# PBS: Periodic Behavioral Spectrum of P2P Applications

Tom Z.J. Fu[1], Yan Hu[1], Xingang Shi[1], Dah Ming Chiu[1], and John C.S. Lui[2]

[1] IE Dept., CUHK
{zjfu6,yhu4,sxg007,dmchiu}@ie.cuhk.edu.hk
[2] CSE Dept., CUHK
cslui@cse.cuhk.edu.hk

**Abstract.** Due to the significant increase of peer-to-peer (P2P) traffic in the past few years, more attentions are put on designing effective methodologies of monitoring and identifying P2P traffic. In this paper, we propose a novel approach to measure and discover the special characteristics of P2P applications, the periodic behaviors, from the packet traces. We call this the "periodic behavioral spectrum" (PBS) of P2P applications. This new finding, learning the characteristics of P2P traffic from a new angle, could enhance our understanding on P2P applications. To show the effectiveness of our approach, we not only provide justifications as to why P2P applications should have some inherent periodic behaviors, but also conduct hundreds of experiments of applying the approach on several popular P2P applications.

## 1 Introduction

There is a significant increase in P2P applications running over the Internet and enterprise IP networks during the past few years. These applications include P2P content distribution applications like BitTorrent, BitComet and eMule, and P2P streaming applications like Sopcast, PPLive, PPStream. Since P2P applications account for a large portion of total Internet traffic, it is important to correctly identify P2P traffic for traffic monitoring and network operations. However, existing approaches to classifying P2P traffic have well known drawbacks: Port-based method is ineffective since many P2P applications rarely use fixed port numbers. Payload signature-based method is more reliable, but constraints like hardware resource limitation, payload encryption and privacy and legal concerns make it ineffective. Hence, it is important to have a better understanding of the characteristics of P2P traffic and thus being able to effectively differentiate it from other conventional applications such as Web and FTP.

In this paper, we propose a novel approach, *Two Phase Transformation*, to measure and discover "periodic" behaviors of P2P applications. In addition, we provide justifications to why P2P applications should have some inherent periodic behaviors. To show the effectiveness of our approach, we carry out a number of experiments by applying this novel approach on several popular P2P applications (such as PPLive, PPStream, eMule etc.). Interestingly, the experimental

S.B. Moon et al. (Eds.): PAM 2009, LNCS 5448, pp. 155–164, 2009.
© Springer-Verlag Berlin Heidelberg 2009

results show that different frequency characteristics of P2P applications could form a periodic behavioral spectrum (PBS), which could be used as a new form of signatures to help to solve the monitoring and identifying problem. This is the main contribution of this paper.

In the rest of the paper, we first introduce three different periodic communication patterns and provide justifications why P2P applications should have these periodic behaviors (section 2). We propose our approach of how to discover the periodic behavioral patterns of P2P applications from packet traces (section 3). Then we show PBS we developed for a number of popular P2P applications, by individually doing experiments and applying our approach to observe their behaviors in isolation, and its application (section 4). Finally we discuss the related work (section 5) and conclusion (section 6).

## 2 Periodic Group Communication Patterns

Independent of the service type (e.g., file sharing, content streaming, or VoIP), a P2P application needs to form an overlay with other peers for reachability. In order to form and maintain this overlay, and often in the use of this overlay, peers inevitably exhibit periodic group communication.

We distinguish between two classes of periodic group communication patterns: (a) *control plane* - that used to form and maintain the overlay; (b) *data plane* - that used to multicast content. In P2P systems, especially those P2P systems performing application layer multicasting, there are basically two kinds of overlays formed:

- Structured overlays: This includes overlays with mesh-based topology, such as ESM[4], and tree-based topology such as NICE[3] and Yoid[6]. In ESM, group members periodically generate refresh messages and exchange their knowledge of group membership with their neighbors in the mesh. Similarly for tree-based topologies, peers also periodically refresh the overlay links so as to maintain the *soft state* information.
- Data-driven overlays: The classic example is BitTorrent[5]. In this case, the topology is more dynamic, driven by which neighbors have the right content needed by a peer. Such dynamic P2P systems are normally bootstrapped by a server known as the *tracker*. All peers may need to periodically update their information to the tracker for system health monitoring.

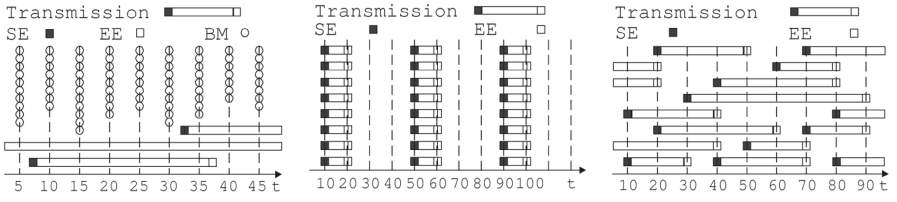
In both kinds of overlays, some active measurements may be used to optimize the efficiency of the overlay. For example, neighboring peers may periodically measure the distance (in terms of round trip time) between each other. In summary, these activities generate periodic group communication patterns.

### 2.1 Terminology for Behavioral Patterns

In this section, we describe three specific periodic group communication patterns that are common for many P2P applications. Note that these three patterns

of periodic group communication behaviors are just examples to illustrate our methodology. The particular values of periodicity of different behaviors are application dependent. A given P2P application may exhibit one or more of such behaviors.

In doing so, we need to define some terminology in our framework. First, time is divided into discrete time intervals. The length of the time interval is quite critical in the ability to identify the periodic behavior, and needs to be carefully chosen. Unless we state otherwise, the length of the time interval is set as 1 second for all our experiments. For the host running the P2P applications (*target host*), it communicates with a number of neighbors. Such communications are organized into a sequence of flows, similar to the flows defined in Netflow, although the inactivity interval that starts and ends a flow is application behavior dependent. The start and end of a flow is indicated with a Start Event (SE) and an End Event (EE), each event has an associated time stamp.




(a) Pattern 1 (b) Pattern 2 (c) Pattern 3

**Fig. 1.** Examples of three periodic communication patterns

**Pattern 1 (Gossip of Buffer Maps):** A popular type of P2P applications is P2P streaming based on a data-driven overlay. This includes live streaming applications such as Sopcast, PPLive, PPStream and P2P Video-on-Demand (VoD) streaming systems. Because of the data-driven approach for forming and maintaining the overlay, all these applications rely on gossip of buffer maps to maintain active links between peers. Typically, one or two packets are enough to present a buffer map.

For P2P Live Streaming systems, because peers only store the streaming content in their RAM (much smaller size compared with VoD) and remove the content as soon as it is played back, the buffer map information changes rather quickly. Each peer needs to periodically exchange its buffer map with neighbors to optimize the scheduling of content exchange to ensure good playback performance. The buffer map exchange period is as short as 5 seconds for some cases. Note, each peer must exchange this information with all its neighbors although it only exchanges content with a subset of these neighbors. For P2P VoD systems, peers also exchange buffer maps with their neighbors periodically, although the period may be longer. Figure 1(a) is an illustration of the traffic pattern for P2P Live Streaming systems. In this figure, we have three flows, e.g., there is a flow which starts right after $t = 5$ and ends between $[35, 40]$. The figure also shows

the periodicity property. For example, every 5 seconds, this node sends out some packets to its neighbors (say at $t = 5$, this node sends out 10 packets describing its buffer maps to its neighbors).

**Pattern 2 (Content flow control):** The second pattern occurs in the data plane. For streamed video content, it often happens that peers download at a higher speed than the playback rate, behaving like file downloading. Although this is good for these peers, the content provider actually prefers the peers download at the pace of playback, to ensure all the peers stay around to help the server in distributing content, rather than watch the content off-line. Content providers thus implement various mechanisms to make peers continue to contribute. One way is to make peers periodically send keep-alive messages to a tracker when they are watching the video, even after the whole video has finished downloading (e.g. in the VoD case[7]). Another way is to perform the Pre-Downloading Control (PDC), which is a form of content flow control to make the download rate match with the playback rate. Such flow control often results in alternating bursts of download activities and sleep periods, as illustrated in Figure 1(b).

**Pattern 3 (Synchronized Link Activation and Deactivation):** It is well-known that BitTorrent implements the *tit-for-tat* mechanism to provide incentives for peers to serve each other. The third pattern of periodic group communication behavior is a direct consequence of how BitTorrent-like protocols might implement the tit-for-tat mechanism. As described in [5,10], each peer uses two timers (10 seconds and 30 seconds) to decide whether to choke and optimistically unchoke neighboring peers, respectively. This results in the synchronization of Start Events (SE) and End Events (EE) at the beginning of the time intervals, as illustrated in Figure 1(c).

## 3 Discovering Periodic Behavioral Patterns

In this section, we describe the approach of how to discover the periodic behaviors of P2P applications, especially the periodic patterns discussed in Section 2.

The overview of the approach is as follows. First, we run a particular P2P application and collect the application's packet trace in a controlled environment where all other network applications are disabled from the target host. While doing so, we only collect packet header information.

Second, we feed the packet trace into three independent and parallel analyzing processes. For each analyzing process, there are two transformation phases. The first one is the transformation from packet-trace to discrete-time sequence, or sequence generator (SG). Three different sequence generators are specifically designed to extract those three periodic patterns. The second transformation phase is the same for all three analyzing processes. It transforms the time-domain sequence to frequency-domain sequence. In this phase, we first apply the Autocorrelation Function (ACF) then the Discrete Fourier Transform (DFT[1]).

[1] In our implementation, Fast Fourier Transform (FFT) is applied.

Finally, we analyze the frequency-domain results derived by ACF and DFT to identify periodic characteristics. In the following section, we present the three sequence generators in detail along with some empirical results.

### 3.1 Sequence Generators

**SG1: Time Series for the Gossip Pattern.** Recall that in our basic model, time is divided into intervals, of length $T$. $X_{in}[i]$ and $X_{out}[i]$ denote time series generated by SG1, where $X_{in}[i]$ represents for the number of source hosts sending data to the target host during the $i^{th}$ interval; and $X_{out}[i]$ is correspondingly the number of destinations which are receiving data from the target host. When the target host is engaged in gossiping, $X_{in}[i]$ and $X_{out}[i]$ represent the number of neighbors gossiping with the target host over the time interval $i$.

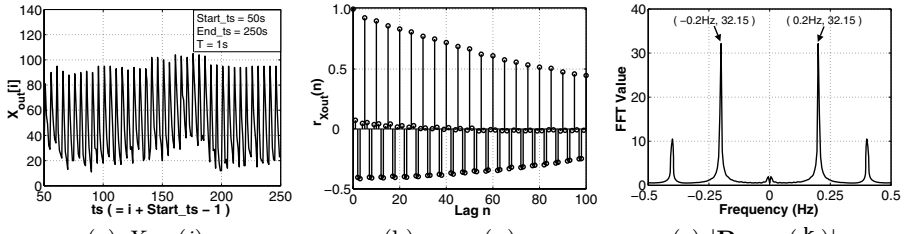
Then the ACF is applied on $X_{in}[i]$ and $X_{out}[i]$ respectively. We denote $r_{Xin}(n)$ and $r_{Xout}(n)$ the result sequences. Finally, we apply DFT on $r_{Xin}(n)$ and $r_{Xout}(n)$ and derive the frequency-domain results denoted by $\mathbf{R_{Xin}}(\frac{\mathbf{k}}{\mathbf{N}})$ and $\mathbf{R_{Xout}}(\frac{\mathbf{k}}{\mathbf{N}})$. Since the ACF and DFT are basic functions in signal processing, the definition and detailed explanation of them can be found in many books and articles, such as[13]. Here we just give the basic formulas. The ACF and DFT are described in Eq. (1) and (2) where $X(i)$ is any input time-domain sequence (e.g., $X_{in}[i]$ or $X_{out}[i]$) and $N$ is the sequence length of $X(i)$.

$$r(n) = \frac{1}{N-n} \sum_{i=1}^{N-n} X(i)X(i+n). \tag{1}$$

$$\mathbf{R}(k) = \sum_{n=0}^{N-1} r(n) e^{-\frac{2\pi i}{N} kn} \quad \text{where } k \in [0, N-1]. \tag{2}$$

The sequence, $\mathbf{R}(0), \mathbf{R}(1), \ldots, \mathbf{R}(N-1)$ is a sequence of $N$ complex numbers (see [13]). For discovering the periodic behavioral patterns, it is sufficient for us to take the modulo of $\mathbf{R}(k)$ to get the magnitude of each frequency component.

For example, Figure 2 shows $X_{out}[i]$ and its ACF and FFT transformations for a PPLive streaming session ($N = 200$). In Figure 2(c), we observe that there



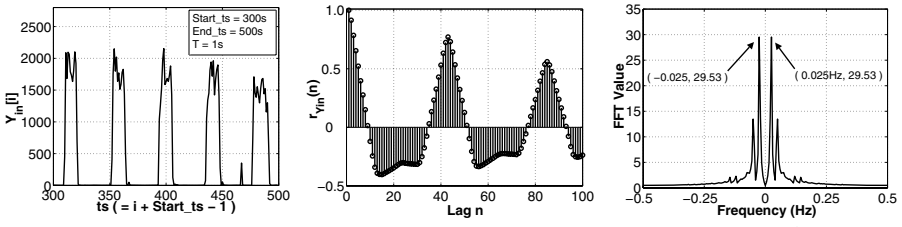

(a) $X_{out}(i)$ (b) $r_{Xout}(n)$ (c) $|\mathbf{R_{Xout}}(\frac{\mathbf{k}}{\mathbf{N}})|$

**Fig. 2.** $X_{out}[i]$, ACF and FFT transformation results for a PPLive streaming session

is a frequency pulse at $f = 0.2$Hz, which means sequence $X_{out}[i]$ has a 5-second periodic characteristic and reveals the periodic gossip pattern.

**SG2: Time Series for Content Flow Control Pattern.** Recall that the content flow control traffic pattern (Figure 1(b)) is about the rate a target host is sending or receiving content from all its neighbors. We represent these as two time-domain sequences, $Y_{in}[i]$ and $Y_{out}[i]$.

The procedure of SG2 is similar to SG1. $Y_{in}[i]$ and $Y_{out}[i]$ are used to accumulate in and out data transmission rate during the $i^{th}$ interval separately, rather than flow count in and out of the target host. After the time-domain sequences are generated, ACF and FFT are applied.




(a) $Y_{in}(i)$ (b) $r_{Yin}(n)$ (c) $|\mathbf{R_{Yin}}(\frac{\mathbf{k}}{\mathbf{N}})|$

**Fig. 3.** $Y_{in}[i]$, ACF and FFT transformation results for a PPStream VoD session

We illustrate an example of PPStream VoD session ($N = 200$). Figure 3 shows $Y_{in}[i]$ and its ACF and FFT transformation results. From the frequency pulse at $f = 0,025$Hz as shown in Figure 3(c), it becomes apparent that PPStream VoD session executes the PDC mechanism in every 40 seconds.

**SG3: Time Series for Synchronized Start and Finish of Flows.** In BitTorrent-like applications, due to the periodic choking and optimistic unchoking mechanism, the occurrences of the data transmission Start Event (SE) and End Event (EE) will also have the periodicity (Refer to Figure 1(c)). This time, the results will be accumulated in time-domain sequences $Z_{in}[i]$ and $Z_{out}[i]$. The algorithm of SG3 is slightly more complicated than the first two algorithms.

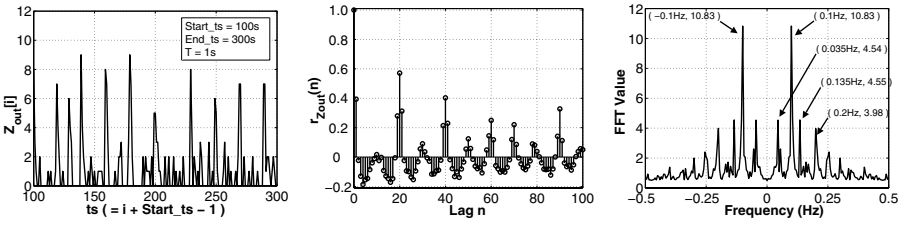
There are three steps of this algorithm. In the first step, all the packets in the input packet trace are reorganized into flows according to their five-tuple information {srcIP, srcPort, dstIP, dstPort and protocol} and then sorted in the ascending order by their *Time Stamp* (TS). Flows destined for the target host are *in-flows*; others are *out-flows*.

In the second step, all the flows are divided into subflows, each subflow with its distinctive SE and EE. Each subflow should correspond to content exchange

| TS: | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Flow: | P1 | | | P2 | | P3 | | | | | P4 | P5 |
| Subflow: | Subflow1 | | | | | | | | | | Subflow2 | |

**Fig. 4.** An example of how to separate a flow into subflows (interval_threshold = 3)

between the target host and one of its neighbors. The rule for marking the beginning and end of subflows is that the time interval of any two consecutive packets of the same flow should not be larger than the given parameter interval_threshold. This is like the *inactivity timer* in Netflow[1]. Figure 4 gives a simple example. In the end, all the triggered events are sorted into ascending order of TS.




(a) $Z_{out}(i)$ (b) $r_{Zout}(n)$ (c) $|\mathbf{R_{Zout}}(\frac{\mathbf{k}}{\mathbf{N}})|$

**Fig. 5.** $Z_{out}[i]$, ACF and FFT transformation results for a BitTorrent session

The third step is similar to the algorithms in SG1 and SG2. The time-domain sequences $Z_{in}[i]$ and $Z_{out}[i]$ represent the total number of SEs and EEs of all the *in-flows* and *out-flows* triggered in the $i^{th}$ time interval respectively.

Figure 5 shows $Z_{out}[i]$, ACF and FFT transformation results for a BitTorrent session ($N = 200$, packet_number_threshold $= 10$, interval_threshold $= 4$). Theoretically, the value of interval_threshold, which determines the start and end of subflows, is likely to affect the frequency characteristics of the sequence. The larger the value is, the fewer number of events will be triggered and there is a higher probability that the application frequency will be buried by noise. So, interval_threshold should take a relatively small value.

In 5(c), we observe that there are four frequencies with large FFT values. In fact, the frequency points $f_1 = 0.035$Hz and $f_2 = 0.1$Hz are the frequency characteristics caused by choking (every 10 second) and optimistic unchoking (every 30 second). The remaining two frequencies, ($f_3 = 0.135$Hz and $f_4 = 0.2$Hz) are the harmonic frequencies, which are the linear combination of the basic frequencies 0.1Hz and 0.035Hz.

## 4 Frequency Characteristics of Popular P2P Applications

In this section, we present the experimental results of the frequency characteristics of several popular P2P applications. These characteristics are derived from the frequency-domain analysis (as discussed in Section 3) of the real packet traces that we captured in a controlled environment.

Packets are captured using Wireshark[2] and each experiment lasts for 30 minutes. When we run each P2P application, we *turn off* all other network applications running on the target machine. In Table 1, we list the frequency characteristics of 15 popular P2P applications and periodic behavioral spectrum (PBS). We also selectively plot the FFTs of these applications in Figure 6. For

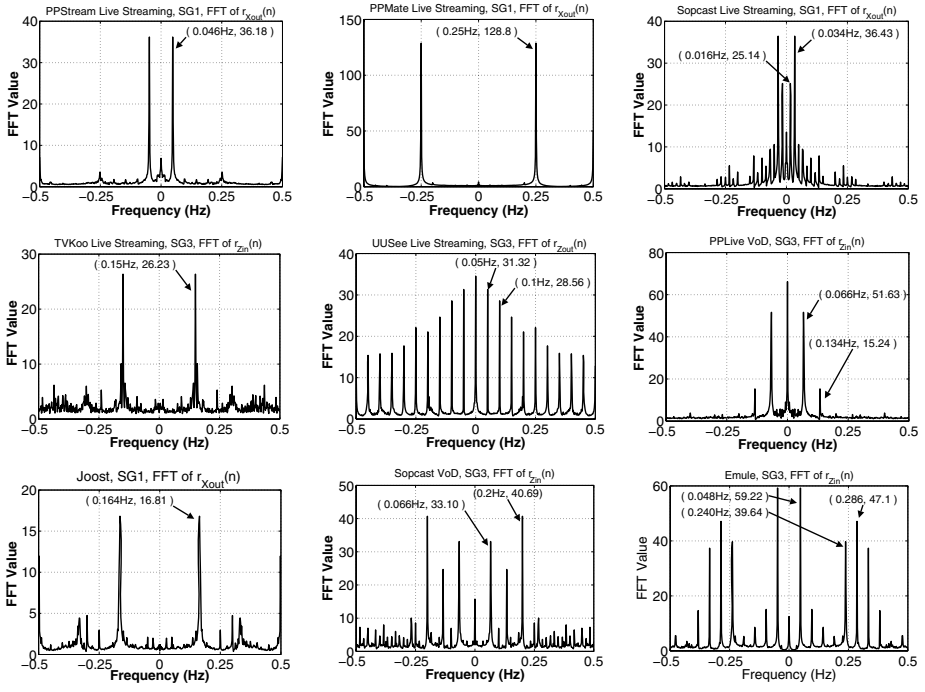



**Fig. 6.** FFT results of nine selective P2P applications

each P2P application, we ran multiple experiments (with different settings) and analyzed the resulting packet trace to double check whether its frequency characteristics showed up every time. For example, for PPLive streaming application, we repeated experiments over six times on different channels of three popularity levels, at least twice per level: the most popular, moderately popular and the least popular. The results confirmed that the traffic exhibited the same frequency characteristics irrespective of a popularity level. In addition, for some of the applications (e.g., PPlive, PPStream), we also carry out the measurement under another controlled environment in which the computer accessed to the Internet was through ADSL instead of the campus Ethernet. Results showed that we could still find the frequency characteristics but the magnitude of FFT values at those frequency points were a bit smaller.

Table 1 shows that most P2P applications have unique fundamental frequencies. Exceptions are PPLive Streaming and TVAnt Streaming, which interestingly share the same frequency characteristics. We believe that the PBS shown in Table 1 can serve as a new form of signatures for classifying P2P applications from mixed packet trace.

Let us present some identification results from the mixed traffic traces collected from our department gateway. The packet header information of each packet was required by the PBS-based approach, but the payload was only used for validation. The measurement duration was two days. After applying the

**Table 1.** The PBS of 15 popular P2P applications

| P2P Application Name | TCP/ UDP) | *In* or *Out* | Fundamental Frequency(Hz) | Harmonic Frequency(Hz) | Effective SGs |
|---|---|---|---|---|---|
| PPStream Streaming | TCP | Both | 0.046 | | SG1, SG2, SG3 |
| PPMate Streaming | | Both | 0.25 | | SG1 |
| | | *Out* | 0.25 | | SG2 |
| PPLive Streaming | Both | Both | 0.2 | 0.4 | SG1, SG2 |
| TVAnt Streaming | Both | Both | 0.2 | 0.4 | SG1, SG2 |
| Sopcast Streaming | UDP | *Out* | 0.016, 0.034 | 0.066 | SG1 |
| | | Both | 0.016, 0.034 | 0.066, 0.1 | SG3 |
| TVKoo Streaming | UDP | Both | 0.15 | | SG1, SG3 |
| UUSee Streaming | UDP | Both | 0.05 | 0.1, 0.15, 0.2, 0.25 | SG3 |
| | TCP | Both | 0.1 | 0.2, 0.3, 0.4 | SG3 |
| TVU Streaming | UDP | *In* | 0.034, 0.066 | 0.1 | SG1, SG3 |
| PPStream VoD | UDP | Both | 0.024 | 0.048, 0.072 | SG1, SG2, SG3 |
| PPLive VoD | UDP | *Out* | 0.1 | 0.2 | SG1 |
| | | Both | 0.066 | 0.134 | SG3 |
| Joost | UDP | Both | 0.164 | 0.328 | SG1, SG2 |
| UUSee VoD | UDP | Both | 0.05 | 0.1, 0.15 | SG1, SG2, SG3 |
| Sopcast VoD | UDP | Both | 0.066 | 0.134, 0.2 | SG3 |
| eMule | UDP | Both | 0.048 | 0.192, 0.24 | SG3 |
| BitTorrent | TCP | Both | 0.034, 0.1 | 0.134 | SG1, SG3 |

PBS-based identification approach, four P2P applications were found and they were PPStream, PPLive, eMule and BitTorrent. We then used a combined method including payload signature checking and manual analysis for validation. The validation results showed that the heuristic approach worked well (with 100% accuracy). Although the PBS-based approach is a prototype in the current stage, we believe that the application of the PBS is promising and valuable.

## 5 Related Works

The tremendous growth of P2P traffic has drawn much attention from researchers. Several studies emphasize on identification of P2P traffic, such as the signature-based payload method in [14] and identifying by transport layer characteristics [8]. Recently, a novel approach named BLINC is proposed by Karagiannis et al. [9]. Although both BLINC and our approach are host-level methods, there is a significant difference between them. BLINC focuses on the behaviors of a host's connection patterns (spatial behaviors) while ours focuses on the periodic behaviors of a given host (temporal behaviors). Moore et al. in [11] apply Bayesian analysis techniques to categorize traffic by application. They also apply FFT to build discriminators [12], but the difference is that their method focuses on each single flow, i.e., applying FFT on the interarrival time of packets belonging to a single flow. Our approach, on the other hand, focuses on the host-level behaviors and we inspect the periodicity of all flows related to the same host.

## 6 Conclusion

In this paper, we first introduce three periodic communication patterns that most P2P applications have and provide concrete justifications. Then we present a novel approach called *Two Phase Transformation* to measure and discover these periodic behaviors of P2P applications. We carry out a large number of experiments applying this approach on several popular P2P applications (such as PPLive, PPStream, eMule etc.), and show the results of different frequency characteristics of P2P applications. These frequency characteristics can form a periodic behavioral spectrum (PBS), which can be used as a new form of signatures to help to monitor and identify P2P traffic.

**Acknowledgments.** We thank the reviewers and our shepherd for providing very helpful technical comments and editing help. This work is partially supported by NSFC-RGC grant N_CUHK414/06 from the Hong Kong government.

## References

1. NetFlow, `http://www.cisco.com/web/go/netflow`
2. Wireshark, `http://www.wireshark.org/`
3. Banerjee, S., Bhattacharjee, B., Kommareddy, C.: Scalable application layer multicast. In: Proc. ACM SIGCOMM 2002 (August 2002)
4. Chu, Y., Rao, S.G., Zhang, H.: A case for end system multicast. In: Proc. ACM Sigmetrics 2000 (2000)
5. Cohen, B.: Incentives build robustness in bittorrent (May 2003), `http://bitconjurer.org/BitTorrent/bittorrentecon.pdf`
6. Francis, P.: Yoid: Extending the multicast internet architecture. White paper (1999)
7. Huang, Y., Fu, T.Z.J., Chiu, D.M., Lui, J.C.S., Huang, C.: Challenges, design and analysis of a large-scale p2p-vod system. In: Proc. ACM SIGCOMM 2008 (2008)
8. Karagiannis, T., Broido, A., Faloutsos, M., Claffy, K.: Transport layer identification of p2p traffic. In: Proc. IMC 2004 (2004)
9. Karagiannis, T., Papagiannaki, K., Faloutsos, M.: Blinc: Multilevel traffic classification in the dark. In: Proc. ACM SIGCOMM 2005 (2005)
10. Legout, A., Liogkas, N., Kohler, E.: Clustering and sharing incentives in bittorrent systems. In: Proc. ACM Sigmetrics 2007 (June 2007)
11. Moore, A.W., Zuev, D.: Internet traffic classification using bayesian analysis techniques. In: Proc. ACM Sigmetrics 2005 (2005)
12. Moore, A.W., Zuev, D., Crogan, M.: Discriminators for use in flow-based classification. Technical report, Intel Research, Cambridge (2005)
13. Oppenheim, A.V., Schafer, R.W., Buck, J.R.: Discrete-time signal processing, 2nd edn. Prentice-Hall, Englewood Cliffs (1999)
14. Sen, S., Spatscheck, O., Wang, D.: Accurate, scalable in-network identification of p2p traffic. In: Proc. WWW 2004 (2004)