Abstract—This paper addresses a simple question: Is there a behavior pattern of an individual node in peer-to-peer systems? It is known that peer-to-peer systems collectively show daily patterns, but the behavior patterns of individual nodes have seldom been studied. If individual nodes have their own behavior patterns with reasonable accuracy, we could greatly improve the efficiency of the system by reducing the overhead to handle unexpected random node failures.

Even though there have been many empirical studies on peer-to-peer systems, most of them are focusing on a collective view to characterize the whole systems in terms of average availability, average session length, and so on. In this paper we present Peer Availability Table (PAT) which is a model to represent a behavior pattern of an individual node based on the measurement of node availability. To judge the existence of the behavior pattern, we measure the performance of PAT using a binary classification test. By answering the basic question, we provide a useful hint for the design of peer-to-peer systems.

Index Terms—Peer-to-peer system, behavior pattern, peer model, availability, KAD

I. INTRODUCTION

Peer-to-peer (P2P) systems provide high scalability and reliability by using peers donated resources, including computing power, network bandwidth, and disk space. Many attractive services have emerged by exploiting it. P2P systems have a significant overheads to manage unexpected node failures, since the fundamental nature of the systems: peers can join and leave at any time without any notice. However, to find out a behavior pattern of an individual node can bring high benefits for reducing these costs. For example, reducing the number of redundant replicas for distributed storage system, the frequency of refreshing time to content-publish for file sharing system, and so on.

First question is, ‘Is there a behavior pattern of an individual node in P2P system?’ If so, how to model the pattern and use it to predict node behavior? We will answer this question. Though it is very important to know that the analysis of individual nodes is inadequate compare to the collective view of nodes. The findings about collective nodes had revealed that there are a diurnal pattern and weekly pattern based on the variation of its population [1], [2], [3], [4]. However these methods are too general to predict a node behavior.

In this paper we emphasize that whether a behavior pattern of an individual node does exist or not. To judge the existence of it, we present a novel technique which is a using a node availability to represent a behavior pattern of an individual node, and a test tool to evaluate the accuracy. Namely, for each node, its behavior pattern is modeled as PAT by analyzing and normalizing its trace data. To show the presence of the behavior pattern, we compare the trace data and its PAT via a binary classification test.

The rest of this paper is organized as follows. Section II describes related works, and then we briefly introduce the KAD trace what we make full use of it. In Section IV, our model to represent the behavior pattern and its verification method are explained. After the analysis of results, we summarize this paper in Section VI.

II. RELATED WORK

To analyze the behavior pattern of nodes, a measurement of P2P system should be done first. The measurement techniques fall into two categories: active and passive probing. Passive ways are constrained to measure a small set of controlled peers to study the traffic pattern and peer dynamics. While, most uses an active network probing method to detect availability of nodes. Steiner et al. studied KAD trace by a global view and peer view [4]. Moreover they generously provide its raw trace data, so we can exploit it in our research. Measurements [5], [6], [7], [8], [9], [10] characterize the P2P file sharing system traffic over the Internet, including Napster, Gnutella, KaZaa, and BitTorrent systems. Not only P2P system, but Bolosky et al. described the host availability of over 50,000 PCs belonging to the Microsoft [11] and Simache et al. studied a UNIX workstation in a distributed environment [12]. But they mainly focused on the collective analysis of those systems not the individual nodes.

Doucet performed a meta-analysis of availability data [1], examining the Microsoft, Gnutella, and Napster traces and revealed two broad pattern of node availability; first, those who are always online, whereas those in the second have diurnal pattern. Bhagwan et al. studied nodes in the Overnet DHT and found diurnal patterns [2], and Tian et al. [3] [13] studied the dynamic pattern of the Maze system. Additionally Steiner et al. discovered weekly pattern in the KAD [4].

A few models of behavior pattern, which is based on the conservative analysis, had typically shown patterns using static host availability. Doucet et al. expressed an availability of node in terms of its fractional downtime [14]. Bhagwan et al.
assumed a pessimistic mean availability based only on hosts online at night [15]. In [16] also used conservative, average-based metrics in their model. Such estimates are useful in a simulation environment, however, the host availability varies widely in real P2P environment, i.e., highly heterogeneous. So that is hard to predict a behavior pattern of an individual node. Other models, which is based on fitting observed traces to well-known statistical distributions such as the Pareto or Weibull distribution, can only provide limited information about how long a random node will stay online. These models cannot predict changes in the host availability over time.

Adar et al. [17] analyzed user traffic in Gnutella and concluded that there is a significant amount of free riding in the system. The tragedy of the digital commons, i.e., free riding, also must be concerned, but it is beyond the scope of our paper. Mickens et al. presented a predictor of individual node availability and its applications [18] using signal analysis and information theory. They developed history based predictors, but the length of prediction is restricted to a few bits for storing a history data. Therefore, their predictability had shown a good result at less than 6 hour look-ahead period even in very stable environment such as nodes belonging to the Microsoft, about 60% nodes were being always online. While they had focused on the predictability, we highlight that whether a behavior pattern of an individual node is existed or not, and its model to represent them for longer period with high resolution.

III. KAD TRACE

KAD is a P2P network, which implements the Kademlia protocol [19] based on distributed hash table (DHT). The users in KAD are also connected to the eDonkey network for publishing and searching, so counting a million or more simultaneous nodes. In this paper, we use a KAD trace from [4] since they provide a rich result which came from the measurement with a high frequency probing and many target nodes. The summary of measurement is described in Table I. Already [4] had shown the results to characterize KAD in terms of metrics such as arrival or departure process of peers, session and intersession lengths, availability, and lifetime. However we have an insight into the data to obtain novel implications, which are omitted or neglected.

A. Data Filtering

<table>
<thead>
<tr>
<th>Period of measurement</th>
<th>2006-09-23 ~ 2007-03-20 (179 days)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Crawl frequency</td>
<td>5 minutes (288 snapshots a day)</td>
</tr>
<tr>
<td>Number of nodes</td>
<td>400,375</td>
</tr>
</tbody>
</table>

TABLE I
ORIGINAL DATASET OF THE CRAWL

It is impossible to analysis the whole nodes in measurement, so the previous work [4] mainly used a subset of nodes that had seen for the first time on first day, because that nodes have slightly longer life time. However, we apply different terms for filtering. Since new nodes join the system for every day, filtering by first seen day extremely limits the total number of target nodes. Thus we filtered out nodes which fail to meet the sampling conditions: nodes that had seen more than once during the last 4 weeks of the measurement period. In this process no preliminary findings were considered to preserve neutrality while data filtering. After data filtering, we get a larger size of subset which has about 9 percent (36,284 nodes) of the whole nodes in the measurement. In addition, we adjust the period of measurement to make its length to 175 days (fully 25 weeks). Namely, remove the data from first day to fourth day during the timescale of the measurement.

B. Data preview

In this section, we show a preview of the sampled nodes which mainly focus on our analysis. In Figure 1(a), approximately 20% of nodes have first seen on the first day, and then linearly increases before the last 4 weeks. Unlikely, the last seen day is crowded around the last 2 weeks. To evaluate a life time we used the time period from the first seen day to last seen day during measurement. The mean of the life time of nodes is about 103.2 days with standard deviation = 53.1 days and median = 105.0 days as shown in Figure 1(b). Compared with the whole data, that means the life times of the sampled nodes are evenly distributed. Figure 1(c) shows the session length which is shifted to slightly longer. Long session time means that nodes are staying for a long time on the system and that makes the system much stable.

Figure 1(d) shows the host availability, which calculated by dividing the number of probes that a node responds to by the total number of probes for each node. The collective average of host availability is only 0.106 with standard deviation = 0.152, and median = 0.05. About 80% of nodes have less than 0.15 host availability but 0.4% of nodes have more than 0.9 host availability. This result shows us the most part of trace are full of offline states, i.e. unavailable. At this point we simply found two behavior patterns: online or offline in
most of the time. This stems from the more the number of probes, the greater the chances of a node being unavailable [2], i.e., that shows a low availability. So, we plot another host availability whose denominator is changed to their life time. As a result, the average of host availability increases to 0.177 with standard deviation = 0.208, and median = 0.09. Therefore, we should consider properties of individual node rather than a given collective average to fully utilize the unbiased host availability. Also, the host availability decreases over longer periods of time motivates the need for periodic refreshes of measurement.

IV. MODELING OF BEHAVIOR PATTERN

In this section, we describe a model to represent a behavior pattern of an individual nodes. Many empirical studies already have shown a diurnal and weekly pattern of nodes. These claims stem from the population analysis of nodes, namely the number of nodes which are online peaked when at midnight during a day and the weekend during a week [4]. However, it is not a behavior pattern of an individual node but an usage pattern of a whole P2P system. To answer whether there is a behavior pattern of an individual node, we design a peer model of the behavior pattern. In our model, a node has a finite state, i.e., online or offline denoted as 1 or 0. There are some issues to model a behavior pattern by using this binary value. First, a resolution of the model that highly depends on the frequency of probing. For instance, if a crawler probes every hour, the upper bound of resolution is 1 hour. Second, a length of the model which means a capacity to show how long a periodic pattern is. For example, if a length of model is 1 day, it could only represent a diurnal pattern. Lastly, a threshold of the model to mark as online at a time slot. Assume that at time slot $i$, the total number of probes is 10 and the number of responds as online is 6. Likely, at time slot $i + 1$, the total number of probes is 10 and the number of responds as online is 4. Then, if a threshold to determine a status as online had configured to 50%, the time slot $i$ will be marked as online while $i + 1$ will be marked as offline. We discuss the effects of these factors, followed by a description of our model of behavior pattern.

A. Peer Availability Table

We apply our previous work to use as a peer model [20]. We develop Peer Availability Table (PAT) which is a model to represent a behavior pattern of an individual node by exploiting a measurement of host availability, and a test tool to evaluate its accuracy. A default PAT indicates a node’s availability at every five minutes for a week length with 50% threshold. So an hour has 12 time slots, a day has 288, and a week has 2,016. Of course, this configuration can be freely changed to apply to other systems. The building process of PAT for each node is as follows. Divide a whole trace data according to the length of a week and adjust the measurement resolution if needed. Count, and then, the number of 1s (probed as online) for each time slots. Finally, normalize PAT by a given threshold. So now, we can discover behavior patterns of individual nodes whose length is up to weekly pattern with very high resolution. To investigate an evolution method of PAT is one of the our future works, since the behavior pattern of nodes might be changed over time.

B. Evaluation of PAT

We adopt a binary classification test, which is the task of classifying the members of a given set of objects into two groups on the basis of whether they have some property or not [21]. In our case, the classification property is the host status, i.e., online or offline. We define Positive as online and Negative as offline. For this test, there are two input streams; pairs of corresponding time slots, one from a real trace and other from its PAT (prediction outcome).

In that setting:

- $TP$ (True Positive) : PAT predicts that a node is online and the real trace also online (correct).
- $FP$ (False Positive) : PAT predicts that a node is online but the real trace offline (incorrect).
- $TN$ (True Negative) : PAT predicts that a nodes is offline but the real trace also offline (correct).
- $FN$ (False Negative) : PAT predicts that a nodes is offline but the real trace online (incorrect).

To measure the performance of the test, the concepts which are Sensitivity, Specificity, Accuracy, Positive Predictive Value (PPV), and Negative Predictive Value (NPV) are often used [22] [23]. Derivations for each terminology are:

$$Sensitivity = \frac{TP}{TP + FN} \quad (1)$$

$$Specificity = \frac{TN}{FP + TN} \quad (2)$$

$$Accuracy = \frac{TP + TN}{TP + FP + FN} \quad (3)$$

$$PPV = \frac{TP}{TP + FP} \quad (4)$$

$$NPV = \frac{TN}{TN + FN} \quad (5)$$

The meanings of these metrics will be explained in Section V. The result of this test is very important to determine whether a behavior pattern of an individual node is existed or not. For instance, if a node has a clear behavior pattern, it would be well represented by PAT. Thus, fewer FP and FN are counted, that result in a high Accuracy. In the other words, the measured performance of PAT is used to judge whether individual node clearly has a behavior pattern or not.

V. RESULTS

We evaluate the performance of PAT, using the binary classification test [21]. Tests for each individual node had done regarding their entire life time. The configurations of PATs are denoted as $<Length \: T \: Threshold \: A \: Availability>$. For instance, WeekT5A3 means that the length of PAT is 2,016 (a week) and the threshold is 50% and the filtering condition by host availability is 30%.
A. Results Overview

In this section we describe an overview of the results. Sensitivity measures the proportion of actual positives which are correctly identified as such (i.e., the percentage of the number of online conditions in real trace which are predicted as online by PAT). With higher sensitivity, fewer online conditions in real trace are labeled as offline. Specificity measures the proportion of negatives which are correctly identified (i.e., the percentage of the number of offline conditions in real trace which are predicted as offline by PAT). With higher specificity, fewer offline conditions in real trace are labeled as online. In theory, sensitivity and specificity are independent in the sense that it is possible to achieve 100% in both. In more practical, less contrived instances, however, there is usually a trade-off, such that they are inversely proportional to one another to some extent [22]. PPV is the proportion of true positives out of all positive predictions. NPV is the same, but for negatives, naturally.

Among these metrics, we should mainly focus on Accuracy, since it shows how well the behavior pattern is represented by PAT. Thus, getting high Accuracy means that nodes clearly have behavior patterns. However, PPV is the most important measure to evaluate a performance of PAT as a predictor. According to the host availability which is very low value, most part of trace is full with offline conditions. It is very hard to predict that a node is really online which happens infrequently. Therefore, PPV, which reflects the probability that a positive test reflects the underlying condition being tested for, shows the ability to predict that a node is really online. We inspect the results in details followed by sections.

B. Presence of Behavior Patterns

To show the presence of a behavior pattern, we compare the real trace data and its PAT (predicted data) via a binary classification test. Since the behavior pattern is modeled as PAT by analyzing and normalizing its trace data for each node, if a node has a definite behavior pattern, which is well represented by PAT. So the test would produce a good outcome. Otherwise, i.e., if a node has a random or very indistinct pattern, that leads to making incorrect PAT. That is to say the test would have poor result. Figure 2 provides the presence of a behavior pattern of an individual nodes clearly. For all nodes, the mean Accuracy of DayT5A0 is 0.874 and median is 0.917. WeekT5A0 has 0.881 mean Accuracy and 0.920 median. These results are impressively high values. About 55% of nodes have more 90% Accuracy in both DayT5A0 and WeekT5A0 case. (It will be explained in next section that the latter is better than the former.) Besides, remains shows moderated Accuracy which mean Accuracy is about 0.75. These result allow us to get a chance to predict a behavior pattern of an individual nodes for reducing a wasteful cost of P2P systems.

C. Types of Behavior Patterns

Daily and weekly patterns have shown by the population of nodes [4]. However, we suspect that not all nodes have the patterns as such. We discover that two types of behavior patterns, by analyzing all the individual nodes PATs: Static pattern and Dynamic pattern. About 86.3% of all nodes have a static pattern which means that no fluctuation between online and offline. Moreover, 93.7% of among them have all 0 PAT, and 6.3% have all 1 PAT. We could not categorize them into a diurnal or weekly pattern, but a static pattern. Only 13.7% of all nodes have a weekly pattern as well as daily pattern. We depict the weekly patterns of individual nodes whose the standard deviation of the host availability is a range in 0.4~0.6 of PAT. As we can see the distinctive weekly patterns in Figure 3.

D. PAT Length Effects

The length of PAT is a capacity to represent a behavior pattern of a node. Namely, Day PAT, Length = 288, can
represent for daily pattern, but Week PAT. Length = 2,016, can represent for weekly pattern as well as daily pattern. Therefore, if Accuracy of Week PAT is higher than Day PAT, it refers that the behavior pattern is longer than a day. Of course we can extend the length to discover longer pattern, we have not yet proceeded due to lack of a trace data which are fairly long period with high resolution. Even so, the length of a week is reasonably applicable in general. Since all Week PATs show better results for all Day PAT configurations, we conclude nodes have weekly pattern. (As shown in Figure 2, even though the difference between WeekPAT and DayPAT seems to be small, but the number of tests are 7 times more (Day vs. Week). Considering it, WeekPAT has much better results.) So we should consider the behavior patterns of nodes as weekly pattern in order to predict precisely according to its variation by day of the week.

E. PAT Threshold Effects

PAT is normalized by a given threshold. Lower threshold cause more predictions as being online (even that has happened with low frequency), but higher threshold cause more predictions as being offline. So it has a big effect on Sensitivity and Specificity. According to the results, Threshold = 50% shows the best accuracy. Figure 4 shows the main cause of this result. Since most traces are filled with 0s (offline), the key factor is related with the prediction of the 0s, which affects TN, FN. Too optimistic threshold makes TN smaller, and pessimistic one makes FN larger. Therefore, Threshold = 50% rare susceptible to TN, FN is a reasonable parameter. In this case, we use the simple threshold as a proportion value only, but a weighting on it must be considered such as freshness or life time of the trace data.

F. Host Availability Effects

To show the effects on Accuracy according to the host availability, we draw the results in Figure 5. Put nodes to divide by the host availability into 3 groups, i.e., 0~0.3, 0.3~0.7, and 0.7~1.0. The higher and lower availability groups have shown high accuracy, while the middle group has shown low accuracy. Trace data and its PATs of the higher group are made up with 1s mostly, and of the lower group are made up with 0s mostly. So they have very high Accuracy. However, PATs of the middle group, which has approximately 0.5 host availability, are sensitive to the Threshold. In Threshold = 50% and host availability is near 50% case, the time slots of PAT are identified as 1 or 0 by a few differences. It makes more errors, FP, FN, unlike the higher and lower group. The middle group has low Accuracy on this account, nonetheless, not too bad to decide that the middle group has no behavior pattern (its mean Accuracy is about 0.75). Thus, we can improve Accuracy if we apply the suitable parameters when building PAT for the middle group which are highly dynamic nodes. For instance, a strategy for highly available group is mainly focused on reducing FN, and for dynamic group is focused on reducing FP. The implication of this result is that we should make suitable strategies to handle groups by the host availability.

G. Filtering Effects

In case of KAD trace, what we used, up to 50% nodes have less than 5% availability, it means that a node could use just 8.75 days during total 175 days. These nodes are pretty much useless when to exploit it in practice. Then, what is the reasonable criteria to filtering out by the host availability? Figure 6 provides an insight into the answer. The suffix − of WeekT5A− refers that filtering out a node if its host availability is less than 0.5. Sensitivity increases along with the filtering criteria but Specificity decreases. As we mentioned above, Specificity is a dominant factor of Accuracy. If so, Accuracy should drop continually. However WeekT5A4 is an inflection point of Accuracy to rise. This result shows that nodes, whose host availability is less then 0.4, are too dynamic to represent by PAT, so we should filter out them; even so, there is plenty of
measurement of node availability, has a meaningful logicality. Present a behavior pattern of an individual node based on the number of sessions, and session length. Therefore, PAT, which represents a correlation between these metrics with lifetime, the number of sessions, and session length, the highly sensitivity (0.867), specificity (-0.858), and accuracy (-0.756) are highly calculate correlations for each metrics. Obviously, Sensitivity a relationship between the evaluated metrics and that, we find a pattern model is more precise length rather than daily one for dynamic patterns. Lastly, to predict a behavior pattern of an individual node with high accuracy, we should consider the properties for individual nodes, especially the host availability. By exploiting a behavior pattern of an individual node, in summary, we can filter out nodes, which are useless, and can utilize nodes, which are proper and timely.

VI. CONCLUSION

P2P systems can provide high scalability and reliability, but that has a significant overheads to manage unexpected node failures. However, if individual nodes have their own behavior patterns with reasonable accuracy, we could greatly improve the efficiency of the system by reducing the overhead to handle unexpected random node failures. Even though there are many empirical studies on P2P systems, the analysis of individual nodes is inadequate compare to the collective view. Compared with previous works, we emphasized that whether a behavior pattern of an individual node does existed or not. To judge the presence of it, we present a novel technique, Peer Availability Table (PAT), which is a model to represent a behavior pattern of an individual node based on the measurement of node availability, has a meaningful logicality.

REFERENCES