Received August 19, 2017, accepted September 11, 2017, date of publication September 19, 2017,
date of current version February 1, 2018.

Digital Object Identifier 10.1109/ACCESS.2017.2754419

A Survey on Content Placement Algorithms for Cloud-Based Content Delivery Networks

MOHAMMAD A. SALAHUDDIN[1], (Member, IEEE), JAGRUTI SAHOO[2], (Member, IEEE), ROCH GLITHO[3,4], (Senior Member, IEEE), HALIMA ELBIAZE[5], (Member, IEEE), AND WESSAM AJIB[5], (Senior Member, IEEE)

[1]David R. Cheriton School of Computer Science, University of Waterloo, Waterloo, ON N2L 3G1, Canada
[2]South Carolina State University, Orangeburg, SC 29117 USA
[3]Concordia University, Montreal, Quebec, QC H3G 1M8, Canada
[4]University of Western Cape, Bellville 7535, South Africa
[5]Université du Québec à Montréal, Montreal, Quebec, QC H2L 2C4, Canada

Corresponding author: Mohammad Ali Salahuddin (mohammad.salahuddin@ieee.org)

This work was supported in part by Ericsson Canada, in part by the NSERC through the SAVI Research Network, and in part by the Quebec FRQNT Post-Doctoral Research Fellowship.

ABSTRACT This paper provides a comprehensive survey of content placement (CP) algorithms for cloud-based content delivery networks (CCDNs). CP algorithms are essential for content delivery for their major role in selecting content to be stored in the geographically distributed surrogate servers in the cloud to meet end-user demands with quality of service (QoS). Evidently, the key objectives of CP, i.e., cost and QoS, are competing. Cost is determined by the underlying cost model of the CCDN infrastructure while the delivered QoS is determined by where the content is placed in the CCDN. Therefore, we provide an overview of the content and the CCDN infrastructure. The overview of the content includes content characteristics and the influence of Online Social Networking on CP. The overview of the CCDN infrastructure includes elasticity and cost model, which affect CP. Our goal is to provide a holistic perspective of the aspects that impact CP algorithms and their efficiency. From the influential factors, we derive a set of design criteria for CP algorithms in CCDNs. We discuss the state-of-the-art CP algorithms for CCDNs and evaluate them against the well-motivated design criteria. We also delineate practical implications and uncover future research challenges.

INDEX TERMS Cloud-based content delivery networks, content placement algorithms, content correlation, content popularity, online social networking relationships, quality of service, resource utilization, user-generated content.

I. INTRODUCTION

Recent advances in utility and cloud computing allow leasing resources, such as storage and bandwidth, to build Content Delivery Networks (CDNs) in the cloud [1]. There is a growing trend to deploy cloud-based CDNs (CCDNs) or to complement traditional CDN infrastructure with cloud-based delivery, management and analytic services. Undoubtedly, there is a move to CCDNs – which is evident by the increase in traffic across datacenters, attributed largely to CCDNs [2].

CCDNs alleviate major limitations of traditional CDNs. In comparison to CDNs, CCDNs have increased scalability [3], flexibility [3], elasticity [3], reliability [4] and security against threats and attacks (e.g., denial of service) [4] and the orders of magnitude lower prices for content storage and delivery [1]. They reduce capital expenditure (CAPEX) and operational expenditure (OPEX) since the cost of deploying and maintaining infrastructure is significantly reduced. This makes CCDNs affordable for small and large-scale content providers, such as small businesses, government and educational organizations [5]. Moreover, CCDNs leverage the agility, scalability and elasticity of the cloud to dynamically provision resources to cater to changing demands [1].

There are various operational subsystems in a content delivery network, including content management and request routing [6]. The content management subsystem, illustrated in Fig. 1, is responsible for selecting content to be replicated and the surrogate server(s) that will host replicated content for meeting end-user requests with quality of service (QoS). The request router has a set of policies for directing end-user requests to surrogate server(s) either for load-balancing or QoS.

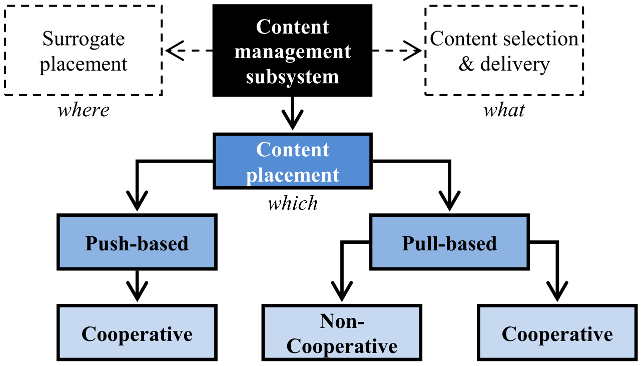


FIGURE 1. Content management subsystem roles and survey scope.

The content management subsystem (CMS) is vital for QoS. In the context of CCDNs, CMS decides *where* to place surrogate servers, *what* to replicate and *which* surrogate servers will hold replicas of content. Though these sub-subsystems are tightly coupled, they are studied jointly [7] and independently. In this paper, we critically review the CP algorithms for CCDNs.

In general terms, content placement algorithms are either pull or push, based on how surrogate servers get content from the origin server. Caching is a popular, pull-based technique employed to increase content availability and reduce content access latency. Push-based algorithms preemptively store content to meet an estimated demand. Cooperative and hybrid content placement algorithms retrieve missing content from the neighboring surrogates or employ push- and pull-based techniques, respectively.

The efficiency of CP algorithms is dependent on cost-effectively placing right content on the right surrogate servers. For QoS, it would mean placing popular content on servers in close proximity to end-users. This is easier said than done. Traditional CDNs were designed for static content [8] and as content evolved, the traditional CDNs evolved to meet the requirements. However, today's content, such as user-generated content and video (UGC/UGV), video-on-demand (VoD) and online gaming, is myriad, high-resolution and volatile due to social media sharing [9]. The petabytes of video traffic flowing through CDNs [10] show that the content catalogue is immense and it is non-trivial to identify and select popular content to store on the capacitated surrogate servers.

Therefore, CP algorithms must accommodate for high resolution content, which is highly susceptible to unpredictability, while it provides low latency. The unpredictability is primarily due to end-user behavior and Online Social Networking (OSN) relationships, which influence the upload time and size of content and various other aspects of content access (e.g., temporal and spatial patterns). These inherent content characteristics and end-user relationships are not adequately supported by traditional CP algorithms [9] and they must be leveraged to design efficient CP algorithms for CCDNs.

Furthermore, the underlying cost model of cloud computing poses a unique set of requirements for the CP algorithms impacting its efficiency, with respect to cost and QoS. For instance, in CCDNs, there is operational cost (e.g., leasing computing resources) that is incurred for the cooperation amongst surrogates across datacenters and for pulling content from the origin server(s). In contrast, in CDNs, there is no operational cost associated with pulling content from the origin or other surrogate servers. Instead, billing is proportional to the traffic delivered to end-users [8]. Readers interested in the evolution of CDN to CCDN and the interplay between the various actors are referred to [11].

The contributions of this survey are as follows:

1) We provide an overview of content and CCDN infrastructure, the two critical components that influence the content placement algorithms for CCDNs. Primarily, cost is determined by the underlying cost model of the CCDN infrastructure while the QoS delivered by CP is determined by where content is placed in CCDN.
2) We present critical and well-motivated design criteria for the CP algorithms for CCDNs based on practical implications from the content characteristics, end-user behavior, OSN relationships and the cloud model.
3) We review and thoroughly discuss the state-of-the-art CP algorithms for CCDNs and evaluate their effectiveness in the light of the design criteria.
4) We identify limitations and future research challenges in designing CP algorithms for CCDNs.

This is a timely survey and, to the best of our knowledge, is the first of its kind. It provides a reference for future research in the area of CP for CCDNs. This survey is unique, since it enlightens researchers to design CP algorithms from a holistic perspective. It emphasizes the effect of the cloud cost model, content characteristics, end-user behavior and OSN relationships on CP algorithms.

There are several surveys ([12]–[15]) on content placement in traditional CDNs, but unlike this survey, they do not tackle the algorithms specific to CCDNs. There is also a survey [16] on CCDN architectures, but it does not discuss algorithms. Furthermore, there is a survey that addresses algorithms [11] for CCDNs. However, it explicitly focuses on surrogate server placement algorithms and does not tackle the algorithms for content placement.

The rest of this paper is organized as follows: In Section II, we present the content placement problem and discuss the flow of content and problem solving approaches for CP. We also discuss various objectives and constraints in modeling CP problem and the parameter settings for solving it. Evidently, the key objectives of CP are competing: QoS versus cost. The delivered QoS is determined by the content placed in CCDN, while the cost is determined by the underlying cost model of the CCDN infrastructure. Therefore, we discuss content characteristics, end-user behavior and non-trivial OSN relationships that influence the efficiency of CP algorithms and derive critical design criteria that meet the requirements of today's content, in Sections III and IV, respectively.

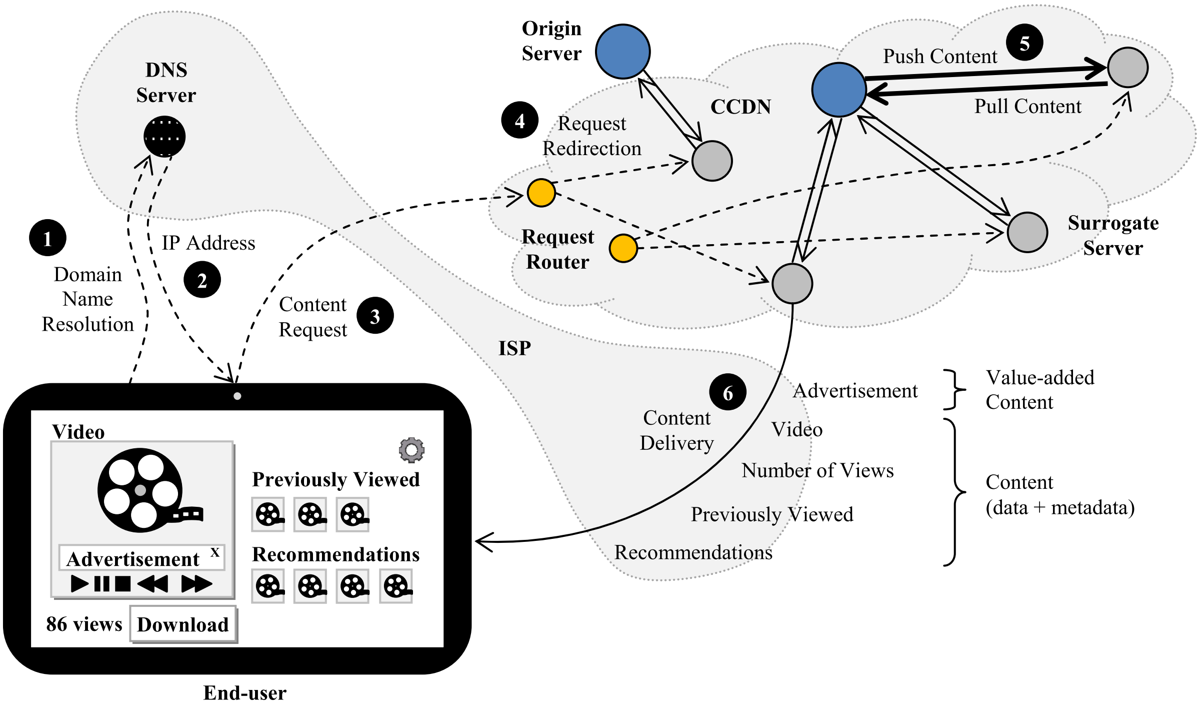


FIGURE 2. End-user and CCDN interaction.

Furthermore, we give an overview of CCDN infrastructure, its elasticity and cost model and discuss the design criteria for CP algorithms to meet the requirements of the underlying CCDN infrastructure in Sections V and VI, respectively. In Section VII, we discuss the pull- and push-based CP algorithms for CCDNs and evaluate them against our well-motivated design criteria. Section VIII deals with practical implications and future research challenges in designing efficient CP algorithms. We conclude in Section IX with an overview of the survey and its contributions.

II. CONTENT PLACEMENT PROBLEM IN CCDNs

In this section, we define the content placement problem and discuss the different strategies for solving the problem and choosing the various simulation and empirical parameter settings for analyzing the algorithm.

Let us begin by depicting the content delivery process, in Fig. 2. Initially, the end-user requests are diverted to request routers in CCDNs from the Domain Name Service (DNS) server as depicted in Steps 1, 2 and 3 in Fig. 2. Today's CCDN architecture consists of a set of surrogate servers organized in a hierarchical structure [17]. The top level is a set of primary surrogate servers, followed by a secondary and a tertiary set of surrogate servers. Generally, the *unicast* and *anycast* DNS names map onto the primary surrogate servers that can redirect requests to secondary and tertiary surrogate servers [17]. In this case, the primary surrogate servers generally interact directly with end-users.

The request router, in Step 4, Fig. 2, redirects end-user requests to other surrogate servers based on geographical proximity and content availability and/or for load balancing [18]. In commercial CCDNs, fine grain load balancing is often employed to redirect requests from a busy server to a less busy server in the same location [17]. Based on the efficiency and implementation of the request router, end-user requests are satisfied with QoS by a content placement algorithm, as in Step 5. Finally, Step 6 in Fig. 2 illustrates content that is delivered to the end-user(s) from CCDN via the Internet Service Provider (ISP).

Content, in the cloud-based content delivery networks (CCDNs), is essentially decomposed into metadata and data [19]. The metadata are the rules used to manage and control content. The content management rules can delineate how to cache content and how content will be updated or when it will be purged. They also determine the duration for storing content. Besides, the rules for controlling and customizing content are based on user profiles as they may have safety features turned on to avoid sensitive content. Similarly, content access may be controlled for distribution based on geographical locations, bans and censorship from governments.

Data is encoded media, which can be static, dynamic or value-added content. Typically, static content appears on a website as text, images or videos with no change over time. For example, the headings and the video file in Fig. 2 are both static content.

Dynamic content is a quasi-static document or template, composed of four different components. The front end is the visual interface component and the back end stores persistent data while the application logic component and user profiles generate dynamic and personalized data. The list of recommended videos in Fig. 2 is dynamically generated by the recommendation engine application based on user profiles. Similarly, the number of likes for a video is retrieved from a persistent database by the application logic.

Value-added content ([6], [19]) are specialized services provided by the CCDN provider. These services could be hidden from the end-users since they affect the metadata. For example, value-added content includes rules for caching, improving QoS, optimizing dynamic content delivery by reusing objects, streamlining mobile content delivery and/or security features (e.g., resilience against denial of service attacks). Value-added services can also be visible to end-users. One example is advertisements that are imposed on top of the delivered content. Essentially, value-added services are content that needs to be managed, maintained and delivered as content through a delivery network [20].

A. PROBLEM STATEMENT

Given that there is a set of origin servers $\mathcal{R}$, a set of surrogate servers $\mathcal{S}$, with $|\mathcal{R}| \ll |\mathcal{S}|$ and a set $\mathcal{T}$ of content. There is also a network graph $G = (V, E)$, where $V = \left\{\mathcal{S} \bigcup \mathcal{R}\right\}$ is the set of vertices representing the location of the origin and surrogate servers and E is the set of edges, where each edge $e_{i,j} \in E$ is a directional network link connecting servers v_i and v_j. For each surrogate server, there is a storage capacity c_i, $\forall 1 \leq i \leq |\mathcal{S}|$ and for each edge between v_i and v_j, there is a bandwidth capacity $b_{i,j}$ and cost $h_{i,j}$ for unit data on the edge link, a communication time $\Gamma\left(e_{i,j}\right)$, uptime, $\Phi\left(e_{i,j}\right)$, and downtime, $\Omega\left(e_{i,j}\right)$, between nodes v_i and v_j, $\forall 1 \leq i, j, i \neq j \leq |E|$.

Each content $\mathrm{t}_m \in T$, has a size v_m, type w_m, a correlation factor $f_{m,n}$ between content t_m and t_n, a popularity index p_m, a refresh/update rate of u_m and a QoS metric q_m, e.g. the end-user perceived latency, $\forall 1 \leq m, n, m \neq n \leq |\mathcal{T}|$. For each content t_m, there is an estimated request rate or demand d_m and a cost for placing t_m on a surrogate s_i is $g_{i,m} \forall 1 \leq i \leq |\mathcal{S}|$, with respect to size and type, v_m and w_m, and failure rate α_i, and each surrogate is bound by its storage capacity c_i, $\forall 1 \leq i \leq |\mathcal{S}|, 1 \leq m \leq |\mathcal{T}|$.

Find a content placement strategy, with the optimal number of replicas (copies) for each content $\mathrm{t}_j \in \mathcal{T}, 1 \leq j \leq |\mathcal{T}|$ and their placement on the surrogate servers in $\mathcal{S}$ such that each content t_j meets its QoS metric $q_j, 1 \leq j \leq |\mathcal{T}|$ while minimizing the cost of content hosting and delivery, and maximizing the bandwidth utilization and QoS for end-users, with respect to the perceived latency, availability and consistency.

Fig. 3 illustrates an instance of this CP problem, along with a feasible solution. In this scenario, there is one origin server with geographically dispersed surrogates catering to end-users around the world, via CCDN. The set of content on the origin server that is contracted to the CCDN provider is delineated in set $\mathcal{T}$ and the end-user demands for content on the different surrogate servers is shown in the figure. The objective is to efficiently replicate and place content on the surrogates to meet the end-user request with QoS, as illustrated in Fig. 3.

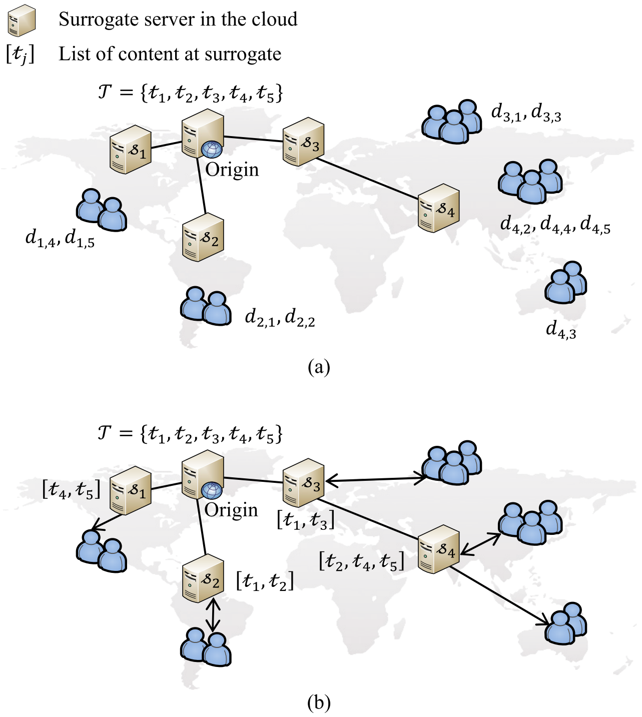

Surrogate server in the cloud
$[t_j]$ List of content at surrogate
$\mathcal{T} = \{t_1, t_2, t_3, t_4, t_5\}$
s_1
Origin
s_3
s_2
s_4
$d_{3,1}, d_{3,3}$
$d_{1,4}, d_{1,5}$
$d_{4,2}, d_{4,4}, d_{4,5}$
$d_{2,1}, d_{2,2}$
$d_{4,3}$
(a)
$\mathcal{T} = \{t_1, t_2, t_3, t_4, t_5\}$
$[t_4, t_5]$
s_1
Origin
s_3
$[t_1, t_3]$
$[t_2, t_4, t_5]$
s_4
s_2
$[t_1, t_2]$
(b)

FIGURE 3. Problem instance and a feasible solution. (a) A problem instance. (b) A feasible solution.

B. CONTENT FLOW IN CONTENT PLACEMENT

In general, content placement can be modeled as push- or pull-based strategies, as illustrated in Fig. 2, depending on the direction of content flow between the origin server(s) and the surrogate servers. Push-based content placement is proactive, whereas pull-based is reactive to end-user requests for content.

In push-based CP, content providers estimate end-user requests or predict content access patterns and replicate content from origin server(s) to surrogates, *prior* to receiving end-user requests for content. The pull-based approach is relatively simpler, where end-user requests instigate the surrogates to download and store content from the origin server(s) or the neighboring surrogate server(s).

Initially, in pull-based CP, all end-user requests will result in a *miss*, since the surrogate will need to download content from the origin or nearby surrogate server(s). Gradually, as the repository on the surrogate server grows, end-user requests will result in a *hit* and end-user requests will be directly satisfied by the surrogate server. The surrogates can cooperate with each other, to download content from another surrogate(s) that is in closer proximity than the origin server, or directly from the origin server in cooperative or non-cooperative schemes, respectively. In push-based CP, contingencies are in-place to cater to unpredicted end-user requests and content access patterns.

Generally, commercial CCDNs employ simple caching such as least recently used (LRU) ([7], [17], [21]) for content placement. The limited storage capacity of cache requires a cache management and replacement strategy. This is simplistic in implementation, such as low bookkeeping (maintenance) and traffic and storage overhead. However, these techniques should be improved for reliability, availability and other performance metrics [22].

Interestingly, simple caching with LRU cache replacement outperforms various push-based content placement algorithms, when the update period is delayed to approximately once in a day [22]. But, cooperative push-based CP algorithms also yield high performance over other content placement algorithms [23]. Generally, the efficiency of pull- and push-based content placement rely significantly on the accuracy of the prediction and estimation models for the prediction of end-user requests or content access patterns [24]. Therefore, caching, pull-based CP, is complementary to push-based CP that can work together to further increase the QoS of CCDNs [25].

This is essential in designing CP algorithms that eventually replace the content stored on the surrogate servers. Therefore, a hybrid ([26], [27]) of pull- and push-based content placement algorithms can leverage both the spatial and temporal correlations in data and the heavy tail distribution exhibited by the popularity of content.

In the following sections, we will discuss the different problem solving techniques for content placement in CCDNs.

C. PROBLEM SOLVING APPROACH

The content placement problem is defined as an optimization problem. In CCDNs, dynamic programming [28], convex programming [29], Lyapunov optimization [30] and Topkis-Veinott's feasible direction algorithm [29] have been used to produce local and global optimal results.

Traditional CP problems in CDNs, as decision problems, are thoroughly investigated ([14], [15], [26], [31]–[33]). They can be classified into replica-aware and replica-blind [31] CP problems, based on surrogates knowledge of the location of the copies, i.e. the replicas of content. The set of replica-blind CP problems are polynomial [31]. However, generic traditional CP problems have been successfully mapped to known NP-Complete and NP-Hard problems, such as facility location, knapsack and k-median ([26], [32], [34], [35]).

Similarly, modeling of CP variants belonging to NP-Complete (NP-C) [36] and NP-Hard (NP-H) [1] has been studied in CCDNs. Therefore, no polynomial time solution exists and hence, efficient heuristics have to be devised for large-scale and practical CP problems However, each step delineated in Fig. 2 is in itself an active area of research and poses numerous research challenges. We will see that the efficiency of content placement algorithms is inherently dependent on content, its intrinsic properties and relationships and the cost model of the underlying infrastructure.

D. FORMAL MODELING

The CP optimization problem has multiple objectives with various competing utility or cost functions. Generally, smaller CP optimization problems are addressed, incorporating only some of the cost functions. Typically, the different cost functions are broadly classified into latency minimization, operational cost minimization or joint minimization of latency and operational cost.

In latency minimization, the objective is to reduce the backbone network traffic, i.e. the traffic between the surrogates and origin server(s). Latency minimization problems include, but are not limited to, objectives that minimize distance and/or traffic between the end-users and the surrogates. Distance metrics can simply be the geodesic distance, Euclidean distance based on network topology, hop counts or complex distance metrics that account for network propagation, queuing, processing and transmission delays. Network traffic is induced between the surrogate servers and the origin server(s) for content retrieval, replication, updates, and/or consistency. The minimization of distance or traffic between the surrogates and origin server(s) explicitly places more content on the surrogates and implicitly improves QoS, with respect to the end-user perceived latency. Generally, a distance metric is used to conjure the end-user perceived latency ([29], [30]). However, network conditions are also used for the end-user perceived latency, such as traffic volume and round trip times (RTT).

Operational cost minimization objectives pertain to the cost of storage, cost of network bandwidth and cost of processing [37], if applicable. The cost of using network bandwidth is attributed to delivering content to end-users, retrieving content from origin server(s) or other surrogates to meet end-user requests, updating content, replicating content and/or maintaining consistency among the copies of content [31]. The cost of processing is included to accommodate for computational resources used in the cloud to compute or execute algorithms for content management. Operational cost minimization guarantees optimal cost for CCDN operations, while latency minimization guarantees QoS for end-users.

Therefore, these are competing objectives that are not mutually exclusive. For example, the cost of using bandwidth in the network is proportional to the traffic induced into the network for content access or retrieval. So, bandwidth utilization is important in CP models such that it does not congest network backbone with CCDN traffic. In such cases, these two objectives work together.

These objectives can be modeled independently or jointly from different perspectives. For instance, from the perspective of VoD and online gaming content providers, content must prioritize latency and consistency, respectively. Fig. 4 illustrates the various utility functions for the CP problem that can be included as the objectives for latency minimization and/or operational cost minimization. Many times, minimization functions can be modeled as maximization functions. For instance, minimizing latency can be modeled as maximizing

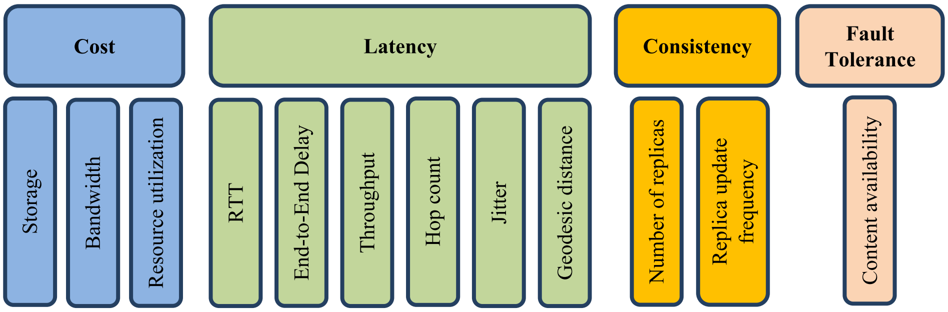


FIGURE 4. Utilities for modeling objectives and constraints.

traffic between the end-users and surrogate servers in close proximity to the end-users.

General utility functions, such as content access or bandwidth and storage cost, are essential for cost and latency minimization objectives. However, where multiple copies, i.e. the replicas, of content are globally distributed to achieve content availability, it becomes necessary to update content and maintain consistency among the replicas. In this case, either origin server(s) can invalidate content or the surrogate servers can validate content [38] for consistency. Since this induces additional bandwidth cost and complexity, it is often not deployed [38]. Furthermore, as content evolves to be more interactive and end-user driven, such as UGC, there is a growing need to include consistency management as an objective in content placement in CCDNs.

CP models and algorithms rarely account for the state of the underlying network as an objective. However, achieving fault tolerance should be a primary objective. In this case, CP models must maintain content availability, within QoS, in a '*lossy*' network, where surrogates and bandwidth could be unavailable due to link failure. Lossy networks are intrinsic characteristics of networks that have intermittent connectivity, such as vehicular clouds [39]. They consist of small-scale datacenters incorporated into processing units in vehicles and roadside infrastructures such as traffic lights. The vehicular cloud can offer CCDN services such that end-users can share and retrieve content while on the go [40]. Therefore, fault tolerance, content availability and consistency will become major objectives of CP in CCDNs, especially as they are overlay on non-traditional cloud infrastructures and underlying technologies.

E. CONSTRAINTS

As illustrated in Fig. 4, a CP model must be subject to various constraints to handle real world scenarios. The hierarchical structure inherently induces heterogeneity into the surrogate servers. Therefore, surrogate servers have different storage and processing capacities and utilization costs. Often, implicit storage constraints are modeled into CP, such as the k-replica constraint [29], which limits the number of replicas of content to k.

Similarly, the underlying network link layer will have different bandwidth capacities and utility cost functions. In CCDNs, the surrogate servers are generally in datacenters in the storage clouds that are divided into regions and zones. The links within zones in a region are low latency and high capacity, in contrast to inter-region bandwidth links.

Often, CP models are required to guarantee either hard or soft QoS metrics, i.e. either meets all or some of the end-user requests for content or meets some predefined latency requirements, such as end-to-end delay. Alternatively, QoS can be modeled as a maximization of the allowable violations in the SLA [39]. For instance, if SLA requires 99% uptime, this implies that up to 1% QoS violations are acceptable without penalty. Therefore, heuristics can be designed to leverage this slack.

It should be noted that various constraints can be modeled as objectives. Therefore, objectives are essentially relaxed constraints and it is not uncommon to find CP models where objectives and constraints are interchanged. A CP model with all utility functions as constraints is very strict as it does not allow the violations of any of the utility functions. However, utility functions in the objective allow violations that are penalized, which is the cost of the model. CP models often include only a few of the constraints or objectives while others are simplified or assumed to hold true.

F. PARAMETER SETTINGS

Algorithms can be validated analytically or via extensive simulation and empirical testing. In contrast to empirical and simulation validation, analytical validation can gauge the sub-optimality of the algorithm by defining a constant factor approximation to the optimal [41] or optimality for small-scale scenarios.

The effectiveness of the algorithms depends on various parameter settings [42]. Beginning with the underlying network model for CCDN, such as hierarchical ([34], [43]), random [31], Waxman [31], power law-based [13], etc. Interestingly, point-of-presence (PoP)-node-to-Internet Protocol (IP) address mappings [1] are complemented by latency information from web traces to build realistic scenarios.

Typically, content is gathered from VoD or UGV providers by crawling ([44], [17]) to gather UGC metadata and statistics or by analyzing HTTP traffic [45] or randomly [46] selecting UGC uploaded to YouTube. The YouTube UGC is decomposed into data and metadata. The metadata consists of static (e.g., title, upload time) and dynamic data (e.g., view count) [44]. The data consists of the static video content and a dynamic related video list, view statistics, including details about referrers and view counts due to the corresponding referrer, as illustrated in Fig. 5.

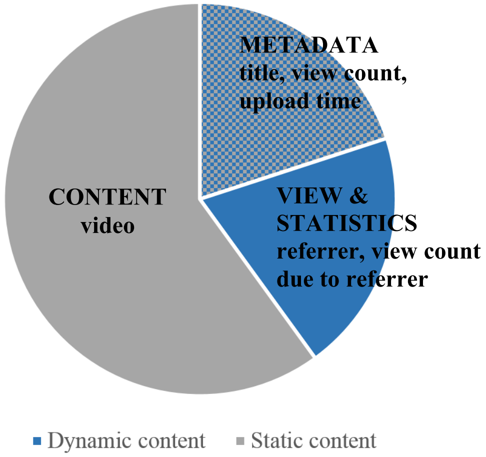


FIGURE 5. User-generated content (e.g., YouTube video) components.

Generally, the probabilistic and statistical moments of a random variable (e.g., mean, variance, percentile, etc.) are employed to describe the popularity and correlation of content. However, various analytical, empirical and learning techniques are also used for setting parameter values, as illustrated in Fig. 6. For example, numerical analysis, entropy and dispersion, similarity analysis and machine learning can be employed for content characterization.

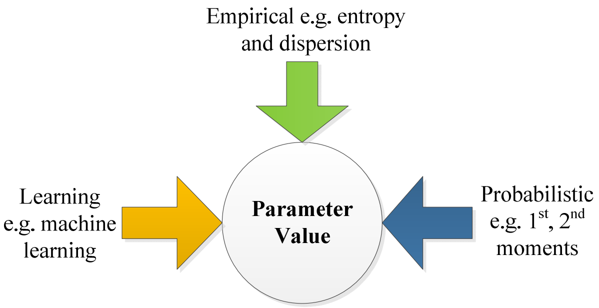


FIGURE 6. Techniques for deducing parameter values.

Linear regression [47] and hybrid regression models [48] are often used for predicting the popularity of content. A Gini coefficient can measure the distributional inequality and the value of the Gini coefficient measures the gap between popular videos and niche video [44]. Entropy and dispersion techniques are employed to study end-user behavior such as the probability of replay, view distribution and patterns [49].

Entropy and peak intensity [46] are used in the study of content correlation and relationships. These techniques can measure the intensity and consistency of the end-user interests in a specific spatial area. Interestingly, anomaly detection and traffic analysis are also employed to uncover content relationships [50]. Reinforcement learning techniques are also used in unsupervised analysis through clustering [50] of data and designing placement algorithms [24].

In this way, content relationships are formulated, defined or configured to study regularities or irregularities, similarity or dissimilarity in content relationships [50], using entropy, temporal similarity and distribution-based techniques. TABLE 1 delineates essential parameters and their typical empirically derived values (cf. Section III-B for discussion on parameters).

TABLE 1. Empirically derived values for essential parameters.

	Parameter	Value
Content	Static content image	$\leq 10^4$ bytes [51]
	Fixed chunk size for UGV (e.g., YouTube)	1.8, 2.5, 3.7 MB [50]
	UGV bitrate	≤ 1 Mbps [50]
	Number of replicas	Power law distribution w.r.t. popularity [29]
End-user	Number of views per day	30–40 [49]
	Total viewing time	≤ 2 hours [49]
	Duration of one view	≈ 3 minutes [49]

III. OVERVIEW OF CONTENT HOSTED ON CCDN, END-USER BEHAVIOR AND OSN RELATIONSHIPS

In this section, we discuss the various characteristics of content that greatly influence the performance of content placement algorithms. We present an overview of characteristics of classical content and then discuss the uniqueness of contemporary content, end-user behavior and OSN relationships.

A. CLASSICAL CONTENT CHARACTERISTICS

An extensive analysis of static and dynamic content indicates that, though there are various content characteristics that are dependent on end-user behavior in the long-term, there are some that are independent of end-user behavior. There are two independent content characteristics required for describing content, size and freshness.

The content size comprises of the size of metadata and encoded media stored on the surrogate servers. Content freshness is the interval between updates for content. Other characteristics of content that are typically dependent on end-user behavior include popularity, age, transmission times and spatial and temporal correlation.

Fig. 7 illustrates a classification of content based on independent content characteristics. The size of static content is best captured by Pareto, a heavy tailed power law distribution [51]. Power law functions have shape parameters that can measure the skew in the data. They have heavy tails and large values along the x-axis that allows analysts

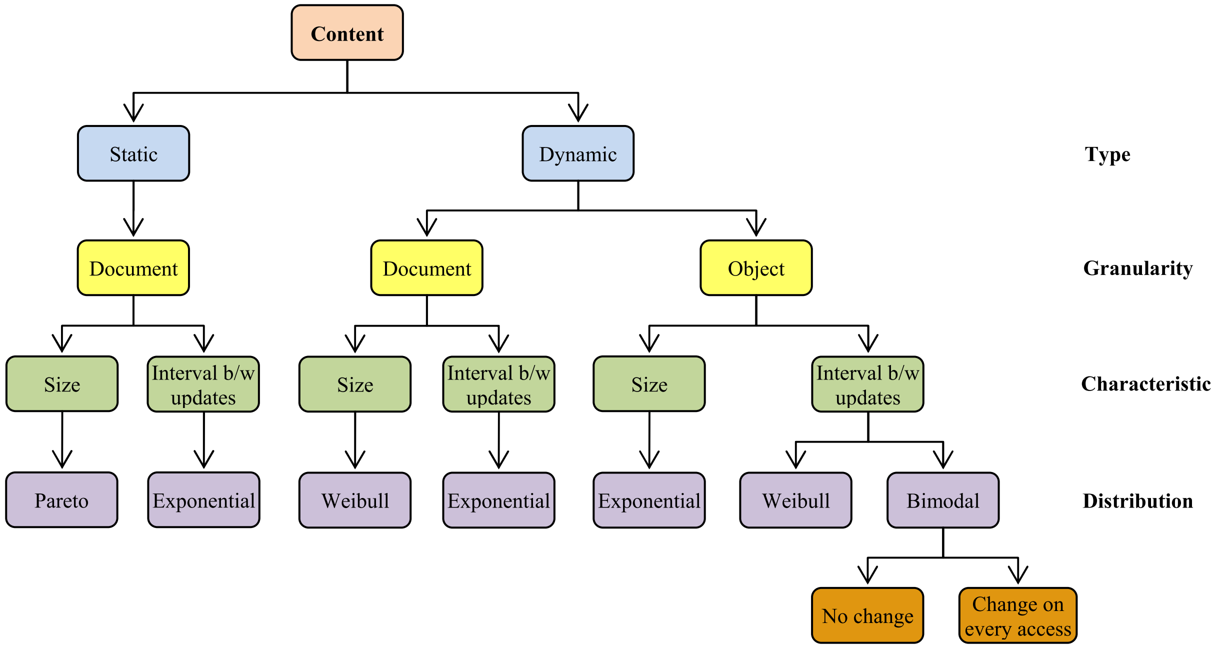


FIGURE 7. Classification of independent content characteristics.

to quantitatively study the data points, that otherwise would have been written off as outliers. In contrast, Gaussian normal distributions capture populations, which have 99.7% of the data within 3 times the standard deviation and the data outside these ranges is typically written off as outliers. The interval between updates of static content is modeled by Weibull, a variant of Exponential functions [52], to capture very short time intervals (within few minutes to a day) between content updates [53].

Similarly, dynamic content is decomposed into coarse documents and finer objects. At the document level, the content sizes are best captured by Weibull distribution [54], whereas Exponential functions best characterize the interval between updates [52]. The size of the objects in a document are found to be exponentially distributed [55]. The freshness time of these objects are best modeled by Weibull Distribution [55]. However, some objects in dynamic documents also exhibit bimodal characteristics [55], requiring either no updates or continuous updates.

Heavy-tailed distributions have been also shown to capture other independent content characteristics, such as the number of end-user requests [51], transmission times [51], idle times [56] and age of content [52]. Though transmission times are vulnerable to the underlying network traffic, it has been shown that they also follow a heavy-tailed distribution.

Popularity is a major characteristic of content to be placed efficiently – i.e. it is popularity that ranks content. The rank is based on the number of end-user requests for content access. The popularity of content closely follows the discrete Zipf distribution such that static content that is not ranked highly can still generate end-user requests.

The estimated time for the end-user requests to arrive is assumed to follow a Poisson process to capture the randomness and independence of the arrival of end-user requests. As a result, the interval time between end-user requests follows the Exponential distribution [54].

Though various characteristics of content are dependent on the end-user behavior, very few are correlated. For instance, there is negligible correlation between popularity, the content size and the update rate for static content [57]. The popularity of static content is uniformly distributed across hot servers [57]. However, objects in dynamic content exhibit temporal and spatial correlation [55]. This implies that there is a high probability of predicting future object requests based on the current object requests. And similarly, objects stored on the same physical server are more likely to be reused in a linked dynamic document than those stored on physically different servers.

B. CONTEMPORARY CONTENT CHARACTERISTICS

On top of the classical content characteristics, contemporary content characteristics include mobility, interactivity and OSN relationships. These characteristics greatly influence the design of CP algorithms in CCDNs [7]. In 2014, 64% of global consumer Internet traffic was video content and 57% of this video traffic was delivered through CDNs [10]. By 2019, the video traffic is estimated to increase by 80% [10] and the traffic coming from mobile devices and applications will increase by ten folds [10]. It is imperative to cater to the near future demands of the global Internet traffic by delivering varying video resolutions in different formats with QoS via efficient content placement algorithms in CCDNs.

Fig. 8 illustrates the various components of video streaming, such as chunk size, format, resolution and playback mode. The resolution is inferred from the bitrate [50] and the playback mode depicts whether video is in playback, fast forward, rewind or pause mode. Traditionally, video content is decomposed into chunks of pseudo-equal sizes and delivered through CDN infrastructure. These components are leveraged to predict end-user behavior and interactivity, which influence the performance of content placement algorithms ([45], [46]).

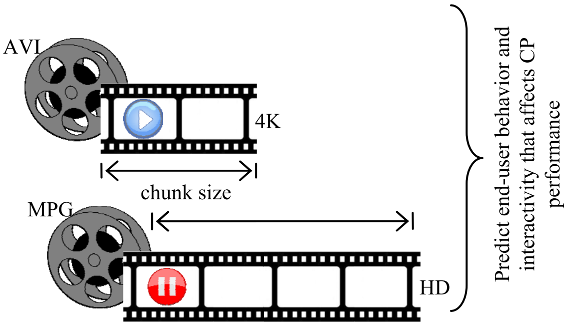

AVI
4K
chunk size
MPG
HD
Predict end-user behavior and interactivity that affects CP performance

FIGURE 8. Components of streaming video content - chunk size, format, resolution, bitrate and playback mode.

For mobile devices, such as tablets, laptops and cell phones, multimedia content is stored in different resolutions and formats [45]. Transcoding techniques are applied to match device requirements to the delivered video content. The chunks vary in size based on mobile or fixed platform, such as personal computer (PC). The chunk sizes are typically fixed in PC and variable in mobile environment. They are often limited by buffer size, dictated by the device [45].

Today's video content hails from providers that offer VoD and UGC services. Videos are selected from catalogues that host dozens of categories, such as Music, Movie, Family, Children, Sport, News, Classic, Featured, Recently Uploaded, etc. Generally, end-users can only play videos through VoD services, whereas end-users can share, edit and delete their own multimedia content through applications that offer UGC services. Furthermore, today's VoD and UGC applications and services also include an OSN perspective, where end-users can interact with each other to suggest, share and rate video content hosted in VoD and UGC applications. This complicates popularity prediction and leaves it susceptible to unpredictability [7].

From the perspective of CP algorithms, the chunking mechanism in video streaming, the video operations (e.g., fast forward and reverse), the spatial and temporal correlations and the complex OSN relationships pose new challenges. Today, videos are generally hosted in VoD systems or as UGV on UGC hosting sites, such as Netflix and YouTube, respectively. End-users can only subscribe to watch videos in the catalogue of VoD providers whereas end-users produce and consume videos hosted by UGC providers.

Both VoD and UGC providers host videos but they are strikingly different [58], as illustrated in Fig. 9. VoD content has a finite, though huge, catalogue and is stored on edge of content delivery network [59] whereas UGC catalogue is seemingly infinite as it is continuously growing with content uploaded by end-users and stored within the content delivery network [59]. VoD content is usually uploaded by VoD content providers at off-peak times and their prediction models can closely estimate the demand for content. Therefore, VoD content is less volatile than UGC [58] and UGV has a virality attribute [60] due to the interactivity of end-users that share and link UGV on OSN [61]. The predictability of VoD greatly reduces the burden for CP algorithms [49] since video content is controlled by the content provider and is not erratically uploaded, edited and deleted by end-users, as in UGC.

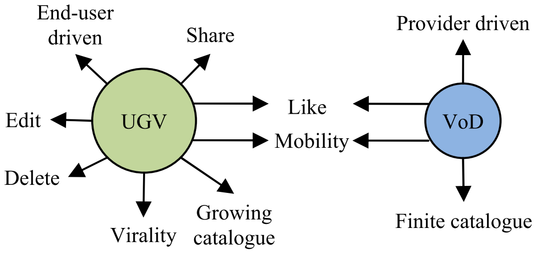


FIGURE 9. Traits of UGV (e.g., YouTube) and VoD (e.g., Netflix).

The VoD and UGC catalogue contain videos that can be classified into a dozen different categories, such as Children, Sports, News, Music, Movie, etc. The number of end-users that view the video is recorded and maintained as the view count in the metadata of the video. Typically, video popularity is inferred by the view count. However, it can be assessed by two different techniques: 1) Based on the temporal evolution of the number of views and 2) the number of views before and after the maximum (the peak of) number of views [46].

Fig. 10 illustrates the lifecycle of a video, which resonates with the temporal evolution of popularity. The first phase of a video is the *hot* phase, when it generates high view counts, usually within the first week of being released/hosted [49]. As videos get older, they enter the *warm* or *lukewarm* [44] phase, where popularity and the view count are attributed to video search, spread by word-of-mouth [49] or appearing on recommended and related video lists [44]. As the video gets older, it enters the *cold* [49] phase, when the popularity significantly wanes and the video hardly generates any view counts.

Though videos are inherently static, only *warm* videos exhibit popularity that is very well estimated by the Zipf function ([49], [58]). The popularity of hot videos is much larger than those predicted by Zipf functions [58]. The popularity of cold videos falls sharply [58], i.e. the popularity of content that lies at the far end of long tail functions, such as Zipf.

It should be noted that different techniques are used to collect data, from an ISP ([50], [58]) or from campus networks [45]. The data from ISP is diverse and depicts

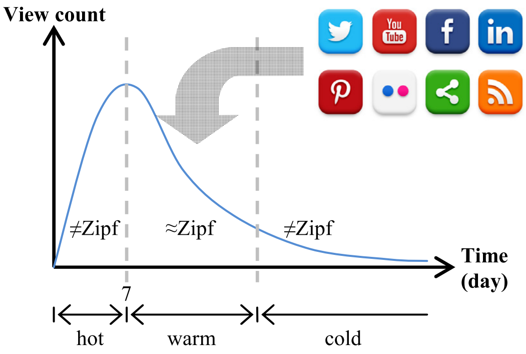


FIGURE 10. Lifecycle of video based on temporal evolution.

end-users whose behavior is very varied. The data from campus networks introduces a natural bias. For example, in a university campus network results are skewed since end-users are primarily students with different content access patterns. Meanwhile, data can be collected by crawling ([17], [44]) to gather UGC metadata and statistics or analyzing HTTP traffic [45] or randomly [46] selecting UGC uploaded to YouTube. However, despite the data collection scheme, popularity distribution of all samples follows the Zipf distribution.

All categories of videos follow similar trends with respect to temporal evolution in popularity ([44], [46], [49]). The key characteristics can be identified as follows: 1) Only a small subset of hot videos is selected to be featured in recommended [49] and trending video lists, which significantly skews their popularity [44] and therefore their view counts are distinctly different from warm and lukewarm videos [46]. 2) Although all categories follow similar trends with respect to popularity, there are some categories that are highly sensitive to the age of video, such as those appearing under News and Sports categories, since less people are interested in stale News [49]. 3) Certain categories are susceptible to local popularity since they are bound by cultural or language barriers while others enjoy global popularity, i.e. their popularity spread to other regions [46].

Most VoD and UGC service providers cater to mobile end-users, such as those accessing VoD and/or UGC videos via tablets and mobile phones. So, mobile content is a special subclass of VoD and UGC. However, content geared for mobile devices require special handling. VoD and UGC service providers must deliver content with QoS, through a communication medium known for slower downlink rates and intermittent connectivity.

Mobile multimedia content is delivered to smart phones, tablets, set-top-boxes and smart TVs that access content via wireless and 3G/4G Internet connections. The wireless medium and the video streaming process pose challenges for the CP algorithms.

C. END-USER BEHAVIOR AND OSN RELATIONSHIPS

End-user behavior and OSN relationships are not easy to be objectively characterized since they are dependent on various external factors and intrinsic human behavior and psychology. However, it is imperative to understand these traits since they can greatly increase the efficiency of CP algorithms in CCDNs [62].

End-users have a short attention span and limited time to devout to watching videos. For instance, an end-user may suffer from constrained eyeball [49], which restricts the time he can watch a video. Typically, 85% of end-users watch videos for less than 90 seconds [50], with a low probability of replay, whether it is due to the end-users' lack of interest or poor quality of experience (QoE) [49]. Such external factors complicate an accurate popularity prediction, which is essential for CP algorithms in CCDNs.

Apart from the general end-users' actions (e.g., play, forward and/or reverse playback positions of video content), end-users can change the resolution of playback and the mode of playback screen. These features are offered by various VoD and UGC service providers, such as Netflix and YouTube. The resolutions range from standard, high or ultra-high definition and the mode of playback screen is either standard or full screen. Generally, VoD and UGC applications playback a video under default resolution and mode settings that the end-users rarely and only slightly change [45].

More recently, VoD and UGC providers have expanded their applications and services by offering Dynamic Adaptive Streaming over HTTP (DASH). This is in contrast to the traditional, fixed, constant quality video streaming mechanism. In DASH, different resolutions are used within a single video playback using many different video bitrates. In case of YouTube, the bitrates are dynamically adapted to varying bandwidths [50]. For YouTube, DASH is increasingly becoming the default option set by end-users since they enjoy near optimal viewing experience with DASH [50]. Though DASH achieves high QoE, its dynamic adaptation complicates CP algorithms for CCDNs.

Other end-user characteristics [49] and VoD content relationships include the relationship of end-user viewing time and view duration, the replay probability and subscription and the end-user membership and its effect on replay probability. As the number of videos the end-user accesses increases, the duration of the videos selected decreases. Users with higher number of view counts tend to have a lower replay probability and they seldom replay the same videos many times. Instead, many different videos may be replayed by a user each for a few times. However, the overall probability of replay is very low. Interestingly, active users tend to have a higher replay percentage. That is, when a user is actively subscribed to a service, there is a higher probability that the user is interested in content and hence the replay probability is higher.

OSN also plays a vital role in aggregating the view count of a UGV. End-users post links to UGVs (their own or those made publically available by others) that are posted on the UGC service providers' website. Service providers (e.g., YouTube) record the number of views from the referral videos and sites. Social sharing [46] is view counts that are attributed to end-users accessing UGC by clicking on a link from an OSN website or application. Fig. 10 illustrates the impact of OSN on the lifecycle of videos hosted in the UGC catalogue.

An average of about one third of the view counts for a YouTube video is based on social sharing [46]. UGV and OSN relationships are complex. Although UGV benefits greatly from social sharing as its popularity rises, social sharing becomes less influential in raising the view count of UGVs [46]. Content can enjoy global or local popularity with respect to geographic regions. OSN relationships can influence this geographic spread in a non-trivial manner, that is, too much or too little social sharing restricts the geographic region of UGVs. However, for hot UGVs, there is no correlation between global popularity and social sharing.

IV. DESIGN CRITERIA OF CP ALGORITHMS BASED ON CONTENT

There are various implications of the inherent characteristics of classical and contemporary content, end-user behavior and their OSN relationships on content for CP algorithms. However, their implications can be broadly classified as those that influence *content access patterns* or *popularity*. As illustrated in Fig. 11, these are critical in designing efficient CP algorithms for CCDNs.

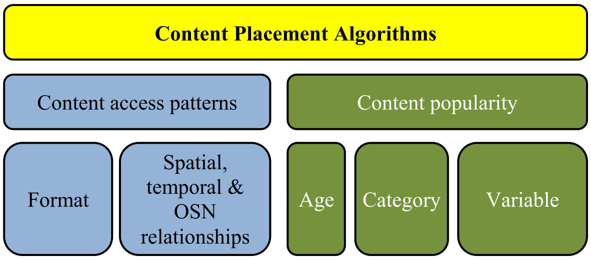


FIGURE 11. Content based design criteria for CP algorithms.

Content access patterns not only account for spatial and temporal locality, but they also include devices that generate the content access request. The heterogeneity in end-user mobile devices requires a set of different resolutions and formats of content to be stored for content delivery. This requires CP algorithms for CCDNs to meet the requirements of content access for heterogeneous end-user devices.

Traditional CDNs were designed for static content [18] and accommodating CP algorithms to cater to content for mobile devices faced numerous challenges [18]. The number of mobile devices is expected to far exceed that of the desktop and hard-wired devices, requiring content providers hosting in CCDNs to keep multiple formats of the same video content. This supports the diversity in hardware and software in the devices. For instance, though YouTube caches multiple formats of a video, it still results in a miss and degradation in QoS, since all video formats are not equally popular and end-users may be redirected to datacenters that are farther away in proximity [45]. These are required for designing effective CP algorithms in CCDNs.

The temporal, spatial and OSN relationships influence content access patterns. End-users within a geographical region have increased similarity. End-users in similar social groups or OSN relationships will have similar content access patterns. Therefore, temporal, social and OSN relationships can influence content access patterns and should be leveraged in CP algorithms to minimize the storage cost and increase QoS in serving the end-user requests.

Another intrinsic property of content is correlations based on recommendation engines or referred videos or sites. It is important that CP algorithms are sensitive to content access patterns of correlated and referred videos since they can significantly impact the popularity of content. Content popularity follows heavy-tail distributions, such as those parameterized by power law and exponential functions. Idealistic distributions (e.g., Gaussian Normal) and power law distributions can capture populations that vary in orders of magnitude ([51], [63]). Their heavy tails, i.e. large values on x-axis, allow to study the data quantitatively, that otherwise would have been written off as outliers. This trend in popularity implies that, though pull-based approach is simple and effective, it will not give high hit ratio as the data in the long tail will generate misses [63]. The cache hit rate is also dependent on the cache size and the eviction probability of the content [49]. Furthermore, the low probability of replay [49] of video content implies that the popularity *will* eventually decrease, suggesting that any content will eventually become stale and require replacement. As long as the stale content occupies space on the cache, it will generate a cache miss.

Nonetheless, CP is commercially achieved via pull-based caching (e.g., UGC on YouTube). This is because bandwidth is a resource in CCDNs that must be optimized and caching with simple LRU yields high byte hit ratio that efficiently utilizes the cloud bandwidth resources. The global file hit ratio is generally low, i.e. around 35%, due to the UGC that lies in the long tail [58]. However, considering a byte hit ratio, the heavy hitters, i.e. the popular files, push the byte hit ratios significantly higher, to about 75% [58].

Moreover, the popularity of content can be arbitrarily modeled by a power law function, but different categories of content are represented by different power law functions. The shape of the power law functions is controlled using shape parameter in a power law function. This influences efficiency in modeling the popularity of content that belong in different categories since popularity is directly proportional to the age of various video categories. Therefore, Least Recently Used (LRU) and Least Frequently Used (LFU) cache replacement strategies alone are not adaptive enough to the dynamically changing popularity of content in different categories [49]. However, FIFO performs best for News category since it corresponds to the observation that stale news looses end-user interest. Therefore, old news videos should be replaced first [49] whereas, for movies, music and TV, LFU enables cache replacement algorithms to effectively reflect the dynamic popularity of these categories [49].

A mixed cache replacement strategy can adapt to changing popularity [49]. For example, LFU and FIFO probabilistically evict the least frequently used or the oldest content and outperform simple LFU and LRU with moderate overhead [49]. Furthermore, the high spatial correlation in content access patterns instigate the preferential caching of videos for different geographic regions [46], as end-users with similar interests are geographically clustered in the same country, town or neighborhood. Furthermore, cache management can also include filtering techniques [58], such that content is stored on the surrogate servers after a certain number of end-user requests have been received. Unfortunately, filtering reduces byte hit ratio since heavy hitters enter cache late and do not maximize bandwidth savings [58]. Therefore, it is critical to design CP algorithms that are sensitive to dynamically changing popularity, age and content category.

V. OVERVIEW OF CCDN INFRASTRUCTURE

In this section, we provide a brief overview of CCDN infrastructure, with respect to cost and elasticity in the resource provisioning of the underlying cloud model. Though traditional CDNs resemble other classical data sharing distributed systems, they are intrinsically different ([8], [18]). For example, similar to CDNs, mesh networks increase connectivity, data grids provide data storage, distributed systems have replication groups, Peer-to-Peer (P2P) networks deliver content for distribution. Yet, they are fundamentally different. Furthermore, today's cloud-based solutions for data sharing have increased resiliency, reliability, accessibility and security.

The infrastructure of traditional CDNs consists of high bandwidth links connecting geographically distributed network elements, such as routers and switches with clusters of surrogate servers. CDNs can be built based on different approaches, depending on the way network elements interact to interpret end-user requests. Software Defined Networking (SDN) ([64]–[66]) and, Network Functions Virtualization (NFV) [60] paradigms are used to build CDNs. Meanwhile, CDNs can be built via the in-network caching and routing of the request to the appropriate surrogate servers. On the other hand, Telco-CDN [67] has been devised for network operators to optimize resources for content delivery, in an Internet Service Provider (ISP) managed CDN infrastructure.

In contrast, overlay CDNs are built by strategically placing content on the surrogate servers in the network and they use request redirection mechanisms that route content requests to the appropriate surrogate servers. In this overlay approach, network elements *only* perform traditional routing duties. There are major drawbacks of traditional CDNs, including a high cost of content hosting [1], complexity of hosting dynamic content [18], dealing with the paramount growth in the size of content and the lack of dynamic scalability of resources [18].

CCDNs can be private or public. In private CCDNs, the content owner owns the cloud infrastructure to store and deliver content to end-users. However, in the case of public CCDNs, content providers can lease cloud resources to build their own CCDN or utilize services provided by the public CCDN infrastructure providers (e.g., Amazon CloudFront [68] and Google Cloud CDN [69]). Irrespective of the type of CCDN, it will incorporate content delivery through an Autonomous System (AS) ([45], [50]) that can include routers, core and access networks, accounting for virtualization and software-defined technologies.

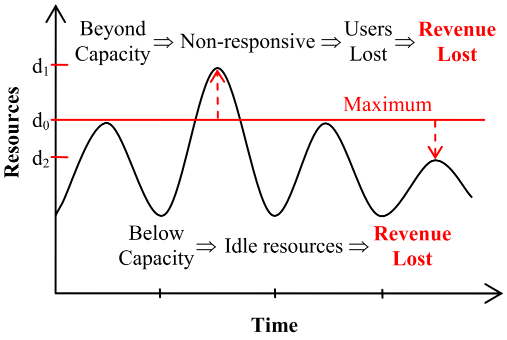

Beyond Capacity ⇒ Non-responsive ⇒ Users Lost ⇒ Revenue Lost
d1
d0
d2
Maximum
Resources
Below Capacity ⇒ Idle resources ⇒ Revenue Lost
Time

FIGURE 12. Effect of dynamically changing demands on traditional CDNs.

Besides, it should be noted that it is not necessary to migrate all content to the cloud or to CCDNs. There are hybrid approaches ([70], [71]), where content can be uploaded to the cloud and distributed via CCDNs when end-user demands for content exceed the content provider's bandwidth capacities. These possibilities can be easily leveraged by CDN providers that move to the cloud and the content providers that do not anticipate extreme scenarios such as flash crowds.

Let us consider a CCDN built by content providers by leasing the cloud resources from a public cloud infrastructure provider. The cloud resources include storage, computation and bandwidth. It consists of regions including various zones, with high capacity datacenters. They provide low latency, high bandwidth links between zones and intra-region communication over the Internet. CCDN providers pay for storage and bandwidth leased on the cloud. The storage cost is based on a flat rate depending on the content catalogue size and the bandwidth cost is decomposed into traffic coming into and going out of zones and regions in the cloud.

The inter-region and intra-region communication links have different bandwidth capacities and costs. Typically, inter-region bandwidth costs are higher and bandwidth capacity is lower, in comparison to intra-region bandwidth costs and capacity. However, leading cloud infrastructure providers (e.g., Google) do not charge for intra-region bandwidth usage or traffic coming into regions. In Fig. 13, we illustrate a cost function for bandwidth usage, with respect to the traffic going out of the regions. The non-increasing cost function is inspired by Google's network rates [72] that decrease per unit cost as the number of consumed units increases. Note that these rates vary from region to region; however, they follow a similar trend. This implies that cost-effective content placement can be achieved by increased resource utilization.

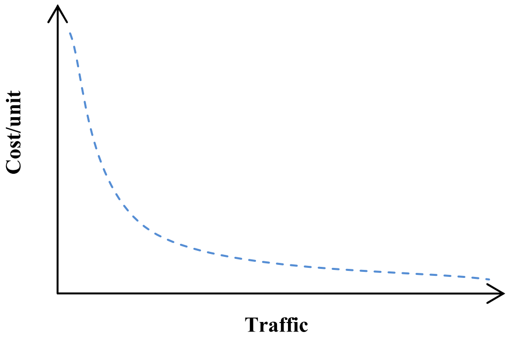


FIGURE 13. Bandwidth cost function.

Dynamic resource provisioning is a severe limitation of traditional CDNs, which causes over, under or at capacity utilization of resources. As illustrated in Fig. 12, content providers *estimate* an average demand for their content and contract CDN providers for the respective resources, say d_0, with Service Layer Agreement (SLA) guaranteed QoS. Static resource allocation is perilous and when demands need resources, say d_1, they exceed the maximum available resources of CDN and the content provider's service is rendered unresponsive while end-users are dissatisfied. This results in the loss of reputation and financial revenue for the content providers. Meanwhile, when demand only utilizes resources, d_2, it is underutilization and results in idle resources that are the eventual loss of revenue.

CCDNs overcome these shortcomings of CDNs [1] with the advent of the cloud and utility computing. Cloud computing infrastructure consists of large-scale datacenters with hundreds and thousands of machines [73], spread across the globe [5], inter-connected with high-bandwidth links, offering low latency. The cloud infrastructure providers offer pay-as-you-go cloud resources, such as storage and bandwidth, which content providers can lease to build an overlay CDN in the cloud, i.e. a cloud-based CDN. CCDN operators can dynamically allocate and de-allocate resources across geographically distributed datacenters to cater to continuously changing end-user demands and popularity of content.

The intrinsic differences in the infrastructure of CCDNs and CDNs limit various operational subsystems to be simply used "as-is" in CCDNs. Therefore, though, CP algorithms in traditional CDNs are mature and a logical predecessor to CP for CCDNs, they cannot be directly applied for CP in CCDNs.

VI. DESIGN CRITERIA OF CP ALGORITHMS BASED ON CCDN INFRASTRUCTURE

In this section, we discuss the design criteria that significantly impact the efficiency of the CP algorithms due to the intrinsic cost model of CCDNs. Generally, CP algorithm must be designed to reduce operational cost, maximize SLA-defined QoS and resource utilization and provision resources to meet the dynamically changing popularity of content and end-user content access patterns.

Operational costs or OPEX are the cost of consuming resources in the cloud. These costs include the cost of storing content on the surrogate servers in datacenters and the cost of consuming bandwidth for the retrieval and update of content. These are costs incurred by CCDN providers using third party cloud infrastructure. For these CCDN providers, it is important to minimize these operational costs without compromising end-user perceived QoS.

Quality of service for end-users is defined as soft or hard guarantee on one or more QoS metrics. A hard QoS guarantee will ensure that the guaranteed QoS metric is never violated. However, soft QoS is implicit and is achieved indirectly, for example, by maximizing traffic between the surrogate servers and end-users and minimizing traffic between the origin server and surrogate servers. However, soft QoS is tolerant to QoS violations.

The QoS metrics can be network health and/or communication metrics quality, such as end-user perceived latency, end-to-end delay, geodesic distance, jitter in delay, hop count, round trip time and/or other network distance functions. Recently, there is a shift from QoS to QoE [50], which is user-centric and subjective. It is hard to monitor and guarantee QoE ([50], [74]). It should be noted that QoS and QoE cannot be used interchangeably since they are inherently different parameters for capturing the end-user satisfaction with CCDN service.

CP algorithms for CCDNs also greatly benefit from resource utilization and dynamic resource allocation. Resource utilization is intrinsic in maximizing the use of already leased resources, rather than leasing new resources. Though it is tightly coupled with operational cost, it is different than just minimizing the cost of resource allocation. Therefore, resource utilization and operational cost are not interchangeable objectives. Explicitly, resource utilization is defined as delivering maximum content from the same leased surrogate server in the cloud before leasing new resources in the cloud.

Resource provisioning is the ability to lease or release resources to meet the changes in the end-user demands. It leverages the elasticity of the cloud to adapt the leased resources such that they can be released to meet lower than expected end-user demands or more resources can be leased to meet a sudden surge in end-user demands. It directly impacts operational costs and QoS design criteria. This design criterion intersects with designing a CP algorithm that dynamically adapts to the popularity of content. This is one major benefit of moving CDNs into the cloud to leverage the elasticity and flexibility of the cloud to meet the evolving interest of end-users in the published content. This enables CP algorithms to move correlated content into and out of the surrogate servers based on the referred content since they affect each other's popularity.

Therefore, critical design criteria based on the cloud for content placement algorithms are *operational cost*, *QoS*, *resource utilization* and *resource provisioning*, as shown in Fig. 14. Though these criteria are tightly coupled, to simplify the implementation and maintenance and to reduce the overhead of the CP algorithms, they are often decoupled.

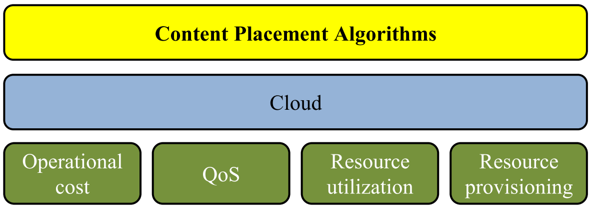


FIGURE 14. Cloud model based design criteria for CP algorithms.

TABLE 2. Classification of CP Algorithms for CCDNs.

Algorithm	Classification	
	Algorithm Type	Content Flow
GRP [1]	Centralized Greedy	Push
DPC and CRB [75]	Centralized Greedy	Push
SNA-GVSP [36]	Centralized Greedy	Pull
Holistic approach [76]	Distributed Greedy	Push
Soft-ConFL [77]	Constant Factor Approximation	Push
DTLM [27]	Distributed Knapsack inspired Heuristic	Hybrid
Enhanced DFS [28]	Depth First Search	Push
Content placement [29]	Convex Optimization	Push
Dynamic algorithm [30]	Lyapunov Optimization	Push
W-SNA [78]	Centralized Greedy	Push
TTL-based cache [79]	Non-Convex Optimization	Pull

VII. EVALUATION OF CP ALGORITHM FOR CCDNs

In this section, we review state-of-the-art CP algorithms for CCDNs. In TABLE 2, we present the classification of CP algorithms for CCDNs based on the algorithm type and the used approach, whether it is push- or pull-based. CP algorithms typically leverage one or more of the following approaches such as centralized and distributed greedy, searching routines, scheduling and allocation routines, tree-based sub-optimal searching techniques, game theoretic approach, simulated annealing approach, genetic algorithms, and optimization and approximation techniques. In TABLE 3, we delineate the CP algorithms designed for CCDNs and identify their objectives, constraints, assumptions, justifications and insights. These algorithms are discussed in the following subsections.

A. PULL-BASED CONTENT PLACEMENT IN CCDNs

Caching is a popular pull-based technique for increasing content availability and reducing content access latency. It is widely employed in a wide range of applications, such as document and data replication. It is the reactive pull-based CP in CCDNs and fetches content on-demand. Since surrogate servers have limited resources, with respect to the size of the catalogue, they must be optimally utilized. Commonly, cache replacement algorithms dictate content and the order in which content should be purged from surrogate servers to make space for new content. The components of pull-based CP algorithms are shown in Fig. 15.

Generally, caching can be decomposed into en-route caching, hierarchical caching [21] and their hybrid.

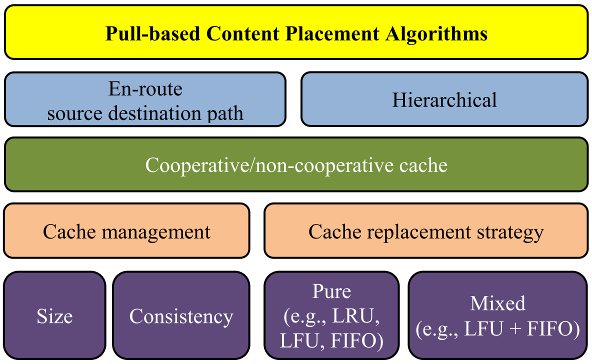

Pull-based Content Placement Algorithms
En-route source destination path
Hierarchical
Cooperative/non-cooperative cache
Cache management
Cache replacement strategy
Size
Consistency
Pure (e.g., LRU, LFU, FIFO)
Mixed (e.g., LFU + FIFO)

FIGURE 15. Components of pull-based CP algorithms.

In en-route caching, end-user requests are served by the first surrogate server that contains content, in the path from the end-user to the origin server. However, in hierarchical caching, the requests are propagated along the hierarchical levels until it can be satisfied.

En-route caching in traditional CDNs has been shown to solve the set of replica-blind CP problems in polynomial time [32]. The algorithm implementations range from k-optimization [32] and dynamic programming [32] to min-cost flow problem [14] – to mention a few. CP algorithms using en-route caching strategies can be extended to include robustness for QoS, surrogate server capacity and content consistency ([31], [38], [80]). Hierarchical caching for CP organizes surrogates into internal and core surrogate servers [81] for jointly minimizing the cost of content access and maximizing the hits. Evidently, a hybrid between en-route [82] and hierarchical caching is often explored in CP to increase content availability and reduce content access times [21].

Cooperative caching for CP includes placement, searching and consistency management [14]. Traditional cooperative caching for CP updates content when a miss generates a request for content with higher utility value than any other content [43]. A utility value is associated with content to denote its demand on the surrogate server while other caching algorithms assume equal utility values for all content.

Generally, traditional commercial CDNs employ caching for CP with LRU cache replacement [21] due to its simplicity and ease of implementation and maintenance. Similarly, pull-based caching for CP in CCDNs [79] devise a cache with a TTL-based cache replacement policy. The pull-based caching accounts for elastic resources in CCDNs and jointly optimizes operational cost and content miss.

The TTL cache replacement policy purges content if it is not accessed within a fixed, predetermined period of time. The content delivery costs include the cost of leasing bandwidth resources in the cloud for content retrieval and the cost of storing content on surrogate servers that pull content on a miss. Therefore, requests cannot be redirected as per traditional en-route or hierarchical caching techniques since they would incur additional operational costs.

TABLE 3. Summary of CP Algorithms for CCDNs.

Algorithm	Objective(s)	Constraint(s)	Assumption(s)	Justification(s)	Insight(s)
Greedy Request with Pre-allocation (GRP) in Chen *et al.* [1], adopted from approximation of the set covering problem	Minimize content access cost. Content access cost accounts for fetching from origin, distributing to end-user, storage and updates	• QoS based on distance	• Tree structure induced from the distribution paths between origin server and end-user		• Cloud-based CDN framework, accounts for upload, download and storage costs • Dynamically adapts to real end-user request patterns and can create new surrogates, intrinsic to CCDNs • Distinguishes between original size of content and size of content requested by end-user
			• Distance metric can capture hop count or delay • No surrogate server storage capacity • No bandwidth capacity	• Benefit of cloud–elasticity, immediate increase in storage and, or bandwidth • QoS distance is implicitly used to bound the bandwidth	
Differential Provisioning and Caching (DPC), and Caching and Request Balancing (CRB) in Hu *et al.* [75], iterative and greedy	Minimize total rental cost, including bandwidth and storage rental and minimize end-user requests routed from origin server	• Bandwidth capacity • Surrogate server storage capacity • Soft QoS • Changing end-user demands • Request routing	• No change in end-user demand within a time period	• If time periods are relatively small, approximately 30 minutes used in performance evaluation, then there is no change in end-user demand within a time period	• Dynamically adapts to changing end-user demands • Two fold content placement algorithm, long (30 minutes) and short term (10 minutes) • Soft QoS guarantees • Multiple surrogates can be assigned to meet same end-user request, i.e. service splitting, that is multi path service delivery
			• Same unit size content	• Since large files, such as videos, are decomposed into multiple smaller files, of same size for caching and one video file request is divided into multiple smaller requests.	
			• No update on content		
SNA-Inspired Greedy Virtual Surrogate Placement (SNA-GVSP) in Papagianni *et al.* [36]	Maximize shortest path betweeness centrality (SPBC)	• QoS based on distance metric • Surrogate storage capacity • Bandwidth capacity • Content access and update	• QoS measured with distance metric	• The maximum routing distance can capture the communication quality that can be measured in terms of hop count or delay. In their absence, geographic distance is sufficient	• Cloud-based CDN accounts for storage, bandwidth and content retrieval costs. • Social Network Analysis (SNA) based greedy heuristic
Holistic approach in Katsalis *et al.* [76], online and cooperative.	Minimize operational cost	• Surrogate server storage capacity • Varying size content	• Number of contents considered in the order of 10^3	• Realistic number of contents on Internet is magnitudes larger, however, with Zipf distribution for popularity and the number of content considered is sufficient for valid comparison	• Cloud-based CDN, consisting of multiple domains • Cooperative • Locality of interest is important in performance
Soft-ConFL in Rappaport and Raz [77]	Joint minimization of cost of content update, content access and surrogate server placement	• Surrogate server capacity constraint	• Content access rate is larger than content update rate • No network underlying link layer bounds • No dynamic content or surrogate placement		• Utilize existing efficient constant factor approximation solutions to the facility location problem and its variants
Distributed Traffic-Latency-Minimization (DTLM) in Guan and Choi [27]	Maximize cost of pushing content to surrogate server	• Surrogate server storage capacity			• Cooperation amongst surrogate servers • Hybrid push-pull optimization model

TABLE 3. ***Continued.*** **Summary of CP Algorithms for CCDNs.**

Enhanced DFS in Wang *et al.* [28], centralized, algorithm, dynamically re-evaluated	Jointly minimize cost of leasing resource and cross-region traffic	• Locality, that is, users within region are assigned to surrogate server in the region	• Startup delay in leasing new surrogate from cloud	• Latency for preparing cloud server	• Systematic organization in cloud and end-user to ensure QoS and minimize cross-region traffic • Dynamically adapt to changing end-user demands
			• Intra-region traffic should be maximized • End-users can assist in content distribution with similarity to peer-to-peer technologies		
Content placement problem in Jin *et al.* [29]	Minimize the cost of storage and bandwidth	• Storage and capacity constraints	• Content sizes follow bounded Pareto distribution • Popularity in terms of download times follows a Zipf distribution	• Verified by real traces • Verified by real traces	• Optimal solution dues to convex optimization model • The optimal number of replicas for a content • Logarithmic relationship for mean hop distance between end-user and content
			• Uniform traffic pattern	• Typically load balancers are in place that ensure approximately equal network traffic across different surrogate servers	
Dynamic algorithm [30] in Hu *et al.*	Minimize cost of storage, bandwidth and replication from source to CCDN node	• Requests must meet QoS metric of average time delay	• Storage is capacitated		• A tradeoff between serving content from source or CCDN node is investigated
Weighted-Social Network Analysis (W-SNA) based heuristic in Salahuddin *et al.* [78]	Minimize cost of storage, bandwidth and degree of QoS violations	• All requests met • Capacitated storage and network links	• Delay is based on LUT using G/G/1 queuing model		• Leverage SLA violations to minimize storage and bandwidth cost
TTL-based cache [79] in Carlsson *et al.*	Optimize cache miss cost, serving from non-nearest surrogate server and cache storage cost	• TTL based cache replacement strategy	• Fixed TTL value rather than exponentially distributed values		• Organize surrogate servers in such a way that some meet only local requests while others meet local and global requests, that is, requests generated within proximity of surrogate server or beyond proximity, respectively.

Naively, if end-user requests are redirected just to minimize the content delivery costs to the nearest surrogate server, which may or may not have content, it will increase the cost of content delivery as there is cost associated with redirection, content retrieval and content storage on the redirected surrogate server. However, if requests were redirected in a top-skewed scheme [79], to the surrogates that have the highest request rates and lowest content delivery costs, then there is a high probability that it will yield a hit since content may be already stored there. Otherwise, since this is a surrogate server that serves a high number of requests, there is a higher probability that even if content is not already there, the pulled content will yield future cache hits. Since caching algorithms are on-demand, they are intrinsically dynamic in resource provisioning [79].

Recently, caching with mixed cache replacement strategies [49] have been devised to accommodate for dynamic popularity based on a short window of observation and the different characteristics of videos in different categories. The considered categories are News, Sports, Movies, TV or Music. Video content from these categories are assigned a probability p for discarding the videos based on the category and age of content. For example, videos in News category are age sensitive so they have a high probability of eviction from the surrogate in the event of a miss and there is a need to replace content on surrogate.

In the academic realm, there has always been a continuous struggle to justify pull- or push-based CP for content delivery [67]. However, a simple caching technique with LRU technique outperforms various push-based CP algorithms, when the update period is delayed [22]. But caching can only increase the maximum hit ratio by 40-50% [83]. In short, caching is an online, distributed, greedy, local variant of push-based CP algorithms [13], which purges content faster than push-based CP algorithms [26].

B. PUSH-BASED CONTENT PLACEMENT IN CCDNs

In the context of push-based content delivery algorithms, the first major step is the assignment of end-users to surrogates that meet their requests, while minimizing cost and maximizing QoS. There is a prerequisite to this assignment, which is the mapping stage when end-users' requests are mapped to the surrogates in CCDN by a request redirector based on geographic locations or for load balancing.

Typically, it is assumed in push-based CP algorithms that the mapping is arbitrary and a set of surrogate servers with a set of end-users are the input to the push-based content delivery algorithm. The end-user requests are estimated and content access patterns are predicted [27] based on prediction models. The surrogate locations are either arbitrary or based on the datacenter locations of large-scale cloud infrastructure providers [78]. An approach to the end-user, surrogate assignment problem is to model it as the re-known Facility Location (FL) problem [77].

Steiner tree approximation [77] can be applied to the end-user, surrogate assignment problem to achieve solutions within a constant factor from the optimal. On the other hand, when the end-user, surrogate assignment problem is modeled as the knapsack problem [27], it eventually begins with the prioritization of surrogate servers. The prioritization is achieved based on the benefit of the surrogate to the entire CCDN, the perceived QoS of the end-users or per-unit weight-based algorithms.

Algorithms that consider the selection of surrogate servers in CCDN are prioritized based on their benefit of reducing network traffic and latency [27]. Uniquely, surrogate servers cooperate to prioritize and store content that minimizes operational cost but maximizes content stored in each level of hierarchy in CCDN, which is organized in tiers based on the inherent technology domains [29]. The components of push-based CP algorithms are shown in Fig. 16.

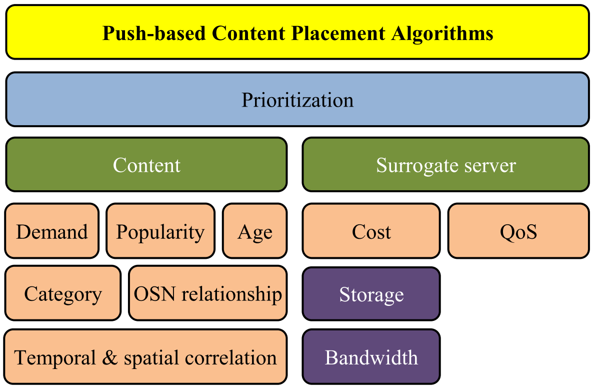


FIGURE 16. Components of push-based CP algorithms.

In QoS-based surrogate server prioritization techniques ([1], [29], [30], [36]), end-users are assigned to surrogate servers such that all end-users are within a QoS distance of the surrogates and their content. The QoS distance can be based on hop-count, delay or geodesic distance [36].

In per-unit weight-based surrogate prioritization techniques, bandwidth-storage ratio ([28], [75]) prioritizes surrogate servers such that the cost of storing and delivering content is optimized. The weighted shortest path metrics [78] are often used to prioritize the surrogates that offer the lowest content delivery cost and the best QoS such that the surrogate is on the shortest path based on betweenness centrality (BC). The BC metric prioritizes surrogates such that the surrogate with the largest number of shortest paths passing through it with the lowest storage and bandwidth costs has the highest priority.

Typically, the end-user, surrogate server assignment problem is tackled by prioritizing surrogates. However, end-users can be clustered based on geodesic distances, complex OSN relationships and similarity preferences in content [30] to implement CP algorithms that are meticulously designed to account for end-user behavior and OSN relationships. The push-based CP algorithms that terminate after end-user, surrogate server assignment are static pre-allocation algorithms ([29], [36], [78]), executed for a static snapshot of time.

The push-based CP algorithm using convex optimization with Topkis-Veinott's feasible direction algorithm [29] minimizes the cost of storage and bandwidth in terms of the mean hop distance between end-users and content. Meanwhile, greedy heuristics ([36], [78]) have been designed to implement end-user, surrogate assignment inspired from the shortest path betweenness centrality (SPBC).

Realistically, there is a discrepancy between the estimated/predicted and real end-user requests since the predicted end-user requests is only as good as the prediction model. The difference depends on the accuracy of the estimation and/or prediction models. Dynamic push-based CP algorithms are pre-allocation algorithms followed by a dynamic adaptation phase that is executed online during live end-user requests.

Dynamic push-based CP algorithms can adapt to changes in end-user requests or content access patterns by either re-provisioning the resources ([28], [30]) or re-routing the end-user requests from busy surrogate servers to under-utilized surrogates [75]. Alternatively, push-based CP algorithms can also dynamically update the content placed on the surrogate servers [1], while incurring minimum operational cost, if end-user requests are violating QoS.

Generally, centralized knowledge is necessary for end-user surrogate server assignment. However, distributed push-based CP algorithms ([27], [76]) have been devised for resilient CP in CCDNs. Moreover, hybrid push-pull CP algorithm [27] for CCDNs has also been employed to minimize operational cost, maximize QoS and dynamically adapt to the changing end-user requests, leveraging the benefits of pull- and push-based CP. Meanwhile, hybrid approaches ([70], [71]) are being investigated to offload content from the origin server(s) to CCDNs, only when the original traditional CDN capacity is exceeded. However, this does not account for a long-term effect on operational cost or on leasing and re-leasing resources when content is pushed to the cloud, with respect to frequently changing end-user demands.

In TABLE 4, we evaluate the CP algorithms for CCDNs against the well-motivated design criteria based on content and the cloud model. It is evident that, though the content characteristics are intrinsic to CP algorithms and impact their efficiency [62], it is often overlooked in the design of CP algorithms. Furthermore, temporal, spatial and OSN relationships affect content access patterns, which are rarely included in CP models and algorithms. Hu *et al.* [7], [30] use OSN relationships to classify end-users into similar groups and achieve 30% improvement in performance of CP algorithm. This is the evidence that content characteristics should be included in the design of CP algorithms for CCDNs.

VIII. PRACTICAL IMPLICATIONS AND RESEARCH CHALLENGES

In this section, we discuss the practical implications and research challenges of designing CP algorithms, with respect to content characteristics, end-user behavior and OSN relationships and the cloud model requirements.

A. CLOUD MODEL

1) OPERATIONAL COST VS. QoS

The dynamic and short-term resource provisioning in the cloud is also the underlying contributor to operational cost. There is a direct tradeoff between operational cost and QoS that a CP strategy offers. It is imperative to scrutinize and quantify the tradeoff or the cost of striking a balance between operational cost and QoS. For example, Jin *et al.* [29] have identified a novel relationship between the number of replicas of content and the mean hop distance between end-users and content. Critical research objectives include (i) the quantification of the effect of the number of replicas on QoS for end-users, (ii) analyzing the cost of offering hard and/or soft QoS, (iii) the effect of overhead pertaining to leasing and initializing resources in the cloud on QoS for end-users, with respect to delay. However, there are some key performance indicators and metrics for measuring and defining the subjective QoE [50]. The challenge lies in scrutinizing the impact of QoE parameters over QoS metrics for CP in CCDNs.

2) PUSH VS. PULL

Despite advances in CP algorithms for CCDNs, various limitations are present for future research. Markedly, a crucial objective includes finding the correct niche for caching in the evolving CP algorithms. They have shown promising results in traditional CDNs [22] and in today's CCDNs [79]. Jia *et al.* [26] propose a hybrid between push- and pull-based CP that can tremendously improve the performance of CP algorithms for client-side web-server proxies. Furthermore, Guan and Choi [27] show the benefits of a hybrid approach by pushing some content pro-actively while pulling others on demand to optimize content delivery using the storage clouds. Therefore, it is important to realize the two approaches are complementary for content placement in CDNs [25]. The key research challenges include analyzing (i) the effect of caching on QoS, (ii) the effect of pushing or prepositioning content on operational cost and (iii) the net gain from pull-based or push-based content placement algorithms.

3) RESOURCE PROVISIONING

It is important to employ a CP algorithm that does not over- or under- utilize the bandwidth and storage capabilities of the surrogate servers [84] simply because resources can be leased and released. The continuous leasing and releasing of resources add additional overhead to the CP algorithm. Therefore, it is an open challenge to strike the perfect balance between performance with respect to cost, QoS, utilization of leased resources and overhead.

4) SIZE OF SURROGATE SERVER

The efficiency of pull-based content placement is directly proportional to the size of storage on the surrogate server. Though cloud resources can be leased to adapt to the size dynamically, it is a challenge to find the balance between operational cost and efficiency of CP algorithms with respect to QoS. Therefore, for pull-based CP algorithms, the challenge that emerges is to select appropriate storage allocation for content placement ([1], [49], [58]).

B. CONTENT CHARACTERISTICS

1) POPULARITY

Though popularity of a video is highly localized, there exists some categories where videos can accumulate enough views to spread to other regions. Let us assume that there exists a threshold on the view count of a YouTube video, which determines whether the video has attained a local or global popularity. The challenge lies in predicting the threshold and scrutinizing its benefit for CP in CCDNs. It remains a challenge to identify the benefits of this knowledge, i.e. whether content is globally or locally popular.

Furthermore, different categories of the catalogue exhibit different popularity with respect to age. Different cache replacement algorithms have been explored to replace content sensitive to the different categories. However, whether different CP algorithms should be designed for different video categories needs further research.

There is a strong correlation between the popularity of video and the popularity of its related videos [44]. Therefore, it is worth investigating the correlation and identifying more relationships, such as the correlation between the *age* of top referrer of a video and the view count of the related video.

2) MINING CONTENT

Future research challenges in CP for CCDNs must include the objective of leveraging analytics that will extract fundamental characteristics of content, enhancing the efficiency of CP algorithms. Researchers have leveraged content characteristics, such as correlation, popularity [29], type (static or dynamic), refresh rates, temporal and spatial locality of end-users [28], social network relationships [78], etc. that can be leveraged to pre-fetch content by CP algorithms.

Pre-fetching closely related content [61] or incorporating complex content dimensions increases content availability and reduces content access times. However, CCDNs cost

TABLE 4. Evaluation of CP Algorithms for CCDNs.

Algorithm	Cloud Model				Content Characteristic				
	Operational Cost	QoS	Resource Utilization	Resource Provisioning	Access Pattern		Popularity		
					Format	Temporal, Spatial & OSN relationship	Age	Category	Variable
GRP [1]	✓	Hard	✓	Dynamic	×	×	×	×	×
DPC and CRB [75]	✓	Soft	✓	Dynamic	×	T	×	×	×
SNA-GVSP [36]	✓	Hard	✓	Static	×	×	×	×	×
Holistic approach [76]	✓	×	×	Static	×	×	×	×	×
Soft-ConFL [77]	✓	×	✓	Static	×	×	×	×	×
DTLM [27]	✓	Hard	✓	Dynamic	×	×	×	×	✓
Enhanced DFS [28]	✓	Hard	✓	Dynamic	×	T, S	×	×	×
Content placement [29]	✓	×	×	Static	×	×	×	×	×
Dynamic algorithm [30]	✓	Hard	×	Static	×	O	×	×	×
W-SNA [78]	✓	Soft	×	Static	×	×	×	×	×
TTL-based cache [79]	✓	×	✓	Dynamic	×	T	×	×	✓

model poses a challenge. Therefore, it is crucial to tackle CP problems with the objective of striking a balance between incorporating content relationships and the cost of leasing resources for content storage and bandwidth. Another research objective can be to scrutinize the effect of content personalization [16] on CP algorithms for CCDNs. Furthermore, recent advances in ubiquitous computing pose a challenge in designing CP algorithms for CCDNs such that the content of different resolutions are stored [85].

3) UGC

Though the videos are inherently static and they display characteristics similar to traditional static content, VoD and UGC requires special handling. First, it has a myriad catalogue, which makes traditional CP algorithms cumbersome and ineffective for CCDNs. Traditional CP algorithms replicate and store content based on a popularity index that is conjured based on a long-term window of observation of content access, which is not feasible for the myriad catalogue hosted by VoD and UGC providers. Second, they require special streaming media servers for streaming videos. And lastly, VoD and UGC providers offer additional services, such as video search, recommended or related video lists and/or trending video lists that pose new challenges for CP algorithms for CCDNs to accommodate evolving and dynamic popularity on today's VoD content and UGC. The size and dynamic nature of today's VoD and UGC systems pose unique requirements for CP algorithms for CCDNs.

4) MOBILE CONTENT

Multimedia content delivery to mobile devices face major challenges. First, it requires the transcoding of multiple device dependent video formats and, second, inefficient video chunking. The inherent mechanism used for video chunk delivery to mobile devices is inefficient [45] and requires multiple TCP connections, whereas it is optimal for wired devices [45]. Therefore, it is a challenge to design efficient video delivery mechanism and improve CP of different video formats to reduce cache miss for mobile content.

5) DYNAMIC CONTENT

Various aspects have to be considered when delivering dynamic content via CCDNs. Dynamic content consists of different components. Therefore, it should be analyzed whether the front-end, back-end, application logic or user profile should be replicated. Generally, the front-end consists of static data and can be replicated by using static content replication strategies. Since the back-end and user profiles both consist of databases, similar strategies can be employed for replicating the database to diverge traffic away from the central database. The database replicating strategies include partial or full replication [18]. For CCDN content replication purposes, the application logic is also replicated to surrogate servers. CCDNs are supporting dynamic content by introducing various optimization algorithms and infrastructure deployments for the replication of dynamic content.

6) AVAILABILITY VS. CONSISTENCY

There is a tradeoff between content availability and consistency management [38]. To ensure content consistency, either the origins can invalidate content or the surrogates can validate content, in the invalidation- or validation-based schemes, respectively [38]. However, consistency mechanisms are not often deployed due to their complexity and cost performance tradeoff [38].

C. END-USER BEHAVIOR

The evaluation reveals that there are various challenges in studying end-user behaviors and their effect on CP algorithms. For instance, can CP algorithm designers leverage the short watch time [45] of end-users to predict the formats and chunks to host across datacenters? Can it be quantified how rarely the cold videos are viewed [49] and deduce a threshold on view count that can be employed to measure when content is cold to leverage it in the design of CP algorithms? End-users who are subscribed to a content service or are just guest end-users behave differently [49]. It remains a challenge to deduce metrics, such as ratio of online active and inactive end-users to adapt or infer CP algorithm characteristics. Since there is a low probability of replay [49], adding metadata to the recorded statistics and information about where the end-user would leave off in the last playback of a certain video should be considered.

1) OSN RELATIONSHIPS

It is non-trivial to study and quantify the complex OSN relationships and their effect on CP [46] and the influence of spreading [44] in social networks on content placement.

It is especially hard to understand, if the popularity of UGV rises, how social sharing becomes less important [46] when the contrary makes more sense. A research challenge would be to find the referrer that can be attributed to voluntary lookup by users due to popularity from word-of-mouth. Can it also be attributed to UGV being listed in the "trending" or "recommended" category? In either of these cases, end-user would arrive at YouTube video without social sharing referral. It may be recommended/trending, therefore views and popularity increase just by navigation rather than sharing. The reason is that social sharing pertains to links that are specifically selected by end-users on a social networking website, whereas non-social UGV are typically referred from the "recommended", "related" or "trending" lists [46].

YouTube recommendation system is a very close second to YouTube search engine for the top sources for driving view counts [44] and there is a strong correlation in popularity of the videos listed under the recommended/related video list and the popularity of the video on the watch page. Consider a video v that has a video r listed in its recommended/related video list, then v is the referrer video and r is the related video. There is a strong correlation in the view count of r with the average view count of its top referrer [44]. Therefore, video v has a high probability of becoming popular if it is on the recommended/related list of a popular video r. However, the position of the recommended video on the recommended list is also vital such that recommended videos on the top have approximately 40% probability of being selected by end-users [44]. That is, the *click through rate* [44] is high for the items listed higher on the recommended video list. These features affect and are in-turn affected by the popularity of the content they host, which pose unique requirements on the CP algorithms for CCDNs.

The recommendation systems and search engines are both top sources for driving view counts [44]. Therefore, this insight can be used to build prediction models for CP algorithms that preemptively host content, since it can be predicted with a high probability that related videos will be clicked. It should be noted that the position on the recommended list is also vital and a top related video has up to 41.6% chance of being clicked [44]. That is the click through rate is high for items listed higher on the recommended video list.

Therefore, the challenge lies in leveraging the recommendation systems to build CP algorithms that can move UGV content based on the recommendations. Also, push-based CP algorithms have a greater chance at success over pull-based in this regard, since caching cannot update content as fast as push-based schemes. A "group of referrer video for a certain video is a good estimate and indicator of view count of video" [44]. This can be leveraged in CP algorithms, by predicting the UGV content that has a high probability of being requested based on the high click-through rate, investigated in [44].

2) REQUEST PREDICTION

End-user request patterns are imperative in the design of efficient push-based CP algorithms. Though the performance of such prediction algorithms has significantly grown in recent years [67], Qui *et al.* [86] show that even using imperfect estimates for content demand are sufficient for CP algorithms in traditional CDNs. However, with CCDNs, an open research challenge includes accurate and efficient demand prediction models and a scientific analysis of their effect on CP algorithms. For example, Zhou *et al.* [44] study prediction models that account for content access frequency, correlation, and popularity. Future research objective can account for the effect of social networking and its complex relationships on the prediction models for end-user requests for content.

D. OTHER IMPLICATIONS

1) REQUEST REDIRECTION

Another research challenge is the effectiveness and cost analysis of a smarter CCDN request redirection scheme, which greatly impacts QoE [50]. Request redirection is aware of the surrogate servers and their content and redirects end-user requests based on load-balancing, QoS or network health metrics. This scheme can utilize neural network or reinforcement learning techniques to build a surrogate-content relationship table such that the redirection is simply a look up in the table. However, the exchange of information between the redirector and the surrogate servers incurs bandwidth and storage costs at the redirector. It is interesting to scrutinize the economic benefits of this redirector and the latency and responsiveness of this scheme. It remains a challenge to evaluate CP algorithms efficiency with [79] and without request redirection techniques.

2) SOLUTION ANALYSIS AND COMPUTATIONAL COMPLEXITY

Tremendous work has been done on studying and classifying traditional CDN CP problems as polynomial, NP, NPC or NP-Hard. Traditional CP problems based on CCDN infrastructure have been successfully mapped onto known NP-Complete and NP-Hard problems such as facility location, knapsack and k-median ([32], [34]). There are special cases and variants of both the decision and optimization of CP problems that are tractable ([15], [26], [31]–[33]) using tree-based [26] algorithms or dynamic programming ([31], [33]). However, unrealistic assumptions, such as surrogate servers have infinite capacity [26] or only one level of hierarchy [15], limits their applicability. Interestingly, CP model is polynomial, when the surrogate servers are assumed to be "replica-blind" [31], i.e. unaware of the location of content replica. In such cases, end-user requests are propagated from one surrogate server to another along the path to the origin, known as en-route caching, to solve the CP problem in polynomial time [32]. The CP problem as a min-cost flow problem [14] is solved optimally in polynomial time while assuming a low rate of updates and a low rate of change in content access patterns. Building such an extensive repository for CP in CCDNs remains a challenge.

3) FOG/EDGE COMPUTING

There is a growing interest in scrutinizing content placement by using smaller, non-traditional surrogate servers, such as cellular-base stations and set-top-boxes ([87], [88]). This area of fog/edge computing in CDN requires research in content delivery techniques across heterogeneous networks, such as cellular/CDN, cloud/cellular edge, etc. [89].

4) HETEROGENEOUS NETWORKS

It is important to realize that the path for content delivery in CCDNs traverses various access networks, ISPs and heterogeneous infrastructures. Researchers can scrutinize the limitations of these domains and leverage them to optimize content storage and delivery by minimizing operational cost and maximizing QoS [29]. For example, the traffic between CCDN and ISPs, CCDN and cellular networks and CCDN and set-top-boxes can all be streamlined. This can be achieved by using low priority edges/links and utilizing surrogate servers within access networks that reduce overall operational costs and increase QoS ([87]–[90]).

5) AT-SCALE TESTING

CP algorithms for CCDNs must include the objective of gaining empirical results from at-scale test beds, especially with SLA and QoS parameters. Since the cost of CP in CCDNs benefits from the elasticity of the cloud, it is also important to setup research problems that analyze the frequency of updating CP strategies. The frequency of the CP algorithm execution can rely on predefined duration or interval, or it can be dynamic to network health statistics and/or end-user perceived QoS parameters. These will be crucial in designing strategic, scalable and novel CP algorithms that can leverage the elasticity of the cloud, inherent to CCDN.

IX. CONCLUSION

By the end of the last millennia, the monthly global Internet traffic had grown to be in the order of petabytes [91], i.e. 10^{15}, which posed a great burden on the antiquated Internet infrastructure. Furthermore, the universally expected response time for end-users has reduced along time. Together, these posed a threat to the QoS for end-users and fueled capital for building and maintaining large-scale CDN infrastructure. In 2013, 36% of the global Internet traffic passed through CDN infrastructures [92].

However, the cost of building and maintaining CDN infrastructures is formidable and is passed onto the content providers. Therefore, building a CDN infrastructure and hosting in CDN is only economically possible for large-scale content providers. Moreover, we are now witnessing the monthly global Internet traffic in the order of exabytes ([93], [94]), that is, 10^{18} bytes. It is estimated that, by 2018, 57% of the global Internet traffic will pass through CDN infrastructures [92]. These unimaginable volumes of traffic and higher QoS requirements, such as lower response times, suggest that traditional CDNs infrastructures and content providers will benefit greatly from the seemingly abundant storage, processing and bandwidth resources of the cloud.

Cloud-based Content Delivery Networks (CCDNs) can greatly reduce CAPEX and OPEX for traditional CDN infrastructures by exploiting cloud datacenters for storage and bandwidth for content delivery. Furthermore, content providers of all sizes can cost-effectively build their own CDNs in the cloud or leverage emerging CCDNs for storing content in the cloud and delivering it via seemingly abundant bandwidth.

This survey has focused on content placement (CP) algorithms for the emerging CCDNs. Our contribution includes a set of well-motivated design criteria for CP algorithms in CCDNs. These design criteria can be decomposed into the requirements that are based on content dynamics and those that must abide by the requirements of the cloud model of CCDNs. We identify that CP algorithms must meet the requirements of content access patterns and those of dynamically changing popularity. Furthermore, the design criteria are extended to include resource provisioning, cost minimization and maximization of QoS and resource utilization. We deduce these criteria by scrutinizing content and its complex relationship with end-users since they are integral in designing effective content placement algorithms for CCDNs.

Next, we discuss and review state-of-the-art CP algorithms for CCDNs and evaluate them against our well-motivated design criteria. As our survey reveals, few CP algorithms are designed to meet the complex relationships exhibited by the inherent content that is hosted on CCDNs. However, the inclusion of content characteristics has shown improvement in performance over traditional caching and cache replacement strategy. Lastly, we summarize the practical implications and uncover research challenges in designing effective content placement algorithms for CCDNs.

REFERENCES

[1] F. Chen, K. Guo, J. Lin, and T. La Porta, "Intra-cloud lightning: Building CDNs in the cloud," in *Proc. IEEE INFOCOM*, Orlando, FL, USA, 2012, pp. 433–441.

[2] "Cisco global cloud index: Forecast and methodology 2015-2020," Cisco, San Jose, CA, USA, White Paper C11-738085, 2016.[Online]. Available: https://www.cisco.com/c/dam/en/us/solutions/collateral/service-provider/global-cloud-index-gci/white-paper-c11-738085.pdf

[3] *Cisco Open Media Distribution*, Cisco, San Jose, CA, USA, 2016.

[4] (2015). *Content Delivery Networks Explained*. Global Dots, accessed: Dec. 15 2016. [Online]. Available: http://www.globaldots.com/content-delivery-network-explained/

[5] J. Broberg, R. Buyya, and Z. Tari, "MetaCDN: Harnessing 'Storage Cloud's for high Perform. content delivery," *J. Netw. Comput. Appl.*, vol. 32, no. 5, pp. 1012–1022, 2009.

[6] M. Pathan and R. Buyya, "A taxonomy of CDNs," in *Content Delivery Networks* (Lecture Notes Electrical Engineering), Berlin, Germany: Springer, vol. 9. 2008, pp. 33–37. [Online]. Available: https://pdfs.semanticscholar.org/6ba0/658e77f3502a5f050b73d0cfbf2a571d0714.pdf

[7] H. Hu, Y. Wen, T. S. Chua, J. Huang, W. Zhu, and X. Li, "Joint content replication and request routing for social video distribution over cloud CDN: A community clustering method," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 7, pp. 1320–1333, Jul. 2016.

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

[9] G. Darzanos, I. Papafili, and G. D. Stamoulis, "A socially-aware ISP-friendly mechanism for efficient content delivery," in *Proc. Int Teletraffic Congr. (ITC)*, Karlskrona, Sweden, 2014, pp. 1–9.

[10] "Cisco visual networking index: Forecast and methodology, 2014–2019," Cisco Syst., San Jose, CA, USA, White Paper C11-481360, 2015. [Online]. Available: http://s2.q4cdn.com/230918913/files/doc_downloads/report_2014/white_paper_c11-481360.pdf

[11] J. Sahoo, M. A. Salahuddin, R. Glitho, H. Elbiaze, and W. Ajib, "A survey on replica server placement algorithms for content delivery networks," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 2, pp. 1002–1026, 2nd Quart., 2016.

[12] M. Al-Shayeji, S. Rajesh, M. Alsarraf, and R. Alsuwaid, "A comparative study on replica placement algorithms for content delivery networks," in *Proc. Int. Conf. Adv. Comput., Control Telecommun. Technol. (ACT)*, Jakarta, Indonesia, 2010, pp. 140–142.

[13] S. U. Khan and I. Ahmad, "Comparison and analysis of ten static heuristics-based Internet data replication techniques," *Parallel Distrib. Comput.*, vol. 68, no. 2, pp. 113–136, 2008.

[14] M. R. Korupolu, C. G. Plaxton, and R. Rajaraman, "Placement algorithms for hierarchial cooperative caching," *J. Algorithms*, vol. 38, no. 1, pp. 260–302, 2001.

[15] A. Leff, J. L. Wolf, and P. S. Yu, "Replication algorithms in a remote caching architecture," *IEEE Trans. Parallel Distrib. Syst.*, vol. 4, no. 11, pp. 1185–1204, Nov. 1993.

[16] M. Wang *et al.*, "An overview of cloud based content delivery networks: Research dimensions and state-of-the-art,"*Transactions on Large-Scale Data- and Knowledge-Centered Systems XX* (Lecture Notes in Computer Science), vol. 9070, A. Hameurlain, J. Küng, R. Wagner, S. Sakr, L. Wang, and A. Zomaya, Eds. Berlin, Germany: Springer, 2015. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-662-46703-9_6

[17] V. K. Adhikari, S. Jain, Y. Chen, and Z. L. Zhang, "Vivisecting YouTube: An active measurement study," in *Proc. IEEE Int. Conf. Comput. Commun. (INFOCOM)*, Orlando, FL, USA, Mar. 2012, pp. 2521–2525.

[18] A. Passarella, "A survey on content-centric technologies for the current Internet: CDN and P2P solutions," *Comput. Commun.*, vol. 35, no. 1, pp. 1–32, 2012.

[19] M. Pathan, R. Buyya, and A. Vakali, "Content delivery networks: State of the art, insights, and imperatives," in *Content Delivery Networks*. Berlin, Germany: Springer-Verlag, 2008, pp. 3–31.

[20] W. Ma, B. Shen, and J. Brassil, "Content services network: The architecture and protocols," in *Proc. Int. Workshop Web Caching Content Distrib.*, 2001, pp. 83–101.

[21] K. Li, H. Shen, F. Y. L. Chin, and W. Zhang, "Multimedia object placement for transparent data replication," *IEEE Trans. Parallel Distrib. Syst.*, vol. 18, no. 2, pp. 212–224, Feb. 2007.

[22] M. Karlsson and M. Mahalingam, "Do we need replica placement algorithms in content delivery networks?" in *Proc. Int. Workshop Web Content Caching Distrib. (WCW)*, Boulder, Colorado, 2002.

[23] G. Pallis, A. Vakali, K. Stamos, A. Sidiropoulos, D. Katsaros, and Y. Manolopoulos, "A latency-based object placement approach in content distribution networks," in *Proc. 3rd Latin Amer. Web Congr. (LA-WEB)*, Buenos Aires, Argentina, Nov. 2005, p. 8.

[24] S. Moharir, J. Ghaderi, S. Sanghavi, and S. Shakkottai, "Serving content with unknown demand: The high-dimensional regime," *ACM SIGMETRICS*, vol. 42, no. 1, pp. 435–447, 2014.

[25] N. Laoutaris, V. Zissimopoulos, and I. Stavrakakis, "On the optimization of storage capacity allocation for content distribution," *Comput. Netw.*, vol. 47, no. 3, pp. 409–428, 2005.

[26] X. Jia, D. Li, H. Du, and J. Cao, "On optimal replication of data object at hierarchical and transparent Web proxies," *IEEE Trans. Parallel Distrib. Syst.*, vol. 16, no. 8, pp. 673–685, Aug. 2005.

[27] X. Guan and B.-Y. Choi, "Push or pull? Toward optimal content delivery using cloud storage," *J. Netw. Comput. Appl.*, vol. 40, pp. 234–243, Apr. 2014.

[28] 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. 2014.

[29] Y. Jin, Y. Wen, K. Guan, D. Kilper, and H. Xie, "Toward monetary cost effective content placement in cloud centric media network," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, San Jose, CA, USA, 2013, pp. 1–6.

[30] H. Hu *et al.*, "Community based effective social video contents placement in cloud centric CDN network," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Chengdu, China, Jul. 2014, pp. 1–6.

[31] 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.

[32] X. Tang and S. T. Chanson, "Coordinated en-route Web caching," *IEEE Trans. Comput.*, vol. 51, no. 6, pp. 595–607, Jun. 2002.

[33] A. Jiang and J. Bruck, "Optimal content placement for en-route Web caching," in *Proc. 2nd IEEE Int. Symp. Netw. Comput. Appl. (NCA)*, Cambridge, MA, USA, 2003, pp. 9–16.

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

[35] I. Baev, R. Rajaraman, and C. Swamy, "Approximation algorithms for data placement problems," *SIAM J. Comput.*, vol. 38, no. 4, pp. 1411–1429, 2008.

[36] C. Papagianni, A. Leivadeas, and S. Papavassiliou, "A cloud-oriented content delivery network paradigm: Modeling and assessment," *IEEE Trans. Dependable Secure Computing*, vol. 10, no. 5, pp. 287–300, Sep. 2013.

[37] I. Cidon, S. Kutten, and R. Soffer, "Optimal allocation of electronic content," in *Proc. 20th Annu. Joint Conf. IEEE Comput. Commun. Soc. (INFOCOM)*, Anchorage, AK, USA, Oct. 2001, pp. 205–218.

[38] X. Tang and S. T. Chanson, "Analysis of replica placement under expiration-based consistency management," *IEEE Trans. Parallel Distrib. Syst.*, vol. 17, no. 11, pp. 1253–1263, Nov. 2006.

[39] M. A. Salahuddin, A. Al-Fuqaha, and M. Guizani, "Software-defined networking for RSU clouds in support of the internet of vehicles," *IEEE Internet Things J.*, vol. 2, no. 2, pp. 133–144, Apr. 2015.

[40] F. Silva, A. Boukerche, T. R. Silva, L. B. Ruiz, E. Cerqueira, and A. A. F. Loureiro, "Content replication and delivery in vehicular networks," in *Proc. Develop. Anal. Intell. Veh. Netw. Appl. (DIVANet)*, Montreal, QC, Canada, 2014, pp. 127–132.

[41] S. Zaman and D. Grosu, "A distributed algorithm for the replica placement problem," *IEEE Trans. Parallel Distrib. Syst.*, vol. 22, no. 9, pp. 1455–1468, Sep. 2011.

[42] S. Khan, A. Maciejewski, and H. Siegel, "Robust CDN replica placement techniques," in *Proc. IEEE Int. Symp. Parallel Distrib. Process. (IPDPS)*, Rome, Italy, 2009, pp. 1455–1468.

[43] S. Borst, V. Gupta, and A. Walid, "Distributed caching algorithms for content distribution networks," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, San Diego, CA, Mar. 2010, pp. 1–9.

[44] R. Zhou, S. Khemmarat, and L. Gao, "The impact of YouTube recommendation system on video views," in *Proc. 10th ACM SIGCOMM Conf. Internet Meas. (IMC)*, Melbourne, VIC, Australia, 2010, pp. 404–410.

[45] A. Finamore, M. Mellia, M. M. Munafò, R. Torres, and S. G. Rao, "YouTube everywhere: Impact of device and infrastructure synergies on user experience," in *Proc. ACM SIGCOMM Conf. Internet Meas. Conf. (IMC)*, Berlin, Germany, 2011, pp. 345–360.

[46] A. Brodersen, S. Scellato, and M. Wattenhofer, "YouTube around the world: Geographic popularity of videos," in *Proc. 21st Int. Conf. World Wide Web (WWW)*, Lyon, France, 2012, pp. 241–250.

[47] A. Tatar, P. Antoniadis, and M. D. De Dia, "Ranking news articles based on popularity prediction," in *Proc. Int. Conf. Adv. Social Netw. Anal. Mining (ASONAM)*, Istanbul, Turkey, 2012, pp. 106–110.

[48] E. B. Abdelkrim, M. A. Salahuddin, H. Elbiaze, and R. Glitho, "A hybrid regression model for video popularity-based cache replacement in content delivery networks," in *Proc. Global Commun. Conf. (GLOBECOMM)*, Washington, DC, USA, 2016, pp. 1–7.

[49] Y. Zhou, L. Chen, C. Yang, and D. M. Chiu, "Video popularity dynamics and its implication for replication," *IEEE Trans. Multimedia*, vol. 17, no. 8, pp. 1273–1285, Aug. 2015.

[50] P. Casas, A. D'Alconzo, P. Fiadino, A. Bär, A. Finamore, and T. Zseby, "When YouTube does not work—Analysis of QoE-relevant degradation in Google CDN traffic," *IEEE Trans. Netw. Service Manage.*, vol. 11, no. 4, pp. 441–457, Dec. 2014.

[51] M. E. Crovella, M. S. Taqqu, and A. Bestavros, "Heavy-tailed probability distributions in the world wide Web," in *A Practical Guide to Heavy Tails*. Cambridge, MA, USA: Birkhäuser Boston, 1998, pp. 3–25.

[52] B. E. Brewington and G. Cybenko, "How dynamic is the Web?" *Int. J. Comput. Telecommun. Netw.*, vol. 33, nos. 1–6, pp. 257–276, 2000.

[53] V. N. Padmanabhan and L. Qiu, "The content and access dynamics of a busy Web site: Findings and implications," in *Proc. Conf. Appl. Technol., Archit., Protocols Comput. Commun. (SIGCOMM)*, New York, NY, USA, 2000, pp. 111–123.

[54] W. Shi, R. Wright, E. Collins, and V. Karamcheti, "Workload characterization of a personalized Web site—And its implications for dynamic content caching," New York Univ., New York, NY, USA, Tech. Rep. TR2002-829, 2002.

[55] W. Shi, E. Collins, and V. Karamcheti, "Modeling object characteristics of dynamic Web content," *J. Parallel Distrib. Comput.*, vol. 63, no. 10, pp. 963–980, 2003.

[56] M. E. Crovella and A. Bestavros, "Self-similarity in world wide Web traffic: Evidence and possible causes," *IEEE/ACM Trans. Netw.*, vol. 5, no. 6, pp. 835–846, Dec. 1997.

[57] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker, "Web caching and Zipf-like distributions: Evidence and implications," in *Proc. Conf. Comput. Commun. (INFOCOM)*, New York, NY, USA, 1999, pp. 126–134.

[58] F. Guillemin, B. Kauffmann, S. Moteau, and A. Simon, "Experimental analysis of caching efficiency for YouTube traffic in an ISP network," in *Proc. Teletraffic Congr. (ITC)*, Shanghai, China, 2013, pp. 1–9.

[59] M. Leconte, M. Lelarge, and L. Massoulié, "Designing adaptive replication schemes in distributed content delivery networks," in *Proc. Int. Teletraffic Congr. (ITC)*, Ghent, Belgium, 2015, pp. 28–36.

[60] D. Krishnaswamy, R. Krishnan, D. Lopez, P. Willis, and A. Qamar, "An open NFV and cloud architectural framework for managing application virality behaviour," in *Proc. IEEE Consum. Commun. Netw. Conf. (CCNC)*, Las Vegas, NV, USA, 2015, pp. 746–754.

[61] I. Kilanioti, "Improving multimedia content delivery via augmentation with social information: The social prefetcher approach," *IEEE Trans. Multimedia*, vol. 17, no. 9, pp. 1460–1470, Sep. 2015.

[62] I. Kilanioti, C. Georgiou, and G. Pallis, "On the impact of online social networks in content delivery," in *Advanced Content Delivery and Streaming in the Cloud*, M. Pathan, R. Sitaraman, and D. Robinson, Eds. Hoboken, NJ, USA: Wiley, 2013.

[63] A. Mahanti, N. Carlsson, A. Mahanti, M. Arlitt, and C. Williamson, "A tale of the tails: Power-laws in Internet measurements," *IEEE Netw.*, vol. 27, no. 1, pp. 59–64, Jan. 2013.

[64] 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.

[65] J. Chandrakanth, P. Chollangi, and C. H. Lung, "Content distribution networks using software defined networks," in *Proc. IEEE Int. Conf. Trustworthy Syst. Their Appl. (TSA)*, Hualien, Taiwan, Jul. 2015, pp. 44–50.

[66] J. Llorca, C. Sterle, A. Tulino, N. Choi, A. Sforza, and A. E. Amideo, "Joint content-resource allocation in software defined virtual CDNs," in *Proc. IEEE Int. Conf. Commun. Workshop (ICCW)*, London, U.K., Jun. 2015, pp. 1839–1844.

[67] Z. Li and G. Simon, "In a telco-CDN, pushing content makes sense," *IEEE Trans. Netw. Service Manage.*, vol. 10, no. 3, pp. 300–311, Sep. 2013.

[68] *Amazon CloudFront—Content Delivery Network (CDN)*. Accessed: Dec. 15, 2016. [Online]. Available: https://aws.amazon.com/cloudfront/

[69] *Google Cloud CDN—Low Latency Content Delivery|Google Cloud Platform*. Accessed: Dec. 15, 2016. [Online]. Available: https://cloud.google.com/cdn/

[70] H. Li, L. Zhong, J. Liu, B. Li, and K. Xu, "Cost-effective partial migration of VoD services to content clouds," in *Proc. IEEE Int. Conf. Cloud Comput. (CLOUD)*, Washington, DC, USA, Jul. 2011, pp. 203–210.

[71] G. Silvestre, S. Monnet, R. Krishnaswamy, and P. Sens, "AREN: A popularity aware replication scheme for cloud storage," in *Proc. IEEE Int. Conf. Parallel Distrib. Syst. (ICPADS)*, Singapore, Dec. 2012, pp. 189–196.

[72] (Oct. 20, 2016). *Google Cloud Storage Pricing|Cloud Storage Documentation|Google Cloud Platform*. Google, Inc., accessed: Jan. 15, 2017. [Online]. Available: https://cloud.google.com/storage/pricing

[73] M. Armbrust *et al.*, "A view of cloud computing," *Commun. ACM*, vol. 53, no. 4, pp. 50–58, 2010.

[74] M. Seufert, S. Egger, M. Slanina, T. Zinner, T. Hoßfeld, and P. Tran-Gia, "A survey on quality of experience of HTTP adaptive streaming," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 1, pp. 469–492, 1st Quart., 2015.

[75] M. Hu, J. Luo, Y. Wang, and B. Veeravalli, "Practical resource provisioning and caching with dynamic resilience for cloud-based content distribution networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 8, pp. 2169–2179, Aug. 2014.

[76] K. Katsalis, V. Sourlas, T. Korakis, and L. Tassiulas, "A cloud-based content replication framework over multi-domain environments," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Sydney, NSW, Australia, Jun. 2014, pp. 2926–2931.

[77] A. Rappaport and D. Raz, "Update aware replica placement," in *Proc. Int. Conf. Netw. Service Manag. (CNSM)*, Zürich, Switzerland, Oct. 2013, pp. 92–99.

[78] M. Salahuddin, H. Elbiaze, W. Ajib, and R. Glitho, "Social network analysis inspired content placement with QoS in cloud-based content delivery networks," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, San Diego, CA, USA, Dec. 2015, pp. 1–6.

[79] N. Carlsson, D. Eager, A. Gopinathan, and Z. Li, "Caching and optimized request routing in cloud-based content delivery systems," *J. Perform. Eval.*, vol. 79, pp. 38–55, Sep. 2014.

[80] X. Tang, H. Chi, and S. T. Chanson, "Optimal replica placement under TTL-based consistency," *IEEE Trans. Parallel Distrib. Syst.*, vol. 18, no. 3, pp. 351–363, Mar. 2007.

[81] A. Araldo, M. Mangili, F. Martignon, and D. Rossi, "Cost-aware caching: Optimizing cache provisioning and object placement in ICN," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Austin, TX, USA, Dec. 2014, pp. 1108–1113.

[82] J. Xu, B. Li, and D. L. Lee, "Placement problems for transparent data replication proxy services," *IEEE J. Sel. Areas Commun.*, vol. 20, no. 7, pp. 1383–1398, Sep. 2002.

[83] L. Zhuo, C.-L. Wang, and F. Lau, "Load balancing in distributed Web server systems with partial document replication," in *Proc. Int. Conf. Parallel Process.*, Vancouver, BC, Canada, 2002, pp. 305–312.

[84] S. R. Srinivasan, J. W. Lee, D. Batni, and H. Schulzrinne, "ActiveCDN: Cloud computing meets content delivery networks," Dept. Comput. Sci., Columbia Univ. New York, NY, USA, Tech. Rep., 2011.

[85] L. Zeng *et al.*, "Monetary-and-QoS aware replica placements in cloud-based storage systems," in *Proc. IEEE CloudCom*, Singapore, Dec. 2014, pp. 672–675.

[86] L. Qiu, V. Padmanabhan, and G. Voelker, "On the placement of Web server replicas," in *Proc. Annu. Joint Conf. IEEE Comput. Commun. Soc. (INFOCOM)*, Anchorage, AK, USA, Apr. 2001, pp. 1587–1596. [Online]. Available: http://ieeexplore.ieee.org/document/916655/

[87] W. Jiang, S. Ioannidis, L. Massoulié, and F. Picconi, "Orchestrating massively distributed CDNs," in *Proc. Int. Conf. Emerg. Netw. Experim. Technol. (CoNEXT)*, Nice, France, 2012, pp. 133–144.

[88] K. Poularakis and L. Tassiulas, "Optimal cooperative content placement algorithms in hierarchical cache topologies," in *Proc. Annu. Conf. Inf. Sci. Syst. (CISS)*, Princeton, NJ, USA, 2012, pp. 1–6.

[89] M. Dehghan, A. Seetharam, B. Jiang, T. He, and T. Salonidis, "On the complexity of optimal routing and content caching in heterogeneous networks," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Kowloon, Hong Kong, Apr./May 2015, pp. 936–944.

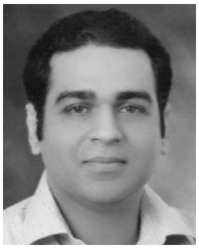
[90] N. Golrezaei, K. Shanmugam, A. G. Dimakis, A. F. Molisch, and G. Caire, "FemtoCaching: Wireless video content delivery through distributed caching helpers," in *Proc. IEEE INFOCOM*, Orlando, FL, USA, Mar. 2012, pp. 1107–1115.

[91] A. Sumits. (Aug. 28, 2015). *History and Future of Internet Traffic*. Cisco Blogs, accessed: Dec. 1, 2016. [Online]. Available: http://blogs.cisco.com/sp/the-history-and-future-of-internet-traffic

[92] Y. Shen. (Sep. 23, 2014). *The Shift to Content Delivery Networks (CDNs) Supports More and Better Customer Video Experiences*. Cisco Blogs, accessed: Dec. 1, 2016. [Online]. Available: http://blogs.cisco.com/sp/the-shift-to-content-delivery-networks-cdns-supports-more-and-better-customer-video-experiences

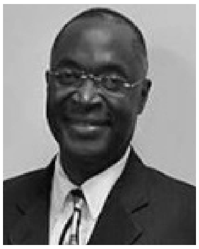
[93] C. Systems. *VNI Forecast Highlights*. Cisco Systems, accessed: Nov. 17, 2016. [Online]. Available: http://www.cisco.com/web/solutions/sp/vni/vni_forecast_highlights/index.html

[94] M. Savage, "Internet traffic will hit 2 Zettabytes By 2019, Cisco Says. Information week—Network computing," Tech. Rep., 2015. [Online]. Available: http://www.networkcomputing.com/networking/internet-traffic-will-hit-2-zettabytes-2019-cisco-says/428421006

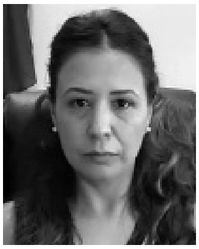


MOHAMMAD A. SALAHUDDIN (S'09–M'15) received the Ph.D. degree in computer science from Western Michigan University, Kalamazoo, MI, USA, in 2014. He is currently a Post-Doctoral Fellow with the David R. Cheriton School of Computer Science, University of Waterloo, Waterloo, ON, Canada. His research interests include wireless sensor networks, QoS and QoE in vehicular ad hoc networks (WAVE, IEEE 802.11p, and IEEE 1609.4), Internet of Things, content delivery networks, software-defined networking, network functions virtualization, and cloud resource management. He serves as a Technical Program Committee Member of international conferences and a reviewer of various peer-reviewed journals, magazines, and conferences.

JAGRUTI SAHOO (M'13) received the Ph.D. degree in computer science and information engineering from the National Central University, Taiwan, 2013. She has been a Post-Doctoral Fellow with the University of Sherbrooke, Canada, and Concordia University, Canada. She is currently an Assistant Professor with the Department of Mathematics and Computer Science, South Carolina State University, USA. Her research interests include wireless sensor networks, vehicular networks, content delivery networks, cloud computing, and network functions virtualizations. She has served as a member of the Technical Program Committee of many conferences and a reviewer of many journals and conferences.



ROCH GLITHO (M'88–SM'97) received the M.Sc. degree in business economics from the University of Grenoble, France, the M.Sc. degree in pure mathematics from University Geneva, Switzerland, and the M.Sc. degree in computer science from the University of Geneva, and the Ph.D. degree (Tekn.Dr.) in tele-informatics from the Royal Institute of Technology, Stockholm, Sweden. He is an Associate Professor of networking and telecommunications with CIISE, Montreal, Canada, where he leads the Telecommunication Service Engineering Research Laboratory. With experience in industry for almost a quarter of a century, he has held several senior technical positions with LM Ericsson, Sweden and Canada, as an Expert, a Principal Engineer, and a Senior Specialist. His industrial experience includes research, international standards setting, such as contributions to ITU-T, ETSI, TMF, ANSI, TIA, and 3GPP, product management, project management, systems engineering, and software/firmware design. He has been the IEEE Communications Society Distinguished Lecturer, an Editor-In-Chief of the IEEE Communications Magazine and an Editor-In-Chief of the IEEE Communications Surveys and Tutorials.



HALIMA ELBIAZE (M'06) received the B.Sc. degree in applied mathematics from the University of MV, Morocco, in 1996, the M.Sc. degree in telecommunication systems from the Université de Versailles in 1998, and the Ph.D. degree in computer science from the Institut National des Télécommunications, Paris, France, in 2002. She is currently an Associate Professor with the Department of Computer Science, Université du Québec à Montréal, QC, Canada, where has been serving since 2003. She has authored and co-authored many journal and conference papers. Her research interests include network performance evaluation, traffic engineering, and quality of service management in optical and wireless networks.

WESSAM AJIB (M'05–SM'16) received the Engineering Diploma in physical instruments from INPG, Grenoble, France, 1996, and the master's and Ph.D. degrees in computer networks from École Nationale Supérieure des Télécommunication, Paris, in 1997 and 2000, respectively. He had been an Architect and Radio Network Designer with Nortel Networks, Ottawa, ON, Canada, from 2000 to 2004. He has conducted many projects on the third generation of wireless cellular networks. He was a Post-Doctoral Fellow with the Electrical Engineering Department, École Polytechnique de Montréal, QC, Canada, from 2004 to 2005. He is currently a Full Professor with the Department of Computer Sciences, Universite du Quebec at Montreal, QC, Canada, where he has been serving since 2005. He has authored and co-authored many journal papers and conferences papers in these areas. His research interests include wireless communications and networks, multiple access design and traffic scheduling.