Analyzing The DC File Sharing Network

Pavel Gurvich
Academic College of Tel Aviv-Jaffa

Noam Koenigstein Yuval Shavitt
School of Electrical Engineering
Tel Aviv University

***Abstract*—This paper investigates the Direct Connect (DC) file sharing network, which to the best of our knowledge, has never been academically studied before. We developed a participating agent, in order to gather protocol specific information. We quantify network characteristics such as distribution of users in hubs, hubs geography, queries distribution and trends in shared folder size. We also characterize the typical DC user: A heavy downloader with a particularly large shared folder.**

Most importantly, we discovered a query duplications problem that drains much of the hubs CPU and bandwidth resources. In the DC network, query facilitation is the most demanding task for hubs and the main factor in the protocol's scalability challenges. We show that in some hubs, up to a third of the queries traffic is duplicated and therefore wasteful. Resolving this problem will dramatically improve hubs performances by reducing the amount of relayed queries and thus permitting larger hub communities.

I. Introduction

Peer-to-peer (P2P) networks are one of the Internet's most popular applications. The number of users and traffic have been growing exponentially since they were first introduced in the turn of the millennium. Previous studies of active file sharing networks focused on Gnutella, BitTorrent, Kazaa, eDonkey and Kad. See [1] for a list of references and a summary of most of these previous measurement studies. These networks differ in the technologies they employ, their topologies and their user behavior. Although Direct Connect (DC) is not as popular as some of the above networks, it is still among the most popular file sharing networks in the world. In a recent report by Ipoque, a Germany-based company that specializes in developing bandwidth managing solutions, DC was found to be the second most popular file sharing network in Eastern Europe (after BitTorrent), with a 17.87% market share [2]. It was also found that most P2P users come from Eastern Europe, where P2P account for 70% of the entire Internet traffic [2]. It is therefore, somewhat surprising that the DC network is maybe the last large file sharing network that hasn't been academically studied.

In this study we devised a participating agent for measuring and characterizing the DC file sharing network. This paper makes two main contributions: First, it is the first measurement study of the DC network. Our main findings are presented in Section II. Second, while investigating this network we discovered a query duplication problem that stems the protocol's scalability potential. Resolving this problem will reduce much of of the hubs CPU and bandwidth requirements and thus allow the hub communities to grow and accept more users. We discuss this in Section III.

A. Measurements in The DC Network

The DC network consists of regular users (clients) connected to one or more central hubs. DC is unique in the world of file sharing networks because it is comprised of hundreds of small communities each connected to a single central hub. Users manually choose hubs to connect to according to their area of interest or geographical location. The first client was developed in 1999 by Jonathan Hess, founder of *NeoModus*. A lock-and-key mechanism was designed to prevent third party clients from joining in. However, this mechanism was soon cracked and today there are many third party clients and hub implementations by different software vendors. The protocol has evolved in many ways since its original design, yet no single entity controls the protocol. There is no official specification of the protocol, thus developers rely on reverse engineering and some unofficial documentation on the web. Our measurement agent was developed in C# using the FlowLib[1] open source library.

Once a day, the agent downloads a DC hub list from a public hub-list server[2]. It then initiate connection to the top 150 largest hubs. Some hubs enforce a minimum limit on a user's shared folder size. The agent does not participate in the actual sharing activity. Instead it falsely reports its shared folder size and fakes its content (free riding). Nevertheless, some hubs reject connections and might redirect the agent to another hub. On average we connected to 122 hubs a day.

Upon connection the agent records the hub's user count and the total amount of shared content. When a user joins or leaves the hub a $MyINFO or $Quit messages are sent by the hub to all users respectively. By monitoring these messages the agent can track the current number of connected users in a hub. The agent also monitors and records search queries received from the clients.

B. Data-Set Statistics

From February to mid April of 2009 we connected to 770 hubs and recorded over 53,740,038 text queries from 1,888,219 different IP addresses. On average, the agent connected to 122 hubs every day and logged 781,513 text queries from 104,960 different IP addresses. After analyzing the data and discovering the protocol's duplications problem, we performed several additional measurements. Therefore, some of the measurements presented here are dated after April 2009.

II. Network Characteristics

A. DC Hubs

DC hubs are central servers that connect communities of users. We resolve the geographical location of peers in hubs using the IP address of their queries. While some of the big hubs have a wide diversity of users nationalities, many hubs are dominated by users from the same country e.g., *ketunkolo.nwg-network.com* (in Fig. 1).

In DC clients such as DC++ users manually choose the hubs to connect to. Users often choose hubs in their local vicinity, which explains the geographical clustering. The hubs themselves are located in 26 countries around the world. The majority of the hubs are located in Romania (273 hubs), Luxembourg (95), Czech Republic (79), France (56), The Netherlands (46) and the US (36).

[1] http://code.google.com/p/flowlib/

[2] We used: ***http://hublist.openhublist.org/hublist.xml.bz2***

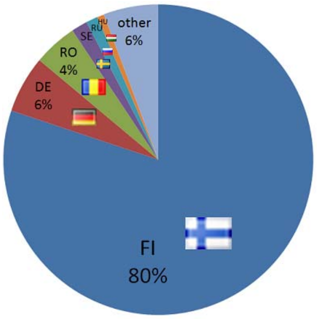


Fig. 1. Queries origin in *ketunkolo.nwg-network.com*

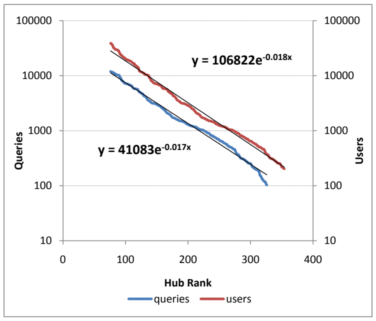


Fig. 2. Exponential distribution of queries and users per hub

The number of users in a hub and its query frequency are highly correlated and reflect its activity. Fig. 2 depicts the distribution of queries and users per hub on a semi-logarithmic graph. When ranked according to the number of queries or number of users, both distributions are exponential with a coefficient of 0.017 and 0.018 respectively. While the top 50 hubs have an average of 3600 users, the rest of the hubs have an average of 680 users connected to them[3]. Hence the DC network consists of a small nucleus of large (mostly international) hubs and a large number of much smaller "community" hubs. As these "communities" become smaller, their number of users and queries frequency decreases exponentially (but with a small exponential coefficient). This result is somewhat surprising, as experience suggests a power-law would often apply for popularity distributions. Nonetheless our measurements indicate differently. One possible explanation is that the small number of large hubs are operated on a high performance commercial servers, while the majority of the hubs are operated by volunteers on regular PCs with domestic Internet bandwidth and thus cannot accept large number of users.

B. Search Queries

Text search queries reflect users taste and areas of interest. Recently, the content of P2P queries had been the focus of several studies in the field of data-mining and marketing [3], [4]. Manually classifying the top 100 most popular text queries, we found that 41% of the queries were related to pornography, 29% related to movies, 16% to music, 3% are searches for a specific file suffix and 1% are TV programs. The majority of the pornography terms relate to video files, thus video files (movies and pornography) consume the majority of the network's traffic. This is in contrast to networks like Gnutella, where it was found that over 68% of the queries relate to music files [3].

We ranked the top 10,000 most popular text queries collected during 4 days in February 2009. Fig. 3(a) shows that the popularity rank follows a power law distribution best fitted by $y = 21765x^{-0.7}$.This result is similar to the distribution of text queries found in other networks [5], [6]. It was suggested to take advantage of such a distribution by caching a small number of queries [5]. This idea will be applicable in the DC network as well. For example, during this period we monitored 2,643,665 text queries containing 905,203 different strings. However, the top 200 search strings (0.022%), were responsible for 10.8% (284,224) of the total queries.

Fig. 3(b) depicts the percentage of daily text queries received in every hour of the day from different geographical locations. The data was averaged over 8 days in March 2009. Fig. 3(b) shows that the evening hours are the time when most users are actively searching new content, while during night time, very little activity is detected. This type of behavior is common to users in different countries. When local time offsets are taken into consideration, the peaks and lows of the graph are highly correlated, which reflects similar users behavior around the globe. In the US this phenomena is weaker due to the spreading of the users over several time zones. Similar behavior was observes in the Gnutella network [6].

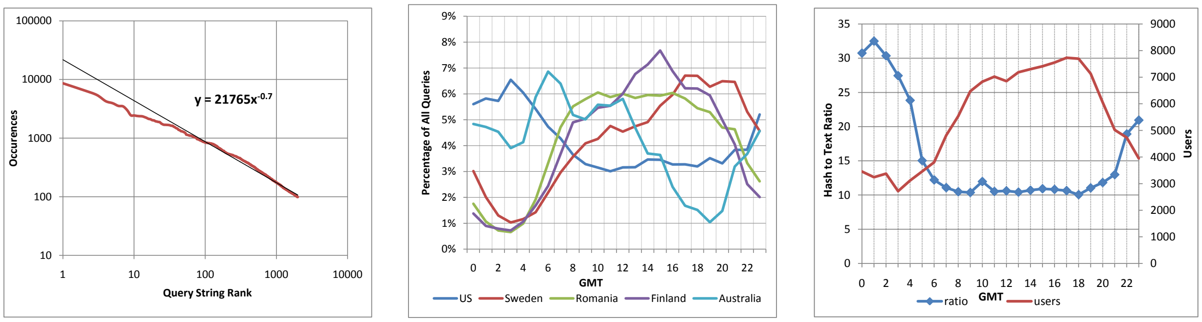
There are two types of queries in the DC network: hash queries and text queries. Hash queries consist of a Tiger Tree hash key that is sent to look for sources of files that are currently being downloaded by the client. These queries are automatically generated by the client SW, after startup or when the number of available sources reduces. Text queries consists of a search string and are manually issued by the users when looking for content to download. The hash to text queries ratio a typical client handles averages between 10 to 30 in favor of the hash queries. This ratio, however is time dependent. Fig. 3(c), for example, depicts a typical Hungarian hub (diablohub.hu) over 24 hours on April the 5th 2009. In this hub, the majority of the hub users are located in Hungary, which is located in the Central European Time Zone (GMT+1). The hash to text ratio (in blue with diamond marks), is inverse to the number of users connected (in solid red). During the night and early morning, the number of connected users drops to a minimum (2700), while the hash to text ratio reaches a maximum (32). While the users count behavior is expected, the time dependance of the hash to text ratio needs an explanation. We therefore suggest that at night time many people leave their computer on to keep downloading while they sleep. Therefore, the ratio between the computer generated queries (hash queries) to human generated queries (text queries) peaks at night time.

C. Average shared size

Hub operators often set strict entry criteria to users, which is usually based on the amount of information a user shares. Although harsh to some users, this requirement reduces leeching effectively. Together with the geographical proximity of users, the DC network is known as the place where "professional" file swappers enjoy high sharing performances.

We monitored the shared folder size of 376,126 users (published in the $MyInfo command) over a period of 4 days

[3] These numbers are an average over 24 hours. During the busy hours, the large hubs have over 10,000 users each.



(a) Zipf law distribution of queries popularity (b) Spatio-temporal behavior of text queries distribution (c) Hash to text ratio and number of users in *diablohub.hu*

Fig. 3. Queries Spatio-temporal properties

Hub	Gigabytes
enigma.evolva.ro	26.7
1.absolutenetwork.se	47.5
dc.rahova-network.ro	36.6
diablohub.hu	30.3
Rock-Massive.THC-Network.org	48.8
klass.idle.ro	35.2
DC.KarlstadRocks.de	76.7

TABLE I
AVERAGE SHARED SIZE

in July 2009. The average shared folder size was 52.7 GB, which is considerably higher than users on other file sharing networks. For example, a recent study in the Gnutella network found that the median value of shared folder is just 650MB [7]. We cross validated our result by dividing hubs' total shared size by the number of connected users (these statistics are published by hubs upon connection). Table I lists the average shared folder size of users in 7 large hubs. The average size ranged from 26.7 GB to 76.7 GB.

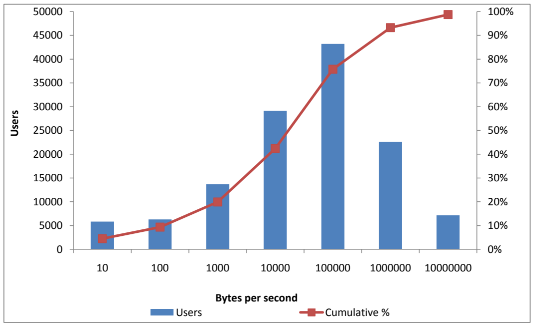


Fig. 4. Change rate in active users shared folder

We were also interested in measuring the change in users shared folder over time. We thus calculated the change in folders size between the first time we saw a user and the last time we saw her. Naturally, many users were idle and did not download any files and some users deleted content from their shared folder. We thus, consider only the 129,998 users who had a positive increase in their folder size and ignore the 198,505 users who had no change and the 47,623 whose shared folder sized decreased. These are the active users, that were downloading at least part of that time period. Some of these users were both downloading and deleting during this time. These numbers are therefore only a lower bound on the amount of sharing performed in the network. Fig. 4 depicts a histogram of the change rate in active users shared folder. On average the active users we monitored increased their shared folder by 74.7 KB per second, which means that at least 9.71 GB were swapped every second, or 3.36 PB over the entire period. Unarguably an impressive statistic.

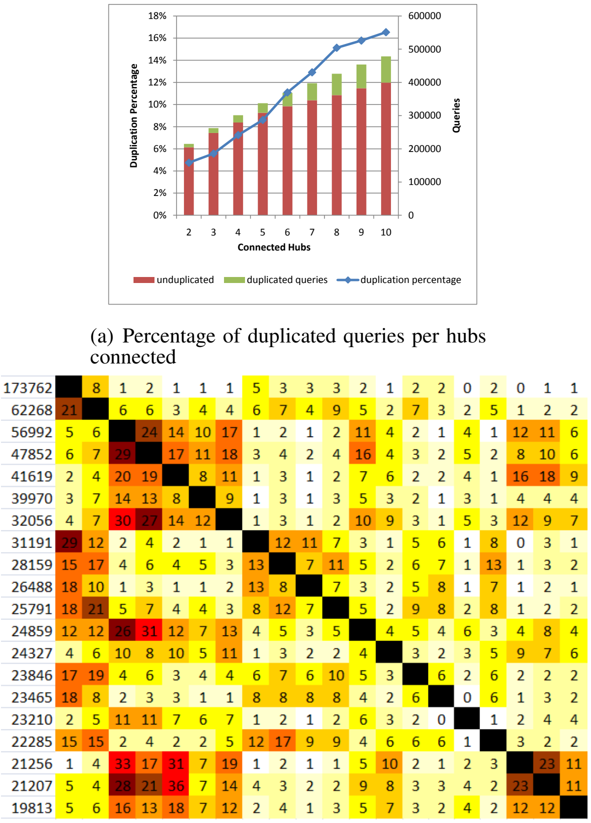
III. THE QUERY DUPLICATION PROBLEM

The hubs most important role is to distribute queries to all connected peers. Clients with relevant content may answer directly to the searching peer, or use the hub as a proxy. Sharing is then performed in a pure P2P fashion. Hubs have no way of knowing which other hubs a user is connected to. By default, clients such as DC++ send search requests to all connected hubs. Inevitably users receive duplicated queries belonging to users with whom they share more than one hub. These redundant queries waste the hubs' bandwidth and CPU resources eventually forcing the hub to reject incoming peers; a known phenomena in the DC network. In this section we show that in some hubs up to a third of the queries are duplicated and therefore wasteful. Thus, the DC duplications problem is a serious challenge that stems the protocol's scalability potential.

We define a *duplicated query* as a query containing the same search string and origin IP address as a previous query coming from a different hub within a short time window (we selected 30 seconds). If an identical query cannot be found, we define the query as *original*. In a series of identical queries the first query is always *original*, while the following queries (if any) are *duplicated*.

On an average day, connected to approximately 122 hubs, we identified that about 40% of the intercepted text queries were duplicated and only 60% were original. This however, is not the common case for regular users, as most users are unlikely to connect to 122 hubs. Fig. 5(a) depicts the percentage of duplicated queries as a function of the number of connected hubs. Since the distribution of users (and queries) per hub is exponential, it is much more likely for a user to connect to two large hubs than to small ones. Therefore, the hubs used to generate the graph are ordered according to their size. As the user connects to new hubs, the percentage of new "unique" queries she receives decreases and the percentage of duplications increases.

Between the two largest hubs the percentage of duplicated queries is 4.7%, however for the smaller hub we see that 19.7%



(a) Percentage of duplicated queries per hubs connected

173762		8	1	2	1	1	1	5	3	3	3	2	1	2	2	0	2	0	1	1
62268	21		6	6	3	4	4	6	7	4	9	5	2	7	3	2	5	1	2	2
56992	5	6		24	14	10	17	1	2	1	2	11	4	2	1	4	1	12	11	6
47852	6	7	29		17	11	18	3	4	2	4	16	4	3	2	5	2	8	10	6
41619	2	4	20	19		8	11	1	3	1	2	7	6	2	2	4	1	16	18	9
39970	3	7	14	13	8		9	1	3	1	3	5	3	2	1	3	1	4	4	4
32056	4	7	30	27	14	12		1	3	1	2	10	9	3	1	5	3	12	9	7
31191	29	12	2	4	2	1	1		12	11	7	3	1	5	6	1	8	0	3	1
28159	15	17	4	6	4	5	3	13		7	11	5	2	6	7	1	13	1	3	2
26488	18	10	1	3	1	1	2	13	8		7	3	2	5	8	1	7	1	2	1
25791	18	21	5	7	4	4	3	8	12	7		5	2	9	8	2	8	1	2	2
24859	12	12	26	31	12	7	13	4	5	3	5		4	5	4	6	3	4	8	4
24327	4	6	10	8	10	5	11	1	3	2	2	4		3	2	3	5	9	7	6
23846	17	19	4	6	3	4	4	6	7	6	10	5	3		6	2	6	2	2	2
23465	18	8	2	3	3	1	1	8	8	8	8	4	2	6		0	6	1	3	2
23210	2	5	11	11	7	6	7	1	2	1	2	6	3	2	0		1	2	4	4
22285	15	15	2	4	2	2	5	12	17	9	9	4	6	6	6	1		3	2	2
21256	1	4	33	17	31	7	19	1	2	1	1	5	10	2	1	2	3		23	11
21207	5	4	28	21	36	7	14	4	3	2	2	9	8	3	3	4	2	23		11
19813	5	6	16	13	18	7	12	2	4	1	3	5	7	3	2	4	2	12	12	

(b) Duplication Matrix: D_{20x20}

Fig. 5. Text Queries Duplication. Data collected on 2/23/2009

of its queries are duplicated (these queries were already sent by the larger hub). Since hubs' size decreases exponentially, this trends escalates as the hubs become smaller. The fourth hub for example, has 66.3% of its queries duplicated in one of the three bigger hubs. Therefore, there is a *diminishing return* effect as a user connects to more large hubs.

We also measured the depth of the duplications for the ten largest hubs. The great majority of the duplicated queries (99.2%) were seen on either 2 hubs (72.6%) or on 3 hubs (26.6%). Only a negligible fraction of the queries (0.8%) were seen on more than 3 hubs.

We define the pairwise duplication matrix between m hubs as $D = [d_{i,j}]_{m \times m}$, where $d_{i,j}$ is the percentage of queries coming from hub i that came also from hub j. The hubs were sorted by size, namely if $i < j$ then hub i is larger than hub j. The diagonal entries $d_{i,i}$ equal zero, as no redundancy exits within the same hub. Fig. 5(b) depicts the duplication matrix of the top 20 biggest hubs in the network. The number of queries collected from each hub is noted on the left. The upper triangle (above the diagonal) represents the duplicated queries between all pairs as percentage of redundant queries seen from the bigger hubs. Similarly the lower triangle represents redundancy percentage seen from the smaller hubs in each pair. The average duplication percentage (without the diagonal) was 6.52%. The average duplication among the lower triangle entries was 8.05% and the duplication percentage in the upper triangle is only 4.78%. A maximum of 36.2% redundancy was seen by one hub located in Sweden (playground.dc-united.se), for queries coming from a bigger Swedish hub (3.absolutenetwork.se). Overall, as much as 68.8% of the hub's queries were duplicated with one of the other 19 hubs. Since users often connect to hubs from their local vicinity, we built smaller duplication matrixes for each country. The average duplication percentage for hubs in the same country was 10.32%, with 13.44% in the lower triangle and 7.45% in the upper triangle.

We further investigate this problem from a different angle: We estimate the number of hubs an average user connects to by tracking user names on user tables (published by hubs). Over a period of 24 hours, we tracked 10,000 users that were connected to one of the hubs we monitored for at least one hour. In this measurement, the average user connected to 2.3 hubs in our sample. However, this number is somewhat misleading for measuring duplication in queries since typically a user starts with an active searching and then continues with a long download period where the computer is left "on" without a human operator. During that period the number of connected hubs slowly drops. We thus measured the maximum number of hubs a user is simultaneously connected to. A number which better reflects the time when the user is active, search for new content and actually generates queries. For the 10,000 users, this maximum was 2.49 on average. These results settles well with the duplication depth histogram we described above, however since the agent is unable to maintain connection to all the hubs in the network at all times, it is possible that the actual numbers for the above measurements (and the numbers in the histogram) are even higher. The results presented here are therefore serve as a lower bound to the duplication problem, yet the severity of the problem is already eminent as is.

IV. Summary

This paper presents a large scale measurement study of the Direct Connect network. We characterize properties such as the distribution of users in hubs, hubs geography, queries popularity distribution and trends in shared folder size. Finally, we uncover a query duplication problem in the protocol, that drains much of the hubs CPU and bandwidth resources.

V. Acknowledgements

This research was supported in part by a grant from the Israel Science Foundation (ISF) center of excellence program (grant number 1685/07).

References

[1] D. Stutzbach and R. Rejaie, "Characterization of p2p systems," in *X. Shen, H. Yu, J. Buford, M. Akon: Handbook of Peer-to-Peer Networking*. Springer, 2009.

[2] Ernesto, "Bittorrent still king of p2p traffic," http://torrentfreak.com/bittorrent-still-king-of-p2p-traffic-090218.

[3] N. Koenigstein, Y. Shavitt, and T. Tankel, "Spotting out emerging artists using geo-aware analysis of p2p query strings," in *International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, 2008.

[4] N. Koenigstein, Y. Shavitt, E. Weinsberg, and U. Weinsberg., "On the applicability of peer-to-peer data in music information retrieval research," in *International Society for Music Information Retrieval Conference (ISMIR 2010)*, August 2010.

[5] K. Sripanidkulchai, "The popularity of gnutella queries and its implications on scalability," Feb. 2001, featured on O'Reilly's www.openp2p.com website.

[6] A. S. Gish, Y. Shavitt, and T. Tankel, "Geographical statistics and characteristics of p2p query strings," in *The 6th International Workshop on P2P Systems*, 2007.

[7] D. Stutzbach, S. Zhao, and R. Rejaie, "Characterizing files in the modern gnutella network," *Multimedia Syst.*, vol. 13, no. 1, 2007.