint optimization of multiple parameters to solve the traffic issues in ICN.

If we take a global view of the network and application layers, we could find the optimal solution for both parts. The general cases are to deploy the equipment or replica at minimal cost and less quantity. Whether we could make the switches and servers achieve the optimal deployment that is a good question. For various scenes in the multimedia field, the deployment of the switches could support the delivery and relieve the part of work for the replica servers. In the meantime, the deployment of the replica servers also needs to consider the circumstance of routing. Thus, the cooperation results benefit both parts and give a more suitable plan for the deployment.

As for future direction, deployment issue could always be important because a good deployment strategy can reduce the repeated traffic in the link and shorten the transmission path, it can also relieve the traffic pressure of the Internet, reduce the load of the source server, and improve the service quality and reliability of the content provider. Here we want to propose two directions that are involved with two scenarios recently emerging: 1. Hybrid SDN [136]. Hybrid sdn refers to a system with both traditional and sdn devices. It is critical to improve the interaction efficiency between the two types of devices. 2. Cloud computing. Cloud computing can realize unified network scheduling and sharing of computing resources network, storage and computing. What kind of deployment strategy should be adopted is still under research.

D. INTEROPERABILITY BETWEEN THE SDN AND CDN

Currently, the main work is to make the strategies simple to perform more directly, including the process from the Intra-CDN layer to the SDN controller layer and the process from the SDN controller layer to the data layer. This part of the work has mostly been done in SDN. Bosshart and Daly [42] improve the Openflow to add the new headers flexibly. They achieve three points: the reassignment for forwarding packets, independent of protocols as forwarding, and the functions independent of hardware. Wang and An [100] present a high-level language for programming distributed collections of network switches and evaluate its performance in several industrial scenarios. Arashloo and Koral [43] propose a centralized programming model SNAP, which modifies a big switch. The role of the big switch was responsible for forwarding packets to multi-switches, so the modification on the big switch was convenient to broadcast quickly. The SNAP solves the problems about how to distribute and assign the packets by modeling the process of input and output to optimize the results and states.

To further solve the issue of interoperability,there are two directions that can be further studied: 1. The programming way. As the API of SDN is low-level, it is hard for people to check or set the configuration, so a high-level programming language could solve this problem to a large extent. It can help people easily check the configuration and the situation of the system even with the way of visualization,and write the protocol could also be easily. What's more,some common methods can be written into library. 2. The middleware way could also help this issue from the bottom of the system. Faced with problems such as discrete data, difficult operation and low system matching. It can provide convenient service interaction and computing capabilities for the development of upper application software, and shorten the development cycle, and also save the system resources of the application itself and reduce the running cost.

E. THE APPLICATIONS OF MACHINE LEARNING ALGORITHMS IN SDN-BASED CDN

With the rise and maturity of machine learning technology, any fields try to integrate machine learning, including SD-CDN. One direction is to analysis the video quality, popularity, and clients' behavior of the content. The basis depends on the realistic platform, just like Netflix, Hulu, and so on, to make the data more convincing. Thus, we could use the available results to do other research as the data-driven or the trace-driven model.

Zhang and Yi [45] inquire about the clients‘ behavior, interactive the process for getting the content, video popularity, session length, hot pots, and so on. This gave the inspiration for making strategies and algorithms. Adhikari and Guo [46] analyze the architecture of CDN in Netflix. Netflix want to focus on the delivery process, so it migrates the Data Center to the cloud. Comparing the delivery process, including bandwidth, link rate, and the selection for edge servers considering geometry, capacity, and type of content in the three CDN service providers, which were Akamai, Limelight, and level-3, it concludes the most critical indexes to the quality. Li and Wu [77] measure the download rate from ISP, temporal effects, and geometry in the PPTV VOD system to research the delivery process and give a strategy to improve the download effectively. Zhang and Liu [78] research the hybrid loose coupling architecture of VOD in Xunlei, then introduce its deployment, cache strategies, and special dual-server mechanism to increase the play rate at first.

Due to the focus on machine learning, especially deep learning, more and more video analysis has emerged. The research topics are not limited to prediction, and it focuses more on finding the problems and solving them in the automation way [101], [111], [112], [113].

Federated learning is another direction of the integration between machine learning and SD-CDN. Federated learning is a new distributed machine learning method, which has attracted much attention recently because of its high matching characteristics with SDN distributed systems. Its workflow is to distribute the training task from the central server to each client. After receiving the task, the client only returns the trained model instead of the training data. Finally, the central server generates models according to the model uploaded by each client [117], [118], [119].

In general, federated learning will benefit SDCDN structure in two ways: 1. It can improve the safety of the system and help protect the privacy of clients. Zhang et al. [120] state that since only the parameters of the model are sent to the center server while the data of certain client is only stored in itself, the safety of data can be guaranteed. Li and Wu [127] also propose a way to protect the system from adaptive poisoning attacks based on federated learning. 2. It can also help reduce the pressure of the central server. Traditionally, central server usually takes most tasks and clients are only responsible for receiving contents, with federated learning [121], [122] that clients would also store and analyze data, the memory of central server can be saved and the pressure of it wolud also be lower.

While it still has some problems and challenges. Federated learning has certain requirements on the equipment of the client because it has to undertake certain storage and computing tasks. In addition, for large international content providers, they need to update the communication protocols of the devices and servers to achieve the task of Federated learning. This will not only generate huge costs, but also create potential security threats. Ma nd Liao [123] propose that the main challenges are: 1. Incentive mechanisms. 2. Security and privacy. 3. Global model aggregation.

According to the characteristics of machine learning. Three future directions are practical: 1. Data analysis and model training,with the large amount data distributed to different clients and the ability to record this information through SDN, these data can be fully used. For example, it can be used to predict what kind of movies or tv series could be more popular. And through this process,network virtualization in SDN-based CDN can also be achieved.It can also used to train models, with the video stream or pictures, the model of pattern recognition can be fully trained in this process. 2. Self-control system. With the rise of multi-layer neural networks, computers can surpass humans in tasks in many fields like alphaGo and OpenAI. So a self-control system that can intelligently doing the task of content distribute is possible, although it may consume a lot of time and money.

F. STANDARDIZATION

The research on SDN characteristics has become the focus of the research on building the next generation network. ONF has issued a series of standardization recommendations and related protocols and technical standards for the next generation network system [131]. In its work, the architecture is divided into three planes: data transmission plane, control plane and application plane. The southbound interface protocol is used between the controller and the switch [132], the communication between the controller and the application plane allows users to customize development according to actual needs, which is also one of the conditions conducive to building a software defined content distribution network. This new infrastructure has been widely concerned and practiced in the industry and academia. The GENI project in the United States, the FIRE project in Europe, the JGN2plus project in Japan, and the SOFIA project in China have successively studied the next generation network. Currently, the typical work achievements include openFlow, 4D [133], Ethane [134], and ForCES.

Another important architecture is I2RS developed by IETF. The core idea of the Routing System Interface (I2RS) Working Group is to open a new interface based on the current routing and forwarding system of traditional network equipment to communicate with the external control layer. The external control layer dynamically sends routing status and policies to each device through the events, topology changes, traffic statistics and other information fed back by the device. The existing management protocols NETCONF and YANG of IETF are selected to realize the opening of the routing system. At present, the I2RS working group has completed most of the data models of the routing information base (RIB) and topology, and is discussing the protocol improvement of NETCONF.

When it comes to the standardization of next generation network,two directions are practical: 1.5G standardization. Compared with 4G network, the innovative development of 5G network architecture, the three application directions and application scenarios of enhanced mobile bandwidth (eMBB), large-scale Internet of Things (mMTC), and ultra-high reliability ultra-low delay communication (uRLLC) of 5G network pose new challenges to the technical capabilities, evolution and carrying demands of optical transmission network. In this context, research and analysis of standardization in 5G mobile communication network architecture will play a positive role in further improving data flow, communication performance, etc. 2. Another similar direction is the next-generation wireless networks(NWN), with the development of IoT, the standardization of it is also important.

G. CROSS-CUTTING APPROACHES

First, we take some research as examples for other problems in SDCDN. Many researchers have paid the topic of choosing the ISP a lot of attention. Gao and Wang [94] solve how to select the ISP by Content Providers based on the peering agreement and design a heuristic algorithm to solve the policy-aware peering, which was an NP-complete problem. Maillé and Simon [95] mainly chose the index of revenue to select the ISP and the cache strategies. Meanwhile, it adds the content popularities to guarantee the QoS to get an optimization through the Lagrangian theory.

Next, the coordination problem between the network layers has been unsolved. Elkotob et al. [96] propose an architecture to coordinate the DWDM layer and network layer based on the video multicast optimized tree to achieve higher effectiveness and lower delay. Or differentiating the CDN used a tool like Akhtar and Hussain [97] develop, it made a difference from the index of user-perceived, then it compares time, location, and performance. So the tool could recognize the different CDN.

The following researches cover many angles which have been mentioned in former sections. Ren and Chan [61] combine delay for the replication, selection, and storage to justify whether the replica needs to be reassigned. Gamehi and Analoui [62] propose a truthful profit-maximizing auction algorithm to choose the edge server. The algorithm decide the range of sharing among peers, and the cost of selecting an edge server is also considered to increase the profit of P2P and decrease the cost of CDN. Torres-Cruz and Rivero-Angeles [73] formulate a comprehensive network to get an optimized result by considering the feature of service, attributes of the network, and behavior of the users.

Finally, other problems related to SDCDN may occur, such as security [65], [114], wireless and updating cache. These also need to combine with current development for SDN and CDN, respectively.

Above all the cross-cutting approaches, the security issue is most concerned as its security architecture design has become an important basis for judging whether the system can be delivered for use. Here we propose two practical directions: 1. Data leakage and malicious data modification. In the OpenFlow switch, it is specified that each input packet must match the record in the flow table item. If the matching fails, it will be discarded or packaged as a packet_in message and sent to the controller. Thus, the attacker can judge the approximate time required for each action by analyzing the time of a packet. In addition, attackers can also determine the configuration of more switches by carefully forging packets. With this information, attackers can forge the same type of packets to launch flooding attacks. Another data leakage is reflected in some sensitive information (such as passwords and permissions) in the system. Once such information is disclosed, the security of SDN will be threatened. How to reduce or avoid the security risks caused by data leakage and malicious data modification in SDN-based CDN is an urgent problem to be solved in the future. 2. Identity recognition. Identity recognition has always been an important research topic in the field of network security. Juniper and other companies point out that the current traditional network security problems mainly come from the weak network identification [135]. Similarly, SDCDN has the same problem.

IV. POTENTIAL RESEARCH DIRECTIONS

According to the previous introduction, we conclude 10 potential research directions where researchers can conduct researches to improve the performance of SDCDN.

A. OPTIMIZING SDN-BASED CDN ROUTING ALGORITHMS

CDN routing algorithms play a crucial role in determining the most optimal path for content delivery to end-users. With the introduction of SDN-based CDN, there is a need to optimize the existing routing algorithms to accommodate the dynamic network topology and SDN controller's ability to improve the network's behavior. The SDN controller can use real-time network information to make informed routing decisions, making SDN-based CDN more efficient than traditional CDNs.

To optimize SDN-based CDN routing algorithms, researchers can conduct several research directions. For example, researchers can explore the use of machine learning algorithms to predict network traffic patterns and predict future traffic demands. By predicting traffic demands, the SDN controller can adjust the routing algorithms to direct traffic to the most efficient paths, reducing network congestion and improving end-users' QoE.

Another research direction is the use of advanced traffic engineering techniques such as traffic splitting, traffic aggregation, and traffic rerouting to optimize SDN-based CDN routing algorithms. For instance, traffic splitting enables the SDN controller to split traffic between multiple paths, while traffic aggregation enables the SDN controller to aggregate multiple traffic flows into a single path, reducing network congestion.

Researchers can also explore the use of dynamic routing algorithms that can adjust the routing paths based on the real-time network conditions. For instance, researchers can explore the use of algorithms that can detect network congestion and reroute traffic to less congested paths, improving network performance and end-users' QoE.

B. INTEGRATING SDN AND AI IN CDN MANAGEMENT

CDN management is a complex task that requires monitoring, control, and optimization of various resources such as cache servers, network links, and load balancing. The integration of SDN and AI has the potential to improve CDN management by providing more intelligent, automated, and efficient solutions. In this direction, research can be conducted to develop algorithms that integrate SDN and AI to manage different aspects of CDN, such as cache placement, load balancing, and network optimization.

One research direction in this area could be to develop AI-based load balancing algorithms that are integrated with the SDN controller. These algorithms can learn from past traffic patterns and predict future traffic demands to optimize the distribution of traffic across the network. Another research direction could be to use AI to optimize the placement of cache servers in the network. AI algorithms can analyze the traffic patterns and user behavior to determine the optimal placement of cache servers to reduce latency and improve user experience.

In addition, AI can also be used to monitor network traffic and detect anomalies and security threats. An AI-based security system can analyze traffic patterns to identify potential security threats and take corrective actions to mitigate them. AI can also be used to optimize the CDN routing policies based on real-time network conditions. This can include dynamically routing traffic around congested network links or optimizing the use of network resources based on the application's QoS requirements.

Another research direction could be to integrate SDN and AI to optimize the use of different CDN technologies, such as video streaming and content pre-fetching. AI can be used to predict the user's viewing habits and dynamically adjust the video quality to optimize the viewing experience. Similarly, AI can be used to pre-fetch content based on user behavior, network conditions, and device capabilities, to reduce latency and improve the user experience.

Overall, the integration of SDN and AI in CDN management has the potential to improve the performance, reliability, and security of CDN services. By developing intelligent algorithms that are integrated with the SDN controller, CDN providers can optimize the use of network resources, reduce latency, and improve user experience.

C. IMPLEMENTING SDN-BASED CDNs FOR IoT APPLICATIONS

The rapid growth of IoT has led to an increase in the volume and diversity of data generated by IoT devices. This has created a need for efficient and scalable CDNs to handle the increased traffic generated by these devices. SDN-based CDNs have the potential to address these challenges by providing programmable network infrastructure that can be dynamically reconfigured to meet changing traffic demands.

One area of research in this direction is to investigate the integration of SDN and CDNs in IoT applications. This involves developing SDN-based CDN architectures that are optimized for IoT traffic patterns, such as the large number of devices generating small amounts of data. SDN can be used to dynamically route traffic based on the location and type of IoT devices, while CDNs can be used to cache and deliver content closer to end-users. AI-based techniques can also be used to predict traffic patterns and optimize network resources for better performance.

Another research direction is to investigate the security implications of SDN-based CDNs for IoT applications. IoT devices are often vulnerable to attacks due to their limited processing power and security features. SDN-based CDNs can provide an additional layer of security by allowing traffic to be dynamically rerouted in response to security threats. However, this also introduces new security risks, such as the possibility of an attacker exploiting vulnerabilities in the SDN controller to gain control of the network. Research in this area can focus on developing secure SDN-based CDN architectures for IoT applications.

In all, the integration of SDN and CDNs in IoT applications has the potential to provide significant benefits, including improved network scalability, reliability, and security. However, further research is needed to develop efficient and secure SDN-based CDN architectures optimized for IoT traffic patterns.

D. SECURING SDN-BASED CDNs

Securing SDN-based CDNs is a critical area of research, given the sensitive nature of the data being transmitted over CDNs. Researchers can explore several directions in this area, including:

1. Designing secure SDN-based CDN architectures: Researchers can explore the design of secure CDN architectures using SDN. This would involve identifying potential security threats and designing architectures that can mitigate these threats.

2. Developing secure SDN-based CDN protocols: Researchers can develop secure protocols for SDN-based CDNs to ensure that data is securely transmitted over the network. This would involve developing secure routing and communication protocols that can resist various types of attacks.

3. Identifying and mitigating SDN-based CDN vulnerabilities: Researchers can identify potential vulnerabilities in SDN-based CDNs and develop mitigation strategies to address these vulnerabilities. This would involve developing techniques for detecting and preventing attacks such as DDoS, malware, and data theft.

4. Incorporating blockchain technology in SDN-based CDNs: Researchers can explore the integration of blockchain technology in SDN-based CDNs to enhance security. This would involve exploring the potential benefits of blockchain technology, such as decentralization and immutability, in securing CDNs.

5. Developing intrusion detection and prevention systems for SDN-based CDNs: Researchers can develop intrusion detection and prevention systems specifically for SDN-based CDNs. This would involve developing techniques for detecting and preventing various types of attacks, such as DDoS and malware attacks.

6. Applying machine learning and AI techniques for securing SDN-based CDNs: Researchers can explore the application of machine learning and AI techniques for securing SDN-based CDNs. This would involve developing algorithms that can detect and prevent attacks in real-time.

7. Studying the impact of SDN on CDN security: Researchers can study the impact of SDN on the security of CDNs. This would involve analyzing the various security threats that can arise as a result of SDN and developing strategies to mitigate these threats.

Securing SDN-based CDNs is a crucial area of research that requires the development of innovative and effective techniques for ensuring the security and privacy of data transmitted over CDNs.

E. ENABLING SDN-BASED CDNs FOR DYNAMIC CONTENT

CDNs have traditionally been designed to deliver static content, such as images, videos, and web pages, efficiently and reliably. However, as more dynamic and interactive content is being generated and consumed on the internet, SDN-based CDNs must adapt to support the delivery of such content.

One method is to explore ways to enable SDN-based CDNs for dynamic content. This includes designing efficient caching algorithms that can effectively cache and deliver dynamic content. One approach is to use techniques such as delta encoding and differential compression to reduce the amount of data that needs to be transferred. Another approach is to use more sophisticated caching strategies that can cache different parts of a dynamic page separately.

Another method is to explore ways to optimize SDN-based CDNs for real-time streaming applications. This involves designing intelligent load balancing and routing algorithms that can ensure high quality of service for real-time streams. This can be achieved by dynamically selecting the best edge servers to serve the stream, based on factors such as available bandwidth, network latency, and server load.

Finally, researchers can also explore ways to enable SDN-based CDNs for personalized content delivery. This involves designing algorithms that can intelligently personalize the content delivered to each user, based on their preferences, past behavior, and other factors. This can be achieved by using machine learning techniques to analyze user data and generate personalized recommendations for content delivery.

F. INTEGRATING SDN-BASED CDNs WITH EDGE COMPUTING

SDN-based CDNs can be integrated with edge computing to enable content delivery at the edge, closer to the end-users. Edge computing refers to the processing of data and computation at the edge of the network, closer to where the data is generated, rather than transmitting the data back to a central data center. By integrating SDN-based CDNs with edge computing, the content can be delivered to the users faster with lower latency, reducing the load on the core network and improving the overall user experience.

So it could be to explore the optimal placement of CDN nodes at the edge to maximize the efficiency of the CDN while minimizing the cost of deploying and maintaining the nodes. This could involve developing algorithms for determining the optimal number of nodes and their placement based on user demand, traffic patterns, and available resources.

Another research direction could be to investigate the use of SDN-based CDNs to support emerging edge computing applications, such as smart cities, autonomous vehicles, and industrial IoT. In these applications, the CDN could be used to distribute content and data to edge devices in real-time, enabling more efficient and effective data processing and analysis.

Finally, researchers could also explore the security implications of integrating SDN-based CDNs with edge computing. With edge computing, data is processed and stored closer to the end-users, making it more vulnerable to cyber attacks. Thus, there is a need for developing secure SDN-based CDN architectures that can protect data and applications at the edge, while also providing efficient content delivery.

G. OPTIMIZING SDN-BASED CDN ENERGY EFFICIENCY

CDNs consume a significant amount of energy due to their large-scale infrastructure and high demand for data processing and transfer. Therefore, optimizing the energy efficiency of SDN-based CDNs is a crucial research direction.

Obviously, one way to achieve energy efficiency is to minimize the number of active servers while maintaining high-quality content delivery. This can be done by optimizing the placement of cache servers in SDN-based CDNs. Researchers can explore new caching strategies that enable dynamic server allocation, content migration, and smart routing to minimize energy consumption while maximizing user experience. Another strategy is to use energy-efficient hardware components, such as low-power processors, network interface cards, and solid-state drives.

Researchers can also design SDN-based CDNs that support energy harvesting technologies. Energy harvesting allows the system to extract energy from the environment (e.g., solar, wind, or thermal) to power the infrastructure. Therefore, energy harvesting can reduce the reliance on the power grid and minimize the carbon footprint of the CDN. We can investigate the feasibility of integrating energy harvesting into SDN-based CDNs and evaluate the trade-off between energy efficiency and performance.

Furthermore, SDN-based CDNs can leverage machine learning algorithms to predict user behavior and network traffic patterns, allowing the system to adapt to changes in demand and reduce energy consumption. Researchers can develop predictive models that incorporate various parameters such as user location, content popularity, and network congestion to optimize energy consumption. Additionally, SDN-based CDNs can use intelligent resource allocation and load balancing techniques to minimize energy consumption while ensuring optimal content delivery.

In summary, optimizing SDN-based CDN energy efficiency requires a multi-disciplinary approach that combines network design, hardware optimization, energy harvesting, and machine learning. Researchers can explore new techniques and algorithms to improve energy efficiency while maintaining high-quality content delivery.

H. IMPROVING SDN-BASED CDN QoS

Improving quality of service (QoS) is an important research direction in SDN-based CDN. In traditional CDNs, QoS is often ensured by replicating content across multiple servers and using load balancing algorithms. However, in SDN-based CDNs, QoS can be improved by leveraging the centralized control provided by SDN.

One possible approach to improving QoS in SDN-based CDNs is to use software-defined traffic engineering. This involves dynamically allocating network resources based on the traffic patterns and the service-level agreements (SLAs) defined for different applications. By dynamically adjusting network resources, SDN-based CDNs can ensure that the most critical applications get the resources they need, while less critical applications are allocated fewer resources.

Another approach to improving QoS in SDN-based CDNs is to use machine learning algorithms to predict traffic patterns and adjust network resources accordingly. This involves collecting and analyzing data on network traffic and user behavior to identify patterns and predict future traffic demands. By using machine learning algorithms, SDN-based CDNs can optimize resource allocation to ensure that QoS requirements are met even under changing traffic conditions.

In addition, SDN-based CDNs can use adaptive streaming techniques to improve QoS for video streaming applications. Adaptive streaming involves dynamically adjusting the quality of the video stream based on the available network resources and the user's device capabilities. By using SDN to dynamically allocate network resources and machine learning algorithms to predict traffic patterns, SDN-based CDNs can optimize adaptive streaming to ensure that users receive the best possible viewing experience.

Overall, improving QoS is an important research direction in SDN-based CDN, and there are many approaches that can be used to achieve this goal. By leveraging the centralized control provided by SDN and using advanced algorithms and techniques, SDN-based CDNs can ensure that users receive high-quality services even under changing traffic conditions.

I. DESIGNING SDN-BASED CDNs FOR 5G NETWORKS

The advent of 5G networks has brought forth numerous opportunities for research in the field of SDN-based CDNs. In this section, we will discuss some research directions for designing SDN-based CDNs for 5G networks.

1. Dynamic network slicing: 5G networks are expected to support dynamic network slicing, which involves dividing the network into multiple virtual networks to provide customized services to different applications. Research can be conducted to explore how SDN-based CDNs can leverage network slicing to provide efficient content delivery to specific applications.

2. Mobile edge computing (MEC): With the emergence of 5G networks, MEC is becoming increasingly popular. SDN-based CDNs can leverage MEC to provide low-latency content delivery to mobile devices. Research can be conducted to explore how SDN-based CDNs can be integrated with MEC to provide seamless content delivery to mobile devices.

3. Network function virtualization (NFV): 5G networks support NFV, which involves virtualizing network functions such as routing, load balancing, and caching. Research can be conducted to explore how SDN-based CDNs can be integrated with NFV to provide efficient content delivery.

4. Multi-access edge computing (MEC) and SDN: MEC provides a distributed computing architecture that extends cloud computing capabilities to the edge of the network. By leveraging SDN-based CDNs, MEC can provide a seamless content delivery experience to end-users.

5. Network slicing and SDN-based CDNs: Network slicing allows for the creation of multiple virtual networks on top of a physical infrastructure, with each slice customized to meet the specific requirements of a particular application or service. SDN-based CDNs can leverage network slicing to provide optimized content delivery to different applications.

6. QoS in SDN-based CDNs: QoS is an essential aspect of CDNs. With the emergence of 5G networks, SDN-based CDNs need to be optimized to provide QoS guarantees to end-users. Research can be conducted to explore how SDN-based CDNs can be optimized to provide QoS guarantees in a 5G environment.

7. Multi-access edge computing (MEC) and SDN: MEC provides a distributed computing architecture that extends cloud computing capabilities to the edge of the network. By leveraging SDN-based CDNs, MEC can provide a seamless content delivery experience to end-users.

8. Network slicing and SDN-based CDNs: Network slicing allows for the creation of multiple virtual networks on top of a physical infrastructure, with each slice customized to meet the specific requirements of a particular application or service. SDN-based CDNs can leverage network slicing to provide optimized content delivery to different applications.

9. Energy efficiency in SDN-based CDNs: The increased demand for content delivery in 5G networks implies a higher energy consumption. Research can be conducted to explore how SDN-based CDNs can be optimized to minimize energy consumption while still providing efficient content delivery.

10. Load balancing and SDN-based CDNs: Load balancing is a critical aspect of content delivery networks. Research can be conducted to explore how SDN-based CDNs can leverage load balancing to provide efficient content delivery in a 5G environment.

J. EVALUATING SDN-BASED CDN PERFORMANCE

SDN-based CDNs have shown great potential in improving the performance of content delivery networks. However, there is still a need to evaluate their performance in different scenarios and configurations. This research direction involves conducting comprehensive performance evaluations of SDN-based CDNs to identify their strengths and limitations.

One aspect of performance evaluation is to measure the throughput, latency, and packet loss of SDN-based CDNs under various loads and network conditions. This involves simulating different network topologies and traffic patterns to analyze how SDN-based CDNs perform in different scenarios. Another aspect of performance evaluation is to assess the scalability of SDN-based CDNs by testing their ability to handle increasing amounts of traffic and the number of connected clients.

Furthermore, performance evaluation can also involve measuring the energy efficiency and resource utilization of SDN-based CDNs. This can help identify ways to optimize the energy consumption of SDN-based CDNs, which is critical for their sustainability and cost-effectiveness.

To evaluate the performance of SDN-based CDNs, researchers can use various tools and methodologies such as simulation tools, testbeds, and real-world experiments. The results of these evaluations can provide insights into how to improve the design and implementation of SDN-based CDNs for better performance and scalability.

V. CONCLUSION

In this paper, we summarize the characteristics and problems of SDCDN architecture which could be widely used in social networks, live streaming, VOD and so on. To face the amount of traffic, the architecture combines the advantages of SDN with CDN to decrease the delay and guarantee the quality. After introducing the trajectory and development of similar related researches, we conclude the eight related domains for continuing the exploration. In the meantime, these are also the dominating challenges in the SDCDN with the development of SDN and CDN. Different from other surveys, we more focus on the breath of the aspects in the SDCDN that there exits many angles even for the one domain. These will not only inspire for the researchers, but also will be our next steps that holds for future work.

REFERENCES

[1] G. Pallis and A. Vakali, "Insight and perspectives for content delivery networks," *Commun. ACM*, vol. 49, no. 1, pp. 101–106, 2006.

[2] A. Vakali and G. Pallis, "Content delivery networks: Status and trends," *IEEE Internet Comput.*, vol. 7, no. 6, pp. 68–74, Nov./Dec. 2003.

[3] G. Peng, "CDN: Content distribution network," 2004, *arXiv:cs/0411069*.

[4] D. Wang, S. Zhang, Y. Xue, and Y. Dong, "Identifying influential factors of CDN performance with large-scale data analysis," in *Proc. Int. Conf. Comput., Netw. Commun. (ICNC)*, Mar. 2018, pp. 873–877.

[5] N. McKeown, "How SDN will shape networking," in *Proc. Open Netw. Summit*, 2011.

[6] S. Shenker, M. Casado, T. Koponen, and N. McKeown, "The future of networking, and the past of protocols," *Open Netw. Summit*, vol. 20, pp. 1–30, Oct. 2011.

[7] Open Networking Foundation. (Apr. 13, 2012). *ONF White Paper-Software-Defined Networking: The New Norm for Networks*. [Online]. Available: https:/www.opennetworking.org/

[8] L. Yang, R. Dantu, T. Anderson, and R. Gopal, "*Forwarding and Control Element Separation (ForCES) Framework*, document RFC 3746, 2004.

[9] A. Farrel, J.-P. Vasseur, and J. Ash, *A Path Computation Element (PCE)-Based Architecture*, document RFC 4655, 2006.

[10] Cisco. (2014). *Cisco's One Platform Kit (OnePK)*. [Online]. Available: http://www.cisco.com/en/U.S./prod/iosswrel/onepk.html

[11] A. Yazdinejad, R. M. Parizi, A. Dehghantanha, Q. Zhang, and K.-K.-R. Choo, "An energy-efficient SDN controller architecture for IoT networks with blockchain-based security," *IEEE Trans. Serv. Comput.*, vol. 13, no. 4, pp. 625–638, Jul. 2020.

[12] M. T. Islam, N. Islam, and M. A. Refat, "Node to node performance evaluation through RYU SDN controller," *Wireless Pers. Commun.*, vol. 112, no. 1, pp. 555–570, May 2020.

[13] A. K. Singh, S. Maurya, N. Kumar, and S. Srivastava, "Heuristic approaches for the reliable SDN controller placement problem," *Trans. Emerg. Telecommun. Technol.*, vol. 31, no. 2, Feb. 2020, Art. no. e3761.

[14] H. E. Egilmez, S. T. Dane, B. Gorkemli, and A. M. Tekalp, "OpenQoS: OpenFlow controller design and test network for multimedia delivery with quality of service," in *Proc. NEM Summit, Implementing Future Media Internet Towards New Horizons*, 2012, pp. 22–27.

[15] J. E. Beasley and N. Christofides, "An algorithm for the resource constrained shortest path problem," *Networks*, vol. 19, no. 4, pp. 379–394, 1989.

[16] V. K. Gurbani, M. Scharf, T. V. Lakshman, V. Hilt, and E. Marocco, "Abstracting network state in software defined networks (SDN) for rendezvous services," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2012, pp. 6627–6632.

[17] H. Nam, K.-H. Kim, J. Y. Kim, and H. Schulzrinne, "Towards QoE-aware video streaming using SDN," in *Proc. IEEE Global Commun. Conf.*, Dec. 2014, pp. 1317–1322.

[18] P. A. L. Rego, M. S. Bonfim, M. D. Ortiz, J. M. Bezerra, D. R. Campelo, and J. N. de Souza, "An OpenFlow-based elastic solution for cloud-CDN video streaming service," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2014, pp. 1–7.

[19] M. Wichtlhuber, R. Reinecke, and D. Hausheer, "An SDN-based CDN/ISP collaboration architecture for managing high-volume flows," *IEEE Trans. Netw. Serv. Manage.*, vol. 12, no. 1, pp. 48–60, Mar. 2015.

[20] L. Chen, M. Qiu, W. Dai, and N. Jiang, "Supporting high-quality video streaming with SDN-based CDNs," *J. Supercomput.*, vol. 73, no. 8, pp. 3547–3561, Aug. 2017.

[21] S. Li, I. Doh, and K. Chae, "Key management mechanism in ALTO/SDN based CDNi architecture," in *Proc. Int. Conf. Inf. Netw. (ICOIN)*, Jan. 2015, pp. 110–115.

[22] H. Owens, II, and A. Durresi, "Video over software-defined networking (VSDN)," *Comput. Netw.*, vol. 92, pp. 341–356, Dec. 2015.

[23] H. Adiseshu, G. Parulkar, and R. Yavatkar, "A state management protocol for IntServ, DiffServ and label switching," in *Proc. 6th Int. Conf. Netw. Protocol*, 1998, pp. 272–281.

[24] Z. Mammeri, "Framework for parameter mapping to provide end-to-end QoS guarantees in IntServ/DiffServ architectures," *Comput. Commun.*, vol. 28, no. 9, pp. 1074–1092, Jun. 2005.

[25] Á. L. V. Caraguay, J. A. P. Fernández, and L. J. G. Villalba, "Framework for optimized multimedia routing over software defined networks," *Comput. Netw.*, vol. 92, pp. 369–379, Dec. 2015.

[26] T. Uzakgider, C. Cetinkaya, and M. Sayit, "Learning-based approach for layered adaptive video streaming over SDN," *Comput. Netw.*, vol. 92, pp. 357–368, Dec. 2015.

[27] C. H. Papadimitriou and J. N. Tsitsiklis, "The complexity of Markov decision processes," *Math. Oper. Res.*, vol. 12, no. 3, pp. 441–450, Aug. 1987.

[28] Q. Wang, K. Xu, R. Izard, B. Kribbs, J. Porter, K.-C. Wang, A. Prakash, and P. Ramanathan, "GENI cinema: An SDN-assisted scalable live video streaming service," in *Proc. IEEE 22nd Int. Conf. Netw. Protocols*, Oct. 2014, pp. 529–532.

[29] J. W. Guck, M. Reisslein, and W. Kellerer, "Function split between delay-constrained routing and resource allocation for centrally managed QoS in industrial networks," *IEEE Trans. Ind. Informat.*, vol. 12, no. 6, pp. 2050–2061, Dec. 2016.

[30] C. Hue, Y.-J. Chen, and L.-C. Wang, "Traffic-aware networking for video streaming service using SDN," in *Proc. IEEE 34th Int. Perform. Comput. Commun. Conf. (IPCCC)*, Dec. 2015, pp. 1–5.

[31] J. W. Kleinrouweler, S. Cabrero, and P. Cesar, "Delivering stable high-quality video: An SDN architecture with DASH assisting network elements," in *Proc. 7th Int. Conf. Multimedia Syst.*, May 2016, pp. 1–10.

[32] X. Shi, Y. Li, H. Xie, T. Yang, L. Zhang, P. Liu, H. Zhang, and Z. Liang, "An OpenFlow-based load balancing strategy in SDN," *Comput. Mater. Continua*, vol. 62, no. 1, pp. 385–398, 2020.

[33] N. Handigol, S. Seetharaman, M. Flajslik, N. McKeown, and R. Johari, "Plug-n-Serve: Load-balancing web traffic using OpenFlow," *ACM SIGCOMM Demo*, vol. 4, no. 5, p. 6, 2020.

[34] N. Medhi and D. K. Saikia, "OpenFlow-based multi-controller model for fault-tolerant and reliable control plane," in *Smart Computing Paradigms: New Progresses and Challenges*. Singapore: Springer, 2020, pp. 61–73.

[35] M. Al-Fares, S. Radhakrishnan, B. Raghavan, N. Huang, and A. Vahdat, "Hedera: Dynamic flow scheduling for data center networks," in *Proc. NSDI*, 2010, pp. 89–92.

[36] A. Ishimori, F. Farias, E. Cerqueira, and A. Abelem, "Control of multiple packet schedulers for improving QoS on OpenFlow/SDN networking," in *Proc. 2nd Eur. Workshop Softw. Defined Netw.*, Oct. 2013, pp. 81–86.

[37] S. Nacakli and A. M. Tekalp, "Controlling P2P-CDN live streaming services at SDN-enabled multi-access edge datacenters," *IEEE Trans. Multimedia*, vol. 23, pp. 3805–3816, 2021.

[38] N. L. M. van Adrichem, C. Doerr, and F. A. Kuipers, "OpenNetMon: Network monitoring in OpenFlow software-defined networks," in *Proc. IEEE Netw. Oper. Manage. Symp. (NOMS)*, May 2014, pp. 1–8.

[39] A. Tootoonchian, M. Ghobadi, and Y. Ganjali, "OpenTM: Traffic matrix estimator for OpenFlow networks," in *Proc. Int. Conf. Passive Act. Netw. Meas.* Berlin, Germany: Springer, 2010, pp. 201–210.

[40] S. R. Chowdhury, M. F. Bari, R. Ahmed, and R. Boutaba, "PayLess: A low cost network monitoring framework for software defined networks," in *Proc. IEEE Netw. Oper. Manage. Symp. (NOMS)*, May 2014, pp. 1–9.

[41] S. Lange, S. Gebert, T. Zinner, P. Tran-Gia, D. Hock, M. Jarschel, and M. Hoffmann, "Heuristic approaches to the controller placement problem in large scale SDN networks," *IEEE Trans. Netw. Serv. Manag.*, vol. 12, no. 1, pp. 4–17, Mar. 2015.

[42] P. Bosshart, D. Daly, G. Gibb, M. Izzard, N. McKeown, J. Rexford, C. Schlesinger, D. Talayco, A. Vahdat, G. Varghese, and D. Walker, "P4: Programming protocol-independent packet processors," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 44, pp. 87–95, Jul. 2014.

[43] M. T. Arashloo, Y. Koral, M. Greenberg, J. Rexford, and D. Walker, "SNAP: Stateful network-wide abstractions for packet processing," in *Proc. ACM SIGCOMM Conf.*, Aug. 2016, pp. 29–43.

[44] D. Wu, Y. T. Hou, W. Zhu, Y.-Q. Zhang, and J. M. Peha, "Streaming video over the internet: Approaches and directions," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 3, pp. 282–300, Mar. 2001.

[45] D. Zhang, Y. Yi, and S. Wang, "User behavior analyses in mobile personal livecasts," in *Proc. IEEE 4th Int. Conf. Comput. Commun. (ICCC)*, Dec. 2018, pp. 2185–2189.

[46] V. K. Adhikari, Y. Guo, F. Hao, V. Hilt, Z.-L. Zhang, M. Varvello, and M. Steiner, "Measurement study of Netflix, Hulu, and a tale of three CDNs," *IEEE/ACM Trans. Netw.*, vol. 23, no. 6, pp. 1984–1997, Dec. 2015.

[47] I. T. Singh, T. R. Singh, and T. Sinam, "Server load balancing with round Robin technique in SDN," in *Proc. Int. Conf. Decis. Aid Sci. Appl. (DASA)*, Mar. 2022, pp. 503–505.

[48] D. Suh, J. Koo, and S. Pack, "Optimal leader selection for minimizing control traffic in distributed SDN controllers," in *Proc. Int. Conf. Electron., Inf., Commun. (ICEIC)*, Jan. 2018, pp. 1–2.

[49] A. Sumiati, P. H. Trisnawan, and M. A. Fauzi, "Implementasi load balancing web server dengan algoritma source IP hash pada software defined network (SDN)," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 4, no. 3, pp. 919–928, 2020.

[50] J. Liu, Y. Shi, L. Zhao, Y. Cao, W. Sun, and N. Kato, "Joint placement of controllers and gateways in SDN-enabled 5G-satellite integrated network," *IEEE J. Sel. Area Commun.*, vol. 36, no. 2, pp. 221–232, Feb. 2018.

[51] O. Ardaiz, F. Freitag, and L. Navarro, "Improving the service time of web clients using server redirection," *ACM SIGMETRICS Perform. Eval. Rev.*, vol. 29, no. 2, pp. 39–44, Sep. 2001.

[52] S. Kumari and S. S. Kang, "Study of traffic based load balancing algorithm in the SDN," in *Proc. 3rd Int. Conf. Inventive Res. Comput. Appl. (ICIRCA)*, Sep. 2021, pp. 312–318.

[53] E. Harahap, Y. Fajar, D. Ahmadi, A. Kudus, and R. Ceha, "Modeling of request routing management on router for content delivery network," *Int. J. Sci. Technol. Res.*, vol. 9, no. 3, pp. 308–315, 2020.

[54] A. Shaikh, R. Tewari, and M. Agrawal, "On the effectiveness of DNS-based server selection," in *Proc. Conf. Comput. Commun. 20th Annu. Joint Conf. IEEE Comput. Commun. Soc. (IEEE INFOCOM)*, 2001, pp. 1801–1810.

[55] B. Krishnamurthy, C. Wills, and Y. Zhang, "On the use and performance of content distribution networks," in *Proc. 1st ACM SIGCOMM Workshop Internet Meas.*, 2001, pp. 169–182.

[56] Q. Fu, B. Rutter, H. Li, P. Zhang, C. Hu, T. Pan, Z. Huang, and Y. Hou, "Taming the wild: A scalable anycast-based CDN architecture (T-SAC)," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 12, pp. 2757–2774, Dec. 2018.

[57] P. K. Dey and M. Yuksel, "Peering among content-dominated vertical ISPs," *IEEE Netw. Lett.*, vol. 1, no. 3, pp. 132–135, Sep. 2019.

[58] A. Ganjam, F. Siddiqui, J. Zhan, X. Liu, I. Stoica, J. Jiang, V. Sekar, and H. Zhang, "C3: Internet-scale control plane for video quality optimization," in *Proc. 12th USENIX Symp. Netw. Syst. Design Implementation (NSDI)*, 2015, pp. 131–144.

[59] A. Alasaad, S. Gopalakrishnan, and V. C. M. Leung, "A hybrid approach for cost-effective media streaming based on prediction of demand in community networks," *Telecommun. Syst.*, vol. 59, no. 3, pp. 329–343, Jul. 2015.

[60] M. Ma, Z. Wang, K. Su, and L. Sun, "Understanding content placement strategies in smartrouter-based peer video CDN," in *Proc. 26th Int. Workshop Netw. Operating Syst. Support Digit. Audio Video*, May 2016, pp. 1–6.

[61] D. Ren, S.-H.-G. Chan, G. Shi, and H. Zhang, "Distributed joint optimization for large-scale video-on-demand," *Comput. Netw.*, vol. 75, pp. 86–98, Dec. 2014.

[62] M. Garmehi, M. Analoui, M. Pathan, and R. Buyya, "An economic mechanism for request routing and resource allocation in hybrid CDN-P2P networks," *Int. J. Netw. Manage.*, vol. 25, no. 6, pp. 375–393, Nov. 2015.

[63] O. B. Maia, H. C. Yehia, and L. D. Errico, "A concise review of the quality of experience assessment for video streaming," *Comput. Commun.*, vol. 57, pp. 1–12, Feb. 2015.

[64] P. Juluri, V. Tamarapalli, and D. Medhi, "Measurement of quality of experience of video-on-demand services: A survey," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 1, pp. 401–418, 1st Quart., 2016.

[65] M. Ghaznavi, E. Jalalpour, M. A. Salahuddin, R. Boutaba, D. Migault, and S. Preda, "Content delivery network security: A survey," *IEEE Commun. Surveys Tuts.*, vol. 23, no. 4, pp. 2166–2190, 4th Quart., 2021.

[66] B. Li, M. J. Golin, G. F. Italiano, X. Deng, and K. Sohraby, "On the optimal placement of web proxies in the internet," in *Proc. Conf. Comput. Commun. 18th Annu. Joint Conf. IEEE Comput. Commun. Soc. Future Now (IEEE INFOCOM)*, Mar. 1999, pp. 1282–1290.

[67] C. Yang, L. Huang, B. Leng, H. Xu, and X. Wang, "Replica placement in content delivery networks with stochastic demands and M/M/1 servers," in *Proc. IEEE 33rd Int. Perform. Comput. Commun. Conf. (IPCCC)*, Dec. 2014, pp. 1–8.

[68] Y. Chen, R. H. Katz, and J. D. Kubiatowicz, "Dynamic replica placement for scalable content delivery," in *Proc. Int. Workshop Peer-Peer Syst.*, 2002, pp. 306–318.

[69] J. Kangasharju, J. Roberts, and K. W. Ross, "Object replication strategies in content distribution networks," *Comput. Commun.*, vol. 25, no. 4, pp. 376–383, Mar. 2002.

[70] W. Benchaita, S. Ghamri-Doudane, and S. Tixeuil, "On the optimization of request routing for content delivery," in *Proc. ACM Conf. Special Interest Group Data Commun.*, Aug. 2015, pp. 347–348.

[71] V. Khare and B. Zhang, "CDN request routing to reduce network access cost," in *Proc. 37th Annu. IEEE Conf. Local Comput. Netw.*, Oct. 2012, pp. 610–617.

[72] H. A. Tran, S. Hoceini, A. Mellouk, J. Perez, and S. Zeadally, "QoE-based server selection for content distribution networks," *IEEE Trans. Comput.*, vol. 63, no. 11, pp. 2803–2815, Nov. 2014.

[73] N. Torres-Cruz, M. E. Rivero-Angeles, G. Rubino, R. Menchaca-Mendez, R. Menchaca-Mendez, and D. Ramirez, "A comprehensive analytical framework for VoD services in hybrid CDN-P2P systems," *J. Netw. Comput. Appl.*, vol. 161, Jul. 2020, Art. no. 102643.

[74] T. Hoeschele et al., "Importance of internet exchange point (IXP) infrastructure for 5G: Estimating the impact of 5G use cases," *Telecommun. Policy*, vol. 45, no. 3, p. 102091, 2021.

[75] T. Bottger et al., "The elusive internet flattening: 10 years of IXP growth," *CoRR*, 2018.

[76] R. Klöti, B. Ager, V. Kotronis, G. Nomikos, and X. Dimitropoulos, "A comparative look into public IXP datasets," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 46, no. 1, pp. 21–29, Jan. 2016.

[77] Z. Li, Q. Wu, K. Salamatian, and G. Xie, "Video delivery performance of a large-scale vod system and the implications on content delivery," *IEEE Trans. Multimedia*, vol. 17, no. 6, pp. 880–892, Jun. 2015.

[78] G. Zhang, W. Liu, X. Hei, and W. Cheng, "Unreeling xunlei kankan: Understanding hybrid CDN-P2P video-on-demand streaming," *IEEE Trans. Multimedia*, vol. 17, no. 2, pp. 229–242, Feb. 2015.

[79] K. D. Joshi and K. Kataoka, "PRIME-*Q*: Privacy aware end-to-end QoS framework in multi-domain SDN," in *Proc. IEEE Conf. Netw. Softw. (NetSoft)*, Jun. 2019, pp. 169–177.

[80] G. Bertrand, E. Stephan, T. Burbridge, P. Eardley, K. Ma, and G. Watson, *Use Cases for Content Delivery Network Interconnection*, document RFC 6770, 2012.

[81] E. Turrini, "An architecture for content distribution internet working," France Telecom-Orange, Univ. Bologna, Bologna, Italy, Tech. Rep. UBLCS-2004-2, 2004.

[82] R. Buyya, A.-M. Pathan, J. Broberg, and Z. Tari, "A case for peering of content delivery networks," *IEEE Distrib. Syst. Online*, vol. 7, no. 10, pp. 3, Oct. 2006.

[83] N. Mowla, I. Doh, and K. Chae, "CSDSM: Cognitive switch-based DDoS sensing and mitigation in SDN-driven CDNi word," *Comput. Sci. Inf. Syst.*, vol. 15, no. 1, pp. 163–185, 2018.

[84] S. D. I. Fabra, S. E. N. Cánovas, G. G. Bofrisco, B. D. Díaz, and A. N. Tchernykh, "Design and optimization of content distribution networks," *Proc. Inst. Syst. Program. RAS*, vol. 31, no. 2, pp. 15–20, 2019.

[85] A.-M.-K. Pathan, J. A. Broberg, K. Bubendorfer, K. H. Kim, and R. Buyya, "An architecture for virtual organization (VO)-based effective peering of content delivery networks," in *Proc. 2nd Workshop Use P2P, GRID Agents Develop. Content Netw.*, Jun. 2007, pp. 29–38.

[86] M. Rodrigues, S. Fernandes, J. Kelner, and D. Sadok, "An analytical view of multiple CDNs collaboration," in *Proc. IEEE 28th Int. Conf. Adv. Inf. Netw. Appl.*, May 2014, pp. 25–32.

[87] L. Amini, A. Shaikh, and H. Schulzrinne, "Effective peering for multi-provider content delivery services," in *Proc. IEEE infocOM*, Mar. 2004, pp. 850–861.

[88] M. Pathan and R. Buyya, "Architecture and performance models for QoS-driven effective peering of content delivery networks," *Multiagent Grid Syst.*, vol. 5, no. 2, pp. 165–195, Jul. 2009.

[89] C. Courcoubetis, L. Gyarmati, N. Laoutaris, P. Rodriguez, and K. Sdrolias, "Negotiating premium peering prices: A quantitative model with applications," *ACM Trans. Internet Technol.*, vol. 16, no. 2, pp. 1–22, Apr. 2016.

[90] M. Pathan and R. Buyya, "A utility model for peering of multi-provider content delivery services," in *Proc. IEEE 34th Conf. Local Comput. Netw.*, Oct. 2009, pp. 475–482.

[91] C. Natalino, A. Sousa, L. Wosinska, and M. Furdek, "Content placement in 5G-enabled edge/core data center networks resilient to link cut attacks," *Networks*, vol. 75, no. 4, pp. 392–404, Jun. 2020.

[92] T. Higuchi, G. S. Pannu, F. Dressler, and O. Altintas, "Content replication in vehicular micro cloud-based data storage: A mobility-aware approach," in *Proc. IEEE Veh. Netw. Conf. (VNC)*, Dec. 2018, pp. 1–4.

[93] H. Yan, Z. Chen, Z. Wang, and W. Zhu, "DRL-based collaborative edge content replication with popularity distillation," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jul. 2021, pp. 1–6.

[94] Q. Gao, F. Wang, and L. Gao, "Routing-policy aware peering for large content providers," *Comput. Commun.*, vol. 67, pp. 23–33, Aug. 2015.

[95] P. Maillé, G. Simon, and B. Tuffin, "Impact of revenue-driven CDN on the competition among network operators," in *Proc. 11th Int. Conf. Netw. Service Manage. (CNSM)*, Nov. 2015, pp. 163–167.

[96] M. Elkotob and K. Andersson, "Challenges and opportunities in content distribution networks: A case study," in *Proc. IEEE Globecom Workshops*, Dec. 2012, pp. 1021–1026.

[97] Z. Akhtar, A. Hussain, E. Katz-Bassett, and R. Govindan, "DBit: Assessing statistically significant differences in CDN performance," *Comput. Netw.*, vol. 107, pp. 94–103, Oct. 2016.

[98] M. Mu et al., "A scalable user fairness model for adaptive video streaming over SDN-assisted future networks," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 8, pp. 2168–2184, Aug. 2016.

[99] X. Huang, S. Cheng, K. Cao, P. Cong, T. Wei, and S. Hu, "A survey of deployment solutions and optimization strategies for hybrid SDN networks," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 2, pp. 1483–1507, 2nd Quart., 2018.

[100] J. Wang, J. An, M. Chen, N. Zhan, L. Wang, M. Zhang, and T. Gan, "From model to implementation: A network algorithm programming language," *Sci. China Inf. Sci.*, vol. 63, no. 7, pp. 1–17, Jul. 2020.

[101] D. S. Berger and R. K. Sitaraman, "AdaptSize: Orchestrating the hot object memory cache in a content delivery network," in *Proc. 14th USENIX Symp. Netw. Syst. Design Implement. (NSDI)*, 2017, pp. 483–498.

[102] J. Zhao, P. Liang, W. Liufu, and Z. Fan, "Recent developments in content delivery network: A survey," in *Proc. Int. Symp. Parallel Architectures, Algorithms Program.*, 2019, pp. 98–106.

[103] R. Amin, M. Reisslein, and N. Shah, "Hybrid SDN networks: A survey of existing approaches," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 4, pp. 3259–3306, 4th Quart., 2018.

[104] B. Zolfaghari, G. Srivastava, S. Roy, H. R. Nemati, F. Afghah, T. Koshiba, A. Razi, K. Bibak, P. Mitra, and B. K. Rai, "Content delivery networks: State of the art, trends, and future roadmap," *ACM Comput. Surv.*, vol. 53, no. 2, pp. 1–34, 2020.

[105] Y. Liu, J. Ding, Z.-L. Zhang, and X. Liu, "CLARA: A constrained reinforcement learning based resource allocation framework for network slicing," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2021, pp. 1427–1437.

[106] K. Bouraqia, E. Sabir, M. Sadik, and L. Ladid, "Quality of experience for streaming services: Measurements, challenges and insights," *IEEE Access*, vol. 8, pp. 13341–13361, 2020.

[107] J. Liu, Q. Yang, and G. Simon, "Congestion avoidance and load balancing in content placement and request redirection for mobile CDN," *IEEE/ACM Trans. Netw.*, vol. 26, no. 2, pp. 851–863, Apr. 2018.

[108] Y. Li, S. Guan, C. Zhang, and W. Sun, "Parameter optimization model of heuristic algorithms for controller placement problem in large-scale SDN," *IEEE Access*, vol. 8, pp. 151668–151680, 2020.

[109] D. Sarkar and N. Rakesh, "Hybrid load scheduling in content delivery network comprising of under populated clusters," in *Proc. 9th Int. Conf. Rel., infocom Technol. Optim. (Trends Future Directions) (ICRITO)*, Sep. 2021, pp. 1–6.

[110] R.-X. Zhang, M. Ma, T. Huang, H. Pang, X. Yao, C. Wu, J. Liu, and L. Sun, "Livesmart: A QoS-guaranteed cost-minimum framework of viewer scheduling for crowdsourced live streaming," in *Proc. 27th ACM Int. Conf. Multimedia*, Oct. 2019, pp. 420–428.

[111] Z. Lu, K. Gai, Q. Duan, K.-K.-R. Choo, J. Li, J. Wu, and Y. Xu, "Machine learning empowered content delivery: Status, challenges, and opportunities," *IEEE Netw.*, vol. 34, no. 6, pp. 228–234, Nov. 2020.

[112] H. Li, L. He, H. Zhang, K. Zhang, X. Li, and C. He, "CDN-hosted domain detection with supervised machine learning through DNS records," in *Proc. 3rd Int. Conf. Inf. Sci. Syst.*, Mar. 2020, pp. 144–149.

[113] Z. Ma, S. Roubia, F. Giroire, and G. Urvoy-Keller, "When Locality is not enough: Boosting peer selection of Hybrid CDN-P2P live streaming systems using machine learning," in *Proc. Netw. Traffic Meas. Anal. Conf. (IFIP TMA)*, Virtual, France, Sep. 2021.

[114] Y. Maleh, Y. Qasmaoui, K. El Gholami, Y. Sadqi, and S. Mounir, "A comprehensive survey on SDN security: Threats, mitigations, and future directions," *J. Reliable Intell. Environ.*, pp. 1–39, Feb. 2022.

[115] A. M. K. Pathan and R. Buyya, "A taxonomy and survey of content delivery networks," Grid Comput. Distrib. Syst. Lab., Univ. Melbourne, Melbourne, VIC, Australia, Tech. Rep. 4, 2007, p. 70.

[116] S. S. Sarma and S. K. Setua, "Design and implementation of a hierarchical content delivery network interconnection model," in *Proc. 3rd Int. Conf. Adv. Comput., Netw. Informat.*, 2016, pp. 71–82.

[117] J. Konečný, H. B. McMahan, D. Ramage, and P. Richtárik, "Federated optimization: Distributed machine learning for on-device intelligence," 2016, *arXiv:1610.02527*.

[118] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Process. Mag.*, vol. 37, no. 3, pp. 50–60, May 2020.

[119] H. Guo, A. Liu, and V. K. N. Lau, "Analog gradient aggregation for federated learning over wireless networks: Customized design and convergence analysis," *IEEE Internet Things J.*, vol. 8, no. 1, pp. 197–210, Jan. 2021.

[120] C. Zhang, Y. Xie, H. Bai, B. Yu, W. Li, and Y. Gao, "A survey on federated learning," *Knowl.-Based Syst.*, vol. 216, Mar. 2021, Art. no. 106775.

[121] Q. Li, Z. Wen, Z. Wu, S. Hu, N. Wang, Y. Li, X. Liu, and B. He, "A survey on federated learning systems: Vision, hype and reality for data privacy and protection," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 4, pp. 3347–3366, Apr. 2023.

[122] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Trans. Intell. Syst. Technol.*, vol. 10, no. 2, pp. 1–19, 2019.

[123] X. Ma, L. Liao, Z. Li, R. X. Lai, and M. Zhang, "Applying federated learning in software-defined networks: A survey," *Symmetry*, vol. 14, no. 2, p. 195, Jan. 2022.

[124] Q. Chen, F. R. Yu, T. Huang, R. Xie, J. Liu, and Y. Liu, "An integrated framework for software defined networking, caching, and computing," *IEEE Netw.*, vol. 31, no. 3, pp. 46–55, May/Jun. 2017.

[125] Q. Chen, F. R. Yu, T. Huang, R. Xie, J. Liu, and Y. Liu, "Joint resource allocation for software-defined networking, caching, and computing," *IEEE/ACM Trans. Netw.*, vol. 26, no. 1, pp. 274–287, Feb. 2018.

[126] G. Li et al., "Multitentacle federated learning over software-defined industrial Internet of Things against adaptive poisoning attacks," *IEEE Trans. Ind. Informat.*, vol. 19, no. 2, pp. 1260–1269, 2022.

[127] X. Han and G. Chen, "Research on SDN based CDN architecture and key technologies," *Internet World*, vol. 8, pp. 17–22, 2014.

[128] D. Kim and Y. Kim, "Enhancing NDN feasibility via dedicated routing and caching," *Comput. Netw.*, vol. 126, pp. 218–228, Oct. 2017.

[129] J. Badshah, M. M. Alhaisoni, N. Shah, and M. Kamran, "Cache servers placement based on important switches for SDN-based ICN," *Electronics*, vol. 9, no. 1, p. 39, Dec. 2019.

[130] J. Badshah, M. Kamran, N. Shah, and S. A. Abid, "An improved method to deploy cache servers in software defined network-based information centric networking for big data," *J. Grid Comput.*, vol. 17, no. 2, pp. 255–277, 2019.

[131] K. Kirkpatrick, "Software-defined networking," *Commun. ACM*, vol. 56, no. 9, pp. 16–19, Sep. 2013.

[132] A. Greenberg, G. Hjalmtysson, D. A. Maltz, A. Myers, J. Rexford, G. Xie, H. Yan, J. Zhan, and H. Zhang, "A clean slate 4D approach to network control and management," *ACM SIGCOMM Comput. Commu. Rev.*, vol. 35, no. 5, pp. 41–54, Oct. 2005.

[133] M. Casado, M. J. Freedman, J. Pettit, J. Luo, N. McKeown, and S. Shenker, "Ethane: Taking control of the enterprise," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 37, no. 4, pp. 1–12, 2007.

[134] P. Berde, M. Gerola, J. Hart, Y. Higuchi, M. Kobayashi, T. Koide, B. Lantz, B. O'Connor, P. Radoslavov, W. Snow, and G. Parulkar, "ONOS: Towards an open, distributed SDN OS," in *Proc. 3rd Workshop Hot Topics Softw. Defined Netw.*, 2014, pp. 1–6.

[135] J. Rexford and C. Dovrolis, "Future internet architecture," *Commun. ACM*, vol. 53, no. 9, pp. 36–40, 2010.

[136] M. Ibrar, L. Wang, G.-M. Muntean, J. Chen, N. Shah, and A. Akbar, "IHSF: An intelligent solution for improved performance of reliable and time-sensitive flows in hybrid SDN-based FC IoT systems," *IEEE Internet Things J.*, vol. 8, no. 5, pp. 3130–3142, Mar. 2021.

[137] H.-A. Tran, S. Souihi, D. Tran, and A. Mellouk, "MABRESE: A new server selection method for smart SDN-based CDN architecture," *IEEE Commun. Lett.*, vol. 23, no. 6, pp. 1012–1015, Jun. 2019.

HUIXIANG YANG received the B.E. degree from Jilin University, in 1987, and the M.S. degree from the Dalian University of Technology, in 2006. She is currently a Professor with the School of Mechatronic Engineering, Changchun University of Technology, China. Her research interests include network security and distributed computing, including theories, models, and algorithms.

HANLIN PAN received the B.E. degree from the School of Software, Tsinghua University, in 2020, and the master's degree from the College of Business, City University of Hong Kong, in 2022. He is currently pursuing the Ph.D. degree with the College of Computer Science, Jilin University.

LIN MA received the B.E. degree in electronic information science and technology and the master's degree from the College of Computer Science and Technology, Jilin University, in 2017 and 2023, respectively, where he is currently pursuing the Ph.D. degree.

• • •