CPCDN: Content Delivery Powered by Context and User Intelligence

Zhi Wang, *Member, IEEE,* Wenwu Zhu, *Fellow, IEEE,* Minghua Chen, *Member, IEEE,* Lifeng Sun, *Member, IEEE,* and Shiqiang Yang, *Senior Member, IEEE*

Abstract—There is an unprecedented trend that content providers (CPs) are building their own content delivery networks (CDNs) to provide a variety of content services to their users. By exploiting powerful CP-level information in content distribution, these CP-built CDNs open up a whole new design space and are changing the content delivery landscape. In this paper, we adopt a measurement-based approach to understanding *why, how, and how much CP-level intelligences can help content delivery*. We first present a measurement study of the CDN built by Tencent, a largest content provider based in China. We observe new characteristics and trends in content delivery which pose great challenges to the conventional content delivery paradigm and motivate the proposal of *CPCDN*, a CDN powered by CP-aware information. We then reveal the benefits obtained by exploiting two indispensable CP-level intelligences, namely *context intelligence* and *user intelligence*, in content delivery. Inspired by the insights learnt from the measurement studies, we systematically explore the design space of CPCDN and present the novel architecture and algorithms to address the new content delivery challenges arisen. Our results not only demonstrate the potential of CPCDN in pushing content delivery performance to the next level, but also identify new research problems calling for further investigation.

1 Introduction

There has been a great trend that large multimedia content providers (CPs) are building and customizing their own content delivery networks to provide content services to their users (throughout this paper, we denote the term *CPCDN*, a CDN powered by CP-level intelligences, as our approach for content providers to optimize content delivery, and use *CDN* to represent a conventional content delivery network). Though having the potential to improve the quality of content delivery for a variety of content providers, the delivery strategies in such CPCDNs have not been thoroughly studied. In this paper, we focus on the design space allowed by CPCDN, and present representative delivery strategies that can improve the content delivery quality.

Today's internet has witnessed a rapid emergence of customized CDNs, e.g., YouTube has long had its Google Global Cache for video delivery. Netflix delivers streaming data by directly collaborating with Internet Service Providers (ISPs) using its own content delivery framework — OpenConnect [26]. Facebook is boosting its edge network in the Open Computing Project [13]. Tencent, one of the largest online content service providers [3], is delivering over 70% of its traffic using its own CDN. Based on the elastic storage and network resources provided by edge cloud service providers like Amazon CloudFront [12], even small Internet content providers can customize their own CDNs [11]. The move to CPCDN is driven by the following new characteristics and trends in content delivery, which pose great challenges to conventional content delivery paradigm.

First, it is becoming popular for content providers to "produce" contents in a realtime manner for different customized *contexts*, e.g., images and videos are dynamically processed for different user devices and network conditions. Since such content production for different contexts is carried out in an online and realtime manner, conventional CDN is unsuitable of ensuring good user experience, due to the inherent decoupling between between CP and CDN. In contrast, knowing the exact context under which contents are consumed, CPCDN is able to exploit *context intelligence* capturing how contents are dynamically processed and synthesized in different contexts, to optimize the user experience and service quality along content delivery.

Second, in the content eco-system, users are no longer just consumers but have already become indispensable *participants* in content generation and distribution [27]. The popular music video "Gangnam Style" as an example: it was a user-generated video and became very popular in 2012 and 2013, as a result of several influential celebrities sharing it on online social networks such as Twitter [31]. Being oblivious to user participation, it is difficult, if not impossible, for conventional CDNs to leverage the inherent user-content preference pattern and user-user social influence to optimize user

Z. Wang is with the Graduate School at Shenzhen, Tsinghua University, Shenzhen, China (email: wangzhi@sz.tsinghua.edu.cn). W. Zhu, L. Sun and S. Yang are with Tsinghua National Laboratory for Information Science and Technology, Department of Computer Science and Technology, Tsinghua University, Beijing, China (e-mail: {wwzhu, sunlf, yangshq}@tsinghua.edu.cn). M. Chen is with the Department of Information Engineering, The Chinese University of Hong Kong, Hong Kong (e-mail: minghua@ie.cuhk.edu.hk).

experience and service quality. Leveraging the powerful *user intelligence* capturing how contents are dynamically shared and distributed among social-networked users, CPCDN is able to improve both the content delivery efficiency and user experience to the next level.

In this paper, to accompany the rapid move to CPCDN, we apply a measurement-based approach to understand *why, how and how much CP-level intelligence can help content delivery*. In particular, our contributions are summarized as follows.

▷ To investigate why CP-level intelligence can help content delivery, we present measurement studies on popular services of Tencent, a representative content provider, whose CPCDN was serving over 2 Tbps traffic of contents from all its services at peak hours in 2013. New characteristics in its content delivery are observed as follows. (1) *Delivery context*. We observe that contents are no longer delivered individually, instead, they are synthesized into various contexts. (2) *Crowd pattern*. According to our measurement study, there are patterns about how users consume contents. In particular, there exist inherent patterns that particular clusters of users will be interested in particular groups of contents. (3) *Social influence*. Our measurement study also shows that certain social behaviors (e.g., sharing and resharing) among online social-network users have great impact on whether contents will become popular, and if so, among which group of users.

▷ To understand how CP-level intelligence can help content delivery and to extract its potential, we systematically explore the design space of CPCDN and present the architecture and algorithms to address the new challenges arisen, using both the context intelligence and user intelligence. (1) Exploiting the fact that contents have diverse importance under different contexts, we propose to prioritize the content delivery according to the context-aware content importance, by opportunistically selecting peer servers and queuing the requests. (2) Based on the crowd patterns, we proactively replicate contents according to the interest groups so that users can fetch the contents they are interested in from nearby CPCDN servers. (3) Based on the social influence of content distribution, we design a new popularity inference strategy using the social relationship and social activity of users, so as to guide the bandwidth reservation for contents becoming increasingly popular due to social influence.

▷ To reveal how much CP-level intelligence can improve the content delivery, we carry out measurement-based studies and extensive empirical evaluation using real-world traces to confirm the benefits of CPCDN in improving content delivery efficiency and user experience, as compared to conventional content delivery paradigms.

The rest of the paper is organized as follows. We present the motivation and framework of CPCDN in Sec. 2. We present measurement studies to guide the design of strategies in CPCDN in Sec. 3. We present the design space of CPCDN in Sec. 4. We use measurement and simulation-based experiments to evaluate the performance of a CPCDN in Sec. 5. We survey related works in Sec. 6. Finally we conclude the paper in Sec. 7.

2 Motivation and Framework

In this section, we present the change of today's content delivery motivating content providers to build their own CPCDNs and the framework of a CPCDN, where content context and user information can be utilized for more intelligent content delivery.

2.1 Changes in Today's Content Delivery

In traditional content delivery, after being produced by the content provider, contents are delivered directly to a group of users via a CDN, which serves as a connectivity "pipe" oblivious to the context and user activity under which contents are consumed, since users are merely receivers at the end of the content flow — there is no feedback information from users to the CDN.

In contrast, in today's content consumption flow as illustrated in Fig. 1, users generate contents [8], provide specification to content providers to synthesize contents into the right contexts [27], and affect how the contents are consumed among users [30]. For example, on Facebook, a user's homepage contains contents generated by different people (*e.g.*, her friends), and is synthesized according to the user's social connections, and contexts (*e.g.*, accessing device, time, and geo-location). These changes have challenged the conventional content delivery paradigm. Specifically, we present how today's content delivery is affected by delivery contexts and users as follows:

C-1 **Content delivery is context-aware**: After produced by the content provider or a user, a content (e.g., an image upload by a user) will be synthesized (e.g., composed in a webpage) with other related contents (e.g., blogs) to be finally provided to a user in a given context. Our measurement studies will show that it becomes a norm rather than an exception, that contents are synthesized differently in different contexts.

C-2 **Content delivery is user-aware**: Rather than the passive receivers of the contents, today, users become important content producers [8] and propagators [9]. User activity and social propagation have greatly changed how contents are consumed by people [34] — users are involved in the content delivery from content generation to content distribution.

The idea of CPCDN can significantly improve conventional CDNs for context-aware and social content delivery. If CDNs can collect information about users and contexts from CPs, they will be able to carry out the strategies proposed in this paper for improving content delivery as well. However, such incorporation is sometimes inadequate since the user and context information

is usually protected by CPs. Thus, in this paper, we only focus on a CPCDN framework that CP and CDN are closely coupled, in which user and context information can be used in a realtime manner.

2.2 CPCDN Framework

Since most of the context and user information is only available to the content providers in the content flow, it is clear that CP can develop intelligent strategies out of the big amount of data to guide content delivery. Being aware of i) how contents are generated and aggregated, ii) how they are processed and synthesized in different contexts, and iii) how they propagate among social-networked users, the CPCDN is able to utilize the context intelligence and user intelligence to not only optimize the under-layer content delivery strategies, but also improve the understanding of upper-layer content characteristics, as illustrated in Fig. 1.

However, in a CPCDN design, we still need to address the new challenges arisen as follows: (1) How to define the delivery contexts and make use of them; (2) How to measure the impact of users on content delivery, and utilize information from online social networks to improve content delivery. In particular, we have to tackle to following research problems to realize CPCDN in a large-scale and dynamically-evolving system:

- Mining the content delivery contexts, *e.g.*, identifying a content item's importance based on how it is requested by users;
- Improving content delivery based on the crowd patterns from users' actions on contents;
- Using the social influence for better social/socialized content delivery.

To address these problems, we propose CPCDN delivery strategies based on context and user intelligences. On one hand, we design the content replication and the user request schedule strategies by knowing the importance of contents when they are delivered in different contexts; on the other hand, we let the CPCDN predict the popularity of contents that are shared among social-networked users, for better network resource allocation.

An objective of this paper is to show that there are indeed potential benefits in exploiting CP-level intelligence in content delivery. Next, we will present our measurement studies shedding insights on potential of a CPCDN.

3 A Trace-Driven Measurement Study of CPCDN

In this section, to establish a concrete understanding on the motivations behind the current move to CPCDN, we present a large-scale measurement study on popular services of one of the largest Chinese content providers, Tencent, which provides popular online content services including web services, online social network services, and online video streaming.

3.1 Representative Content Services

We have used the following representative content services to perform measurement studies of a CPCDN.

WWW service: Since invented over 20 years ago, the great success of WWW has made web objects the dominant form of content delivery, and it is becoming a new trend that webpages are dynamically generated [6], i.e., the same content component (component for short) can be composed into different webpages, while a webpage contains up to hundreds of multimedia components. These components are dynamically organized and delivered in different contexts, e.g., the webpage under the same URL composes different components when it is accessed by different users at different times.

Online social network service: The online social network service has been increasingly popular in recent years [17]. In an online social network, contents are generated by users and propagate through the social connections [20]. It has changed content delivery in a sense that social relationship and user behavior will influence which contents will be viewed by which users.

Online video streaming: The traffic of online video services still dominates the Internet bandwidth. HTTP-based streaming (e.g., DASH) that takes full advantage of the CDN deployment, has emerged as a promising approach to provide high quality-of-experience to Internet users. In such video streaming, videos are served as "chunks" by CDNs. How to effectively handle context- and social-aware video content delivery is a challenge to today's CDNs.

3.2 Measurement Setup

We conduct a *trace-driven* measurement study of content delivery for the above content services in Tencent, which has deployed tens of thousands of servers spanning major ISPs in China including China Telecom, China Unicom and China Mobile, and several overseas ISPs. In 2012, Tencent CPCDN serves over 70% of the whole company's content delivery traffic. Our measurement studies are based on a set of different types of traces provided by Tencent, covering Sec. 3.1.

$\triangleright$ To study the context awareness in content delivery (i.e., C-1 in Sec. 2), we use traces from WWW services in Tencent. In particular, we collected traces from Tencent's portal website [3], which provides webpages with a variety of multimedia contents updated frequently. Our traces contain the structures of these webpages (i.e., how a webpage is composed with different components) at different times. Besides, to show that knowing the delivery context can improve content delivery strategies, we have also collected records of over 3.39 billion TCP connections from peering servers located at 55 regions in May 2013. These TCP connections were established for users to download contents with sizes varying from tens of bytes to 4.8 GB. Each of the trace items contains the following information: the timestamp indicating when a

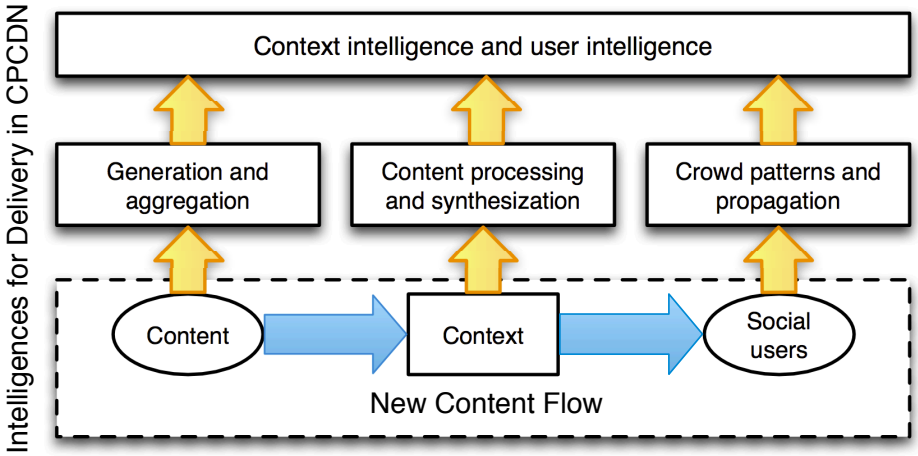


Fig. 1. CPCDN powered by intelligences mined from the new content flow.

TCP connection was established, the client IP, the number of downloaded bytes and the connection duration.

▷ To study the user awareness in content delivery (i.e., C-2 in Sec. 2), we have obtained traces from Tencent Weibo [2], a Twitter-like microblogging system in China. Some of the microblogs record how contents (e.g., videos) are shared between users, which can be used to study how users affect content delivery. We obtained Weibo traces containing the microblog items, including ID, name, IP address of the publisher, time stamp when the microblog is posted, IDs of the parent and root microbloggers if it is a re-post, and content of the microblog.

Next, we present the observations made.

3.3 Dynamic Delivery Context

It is of importance to provide swift webpage viewing experience to users, which is mainly determined by the delay between the time a user requests a webpage and the time the webpage is successfully rendered by the user's browser [37]. A large fraction of webpages today contain hundreds of components (including not only basic HTML codes, but also multimedia objects like images), which take a lot of time to download. In January 2013, the webpage of Tencent portal website contained over 140 components, whose size varied from 29 bytes to 168 KB, with an average size of 13.8 KB. Content providers dynamically compose dynamic components into a highly customized *context*, e.g., the webpages of Tencent portal are dynamically generated for individuals according to their device, preference, time and geo-location information. Semantic information, timeliness, interconnections and other impact factors can be considered to infer the content importance for specific application scenarios like general webpages, social networks, and online video streaming networks, e.g., the text information can be used to infer if the component is related to what the user has requested. It is challenging for traditional CDNs to deliver components efficiently, without knowing their importance to user experience in different contexts.

3.3.1 Heterogeneous Component Importance

We study the importance of components in different contexts. In our study, The context is inferred from a content component's size and position in webpages — a simple yet efficient context inference. An importance level of a component c in webpage w is calculated as follows $I(c,w) = z(c,w) \times t(c,w)$, where $z(c,w)$ is the normalized size of component c in webpage w, and $t(c,w)$ is the normalized height of the component c in webpage w, as illustrated in Fig. 2(a). The intuition of the importance level is that a component has a larger importance level if it is more visually important to the content provider and viewer. Though our approach is not limited to this definition, we present observations made based on this simple calculation. In our measurement, we have made the following observations of component importance: (1) Components in the same webpage have different importance levels. Only a few of them have a very large importance level, and their delivery performance determines the quality of experience for the webpage viewer. (2) The same component has different importance levels when it is composed in different webpages. It is clear that the same component has distinct importance level in different webpages, meaning that its delivery has a dynamic priority for different contexts.

3.3.2 Potential of Component Prioritization

Since components are timely generated, CPCDN which knows the delivery contexts can improve the content delivery by prioritize the content delivery of different components. The prioritization of components enables the CPCDN to allow users to be redirected to download important contents from fast peering servers.

Based on our TCP traces of the Tencent CDN peering servers, we are able to measure the time for a user to

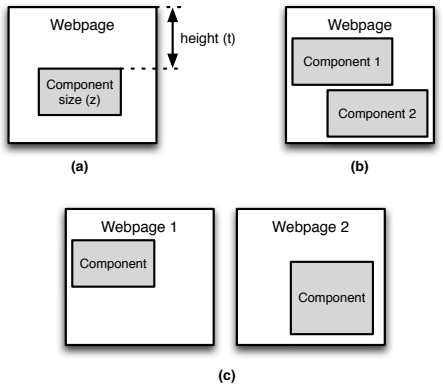


Fig. 2. Importance of components in webpages.

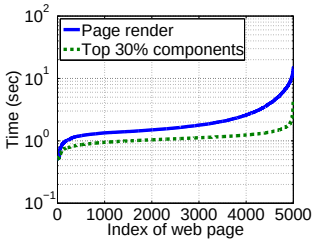


Fig. 3. Page render time versus top 30% components delivery time for subpages of the Tencent portal website.

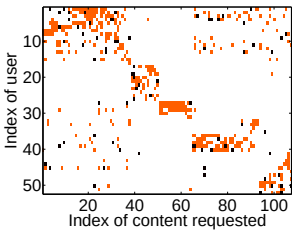


Fig. 4. New requests issued in the co-clustered user-content matrix.

download different components in webpages. In Fig. 3, the two curves are the render time (i.e., the delivery time of the first 30% components in position), and the delivery time of the top 30% components that are downloaded faster than others, respectively. We observe that the page render time is $2-10$ times larger than the download time of the top 30% components in the page. The reason is that the browser fails to render a page due to the absence of some components un-downloaded (e.g., the framework HTML, the CSS style file, and critical multimedia contents). This observation indicates that we can reduce the page render time by strategically prioritizing the delivery of important contents.

In a CDN system, servers are deployed at different geographic locations and in different ISPs. As a result, the network performance of different servers can be different. Table 1 illustrates the average download speed (Kbps) experienced by the same group of users, when they downloaded from 7 CDN sites in Tencent CPCDN in one day. We observe that the average download speed varies from 250 Kbps to over 500 Kbps when users download from servers deployed in different sites. The observation indicates that the performance of peering servers is heterogeneous to a user — when a content is downloaded from different servers, the delivery delay varies significantly.

We further demonstrate users' preference of different

TABLE 1
Average download speed of peering servers deployed in different regions and ISPs on May 4, 2013 (Kbps).

	Beijing	Zhejiang	Guangdong	Shaanxi
Telecom	366.8	281.4	338.7	249.4
Unicom	512.2	–	462.8	–
Mobile	–	491.8	–	–

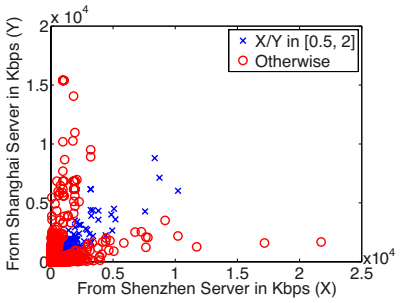


Fig. 5. Comparison of average download speeds of users downloading from different peering servers (May 4, 2013).

peering servers. Based on the TCP traces of the peering servers, we compare the download speeds of about 150 users who downloaded from different peering servers in the same 10 minutes on May 4, 2013. In Fig. 5, each sample is the average download speed of a user downloading from a peering server deployed in Shanghai, versus the average download speed of the same user downloading from a peering server in Shenzhen, both in the same ISP. We observe that for over 79% of the users, their download speeds differ over 2 times when they download from servers located at different regions, indicating that users have different preferences of peering servers.

In summary, we observe that web content delivery has the following characteristics that suggest a context-aware design: (1) Components are delivered in dynamic contexts, and have different importance to users when composed in different contexts; (2) Users have different preference of peering servers in a CDN, i.e., they download at different speeds from different servers; (3) Since components are timely composed into different contexts, CPCDN which knows the context information and controls user redirection is able to proactively prioritize component delivery. We will present the detailed design in Sec. 4.

3.4 Delivery Influenced by Users

Conventional content delivery is also oblivious to the impact of social propagation of contents, i.e., how downloads are driven by users' resharing contents in the online social network. Content delivery can be highly influenced by the social relationship (e.g., friending in Facebook, following in Twitter, *etc.*) and social behavior (e.g., sharing, commenting, *etc.*) [10]. Using such social

information from the online social network can fundamentally improve the delivery for social/socialized contents.

3.4.1 Crowd Pattern

Conventional CDN serves contents in a passive way, i.e., no historical information is used by the CDN when serving a content request. This is not efficient for today's content delivery, where users show predictable preference of contents. In today's content delivery, there exists interest patterns between a group of users and a cluster of contents, i.e., the crowd pattern. To illustrate the crowd pattern and its potential benefit in content delivery, we perform a case study on how users share videos in Tencent Weibo, in November 2011. We summarize the *user-content* activity information into a matrix (referred to as a user-content matrix in this paper) — an entry 1 (*resp.*, 0) indicates that the corresponding content is (*resp.*, is not) shared by the user. We have applied a co-clustering algorithm [15] to the matrix formed by 500 users randomly selected and the contents these users shared. The co-clustering results are illustrated in Fig. 4. Each yellow/light sample indicates that a content has been shared by a particular user. We observe that there are several user-content "clusters", where users in the same user cluster tend to request the contents in the corresponding content cluster. This observation suggests that users can be clustered into groups with similar interests. Furthermore, the black/dark samples are the shares issued by the same 500 users in 1 day, a week later after the co-clustering in the first round. We observe that most of the new samples are scattered in the ranges of the clusters based on the previous user-content co-clustering.

It is clear that there exists crowd patterns in today's content sharing. Knowing users' activities to contents, CPCDN is able to predict users' preference by mining these activities for efficient content replication.

3.4.2 Characteristics of Social Propagation

Next, we present the characteristics of social propagation, and that such characteristics are affecting content delivery.

▷ The life span of social contents is affected by user behaviors. Fig. 6 illustrates the life span of contents shared on Tencent Weibo. Each sample in this figure represents the number of reshares of a content versus the time lag between when the content is published (on Tencent Weibo) and when the content is reshared. We observe that most of the reshares are issued in early hours after a video is published, indicating that the newly published/shared contents are the ones that attract most of the users. The reason is that in today's online sharing paradigm, contents are not directly exposed to users to browse, instead, they reach users through social connections, who are more likely to pay attention to the most recent information [9].

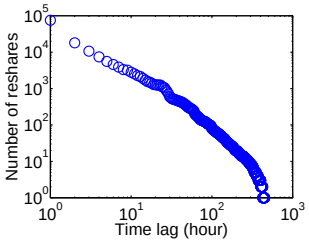


Fig. 6. Social content popularity versus the time lag after the content has been published.

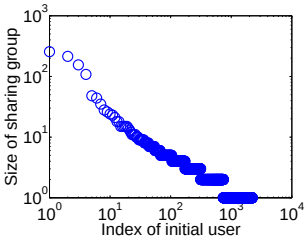


Fig. 7. The same content shared by different users ends up with different level of popularity.

▷ Change of popularity distribution of social contents. Fig. 7 illustrates how a content will attract people when it is shared by different users. In this experiment, the same video can be shared by different people to the online social network from time to time. Each sample in this figure represents the number of people who are attracted to reshare versus the index of the user who initially shares the content. We observe that when the same content is shared by different users, the number of receivers varies significantly.

As a matter of fact, it is no longer easy to perceive the content popularity in the traditional CDN paradigm, e.g., when the CDN notices that a content is becoming popular, it replicates the content to more peering servers. The effectiveness of the traditional strategy is based on the assumption that the popularity pattern of a content lasts for a relatively long time. However, this assumption is no longer true — the life span of contents is short and the popularity is dynamically affected by people in the online social network.

3.4.3 Social Popularity

Next, we present that it is possible to use the social influence (e.g., how many viewers can be attracted when a user shares a particular content in the online social network) to learn the social content popularity.

▷ *Influence and social connections*. We first study the correlation between a user's social influence and her social connections. Fig. 8 illustrates the number of all reshares attracted by a user versus the number of her followers. We observe that there is a general trend that a user with more followers is able to attract more reshares, when sampling the users with a follower number larger than 50, which is a typical number for average users in a social-network system [18].

▷ *Global influence vs. local influence*. The global influence (i.e., the number of both direct and indirect followers/friends that are attracted to join a particular content propagation) can be inferred from their local influence (i.e., the number of only the direct followers/friends attracted by a social propagation). Fig. 9 illustrates the global influence versus the local influence of $1,226$ influential users randomly selected from the Weibo traces. Each sample in this figure represents the

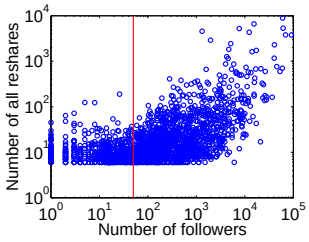


Fig. 8. Number of all attracted reshares versus number of a user's followers.

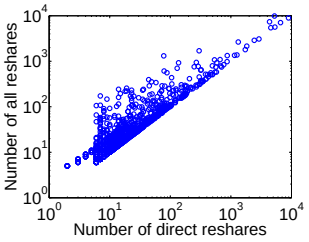


Fig. 9. Number of globally attracted reshares versus number of locally attracted reshares.

global influence of a user versus the local influence of the user. We observe a relatively strong correlation between the two influences, indicating a local influence can well estimate the global influence of a content.

To summarize, how social contents are downloaded can be highly affected by how contents are shared by people and propagating in the online social network. In our study, we observe that information from a social network can be used to improve the content delivery in the CPCDN design.

4 Using Context and User Information for Content Delivery in CPCDN

To further investigate the advantage of a CPCDN design, in this section, we present how the CPCDN paradigm can improve the content delivery, with a few content delivery examples using design principles summarized from our measurements.

4.1 Context-Aware Request Schedule

In this subsection, we present the server redirection and request queuing strategies based on the content context.

4.1.1 Request Schedule based on Component Importance

According to our measurement studies, contents have heterogeneous importance when they are delivered in different context. In the context of web content delivery, an importance index $v_p(c)$ can be defined to represent the important level of content c when it is composed in webpage p (context). We assume that the size, position available to the CPCDN can be used for the calculation of $v_p(c)$. A component with a larger $v_p(c)$ will be considered more important than a component with a smaller $v_p(c)$. Notice that different types of semantic information can be used to infer the content importance for different applications.

4.1.2 Context-Aware Bottleneck Set

A CPCDN has the content and user information (e.g., content composition and user profile), and can monitor the network information (i.e., the delay and bandwidth between peering servers and users), making context-aware content delivery possible. CPCDN allows users to simultaneously download all the components in the webpages, and determines the priority of different content components based on the following information: (1) *CP information*. We assume a user u is requesting a webpage containing a set of content components $\mathcal{G} = \{c_1, c_2, \ldots, c_n\}$, where c_i is the ith content component in this webpage; (2) *CDN information*. Let $E(c)$ denote the set of servers that have replicated content c, i.e., servers in $E(c)$ are the peering servers of content c.

A bottleneck component set which contains the components most likely to affect the user experience is used for the prioritization. Based on our measurement study, we construct the bottleneck content component set $\mathcal{D}$ according to the content importance index, as follows:

1) Let $\mathcal{D} = \{\}$;
2) Add critical components from $\mathcal{G}$ to $\mathcal{D}$, including the framework HTML, JS and CSS, *etc.*;
3) Add contents with an importance index $v_p(c) \geq \theta$, where θ is a parameter to select the critical components. θ is dynamically changed according to the composition of the webpages, and in our experiments, θ is selected to include the content components on the first screen of the web pages.

Requests for contents in $\mathcal{D}$ are then the critical ones to be delivered to users requesting a particular webpage.

4.1.3 Strategies of Server Redirection and Requesting Queuing

Being aware of the bottleneck contents requested by different users, the CPCDN is able to strategically schedule these requests. Our design of the context-aware schedule includes the following strategies: (1) the server redirection and (2) the request queuing.

▷ *Server redirection* is the strategy to choose a peering server to respond to a user's request when there are multiple candidate servers. In a traditional CDN, peering servers are selected without considering the context in which the contents are delivered, e.g., the GSLB (Global Server Load Balance [19]) which is typically used to redirect user requests, only considers the load balance of the servers. As a result, requests for contents with different importance levels are served using the same approach. In a CPCDN paradigm, by inferring the importance of the content components, users' preference of different servers, and the servers' load (Sec. 3), we can strategically assign "fast" servers to respond to the requests for bottleneck contents in $\mathcal{D}$ (recall that a content c is served by multiple candidate servers in $E(c)$), and let the "slow" servers serve requests for less important contents.

▷ *Request queuing* is the other strategy used to determine which requests are served first at a particular peering server. In our design, a CPCDN peering server always prioritizes the critical requests in $\mathcal{D}$ over the other requests in $\mathcal{G} - \mathcal{D}$ received. To let the queuing strategy also perform in a fair manner, the prioritization only

adjusts the queue in the same timeslot — $\mathcal{Q}_T$ (referred to as a *timeslot queue*), which is the priority queue where requests received in timeslot T are stored, and the *global request queue* at the peering server is the sequence of multiple timeslot queues. Since the requests are only prioritized within a timeslot queue, the requests in $\mathcal{G}-\mathcal{D}$ can be finally served within a given time threshold determined by the timeslot length. Assume the original expected waiting time for a request to be served is x_0, then the waiting time for bottleneck components then can be reduce by $\frac{(1-\sigma)x_0}{K}$, where K is the number of timeslot queues maintained by the peering server, and σ is the expected fraction of bottleneck contents. The rationale is that a bottleneck content can be prioritized over other contents within the same timeslot queue. The system can choose K to tradeoff effectiveness of delivering bottleneck components and fairness of delivering other contents.

4.2 Crowd-Based Content Replication

Next, we present our replication design in CPCDN, using the intelligence of crowd patterns. In particular, we perform content replication across peering servers based on the user group preference derived from the co-clustering.

4.2.1 User-Content Relevance based on Co-clustering

According to the observed crowd patterns, the key to predict a user's preference is to infer the preference of her user group. In particular, users or contents that are clustered into the same group share similar properties [16]. Based on the user-content matrix $\mathbf{M}$ generated using the historical request records, we co-cluster users and contents. Using the x- and y-indices of user activities in the user-content matrix in Sec. 3.4.1 as their 2D locations, we can calculated the *Euclid distance* between these activities. We use a distance threshold to identify which cluster a user-content activity belongs to, i.e., an activity belongs to a cluster if the distance between the activity and the center of the cluster (an output of the co-clustering algorithm) is smaller than the threshold. This threshold can be determined by a content provider according to its content services/types, or automatically learnt from classification algorithms. Let $\mathcal{U}$ denote a set of the corresponding users clustered and $\mathcal{C}$ denote the set of contents clustered in the same group. In our design, we use a cluster number that Tencent uses in its social network service for the number of types of celebrities [7].

Based on our co-clustering results, we define a user-content relevance index e_{uc} which represents the possibility level for user u to request content c in the future according to the clustering, which is calculated as follows:

$$e_{uc} = \frac{\sum_{i\in\mathcal{U}} \mathbf{M}_{ic} \sum_{j\in\mathcal{C}} \mathbf{M}_{uj}}{|\mathcal{U}||\mathcal{C}|}. \quad (1)$$

A large e_{uc} indicates that user u is likely to request content c according to the historical crowd patterns. The rationale is that a large e_{uc} indicates that (1) the user has already requested more contents in the same cluster which content c belongs to, and (2) the content has already been frequently requested by users in the same group. Note that e_{uc} is not the exact probability for user u to request c, but is the index for ranking the contents.

4.2.2 Replication based on User-Content Relevance

Based on the relevance information, the CPCDN is able to proactively replicate contents to servers that are close to the users. We calculate a regional replication vector for content c based on the user locations, which can be inferred from their profiles hosted by the CPCDN. The regional replication vector is calculated as follows: $V_c = \{v_{c1}, v_{c2}, \ldots, v_{cR}\}$, where $v_{cr} = \sum_{u|\mathbf{r}(u)=r} e_{uc}$ ($\mathbf{r}(u)$ denotes the region of user u). The CPCDN replicates a content to a new region that will incur a high request level. Based on the above knowledge, we design the heuristic Algorithm 1 to perform the replication for the newly published contents, which are the ones a traditional CDN has no or little popularity information to handle.

This algorithm is carried out periodically, so that a content can be replicated to more and more regions, if it keeps the popularity. In this algorithm, $\mathcal{R}$ is the set of the candidate replications (each entry in $\mathcal{R}$ is a pair (c, r) indicating content c is to be replicated to region r), $\mathcal{A}$ is the set of all contents that are published in the recent timeslot (a CPCDN is aware of all the newly published contents). The matrix $\mathbf{M}$ is constructed by retrieving the request records of users who have already requested contents in $\mathcal{A}$.

We push all the content-region pairs into the candidate set $\mathcal{R}$, and rank them according to the replication index v_{cr} — a larger v_{cr} indicates that c should be replicated to region r where more users are requesting it. The replication is iteratively performed until the server replication load is exceeded.

4.3 Influence-Based Video Popularity Prediction

We present how CPCDN utilizes the user intelligence in content delivery, with respect to the social influence. Fig. 10 illustrates the general idea of utilizing the user intelligence for content delivery. After contents are shared in the online social network among users, their popularity can be predicted from the social relationship and social activity. In particular, CPCDN is able to perform bandwidth reservation for social contents that become dramatically popular due to influential people sharing them. Our design utilizes both the traditional popularity and the social influential information for the content popularity prediction in the social content delivery.

▷ *Traditional content popularity*. In our design, the traditional content popularity (i.e., the popularity of a content perceived in traditional platforms) is important

Algorithm 1 Crowd-pattern based replication.

procedure REPLICATION
  Candidate replication set $\mathcal{R} = \Phi$
  Co-cluster the user-content matrix $\mathbf{M}$ involving the contents in $\mathcal{A}$
  Update the user-content relevance index e_{uc} using Eq. 1
  for all $c \in \mathcal{A}$ **do**
    Update V_c using the relevance index
    Push all (c, r) into $\mathcal{R}$
  end for
  Rank $\mathcal{R}$ in v_{cr}'s descending order
  for all (c, r) in the ranked $\mathcal{R}$ **do**
    Content c is replicated at r
    Update regional server load
    break if servers are fully replicated
  end for
end procedure

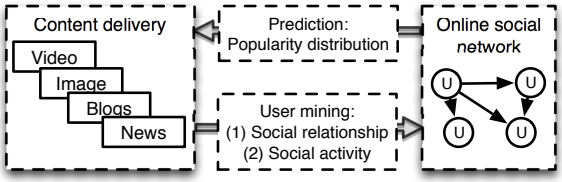


Fig. 10. Guiding social content delivery in CPCDN using predicted popularity based on the user intelligence.

because the online contents can be distributed by not only the new online social network, but also traditional approaches (e.g., centralized content portal). As a result, the number of requests still reflects the content's intrinsic popularity level.

▷ *Social influential index*. On the other hand, after a content is distributed along the social connections. The popularity of a content is highly affected by users. We incorporate the social influence of users into our popularity inference. Our measurement studies have shown that when contents are provided to users in the online social network, user influence can dramatically affect the content popularity. According to our measurement insights, we estimate the influence of users using the knowledge of their social connections and their direct influence level.

Social popularity index. In our design, we make the most intuitive combination of the two aspects. We design a *social popularity index* of a content c as follows:

$$x(c) = \alpha p(c) + (1 - \alpha) \sum_{u \in \mathcal{S}(c)} (f(u)\rho(u)),$$

where $p(c)$ is the traditional popularity of content c, which is calculated as the number of recent viewers of content c, $f(u)$ is the follower number of user u, $\rho(u)$ is the average resharing ratio of user u, i.e., the average fraction of followers that have shared the contents posted by user u in the recent historical time window, and $\mathcal{S}(c)$ is the set of users who have shared content c. A content with a large $x(c)$ is likely to attract more requests in the near future.

The rationale is that a content with a large social popularity index is either very popular in the recent time window, or has been shared by many influential people. In our experiments, α is learnt from the traces using the collected social factors and the real content popularity in a recent time window: α is assigned a larger value for contents with a more "inherent popularity", i.e., the correlation between content popularity and social influence is weak. In our experiments, we use the fraction of the influenced users (i.e., users whose friends are already resharers of a content) to represent the inference level of a particular content.

5 Measurement-Based Performance Evaluation

In this section, we present the measurement-based evaluation of our intuitive designs in Sec. 4, using simulation experiments, to demonstrate the potential of CPCDN.

5.1 Experiment Setup

CPCDN peering servers. In our simulation, 50 peering servers are deployed at different sites (a region-ISP pair) to serve the contents — the number of replicas of each content is proportional to its popularity calculated as the number of historical requests. In our experiments, we randomly set the capacities of the servers at different levels.

User activity. $2,000$ users are selected from our traces, and we randomly assign download speeds from the TCP records to simulate their download speeds from the peering servers. We use the records in the traces to generate our strategies, including the context-aware request schedules, the crowd-based replication and the social popularity prediction, assuming that such information is available to a CPCDN. Users' activities are driven by the traces as well, which will be presented in detail in the following experiments.

5.2 Performance of Context-Aware Request Schedule

We measure how fast users can receive $1,000$ webpages under the Tencent portal website, which share common content components including images, texts and HTML/JS framework codes. We use the first-screen load delay as a metric, which is the time a user spends on receiving the content components to be displayed on the first screen of the browser. Prior studies show that typical web browsing behaviors demonstrate that it is common for users to move across multiple webpages in one browsing "session", and the first-screen load delay has a significant impact on the quality of experience in web browsing. For instance, in 2009 Amazon demonstrated

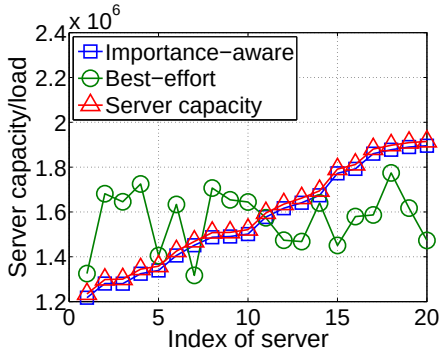
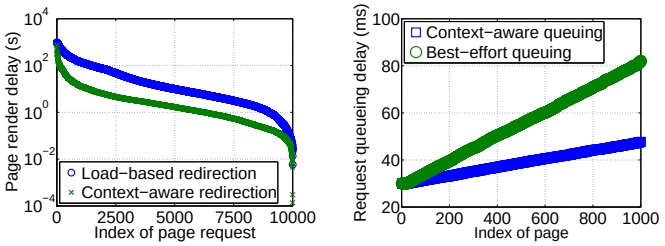


Fig. 11. Server load under different request schedule



(a) First-screen load delay under different server redirection strategies.

(b) Request queuing delay under different queue schemes.

Fig. 12. Evaluation of the context-aware server redirection and request queuing.

that every 100 milliseconds of latency which occurred on their website resulted in a 1% loss in sales, because of impatient users leaving the webpages.

In our experiments, we set the window size to be 1024x768. We simulate 10 million webpage requests, each of which randomly selects one among the 1,000 webpages. Multiple connections are created to download the components composed in a webpage simultaneously.

First, we evaluate server load balance improved by the request scheduling. We compare our bottleneck-based balancing with a best-effort scheme, where users are always served by the best-bandwidth servers inferred from historical information (*e.g.*, the GLSB generally gives the server list to users using the location information). As illustrated in Fig. 11, each sample represents the server load or server capacity versus the server index. We observe that our CPCDN strategy can schedule the requests to servers well according to their capacity and loads, so that servers will not be overwhelmed by the requests; while in the best-effort scheme, since the schedule is based on static information, users can be redirected to servers that are currently heavily loaded.

Next, we evaluate first-screen load delay reduced by the context-aware server redirection. We compare it with a load-based server redirection, where requests for contents in a webpage are redirected to a server with the lowest load currently. In our experiments, each webpage contains 20 bottleneck content components, and the average number of replicas of a content component is 3. The speed of users downloading contents from these servers is set according to the TCP traces. We generate 10,000 requests from among 2,000 users (each user will randomly select 5 webpages to request). As illustrated in Fig. 12(a), each sample represents the first-screen load delay versus the index of the request. We observe that our server redirection strategy can significantly reduce the first-screen load delay — for a large fraction of the webpage views, the first-screen load delay can be reduced by over 90%. The reason is that our strategy is designed to match the importance of contents in different contexts with the performance of peering servers to a user.

We also compare our prioritization strategy implemented on the servers, with the conventional best-effort scheme, where requests are only queued according to their arrival time stamps. Fig. 12(b) illustrates the webpage delivery time versus the index of the webpage requests. We observe that our context-aware queuing strategy based on the context information significantly reduces the webpage delivery time, compared with the best-effort strategy. An average of 50% reduction of the queueing delay can be achieved by our prioritization, indicating a large improvement that can be adopted by the web server implementation.

5.3 Performance of Crowd-Based Replication

The user ID and content ID in the traces provided by Tencent allow us to track which contents a user has requested. Our evaluation is based on traces collected from Tencent Weibo in 10 days — the full matrix recording the user-content requests contains 2.1% of "1" entries. In our experiments, by adjusting the number of days/hours of traces for the matrix co-clustering, we are able to generate different sparsities, e.g., choosing the records of the most recent 1 day generates a matrix with a sparsity 0.2%.

We use the user requests on the last day as the ground truth, and vary the length of the previous time slot to change the sparsity of the matrix. Fig. 13 compares the crowd-based replication with a popularity-based scheme, in which contents are replicated only according to their historical popularity. The metric is the local download fraction, i.e., the fraction of users that can download from servers located in their own regions. The rationale is that if a user downloads from a local server, a small delay and large bandwidth is expected. Each sample in this figure represents the fraction of requests served by the local servers versus the number of regions each content can be replicated to. We observe that for the typical social contents, their local download fraction achieved by our crowd-based replication outperforms the popularity-based approach by up to 50%, especially when the contents are not fully replicated (i.e., replicated at all the peering servers). The reason is that crowd patterns reflect the potential users of contents that have short popularity. In today's content replications, only the most popular content items can be fully-replicated

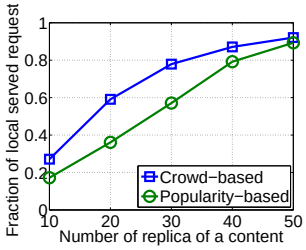


Fig. 13. Local download fraction versus the number of replication points.

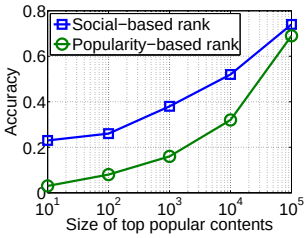


Fig. 14. Popularity content prediction accuracy versus the number of top popularity social contents.

according to the administrators' operations, indicating that our design has a significant improvement to the conventional content replication strategies for regular contents.

5.4 Performance of Social Content Popularity Prediction

Finally, we evaluate the influence-based content popularity prediction. Our evaluation uses the dataset with $300,000$ videos shared by users in Tencent Weibo in 10 days. The traces are divided into 5 groups, each of which contains the traces of 2 consecutive days — traces of the first days are used as training dataset, and the traces of the second days are used as ground truth. In particular, we use how users reshare contents and their follower numbers in the first day to calculate the content social popularity index, according to our design in Sec. 4.3. Contents are ranked in the social popularity index's descending order, and compared with the index of the ground truth.

We compare our ranking strategy with a popularity-based scheme, in which contents are ranked only according to their traditional popularity (i.e., number of views) in the previous timeslot. We calculate the fraction of overlapped contents between the true rank and the predicted ranks by both strategies as the performance metric — a larger fraction indicates more accurate prediction. In Fig. 14, each sample represents the fraction of overlapped contents versus the number of the most popular contents. We observe that our ranking based on the social influence works much better than the popularity-based scheme, especially for the popular contents. An improvement of 3 times can be achieved when the number of the most popular contents is around 100, indicating that our prediction can detect flash crowds of social contents, which can be common in online social networks [32].

6 Related Work

6.1 Traditional Content Delivery

Originally, content delivery networks were proposed to address the problem that Internet service quality perceived by customers were largely unpredictable and unsatisfactory when contents are only placed at one original server. By caching the contents to edge peering points, users can download from servers that are close to them, achieving a better download experience with small propagation delay and large bandwidth [14]. Ager *et al.* [4] have studied the content hosting in such edge content delivery infrastructures.

There are a lot of CDNs for a content provider to choose to deliver its contents. The dilemma of selecting one between the multiple CDN providers is that they have different strengths and weaknesses [1], e.g., they have deployed servers at different geographical regions, covering different user groups [28]. A practice today's content providers use is to employ multi-CDN [25] (i.e., multiple traditional CDNs) or hybrid-CDN [38] (i.e., peer-assisted CDN) to deliver their contents — users are redirected to different CDNs according to a set of rules. This however has not solved the fundamental problem, i.e., how can the CP-level information that is increasingly changing the content delivery be used by the delivery systems?

Today, contents are dynamically generated [21], socially propagating among social-networked users, and increasingly attracting popularity in the content ecosystem [8]. It is challenging for a traditional CDN to provide good service without considering the content context and user influence in a one-size-fits-all manner.

6.2 New Trends in Content Delivery

First, content delivery can be highly dynamic and personal. Li *et al.* [24] have observed that in the video content delivery, multiple versions of videos have to be generated for users with different devices to receive the video contents. Brewington *et al.* [6] have studied the webpages and observe that the contents in webpages are highly dynamical. The context in content delivery includes not only the contents themselves, but also the client, server and network requirements [22]. Verbert *et al.* [33] have surveyed the adaptive content recommendation according to the user context including where the content context. In our study, we are focused on the context information that is particularly owned by the content provider.

Second, online social network has greatly changed the content delivery, e.g., the distribution of social contents is shifted from a "central-edge" manner to an "edge-edge" manner. Bakshy *et al.* [5] have studied the social influence of people in the online social network, and observed that users can be very influential in the online social network. Li *et al.* [23] study the content sharing in the online social network, and observed the skewed popularity distribution of contents and the power-law activity of users. Pujol *et al.* [29] have designed a social partition and replication middle-ware where users' friends' data can be co-located in the same server. In our previous studies [36], [35], we investigated the possibility to infer

users' preference of contents according to their social profiles and behaviors, and the way to allocate network resource at edge CDN servers using social predictions based on information collected from social networks. In a CPCDN, the social relationship and social activities of users allow the new design space to explore how the make use of the user intelligence for content delivery.

Related works have been studying the content delivery and CP-level intelligence in a separate way, i.e., content delivery network is optimizing the delivery for individual contents and individual users requests, without using the upper-layer information including context and user intelligence. In our study, we are focused on the CPCDN design space where the upper information and intelligence can be utilized in content replication, request scheduling and bandwidth reservation.

7 Concluding Remarks

CPCDN is based on the idea that CP-level intelligence can improve the performance of content delivery. Our measurement studies illustrate the benefits of exploiting two indispensable intelligences in content delivery: context intelligence and user intelligence. By systematically exploring the CPCDN design space, we design architecture and algorithms to extract the maximum potential of CPCDN. In particular, we use the context intelligence to schedule the component delivery according to their diverse importance under different contexts, we utilize the user-content crowd pattern to replicate contents close to users of potential interests, and furthermore we leverage the user-user influence to provision resources for social contents that are likely to attract flash crowds. Our measurement-based evaluations demonstrate the effectiveness of our CPCDN design — as compared to the conventional CDN approaches as follows: the webpage load delay is reduced by context-aware request schedule, the local download improvement is achieved when the crowd patterns are considered in content replication; and a significant improvement is achieved in the prediction of popular social contents that attract flash crowds. Our results demonstrate the potential of CPCDN in enhancing content delivery performance for today's online services.

Acknowledgment

This work is supported in part by the National Basic Research Program of China (973) under Grant No. 2011CB302206, the National Natural Science Foundation of China under Grant No. 61210008, 61272231 and 61402247, SZSTI under Grant No. JCYJ20140417115840259, the research fund of Tsinghua-Tencent Joint Laboratory for Internet Innovation Technology, Beijing Key Laboratory of Networked Multimedia, and the University Grants Committee of the Hong Kong Special Administrative Region, China (General Research Fund Project No. 14201014).

References

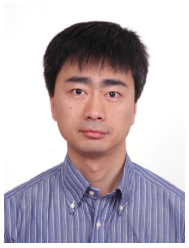
[1] CDN Expert Online. http://cdnexpertonline.com/node/45.
[2] Tencent Weibo. http://t.qq.com/.
[3] Tencent Portal Website. http://www.qq.com.
[4] B. Ager, W. Mühlbauer, G. Smaragdakis, and S. Uhlig. Web Content Cartography. In *ACM Internet Measurement Conference (IMC)*, 2011.
[5] E. Bakshy, J. Hofman, W. Mason, and D. Watts. Everyone's an Influencer: Quantifying Influence on Twitter. In *ACM International Conference on Web Search and Data Mining (WSDM)*, 2011.
[6] B. Brewington and G. Cybenko. How Dynamic Is the Web? *Computer Networks*, 33(1):257–276, 2000.
[7] T. celebrities, http://zhaoren.t.qq.com/rank.php.
[8] M. Cha, H. Kwak, P. Rodriguez, Y. Ahn, and S. Moon. I Tube, You Tube, Everybody Tubes: Analyzing the World's Largest User Generated Content Video System. In *ACM SIGCOMM*, pages 1–14, 2007.
[9] M. Cha, A. Mislove, and K. Gummadi. A Measurement-Driven Analysis of Information Propagation in the Flickr Social Network. In *ACM International Conference on World Wide Web (WWW)*, 2009.
[10] M. Cha, A. Mislove, and K. P. Gummadi. A Measurement-driven Analysis of Information Propagation in the Flickr Social Network. In *ACM International Conference on World Wide Web (WWW)*, 2009.
[11] F. Chen, K. Guo, J. Lin, and T. La Porta. Intra-Cloud Lightning: Building CDNs in the Cloud. In *IEEE International Conference on Computer Communications (INFOCOM)*, 2012.
[12] CloudFront. http://http://aws.amazon.com/cloudfront/.
[13] F. O. Computing. https://www.facebook.com/note.php?noteid =10150144039563920.
[14] A. Datta, K. Dutta, H. Thomas, D. VanderMeer, and K. Ramamritham. Proxy-Based Acceleration of Dynamically Generated Content on the World Wide Web: an Approach and Implementation. *ACM Transactions on Database Systems*, 29(2):403–443, 2004.
[15] I. Dhillon. Co-Clustering Documents and Words Using Bipartite Spectral Graph Partitioning. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, 2001.
[16] A. Di Marco and R. Navigli. Clustering and Diversifying web Search Results With Graph-Based Word Sense Induction. *Computational Linguistics*, 39(3):709–754, 2013.
[17] N. B. Ellison, C. Steinfield, and C. Lampe. The Benefits of Facebook "Friends:" Social Capital and College Students' Use of Online Social Network Sites. *Journal of Computer-Mediated Communication*, 12(4):1143–1168, 2007.
[18] B. Huberman, D. Romero, and F. Wu. Social networks that matter: Twitter under the microscope. *Available at SSRN 1313405*, 2008.
[19] S. Kommula, I. Hsu, R. Jalan, D. Cheung, et al. Global Server Load Balancing, Aug. 7 2007. US Patent 7,254,626.
[20] H. Kwak, C. Lee, H. Park, and S. Moon. What Is Twitter, a Social Network or a News Media? In *ACM International Conference on World Wide Web (WWW)*, 2010.
[21] I. Lazar and W. Terrill. Exploring content delivery networking. *IT Professional*, 3(4):47–49, 2001.
[22] T. Lemlouma and N. Layaïda. Adapted Content Delivery for Different Contexts. In *IEEE Symposium on Applications and the Internet*, 2003.
[23] H. Li, H. Wang, and J. Liu. Video Sharing in Online Social Network: Measurement and Analysis. In *ACM Network and Operating System Support for Digital Audio and Video (NOSSDAV)*, 2012.
[24] Z. Li, Y. Huang, G. Liu, F. Wang, Z. Zhang, and Y. Dai. Cloud Transcoder: Bridging the Format and Resolution Gap between Internet Videos and Mobile Devices. In *ACM Network and Operating System Support for Digital Audio and Video (NOSSDAV)*, 2012.
[25] H. Liu, Y. Wang, Y. Yang, A. Tian, and H. Wang. Optimizing Cost and Performance for Content Multihoming. In *ACM SIGCOMM*, 2012.
[26] N. OpenConnect. http://signup.netflix.com/openconnect.
[27] T. O'reilly. What is web 2.0, 2005.
[28] I. Poese, B. Frank, B. Ager, G. Smaragdakis, and A. Feldmann. Improving Content Delivery Using Provider-Aided Distance Information. In *ACM Internet Measurement Conference (IMC)*, 2010.
[29] J. Pujol, V. Erramilli, G. Siganos, X. Yang, N. Laoutaris, P. Chhabra, and P. Rodriguez. The Little Engine(s) That Could: Scaling Online Social Networks. *ACM SIGCOMM Computer Communication Review*, 40(4):375–386, 2010.
[30] S. Scellato, C. Mascolo, M. Musolesi, and J. Crowcroft. Track Globally, Deliver Locally: Improving Content Delivery Networks by Tracking Geographic Social Cascades. In *ACM International Conference on Multimedia (Multimedia)*, 2011.
[31] G. Style. http://en.wikipedia.org/wiki/gangnam_style.

[32] G. Szabo and B. A. Huberman. Predicting the Popularity of Online Content. *Communications of the ACM*, 53(8):80–88, 2010.
[33] K. Verbert, N. Manouselis, X. Ochoa, M. Wolpers, H. Drachsler, I. Bosnic, and E. Duval. Context-Aware Recommender Systems for Learning: a Survey and Future Challenges. *IEEE Transactions on Learning Technologies*, 5(4):318–335, 2012.
[34] Z. Wang, L. Sun, X. Chen, W. Zhu, J. Liu, M. Chen, and S. Yang. Propagation-based Social-aware Replication for Social Video Contents. In *ACM International Conference on Multimedia (Multimedia)*, 2012.
[35] Z. Wang, L. Sun, C. Wu, and S. Yang. Guiding Internet-Scale Video Service Deployment Using Microblog-based Prediction. In *IEEE International Conference on Computer Communications (INFOCOM)*, 2012.
[36] Z. Wang, L. Sun, W. Zhu, S. Yang, H. Li, and D. Wu. Joint Social and Content Recommendation for User Generated Videos in Online Social Network. *IEEE Transactions on Multimedia*, 15(3):698–709, April 2013.
[37] J. Wei and C.-Z. Xu. Measuring Client-Perceived Pageview Response Time of Internet Services. *IEEE Transactions on Parallel and Distributed Systems*, 22(5):773–785, 2011.
[38] H. Yin, X. Liu, T. Zhan, V. Sekar, F. Qiu, C. Lin, H. Zhang, and B. Li. Design and Deployment of a Hybrid CDN-P2P System for Live Video Streaming: Experiences With Livesky. In *ACM International Conference on Multimedia (Multimedia)*, 2009.

Zhi Wang (S'10 M'14) received his B.E. and Ph.D. degrees in Computer Science in 2008 and 2014, from Tsinghua University, Beijing, China. He is currently an assistant professor in the Graduate School at Shenzhen, Tsinghua University. His research areas include online social network, mobile cloud computing and big-data systems. He received the Best Paper Award at ACM Multimedia 2012. He is a member of IEEE.

Wenwu Zhu (M'97 SM'01 F'10) received the Ph.D. degree from New York University Polytechnic School of Engineering in 1996 in Electrical and Computer Engineering. He is with Computer Science Department of Tsinghua University as Professor of "1000 People Plan" of China. His current research interests are in the area of multimedia cloud computing, social media computing, multimedia big data, and multimedia communications and networking. He served(s) on various editorial boards, such as Guest Editor for the Proceedings of the IEEE, IEEE T-CSVT, and IEEE JSAC; Associate Editor for IEEE Transactions on Mobile Computing, IEEE Transactions on Multimedia, and IEEE Transactions on Circuits and Systems for Video Technology; Leading Editor of the Area "Computer Networks and Distributed Computing" of Journal of Computer Science and Technology. He received the Best Paper Award in ACM Multimedia 2012, the Best Paper Award in IEEE Transactions on Circuits and Systems for Video Technology in 2001, and the other 4 international Best Paper Awards.

Minghua Chen (S'04 M'06 SM'13) received his B.Eng. and M.S. degrees from the Dept. of Electronic Engineering at Tsinghua University in 1999 and 2001, respectively. He received his Ph.D. degree from the Dept. of Electrical Engineering and Computer Sciences at University of California at Berkeley in 2006. He joined the Dept. of Information Engineering, the Chinese University of Hong Kong in 2007, where he is currently an Associate Professor. He received the Eli Jury award from UC Berkeley in 2007 and The Chinese University of Hong Kong Young Researcher Award in 2013. He also received several best paper awards, including the IEEE ICME Best Paper Award in 2009, the IEEE Transactions on Multimedia Prize Paper Award in 2009, and the ACM Multimedia Best Paper Award in 2012. He is currently an Associate Editor of the IEEE/ACM Transactions on Networking. His recent research interests include energy systems (e.g., microgrids and energy-efficient data centers), distributed optimization, multimedia networking, wireless networking, network coding, and secure network communications.



Lifeng Sun (M'05) received his B.S. and Ph.D. degrees in System Engineering in 1995 and 2000 from National University of Defense Technology, Changsha, Hunan, China. He was an postdoctoral fellow from 2001 to 2003, an assistant professor from 2003 to 2007, an associate professor from 2007 to 2013, and currently a professor all in the Department of Computer Science and Technology at Tsinghua University. His research interests lie in the areas of online social network, video streaming, interactive multi-view video, and distributed video coding. He is a member of IEEE and ACM.

Shiqiang Yang (M'97 SM'08) received the B.E. and M.E. degrees in Computer Science from Tsinghua University, Beijing, China in 1977 and 1983, respectively. From 1980 to 1992, he worked as an assistant professor at Tsinghua University. He served as the associate professor from 1994 to 1999 and then as the professor since 1999. From 1994 to 2011, he worked as the associate header of the Department of Computer Science and Technology at Tsinghua University. He is currently the President of Multimedia Committee of China Computer Federation, Beijing, China and the co-director of Microsoft-Tsinghua Multimedia Joint Lab, Tsinghua University, Beijing, China. His research interests mainly include multimedia procession, media streaming and online social network. He is a senior member of IEEE.