Profiling the Performance Anomalies of Multi-Source Media Downloading at Scale in the Wild

Xi Chen, Xinlei Yang, and Zhenhua Li

Tsinghua University, China
{chenxi198909, yangxinlei19971105, lizhenhua1983}@gmail.com

Abstract. As one of the most fundamental and pervasive Internet services, media file downloading has undergone several generations of enabling technologies. Unfortunately, the performance today is still far from satisfactory. As the state-of-the-art approach to accelerating media file downloads, multi-source downloading enables user clients to utilize multiple data sources and various content delivery techniques. However, without careful designs, multi-source downloading can result in worse performance with higher overhead, referred to as an *anomaly*.
This paper conducts the first empirical study to quantitatively understand the performance anomalies of multi-source media downloading, based on the production logs of a large-scale system serving 179M media file downloads for 37M users (including both PC and mobile users) per day. We reveal the characteristics and root causes of manifold anomalies with regard to seven types of data sources. In particular, 23% of the downloads accelerated by using multiple data sources become slower than the original single-source downloading, and there are sweet spots between the number of data sources used and the download speed. Also, we exploit some unconventional metrics (*e.g.,* diversity of participation time) to explain some counter-intuitive anomalies. Accordingly, we provide a series of practical and applicable implications to effectively address the anomalies.

1 Introduction

As one of the most fundamental and pervasive Internet services, media file downloading has undergone several generations of enabling technologies, including traditional client-server (C/S) models, content delivery networks (CDNs), peer-to-peer (P2P) networks, and cloud-based techniques. However, today's network infrastructure is in dire straits to catch up with the continuous growth in the user base, the data throughput (which is reported to be increasing by double digits every year) and energy consumption [1–3]. Consequently, the performance of media file downloading, in terms of both download speed and success rate, is still far from satisfactory [4–6].

To accelerate the media file downloads, *multi-source downloading* has been adopted by many Internet service and content providers today as the state-of-the-art downloading approach. With multi-source downloading, a user client simultaneously fetches different parts of the requested file from multiple data sources through various content delivery techniques and protocols. For example, in a P2P downloading system, a user client can maintain tens of TCP connections with different peers to fetch different data chunks of the requested file concurrently. After P2P is extended to P2SP (peer-to-server&peer), the user client can also fetch data chunks from dedicated servers, in

MOBIMEDIA 2018, June 21-22, Qingdao, People's Republic of China

DOI 10.4108/eai.21-6-2018.2276556

addition to the peer links [7–9]. Intuitively, multi-source downloading can effectively improve the performance of media file downloading, especially for large-sized files. The efficacy is indeed confirmed by previous studies [10, 11].

However, as time goes on, the interactions between a user client and the multiple data sources become far more complex today than those were a few years ago. If not designed properly, multi-source downloading can result in worse performance with higher network and monetary overhead [12–14], in comparison to the original single-source downloading. This phenomenon is referred to as an performance *anomaly* of multi-source downloading. Although anomalies are constantly experienced by end users and are known to the research community, there is no systematic effort towards comprehensively understanding the anomalies in real-world systems at scale, let alone guiding users and developers to address the anomalies.

This paper conducts the first empirical study to quantitatively understand the anomalies of multi-source media downloading, including their characteristics, root causes, and implications for the design of relevant systems. Our study is based on M-Downloader, a large-scale multi-source downloading system operated by a major Internet content provider. M-Downloader serves 179M (million) file download requests (including about 71% media files) issued from 37M users (including both PC and mobile clients) on a daily basis. For each request, M-Downloader can use up to seven types of data sources, including original C/S links (mostly in HTTP and FTP), free C/S mirrors, *charged* C/S mirrors, free CDNs, *charged* CDNs, ISP caches, and P2P data swarms. Here charged means that M-Downloader must pay for the upload traffic of the used data sources.

M-Downloader schedules data sources using a *progressive* procedure. Immediately after receiving a user's download request, the M-Downloader client, installed on the user's device, starts to download the requested file content from the original data source. Meanwhile, the back-end cloud of M-Downloader attempts to help the client accelerate the download. It searches the available data sources and notifies the client to upgrade the initial single-source downloading to multi-source downloading. This progressive procedure provides the baseline for performance analysis, as we can comparatively study the download performance using multiple data sources versus the original source, and identify the performance anomalies.

Based on the analysis of the dataset from M-Downloader, we reveal manifold performance anomalies of multi-source media downloading and dissect common misunderstandings among both users and developers. To understand the root causes of anomalies (including some counter-intuitive ones), we exploit certain unconventional metrics (*e.g.,* diversity of participation time, abandonment rate, and estimated remaining download time), together with common metrics like download speed and time, number of data sources used, file size and popularity, *etc.* Accordingly, we provide a series of practical and applicable implications for effectively addressing the anomalies. Our major findings and implications are summarized as follows:

- *Popularity does not mean abundance*. It is commonly believed that the more popular a file is, the more data sources exist on the Internet. Thus, downloading a more popular media file would be faster and more likely to succeed. Nevertheless, our dataset reveals that the file popularity is not strongly correlated with the abundance of data sources (the correlation coefficient is merely 0.16). This is one of the

fundamental causes that make the performance of multi-source media downloading unstable and unpredictable. Therefore, using multiple data sources for media downloading is not a trivial panacea but needs in-depth investigation.

- *The diversity of participant time of data sources influences the performance of multi-source media downloading*. M-Downloader accelerates a media file download by upgrading the original single-source to multi-source downloading. Surprisingly, we observe that 23% of the accelerated downloads become slower or even fail. This counter-intuitive phenomenon is explained by the large *diversity of participation time* of data sources (refer to Eq. 1 for its formal definition). In other words, the download process mainly relies on a small subset of data sources that participate for a long period, while being distracted by other short-period data sources. Thus, to achieve high performance, the data sources need to be carefully probed and selected. Specifically, once multi-source downloading becomes slower than single-source downloading and the diversity of participation time is larger than the threshold (0.25), multi-source downloading should be degraded.
- *Overusing data sources hurts the download performance*. One common practice to improve download performance is adding additional data sources. However, we show that overusing data sources hurts the download performance. For example, when multi-source downloading outperforms the original single-source downloading, the speed growth is 331 KBps when 5 data sources are used while only 120 KBps when 22 data sources are used. In addition, using more data sources brings more potential bottlenecks (in the multiple data connections), especially for the downloads of small files. Based on our dataset, we quantify the sweet spots between the number of data sources used and the download speed by taking the file size and the diversity of data sources into account.

Roadmap. The reminder of the paper is organized as follows. § 2 describes the working principle and dataset organization of M-Downloader. § 3 presents the in-depth analysis of multi-source media downloading. The related work is reviewed in § 4, and we conclude the paper in § 5.

2 System and Dataset

In the M-Downloader system, when a user wants to acquire a file from a data source (say S_0), she/he issues a *download request* to the back-end cloud through the front-end client and get the file from this file firstly. Once receiving the download request, the cloud first maps S_0 onto all the other data sources (say S_1, S_2, $\cdots$, S_n) that provide the same file content, and then randomly picks a few (say m) data sources for the client. After getting the m data sources ($m > 0$), the client also randomly picks a few (say c) data sources to set up TCP/UDP connections with. Once a TCP/UDP connection is successfully established, the client starts downloading a chunk of the wanted file from the corresponding data source. At this time, the download is upgraded from single-source downloading to multi-source downloading. Finally, when the download task is

successful, timed out, or abandoned by the user, the client sends a *log report* to the cloud which records detailed information of the data sources used during the download.

To understand the performance characteristics of multi-source media downloading, we study a large-scale dataset collected from the M-Downloader system. The dataset contains the complete running logs of the system during a whole week (July 13–19, 2015), involving 1,364,122,406 download tasks (with 71.1% media download tasks), 57,538,801 users, and 9,827,109 unique files. Among these download tasks, the majority (59%) utilized multiple (≥ 2) data sources, and the remainder (41%) only used the original (single) data source. From the dataset, we find that M-Downloader can use up to seven types of data sources for multi-source downloading. The seven types of data sources cover almost all popular content delivery techniques and protocols at present, and they are each briefly profiled as follows:

1. *Original C/S data sources* mostly transfer data using HTTP and FTP protocols, and typically upload data by a single server (cluster).
2. *Free C/S mirrors*. For a file originally provided by a C/S data source, when it can also be downloaded from other C/S data sources in a free manner, these other data sources are called free C/S mirrors.
3. *Charged C/S mirrors* only serve the content delivery systems who have paid. As for M-Downloader, once its clients download content from charged C/S mirrors, it has to pay for the network traffic or bandwidth.
4. *Charged CDN data sources*. CDN optimizes the performance of content distribution by strategically deploying edge servers at multiple locations. An end user usually obtains a copy of content from a nearby edge server.
5. *Free CDN data sources* are deployed by M-Downloader, so they are only free to M-Downloader users.
6. *ISP caches* are deployed by ISPs to reduce the expensive cross-ISP network traffic during file downloads. Once a file is cached in ISPs' server, subsequent requests for the file are directly satisfied by the cached copy.
7. *P2P data sources* mostly transfer data using BitTorrent and eMule protocols. They organize numerous end-user devices into peer-to-peer data swarms in which shared content is directly delivered among interested peers.

3 Anomaly Analysis

Having understood the working principle and dataset organization of M-Downloader in § 2, in this section we investigate why, when, and how multi-source (media) downloading generates performance anomalies from an empirical perspective.

3.1 File Popularity vs. Data-Source Abundance

File popularity denotes how many times a file was requested for in one week. As we all know, the download system (*e.g.,* Thunder, iTudou, *etc.*) actively maintains a Data

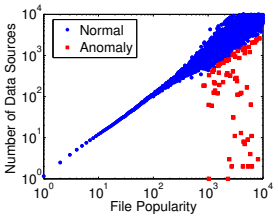

Fig. 1: Relationship between file popularity and number of data sources.

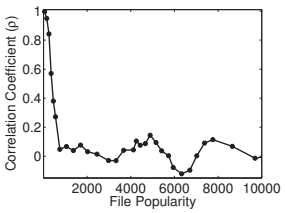

Fig. 2: Correlation between file popularity and number of data sources.

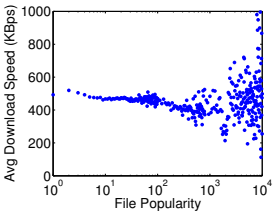

Fig. 3: Relationship between file popularity and average download speed.

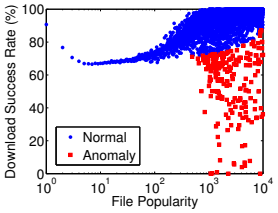

Fig. 4: File popularity vs. download success rate.

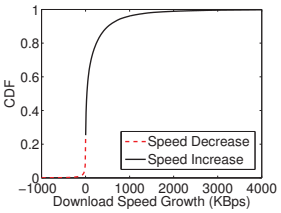

Fig. 5: CDF of download speed growth.

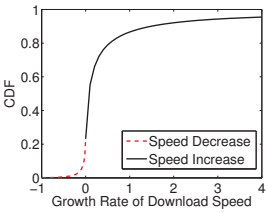

Fig. 6: CDF of growth rate of download speed.

Source Map where every file is associated with a unique identifier, *i.e.,* the file hash. Hence, multiple files are considered identical as long as they have the same file hash, and the file popularity is calculated based on the file hash.

People usually think that the more popular a file is, the more data sources exist in the Internet for this file, and thus downloading this file would be faster and more likely to succeed. Nevertheless, our measurement results reveal that this is not always the truth. As illustrated in Fig. 1, the number of data sources is generally proportional to the file popularity, which is especially evident when the file popularity falls below 100. However, when the file popularity exceeds 1000, many "anomalies" turn up as red points in Fig. 1.

Quantitatively, we draw the correlation coefficient (ρ) between file popularity and number of data sources in Fig. 2. Here $\rho = \frac{Cov(X,Y)}{\sqrt{D(X)}\sqrt{D(Y)}}$, where X is the file popularity, Y is the number of data sources, Cov is the covariance which is calculated as $E(XY) - E(X)E(Y)$, and $D(X)$ is the variance of X. Obviously, when the file popularity is below 100, ρ is close to 1.0. As the file popularity increases, ρ dramatically decreases to between -0.1 and 0.3, making the corresponding download performance highly unstable and even poor. This is confirmed by the relationship between file popularity and average download speed as shown in Fig. 3, as well as the relationship between file popularity and download success rate as shown in Fig. 4. Across all file popularities, the overall ρ is merely 0.16. This is why multi-source downloading does not work as a trivial "panacea" (for media download optimization) and deserves in-depth investigation.

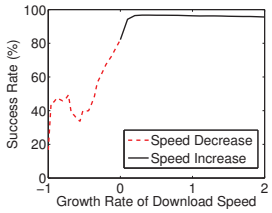

Fig. 7: The success rate of different speed change rate.

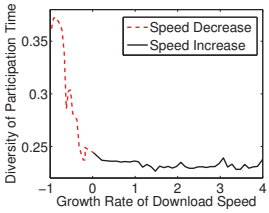

Fig. 8: Diversity of participation time *vs.* growth rate of download speed.

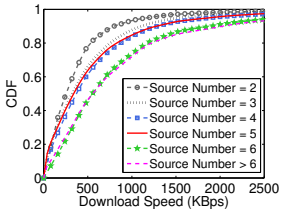

Fig. 9: CDF of download speed for different numbers of data sources used.

3.2 Multi-Source vs. Single-Source Downloading

Our collected dataset shows that in a whole week, 0.57 billion downloads are using a single data source with an average speed of 237 KBps and a success rate of 96.7%, while 0.8 billion are using multiple (2.94 in average) data sources with an average speed of 728 KBps and a success rate of 98.4%. This general statistic comparison seems to present that multi-source downloading definitely outperforms single-source downloading. Nevertheless, detailed examination on the effect of upgrading (from original single-source downloading to multi-source downloading) reveals unexpected performance degradation.

The performance degradation first appears in the download speed. Fig. 5 shows the distribution of the download speed growth (which can be negative) when original-source downloading is upgraded to multi-source downloading. Surprisingly, after the upgrading phase, 23% of downloads (depicted as the red dashed curve in Fig. 5) become slower. Additionally, approximately 37% of downloads are trivially accelerated by almost zero KBps. To make things clearer, we plot the distribution of the growth rate of download speed in Fig. 6. Once again, we notice that nearly 37% of downloads are speeded up by a small percentage.

In addition, the performance degradation also appears in the download success rate. As indicated in Fig. 7, there is an obviously positive correlation between the download success rate and the acceleration effect. Specifically, when a download is slightly accelerated, its success rate would exceed 80%; when a download is considerably accelerated, its success rate would be as high as 94%. On the contrary, when a download is decelerated, its success rate can hardly reach 80% (shown as the red dashed curve in Fig. 7), sometimes even falling below 50%.

The above two paragraphs reveal the second counter-intuitive phenomenon in our study, *i.e.,* multi-source downloading is sometimes worse than the original single-source downloading in terms of both download speed and success rate. Seeking for a reasonable explanation to this phenomenon, we examine a number of metrics and eventually note that it is attributed to a large *diversity of participation time* of data sources, which

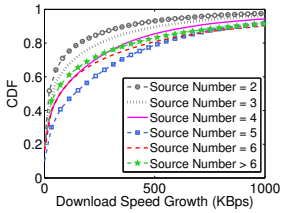

Fig. 10: CDF of download speed grow -th for different # of data sources used.

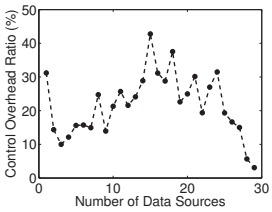

Fig. 11: Control overhead ratio *vs.* number of data sources used.

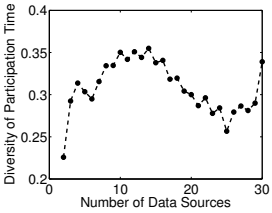

Fig. 12: Diversity of participation time for different # of data sources used.

is measured by the standard deviation divided by the range, *i.e.,*

$$\frac{\sqrt{\frac{1}{N}\sum_{i=1}^{N}(T_i-\overline{T})^2}}{T_{max}-T_{min}}, \tag{1}$$

where T_i denotes the participation time of the i-th data source, and $\overline{T}$ denotes the average participation time of the data sources used. As described in § 2, when multiple data sources are used during a download, all these data sources do not upload data all the time. Instead, each data source uploads data for a specific participation time. Hence, the diversity of participation time basically reflects the heterogeneities of contributions among the data sources used.

Fig. 8 quantifies the obviously negative correlation between the diversity of participation time and the growth rate of download speed. Specifically, when the diversity of participation time is small (< 0.2), a download task can usually benefit from using multiple data sources. In fact, this means that all data sources used are making similar contributions. On the other hand, once the diversity of participation time is large (> 0.25), the user can hardly benefit from multi-source downloading. Essentially, the download process mainly relies on a small subset of data sources that participate for a long period, while being distracted by other short-period data sources. Thus, to achieve effective acceleration by multi-source downloading, the data sources need to be carefully probed and selected. Specifically, once multi-source downloading becomes slower than single-source downloading and the diversity of participation time is larger than the threshold (0.25), multi-source downloading should be degraded.

3.3 How Many Data Sources Should Be Used

One common practice to improve media download performance is adding additional data sources. However, in this part we will show that overusing data sources hurts the download performance. Besides, we will quantify the sweet spots between the number of data sources used and the download speed.

First, we wonder how the download speed grows as more data sources are used. Fig. 9 and Fig. 10 shows the distribution of download speed and download speed

Table 1: Control overhead ratio of download tasks with different file size.

File Size (MB)	**<1**	**1-10**	**10-100**	**100-300**	**300-1024**	**>1024**
Control Overhead Ratio (%)	92.3	6.0	3.5	2.8	3.0	2.1

Table 2: Sweet spots among the file size, recommended number of data sources, diversity of participation time, and download speed.

File Size (MB)	**Recommended Number of Data Sources**	**Diversity of Participation Time**	**Download Speed (KBps)**
<1	1	0	206
1-10	5	0.25	811
10-100	9	0.32	1370
100-300	7	0.32	1302
300-1024	6	0.32	1078
>1024	4	0.40	977

growth, respectively. In Fig. 9, we notice that when more than 6 data sources are used, the acceleration effect will be trivial. In Fig. 10, we further figure out the average download speed growths when different numbers (including 2, 3, 4, 5, 6, and more than 6) of data sources are used: 126 KBps, 171 KBps, 246 KBps, 338 KBps, 332 KBps, and 275 KBps. Generally speaking, the maximum download speed growth is achieved when 5 or 6 data sources are used.

Next, we examine the control overhead ratio (*i.e.,* the ratio of the control overhead traffic over the file size) of download tasks when different numbers of data sources are used to download files in different sizes. The results are plotted in Fig. 11 and Table 1. As shown in Fig. 11, the control overhead ratio is remarkable when a single data source is used or 15 to 20 data sources are used. Why does single-source downloading generate such a high control overhead ratio? The answer can be found when Table 1 is also taken into consideration. In fact, a large portion of single-source downloading tasks are for small files whose size is less than 1 MB, which leads to a high control overhead ratio.

In § 3.2, we have understood that the diversity of participation time greatly affects the performance of multi-source downloading. Following this understanding, we further scrutinize the diversity of participation time when different numbers of data sources are used. The corresponding results are presented in Fig. 12, from which we find that when the number of data sources used is between 1 and 6, the diversity of participation time is relatively small. This explains our aforementioned observation (from Fig. 10) that the maximum download speed growth is achieved when 5 or 6 data sources are used.

From all the above analysis, we conclude that the performance of multi-source media downloading heavily relies on the number of data sources used coupled with the file size and diversity of participation time. Hence, we comprehensively examine the relationship between the download speed and the three impact factors, and then list in Table 2 the recommended number of data sources used in various cases. In other words, Table 2 quantifies the sweet spots between the number of data sources used and the download speed by taking file size and diversity of participation time into account. In

particular, as the file size increases, the recommended number of data sources used and the download speed all exhibit a bell-shaped ($\bigcap$) variation pattern, coupled with the slight change of the diversity of participation time. The two bell-shaped variations consistently suggest that the designers of multi-source media downloading should make a considerate tradeoff among manifold impact factors (*e.g.,* by following Table 2) to achieve desirable performance.

4 Related Work

As the state-of-the-art approach to accelerating media file downloads, multi-source downloading has attracted wide attention in recent years, particularly in the following novel forms of cloud-based CDN, hybrid CDN-P2P, and open-P2SP.

Cloud-based CDN. CDN is traditionally employed by cloud service providers to accelerate their content delivery. Recently, some CDN service providers have begun to enhance their content delivery performance by purchasing resources from clouds [15, 16]. In this way, bandwidth/storage resources pervasively existing in the Internet could be fully and collaboratively utilized.

Hybrid CDN-P2P. Yin *et al.* designed and deployed LiveSky, a hybrid CDN-P2P media streaming system [17]. By benefiting from both CDN and P2P, its clients work well even under bandwidth constraints. Besides, Aditya *et al.* comprehensively evaluated another hybrid CDN-P2P system called Akamai NetSession, and proposed a method for reliable client and resource accounting [18].

Open-P2SP. As a generalized and extended mode of P2SP, open-P2SP integrates various third-party servers, content, and data transfer protocols across the Internet. Li *et al.* presented the key challenges, practical designs, and real-world performance of an open-P2SP system named QQXuanfeng [19]. Besides, Dhungel *et al.* made a measurement-based study of Xunlei, perhaps the biggest open-P2SP system at present [20].

5 Conclusion

In this paper, we reveal manifold performance anomalies of multi-source media downloading by analyzing a large-scale dataset provided by the M-Downloader system. We investigate their characteristics, root causes, and implications for addressing the performance anomalies. In particular, we exploit some unconventional metrics to understand the root causes of some surprising anomalies, and (for the first time) quantity the sweet spots between the number of data sources used and the download speed. Our work provides solid experiences and helpful heuristics to the designers of relevant systems.

6 Acknowledgements

This work is supported in part by the High-Tech Research and Development Program of China (“863China Cloud” Major Program) under grant 2015AA01A201, the National Natural Science Foundation of China (NSFC) under grants 61471217, 61432002, 61632020 and 61502271.

References

1. Y. Zaki, J. Chen, T. Pötsch, T. Ahmad, and L. Subramanian, "Dissecting web latency in Ghana," in *Proc. of ACM IMC*, 2014.
2. X. Ji, Y. He, J. Wang, K. Wu, D. Liu, K. Yi, and Y. Liu, "On improving wireless channel utilization: A collision tolerance-based approach," *IEEE Transactions on Mobile Computing*, vol. 16, no. 3, pp. 787–800, 2017.
3. X. Zheng, Z. Cao, J. Wang, Y. He, and Y. Liu, "Interference resilient duty cycling for wireless sensor networks under co-existing environments," *IEEE Transactions on Communications*, vol. PP, no. 99, pp. 1–1, 2017.
4. S. Sundaresan, N. Feamster, R. Teixeira, and N. Magharei, "Measuring and mitigating web performance bottlenecks in broadband access networks," in *Proc. of ACM IMC*, 2013.
5. T. Flach, E. Katz-Bassett, and R. Govindan, "Diagnosing slow web page access at the client side," in *Proc. of ACM CoNEXT Student Workshop*, 2013.
6. S. Sundaresan, W. De Donato, N. Feamster, R. Teixeira, S. Crawford, and A. Pescapè, "Broadband Internet performance: a view from the gateway," in *Proc. of ACM SIGCOMM*, 2011.
7. Y. Huang, T. Z. Fu, D.-M. Chiu, J. Lui, and C. Huang, "Challenges, design and analysis of a large-scale P2P-VoD system," in *Proc. of ACM SIGCOMM*, 2008.
8. C. Wu, B. Li, and S. Zhao, "On dynamic server provisioning in multichannel P2P live streaming," *IEEE/ACM Transactions on Networking*, vol. 19, no. 5, pp. 1317–1330, 2011.
9. Y. Sun, F. Liu, B. Li, and X. Zhang, "Fs2you: Peer-assisted semi-persistent online storage at a large scale," in *Proc. of IEEE INFOCOM*, 2009.
10. C. Jiang, Y. Chen, Y. Ren, and K. R. Liu, "Maximizing network capacity with optimal source selection: A network science perspective," *IEEE Signal Processing Letters*, vol. 22, no. 7, pp. 938–942, 2015.
11. G. Nam and K. Park, "Analyzing the effectiveness of content delivery network interconnection of 3G cellular traffic," in *Proc. of The ACM 9th International Conference on Future Internet Technologies*, 2014.
12. "My P2P download is slower than my regular one," http://www.dslreports.com/forum/r9669266-My-P2P-download-is-slower-than-my-regular-one.
13. M. Adler, R. K. Sitaraman, and H. Venkataramani, "Algorithms for optimizing the bandwidth cost of content delivery," *Computer Networks*, vol. 55, no. 18, pp. 4007–4020, 2011.
14. T. Bektas and O. Ercetin, *CDN Modeling*, 2014.
15. V. K. Adhikari, Y. Guo, F. Hao, and V. Hilt, "A tale of three cdns: An active measurement study of hulu and its cdns," in *Computer Communications Workshops*, 2012, pp. 7–12.
16. Z. Li, Y. Dai, G. Chen, and Y. Liu, "Content distribution for mobile Internet: A cloud-based approach," 2016.
17. 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 *Proc. of ACM Multimedia*, 2009.
18. P. Aditya, M. Zhao, Y. Lin, A. Haeberlen, P. Druschel, B. Maggs, and B. Wishon, "Reliable client accounting for P2P-infrastructure hybrids," in *Proc. of USENIX NSDI*, 2012.
19. Z. Li, Y. Huang, G. Liu, F. Wang, Y. Liu, Z.-L. Zhang, and Y. Dai, "Challenges, designs, and performances of large-scale open-P2SP content distribution," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 11, pp. 2181–2191, 2013.
20. P. Dhungel, K. Ross, M. Steiner, Y. Tian, and X. Hei, "Xunlei: Peer-assisted download acceleration on a massive scale," in *Proc. of PAM*, 2012.