Inference on BitTorrent Mainline DHT from Network Evolution Time Series

Liang Wang and Jussi Kangasharju

Department of Computer Science, University of Helsinki, Finland

Abstract. Network size is a fundamental statistic for a peer-to-peer system and is generally considered to contain valuable information for analyzing system dynamics. BitTorrent Mainline DHT, as the largest peer-to-peer system today, has been attracting more and more attention in the research community because it provides us an excellent opportunity to study the global user behavior in a content distribution system. However, most existing work only considers the metric by itself and does not explore what features could be extracted from this fundamental metric. What's more, the similarity in the usage pattern of Mainline DHT has not been well studied in a quantitative way and how to cluster them remains as an open question. In this paper, we apply frequency-based feature extraction to such time series data. By using the proposed algorithm, our system successfully discovers and clusters the countries of similar user behavior and captures the anomalies like Sybil attacks and other real-world events with high accuracy. In addition, our result on Mainline DHT provides an interesting and alternative view the on global usage pattern in BitTorrent system.

Keywords: BitTorrent Mainline DHT, User behavior characterization, anomaly detection, Fourier transform, Feature extraction, Classification

1 Introduction

System measurement has its irreplaceable importance in the Peer-to-Peer (P2P) research. The knowledge gained from the measurement data can be used as an invaluable input to help us improve the current system design. The measured data can certainly be simply reported as-is, which makes obtaining deeper insight from the measurements difficult. While such "naive" results hold some interest in themselves, more advanced methods yield more insight and enable automatic comparison of many interesting features. Therefore, instead of merely reporting the crude system statistics, the measurement data is expected to expose the non-trivial relations between the system dynamics. However, such insight from the statistics can only be obtained if the data is properly processed, in other words, when the useful information is extracted.

In a P2P system, the network size is a fundamental system statistic, and network evolution data describes how the size changes as a function of time. P2P technology has been widely used for large-scale content distribution and has contributed significant amount of traffic to the overall traffic on the Internet, also because P2P traffic is highly correlated with its network size. Analyzing the network evolution can help us better understand the traffic pattern in a content distribution network. Therefore, the network size as a system metric is generally considered containing valuable information which can be further used for traffic classification, user behavior characterization, anomaly detection and so on. This observation emphasizes the significance of the work and urges us to seek for proper tools to analyze the data.

In this paper we apply Fourier transform to extract representative features from a time series of the measurements on the size of BitTorrent Mainline DHT (MLDHT) network. Because Fourier transform is not sensitive to issues like temporal alignment of different samples, it also provides us with an easy way of obtaining a "fingerprint" of the data, which can be further used to detect anomalies. Fourier transform is a well-known and widely-used signal processing technique in many fields, including network measurement and anomaly detection [11–14]. We must emphasize that we are by no means the first one who applied such technique in network measurements.

However, despite of its wide application in many fields, Fourier transforms (or other similar methods) have not been commonly used in analyzing MLDHT system even though a lot of research has focused on actively collecting the raw data [3–9]. Notably, MLDHT is the largest P2P system today, and has been attracting more and more attention in the research community because it provides us an excellent opportunity to study the user and system behaviors in a global content distribution network. Though in many existing work, the focus is on different aspects of this system ranging from security and topology to economic incentives, the general methodology is similar in the sense that the common goal is to find a pattern by extracting as much information as possible from time series data. For this kind of work, Fourier transforms and other mechanisms, such as wavelets, can provide interesting insights.

Using the proposed algorithms, we are able to extract meaningful insight from a simple time series of MLDHT network evolution data. Past work that have looked at this kind of data typically have been limited to describing quantitative numbers about the size evolution or the usage in different countries [3, 8]. Our work shows that we can classify different countries according to their usage behavior and easily visualize how different countries group together. Naturally, more sophisticated methods, such as wavelets, could also be used and would likely yield additional insights; our goal in this paper is to highlight the benefits of using proper time series analysis techniques and open the road for further research in a broader range of content distribution networks.

Specifically, our contributions are as follows:

1. We propose using frequency-based feature extraction to analyze BitTorrent MLDHT network evolution. The simple algorithm is also applicable to other time series data in P2P measurements.
2. We show the frequency-based features are robust to the noisy data and can be used to characterize user behaviors and detect system anomalies. Our

clustering result exposes interesting and non-trivial connections between the countries which cannot be simply explained with geographical proximity.

3. We implement our algorithm and evaluate it on a realistic online monitoring system. We report its actual performance along with some interesting findings in the paper.

The rest of the paper is organized as follows. Section 2 gives a brief introduction on Fourier transform and related work. Section 3 presents our method of extracting frequency-based features and Section 4 evaluates it on the realistic system. Finally, Section 5 concludes the paper.

2 Background

In this section, we give a very brief introduction on both Fourier transform and BitTorrent MLDHT. The introduction does not intend to be comprehensive but only covers the necessary background for understanding the system context.

The original idea of Fourier transform is to decompose a function[1] into a summation of a series of sinusoids waves, or in other words, projecting a function from one function base to another. Often, a discretized form consisting only of T consecutive sampling points is used which leads to the discrete Fourier transform. In complex form, it can be written as follows:

$$f[t] = \sum_{k=0}^{T-1} \widehat{f[k]} e^{j2\pi kt/T} \qquad t \in \{0, 1, 2 ... T-1\} \tag{1}$$

$$\widehat{f[k]} = \frac{1}{T} \sum_{t=0}^{T-1} f[t] e^{-j2\pi kt/T} \quad k \in \{0, 1, 2 ... T-1\} \tag{2}$$

where j is the imaginary unit and $\widehat{f[k]}$ denotes the Fourier coefficient of frequency k in the complex form. Fourier transform in the complex plane is much simpler than the trigonometric form and can be efficiently calculated using *Fast Fourier Transform* (FFT). For a more thorough overview of Fourier transform and related techniques, please refer to [1, 2].

Our measuring target in the paper is MLDHT which is a implementation of the distribute hash table to provide decentralized tracker service in BitTorrent system. MLDHT is a simplified version of Kademlia-based protocol, it only supports four control messages (i.e. `PING`, `FIND_NODE`, `GET_PEERS`, and `ANNOUNCE_PEER`) to keep the minimum functionality. The distance between two nodes is calculated as the `XOR` of their IDs. Node ID in MLDHT is 160-bit long and is not persistent, i.e., every time a node joins the system, it will generate a random ID on the fly. For those who are interested in MLDHT protocol, please refer to [3] for more technical details.

[1] In this paper, we use function, signal and time series interchangeably as long as it does not cause ambiguity in the context.

Despite of the simplicity of its protocol, MLDHT scales very well. Actually, MLDHT is the largest P2P system today with 15–27 million concurrent users online and 10 millions regular daily churn [3]. Because of its popularity in the real-world, many modern P2P softwares support the MLDHT protocol by default and it has in fact evolved into a peculiar ecosystem [**?**]. This evolution means that it is not restricted to any particular type of content or application, and therefore obtaining a system level view is paramount to a better understanding of this ecosystem. The MLDHT has been catching more and more research attention these year since it provides us an excellent chance to study user behaviors in a global P2P system.

3 Frequency-Based Feature

To compare user behavior and detect behavior changes, as well as to capture anomalies, we first need to identify a robust feature set. As mentioned, we apply Fourier transform to country-level network size evolution and then generate a frequency-based feature set as its unique "fingerprint". In this section, we formulate the relevant definitions and detail the actual steps of feature extraction from network size evolution. Then we show how the similarities derived can be used to discover patterns and detect anomalies. Fig. 1 depicts the general workflow of the proposed method. Starting from the raw time series data, i.e., estimated number of users in a country or the whole system, we perform the Fourier transform to obtain the frequency domain features. Using singular value decomposition, we reduce the dimensionality of the feature vector to allow for easier comparison of different samples. Below we will detail these steps.

3.1 Feature Extraction with FFT

Our data sets are collected from a monitoring system similar to the one in [3], which continuously estimates the network size of MLDHT by crawling its ID space. Each crawl generates a full list of IP addresses in a subspace. Using the methodology presented in [3], we can get a good estimate of the total network size. System-level network evolution can be obtained by a series of such measurements, then further decomposed to country-level (even city-level) evolution based on their IP blocks. We use the MaxMind service[2] to perform the mapping of IP addresses to geographical locations.

Definition 1 *Network evolution of a specific country c is a time series $f_c(t)$ which gives the network size of c at time t.*

For simplicity of notation, we drop the subscript c if it causes no confusion in the context. Our actual measurements show that $f(t)$ is practically a stationary function with a period of one week for almost all the countries, because user behavior is generally different in the weekends than weekdays, but similar from

[2] www.maxmind.com

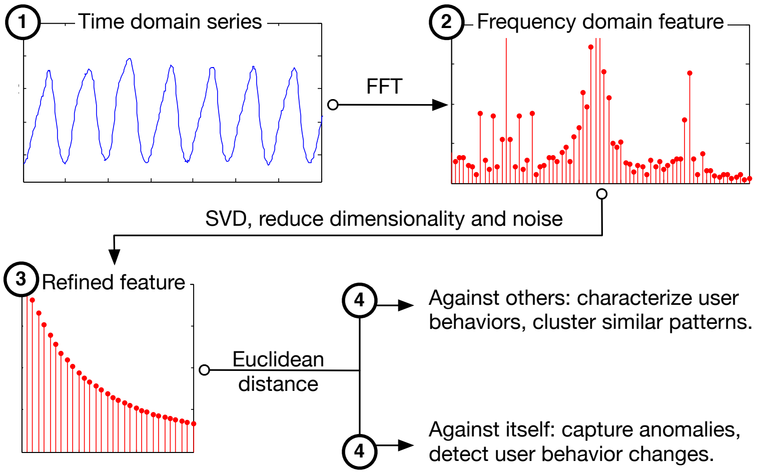

1
Time domain series
FFT
2
Frequency domain feature
SVD, reduce dimensionality and noise
3
Refined feature
Euclidean distance
4
Against others: characterize user behaviors, cluster similar patterns.
4
Against itself: capture anomalies, detect user behavior changes.

Fig. 1: The general workflow from feature extraction to user behavior characterization and anomaly detection.

one week to another. So in our calculation, we adopt a period window of seven consecutive days (irrelevant of the starting point) which consists of T discretized sampling points $f[0]$, $f[1]$, $f[2]$ ... $f[T-1]$. Fourier transform of this time series gives another complex vector of the same length which we will use as the starting point of the frequency-based feature set.

Definition 2 *Frequency-based feature set F of a specific network evolution series $f(t)$ is a real vector where element k represents the normalized Fourier coefficient associated with frequency k in $f(t)$.*

Specifically,

$$F[k] = \frac{\widehat{f[k]}}{\parallel \widehat{f} \parallel} = \frac{1}{T \parallel \widehat{f} \parallel} \left| \sum_{t=0}^{T-1} f[k] e^{-j2\pi kt/T} \right| \tag{3}$$

where the normalizer $\parallel \widehat{f} \parallel$ equals

$$\parallel \widehat{f} \parallel = \frac{1}{T} \sqrt{\sum_{k=0}^{T-1} \left| \sum_{t=0}^{T-1} f[k] e^{-j2\pi kt/T} \right|^2} \tag{4}$$

$\parallel \cdot \parallel$ above denotes Euclidean norm. E.q.(3) is simply the Fourier coefficient of frequency k normalized by $\parallel \widehat{f} \parallel$. E.q.(4) calculates the length of the raw coefficient vector in T-dimensional space which is used as an normalizer in (3). The purpose of normalization is to filter out the effect from the absolute network size but focusing on the pure pattern of a curve. Essentially, we are using a normalized vector of Fourier coefficients as the feature set, where each element represents the amplitude of its associated frequency component in the original signal $f(t)$.

High dimensional features are usually noisy, redundant and expensive to compare. Common technique for dimensionality reduction is *Singular Value Decomposition* (SVD). Fig. 2a shows the semi-log plot of the singular values in a feature set. We can see the first two components dominate the system dynamics. More precisely, the first principal component itself captures over 77% of the energy. In practice, we only keep the first 40 principal components (denoted by u) in order to retain 99.20% of the system energy. The refined feature set can be calculated with u as follows, and $k = 40$ in our case.

$$F^* = [u_0, u_1, u_2 ... u_{k-1}]' F \tag{5}$$

The actual computation can be done efficiently since FFT and SVD are standard routines in scientific computing libraries. Once the feature set is successfully extracted, we can use it to characterize user behavior and further look for similar patterns with properly defined similarity metric as follows.

Definition 3 *Similarity $S_{a,b}$ of two feature sets F_a^* and F_b^* is defined as their Euclidean distance in k-dimensional space.*

$$S_{a,b} = \| F_a^* - F_b^* \| \tag{6}$$

The code in Algorithm 1 shows how the features are actually extracted and pairwise similarity is calculated. The key input M is a matrix where each column vector represents a country's network evolution of a 7-day window. The column vector contains 2048 discretized sampling points. k is the number of principal components we want to keep in the final feature set. The rest of the code is easy to understand. Lines 4 - 8 iterate all the countries, where line 5 calculates the Fourier coefficients using FFT, and line 6 normalizes it to a real unit vector. Note that in our case *abs* function translates each complex element in z into corresponding amplitude. Lines 9 and 10 reduce the feature set dimensionality by taking the first k principal components. Function *pdist* in line 11 calculates the pairwise distance for a given point set. Eventually, each column in the output matrix F represents a k-dimensional feature set of a country, and S contains their pairwise similarity metrics.

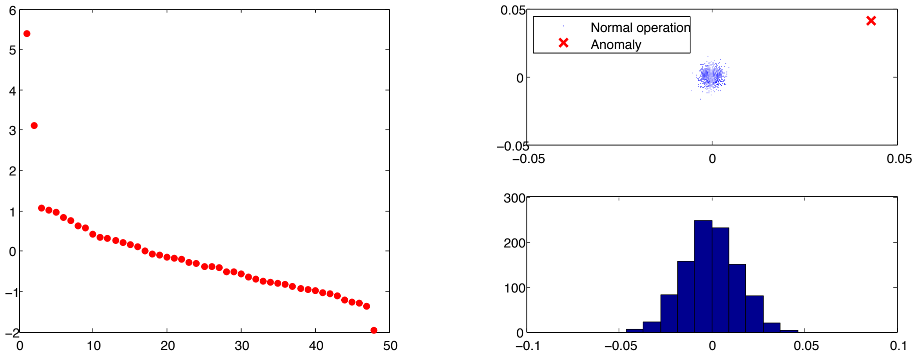
The computation overhead of Algorithm 1 mostly comes from `fft` and `svd` two operations. For an $m \times n$ matrix, the complexity of `svd` is $\mathcal{O}(4m^2n + 8mn^2 + 9n^3)$ if we compute all u, s and v three components in line 9. For a time series of n points, the complexity of `fft` is $\mathcal{O}(n \log n)$ [1]. Another operation that may cause certain overhead is to calculate the pairwise distances using `pdist`. Because the countries that we are interested in are much less than hundreds, it does not introduce too much overhead in practice, especially for offline post-analysis. However, for an online monitoring system, the computation cost needs to be taken into account more seriously. In the actual implementation, we only calculate the principle components once per day due to its rather stable behavior.

Algorithm 1 Feature extraction and pairwise similarity

```
**Input:** Network evolution matrix $M$, $k$ first components
**Output:** Feature matrix $F$, similarity matrix $S$
$[h, w] = \text{size}(M)$
**for** $i = 1 : w$ **do**
    $z = \text{fft}(M(::, i))$
    $z = \text{abs}(z)\ /\ \text{sqrt}(z' * z)$
    $F = [F\ z]$
**end for**
$[\ u, s, v\ ] = \text{svd}(F)$
$F = u(:, 1 : k)' * F$
$S = \text{pdist}(F', \text{'euclidean'})$
```

Note this will not degrade the accuracy of the anomaly detection. Developing more efficient online monitoring algorithms is already out of the scope of this paper and is reserved as our future work.

3.2 Anomaly Detection Based on Fluctuation

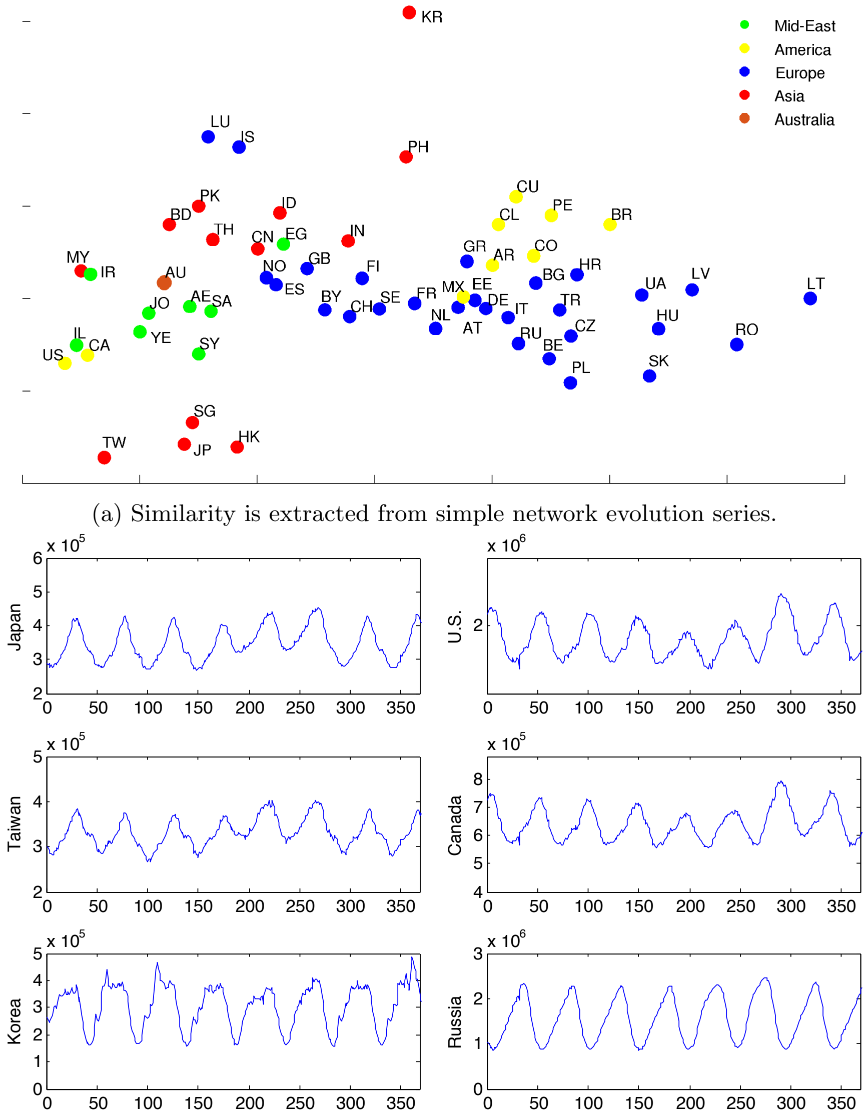


(a) Semi-log plot of the principal components in the feature set. The first two dominate the dynamics.

(b) Feature fluctuation due to the Sybil-attack, and feature distribution during normal operations.

Fig. 2: Feature set is dominated by only few components, and is rather stable during the system normal operations.

In a broad sense, a system anomaly indicates a significant change in a certain system metric which is usually reflected as fluctuation or drift in the corresponding time series. Such anomalies might be due to service breakdowns, system attacks, or drastic changes in use pattern. Detecting such anomalies with satisfying sensitivity and fall-out rate is generally difficult, especially with noisy data.



(a) Similarity is extracted from simple network evolution series.

(b) JP, TW and KR

(c) US, CA and RU

Fig. 3: Clustering of countries based on the similar use patterns, and the original network evolution of some selected countries. In figure (b) and (c), x and y axes are *time* and *network size* respectively.

For an ideal stationary periodic signal, the feature set is time-invariant and remains constant since it has been projected to the frequency domain. Even in a practical setting where some noise is inevitable, the feature set does not vary much if the overall evolution remains constant. In other words, the feature set in each measurement should only fluctuate around its mean and at a very small scale. Fig. 2b plots the variation of system-level features with the mean subtracted from the values. The lower figure shows that the feature distribution during normal operations is very narrow and Gaussian-like. In the upper figure, we can see all the corresponding blue points scattered around the origin within a very small area. The big red cross in the top-right corner marks the deviant feature measured when a Sybil attack was launch in September 2011 [10]. We can see it deviates very far away from the center and this deviation can be used as an indicator of an anomaly. Note that this only indicates a possible anomaly but does not (necessarily) yield information about the nature of the anomaly.

Therefore, we can take advantage of the stability of the frequency-based feature and use it as an anomaly indicator. Technically, it is done by reusing the similarity metric introduced in Section 3.1 against a signal itself, namely $S_{a,\bar{a}}$.

Definition 4 *Anomaly indicator for an entity a is defined as $S_{a,\bar{a}} = \| F_a - \mu_a \|$, where F_a and μ_a denote the current measurement on F_a^* and its average value respectively.*

By definition $S_{a,\bar{a}}$ is the deviation of the current measured F_a^* to its own average. In other words, it measures how similar a signal to itself on average in each measurement. In practice, sample mean is used in place of the true average. An alarm can be triggered if the deviation is large enough, e.g., $S_{a,\bar{a}} > 3\sigma_a$ and σ_a^2 denotes the variance of F_a^*. In the actual implementation, we incorporate supervised machine learning technique to achieve good balance between sensitivity and fall-out rate.

4 Evaluation in the Wild

In the following, we briefly report the performance of the proposed method on a realistic monitoring system with some selected results. We evaluate our method in terms of its effectiveness of clustering similar user behaviors and the accuracy of capturing anomalies. We extended the basic system proposed in [3] by incorporating the feature extraction and anomaly detection modules, and reimplemented it on Spark. The system is not only able to quickly respond to the continuous real-time samples, but also capable of fast processing terabytes historical archives.

4.1 Discover Similarity in Use Patterns

The evolution of network size represents how and when the system is accessed and used, which further reflects the characteristics of the use patterns from a group of geographically-close users. The frequency-based feature is expected to capture the similarity in such patterns and reflect in the proposed similarity metric defined in Section 3.1.

To gain an intuitive understanding of the effectiveness of such mechanisms, we use the first two principal components in the feature set and plot them in

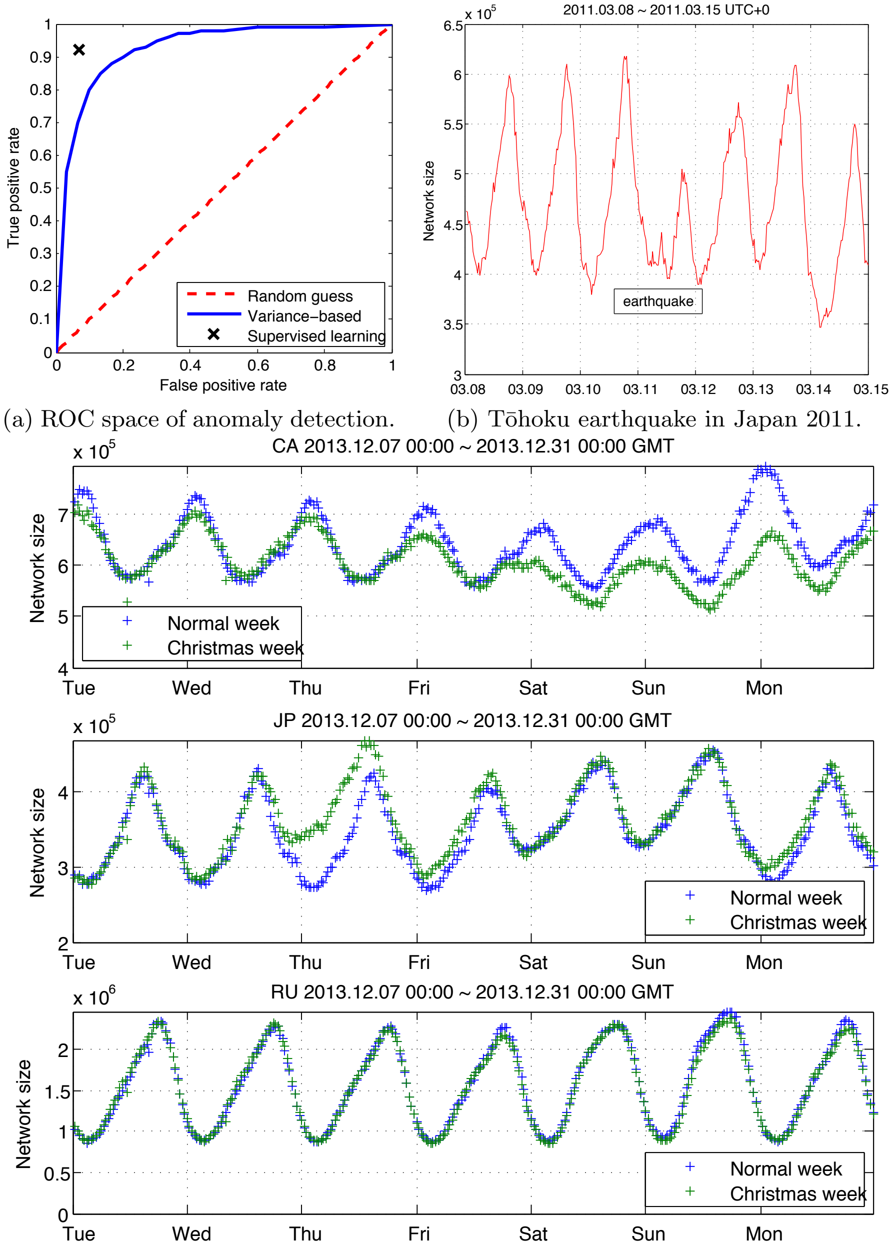
Fig.3a. Countries are marked with different colors according to their geographical regions, and their coordinates in the figure are determined by the two principal components. At the first glance, most of the countries gather together roughly based on their geographical proximity. This is understandable since the geographical closeness usually indicates that the countries share social, cultural, and economic bonds which lead to similar user behavior. For example, among the European countries, most East European countries stay at the right of the figure while the most Western ones gather in the middle.

However, we can see some interesting phenomena among some sets of countries. Comparing Japan, Taiwan, and Korea, we see that Korea is mapped to a very different place in Figure 3a and if we compare the actual daily patterns of these three countries, as shown in Figure 3b, we see that the Korean pattern is clearly different from that of Japan or Taiwan. Similar observation holds when comparing Korea with Singapore or Hong Kong which map close to Japan and Taiwan in Figure 3a. Likewise, Figure 3c compares USA, Canada, and Russia, and the visual differences in the patterns are also reflected in the placement of the three countries in Figure 3a. (USA and Canada are towards the left, Russia is in the middle.)

We have compared other groupings of countries and have seen that countries that map close to each other in Figure 3a have visually similar daily patterns. However, data in Figure 3a only serves to classify countries into different groupings. The distances between different countries do not really serve as an indicator of *how* different their daily patterns are; instead it only indicates the existence of a difference.

4.2 Detecting Nontrivial Anomalies with Learning

Despite different potential causes, the system anomalies are reflected as statistically significant changes in the feature by definition. Anomaly detection takes advantage of the stability of frequency-based features, and issues an alarm when the deviation is large enough. As we have introduced in Section 3.2, the threshold of triggering an alarm can either be manually set or automatically learnt from labelled data. The blue solid line in the Fig. 4a shows the performance by simply varying variance-based threshold in the ROC space. Recall that ROC curve shows the trade-off between sensitivity (true positive rate) and fall-out (false positive rate) in a classification system and increasing the threshold improves fall-out by sacrificing sensitivity. The red dashed line in Fig. 4a represents a pure guess, which serves as a baseline that any rational algorithm should be above and the top-left corner represents perfect accuracy. Our evaluation shows that simple variance-based threshold can already achieve good accuracy. In the actual system, we adopted an naive Bayesian classifier trained on a data set of 500 labelled anomalies. More precisely, we used two labelled data sets from [10] to serve as the ground truth in the evaluation. One data set is used for training the classifier as mentioned above, and the other is used for cross-validation. The accuracy of the classifier is plotted as a black cross in the same Fig. 4a.



(a) ROC space of anomaly detection.

(b) Tōhoku earthquake in Japan 2011.

(c) Behavior drift in 2013 Christmas.

Fig. 4: Accuracy of anomaly detection with different threshold tuning techniques, and some captured real-world events.

As we can see, frequency-based feature with basic supervised learning leads to a high-accuracy detection module.

Besides the large-scale Sybil attack mentioned in Section 3.2 and many others, our anomaly detection system also successfully captured a lot of interesting real-world events. For example, there was a severe earthquake happened in Tōhoku, Japan on March 11th, 2011. Figure 4b shows the network evolution of the corresponding week. Our feature extraction method would have indicated an alarm due to the large-scale network breakdown which caused tens of thousands of computers disconnected from the network. While Figure 4b illustrates the impacts from natural causes, Figure 4c on the right shows how the social and cultural events can change user behaviors. Figure 4c plots the Christmas week in 2013 with green lines for Canada, Japan, and Russia (from top to bottom). To highlight the pattern drift, the network evolution of a normal week is also plotted in the same figure with blue line. Interestingly, we notice while there was significant drop in the network size during Christmas in Canada (actually in most western countries), this number increased quite a lot in Japan (also in many other Asian countries). Russia, on the other hand, did not change much during this time period. One possible explanation could be that in Russia Christmas is celebrated according to the Orthodox calendar, meaning that it falls a few weeks later than in Western countries; in other words, the week shown in Figure 4c is simply a normal week compared with another normal week.

4.3 Summary

As the results above show, the features extracted via Fourier transform allow us to look at the data from another point of view and observe different phenomena more easily. For example, characterizing the diurnal patterns in different countries is readily visible in Figure 3 without the need to compare every country pairs visually.

5 Conclusion

In this paper, we used Fourier transform as a simple yet powerful tool to extract representative information from time series data in BitTorrent MLDHT measurements. Applying frequency-based feature extraction on MLDHT network evolution, we showed that this fundamental system metric contains rich information. The extracted frequency-based feature can be effectively used to characterize user behaviors and detect system anomalies. We implemented a monitoring system with the proposed algorithm and evaluated its actual performance in the realistic environment with many interesting real-world findings. Our algorithm successfully captured anomalies with high accuracy and discovered the clusters of usage patterns. The clustering result further exposed interesting and non-trivial connections between the countries which cannot be simply explained with geographical proximity. In the future, we plan to apply this technique on a broader range of data sets and also explore its other potential applications in P2P monitoring and measurement research.

References

1. R. N. Bracewell, "Fourier transform and its applications," 1980.
2. P. Bloomfield, *Fourier analysis of time series: an introduction.* John Wiley & Sons, 2004.
3. L. Wang and J. Kangasharju, "Measuring large-scale distributed systems: case of bittorrent mainline dht," in *Peer-to-Peer Computing (P2P), 2013 IEEE Thirteenth International Conference on*, Sept 2013, pp. 1–10.
4. C. Decker, R. Eidenbenz, and R. Wattenhofer, "Exploring and improving bittorrent topologies," in *Peer-to-Peer Computing (P2P), 2013 IEEE Thirteenth International Conference on*, Sept 2013, pp. 1–10.
5. R. Farahbakhsh, A. Cuevas, R. Cuevas, R. Rejaie, M. Kryczka, R. Gonzalez, and N. Crespi, "Investigating the reaction of bittorrent content publishers to antipiracy actions," in *Peer-to-Peer Computing (P2P), 2013 IEEE Thirteenth International Conference on*, Sept 2013, pp. 1–10.
6. Z. Jelveh and K. Ross, "Profiting from filesharing: A measurement study of economic incentives in cyberlockers," in *Peer-to-Peer Computing (P2P), 2012 IEEE 12th International Conference on*, Sept 2012, pp. 57–62.
7. J. Timpanaro, T. Cholez, I. Chrisment, and O. Festor, "Bittorrent's mainline dht security assessment," in *New Technologies, Mobility and Security (NTMS), 2011 4th IFIP International Conference on*, Feb 2011, pp. 1–5.
8. C. Zhang, P. Dhungel, D. Wu, and K. Ross, "Unraveling the bittorrent ecosystem," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 22, no. 7, pp. 1164–1177, July 2011.
9. G. Mega, A. Montresor, and G. Picco, "On churn and communication delays in social overlays," in *Peer-to-Peer Computing (P2P), 2012 IEEE 12th International Conference on*, Sept 2012, pp. 214–224.
10. L. Wang and J. Kangasharju, "Real-world sybil attacks in BitTorrent Mainline DHT," in *Proceedings of IEEE Globecom*, Dec. 2012.
11. P. Barford, J. Kline, D. Plonka, and A. Ron, "A signal analysis of network traffic anomalies", in ACM SIGCOMM Workshop on Internet measurment, 2002.
12. L. Quan, J.Heidemann, and Y. Pradkin, "When the internet sleeps: correlating diurnal networks with external factors," In ACM IMC '14, 2014.
13. Z. Bischof, J. Otto, and F. Bustamante., "Distributed systems and natural disasters - BitTorrent as a global witness", in ACM CoNEXT Special Workshop on the Internet and Disasters (SWID), 2011.
14. G. Bartlett, J. Heidemann, and C. Papadopoulos, "Using Low-Rate Flow Periodicities for Anomaly Detection", Technical report, 2009

Appendix

	BR	CA	CN	EG	FI	GB	HK	IL	IN	IT	JP	KR	NL	PL	RU	SE	US
BR	0.00	8.92	6.00	5.56	4.23	5.16	6.46	9.10	4.46	1.81	7.34	3.61	3.02	1.09	1.69	3.95	9.31
CA	8.92	0.00	2.95	3.39	4.69	3.77	2.59	0.19	4.48	7.16	1.72	5.78	5.92	8.22	7.33	4.97	0.39
CN	6.00	2.95	0.00	0.44	1.78	0.85	1.13	3.13	1.54	4.28	1.64	2.88	3.06	5.37	4.47	2.09	3.34
EG	5.56	3.39	0.44	0.00	1.35	0.42	1.35	3.56	1.10	3.84	2.00	2.48	2.63	4.94	4.03	1.67	3.78
FI	4.23	4.69	1.78	1.35	0.00	0.94	2.31	4.87	0.31	2.49	3.16	1.65	1.28	3.59	2.68	0.34	5.08
GB	5.16	3.77	0.85	0.42	0.94	0.00	1.53	3.94	0.71	3.43	2.30	2.22	2.21	4.52	3.62	1.25	4.15
HK	6.46	2.59	1.13	1.35	2.31	1.53	0.00	2.79	2.19	4.66	0.90	3.75	3.44	5.68	4.82	2.53	2.97
IL	9.10	0.19	3.13	3.56	4.87	3.94	2.79	0.00	4.66	7.35	1.91	5.94	6.11	8.41	7.52	5.16	0.22
IN	4.46	4.48	1.54	1.10	0.31	0.71	2.19	4.66	0.00	2.75	3.00	1.62	1.56	3.86	2.95	0.65	4.87
IT	1.81	7.16	4.28	3.84	2.49	3.43	4.66	7.35	2.75	0.00	5.55	2.36	1.24	1.12	0.22	2.19	7.55
JP	7.34	1.72	1.64	2.00	3.16	2.30	0.90	1.91	3.00	5.55	0.00	4.48	4.32	6.58	5.71	3.40	2.08
KR	3.61	5.78	2.88	2.48	1.65	2.22	3.75	5.94	1.62	2.36	4.48	0.00	1.77	3.40	2.58	1.68	6.16
NL	3.02	5.92	3.06	2.63	1.28	2.21	3.44	6.11	1.56	1.24	4.32	1.77	0.00	2.31	1.41	0.96	6.31
PL	1.09	8.22	5.37	4.94	3.59	4.52	5.68	8.41	3.86	1.12	6.58	3.40	2.31	0.00	0.91	3.27	8.60
RU	1.69	7.33	4.47	4.03	2.68	3.62	4.82	7.52	2.95	0.22	5.71	2.58	1.41	0.91	0.00	2.37	7.72
SE	3.95	4.97	2.09	1.67	0.34	1.25	2.53	5.16	0.65	2.19	3.40	1.68	0.96	3.27	2.37	0.00	5.36
US	9.31	0.39	3.34	3.78	5.08	4.15	2.97	0.22	4.87	7.55	2.08	6.16	6.31	8.60	7.72	5.36	0.00

Table 1: The number in the table is a numerical presentation of Fig.3a, which shows the Euclidean distance of any two selected countries. Small distance indicates there is a high degree of similarity between the two countries.