IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

SPECIAL SECTION ON DATA MINING FOR INTERNET OF THINGS

Received October 20, 2020, accepted October 30, 2020, date of publication November 10, 2020, date of current version December 7, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3037164

An Efficient Content Distribution Network Architecture Using Heterogeneous Channels

YUQIANG WEN[1], YUQIANG CHEN[2], MENG-LIANG SHAO[3], JIAN-LAN GUO[2], AND JIA LIU[1]

[1]Guangzhou Nanyang Polytechnic College, Guangzhou 510925, China
[2]Dongguan Polytechnic, Dongguan 523808, China
[3]South China Institute of Software Engineering GU, Guangzhou 510990, China

Corresponding author: Yuqiang Chen (chenyuqiang@126.com)

This work was supported in part by the Key Natural Science Research Projects of New Generation of Information Technology in Focus Areas in 2020 of General Universities in Guangdong Province Key Technology Research on Channel Heterogeneous CDN under Grant 2020ZDZX3108, in part by the Characteristic Natural Science Innovation Projects of General Universities in Guangdong Province in 2020 Research on Control System and Key Technology of Intelligent Arm Type Iron Roughneck under Grant 2020KTSCX380, in part by the Key Research Natural Science Projects of General Universities in Guangdong Province in 2020 Research and Development of Key Technologies of Real-Time Big Data Processing Platform, in part by the 2018 Guangzhou Planned Science and Technology Project—Research and Development of Key Technologies for Real-time Big Data Processing Middleware under Grant 201804010402, in part by The Information Technology Innovation and Application Base Project 2018 of Guangzhou Nanyang Polytechnic College under Grant NY-2018CQPT-01, in part by the 2018 Guangdong Provincial Department of Education University Youth Innovative Talents Research Project—Research on Interactive Design of 3D Animation Based on Virtual Reality Technology under Grant 2018GkQNCX042, in part by the 13th Five-Year Plan of Philosophy and Social Science Development in Guangzhou co-constructed the project of 2020 Research on the Mechanism of Urban Waste Classification and Recycling in the Artificial Intelligence Environment under Grant 2020GZGJ315, and in part by the Key Projects of Social Science and Technology Development in Dongguan under Grant 2020507156156.

ABSTRACT With the popularization and development of the IoT(Internet of Things), more and more data needs to be transmitted over the Internet, which leads to the deterioration of network quality. The CDN (Content Distribution Network) technology is an important theoretical model to ensure network QoS (Quality of Service). To improve the QoS, we extend current CDN system to the hierarchical CDN system, that traditional CDN system is just a special case of hierarchical CDN system. In the traditional CDN system and the hierarchical CDN system, by analyzing the bandwidth between subsystems, we found the inter-system bandwidth is the bottleneck that impedes CDN system expansion. To address this problem, a new kind of distributed system architecture is proposed in this paper. This new architecture uses broadcast channels to distribute broadcast type data and still using bidirection uni-cast channels for other type of data, so we call the new architecture as CHCDN (Channel Heterogeneous CDN) in this paper. The new architecture is analyzed and compared with the traditional CDN system architecture and hierarchical CDN system architecture. Moreover, the experimental simulation result has shown that the CHCDN system features better in real-time cache updating and features with higher data transmission efficiency than the hierarchical CDN system, which indicating that the new architecture has great potential for being widely used in distributed computing.

INDEX TERMS Real time CDN cache update, Internet of Things (IoT), channel heterogeneous CDN (CHCDN), network QoS, distributed computing.

I. INTRODUCTION

Network traffic shows an increasing trend [1]. The CDN technology has been widely used to guarantee network QoS in the wake of data traffic growth. Today, many scholars have conducted fruitful research on the CDN technology.

In [2], Dehghan *et al.* studied the joint routing problem and cache replacement problem to optimize the total delay of user requests. This kind of problem is a NP-complete problem. To analyze the complexity of the optimization problem, Dehghan *et al.* proposed two sub-problems of non-cached path delay congestion sensitivity and non-cached path delay congestion insensitivity. In the non-cache path delay congestion insensitive problem, if there is only one user request for one datum in the system or only two cache devices in the network, the problem can be reduced to a polynomial time solvable problem. However, in an actual system, its optimization assumption is difficult to meet, and the real-time performance

The associate editor coordinating the review of this manuscript and approving it for publication was Mu-Yen Chen.

of the optimization algorithm proposed in [2] is difficult to achieve. In [3], a system model with the maximum delay of point-to-point content transmission as QoS constraint condition was proposed. The optimization model was a NP-Hard nonlinear problem; the system was linearized in three different ways, and two linear models were solved respectively by specific optimization algorithms based on Benders decomposition and Lagrangean relaxation algorithm to obtain the final system scheduling decision. However, in the process of resource scheduling, the scheduling activity itself aggravated the congestion of the network. Only when the scheduling was completed could users access hot resources smoothly. In [4], Aioffi *et al.* proposed a statistical prediction-based double exponential smoothing heuristic dynamic content replacement online algorithm to optimize CDN resource scheduling. The algorithm predicted user requests to optimize cache updates. This proposed optimization algorithm could achieve the optimization of the overall network traffic, but its model lacked quantitative indicators of QoS and had poor effect on QoS optimization. In [5], Tang studied the cache content replacement problem with QoS constraints by the graph theory method, and found a polynomial complexity replacement strategy for the replica-blind class replacement sub-problem proposed in their paper, but the data transmission problem during replacement was not considered, and the proposed problem solving algorithm could only perform static analysis on the system. SLA (service level agreement) is a QoS agreement signed by CDN suppliers and users, which contains QoS terms. In [6], Kolisch *et al.* proposed a system model with SLA as the constraint objective, and the system optimization model was a NP-Hard problem. They used different routing rules in the multicast phase to effectively reduce the transmission time, and a dynamic cache update algorithm was implemented using a simulated annealing algorithm to meet SLA constraints. Fatin [7] try to minimize the storage and the delivery costs and at the same time to satisfy SLA, they use genetic algorithm (GA) to extract a mixed integer programming (MIP) formulation for the problem's optimum solutions.With SLA as the constraint, the improvement of cache update real-time was not the fundamental optimization goal of its algorithm. In [8], Alabbasi *et al.* studied the over-the top video streaming system based on hierarchical CDN, and proposed using SDTP (stall duration tail probability) as the optimization target. Through two-stage probability scheduling of downloading video clips and playing clips, they realized optimization method of multiple system variables such as bandwidth allocation under QoS constraints, multi-layer cache updating, user request scheduling and the like. However, because the algorithm dynamically selected the server where the video files were placed and could not ensure the real-time bandwidth allocation, the scheme could not solve the problem when the code rate of the dynamic video stream changed. Other methods were also proposed in previous studies. For instance, the peer-to-peer network Tapestry was used as the core of traffic updating and was deployed in IDC; the traffic was distributed through the d-trees and Tapestry was deployed close to the backbone network so as to achieve cache updating efficiency [9]. Similarly, with the popularization of cloud computing and NFV (Network Functions Virtualization) and other technologies, CDN has also migrated to the cloud [10], [11]. Cloud update can achieve faster update speed and lower cost driven by mass scale and new technologies for cloud computing [12]–[14]. The problem of CDN update has been transformed into a data center management problem. At this time, the goal of CDN optimization was to reduce the cost of optimization of the data center [15]; In the methods mentioned in [9]–[11], [15], there was no essential change in the real-time nature of the CDN system itself, and the CDN efficiency was improved by external factors. Some studies have proposed shared channel schemes [16]–[19] in the wireless network environment. These schemes were helpful to improve QoS, but the broadcast channel sharing scale of these schemes was too small and the improvement was not significant. Leyva-mayorga *et al.* [20] propose a network-coded cooperation (NCC) protocol in the 5G network environment, this scheme was helpful to improve bandwidth usage, only WIFI part use broadcast channel sharing technology, all over improvement was not significant. New network architectures such as the edge computing scheme [21] or fog computing scheme [22], [23] or ICN (Information Centric Networking) based architecture scheme [24]–[26] are more reasonable. However, in the new architecture, how to realize and optimize the underlying cache still depends on the development of the CDN technology, and these underlying problems will not disappear automatically due to the adoption of those new network architectures.

The above mentioned studies on CDN has made a substantial contribution to the improvement of network QoS by using CDN technology, making CDN technology mature continuously. However, none of the above mentioned algorithms can solve the real-time problem of QoS. This paper proposes a scheme of constructing the CDN system using broadcast channel -CHCDN system, which can update cache contents in real time. The CHCDN system greatly improves the QoS level of the network. In the following sections, the bottleneck of the traditional CDN system, namely the inter-system bandwidth problem, is analyzed, and the improvement factors of the CHCDN scheme on the transmission efficiency and QoS effect of the CDN system are explored, and at last, the excellent performance of the CHCDN system is verified through experimental simulation.

II. CDN SYSTEM MODEL

To understand current CDN system we need to trace back to the time of the creation of CDN system.

A. CDN THEORETICAL MODEL

The random tree model is the framework basis of CDN systems [27], [28] . At present, most CDN systems are based on this model. The random tree model mainly comprises browsers, servers and cache devices, as shown in Fig.1:

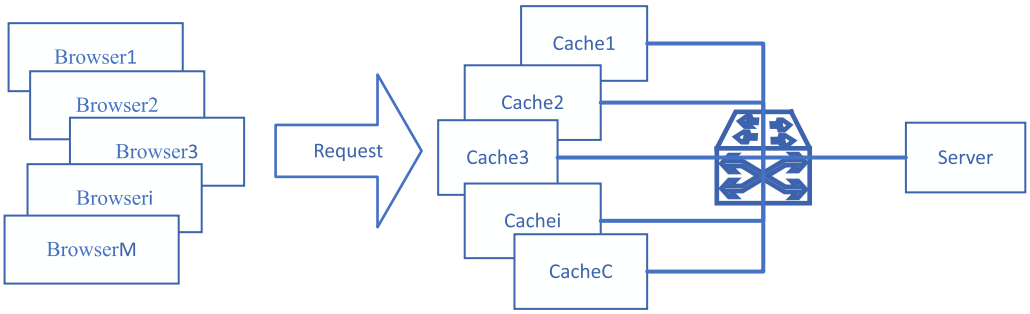

FIGURE 1. Random tree system model.

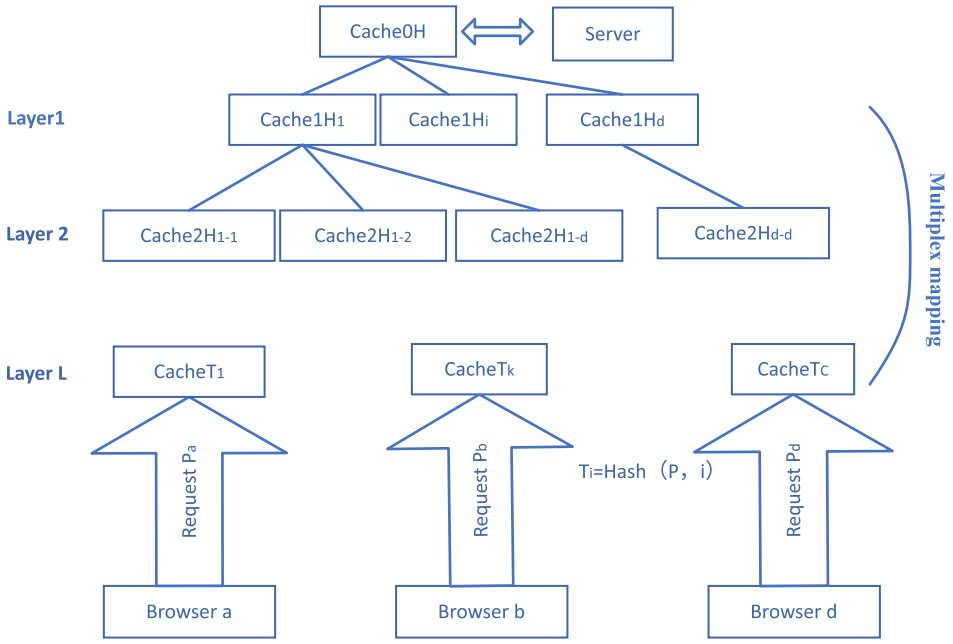

FIGURE 2. Construction of the random tree.

The system working protocol for the random tree model system is as follows:

The browser sends a request to a page on the server, and the request will be sent to a Cache device. The Cache device that accepts the browser request can be regarded as a logical leaf node. After receiving the request, the Cache device inquires whether there is this page locally or whether it is requesting this page itself. If yes, it returns the page directly to the browser. Otherwise, the request count is accumulated for the corresponding page and the request is forwarded to the next node on the path. When the Cache device obtains a page, if the page count exceeds the threshold (q), the Cache device will cache the page. After receiving the page request, the server will return to the corresponding page.

The path where the browser request is satisfied is logically constructed as the path from the leaf node to the root node, thus forming a tree. Nodes in the tree will be mapped to a physical Cache device by a hash function, thus forming a random tree, as shown in Fig.2:

By solving the random tree model problem, the key parameters related to the QoS of the system can be obtained, including the longest path of the random tree that the browser requests the page to go through, and the maximum traffic of each cache device under the probability of 1-1/N, specifically:

1) The longest path where the request is satisfied *MaxDelay*:

$$MaxDelay = 2 * \log_d C \tag{1}$$

C in (1) represents the number of Cache devices in the random tree model system; d represents the degree of the balanced logic tree;

2) Under the circumstance of the (1-1/N) probability, the maximum access MaxLoad of any Cache device is:

$$MaxLoad = \rho(2\log_d C + O(\frac{\log N}{\log\log N})) + O(\frac{dq\log N}{(\log\frac{dq}{\rho}\log N)} + \log N) \tag{2}$$

N in (2) represents a measure of probability; ρ denotes Cache processing capability; q indicates the count threshold, when the count exceeds q, the Cache device will cache this page.

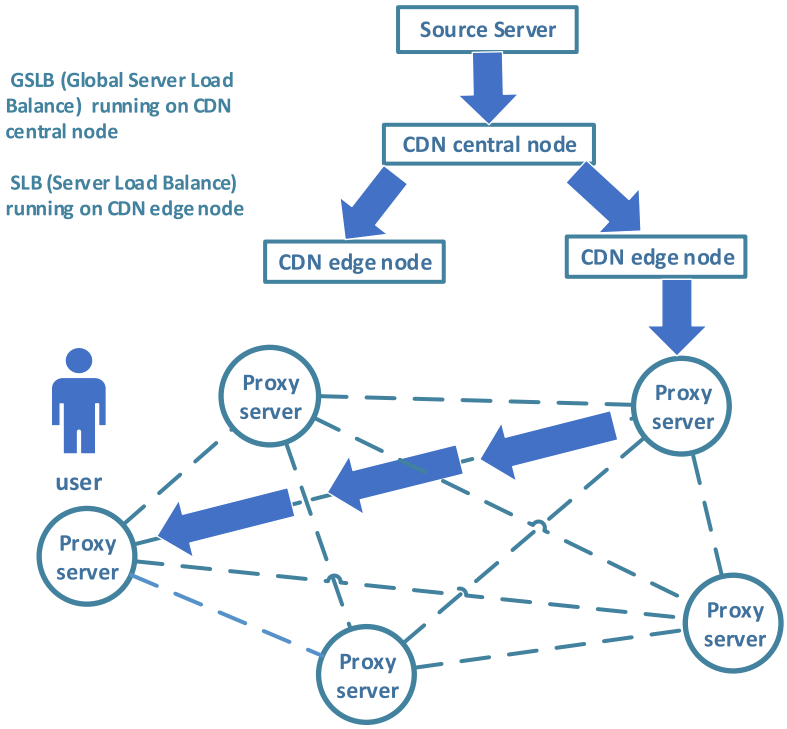

FIGURE 3. Current CDN system.

In random tree model, the cache servers can cache all data the server have, and the inter cache link's bandwidth is enough for inter cache communication. So, this model now is used in proxy servers' level of CDN system, which is the level directly connected with end user. And this level of proxy servers connected with CDN edge node, in this scene CDN edge node is the server in random tree model, and the proxy servers are cache servers in random tree model. As shown in Fig.3:

It is impossible for a single isolated system (a CDN edge node) to store all contents and directly process all user requests, so only part of request can be severed by cached content, the number of requests severed by cached content compared to the total requests number is hit rate of this cache. The new algorithm proposes to optimize the resource placement and access problems between CDN nodes [2]–[11], but these optimization algorithms are not real-time. If the system is expanded and QoS of the system is guaranteed to remain unchanged, the resource optimization algorithm must be run in real-time. To achieve real-time updating of resources, if the system still uses the random tree model and all resources are evenly distributed in the system, then according to (2), since the total page request volume of the system and the cache volume in the system are huge, MaxLoad will be the task volume that cannot be handled by a single node. in addition, the nodes will realize communication through WAN links to ensure the algorithm of the random tree model to run, and the WAN traffic consumption is also huge.

B. THEORETICAL MODEL OF THE HIERARCHICAL CDN SYSTEM

A random tree model system cannot have all resources. In order to ensure the quality of CDN service and improve the scalability of the system, it's natural to extend current CDN system to a hierarchical tree like structure system with the root node of the tree being the internet. The system as a whole can store huge amounts of resources. Each node in the system is a subsystem. The interior of the subsystem is built according to the random tree model, as shown in Fig.4.

Users access the Internet through Layer 1. Cache in each layer of the hierarchical model can serve users in its lower layer. User requests will only be forwarded to higher layer systems when the lower layer systems cannot meet the requirements. All non-broadcast data traffic will pass through the Internet interface of the system. The quality of the user's online surfing experience is calculated as follows:

$$l_{user} = r_b * \sum_{n=1}^{L_{int}} (\alpha_n * Q_n * \prod_{i=1}^{n} l_i) + (1 - r_b) * \prod_{n=1}^{L_{int}} l_n$$

$$s.t. \sum_{n=1}^{L_{int}} \alpha_n = 1, \quad 0 \le \alpha_n \le 1, \ 0 \le r_b \le 1,$$

$$0 \le Q_n \le 1, \quad 0 \le l_n \le 1, \ 0 \le n \le L_{int1} \quad (3)$$

l_{user} consists of broadcast data transmission QoS and non-broadcast data transmission QoS. In (3), r_b represents the proportion of users accessing broadcast resources, Q_n represents the CDN service quality of Layer n cache,l_n represents the

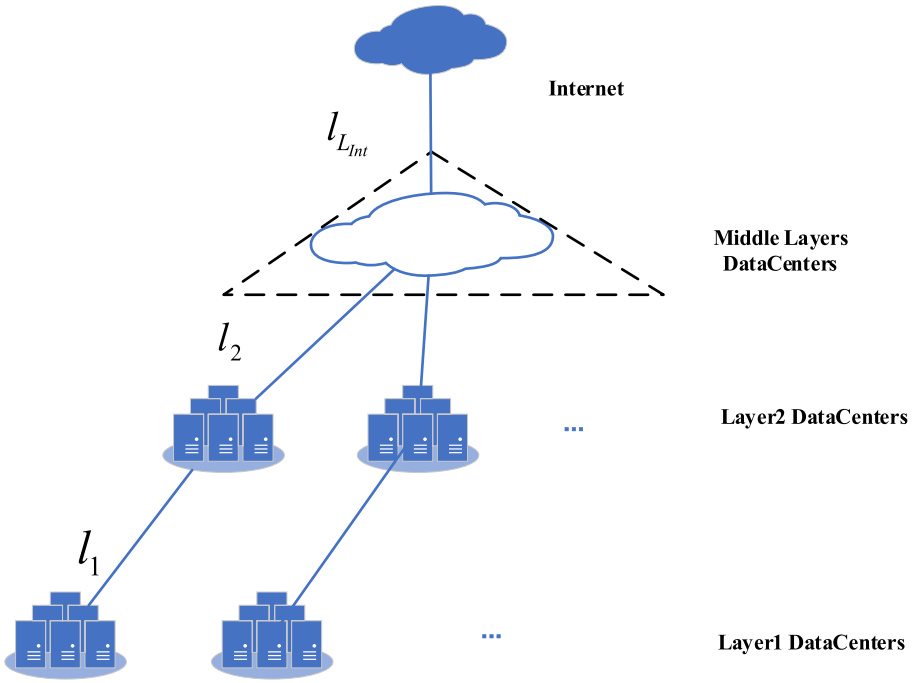

FIGURE 4. Hierarchical CDN system.

link quality from Layer n to Layer (n + 1), α_n represents the hit rate of Layer n, and L_{int} represents the hierarchical distance between users and the Internet. α_n is a random variable produced by subsystem, we do not restrict the resource allocation algorithm in subsystem to get α_n, we use α_n as control variable to analyze the inter layer bandwidth. If the l_n of a layer is very low, then the CDN service of a layer can hardly improve l_{user}. For example, GSLB (Global Server Load Balance) and SLB (Server Load Balance), the two-tier CDN scheduling schemes, as shown in Fig.3, are equivalent to only one-tier CDN cache system under QoS constraints. In this scheme, the higher-level CDN contributes to increasing Q_1 but the higher-level CDN contribute little to improve l_{user}. After all levels of cache in the hierarchical CDN network have been updated, through the cooperation of CDN layers, users will have a better online experience than the traditional CDN two-layer scheduling scheme. So we will use the hierarchical CDN network as benchmark architecture to compare our novel network architecture in later sections. Bandwidth between systems after layering is WAN bandwidth.

For the nodes in each layer in the hierarchical CDN network framework, each node in the lower layer is regarded as a user of this layer. The internal implementation of each node is still a random tree model system. The total request amount R_{i+1} for all users in layer i + 1 to access broadcast resources depends on the sum of requests for broadcast resources missed by all users in the lower layer, and the recurrence formula of R_{i+1} is:

$$R_{i+1} = (1 - \alpha_i) \sum R_i \quad s.t.\ 0 < \alpha_i < 1, \quad 1 \le i \le L_{int} \tag{4}$$

in (4) α_i is a random variable produced by subsystem, it represents the hit rate of Layer i, R_i is random variable which follow empirical distribution of ZIPF.

If the number of cache servers in each tier of data center is C_i, then the processing capacity requirement of each cache service is ρ_i:

$$\rho_i = R_i / C_i \quad s.t.\ 1 \le i \le L_{int} \tag{5}$$

The total amount of storage required in a subsystem is determined according to the user's behavior habits. In order to reduce the bandwidth of the subsystem to the superior system, the subsystem user's access should be satisfied in the subsystem as much as possible, that is, the hit rate α_i should be as high as possible, which requires a larger storage capacity in the system. We assume that the user behavior is similar, and the amount of network traffic per unit time is limited. We assume the network traffic per unit time is D_i, the repetition rate of all hot spot data is θ_i, and M_i is the number of users, the non-repetitive storage capacity S_i required per unit time of the system is:

$$S_i = M_i * D_i * \alpha_i * (1 - \theta_i) \quad s.t.\ 0 < \theta_i < 1 \tag{6}$$

D_i is upper bounded by ISP's (Internet Service Provider) service contracts. With M_i increasing, the chances that some users will have same interest in same content will increase, so θ_i will increase, then α_i can be increased with M_i increase, we can just have brief discussion about the relations between resource availability constraints and α_i here. The more deep

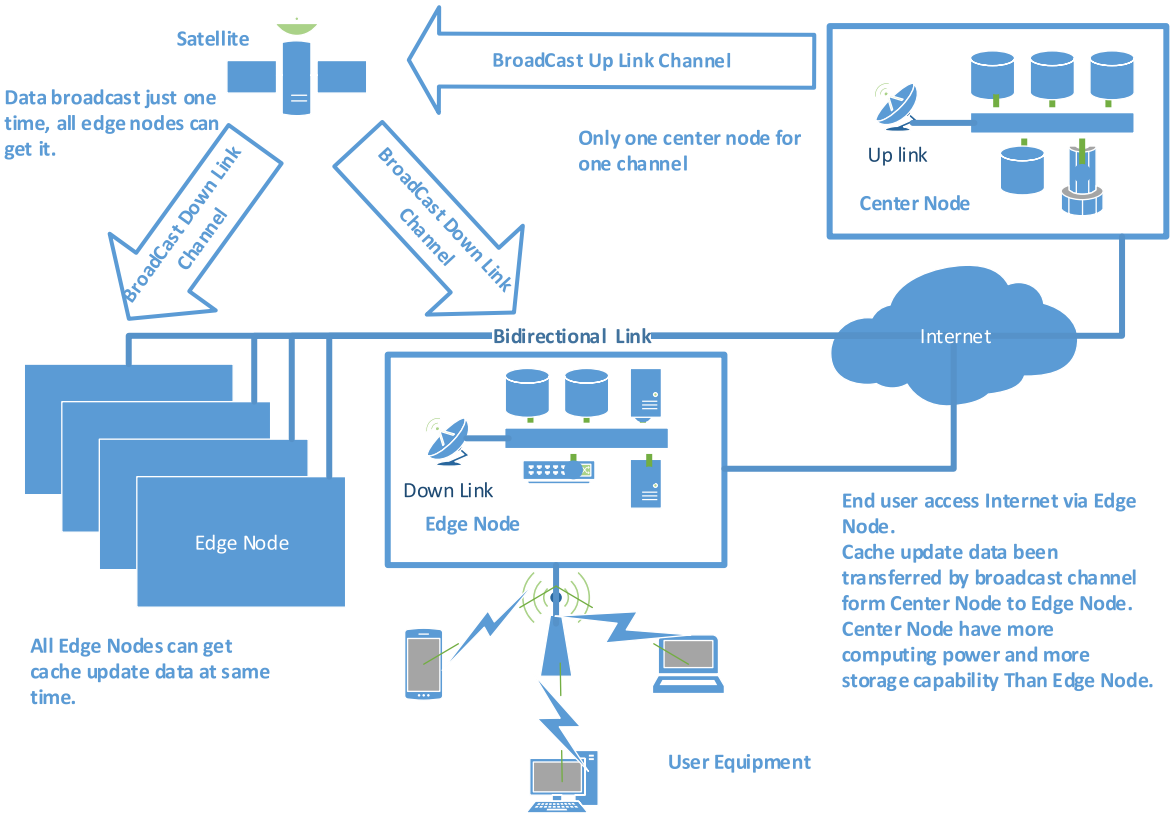

FIGURE 5. Architecture of CHCDN system.

discussing on this topic is out of scope of this paper, and the topic deserve another whole paper.

Based on (3), (4), (5), (6), we can analyze the performance of the layered CDN system. Since the above analysis is made without updating CDN cache content, these conclusions are only static analysis of the system. The QoS guarantee of the network depends on the improvement on the heterogeneous CDN system when Cache content of the hierarchical CDN system is dynamically updated.

III. IMPROVEMENT ON THE CHCDN SYSTEM

The data that needs to be transmitted through the CDN system must meet a precondition, that is, the data obtained by all users is the same when the data is sent to users. We call data with such characteristics broadcast data. For example, some video data, web static data, software upgrade packages and some GIS (Geographic Information System) data. Based on the broadcast nature of data and the nature of broadcast channels, we propose a new CDN architecture called CHCDN [29]. The architecture has the characteristics of high content distribution efficiency, real-time cache update and strong system scalability.

A. ANALYSIS OF THE CHCDN SYSTEM ARCHITECTURE

After the CDN system joins the broadcast channel, the updated content is transmitted through the broadcast channel, only once, and all cache nodes can receive the relevant data content at the same time. Its architecture is as shown in Fig.5.

The central node is responsible for collecting and transmitting the broadcast data. The central node is a computing cluster and is also connected to the Internet. Broadcast data is sent to edge nodes through broadcast channels, taking satellite channels as an example. Edge nodes provide services for local users by receiving data from broadcast channels and caching the data locally. Here, the edge node is an edge computing center and is also connected to the Internet.

The CHCDN system makes full use of broadcast channel characteristics. In order to make Cache update more effective, its interaction protocol is improved as follows:

The central node arranges the content to be released through the broadcast channel according to the obtained information prediction [30]–[33] or the channel reservation of the content operator. At the same time, the network traffic information related to immediate hot events is summarized, and different bandwidths and priorities are allocated to different data. The distribution of data is sorted according to priority and then transmitted through broadcast channels. In data organization, different types of data will be classified, and the receiver of broadcast channel will easily filter out the data it needs. The edge node consists of a broadcast channel data receiving end and an edge computing center connected

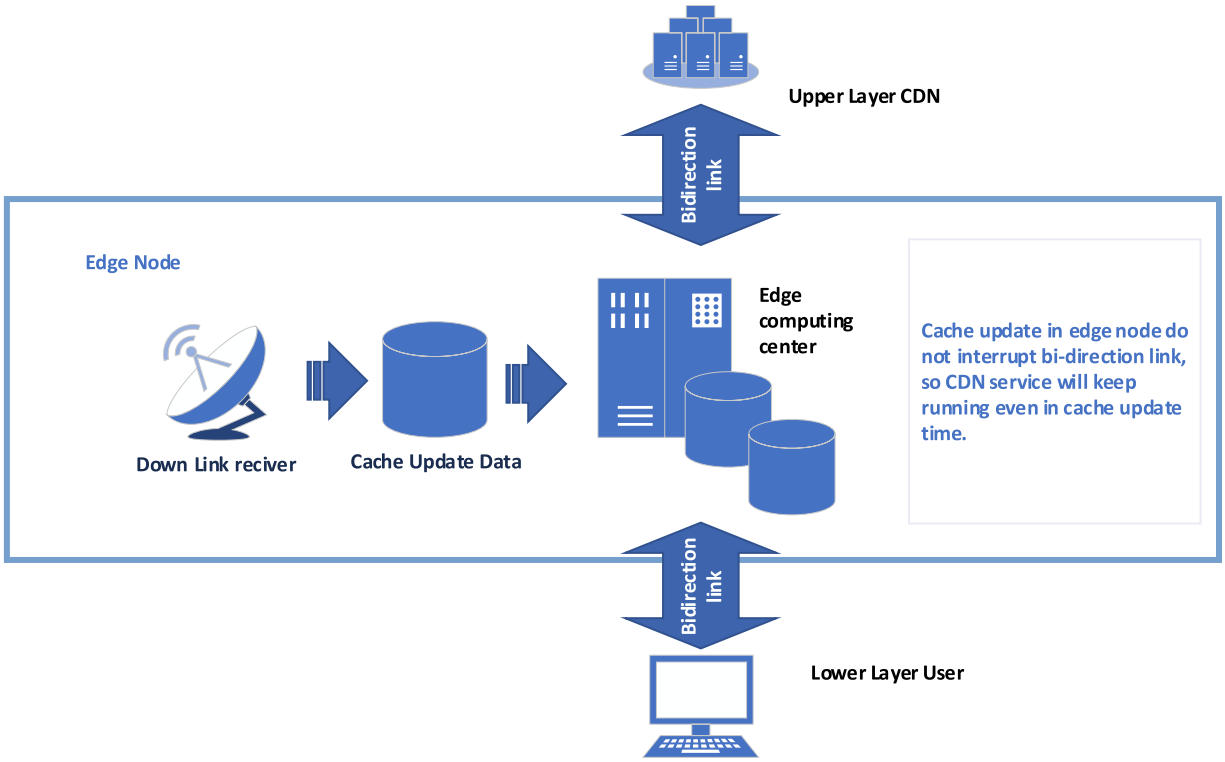

FIGURE 6. CHCDN edge nodes.

to the internet. After the update data is acquired through the broadcast channel,it will be processed and enter the edge computing center,and its architecture is shown in Fig.6. The edge node classifies the obtained broadcast channel data, and directly filters out some contents if the node is not interested in them. If some of the data need to be delivered in real time, the edge node will submit the data to the corresponding application program for further processing. If some of the data is non-real-time data, the edge node will store the data in the hard disk. For a missed user request to an edge node, the node will forward the request to the upper-layer CDN system for processing through a bidirectional channel. Traditional CDN updates rely on bidirectional networks, while CHCDN edge nodes, due to the addition of a broadcast channel receiver,when there is cache data update, update data can be delivered to all edge nodes through the broadcast channel, and simultaneously can communicate with upper-layer CDN and lower-level customers through the bidirectional channel,so CDN services in edge nodes will not be interrupted by cache update.

The CHCDN system model has only two levels, the central node and the edge node. Except for the central node, all other nodes belong to edge nodes in CHCDN system. The hierarchical system refers to the hierarchy in a bidirectional point-to-point network, so all intermediate nodes and edge nodes in the system can receive the content delivered by CHCDN at the same time. For users, they do not know the existence of broadcast channels, so when analyzing CHCDN performance, it should be done in a hierarchical model.

B. PERFORMANCE ANALYSIS OF THE CHCDN SYSTEM

Since hierarchical systems can be defined nested, the hierarchy of the system will have different definitions according to the different problems analyzed, and the following relationship can be obtained by rewriting according to (4):

$$\begin{aligned} R_{w_1} &= (1-\alpha_e)\sum R_e, \\ R_{w_i} &= (1-\alpha_{w_{i-1}})\sum R_{w_{i-1}} \\ s.t.\ e &\geq 1, \quad e \leq w_i \leq L_{int} \end{aligned} \tag{7}$$

In (7), R_{w_1} represents the total number of broadcast data access requests at the w_1 layer and α_e represents the hit rate of the edge convergence layer; $\sum R_e$ represents the sum of the visits of all edge layers under the w_1 layer, and the sum in the subsequent formulas represents the sum of the corresponding layers. Similarly, R_{w_i} represents the total requests of the w_i layer, and α_{w_i} represents the hit rate of the w_i layer.

Equation (5) and (7) are the basis of the layer-to-layer relationship in the hierarchical model, and we can deduce the key parameters related to QoS of each layer:

1) Because the effective data acquisition ratio from the lower layer to the higher layer via WAN is $1-\alpha_{w_{i-1}}$ and α_{w_i} via the inside of the edge system, and the transmission path length of the data transmitted through the broadcast network is only one hop, therefore, based on derivation of (1), the total access path length is $MaxDelay_H$, as expressed by (8):

$$MaxDelay_H = 2\sum(1-\alpha_{w_i})\log_{d_{w_{i+1}}} C_{w_{i+1}}$$

$$+2\alpha_e \log_{d_e} C_e + 1 \quad (8)$$

where $d_{w_{i+1}}$ represents the degree of random tree composed of Cache in the w_{i+1} layer, $C_{w_{i+1}}$ represents the number of Cache nodes in the w_{i+1} layer, d_e represents the degree of Cache random tree in the edge computing center, and C_e represents the number of caches in each edge computing center. d_{w_i} also represents the communication capability of each layer, with little change above the secondary edge layer. It is easy to deploy a large amount of storage at high level. α_{w_i} increases with the increase of the hit rate, so $MaxDelay_H \leq MaxDelay_{H1}$, where $MaxDelay_{H1}$ is:

$$MaxDelay_{H1} = 2(1-\alpha_e)\log_{d_{w_1}} C_w + 2\alpha_e \log_{d_e} C_e + 1$$
$$C_w = \sum(d_{w_i} C_{w_i}) \quad (9)$$

Through system classification, most users' needs are fulfilled at the edge. d_e can be large while the number of C_e is not large, and α_e will approach 1. Because bandwidth and processing capacity limit d_{w_i} is less and C_w quantity is moderate, the maximum delay of the hierarchical system is smaller than that of the non-hierarchical system.

2) According to (2), it can be deduced that under the probability of 1-1/N, the maximum access number InnerMaxLoad of Cache in each data center in the middle layer is:

$$InnerMaxLoad = priLoad + secLoad$$
$$priLoad = \rho_{w_i}(2\log_{d_{w_i}} C_{w_i} + O(\frac{\log N}{\log\log N}))$$
$$secLoad = O(\frac{d_{w_i} q_{w_i} \log N}{\log(\frac{d_{w_i} q_{w_i}}{\rho_{w_i}} \log N)} + \log N) \quad (10)$$

ρ_{w_i} represents the processing capacity of each Cache in the layer, calculated by (5), and q_{w_i} represents the counting threshold of pages to be saved in the w_i layer cache. After layering, InnerMaxLoad is the number of tasks that can be carried out.

3) Since after the classification, the exchange messages between different data centers at the same level will be forwarded through the upper layer, and all missed traffic will be forwarded through the upper level link interface. The upper bandwidth required by each layer WB_{w_i} is:

$$WB_{w_i} = R_{w_i}/t = (1-\alpha_{w_i}) * \sum(R_{w_{i-1}}/t)$$
$$= (1-\alpha_{w_i}) \sum WB_{w_{i-1}} \quad (11)$$
$$\max_{e \leq w_i \leq L_{int}} WB_{w_i} = \sum WB_e \quad (12)$$

In (11), t represents the time length for generating R_{w_i} requests, and WB_e in (12) is the edge layer access bandwidth. Equations (7), (8), (9), (10), (11) shows that the hierarchical structure plays a very good smoothing role in the service burden sharing of the system as a whole, which illustrates the necessity of edge calculation. Equation (11) describes the communication bandwidth that the system must guarantee for each node in the WAN, which belongs to the internal scheduling bandwidth of the system.

4) WAN data traffic. If no broadcast channel is used, all valuable broadcast data will be transmitted from the central node (data not in the central node will also be transmitted to the central node through CDN scheduling), and at least q_{w_i} times will be transmitted through each layer. The amount of data that the system needs to transmit over the WAN is:

$$Q_{w_i} = \sum(q_{w_i} S_{w_i} d_{w_i} C_{w_i}) \quad (13)$$

In (13), represents the storage capacity required in the layer CDN system.

Judging from (11),(12),(13), if the number of edge nodes continue to increase, the data volume will rise rapidly and the inter-layer link quality l_n will drop sharply. According to (3), this will lead to a sharp drop in l_{user}. The CHCDN system can completely free wide area network channels from broadcast data distribution, and the content distribution traffic of wide area network can be reduced from Q_{w_i} described in (13) to 0. When a network hot event occurs, using CHCDN to transmit hotspot data can prevent network paralysis caused by WB_{w_i} from reaching (12). Therefore, the operation of CHCDN system will improve the system transmission efficiency and network QoS at the same time.

IV. EXPERIMENT ON THE CHCDN SYSTEM AND SIMULATION

To verify the characteristics of the CHCDN system, the author developed a verification software called FNCTB (Fusion Net Cache Test Bed)V0.1 using the Python language. This program is based on DES (Discrete Event Simulation) simulation and realizes the transmission simulation of broadcast channel and point-to-point network channel without using the OMNeT++ Library [34], [35]. The architecture is as shown in Fig.7.

The Event STACK module constructs an efficient data structure for DES event processing. Event loop implements the time advance logic of DES simulation. The input simulator can simulate various network user behaviors and form input to the target simulation system. Event STACK, Event loop and Input simulator together form the Event Process Simulator subsystem to realize the bottom-level operation mechanism of DES simulation. BroadcastNIC simulates broadcast channel transmission, P2PNIC simulates bidirectional channel transmission, and ComputerNode simulates intelligent devices on the network with logical processing capability. TaskItem simulates data requests and responses. ComputerNode and TaskItem participate together to implement a callback mechanism for extension of system functions. Toplogical Manager is used to deal with the calculation of graph theory related problems. Configure File Manager makes the definition of simulation target system flexible and

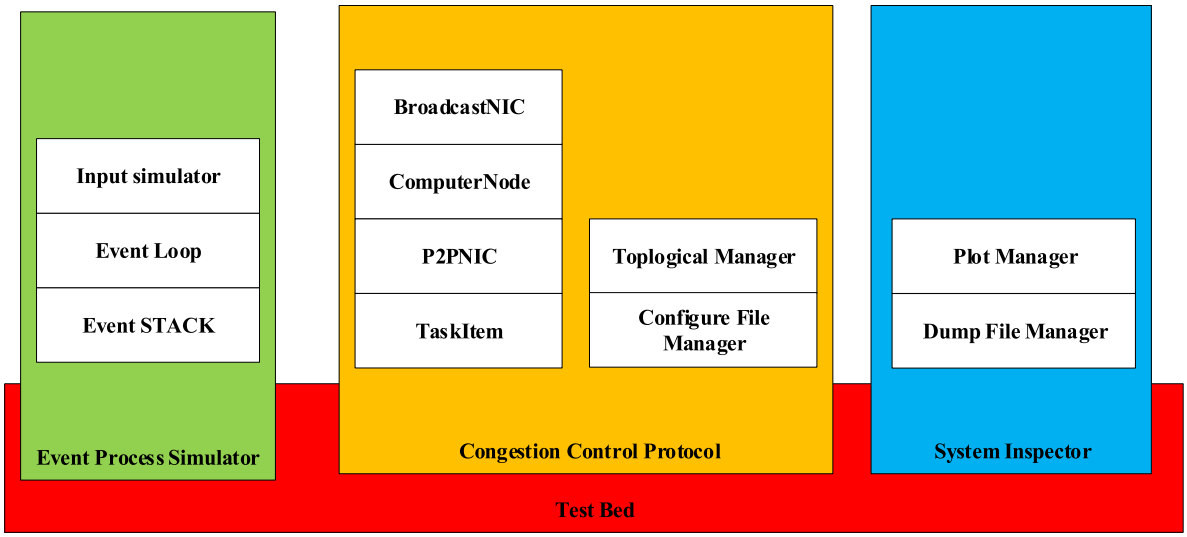

FIGURE 7. FNCTB V0.1 software system architecture.

efficient. BroadcastNIC, P2PNIC, ComputerNode, TaskItem, Toplogical Manager and Configure File Manager together have implemented the Congestion Control Protocol subsystem. The Congestion Control Protocol subsystem is the mechanism for running the business logic simulated by CHCDN. Dump File Manager makes system debugging easy. Plot Manager can analyze dump data and generate reports. Together they form the System Inspector subsystem. The three subsystems of Event Process Simulator, Congestion Control Protocol and System Inspector constitute a fully functional test bed for testing CHCDN problems.

In FNCTB V0.1, network nodes can be configured with different memory sizes, CPU processing capacities, interface bandwidths and disk storage capacities, and implement the CDN system operation infrastructure such as network flow control algorithm and CDN cache replacement algorithm through the callback mechanism. The network topology can be customized. Since CDN is an overlay network [36], [37], the verification software directly realizes the transmission and flow control of data packets between nodes without simulating the details of low-level networks. The software can collect data of system operation according to needs. We use FNCTB V0.1 to simulate and analyze the behaviors of hierarchical CDN network and CHCDN network. The time precision used in simulation is 10^{-5}s, and the balance between simulation precision and simulation time is achieved by controlling input granularity. FNCTB V0.1 currently uses only a single thread and runs the simulations related to this section on a PC with AMD A10-5800K CPU and Ubuntu 18.04 OS, each running for about 1.5 hours.

A. HIERARCHICAL CDN NETWORK SIMULATION

The systematic network topology used in this simulation is shown in Fig.8:

TABLE 1. Network node hierarchy.

Layer	NodeName	Number of Nodes
Layer4	A	1
Layer3	B,C	2
Layer2	D,E,F,G,H,I,J,K	8
Layer1	L1-L4,M1-M4,N1-N4,O1-O4, P1-P4,Q1-Q4,R1-R4,S1-S4	32

TABLE 2. Network bandwidth settings.

Layer	Uplink Bandwidth	Downlink Bandwidth
Between Layers3-4	30000KB/S	30000KB/S
Between Layers2-3	12000KB/S	12000KB/S
Between Layers1-2	3000KB/S	3000KB/S

A represents the top-level node, which also means the Internet, all content requests will be satisfied in Node A.

B–K are cache nodes, B, C are high-level cache nodes, and D–K are low-level secondary edge cache nodes.

Ln-Sn, n = 1, 2, 3, 4 are edge nodes, simulating user traffic convergence point.

The content replacement algorithm of CDN in B–K adopts the LRU (least recently used) replacement strategy, with replacement threshold q = 2. If we choose other cache algorithm for simulation, the converge time for simulations may be different, but the conclusion is same. The processing capacity of all nodes in the experimental setup is large enough to not cause performance bottleneck. Therefore, the simulation results depend entirely on the network-related characteristics.

Data parameters of each part are shown in Table.1,Table.2, and Table.3.

The content data requested by the simulated user is divided into 20 data sets of equal size, each request of each edge node

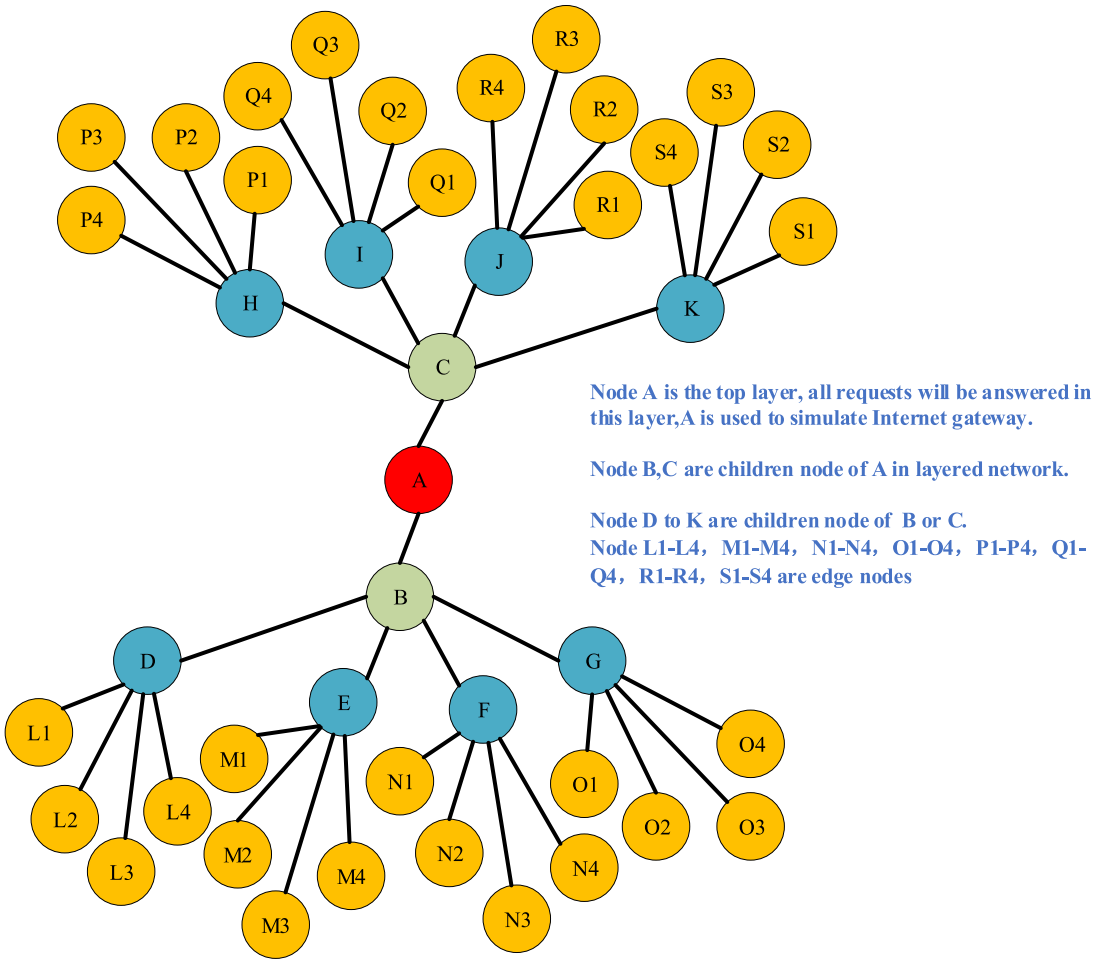

FIGURE 8. Topology of the simulation of the hierarchical CDN system.

TABLE 3. Size and bandwidth for data packets.

Type	Min Transmission Bandwith	Size
Uplink	600KB/S	2000KB
Downlink	600KB/S	2000KB

TABLE 4. Frequency of experimental data packets.

Packet No.	Frequency	Packet No.	Frequency
1	0.2123	11	0.0312
2	0.1220	12	0.0298
3	0.0882	13	0.0273
4	0.0700	14	0.0257
5	0.0586	15	0.0243
6	0.0506	16	0.0231
7	0.0448	17	0.0220
8	0.0402	18	0.0210
9	0.0366	19	0.0210
10	0.0366	20	0.0193

is randomly generated, and its request probability distribution follows ZIPF distribution with $Z = 0.8$, and the data distribution is shown in Fig.9.Details are shown in Table 4.

The size of each packet is 2,000KB, and the minimum transmission bandwidth of each packet is 600KB/S. In the system simulation, each data packet allocates the bandwidth on the transmission path according to the minimum transmission bandwidth.

B–K have no content cached at the beginning of the simulation. B, C set the cache space to contain 11 data packets, D–K set the cache space for 5 data packets. When Cache receives a content request that is not cached, it must forward the request to the network interface of the upper node, and then the upper node returns the content.

During system simulation, edge nodes request and receive content at full load at line speed. To achieve good simulation accuracy and make the simulation calculation load bearable, the request for each edge node is input in batches of 10 packets. Request packets are continuously injected into edge nodes Ln-Sn, $n = 1, 2, 3, 4$ from 0-300 s. The simulation will be completed in 320 s.

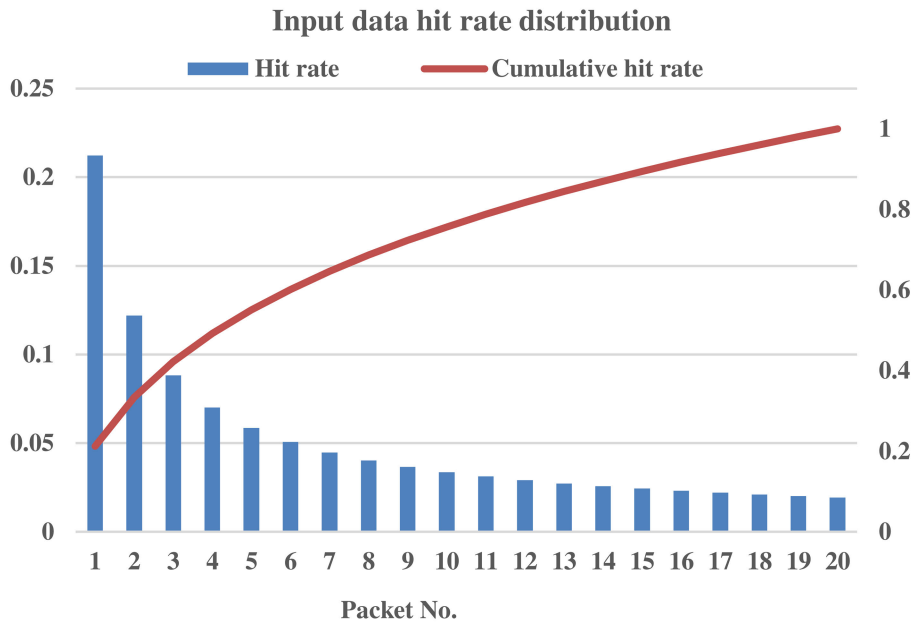

FIGURE 9. Hit rate distribution of request data.

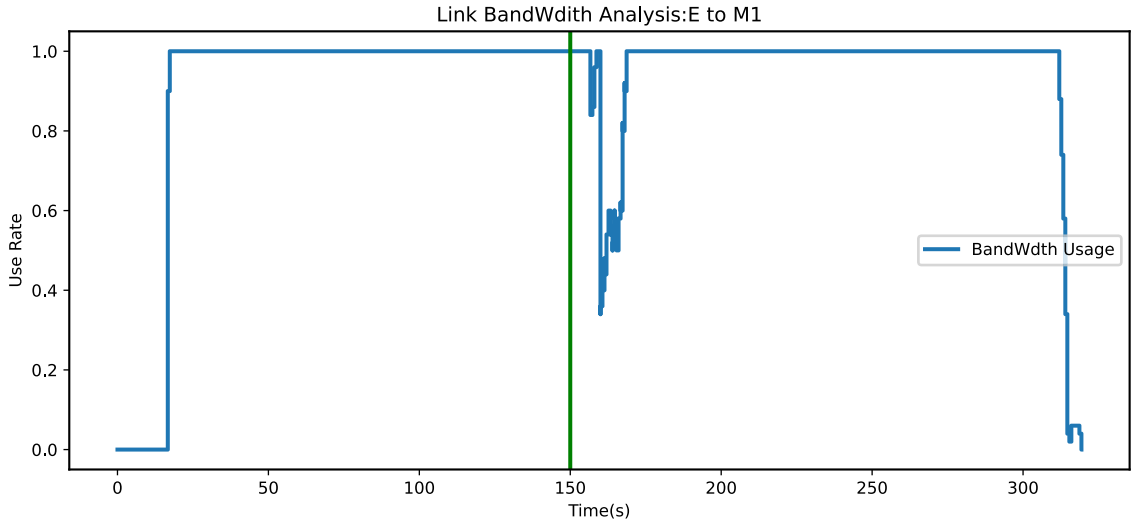

FIGURE 10. Hierarchical CDN updating E-M1 bandwidth utilization rate.

We selected three links, i.e. E-M1, B-E, and A-B to observe the system bandwidth changes, as shown in Figs.10-Fig.12.

To make the update changes more obvious and easier to observe, the content update operation updated the 5 cache contents of the edge cache node, and the updated content's hit rate was still the top 5 hit rate. After the content is updated, the transmission of request and response data for expired content in the system is cleared so as to observe the impact of content update on bandwidth at all levels. The sudden sag in bandwidth utilization in Fig.10 and Fig.11 is to eliminate the influence of residual data packets.

As can be seen from Fig.10, the B-E link has a stable incoming traffic at about 100s, and the A-B link has a stable incoming traffic at about 150s in Fig.12. The traffic drop after the first peak of the A-B link is another round of B's cache content update triggered after the completion of D–G cache update. After the new balance is reached, the content stored in D–G is not stored in B. The bandwidth utilization rate after entering the stable working phase is calculated by (11) as B-E link: 0.449 and A-B link: 0.132 respectively, which are marked with red lines. The simulation content update time is 150s, which is marked with a green line in the figures.

By comparing Fig.11 with Fig.12, we can clearly see that the content update takes up the network bandwidth and has an amplification effect in high-level networks. The amplification factor is equal to the fan-out ratio of nodes, and its peak value

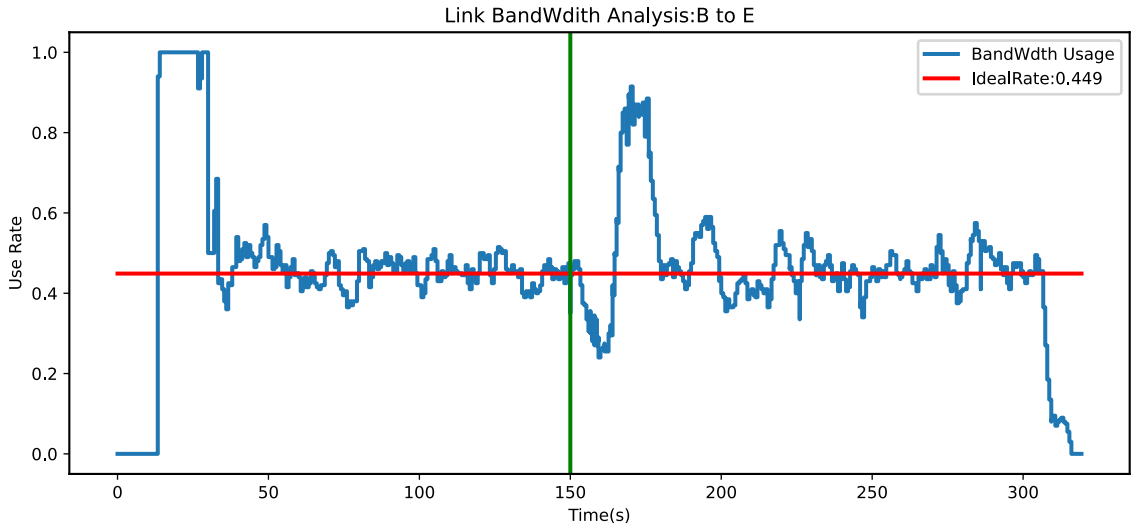

FIGURE 11. Hierarchical CDN updating B-E bandwidth utilization rate.

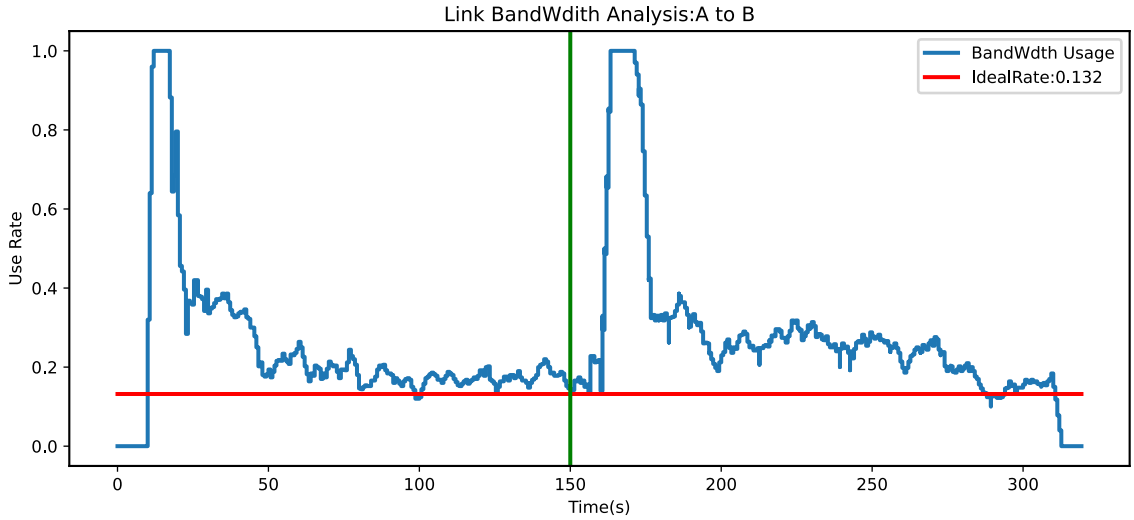

FIGURE 12. Hierarchical CDN updating A-B bandwidth utilization rate.

is calculated as 48000 KB/s by (12). Experiments also show that although cache mechanism is eased, A-B link can still be paralyzed by this content update alone. If in a real system, due to the huge fan-out ratio of high-level cache, small changes in low-level cache (e.g., cache content expires) will also cause huge fluctuations in high-level bandwidth. Moreover, due to the content update of the lower layer cache, the content update of the higher layer cache will be triggered again, and the update bandwidth convergence speed of the higher layer network is slower than that of the lower layer network. If there are sudden and concentrated content requests that cannot update the cache content in time, or do not have time to update the cache, this situation will be disastrous for a hierarchical CDN network, no matter whether or not the cache content is updated through the existing network, the network will be paralyzed immediately.

B. CHCDN NETWORK SIMULATION

The parameters of this simulation were mostly the same as the parameters of the previous layered CDN network simulation, except that the broadcast channel was selected for content update. The content replacement algorithm of CDN in B–K adopts the LRU (least recently used) replacement strategy, with replacement threshold $q = 2$, same as previous layered CDN network simulation. The peer-to-peer network topology is the same as that in Fig.8. On this basis, all cache nodes are configured with broadcast data receivers. After receiving the updated data, the nodes decide what to update according to their own conditions. The processing after data update is the same as that in hierarchical network, and the request and response data transmission of expired content in the system will be cleared. The broadcast channel adopts a transmission queue mechanism, and each package of the

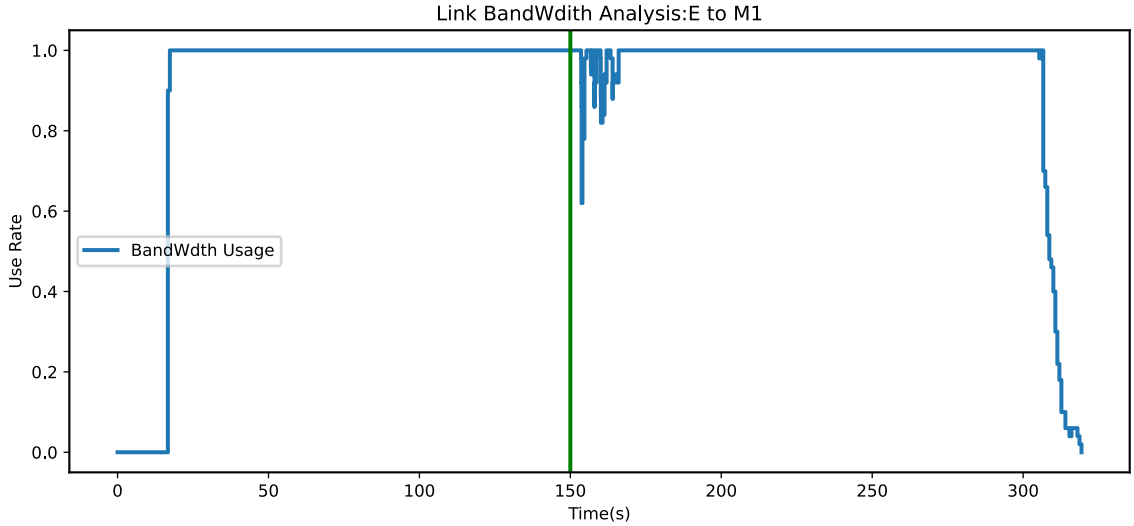

FIGURE 13. CHCDN updating E-M1 bandwidth utilization rate.

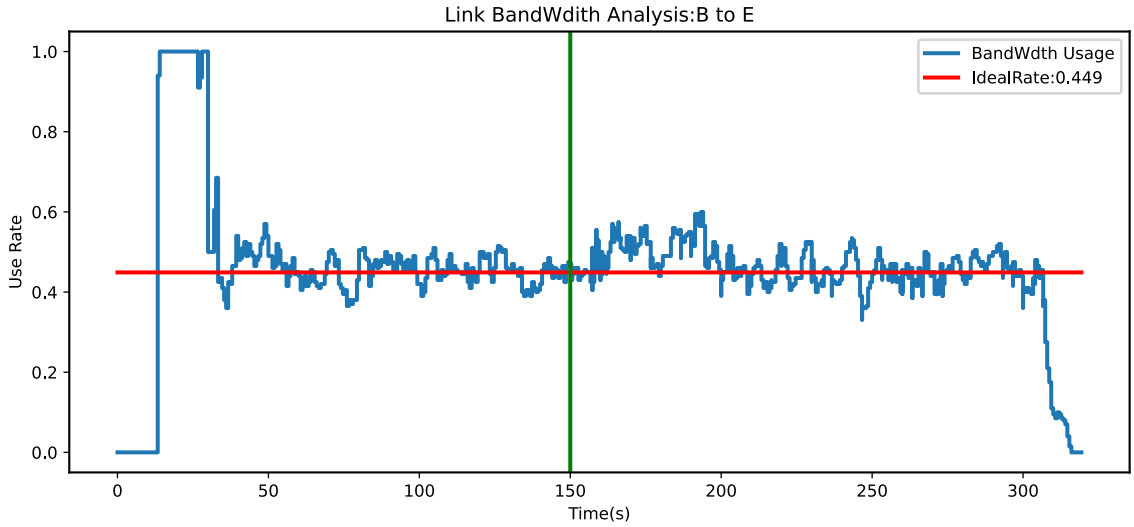

FIGURE 14. CHCDN updating B-E bandwidth utilization rate.

TABLE 5. Broadcast channel simulation parameters.

Number of Channels	Channel Bandwidth	Bandwidth per Task
1	3000KB/S	3000KB/S

content transmission will monopolize the channel. Because another random input process is used, the curves of the first 150 s of the CHCDN update experiment are not exactly the same as those in Fig.10- Fig.12.

The broadcast channel and corresponding data transmission parameters are shown in Table 5:

Similarly, the insertion and transmission of broadcast content started at time 150 s.

We also selected three links, i.e. E-M1, B-E, and A-B to observe the system bandwidth changes, as shown in Fig.13-Fig.15:

By comparing Fig.11 with Fig.14 and Fig.12 with Fig.15, we can see that updating the cache content of the CDN through the broadcast channel has little fluctuation on the bandwidth utilization. At the same time, it is also simulated that if there is a sudden and concentrated content request and there is no content update to cache in the early stage, the cache update can be implemented in real time by using CHCDN, thus the network QoS will not be affected by the sudden request. Therefore, the simulation experiment proves the huge advantages of CHCDN in improving network QoS. In addition, in terms of transmission efficiency, updating and using CHCDN only transmitted data once, which is also a huge advantage of CHCDN.

C. LIVE VIDEO STREAM DATA DISTRIBITION

To distribute live video stream data that that large scale users watched without interaction, currently the CDN sys-

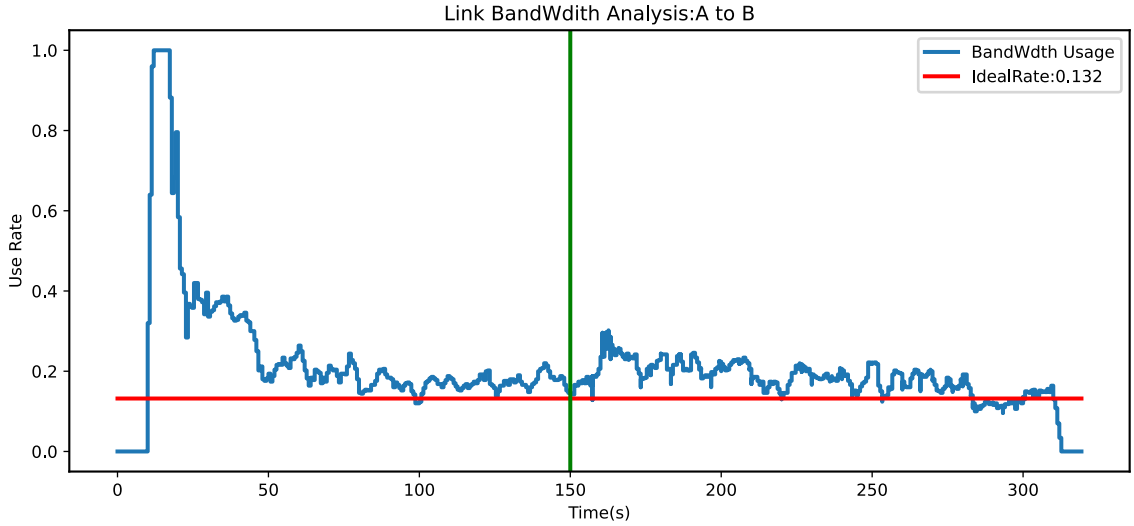

FIGURE 15. CHCDN updating A-B bandwidth utilization rate.

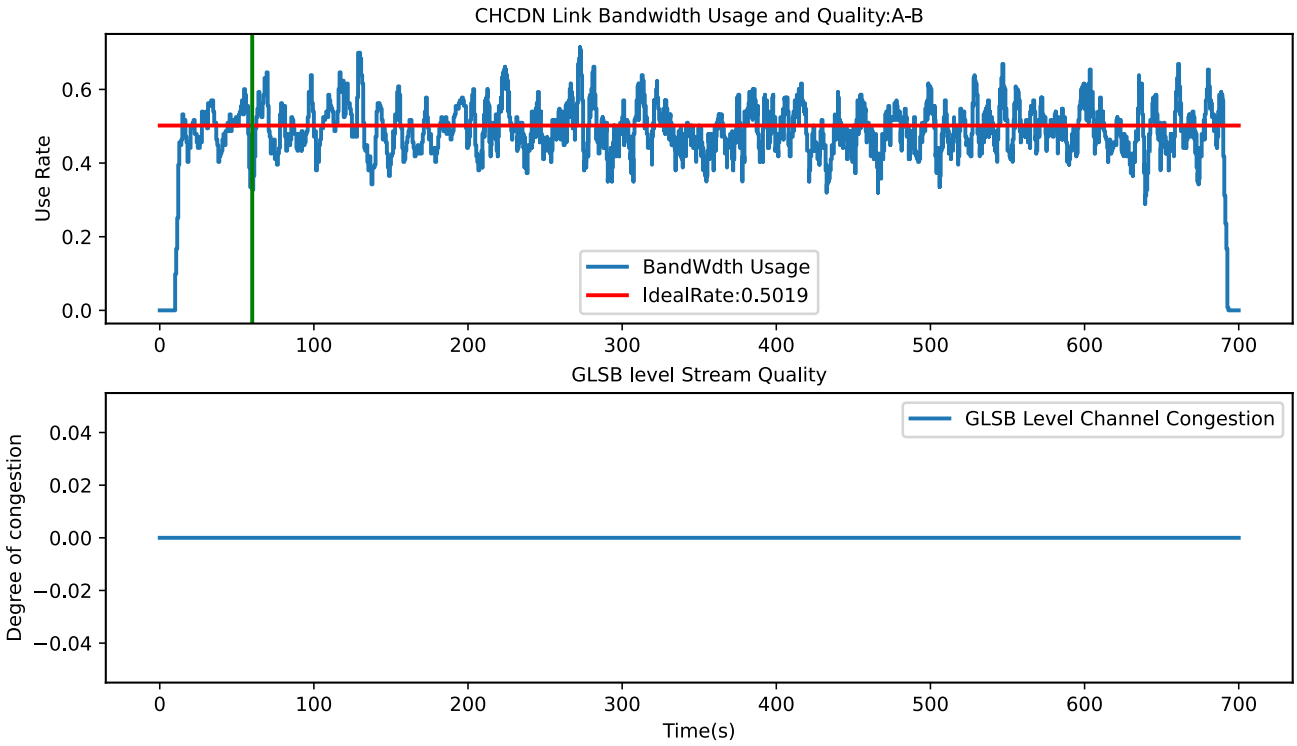

FIGURE 16. CHCDN link bandwidth usage and congestion: A-B.

tem use multi layers structure to distribute the live stream to edge cache servers, then the user access the live stream from proxy servers through edges cache servers. The system structure is similar to Fig.8, the different is that the height of the tree is much higer than Fig.8 and those higher level nodes are not fully functional CDN nodes,they just transfer live streams; only two layers,the LSB layer and GLSB Layer, are fully functional CDN nodes. So we simulate those system using Fig.8 's network topology, but assign different meaning for different nodes as in Table.6.

Now we allocate fixed bandwidth for live video stream data and other contents for each level of network as in Table.7 and Table.8. In Table.8, we do not need to allocate live video bandwidth for Source-GLSB level and GLSB-LSB level unicast channel,since we use broadcast channel for data transmit.

Since all contents in traditional CDN system use same network to transmit,even live video stream already allocate

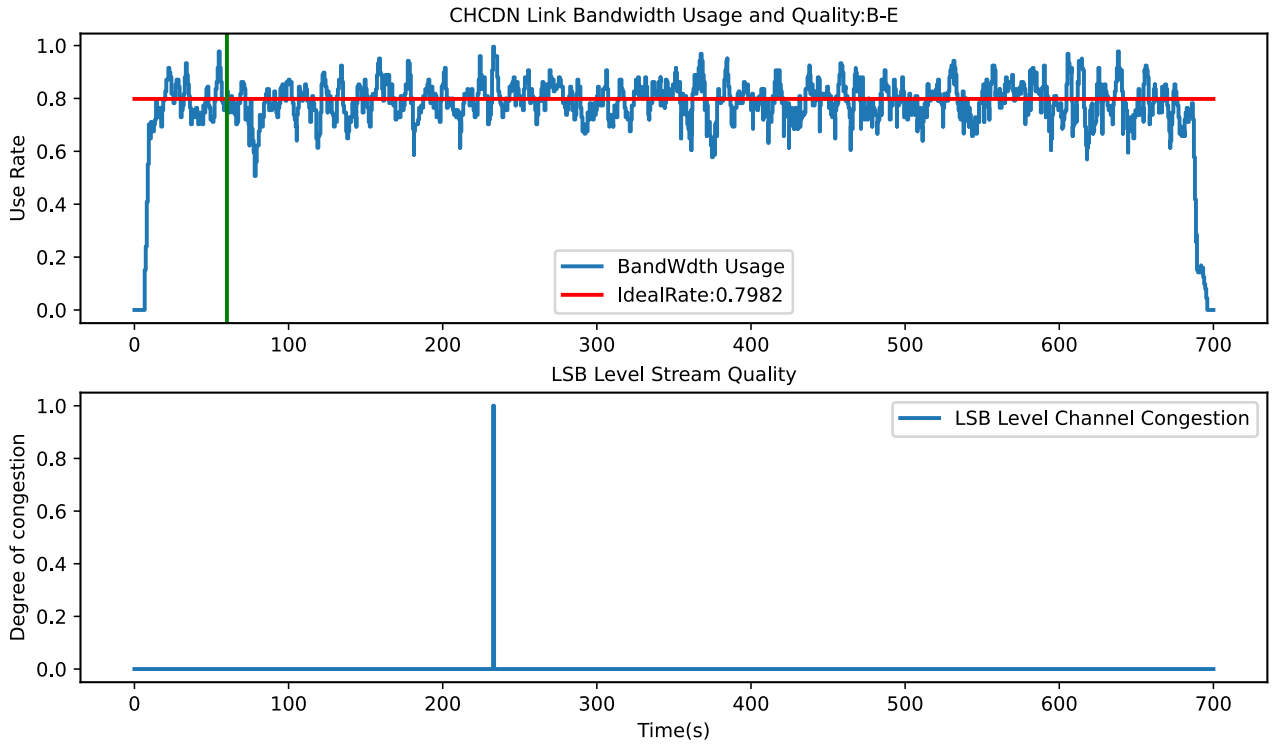

FIGURE 17. CHCDN link bandwidth usage and congestion: B-E.

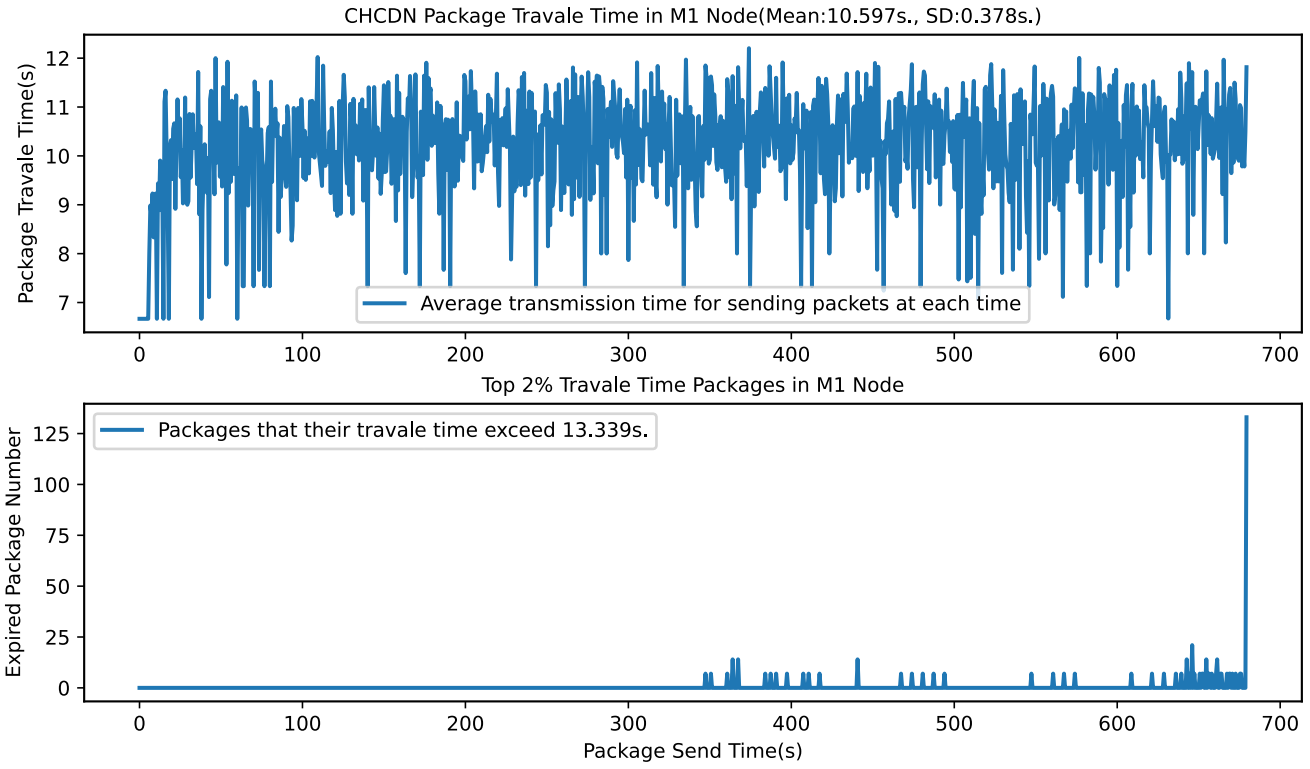

FIGURE 18. CHCDN packages travel time in M1.

bandwidth before, but if the network congestion occurs, the live video stream quality will be affected. As state of art CDN system implemented by Alibaba for 2018 FIFA live video stream, when congestion occurs the live video stream

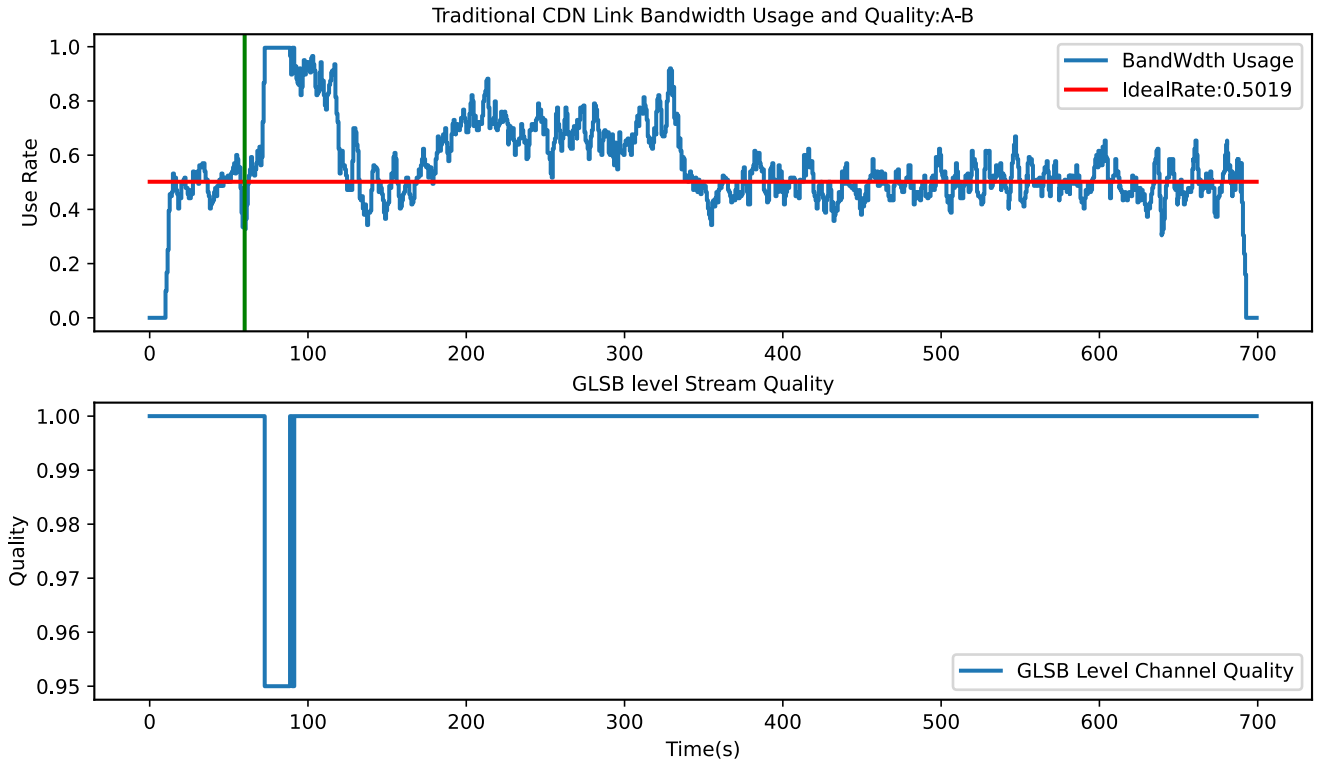

FIGURE 19. Traditional CDN link bandwidth usage and quality: A-B.

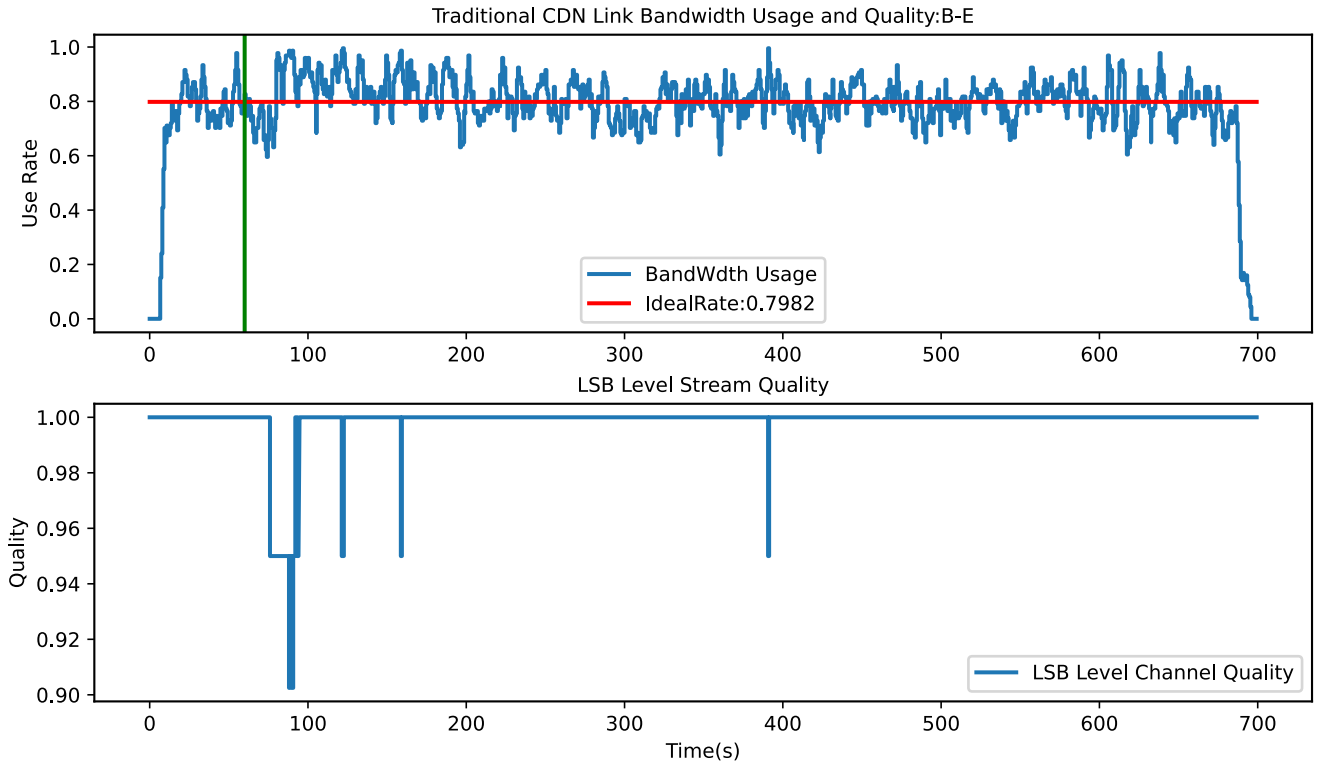

FIGURE 20. Traditional CDN link bandwidth usage and quality: B-E.

quality will drop to 0.95 of previous quality on each links, But for live video steam, people will obviously feel discomfort if the video quality is below 0.95. So for our simulation we define congestion condition as the usage of bandwidth of

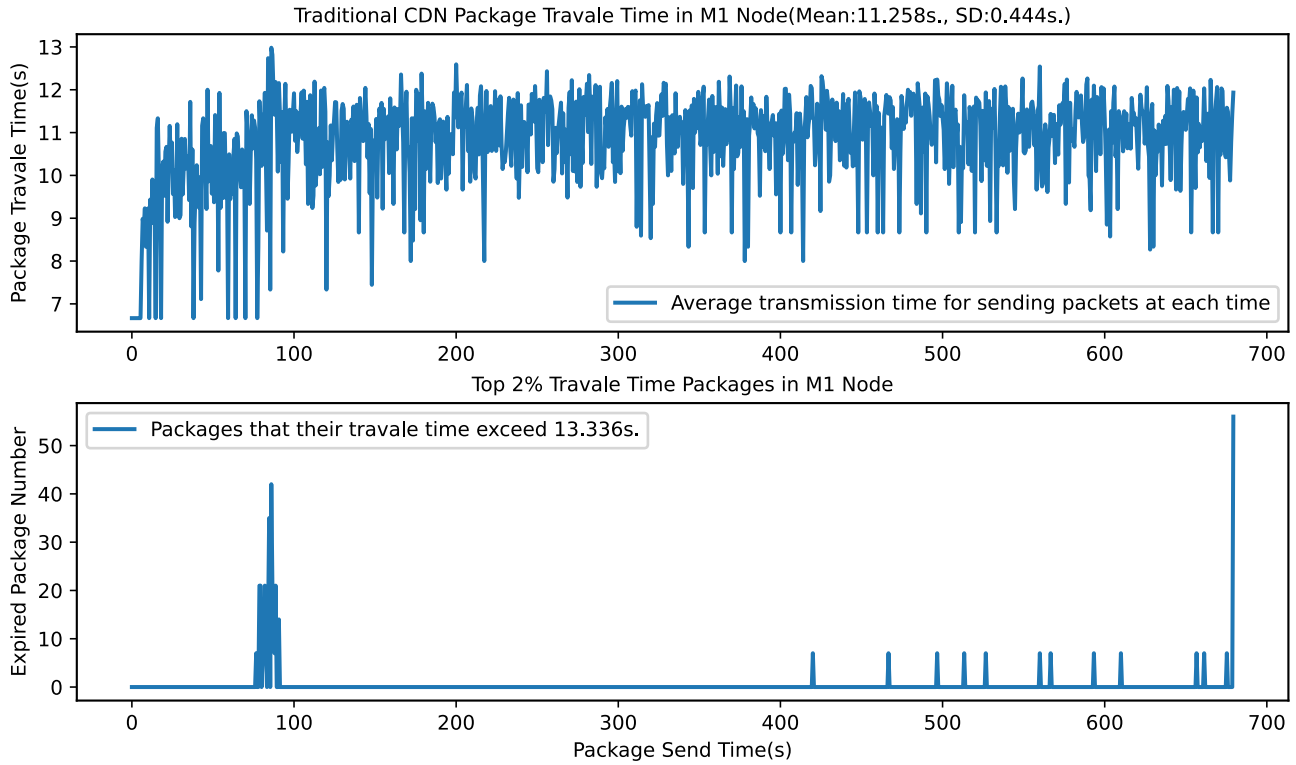

FIGURE 21. Traditional CDN packages travel time in M1.

TABLE 6. Network node hierarchy.

Layer	NodeName	Function for live video and other contents
Source	A	Source Node
GLSB	B,C	GLSB Nodes
LSB	D,E,F,G,H,I,J,K	LSB Nodes
Proxy	L1-L4,M1-M4, N1-N4,O1-O4, P1-P4,Q1-Q4, R1-R4,S1-S4	Proxy Nodes

TABLE 7. Traditional CDN network bandwidth setting.

Layer	For live video	For other contents
Source-GLSB	22100KB/s	78900KB/s
GLSB-LSB	52500KB/s	67500KB/s
LSB-Proxy	52500KB/s	3000KB/s

TABLE 8. CHCDN network bandwidth setting.

Layer	For live video	For other contents
Source-GLSB	0 KB/s	78900KB/s
GLSB-LSB	0 KB/s	67500KB/s
LSB-Proxy	52500KB/s	3000KB/s
BroadCast	25000KB/s	3000KB/s

other(not live video) contents above 0.985, and if congestion happened the live video stream quality will degrade to 0.95 of previous link quality. And the end user's experienced live video stream quality is accumulation quality of the whole transmit path.

We simulate network transmission begin with the stable cache allocation on each nodes. B–K node's cache space is same as previous simulation,the cache replace policy is same as previous simulation. And for other content's requests(not live video contents), the request data distribution same as Table3 and Table4. At simulation time 60s, we make one package,the No.2 package update. And we can get the traditional CDN system simulation data as Fig.19–Fig.21,and the CHCDN system simulation data as Fig.16–Fig.18.

As the simulation shows in Fig.19 and Fig.20, using traditional CDN system to deliver live video stream,the video quality is vulnerable to network congestion. One package update will annoy the user's live video watching experience from 72.667s to 391.333s. In CHCDN system, the same update affect almost nothing, in Fig.17 at time 232.667s, the network congestion occured, but this do not affect live video watching experience of users, the CHCDN system use broadcast channel to transmit live video data, so the CHCDN system's live video quality is always 1. We do not show CHCDN system's live video quality in simulate result, we just show the ''Degree of congestion''. For the other contents(not live video contents) transmit quality compare, we can use Fig.18 and Fig.21, we notice that traditional CDN system

TABLE 9. Other contents transmit quality compare.

Performance index	Traditional CDN	CHCDN
Average package travel time	11.258s	10.597s
Package travel time standard deviation	0.444s	0.378s
Travel time threshold of top two percent	13.336s	13.339s

have longer average package travel time and larger package travel time SD(Standard Deviation) than CHCDN system. To compare the worst experience, we use threshold of top two percent travel time, the simulation result show that CHCDN system is a little bit higher but no big difference in this performance index. We use Table.9 to summarize not live video contents transmit quality compare.

If a bidirectional channel communication is used to simulate broadcast channel communication for real-time content update, the effect of CHCDN update can be achieved to a certain extent. However, there are several drawbacks in doing so.

1) For each content update path, the bidirectional channel must be used to simulate the broadcast channel.
2) Both ends of each content-update bidirectional channel should control transmission and receipt all the time.
3) Even if a lot of resources are invested to realize the above two points, due to the node processing capability and buffers management variance on the path, the bidirectional channel transmission cannot achieve a network QoS that is the same as that of CHCDN transmission.

As there are no economical and effective bidirectional update channels, as mentioned in previous studies [2]–[11], [15]–[21], [24]–[26], CDN can now only optimize the use of other resources in the system for QoS enhancement, yet CHCDN is the fundamental way to solve this problem.

V. CONCLUSION

This paper has offered an exhaustive analysis of the bandwidth problem in CDN systems, especially for the real-time update of cache, proposes a layered CDN model that maintains QoS. To work smoothly, the layered CDN model needs to ensure the bandwidth between CDN systems. If there is content that needs to be updated in real time, bandwidth between hierarchical CDN systems will encounter bottlenecks. This paper puts forward a systematic solution of CHCDN to this problem and makes theoretical analysis. Finally, the performance differences of the hierarchical CDN model and the CHCDN model in cache content updating are compared by system simulation. The results of data simulation show the advantages of the CHCDN architecture in distributed computing environment.

To effectively apply the CHCDN distributed system architecture, it is necessary to change the way of distributed software development, which is the difficulty and opportunity to popularize the CHCDN system. Therefore, in future studies, we will look for an advantageous application area of the CHCDN architecture, and take this as a breakthrough to promote the development and application of this new architecture.

REFERENCES

[1] Cisco, "Cisco visual networking endex: Global mobile data traffic forecast update, 2017–2022," Cisco, San Francisco, White Paper, Feb. 2019. [Online]. Available: https://hollandfintech.com/wp-content/uploads/2019/02/white-paper-c11-738429.pdf

[2] M. Dehghan, B. Jiang, A. Seetharam, T. He, T. Salonidis, J. Kurose, D. Towsley, and R. Sitaraman, "On the complexity of optimal request routing and content caching in heterogeneous cache networks," *IEEE/ACM Trans. Netw.*, vol. 25, no. 3, pp. 1635–1648, Jun. 2017.

[3] T. Bektaş, J.-F. Cordeau, E. Erkut, and G. Laporte, "Exact algorithms for the joint object placement and request routing problem in content distribution networks," *Comput. Oper. Res.*, vol. 35, no. 12, pp. 3860–3884, Dec. 2008.

[4] W. M. Aioffi, G. R. Mateus, J. M. Almeida, and R. C. Melo, "Dynamic content placement for mobile content distribution networks," in *Web Content Caching and Distribution. WCW* (Lecture Notes in Computer Science), vol. 3293, C. H. Chi, M. van Steen, and C. Wills, Eds. Berlin, Germany: Springer-Verlag, 2004, doi: 10.1007/978-3-540-30471-5_2.

[5] X. Tang and J. Xu, "QoS-aware replica placement for content distribution," *IEEE Trans. Parallel Distrib. Syst.*, vol. 16, no. 10, pp. 921–932, Oct. 2005.

[6] R. Kolisch and A. Dahlmann, "The dynamic replica placement problem with service levels in content delivery networks: A model and a simulated annealing heuristic," *OR Spectr.*, vol. 37, no. 1, pp. 217–242, Jan. 2015.

[7] H. Z. Fatin, S. Jamali, and G. Z. Fatin, "Data replication in large scale content delivery networks: A genetic algorithm approach," *J. Circuits, Syst. Comput.*, vol. 27, no. 12, Nov. 2018, Art. no. 1850189.

[8] A. O. Al-Abbasi, V. Aggarwal, and M.-R. Ra, "Multi-tier caching analysis in CDN-based Over-the-Top video streaming systems," *IEEE/ACM Trans. Netw.*, vol. 27, no. 2, pp. 835–847, Apr. 2019.

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

[10] F. Wang, J. Liu, M. Chen, and H. Wang, "Migration towards cloud-assisted live media streaming," *IEEE/ACM Trans. Netw.*, vol. 24, no. 1, pp. 272–282, Feb. 2016.

[11] M. Mangili, F. Martignon, and A. Capone, "Stochastic planning for content delivery: Unveiling the benefits of network functions virtualization," in *Proc. IEEE 22nd Int. Conf. Netw. Protocols*, Raleigh, NC, USA, Oct. 2014, pp. 344–349.

[12] L. Guo, H. Shen, and W. Zhu, "Efficient approximation algorithms for multi-antennae largest weight data retrieval," *IEEE Trans. Mobile Comput.*, vol. 16, no. 12, pp. 3320–3333, Dec. 2017.

[13] Z. Shen, P. P. C. Lee, J. Shu, and W. Guo, "Encoding-aware data placement for efficient degraded reads in XOR-coded storage systems: Algorithms and evaluation," *IEEE Trans. Parallel Distrib. Syst.*, vol. 29, no. 12, pp. 2757–2770, Dec. 2018.

[14] B. Lin, W. Guo, N. Xiong, G. Chen, A. V. Vasilakos, and H. Zhang, "A pretreatment workflow scheduling approach for big data applications in multicloud environments," *IEEE Trans. Netw. Service Manage.*, vol. 13, no. 3, pp. 581–594, Sep. 2016.

[15] Q. Zhang, Q. Zhu, M. F. Zhani, R. Boutaba, and J. L. Hellerstein, "Dynamic service placement in geographically distributed clouds," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 12, pp. 762–772, Dec. 2013.

[16] C. Huang, A. Wang, and J. Li, "Understanding hybrid CDN-P2P: why limelight needs its own Red Swoosh," in *Proc. 18th Int. Workshop Netw. Operating Syst. Support Digital Audio Video*, Braunschweig, Germany, 2008, pp. 75–80.

[17] H. Cheng, N. Xiong, L. T. Yang, and Y. S. Jeong, "Distributed scheduling algorithms for channel access in TDMA wireless mesh networks," *J. Supercomput.*, vol. 63, no. 2, pp. 407–430, 2013.

[18] L. Guo and H. Shen, "Efficient approximation algorithms for the bounded flexible scheduling problem in clouds," *IEEE Trans. Parallel Distrib. Syst.*, vol. 28, no. 12, pp. 3511–3520, Dec. 2017.
[19] R. Skehill, M. Barry, W. Kent, M. O'Callaghan, N. Gawley, and S. Mcgrath, "The common RRM approach to admission control for converged heterogeneous wireless networks," *IEEE Wireless Commun.*, vol. 14, no. 2, pp. 48–56, Apr. 2007.
[20] I. Leyva-Mayorga, R. Torre, V. Pla, and S. Pandi, "Network-coded cooperation and multi-connectivity for massive content delivery," *IEEE Access*, vol. 8, pp. 15656–15672, 2020.
[21] T. He, H. Khamfroush, S. Wang, T. La Porta, and S. Stein, "It's hard to share: Joint service placement and request scheduling in edge clouds with sharable and non-sharable resources," in *Proc. IEEE 38th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Venue Vianna, Austria, Jul. 2018, pp. 365–375.
[22] J.-L. Guo, Y.-Q. Chen, H.-D. Yang, C.-M. Chen, Y.-C. Chen, H. Zhang, and Z. Zhang, "Study on secrecy capacity of wireless sensor networks in Internet of things based on the Amplify-and-Forward compressed sensing scheme," *IEEE Access*, vol. 7, pp. 185580–185589, 2019.
[23] H. Cheng, Z. Su, N. Xiong, and Y. Xiao, "Energy-efficient node scheduling algorithms for wireless sensor networks using Markov random field model," *Inf. Sci.*, vol. 329, pp. 461–477, Feb. 2016.
[24] Y. Cui, F. Lai, E. Yeh, and R. Liu, "Enhanced VIP algorithms for forwarding, caching, and congestion control in named data networks," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Washington, DC, USA, Dec. 2016, pp. 1–7.
[25] M. Dehghan, A. Seetharam, B. Jiang, T. He, T. Salonidis, J. Kurose, D. Towsley, and R. Sitaraman, "On the complexity of optimal routing and content caching in heterogeneous networks," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Anaheim, CA, USA, Apr. 2015, pp. 936–944.
[26] E. Yeh, T. Ho, Y. Cui, R. Liu, M. Burd, and D. Leong, "Forwarding, caching and congestion control in named data networks," 2013, *arXiv:1310.5569*. [Online]. Available: http://arxiv.org/abs/1310.5569
[27] D. R. Karger, A. Sherman, and A. Berkheimer, "Web caching with consistent hashing," in *Proc. Austral. World Wide Web Conf.*, vol. 31, no. 11. Ballina, NSW, Australia, 1999, pp. 1203–1213.
[28] D. R. Karger, E. Lehman, and T. Leighton, "Consistent hashing and random trees: Distributed caching protocols for relieving hot spots on the world wide Web," in *Proc. 29th Annu. ACM Symp. Theory Comput.*, Ramat-Gan, Israel, 1997, pp. 654–663.
[29] Y. Q. Wen, "Methods, devices and architectures for data request and sending in the CDN system," China Patent 10 271 393, Aug. 27, 2014.
[30] Z. Yu, Z. Yu, and Y. Chen, "Multi-hop mobility prediction," *Mobile Netw. Appl.*, vol. 21, no. 2, pp. 367–374, Apr. 2016.
[31] D. Yang, X. Liao, H. Shen, X. Cheng, and G. Chen, "Relative influence maximization in competitive social networks," *Sci. China Inf. Sci.*, vol. 60, no. 10, Oct. 2017.
[32] X. Zheng, D. An, X. Chen, and W. Guo, "Interest prediction in social networks based on Markov chain modeling on clustered users," *Concurrency Comput., Pract. Exper.*, vol. 28, no. 14, pp. 3895–3909, Sep. 2016.
[33] K. Guo, W. Guo, Y. Chen, Q. Qiu, and Q. Zhang, "Community discovery by propagating local and global information based on the MapReduce model," *Inf. Sci.*, vol. 323, pp. 73–93, Dec. 2015.
[34] K. Stamos, G. Pallis, and A. Vakali, "CDNsim: A simulation tool for content distribution networks," *ACM Trans. Model. Comput. Simul.*, vol.20, no. 2, p. 10, 2010.
[35] P. Wunner, S. May, S. May, and S. Dengler, "Development and testing of automotive Ethernet-networks together in one tool–OMNeT++," 2014, *arXiv:1409.1026*. [Online]. Available: http://arxiv.org/abs/1409.1026
[36] D. Clark, B. Lehr, and S. Bauer, "Overlay networks and the future of the Internet," *Commun. Strategies*, vol. 63, pp. 109–129, Jul. 2006.
[37] T. Vinh Nguyen, F. Safaei, P. Boustead, and C. Tung Chou, "Provisioning overlay distribution networks," *Comput. Netw.*, vol. 49, no. 1, pp. 103–118, Sep. 2005.

YUQIANG WEN received the M.S. degree from the Huazhong University of Science and Technology. He is currently a Lecturer with the Department of Information Engineering, Guangzhou Nanyang Polytechnic College. His current research interests include distributed computing, network theory, and big data.

YUQIANG CHEN received the Ph.D. degree from the Guangdong University of Technology. He is currently a Professor with the Department of Computer Engineering, Dongguan Polytechnic College. His current research interests include network security, mobile Internet, big data, the Internet of Things (IoT), and cryptography.

MENG-LIANG SHAO received the M.S. degree from the Nanjing University of Software Engineering. He is currently an Associate Professor with the Department of Computer Science, South China Institute of Software Engineering GU. His current research interests include big data, distributed computing, and mobile Internet.

JIAN-LAN GUO is currently an Associate Professor with the Department of Computer Engineering, Dongguan Polytechnic College. Her current research interests include network security, mobile Internet, the Internet of Things (IoT), and cryptography.

JIA LIU received the bachelor's degree from Yangtze University. He is currently a Senior Engineer with the Department of mechatronics and automobile, Guangzhou Nanyang Polytechnic. His current research interests include electrical automation, intelligent control, and the Internet of Things (IoT).