Zhenhua Li · Yafei Dai
Guihai Chen · Yunhao Liu

Content Distribution for Mobile Internet: A Cloud-based Approach

Springer

Content Distribution for Mobile Internet: A Cloud-based Approach

Zhenhua Li · Yafei Dai · Guihai Chen
Yunhao Liu

Content Distribution for Mobile Internet: A Cloud-based Approach

Springer

Zhenhua Li
Tsinghua University
Beijing
China

Guihai Chen
Shanghai Jiao Tong University
Shanghai
China

Yafei Dai
Peking University
Beijing
China

Yunhao Liu
Tsinghua University
Beijing
China

ISBN 978-981-10-1462-8 ISBN 978-981-10-1463-5 (eBook)
DOI 10.1007/978-981-10-1463-5

Library of Congress Control Number: 2016943326

Printed on acid-free paper

This Springer imprint is published by Springer Nature
The registered company is Springer Science+Business Media Singapore Pte Ltd.

Preface

Content distribution (also known as *content delivery*) is the most fundamental function of the Internet, i.e., distributing *digital content* from one *node* to another node or multiple nodes. Here digital content includes webpage, image, software, audio, video, and so on; a node can be a large server cluster, a personal computer, a smartphone, a tiny sensor, *etc.* Typical content distribution systems include Akamai, BitTorrent, eMule, Skype, Xunlei, QQXuanfeng, and so forth. Since Amazon's launch of Elastic Compute Cloud (EC2) in 2006 and Apple's release of iPhone in 2007, Internet content distribution has illustrated a strong trend of *polarization*. On one hand, great fortune has been invested in building heavyweight and integrated data centers, in order to achieve the economies of scale and the cost efficiency for content distribution. On the other hand, end user devices have been growing more lightweight, mobile, and heterogeneous, thus posing rigorous requirements on the traffic usage, energy consumption, speed, and latency of content distribution.

Through comprehensive real-world measurements, we observe that existing content distribution techniques often exhibit poor performance under the settings of cloud computing and mobile Internet. To address the issue, this book investigates content distribution for mobile Internet with a cloud-based approach, by designing novel traffic-saving, energy-efficient, high-speed, and delay-tolerant content distribution techniques and frameworks that automatically adapt to mobile scenarios.

The major content of this book is organized in six parts, which are further elaborated into ten chapters. Specifically, we start with the background and overview in Part I. Then, since cellular traffic is the most expensive among all Internet traffic, its cloud-based optimization is first explored in Part II. Next, video content dominates the majority of Internet traffic, whose delivery deserves deep investigation in Part III. Moreover, P2P content distribution incurs little infrastructure cost and can scale well with the user base; however, its working efficacy can be poor and unpredictable without the assistance of cloud platforms, which is carefully addressed in Part IV. In addition, as an advanced paradigm of content distribution, cloud storage services like Dropbox and Google Drive have quickly gained enormous popularity in recent years, which are widely studied in Part V. At last, we summarize the major research contributions and discuss the future work in Part VI.

To be more specific, we summarize the main body of the book as follows:

Part II *Cloud-based Cellular Traffic Optimization.* As the penetration of 3G/4G/5G data networks, cellular traffic optimization has been a common desire of both cellular users and carriers. Together with the Baidu PhoneGuard team, we design and deploy TrafficGuard, a third-party mobile traffic proxy widely used by over 10 million Android devices (Chap. 2). TrafficGuard effectively reduces cellular traffic using a network-layer virtual private network (VPN) that connects a client-side proxy to a centralized traffic processing cloud. Most importantly, it works transparently across heterogeneous apps, so it is not constrained to any specific app.

Part III *Cloud-based Mobile Video Distribution.* Driven by the special requirements of mobile devices on video content distribution, we measure and analyze the industrial "cloud downloading" (Chap. 3), "cloud transcoding" (Chap. 4), and "offline downloading" (Chap. 5) services based on the Tencent Xuanfeng system and popular smart home routers. In particular, we diagnose their respective performance bottlenecks and propose the corresponding optimization schemes.

Part IV *Cloud-assisted P2P Content Distribution.* Through large-scale measurements and analysis of industrial cloud-assisted peer-to-peer (P2P) systems like QQXuanfeng and Xunlei, we extract the basic model of "cloud tracking" content distribution in Chap. 6. Further, we design the "cloud bandwidth scheduling" algorithm to maximize the cloud bandwidth multiplier effect in Chap. 7.

Part V *Cloud Storage-oriented Content Distribution.* We are the first to discover the "traffic overuse problem" that pervasively exists in today's cloud storage services. Also, we propose and implement a variety of algorithms to address the problem, such as BDS—batched data sync, IDS—incremental data sync, ASD—adaptive sync defer (Chap. 8), and UDS—update-batched delayed sync (Chap. 9).

In this book, we provide a series of useful takeaways and easy-to-follow experiences to the researchers and developers working on mobile Internet and cloud computing/storage. Additionally, we have built an educational and experimental cloud computing platform (http://www.thucloud.com) to benefit the readers. On top of this platform, the readers can monitor (virtual) cloud servers, accelerate web content distribution, explore the potentials of offline downloading, acquire free cloud storage space, and so forth. Should you have any questions or suggestions, please contact the four authors via *lizhenhua1983@gmail.com*, *dyf@pku.edu.cn*, *gchen@cs.sjtu.edu.cn*, and *yunhaoliu@gmail.com*.

Beijing, China
May 2016

Zhenhua Li
Yafei Dai
Guihai Chen
Yunhao Liu

Acknowledgments

First of all, we deeply appreciate the Chinese Association for Artificial Intelligence (CAAI) for the precious CAAI Ph.D. Thesis Award. In 2015, only eight Ph.D. theses across China received this award. To my knowledge, it was this award that drew the initial attention of Springer on my academic research results. After that, the editors Xiaolan Yao, Jane Li, and Celine Chang have kept in contact with me and provided plenty of useful information and materials for my preparing the book. Specifically, I wish to thank the following people for their contributions to this book:

- Prof. Christo Wilson (@ Northeastern University) guided me closely in my four best researches published in NSDI'16, IMC'15, IMC'14, and Middleware'13. I have learnt so much from him regarding how to properly tell a story, organize a manuscript, present a viewpoint, draw a figure, list a table, and so forth.
- Dr. Tianyin Xu (@ UCSD) helped me closely in conducting the researches published in NSDI'16, IMC'15, and IMC'14. I sincerely appreciate his devotion, honesty, patience, and selflessness during our collaboration.
- Zhefu Jiang (@ Cornell University and Google) helped develop the Linux kernel extension to our proposed UDS (update-batched delayed synchronization) middleware for cloud storage services. In addition, he provided valuable suggestions on the design and implementation of TrafficGuard.
- Weiwei Wang, Changqing Han, Junyi Shao, Xuefeng Luo, Min Guo, Cheng Peng, and Tianwei Wen (@ Baidu Mobile Security) helped develop the TrafficGuard system. Meanwhile, Xin Zhong, Yuxuan Yan, and Jian Chen (@ Tsinghua University) helped evaluate the client-side latency penalty and battery consumption of TrafficGuard.
- Yan Huang, Gang Liu, and Fuchen Wang (@ Tencent Research) helped develop the Xuanfeng system. Meanwhile, Yinlong Wang and Zhen Lu (@ Tsinghua University) helped evaluate the performance of multiple offline downloading systems including smart home routers.

- Prof. Yao Liu (@ SUNY Binghamton) helped evaluate the performance of Dropbox and WeChat in the US. I have learnt a great deal from her concerning measurement-driven research.
- Prof. Ben Y. Zhao (@ UCSB) guided me in conducting the researches published in NSDI'16 and Middleware'13. Also, I used to be a teaching assistant for his summer holiday course at Peking University.
- Prof. Zhi-Li Zhang (@ University of Minnesota) offered me freedom, trust, and valuable guidance during my 1-year visiting life at the University of Minnesota. Also, Dr. Cheng Jin (@ University of Minnesota) helped evaluate the performance of multiple cloud storage services in the US.
- Bob and Joanne Kraftson, as well as Mark and Jaycelyn Colestock, helped me a lot when I was visiting the University of Minnesota. They brought me warm, happy, and unforgettable memories.
- Prof. Xiang-Yang Li (@ University of Science and Technology China and Illinois Institute of Technology) provided useful suggestions on my research published in NSDI'16. I'm impressed by his hard working and diligence.
- Prof. Roger Zimmermann (@ National University of Singapore) helped me with the research published in JCST'15. I am impressed by his patience and elegance.
- Dr. Tieying Zhang (@ Chinese Academy of Sciences) helped me with the research published in IWQoS'12. Also, we collaborated on a number of other researches.
- Dr. Ennan Zhai (@ Yale University) kept in touch with me on a number of topics. Recently, our collaboration produced the first INFOCOM paper in my career.
- Dr. Yiyang Zhao, Dr. Jian Li, Zihui Zhang, Zhinan Ma, Lin Cui, Shuang Xin, Kun Duan, Songyang Wang, and Yong Yi (@ Tsinghua University) helped me a great deal in dealing with numerous affairs and issues in both my work and life.

This work is supported in part by the High-Tech Research and Development Program of China ("863—China Cloud" Major Program) under grant 2015AA01A201, the National Basic Research ("973") Program of China under grants 2014CB340303 and 2012CB316201, the National Natural Science Foundation of China (NSFC) under grants 61432002 (State Key Program), 61471217, 61232004, 61472252 and 61133006, and the CCF-Tencent Open Fund under grant IAGR20150101.

Beijing, China Zhenhua Li
May 2016

Contents

Part III Cloud-Based Mobile Video Distribution

Acronyms

AP	Access point, or WiFi home router
ASD	Adaptive synchronization defer
BDS	Batched data synchronization
BT	BitTorrent
C/S	Client/Server content distribution
CCN	Content centric networking
CDN	Content distribution (delivery) network
CloudP2P	Hybrid cloud and P2P content distribution
DASH	Dynamic adaptive streaming over HTTP
EC2	Elastic compute cloud
HLS	HTTP live streaming
IDS	Incremental data synchronization
NDN	Named data networking
P2P	Peer-to-Peer content distribution
P2SP	Peer-to-Server&Peer content distribution
PUE	Power usage effectiveness
QoS	Quality of service
RFID	Radio frequency identification device
S3	Simple storage service
SDN	Software-defined networking
TUE	Traffic usage efficiency
UDS	Update-batched delayed synchronization
VBWC	Value-based web cache
VoD	Video on demand
VPN	Virtual private network
WWW	World Wide Web, or "World Wide Wait"

Part I
Get Started

Chapter 1
Background and Overview

Abstract This chapter presents the background and overview of the book. First, we introduce the basic concept and history of Internet content distribution. Next, we illustrate the "heavy-cloud versus light-end" polarization of Internet content distribution under the novel settings of cloud computing and mobile Internet. Afterward, we review various frontier techniques that attempt to address current issues of Internet content distribution. At the end, we outline the entire book structure.

1.1 Internet Content Distribution

Content distribution (also known as *content delivery*) is the most fundamental function of the Internet, i.e., distributing *digital content* from one *node* to another node or multiple nodes. Here digital content includes webpage, image, document, email, software, instant message, audio, video, and so forth. A node can be a giant data center, a large server cluster, a home router, a personal computer, a tablet or smartphone, a tiny sensor or RFID (radio frequency identification device), *etc.* From a historical perspective, existing techniques for Internet content distribution can be generally classified into the following four categories:

1. *Client/Server* (*C/S*). The tidal wave of the Internet first rose at the Silicon Valley of the US in 1990s, represented by several pioneering companies like Yahoo! and Netscape. At that time, digital content distributed over the Internet is mainly composed of webpages, images, documents, and emails. Because these content is limited in terms of type, quantity, and capacity (size), using the simplest client/server technique (as shown in Fig. 1.1) to directly deliver content through TCP/IP connections can usually meet the requirements of Internet users.
   C/S content distribution was first embodied by the classical UNIX/BSD Socket implementation (IETF RFC-33) in 1970s. Notably, one author of the UNIX/BSD Socket is Vinton Cerf, known as the "Father of the Internet," the ACM Turing Award winner in 2004, and the president of ACM during 2012–2014.
2. *Content Distribution Network* (*CDN*). As the Internet became more ubiquitous and popular, its delivered digital content began to include large-size multimedia content, particularly videos. Meanwhile, the quantity of content underwent

Z. Li et al., *Content Distribution for Mobile Internet: A Cloud-based Approach*,
DOI 10.1007/978-981-10-1463-5_1

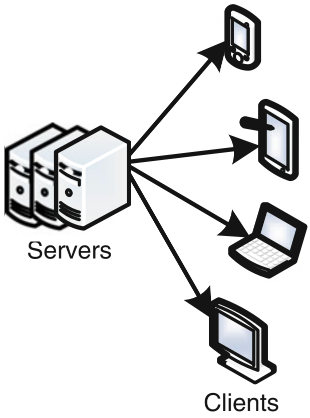


Fig. 1.1 Client/server content distribution. (Reprinted with permission from [12].)

an exponential growth. Both issues led to severe congestions on the Internet when most content was distributed in the C/S manner. As a consequence, "World Wide Web" (WWW) gradually deteriorated into "World Wide Wait". In 1995, Tim Berners-Lee, the inventor of WWW, posed a challenging problem to his colleagues at the Massachusetts Institute of Technology: "Can we invent a fundamentally new and better way to deliver Internet content?"

Interestingly, the man who gave the first solution to this problem was just his office neighbor, Prof. Tom Leighton. Leighton's research team proposed the idea of CDN, namely content distribution network or content delivery network, and founded the first CDN company called Akamai in 1998. CDN optimizes the performance of Internet content distribution by strategically deploying *edge servers* at multiple locations (often over multiple ISP networks) [9]. These edge servers cooperate with each other by replicating or migrating content according to content popularity and server load. An end user usually obtains a copy of content from a nearby edge server, so that the content delivery speed is greatly enhanced and the load on the original data source is effectively reduced.

To date, CDN has been the most widely used technique for accelerating Internet content distribution. Besides Akamai, representative CDN service providers include Limelight (founded in 2001), Level3 (founded in 1998), ChinaCache (founded in 1998), ChinaNetCenter (founded in 2000), and so on.

3. *Peer-to-Peer* (*P2P*). Although CDN is widely used across the Internet, it is subject to both economical and technical limitations. First, CDN is a charged facility that only serves the content providers who have paid (typically popular websites like YouTube and Netflix), rather than a public utility of the Internet. Moreover, even for those content providers who have paid, CDN is not able to accelerate the distribution of all their content, since the bandwidth, storage, and coverage of a CDN are constrained. Then the question is: can we simply leverage the resources of content receivers to accelerate Internet content distribution? More specifically, now that every end-user device possesses a certain amount of bandwidth, storage,

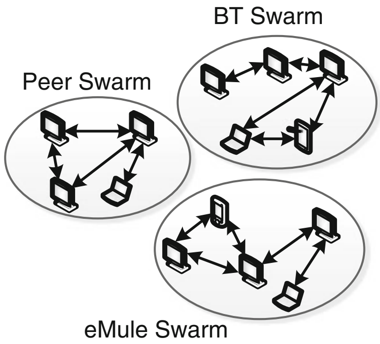


Fig. 1.2 Peer-to-peer content distribution. (Reprinted with permission from [12].)

and coverage, can we organize the numerous end-user devices into peer-to-peer data swarms in which shared content is directly delivered among interested peers (as demonstrated in Fig. 1.2)?

In 1999, the Napster music sharing system offered a solid and splendid answer to the above question—50 million users joined Napster in 6 months. Although Napster was then shut down soon for copyright reasons, it started the prosperity of P2P content distribution. Following the step of Napster, a series of well-known P2P systems quickly appeared, such as BitTorrent (abbreviated as BT), eDonkey/eMule, KaZaa, Skype, and PPLive [4]. These systems confirmed the huge power of content distribution concealed in Internet end users.

P2P content distribution also bears its intrinsic limitations. First, end-user devices do not work stably (i.e., highly dynamic). Second, end-user devices are diverse in bandwidth and storage capabilities (i.e., highly heterogeneous). Third, the users and content in peer swarms are short of reputation and quality authentication (i.e., highly unreliable). These limitations make it difficult to predict and control the performance of P2P content distribution, and thus the quality-of-service (QoS) perceived by users can hardly be guaranteed.

4. *Hybrid content distribution*. To address the potential limitations and meanwhile inherit the advantages of C/S, CDN and P2P, hybrid content distribution techniques came into being. They aim to integrate the stability and reliability of C/S and CDN, as well as the economy and scalability of P2P. Inevitably, their designs, implementations, and deployments are often quite complicated. As a matter of fact, most popular P2P systems (e.g., Skype, PPLive, PPStream, and UUSee) have transformed their network architectures to a hybrid mode. On the other side, many C/S and CDN-based systems (e.g., Youku, Tudou, and LiveSky [30]) have integrated P2P techniques.

1.2 Cloud Computing and Mobile Internet

Since Amazon's launch of EC2 in 2006 and Apple's release of iPhone in 2007, Internet content distribution has demonstrated a strong trend of *polarization*:

- On one hand, great fortune has been invested in building heavyweight and integrated data centers ("heavy-cloud") all over the world. Recent years have witnessed great successes of cloud computing [3] (e.g., Amazon EC2, Google App Engine, Microsoft Azure, Apple iCloud, Aliyun, and OpenStack), big data processing (e.g., Apache Hadoop, Cassandra, and Spark), cloud storage (e.g., Amazon S3, Dropbox, Box, Google Drive, and Microsoft OneDrive), virtualization (e.g., VMware, VirtualBox, Xen, and Docker), and so forth. Based on these cloud platforms, today's content distribution systems can reliably host a huge amount of content, purchase any amount of ISP and CDN bandwidth on demand, and adaptively migrate content and schedule bandwidth, in order to achieve the economies of scale and the cost efficiency [2].
- On the other hand, end user devices have been growing more lightweight and mobile ("light-end"), as well as highly heterogeneous in terms of hardware, software, and network environments. The release of iPhone and iPad, together with the flourishing of Android Open Handset Alliance, have substantially extended the functions of mobile devices from traditional voice calls and text messages to almost all kinds of Internet applications. Besides, growth in mobile devices greatly outpaces that of personal computers. Nevertheless, most existing content distribution techniques are still geared for personal computers at the moment. When applied to mobile scenarios, they often exhibit unsatisfactory or even poor performance. Mobile devices have diverse sizes and resolutions of screens, support different formats of content and applications [23], and are quite limited in traffic quotas (while working in 2G/3G/4G modes), processing capabilities, and battery capacities [22], thus posing rigorous requirements on the traffic usage, energy consumption, speed, and latency of content distribution.

Although "heavy-cloud" seems on the other side of "light-end", it is not on the opposite of "light-end". In essence, the evolution of "heavy-cloud" is motivated by the requirements of "light-end", i.e., only with more heavy back-end clouds can front-end user devices become more light. A representative case is that every new generation of iPhone is lighter, thinner, and yet more powerful than its predecessors, while more relies on iCloud services. In the era of cloud computing and mobile Internet, cloud platforms have to collaborate more tightly with end user devices, and thus the gap between Internet content and mobile devices can be better filled.

1.3 Frontier Techniques

In recent years, various frontier techniques for Internet content distribution have emerged in both academia and industry. Below we review typical frontier techniques that are especially related to the "heavy-cloud versus light-end" scenarios:

- *Multi-CDN*. To mitigate the limitations of a single CDN, a content provider can simultaneously purchase resources from multiple CDNs and then allocate the resources on her own, as demonstrated in Fig. 1.3. A representative case of Multi-CDN is Hulu [1], a novel video content provider that makes use of three CDNs: Akamai, Limelight, and Level3. Additionally, Hulu in itself builds a relatively small-scale cloud platform for scheduling the resources from the three CDNs. With such efforts, Hulu effectively addresses the limitations of a single CDN in cost efficiency, available bandwidth, and ISP/geographical coverage.
- *Private BitTorrent* and *Bundling BitTorrent* are both extensions to the classical BitTorrent protocol. The former restricts BitTorrent's working within a more narrow, more homogeneous, and more active user group, e.g., the students in the same campus. By trading the user-group coverage for the authentication of users and shared content, private BitTorrent can remarkably enhance the engagement of users, the quality of shared content, and the performance of content distribution [24, 27, 31].
  By bundling a number of related files to a single file, the latter motivates BitTorrent users to stay longer online and share more data with others. Hence, the whole BitTorrent system can be boosted to a more healthy and prosperous status [5, 10, 26]. Typically, bundling BitTorrent combines a number of episodes of a TV series to a single, large file—this simple and useful idea has been adopted by numerous BitTorrent (and even eMule) websites.
- *Peer-to-Server&Peer* (*P2SP*). As illustrated in Fig. 1.4, P2SP allows and directs users to retrieve data from both peer swarms and content servers, so it is the integration of P2P and C/S. Till now, P2SP has been adopted by plenty of media streaming systems, such as Spotify, PPLive, PPStream, UUSee, and Funshion [6, 29]. It is worth noting that the implementation complexity of P2SP is usually higher than that of P2P or C/S. If not properly designed, P2SP can generate worse performance with higher overhead.

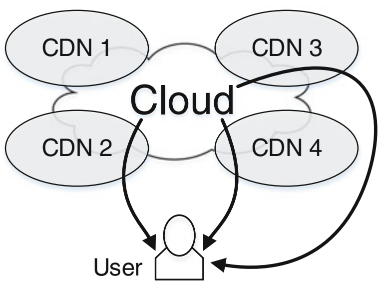


Fig. 1.3 Multi-CDN content distribution with a centralized cloud for scheduling the resources from multiple CDNs

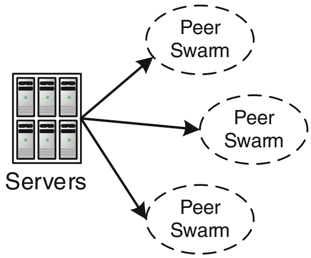


Fig. 1.4 P2SP content distribution by integrating P2P and C/S. (Reprinted with permission from [12].)

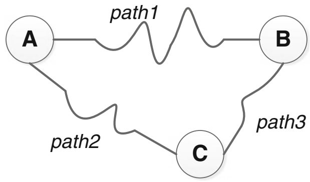


Fig. 1.5 Triangle inequality violation in the Internet, i.e., *path2* + *path3* < *path1*

Furthermore, P2SP can be extended to Open-P2SP [12], which outperforms P2SP by enabling users to retrieve data across heterogeneous protocols and content providers. For example, a user can simultaneously download from a BitTorrent swarm, an eMule swarm, an HTTP server, and a RTSP server. Representative examples of Open-P2SP include Xunlei, QQXuanfeng, FlashGet, Orbit, and QVoD. Naturally, Open-P2SP is very complicated and difficult to implement, since it involves not only technical problems but also business/copyright issues.

- *Detour routing* [25] originates from the pervasive *triangular inequality violation* phenomena of the Internet. As depicted in Fig. 1.5, suppose we want to deliver a file f from node A to node B in the Internet, and the shortest-hop path from A to B is *path1*. Theoretically, delivering f along *path1* should be the fastest. But in practice, because today's Internet is complicated by too many artificial factors, it may be faster if we select an appropriate intermediate node C to forward f from node A (along *path2*) to node B (along *path3*), which is referred to as detour routing. Therefore, we observe the triangular inequality violation phenomenon when the sum of two sides (*path2* + *path3*) of a triangle is smaller the third (*path1*). Particularly, when nodes A and B locate at different ISP networks, triangular inequality violations would frequently occur. In this case, detour routing will be an effective remedy for the defects of today's Internet. Representative examples of detour routing include "offline downloading" [19] and quality improving of online gaming [25].
- *Dynamic Adaptive Streaming over HTTP* (*DASH*). HTTP (web) servers are the earliest and most widely used in the Internet. They are originally designed for serving webpage requests rather than media streaming. However, due to their large scale and enormous popularity, HTTP servers are often used for media streaming in practice. To make HTTP servers more suitable for media streaming, web professionals

developed DASH as an extension to the original HTTP protocol.[1] Meanwhile, web clients need adjusting so that they can adaptively select an appropriate streaming bit rate according to real-time network conditions. At present, DASH remains an active research topic and possesses considerable space for performance optimization [8, 28].

In general, all the above efforts attempt to address current issues of Internet content distribution by extending or upgrading existing techniques and frameworks. In other words, they act as remedies for defects and seek for incremental improvements. As time goes on, the "heavy-cloud versus light-end" polarization of Internet content distribution may become more and more severe, and thus we need to explore innovative techniques and frameworks. This is the basic starting point of our research presented in this book.

1.4 Overview of the Book Structure

Based on comprehensive real-world measurements and benchmarks, we observe that existing content distribution techniques often exhibit poor performance under the settings of cloud computing and mobile Internet. To address the issue, this book investigates content distribution for mobile Internet with a cloud-based approach, in the hopes of bridging the gap between Internet content and mobile devices. In particular, we propose, design, and implement a series of novel traffic-saving, energy-efficient, high-speed, and delay-tolerant content distribution techniques and frameworks that automatically adapt to mobile application scenarios.

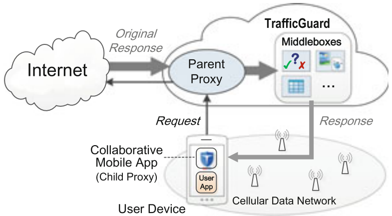
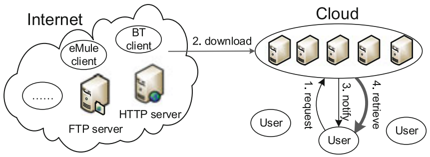
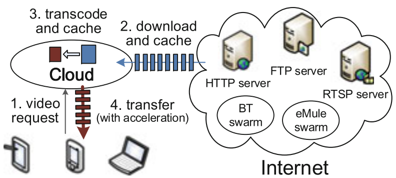
Besides theoretical and algorithmic contributions, our research pays special attention to its *real effect*. Specifically, we discover practical problems in real systems, solve the problems under real environments, and achieve real performance enhancements. Our research stands on real-world production systems such as TrafficGuard (i.e., a cross-app cellular traffic optimization platform of Baidu), QQXuanfeng (i.e., the major content distribution platform of Tencent), CoolFish (i.e., a video streaming system of the Chinese Academy of Sciences), and Dropbox (i.e., one of the world's biggest cloud storage services). A series of useful takeaways and easy-to-follow experiences are provided to the researchers and developers working on mobile Internet and cloud computing/storage.

The main body of this book is organized in four parts: *cloud-based cellular traffic optimization* (Chap. 2), *cloud-based mobile video distribution* (Chaps. 3, 4, and 5), *cloud-assisted P2P content distribution* (Chaps. 6 and 7), and *cloud storage-oriented content distribution* (Chaps. 8 and 9). Below we provide a brief overview of each part.

Part II: Cellular traffic is the most expensive among all Internet traffic, so it is first explored in Chap. 2. As the penetration of 3G/4G/5G data networks, cellular traffic optimization has been a common desire of both cellular users and carriers. Together

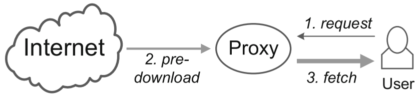
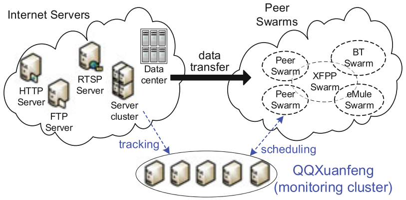
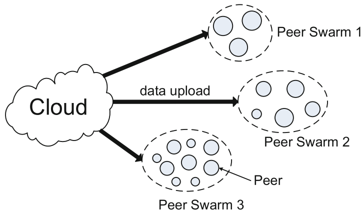
[1]DASH is called HTTP Live Streaming (HLS) by Apple and Smooth Streaming by Microsoft.

Table 1.1 Chapter structure

Title	System	Architecture	Publication
Cross-application cellular traffic optimization (Chap. 2)	百度手机卫士 shoujiweishi.baidu.com		NSDI'16 [17]
Cloud downloading for unpopular videos (Chap. 3)	Download Offline lixian.qq.com		*JCST'15* [16] ACM-MM'11 (Long Paper) [7]
Cloud transcoding for mobile devices (Chap. 4)	 xf.qq.com		NOSSDAV'12 [13]

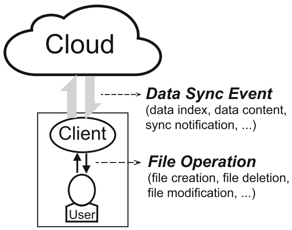
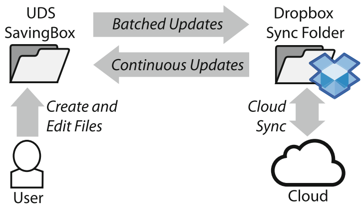
(continued)

Table 1.1 (continued)

Title	System	Architecture	Publication
Offline downloading: a comparative study (Chap. 5)			IMC'15 [19]
Cloud tracking or Open-P2SP (Chap. 6)	xf.qq.com www.xunlei.com		*TPDS'13* [12] ACM-MM'11 (Doctoral Symposium) [11]
Cloud bandwidth scheduling (Chap. 7)	xf.qq.com www.cool-fish.org		IWQoS'12 [20]

(continued)

Table 1.1 (continued)

Title	System	Architecture	Publication
Towards network-level efficiency for cloud storage services (Chap. 8)	Google Drive, Dropbox, iCloud, SkyDrive, IDrive sync, ubuntu one, 百度云, 360云盘, SugarSync		TCC'15 [32] (Spotlight Paper) IMC'14 [14] *JTST'13* [21]
Efficient batched synchronization for cloud storage services (Chap. 9)	Dropbox www.dropbox.com		Middleware'13 [18] *CCCF'14* [15] (*Cover Article*)