Architecture and Download Behavior of Xunlei:A Measurement-Based Study

Min ZHANG
Beijing Jiaotong University
Beijing, China, 100044
mia.minzhang@gmail.com

Wolfgang JOHN
Chalmers University of Technology
Goteborg, Sweden
wolfgang.john@chalmers.se

Changjia CHEN
Beijing Jiaotong University
Beijing, China, 100044
changjiachen@sina.com

***Abstract*—this paper presents a measurement-based study of Xunlei, a young but increasingly popular P2P application currently mainly used in China. Due to its proprietary protocols and data encryption, packet-level data was analyzed in order to infer the network architecture and the superior download behavior, Xunlei's main key to success. We identify Xunlei's core-server layer with a clear division of tasks. We found that Xunlei increases download speeds by applying multi-source technology in an effective way, which includes (inappropriate) direct linking. Finally, we reveal that Xunlei utilizes UDP as main transport protocol both for bulk file transfer and IPTV streaming. As the first comprehensive study on Xunlei, this paper provides new insights for researchers and application designers, and contributes to a better understanding of the Internet.**

Keywords- Internet measurement; Xunlei; P2P; Bandwidth theft

I. Introduction

Peer-to-Peer applications may well account for a majority of modern Internet traffic [1], [2]; meanwhile, new P2P applications emerge rapidly on the Internet. In this paper, we study an interesting P2P application: the Xunlei download manager. The results might contribute to a better understanding of modern Internet and also provide new insights to designers of network application.

Today, Xunlei is probably the most popular file-sharing and streaming-media application in China. According to records on Wikipedia, over eighty million users installed Xunlei and its web site attract over fifty million hits per day; even more impressive are the user numbers of up to 1,880 millions claimed on the official Xunlei website. Xunlei attracts users due to great performance in searching files and downloading them with rapid download rates. It not only has all features of other download tools, such as searching, publishing, downloading, but it also aims to integrate resources on the Internet and transfers all kinds of files for users as fast as possible with multi-source technology. This is done by supporting file sharing and streaming protocols such as BitTorrent, eMule, FTP, and RTSP. Besides downloading files, Xunlei also provides IPTV functionality which is embedded in web pages, which makes it convenient for users to watch movies and TV programs. Due to its popularity, Xunlei even attracts attention of big media companies such as Google, Geyuan Ventures and IDGVC. Although Xunlei does not provide an official English version (yet) and only runs on Windows machines, it gained significant popularity in regions outside China as well, mainly due to Chinese migrants [1]. Furthermore, translated and ported versions are likely to be available soon, eventually allowing Xunlei to spread worldwide.

Besides regional limitations, we consider that Xunlei's proprietary nature and use of data encryption are the main cause for the very limited attention of the scientific community. In this paper, we seek to fill this gap by continuing an earlier study [3] in order to provide a more detailed study on Xunlei.

A. Resource searching and sharing mechanisms

We want to introduce Xunlei's resource searching policy and contrast it to traditional resource searching mechanisms, as observed in related P2P applications:

1. Client/Server (C/S) systems:
   - *Traditional Web downloads*: A central server provides content to connected clients directly.
2. Peer-to-Peer (P2P) systems:

By organizing own overlay networks, peers can exchange content among themselves:

   - *Gnutella*: Flooding querying for finding the targeted file.
   - *BitTorrent*: Uses torrent files and BitTorrent servers as querying agencies.
   - *EDonkey*: Uses eDonkey servers as querying agencies.

3. Xunlei resource system:

Different from C/S and P2P systems, Xunlei not only indexes files of peers (Xunlei's own overlay network), but also utilizes resources of other servers. In this paper, we classify download resources into three groups:

- *Internal server*: a server belonging to the web site from which users request target files;
- *External server*: a server not belonging to the web site from which users request target files;
- *Peer*: individual Xunlei client hosting the target files;

B. Direct Linking (Leeching)

While internal server and peer download match previous resource searching/sharing systems, exploitation of external servers is regarded as bandwidth theft in form of *direct* *linking*. Direct linking [4] is also known as inline linking or leeching, and is a stealthy behavior of taking advantage of web sites by getting free download services. A common form of direct linking is the use of linked images, from one web site into a web page belonging to a second web site. The second web site is said to leech if such links are not agreed upon in advance. In Xunlei, we define direct linking as follows: when a user uses Xunlei to download a file from web site A (the internal server), we regard the client as leecher if it downloads files from further, external servers besides the internal server and peers. This type of leeching explains Xunlei's high download rates. However, direct linking is controversial since it can do harm to advertisement profits of the external sites.

C. Contributions of this Paper

First, we use recorded Xunlei traffic to infer the architecture of the Xunlei core-system, by disclosing its server structure and client workflow. We then analyze the download behavior of Xunlei by thorough observation of a series of experimental download processes for both bulk and streaming data. The main results of the experiments are as follows:

- We discover the complex server infrastructure of Xunlei and identifying good cooperation and clear division of tasks between Xunlei core servers.
- We observe that Xunlei provides multiple resources for speeding up download processes, including exploitation of external servers by (inappropriate) direct linking.
- We show that Xunlei's multi-source download manager works effective, introducing little download-redundancy.
- We reveal that Xunlei mainly utilizes UDP as transport protocol for both, bulk file-sharing and IPTV steaming.

The remainder of the paper is organized as follows. We introduce related work in section II. Section III describes the experiment and measurement setup. Based on the resulting data sets, the following section presents the architecture of Xunlei. The download behavior is then described in section
V. Finally, section VI summarizes and concludes the paper.

II. Related Work

To our knowledge this work is the first comprehensive study on Xunlei, while there have been many measurement studies for other P2P applications. In 2002, Saroiu et al. [5] shows that there is heterogeneity and a lack of cooperation across peers participating in Napster and Gnutella by measuring end-user hosts in both networks. In Miller [6], a characterization of types of files in the Gnutella network by analyzing captured data was provided. The study showed that multimedia files are the most common types for which users search in Gnutella.

Recently, measurement studies focused more on the increasing popular eMule and BitTorrent systems. Yang et al. [7] describe a valuable measurement study of eDonkey, providing an evaluation of capacity and characterization of the distributions of servers, clients and files. A comprehensive measurement study of Kad is undertaken in Zhou et al. [8]. It not only describes the structure and processes of the Kad-protocol in detail, but also evaluates the performance of Kad with the focus on the data item distribution process.

Although P2P applications mentioned above have open protocols (and sometimes even publicly available source code), newly emerging applications tend to use proprietary communication protocols to prevent detection or filtering by network or content owners. Joost, a popular distributing online video system, is one example. Nevertheless, Hall et al. [9] provided a good description of Joost including application-, network-, and peer-behavior. Liang et al.[10] is a measurement study on KaZaA, another encrypted file-sharing application. It describes a measurement setup and infers KaZaA's overlay structure and dynamics, its neighbor selection, use of dynamic port numbers, etc. Finally, Baset and Schulzrinne [11] analyze key functions of Skype such as login, NAT and firewall traversal, call establishment and media transfer.

III. Measurement Setup

The measurement node is a 1.8GHz computer with a 100Mb/s NIC which is behind a network address translator (NAT) in Beijing Jiaotong University LAN. All measurements are taken on this node. For the experiments in this paper Xunlei version 5.7.4.401. was used.

A. Experiments and Data Collection

We carried out two kinds of experiments. The first one should help to understand Xunlei's architecture. A packet-based analysis of initial experiments revealed three communication phases (login, idling and downloading) between Xunlei servers and clients. We then recorded and analyzed all packets for each specific phase with the help of WireShark. The results of this experiment are presented in section IV.

The second experiment focuses on the download behavior of Xunlei. In order to evaluate how Xunlei servers reply to heterogeneous download requests, we download different kinds of files and recorded entire sessions. The chosen target files vary in file-types, sizes and content. As a result, section V provides a packet-level analysis regarding download resources, redundancy and data transfer properties of bulk-files downloads and IPTV streaming.

IV. Resutls: Architecture of Xunlei

As a piece of popular commercial download software, Xunlei really brings great download experiences to ordinary users. In order to reveal the mechanisms behind this successful product, we started by collecting traffic of complete Xunlei sessions. We also took advantage of information in Xunlei's log files. With the help of reverse DNS lookups it was possible to identify Xunlei servers, which then have been filtered from the raw data based on IP addresses. The filtered dataset finally allowed us to reveal the core server infrastructure of Xunlei, which we contrast to eDonkey servers in order to achieve a better understanding. After that, in subsection IV-B, a brief description of the client-server communication is provided.

A. Core Servers

Xunlei differs from eDonkey’s client/server architecture [12]. First of all, the number and IP addresses of eDonkey servers change frequently, while those of Xunlei servers are static. In our experiments, forty-six IP addresses are identified as Xunlei servers. Most of these servers are located in three provinces of China according to *www.ip2location.com*.

Furthermore, Xunlei servers are structured in a more complicated and systematical way than those of eDonkey. In the eDonkey network, each server is mainly in charge of indexing files for clients; but Xunlei servers cooperate effectively with clear-cut divisions of work. The following list includes dedicated servers identified:

- Client initialization
- Session tracking
- Nodes server directory
- Nodes server
- Resource management (providing resource reports with external servers and peers to clients)
- Indexing advertisements
- Storing advertisements
- Picture server
- Virus Scan
- Statistics
- Keep-alive server

For example, at startup Xunlei clients initialize sessions at initialization and tracking servers, followed by contacting and a couple of servers dedicated to provide advertisements.

B. Client-Server Communication

Xunlei uses both TCP and UDP as transport protocol during its operation. Even though there are some default ports used by clients for login or downloading, Xunlei can dynamically change port numbers, e.g. falling back on standard ports such as 80, 8080 and 443. The Client-server communication in Xunlei can be divided into three phases:

- *Login*: When establishing a connection to the Xunlei network, the client performs a login sequence with Xunlei servers. After initiating the session with the *client initialization server* via TCP and the *session tracking server* on UDP/4000, a list of available *nodes servers* is requested at the *nodes server directory* via UDP. If the request is unreplied, the continues to connect to another instance of the *nodes server directory*. Once a *nodes server* is successfully announced to the client, this is reported to the *session tracker* and the login procedure is finished.
- *Downloading*: The network system of Xunlei is illustrated schematically in Fig.1 in order to outline the download mechanism. When a client ('Bob') starts a download from one specific web site ('server 1') by requesting an URL (1), the client launches an encrypted TCP connection to the Xunlei resource management server (2). This request is always preceded by a DNS lookup of a fixed hostname (hub5sr.sandai.net), which leads us to suspect that a form of DNS load balancing is used to access Xunlei's resource management infrastructure. The contacted resource management server then identifies additional sources for the requested data by translating the URL into a unique hash value and provides Bob with the resulting 'resource report' including further locations where the requested file could be downloaded (2). Bob can then download the data in parallel using Xunlei's multi-source technology (3), which we will further analyze in section V.
- *Idling*: During the idling state, Xunlei client keeps regular interaction with servers besides updating advertisements. This includes ICMP messages every 10 seconds to the *keep-alive server* and UDP updates to the *nodes server* and the *session tracker* in intervals between 45 and 300 seconds. These are heartbeat packets used for updating and reporting the client status.

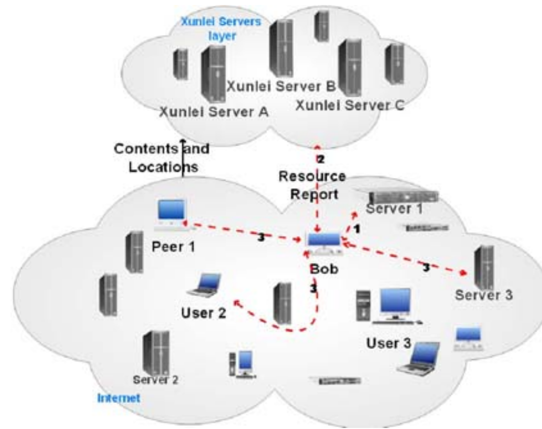

Figure 1. The Xunlei Architecture

V. RESULTS: DOWNLOAD BEHAVIOR

After section IV provided a general understanding of Xunlei’s infrastructure and workflows, we are now interested in the resource allocation provided by Xunlei core servers. This is also considered to be the main selling feature of Xunlei: multi-source technology exploiting abundant resources in order to speed up the download process. Our analysis shows that various supplement resources are returned by Xunlei when a client requests files. In the following subsections, we first evaluate the type and efficiency of the resources exploited by Xunlei. We will then examine data redundancy for Xunlei’s multi-source download process. Finally, we compare protocol usage for bulk file-sharing and IPTV streaming in the Xunlei system, since especially IPTV attracts increasing attention from both

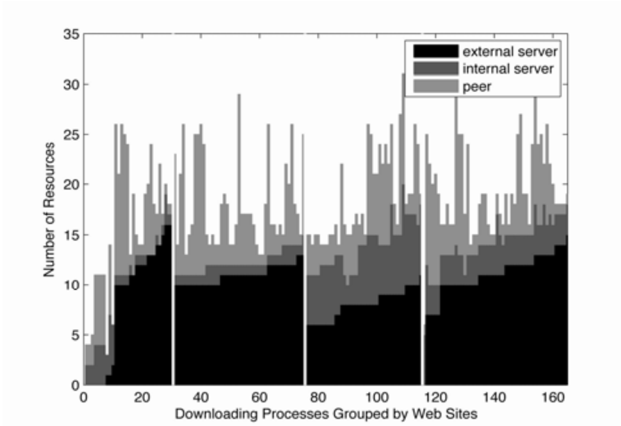

Figure 2. # of resources (host) per resouce type for download processes clustered by web sites. Inside clusters, download processes are sorted by number of external servers.

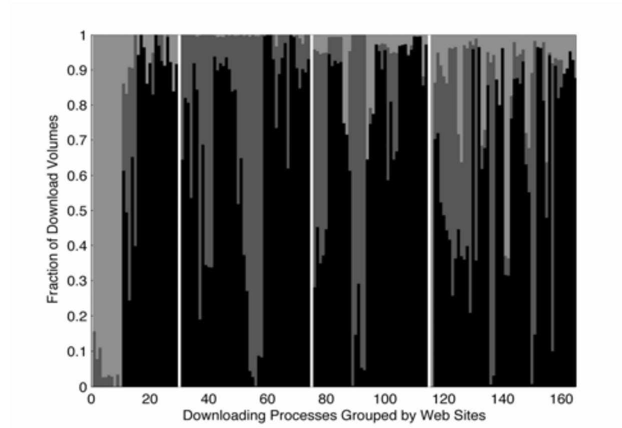

Figure 3. Data fractions per resouce type for download processes clustered by web sites. Inside clusters, download processes are sorted by number of external servers.

users and the researcher community.

A. Resource allocation

In section I-A, we introduced three types of possible resources: *Internal servers* belonging to the specific web site from which files are requested; *External servers* not belonging to the specific web site; and *Peers*, i.e. individual clients who currently host the target file.

In order to investigate the types of resources provided by Xunlei resource allocation servers, we chose ten popular files and four portal web sites as our experimental subjects. Each file is download from each web site in five consecutive times. Since some web sites lack some of the target files, the experiment allowed us to record 165 download sessions.

The number of allocated resources per session is illustrated in Fig.2(a). The x-axis lists the 165 download processes grouped according to web sites (divided by vertical white spaces) and sorted by number of external servers inside each cluster. The y-axis is numbers of resources according to their resource type. From the figure it is evident that Xunlei servers provide many additional resources, in which external servers (exploited by direct linking) and peers account for the majority.

Large quantities of available resources however do not necessarily mean effective downloads. The quality of the provided resources has been evaluated by examining the downloaded data fractions per resource type. The results are summarized in Fig.2(b) for the same 165 download processes. It is shown that external servers and peers provide most data in many download processes. Only 13 out of 165 (8%) download processes retrieved more than 50% of the downloaded data volume from internal servers. This shows that the additional resources provided by Xunlei servers indeed improve client download performance.

However, we also want to point out the improperness of downloading (i.e. leeching) data from external servers. For a web site, view counts are important, since more visits would attract advertisers. Usually, only users that consume the advertisement can take advantage of the provided service.
web site, view counts are important, since more visits would

attract advertisers. Usually, only users that consume the advertisement can take advantage of the provided service. The behavior that Xunlei client downloads data from external server breaks this convention. It gets free download services without actually browsing the sites.

Redundancy	Number of downloads	Proportion
R(i)<1.2	210	54.4%
1.2<R(i)<1.5	131	33.94%
1.5<R(i)<2.0	41	10.62%
R(i) > 2.0	4	1.04%

TABLE I. DOWNLOAD DATA REDUNDANCY STATISTICS

B. Download Redundancy

Effectiveness of multi-source downloading is not guaranteed if much duplicate data is retrieved from different resource. In order to evaluate the data redundancy introduced by Xunlei, we define the redundancy of download *i* as R(i):

$$R(i) = \frac{S(i)_{raw}}{S(i)_{s\tan dard}} \qquad (1)$$

Where $S(i)_{raw}$ is the down-data size in i_{th} download and $S(i)_{standard}$ is the real size of target file.

Table I provides a summary of redundancy values of 386 download sessions from several sites, consisting of various different file types and files sizes. We took the sum of valid TCP and UDP data (packet headers removed) as $S(i)_{raw}$. Considering network packet loss and retransmissions, R(i) presents an upper bound of its actual value. The results indicate that Xunlei's multi-source allocation works effectively for downloading from numerous sources, with almost 90% of these downloads experiencing data redundancy of less than factor 1.5.

C. Bulk transfer vs IPTV

In this subsection, we seek to understand sharing properties of bulk files-sharing and IPTV in Xunlei system.

Xunlei provides a site for navigating users to search video files. Users would search, download, even browse videos online. We downloaded (in case of bulk files) and streamed (in case of IPTV) four popular video files in order to analyze port and protocol usage as well as up- and download- data volume. Upload traffic is generated when the client serves as peer for other Xunlei clients.

Table II shows that most data is transported by UDP packets with a byte signature of {0x32, 0x00, 0x00, 0x00} in the four initial payload bytes. The source and destination ports used appear randomly without any predictable pattern. Different from bulk file-sharing, where upload volume reaches about 12% of download volume, there is a considerable amount of upstream data for IPTV, reaching about 70% of the download volume.

Mediafile			File1	File2	Str1	Str2
Filetype			bulk	Bulk	IPTV	IPTV
Filesize (MB)			198.8	187.1	42.1	168.6
Time (sec)			131.4	1515	78	300
Down	Bytes	Total (MB)	204.0	225.2	25.6	96.6
		TCP (%)	0.4	0.1	0.02	0.08
		UDP(%)	99.6	99.9	99.98	99.92
	Pkts	Total (K)	220.0	218.6	30.2	176.1
		TCP (%)	0.5	0.4	0.18	0.21
		UDP(%)	99.5	99.6	99.82	99.79
Up	Bytes	Total (MB)	24.5	27.7	17.6	63.0
		TCP (%)	0.9	0.4	0.03	0.05
		UDP(%)	99.1	99.6	99.97	99.95
	Pkts	Total (K)	161.3	203.8	28.5	199.5
		TCP (%)	0.8	0.5	0.17	0.14
		UDP(%)	99.2	99.5	99.83	99.86

TABLE II. DOWN-AND UPLOAD VOLUMES VS PROTOCOLS

VI. SUMMARY AND CONCLUSION

In this paper, we have studied Xunlei, the most popular download application in China. This popularity triggered our interests to carry out this first study on Xunlei.

Data transmission in Xunlei is mainly encrypted, which forced us to study packet level features in order to infer Xunlei's application behavior. We focus on describing Xunlei's servers and client-server communication. After analyzing download processes, we observe that Xunlei servers provide abundant resources (both servers and peers) to clients for accelerated download. Most of the download data is actually provided by external servers via direct linking or by peers. The multi-source download approach was shown to be rather efficient and introduces little data redundancy. We also present some preliminary results of bulk file-sharing and IPTV in Xunlei. We reveal that Xunlei uses UDP even for bulk data transfer, which is in contrast to comparable P2P systems such as BitTorrent and eDonkey, which transfer data via TCP and use UDP for signaling purposes only [13]. Further exploring bulk file-sharing and IPTV streaming is subject of future work, in which we want to evaluate the compatibility with other P2P applications.

Based on the results of this paper, we conclude that Xunlei indeed provides users with many resources, though some of them are provided inappropriately from external servers via direct linking. This strategy however contributes significantly to Xunlei's superior performance. Xunlei inspires us to contemplate a business driven network architecture (BDNA) by diverting the inappropriate direct linking into the right and legal path. Acting as an intermediary platform, BDNA could bring the relationship between resource service providers and users into commercial operations. Resource service providers could so publish and capitalize their services, while users could bid for different services. Taking the role of an intermediator, BDNA could guide users to requested resource service providers in the same fashion as Xunlei allocates resources.

ACKNOWLEDGMENT

Thanks go to Nevil Brownlee, and kc claffy (CAIDA) for feedback and comments. This work was supported in part by the Chinese NSFC under Grant 60672069 and 60772043, China 973 2007CB307101 Chinese Ministry of Education under grant 20050004033 and Beijing Jiaotong University under grant 2005SM006.

REFERENCES

[1] H. Schulze and K. Mochalski, "Internet study 2008/2009,", *IPOQUE*.

[2] W. John, S. Tafvelin, and T. Olovsson, "Trends and differences in connection-behavior within classes of internet backbone traffic," in *PAM'08: Passive and Active Network Measurement Conference*, 2008, pp. 192–201.

[3] M. Zhang, C. Chen, and N. Brownlee, "A Measurement-Based Study of Xunlei," in *PAM'09: Passive and Active Network Measurement Conference, Student Workshop*, 2009.

[4] M. Zhang and C. Chen, "Behaviors of bandwidth theft in xunlei," in *ICCSIT*, 2009.

[5] S. Saroiu, P. K. Gummadi, and S. D. Gribble, "A measurement study of peer-to-peer file sharing systems," *In. Proc. of Multimedia Computing and Networking. (MMCN)*, 2002.

[6] J. Miller, "Characterization of data on the gnutella peer-to-peer network," *In. Proc. of the 1st IEEE Consumer Communications and Networking Conference*, 2004.

[7] J. Yang, H. Ma, W. Song, J. Cui, and C. Zhou, "Crawling the edonkey netwok," *In. Proc. of the fifth International conference on Grid and Cooperative Computing workshops*, 2006.

[8] M. Zhou, Y. Dai, and X. Li, "A measurement study of the structure overlay network in p2p file-sharing systems," *Advances in Multimedia*, 2007.

[9] Y.-J. Hall, P. Piemonte, and M.-Weyant, "Joost: A measurement study," *Technical Report, School of Computer Science, Carnegie-Mellon University*, May 2007.

[10] J. Liang, R. Kumar, and K.W. Ross, "The kazaa overlay: A measurement study," *Computer Networks Journal*, 2005.

[11] S. A. Baset and H. G. Schulzrinne, "An analysis of the skype peer-to-peer internet telephony protocol," in *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings*, 2006.

[12] O. Heckmann and A. Bock, "The edonkey 2000 protocol," *Technical Report KOM-TR-08-2002, Multimedia Communications Lab, Darmstadt University of Technology*, December 2002.

[13] M. Zhang, M. Dusi, W. John, and C. Chen, "Analysis of udp traffic usage on internet backbone links," in *SAINT'09: 9th Annual International Symposium on Applications and the Internet*, 2009.