Delft University of Technology
Parallel and Distributed Systems Report Series

Understanding User Behavior in Spotify

Boxun Zhang, Gunnar Kreitz, Marcus Isaksson, Javier Ubillos
{B.Zhang}@tudelft.nl

Guido Urdaneta, Johan A. Pouwelse, and Dick Epema
January 2013

report number PDS-2013-001

ISSN 1387-2109

Published and produced by:
Parallel and Distributed Systems Section
Faculty of Information Technology and Systems Department of Technical Mathematics and Informatics
Delft University of Technology
Mekelweg 4
2628 CD Delft
The Netherlands

Information about Parallel and Distributed Systems Report Series:
reports@pds.ewi.tudelft.nl

Information about Parallel and Distributed Systems Section:
http://www.pds.ewi.tudelft.nl/

Abstract

Spotify is a peer-assisted music streaming service that offers instant access to a vast music catalogue, and it has gained worldwide popularity in the past few years. Until now, little has been published about user behavior in such services. In this paper, we study the user behavior in Spotify by analyzing a massive dataset collected between 2010 and 2011. Firstly, we investigate the system dynamics including session arrival patterns, daily variations of session length, and playback arrival patterns. Secondly, we analyze the behavioral patterns of individual users on multiple devices and single devices, respectively. Our analysis reveals the patterns of users switching between multiple devices on successive sessions and the favorite times of day for Spotify users. We also show the correlations between both the length and the downtime of successive user sessions on single devices. Our findings not only deepen our understanding of user behavior in Spotify, but also shed light on user behavior in other music streaming services. We believe the results may also be used for building other types of services such as P2P social networks. In particular, our analysis of mobile user behavior provides valuable insights for developing systems on mobile platforms.

Contents

List of Figures

List of Tables

1 Introduction

Spotify – a peer-assisted music streaming service – has gained worldwide popularity in the past few years. Spotify provides millions of users instant access to over 20 million tracks. Different from many commercial music streaming services, Spotify transfers data from both its servers and a proprietary Peer-to-Peer (P2P) network. Using P2P technology considerably increases Spotify's scalability and reduces server workload and bandwidth requirements. Our previous studies [1] have introduced the technical architecture of Spotify and analyzed Spotify's P2P network using a one-week dataset. Until now, little is known about the user behavior in Spotify. In this paper[1], we conduct an empirical study of user behavior in Spotify by analyzing a larger dataset.

User behavior can be fundamentally different in music streaming and video streaming services, because unlike watching videos, listening to music usually does not require constant attention. Hence, people can focus on more important tasks such as working and dining with music played in background. In addition, a music streaming app can turn a smartphone into an MP3 player with millions of tracks and can be used at almost anytime, anywhere, which is not practical for video streaming. Despite the increasing popularity of music streaming services nowadays, few studies [2] have examined the user behavior in those services. Such knowledge is crucial for improving the system design and operation. In particular, the explosively increasing adoption of smartphones and tablets urges the understanding of the usage pattern of music streaming services on mobile platforms.

Our dataset was collected by Spotify between 2010 and 2011, and it covers users in Sweden, UK, and Spain. We focus our study on two aspects: aggregated user behavior, which we define as system dynamics, and individual user behavior. For system dynamics, we first investigate the arrival pattern of user sessions in Spotify. Besides session arrivals, we also examine the daily variations of session length and playback arrivals. For individual user, we investigate separately the behavior on multi-devices and single-device behavior, including both desktop and mobile devices. Our analysis of multi-device behavior reveals how Spotify users switch between multiple devices, and we analyze the temporal properties of user sessions. For single-device behavior, we focus on correlations of successive sessions. To the best of our knowledge, we are the first to study the user behavior in music streaming service using a large real-world dataset. Our findings are not only key for improving system design and operation of Spotify, but also provide valuable insights to understand user behavior in general music streaming systems. We believe that such knowledge can also be used for designing systems that might have similar usage pattern, such as P2P social networks.

Our main findings include:

1. We find that not only the session arrivals, but also session length and playback arrivals exhibit strong daily patterns in Spotify.

2. We show that the session arrivals in Spotify can be modeled as a non-homogenous Poisson process, and the session arrivals in 1-hour and 10-minute intervals can be well modeled by homogenous Poisson process.

3. We observe strong "inertia" of Spotify users to continue successive sessions on the same device, and the probability of switching from desktop to mobile devices is between 20-30%.

4. We find that most Spotify users have their favorite times of day, during which period they spend large fractions of their sessions. In contrast, most Spotify users spend a few but very long sessions in times different from their favorite times.

5. We find the length of the first session can be used as indicator for both the length of successive sessions, as well as the successive downtime on the same device.

The rest of the paper is organized as follows: Section 2 provides an overview of Spotify and its P2P network. Section 3 introduces the dataset used for this study and data sanitization. Section 4 and 5 present our analysis

[1] A 5-page version of this paper is accepted for publication in the INFOCOM-mini conference, 2013.

of system dynamics and user behavior in Spotify, respectively. Section 6 compares our findings with related studies. Section 7 concludes this paper.

2 Spotify Overview

In this section, we give an overview of Spotify, both client and protocol. We focus the description narrowly around aspects relevant to this paper, and refer to [1] for more details on the Spotify protocol.

2.1 Clients

Users can search and browse Spotify's catalogue, and freely select tracks to play. They can also organize tracks into playlists, which are synchronized across all their devices. Tracks and playlists can also be directly linked to via URIs. To access the service, users need to use client software developed by Spotify, which is freely available for download. They also require an account with the service, which is either free (financed via advertisement), or paid-for in a subscription. We describe user account types further in Section 3.1.

Client software is available for Windows, OS X, and Linux, as well as several smartphone operating systems. The application can run in the background to play music on both Android and iOS. There are also Spotify clients integrated into set-top boxes and music players, such as Sonos. Data from these are excluded from our measurements. In addition to streaming tracks, the application can also play local mp3 files stored on the computer or phone. With the more expensive paid-for account (Premium), users can also synchronize tracks from the Spotify catalogue for offline playback. However, being streaming software, the main use case is to use the client online.

2.2 Protocol Overview

Spotify uses a proprietary network protocol designed specifically for on-demand music streaming. The protocol is based on a peer-assisted architecture, streaming from Spotify's data centers, and offloaded by a P2P network running on the computer clients. The smartphone clients do not use P2P, and are are client-server applications. Spotify clients use local storage to cache content the client has accessed, to decrease the network load of data transmission. The cache is such that it cannot be used for playback when the client is offline. A previous study [1] showed that over 55% of the data accessed in the Spotify system comes from the playing client's local cache. The study also showed that P2P serves a further 35% of the data volume.

During our measurement period, Spotify operated data centers in Stockholm and London. Clients were directed to the closest data center by GeoDNS, detecting the country of the client making a DNS query. The client receives a pool of servers to connect to via DNS, and randomizes the order in which it attempts to contact them.

2.3 Connection Behavior

While UDP is the most common transport protocol in streaming applications, Spotify instead uses TCP. Firstly, having a reliable transport protocol simplifies protocol design and implementation. Secondly, TCP's congestion control is friendly to itself (and thus other applications using TCP), and the explicit connection signaling helps stateful firewalls. Thirdly, as streamed tracks are shared in the peer-to-peer network, the re-sending of lost packets is useful to the application.

When the client starts, it prompts for user credentials, which can be stored for future starts and are then used automatically to log in. While the client is running, it tries to always keep an open TCP connection to a Spotify server to keep down playback latency. In the connection, the client authenticates the user, and we refer to the establishment of a TCP connection as logging in, and the closing of the connection as logging out. If the client detects that the socket was disconnected, it will try to reconnect. It may either detect a disconnection via the TCP socket, or by the server not responding to a heartbeat message the client sends every 2 minutes.

Smartphone clients distinguish the local connection medium to determine whether they connected via 3G or WiFi. If the client is connected via a 3G network, and a WiFi network becomes available, the client disconnects the socket over 3G and re-connects to a Spotify server using the WiFi network.

3 Dataset

In this section, we first introduce the collection of the datasets. Second, we present our data sanitization method. Third, we define the terms used through the analysis.

3.1 Trace collection

In Spotify, both clients and servers perform continuous measurement of the system. For this study, we were granted access to the Hadoop cluster that is used to store and analyze Spotify's log data. From the cluster, we extracted session information of Premium users in July 2010 and March 2011, covering Sweden, UK, and Spain. The Premium dataset not only enables us to study long-term user behavior in Spotify, but also allow us to make comparative studies of user behavior on desktop and mobile devices possible, as a Premium account was required to use the smartphone client.

3.2 Data Sanitization

One issue we observe from our datasets is that a non-negligible fraction of sessions were logged out due to "idle-timeout". A "idle-timeout" logout happens when the client fails to send its heartbeat messages (Sec. 2.3) for 10 minutes, which causes it to be logged out by the Spotify server. This can happen when users experience poor network connections, such as walking around a building with poor WiFi coverage or roaming in an unstable 3G network. When a client is logged out due to "idle-timeout" (or for other reasons), it will try to reconnect immediately until a new connection to the server is established, which in turn will be logged as a new session.

Sessions generated by this reconnection mechanism do not reflect user behavior as they are not initiated by users themselves. Thus, we need to merge such sessions that should be one single session but are separated by "idle-timeout" logouts. To do so, we first merge close successive sessions on the same device of individual users that are marked as "idle-timeout". After merging all such sessions, we subtract 10 minutes from merged sessions that are logged out due to "idle-timeout", since this is the last point in time where we know the user had a working network connection. We say two sessions are closely successive if the *downtime* – the time between the login time of the successive session and the logout time of the current session – is below a certain threshold. We expect reconnection times to vary slightly, so we experiment with different thresholds, varying from 15 seconds to one minute, on a small sample dataset. Figure 1 summarizes the fraction of sessions left after merging with different thresholds. We find that around 25% and 75% sessions in the original desktop and mobile datasets can be merged, respectively. The percentage of merged sessions in mobile dataset is much higher than in the desktop dataset because mobile users are more likely to experience unstable network conditions, and mobile clients automatically disconnect 3G sessions when WiFi becomes available. We also notice that the result is insensitive to different thresholds. The reason is that user-initiated sessions typically arrive at a much lower rate, so it is unlikely for our algorithm to merge successive sessions initiated by a user rather than by the reconnection mechanism. In this study, we choose 30 second as the merging threshold for our datasets.

3.3 Terminology

In our analysis, *session length* is the time between the login time and the logout time for merged sessions. A *playback* is the action of a user playing a track from any source such as playlist or search results. We will use

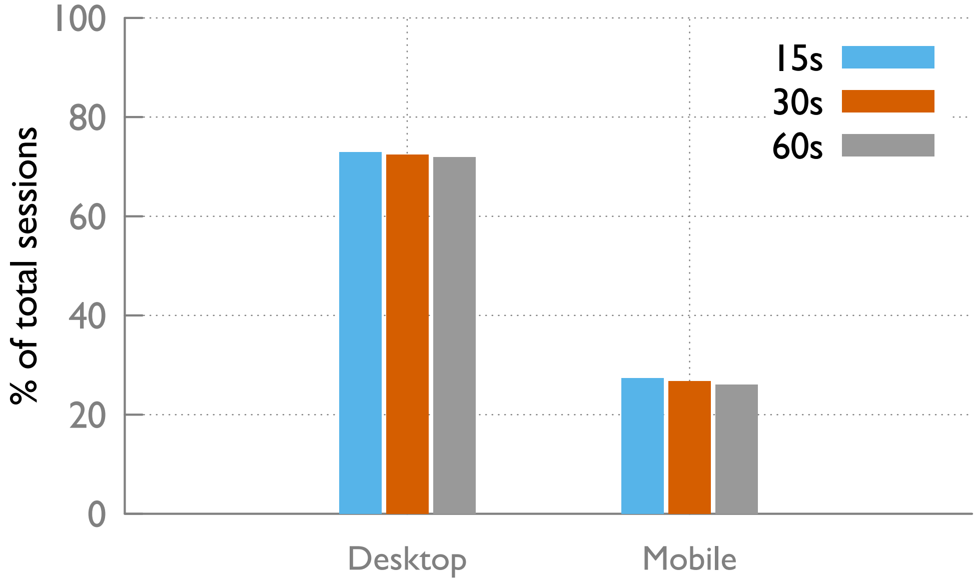


Figure 1: Sessions left after merging with different thresholds.

desktop to refer to regular computers, which includes both desktops and laptops. We use *mobile* to refer to both smart phones and tablets.

4 System Dynamics

In this section, we present our analysis of system dynamics – the aggregated user behavior – in Spotify. We examine the daily patterns of session arrivals, session length, and playback arrivals, respectively. Unless otherwise stated, we present only the results of the Sweden 2011 dataset, as many findings are similar among all of our datasets

4.1 Session arrival

Knowledge of the session arrival pattern is the first step to understand the system usage pattern, and it is also key in designing effective service provisioning mechanisms. Strong daily patterns have been observed in both Internet backbone traffic [3] and many P2P systems [4–10]. Figure 2 shows the number of new sessions in Spotify within 1-hour intervals in a five-day period. The data has been normalized by the total number of daily sessions in the respective category, and the time in this figure is local time. We observe a strong daily pattern and significant variation of hourly arrival rates: the session arrival rate is lowest around 2 am, and increases sharply until 9-10 am, which we define as the *morning peak*. After the morning peak, the arrival rate drops slightly during the lunch break in weekdays. After the lunch break, the session arrival rate increases again and reaches the daily peak around 6-7 pm – the *evening peak*. Then, the arrival rate drops sharply to its lowest point in a day.

An interesting observation here is that the morning peak of mobile sessions in weekdays is often one hour ahead of desktop sessions, which we believe is because the Spotify mobile app is often used while commuting. This hypothesis can also explain the earlier evening peak of mobile sessions. Another observation is the *weekend effect*: during weekends, both the morning peak and lunch break dip of mobile sessions disappear, and the

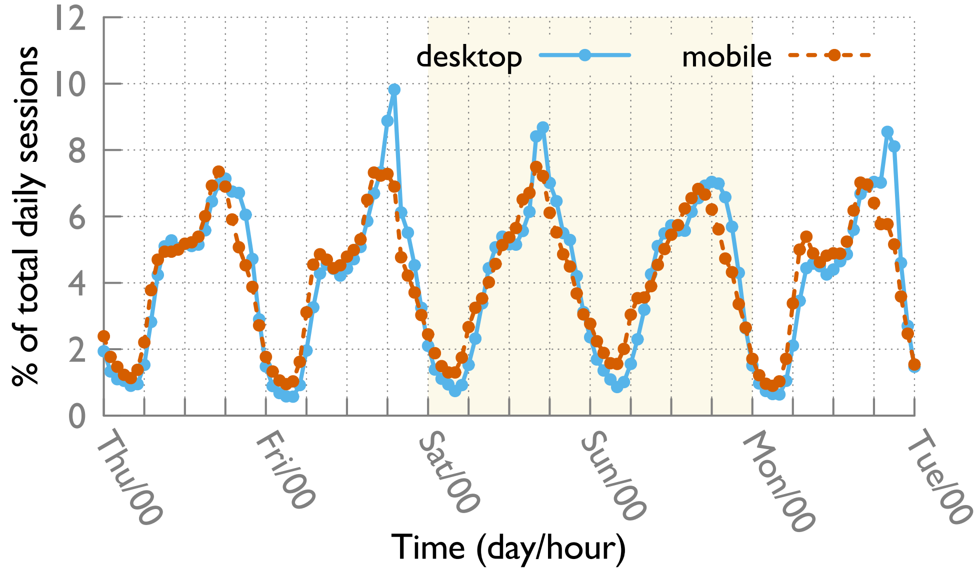


Figure 2: Hourly session arrivals in Spotify.

"commuting effect" also disappears. For desktop sessions, we find that the lunch break dip is noticeably less pronounced than in weekdays.

The high variation of session arrival rates clearly indicates that session arrivals in Spotify cannot be modeled as a homogeneous Poisson process. Thus, we assume the session arrivals in Spotify as a non-homogenous Poisson process. The non-homogenous Poisson process is a Poisson process with its rate parameter λ changing over time. The expected number of events between time a and time b is:

$$\lambda_{a,b} = \int_a^b \lambda(t)\mathrm{d}t \tag{1}$$

To simplify our analysis, we postulate that $\lambda(t)$ is constant in small time intervals so that we can model session arrivals in each interval as a homogenous Poisson process. This approach has also been used to model the traffic of various web systems [11]. Session arrivals in a time interval can be modeled as a homogenous Poisson process if the inter-arrivals are exponentially distributed, and the arrivals are independent from each other.

We use the maximum likelihood estimation (MLE) to fit exponential distributions for session inter-arrivals, and the fitting results are tested against the *Kolmorogov-Smirnov* test (KS-test) with a significance level of 5%, below which the null hypothesis that the fitted exponential distribution represents the empirical data is rejected. We test session arrivals for independence by examining the autocorrelations of the inter-arrivals. If session arrivals are independent, the corresponding inter-arrivals should be a random sequence. For a random sequence, the probability that the autocorrelation at any lag exceeds $1.96/\sqrt{n}$ is below 5%, where n is the number of samples in the sequence [11].

The session arrival patterns can be different across countries in the same year and can also vary over time for the same country. Thus, we perform our test separately on session arrivals for each country (SE, UK, ES) and for each year (2010 and 2011) for a two week period with 1-hour and 10-minute intervals, respectively. In total, the test is performed on 12 datasets, and the results are summarized in Figure 3. The percentage of exponential inter-arrivals and independent session arrivals is above 95% for both the desktop and mobile datasets with 10-minute interval. For 1-hour intervals, the percentage of independent session arrivals of some datasets can drop below 80%. The reason some 1-hour intervals fail the independence test is that during a

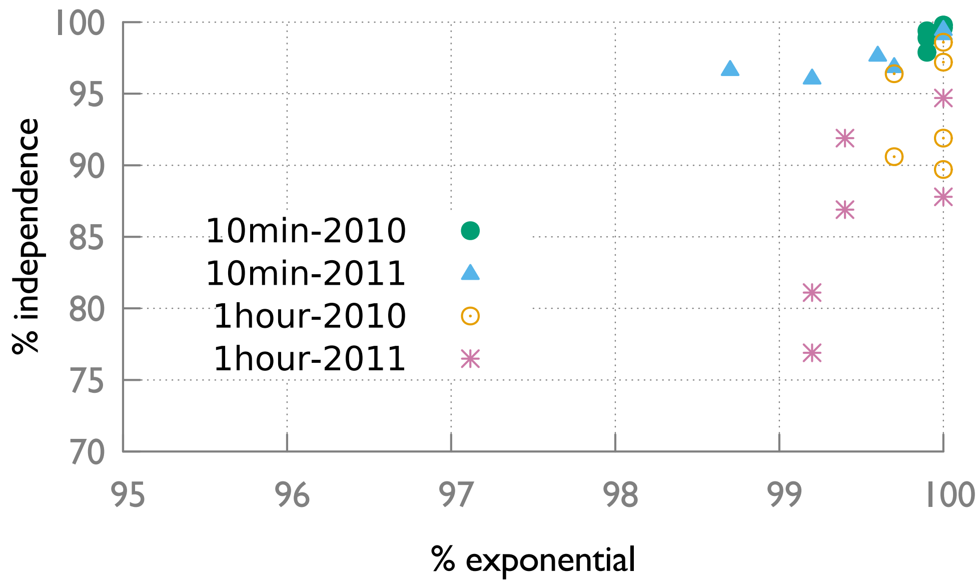


Figure 3: Poisson test results for 1-hour and 10-minute intervals.

certain time of a day, 1 hour is long enough for the daily pattern to take effect (e.g., the morning peak) so the lag-1 autocorrelation of inter-arrivals can be significant.

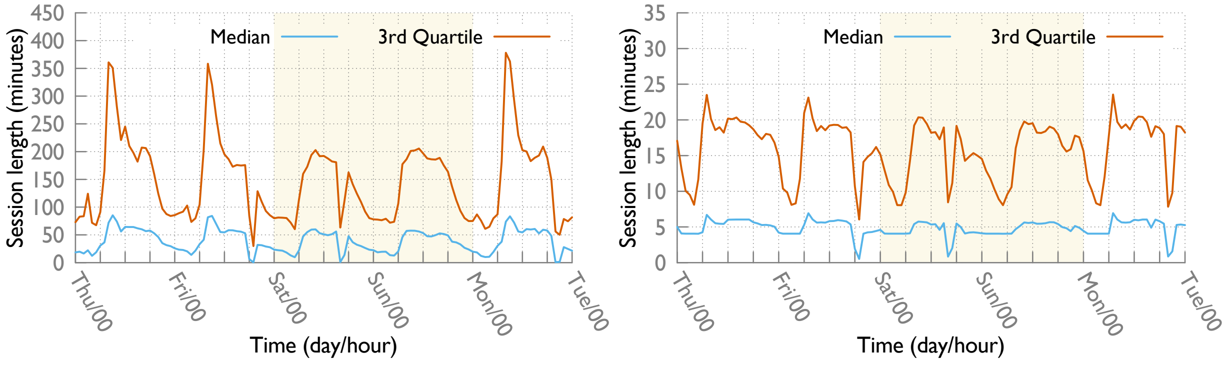
Accurate one-hour forecast of session arrival rates in Spotify can be made by a seasonal ARIMA $(1, 0, 1) \times (1, 0, 1)$ model with a period of 24 (because of the daily patterns of our data). Forecasts are made each hour and each forecast is made based on the data of the last three days. A logarithm transformation is also performed before applying the model to limit the effect of some big spikes (evening peaks) in our data. This model is useful for developing service provisioning in systems with high churn. For space reasons, we omit further details.

4.2 Session length

Besides session arrival patterns, session length is another key property for characterizing important properties such as churn rates in P2P system. A particular question for us is *Does the session length distribution also exhibit daily patterns like session arrivals?* To answer this question, we compute the median and the 3$^{\text{rd}}$ quartile of the length of new sessions starting in each hour, which is shown in Figure 4. An interesting observation is that the length of desktop sessions peaks in the morning and then decreases almost monotonously until late night. The peak of session length matches fairly well with the morning peak of session arrivals, which we believe is because many users launch Spotify to have “background music” at work. Our explanation is confirmed by the weekend effect of session length: the morning peak of session length appears in weekdays but disappears during the weekend. After the morning, the session length of desktop users is very similar in weekdays and weekends.

From Figure 4(b), we find that mobile sessions are much shorter than for desktop sessions, and its morning peak of session length is also much less significant than desktop sessions. We also notice that the median mobile session length exhibits small variation over time. However, we remark that the number of mobile sessions is in fact much larger than the number of desktop sessions, which suggests that the usage pattern is dramatically different between desktop and mobile users.

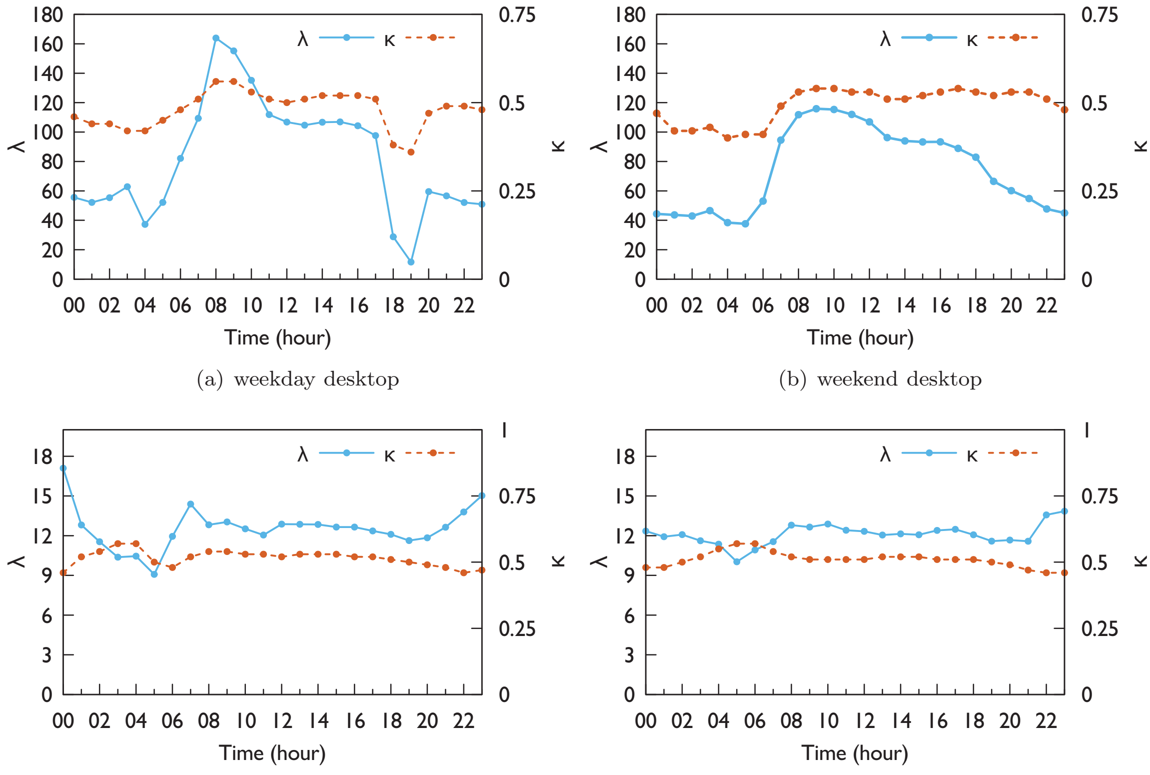
When modeling the session length distribution in P2P systems, the daily pattern is usually not considered [7, 12, 13]. However, as our data indicated significant variation, we argue that capturing the daily pattern is important for building a realistic model. Thus, we perform curve fitting for the length of sessions beginning



(a) desktop sessions

(b) mobile sessions

Figure 4: Daily patterns of session length (Different vertical scales in figures).



(a) weekday desktop

(b) weekend desktop

(c) weekday mobile

(d) weekend mobile

Figure 5: Weibull parameters for desktop session length and mobile session length.

in each hour separately, rather than fitting all sessions in our dataset altogether. We use five widely-used probability distributions, namely Weibull, log-normal, exponential, general Pareto, and gamma distribution. We find that all these distributions except exponential can fit our data for over 95% of intervals. Among the fitting distributions, we find Weibull distribution the best for modeling the desktop session length, as it has the

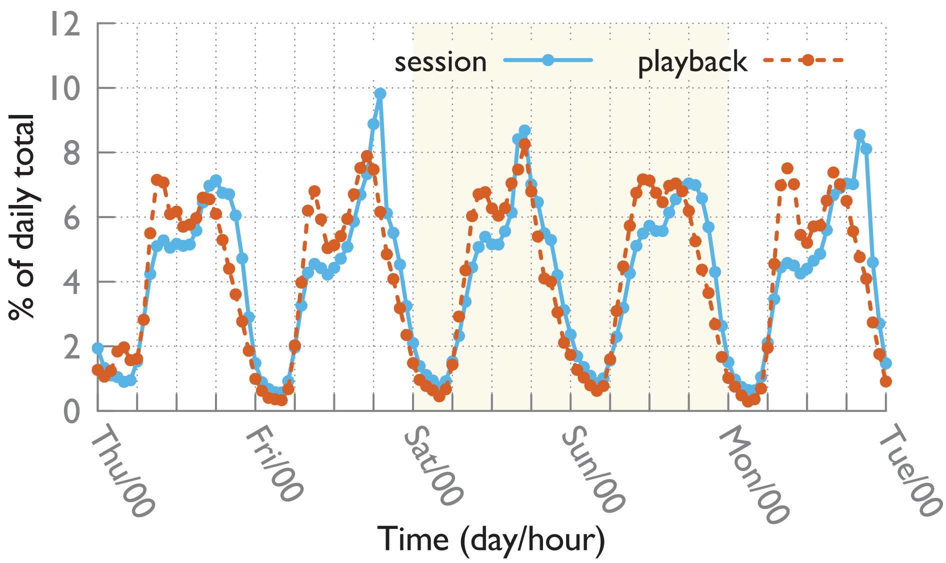


Figure 6: Correlation between session arrivals and playback arrivals.

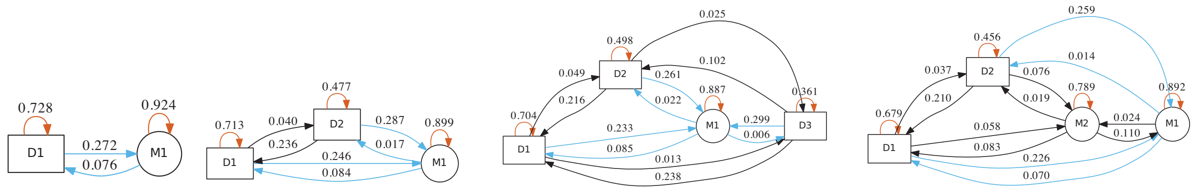
smallest *D Statistic* (the largest discrepancy between the fitting distribution and the empirical data) for around 80% intervals. For mobile sessions, Weibull and log-normal distribution is the best fit for the session length in our 2011 and 2010 datasets, respectively. Figure 5 shows the daily variation of Weibull parameters during a typical weekday and weekend: λ and κ are the scale and shape of Weibull distribution, respectively.

The high variations of session arrival rates and session length consequently make the churn rates in the system vary dramatically across day. Therefore, a more effective churn handling mechanism can be built by considering this daily variation of churn rates. For example, different peer discovery algorithms can be used during the peaks in the morning and evening, respectively: the algorithm used in the morning is for handling (relatively) low churn with long sessions, while the algorithm used in the evening is for handling high churn with short sessions.

4.3 Playback arrival

Besides session length, another important metric to measure the user activeness in music streaming services is the number of playbacks in a session. We define the total number of playbacks in sessions that start in each hour as the hourly playback arrival. As we have observed clear daily patterns of session arrivals and session length, we wondered whether the hourly playback arrivals also exhibit similar patterns. Note that a playback does not necessarily generate traffic to the servers, as a large fraction of tracks are cached locally and/or transferred from P2P network [1].

Figure 6 shows the hourly playback arrivals together with hourly session arrivals. The hourly playback arrivals are normalized the same way as the hourly session arrivals. For desktop users, we observe both the morning peak and evening peak for playback arrivals. However, we find the daily pattern of playback arrivals differs from that of session arrivals. Take the Monday in Figure 6 as an example: the morning peak of session arrivals contribute about 4.5% of total daily sessions but generate nearly 8% of total daily playbacks. This disproportionally high percentage of playbacks indicates that morning sessions are more active in terms of number of playbacks, which can be explained by the long session length in the morning (Figure 4). The evening (or afternoon) peak of playback arrivals is much less significant than the evening peak of session arrivals, which



(a) (b) (c) (d)

Figure 7: Device switch probability for different user groups (red arrows – highest switch probability of each device; blue arrows – probability of switching from and to the most frequently used mobiles.

in turn can be explained by the shorter session length in the evening. In contrast, the playback arrivals of mobile users match fairly well with the hourly session arrivals. This means, unlike desktop sessions, mobile sessions are similarly active in terms of number of playbacks through the day. We believe this is due to the different ways of Spotify users using the desktop and mobile clients. For example, it is rare for mobile users to have very long sessions in the morning, while many desktop users tend to have "background music" during that time. The unmatched arrivals of session and playback for desktop users indicates that other metrics than the session arrival rates are also necessary to accurately capture user activity or system workload.

5 User behavior

In this section, we explore the behavioral patterns of individual Spotify users on multiple devices and single device, respectively. Similarly to Section 4, we present only the results of the Sweden 2011 dataset unless otherwise stated.

5.1 Device switch patterns

Many users have Spotify client installed on multiple devices, but it is not clear how they switch between those devices when using Spotify. Now, we investigate the patterns of Spotify users switching between multiple devices on successive sessions, including both desktops and mobiles. Since users with different numbers of devices may behave differently, we first group users based on the numbers of desktops and mobiles on which they use a Spotify client on. In this section, we focus only on several typical user groups: users with one desktop and one mobile, users with two desktops (e.g., one at work and the other at home) and one mobile, and users with three desktops and one mobile. In addition, we examine the behavior of users with two desktops and two mobiles as a comparison to users with only one mobile.

We study the device switch behavior by measuring the probability of users switching between different devices in successive sessions. First, we rank the desktops and mobiles of a user separately by the usage frequency (number of sessions) of each device. Take a user with p desktops and q mobiles as an example, the desktops and mobiles of that user are ranked respectively as $D_1 ... D_p$ and $M_1 ... M_q$ based on usage frequency, so we can have pairs of ranked devices used in successive sessions (e.g., $< D_1, M_1 >$). Then we can obtain the device switch probability by repeating this process for all users in our dataset.

Figure 7 illustrates the device switch probability of our interested user groups. Firstly, we find that Spotify users have strong "inertia" to continue successive sessions on both the most frequently used desktop and mobile devices, but the inertia on mobile devices is considerably bigger than on desktop devices. Secondly, we find that the probability of continuing the next session on the most used desktop and mobile devices decreases insignificantly as the total number of devices increases. Thirdly, we find that the probability of continuing

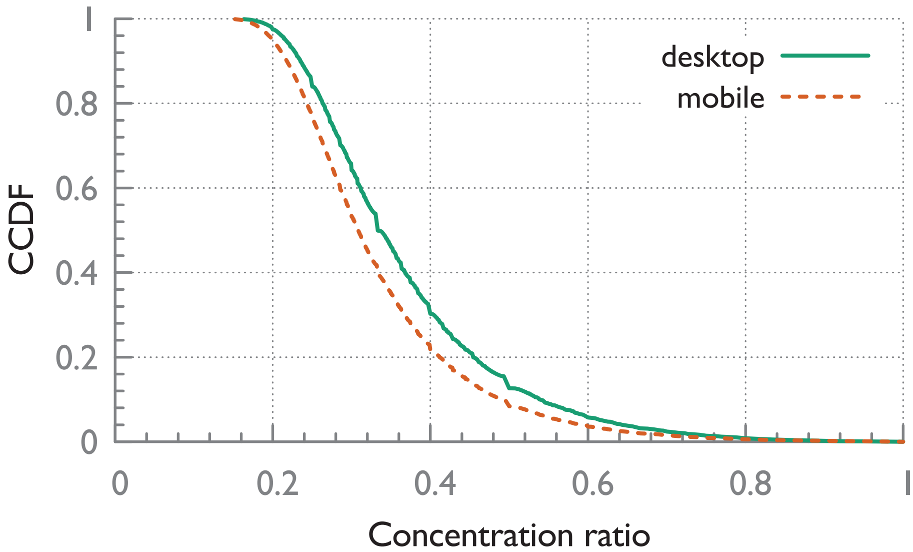


Figure 8: Concentration ratio.

successive sessions on the same device is much lower for less used desktops ($D2, D3$), while it is as high as 0.789 for the least used mobile ($M2$). Last, we notice that the probability of switching from a desktop to the most used mobile is between 0.226 - 0.299. The lesser a desktop is used, the higher the probability of switching to the most used mobile for the successive session.

5.2 Favorite times of day

We now look into the temporal aspect of the multi-device behavior in Spotify. In particular, we are interested in answering the questions *Do users have favorite time of day to use Spotify?* Before diving into details, we first define the terms that are used in the following analysis: *favorite time* is the time period of a day during which the largest fraction of a user's sessions occur. *Concentration ratio* is the ratio between the number of sessions occur in the favorite time and a user's total sessions. *Long session time* is the time period of a day that a user has the longest average session length.

To find the favorite times of Spotify users, we first divide equally the time of a day – 00-24h – into eight parts, and we count the sessions of each user started in each part of a day. Then, the favorite time for each user is the part of the day with the most sessions. Figure 8 shows that concentration ratios is quite high for large fraction of users. More than half of desktop users have over 35% of their sessions in their favorite times, and 30% for half of the mobile users. Another finding is that the concentration ratio of mobile users is lower than that of desktop users. We believe it is because mobiles have better accessibility than desktops so users have more opportunities to spread their mobile sessions across different times of a day.

From Figure 9, we find that the favorite times of large fractions of both desktop and mobile users spread between 12 - 24 pm. The most popular favorite time for mobile users is between 15-18 pm, which is three hours earlier than the most popular favorite time for desktop users (18-21 pm). We believe this is because many Spotify users tend to use desktops rather than mobiles after they arrive home in the evening.

We notice that the favorite time of a large fraction of users is in the second half of a day, which is the time period when many short sessions occur (Figure 4 in Sec. 4). By comparing the average length of sessions in the favorite time and that of all sessions of a user, we find that for many users, the sessions in their favorite times are

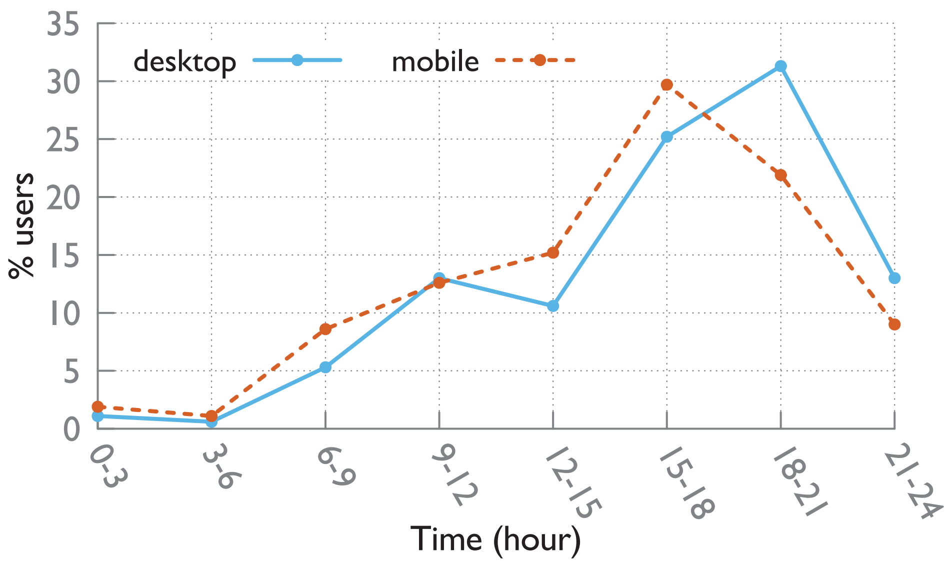


Figure 9: Favorite time distribution.

much shorter than their sessions on average, which is shown in Figure 10. After comparing the average session length in different time periods of a day, we find that for 92% of all users, their favorite times are different than their long session times. The average session length in the long session times of around 50% users are twice as long as the average user session length, and four times as long as the average user session length for about 20% users.

In contrast to the favorite time, we find that the long session time of many desktop users are in the morning, while the long session times of a smaller fraction of desktop users are in late night or early morning, which is shown in Figure 11. The distribution of long session time for mobile users is much more even across day, and the lowest fractions occur between 3-6 am.

5.3 Correlations of successive sessions

We have shown earlier that Spotify users have strong "inertia" on the same device, and now we study the user behavior on single device, with a focus on the correlations of successive user sessions. Our analysis includes the correlation between the length of successive sessions and the correlation between the length of the first session and the successive downtime.

We adopt the approach used in [14] to examine the correlation of the length of successive user sessions on the same device: a user with n sessions have $n-1$ consecutive sessions pairs. We group all session pairs of all users by the length of the first session in each pair. Then, for each group with the first session of length x, we compute the median and mean length of second sessions. Figure 12(a) shows an almost linear correlation between the length of consecutive sessions for desktop users, which has also been observed in Gnutella and Kad [14]. This correlation means that the length of the current session can be used as an indicator for the successive session on the same device. However, the large gap between the mean and median values indicates high variations of the length of successive sessions.

To study the correlation between session length and successive downtime, we change the method for computing the correlation between session length: instead of pairs of the length of the successive sessions, we analyze pairs of the length of first session and the successive downtime. From Figure 13(a), we observe that the median

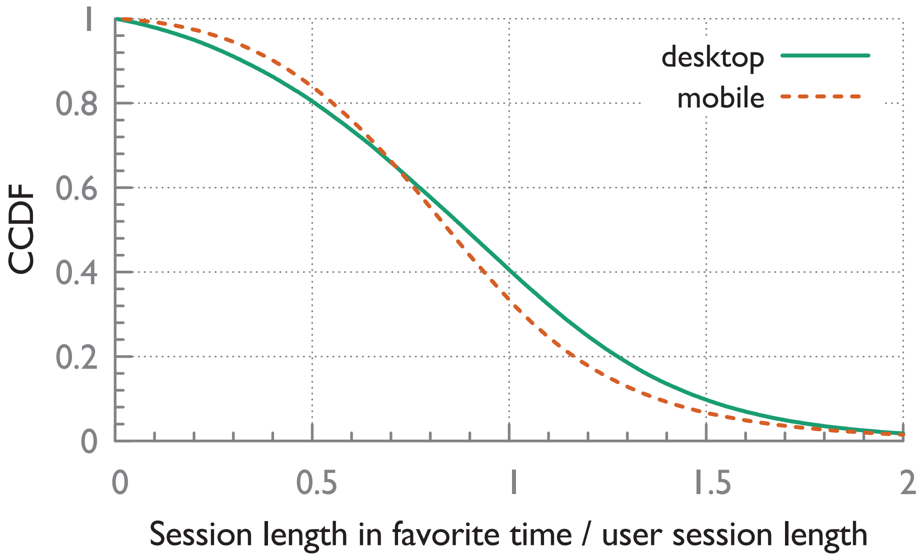


Figure 10: Ratio of the average session length in favorite time and the average user session length.

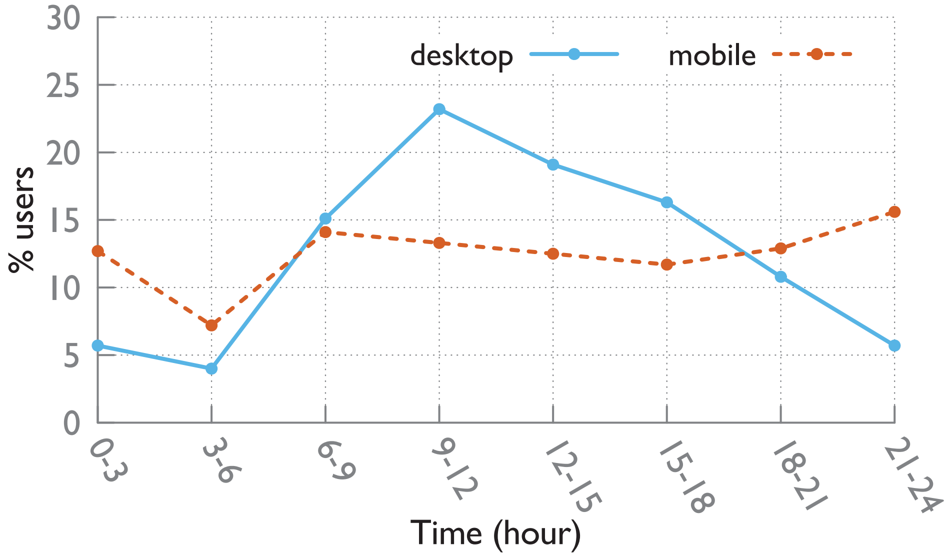
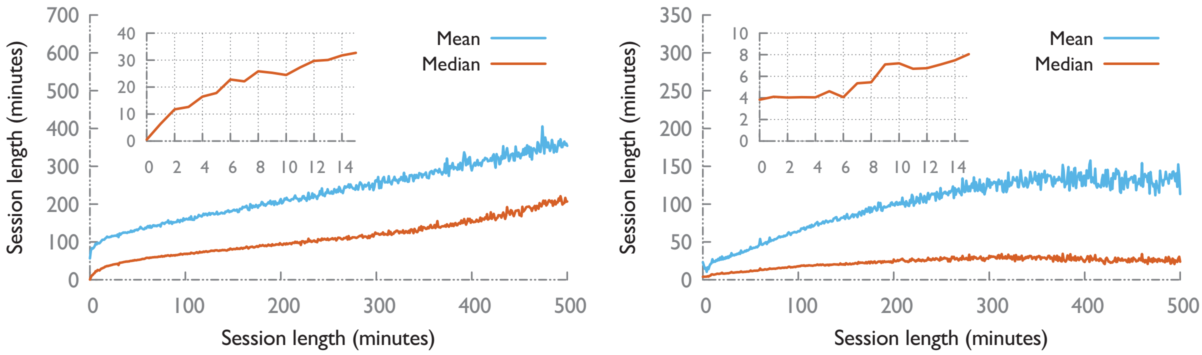


Figure 11: Long session time distribution.

downtime increases as the uptime of session length increases. Here we are particularly interested in the short sessions, because such sessions are the main sources of churn in a system. For sessions shorter than 15 minutes, we observe a linear correlation of session length and the successive downtime, which means many users return to the system shortly after a short session.

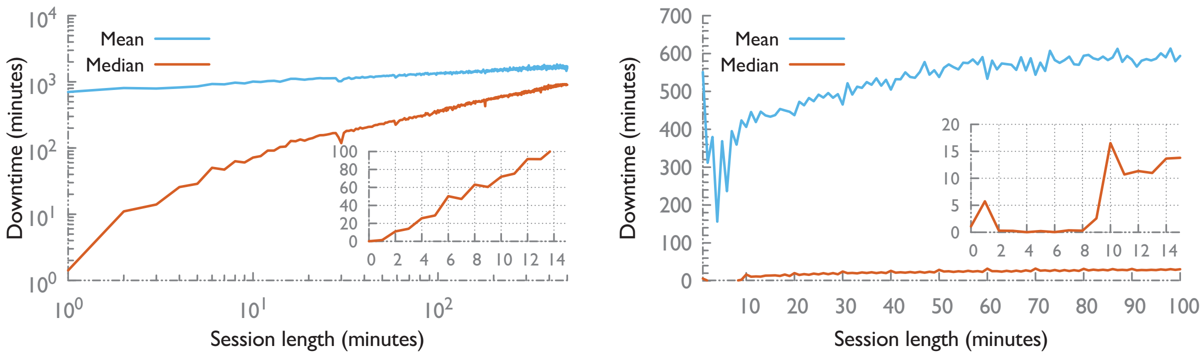
Successive mobile sessions on the same device exhibit similar correlations as desktop sessions. Figure 12(b) shows the correlation between the length of successive mobile sessions. Since mobile sessions are considerably



(a) desktop

(b) mobile

Figure 12: Correlations of length of successive sessions on single device.



(a) desktop

(b) mobile

Figure 13: Correlations of current session length and the successive downtime on single device.

shorter than desktop sessions, both the median and mean length of successive session converge quickly. For sessions shorter than six minutes, the length of 50% successive sessions are shorter than four minutes. Figure 13(b) shows the correlation between the length of current session and the successive downtime. The mean and median downtimes are much shorter than desktop sessions, which is because mobile users have much shorter inter-arrival times than desktop users. For sessions of 2-8 minutes, 50% of the successive downtimes are less than one minute! The peak of mean downtime when the session length is around one minute is probably due to users open the Spotify app accidentally, and then they quickly close the app. These findings confirm the intuition that mobile users can generate much higher churn rates than desktop users.

These findings can be used for improving peer discovery algorithms. For example, Spotify uses centralized trackers [1] to provide peer discovery service in its P2P network. In the current implementation, the entry of a peer is removed immediately after that peer logs out. According to our findings, the tracker can keep the entry of a user after that user logs out a short session (e.g., less than two minutes), since that user is likely to log in again quickly.

6 Related Work

Several previous works study user availability in various contexts. These include both external measurements [6–9] and studies using data from a service provider [2, 4, 5, 10, 15]. There are significant differences between the results of studies, as discussed by Dell'Amico et al. [10], due to both differences in the service studied and measurement methodology.

In a study on Skype [8], only supernodes were measured leading to very high uptimes reported, as supernodes are selected from the most available nodes. In a study of an Italian IM service [10], a weekend effect was observed, whereas no such effect was present in a study of Kad [9]. We remark that the weekend effect in [10] was different from what we observe in Spotify in that they saw usage dropped significantly in the weekend.

Many studies of P2P video streaming systems [2, 4–7, 15] have been conducted in the past years. Session arrivals in many P2P video streaming system [4, 6, 7] exhibit strong daily patterns, and similar patterns have been observed in individual channel of such systems [5, 15]. Compared to Spotify, both the morning peak and evening peak in P2P video streaming systems arrive later: the morning peak of session arrivals in these systems comes around lunch time, and the evening peak arrives toward midnight. We believe these differences of the peak times are caused by the types of content provided in Spotify and those video streaming systems. For example, many users listen to music in the morning but few watch movies.

Several studies [5–7] find that large fractions of user sessions in individual video channels end within ten minutes, which is significantly shorter than the sessions of desktop users in Spotify. It is suggested [6] that the short sessions in video streaming systems is due to impatience of users and the "intro sampling" behavior. In addition, a measurement study of peer lifetime in P2P video streaming system shows that the median of peer lifetime is less than 20 minutes. A study [2] of RealAudio traffic shows that the median user lifetime in sports and talk shows stream channels is around 50 minutes, similar to the median desktop session length in Spotify. A study [4] of a mobile IPTV system find that the average session length in each channel is around only 3 minutes caused by channel surfing [16], and users tend to stay longer in a channel between 0-6 am, which is very different from the peak time of session length (9-10 am) in Spotify.

Several studies [17–19] have pointed out that using multiple devices is increasingly common nowadays and mobile phones are emerging as the primary computing devices for some users. However, all these studies are based on datasets and interview feedbacks collected from small groups of users. To the best of our knowledge, we are the first to measure the device switch probability in a very large user base. Correlations between successive sessions in several P2P file-sharing systems have been studied in [14], and in this study, we take one step further by studying the correlation between session length and downtime of successive sessions.

7 Conclusion

Spotify has gained worldwide popularity in the past few years, but little is known about the behavioral patterns of its users, or that in other music streaming systems. In this paper, we study the user behavior in Spotify by analyzing a large dataset collected between 2010 and 2011. Our measurements was based on data collected directly by Spotify for a very large user set.

We found that in Spotify, not only session arrivals, but also session length and playback arrivals exhibit daily patterns. For individual users, we first studied the behavior of switching between desktop and mobile devices for using Spotify. Second, we found that Spotify users have their favorite times of day to access the service. Third, we observed clear correlations between the session length and downtime of successive user sessions on single devices. Our findings greatly deepen our understanding of user behavior in Spotify, and also provide new insights of user behavior in other music streaming services.

References

[1] G. Kreitz and F. Niemelä, "Spotify – Large Scale, Low Latency, P2P Music-on-Demand Streaming," in *IEEE P2P Computing*, 2010. 4, 5, 11, 16

[2] A. Mena and J. Heidemann, "An Empirical Study of Real Audio Traffic," in *IEEE INFOCOM*, 2000. 4, 17

[3] K. Thompson, G. Miller, and R. Wilder, "Wide-area Internet Traffic Patterns and Characteristics," *IEEE Network*, 1997. 7

[4] Y. Li, Y. Zhang, and R. Yuan, "Measurement and Analysis of a Large Scale Commercial Mobile Internet TV System," in *ACM IMC*, 2011. 7, 17

[5] Y. Huang, T. Z. Fu, D.-M. Chiu, J. C. Lui, and C. Huang, "Challenges, Design and Analysis of a Large-scale P2P-VoD System," in *ACM SIGCOMM*, 2008. 7, 17

[6] H. Yu, D. Zheng, B. Zhao, and W. Zheng, "Understanding User Behavior in Large-scale Video-on-Demand Systems," in *ACM EuroSys*, 2006. 7, 17

[7] X. Hei, C. Liang, J. Liang, Y. Liu, and K. Ross, "A Measurement Study of a Large-scale P2P IPTV System," *IEEE Transactions on Multimedia*, 2007. 7, 9, 17

[8] S. Guha, N. Daswani, and R. Jain, "An Experimental Study of the Skype Peer-to-Peer VoIP System," in *IPTPS*, 2006. 7, 17

[9] R. Bhagwan, S. Savage, and G. M. Voelker, "Understanding Availability," in *IPTPS*, 2003. 7, 17

[10] M. Dell'Amico, P. Michiardi, and Y. Roudier, "Back To The Future: On Predicting User Uptime," *CoRR*, 2010. 7, 17

[11] V. Paxson and S. Floyd, "Wide Area Traffic: the Failure of Poisson Modeling," *IEEE/ACM Transactions on Networking*, 1995. 8

[12] S. Saroiu, P. Gummadi, S. Gribble *et al.*, "A measurement study of peer-to-peer file sharing systems," in *MMCN*, 2002. 9

[13] J. Pouwelse, P. Garbacki, D. Epema, and H. Sips, "The BitTorrent P2P File-sharing System: Measurements and Analysis," *Peer-to-Peer Systems IV*, 2005. 9

[14] D. Stutzbach and R. Rejaie, "Understanding Churn in Peer-to-Peer Networks," in *ACM IMC*, 2006. 14, 17

[15] Z. Liu, C. Wu, B. Li, and S. Zhao, "UUSee: Large-scale Operational On-demand Streaming with Random Network Coding," in *INFOCOM*, 2010. 17

[16] M. Cha, P. Rodriguez, J. Crowcroft, S. Moon, and X. Amatriain, "Watching Television over an IP Network," in *ACM IMC*, 2008. 17

[17] A. Karlson, B. Meyers, A. Jacobs, P. Johns, and S. Kane, "Working Overtime: Patterns of Smartphone and PC Usage in the Day of an Information Worker," *Pervasive Computing*, 2009. 17

[18] D. Dearman and J. Pierce, "It's on My Other Computer!: Computing with Multiple Devices," in *ACM SIGCHI*, 2008. 17

[19] S. Kane, A. Karlson, B. Meyers, P. Johns, A. Jacobs, and G. Smith, "Exploring Cross-Device Web Use on PCs and Mobile Devices," 2009. 17