A Task-Based Model for the Lifespan of Peer-to-Peer Swarms

Yong Zhao[123] Zhibin Zhang[2] Ting He[4] Alex X. Liu[5] Li Guo[1] Binxing Fang[1]

[1] Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
[2] Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China
[3] Graduate University, Chinese Academy of Sciences, Beijing, China
[4] IBM T.J. Watson Research Center, Hawthorne, NY, USA
[5] Department of Computer Science and Engineering, Michigan State University, MI, USA

zhaoyongcool@gmail.com, zhangzhibin@ict.ac.cn, the@us.ibm.com, alexliu@cse.msu.edu

Abstract. Peer-to-Peer (P2P) techniques are broadly adopted in modern applications such as Xunlei and Private Tracker [1, 2]. To address the problem of service availability, techniques such as bundling and implicit uploading are suggested to increase the swarm lifespan, i.e., the duration between the birth and the death of a swarm, by motivating or even forcing peers to make more contributions. In these systems, it is common for a peer to join a swarm repeatedly, which can introduce substantial bias for lifespan modeling and prediction. In this paper, we present a mathematical model to study the lifespan of a P2P swarming system in the presence of multi-participation. We perform evaluations on three traces and a well-known simulator. The result demonstrates that our model is more accurate than previous ones.

Keywords: peer-to-peer; modeling; evolution; lifespan

1 Introduction

Peer-to-peer (P2P) systems have seen a tremendous growth in the past decade for its scalability and high downloading speed [3]. They are widely used for content sharing and online video streaming. The lifespan of the P2P swarm for a resource is defined as the time duration from the time that the resource is shared in the system to the time that the number of peers in the swarm becomes below a predefined threshold, such as one.

To improve the availability, extending the lifespan of P2P swarms is critical for modern P2P systems. For example, peer-assisted systems, such as FS2You [4] with both dedicated servers and peers, if a swarm died servers must afford all the uploading bandwidth. It's reported that cold files that involved little peers consume 54% of the bandwidth of servers in FS2You system [4]. As another example, for private tracker systems, peers have to maintain a certain uploading-downloading ratio for better availability (*i.e.*, they must upload more in order to download more). But a selfish peer with high uploading bandwidth tend to leave quickly once it achieves the ratio, which can hurt the swarm lifespan aggressively. In this work, to better understand the lifespan of P2P swarms, we model P2P swarm evolution from a nontrivial view and figure out the important factors that impact lifespan, which is important for P2P system design and performance tuning.

Prior studies on P2P swarm model are based on single-participation, *i.e.*, assuming that a peer joins a swarm only once, [5, 6]. However, measurement studies observe that this assumption does not hold in reality [7, 8]. In modern systems, such as Xunlei [1] and private tracker systems [2, 9], reward generous peers who provide more uploading by giving them higher downloading speed and punish selfish peers. Generous peers are motivated and selfish peers are forced to share resources with other peers. Therefore, we observe that a peer often participate a swarm repeatedly, which can be the main source of bias for the prior models. Though a peer may join one swarm multiple times, only the first participation can be modeled and others are ignored. Consequently, only the interval between the first arrival and leave can be counted in the model as the peer online time. These limitations will make the prior models underestimate the peer online time and the swarm lifespan. To our best knowledge, the only prior work on P2P swarm modeling that assumes multi-participation is by Menasche *et al.* [8]. They modeled the content availability by a new metric called "busy period", which is the uninterrupted intervals during which the content is available. However, their model only depict the relationship between peer arrival rate and swarm lifespan, leaving other factors like peer online time out of their discussion.

In this work, to address multi-participation when modeling swarm lifespan, we combine the series of participation of the same peer into a single process called *task* to incorporate the interrelationship of consecutive behavior of the same peer. By regarding each task as an alternating renewal process that switch between online and offline, the number of active peers of a swarm can be obtained during the evolution. As the lifespan of a swarm is the duration between the swarm birth and death, and as the birth point can be observed easily, we derive the death point through solving the evolution equation by setting active peers as a threshold. We show that by employing a subexponential decaying process approximation in the model, a closed-form solution can be obtained. As lifespan is very difficult to measure and predict in reality, we present a new lifespan metric according to our model, half-life, which is defined as the time in which the number of active peers decreases from a start to its half. The evaluation based on real traces and extensive simulations verifies that our model is more accurate than the state-of-the-art fluid model.

In summary, we make four major contributions. (1) *Task-based churn model*: We present a task-based churn model by combing the series of participation of the same peer into a task and characterize the task-based churn.(2) *Task-based evolution model*: We present a novel model to depict swarm evolution based on task-based assumption. The model can be used to analyze the lifespan of swarming systems. (3)*Lifespan model*: We present a closed-form solution of lifespan through approximation and an efficient metric, half-life. (4)*Experimental validation*: We perform extensive experiments on both three real traces and simulations to compare the accuracy of the task-based model with the fluid model. The results show that our model is more accurate then prior ones.

The remainder of the paper is organized as follows. Section 2 presents related works on swarming systems. Then we propose our task-based churn model in Section 3. In Section 4, we demonstrate our evolution model and the lifespan model. In addition, we also propose a half-life based method to measure lifespan more efficiently. Section 5 and 6 present our experiment setup and results. Finally, we conclude the paper.

2 Related Works

Although many studies aimed to model or improve the availability of P2P systems, most of them made unrealistic assumptions to bypass the complexity. Before introducing related models, we present some measurement studies of real systems. Daniel *et al.* have performed comprehensive measurements on Gnutella, KAD and Bittorrent systems [7]. In their paper, three of important conclusions are highly related with our work. First, they have found that the inter peer arrival time follows exponential distribution. Second, the online session length is better described by Weibull distribution. The third is that past session length of a peer is a good predictor of the rest, which means consecutive behaviors of the same peer are related. In modern P2P systems, peers have new patterns according to the measurements of Private Tracker systems [2, 9], or more specifically peers are more patient, active and eager to upload than before.

For P2P evolution modeling, Qiu *et al.* have proposed a simple fluid model to describe BitTorrent-like system and studied the steady-state network performance [5]. Based on extensive measurements on real BitTorrent systems, Guo *et al.* have found the peer arrival rate follows the exponential process and modeled the swarm lifespan with an improved fluid model.[6]. Kaune *et al.* also focused on the availability in modern systems. They performed widespread measurements and found that seeders have a significant impact on swarm availability [10]. Then they tried different incentives to improve the availability and gave a comparison study on them [11].

3 Task-Based Churn

In this section, we firstly present the task definition and task-based churn, then we model task-based churn with the help of three characteristics: task arrival rate, task duration and task availability.

3.1 Task

In modern P2P systems, incentives are exploited to encourage peer's more and longer contribution to enhance the availability of the whole system, which lead the multi-participation to prevail. Take Xunlei for example, which is the most widely used private P2P system in China, peers are forced to upload their downloaded files by implicit uploading every time they join it. This phenomenon of

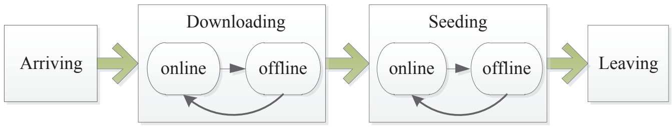

Fig. 1: A peer lifecycle in a task

multi-participation is also mentioned in other literature [7, 8, 12]. However, to the best of our knowledge, no evolution or lifespan model was provided in the presence of multi-participation.

To address multi-participation, we define the process starts from a peer's first arrival to the last departure of a swarm as one basic unit called *task*. When a peer ends a task in a swarm, the peer will no longer be back to the swarm again. Figure 1 shows the lifecycle model of a peer in a swarm during a task. More specifically, a peer will experience four states in its lifecycle in a task: Arriving, Downloading, Seeding and Leaving. Compared with the session defined in fluid model, assuming that a peer has single participation in a swarm and should not be back after the first departure, task can be more general, insightful and practical.

In order to better understand multi-participation, we collected tracker log trace from a nationwide private tracker system in China (cgbt trace). Users in this system should hold an upload-to-download ratio to maintain its access right. The system currently has 116,679 registered users and 132,777 torrents. The average upload-to-download ratio is 2.8. The trace contains all peer requests that were posted to the tracker from June 1, 2010 to July 4, 2010. In the trace, there are 447,141 swarms and 843,242 peers all together. The result in Figure 2 shows that more than 75% of peers join the same swarm repeatedly. Furthermore, 30% of peers participate in the same swarm for more than 10 times, which suggests that they are very "patient". This finding shows the limitations of the fluid model of characterizing the swarm with impatient peers.

3.2 Modeling and Characterization of Task-Based Churn

The dynamics of peer activities when we view it in a task perspective is called task-based churn. In order to build task-based evolution model, we are interested in three characteristics of tasks in the churn: task arrival rate, task duration and task availability.

Task arrival rate profiles the pattern of task arrivals. As the downloading request of a torrent file, which can be regarded as the creation of tasks, decreases exponentially [6], we assume that the task arrival rate of a swarm follows an exponential decreasing rule with time t.

$$\lambda(t) = \lambda_0 \exp(-\frac{t}{\tau}), \tag{1}$$

where λ_0 is the initial arrival rate and τ is the attenuation parameter. In other words, τ indicates the decreasing speed of task arrival rate. To consolidate our assumption, we fit the arrival rate of each swarm in cgbt trace and plot all

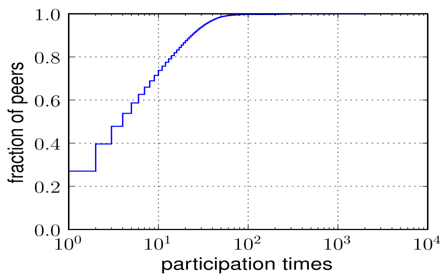

Fig. 2: The CDF of peer's participation counts

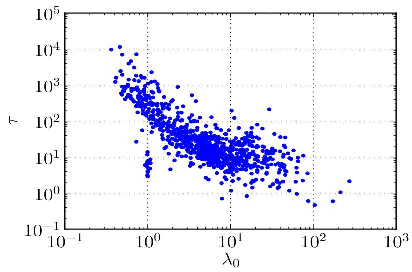

Fig. 3: The parameters of exponential fitting for each swarm

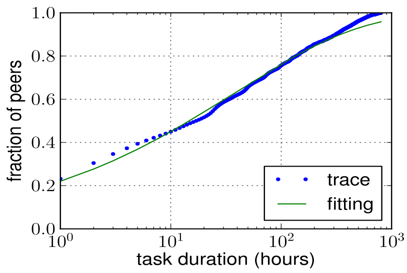

Fig. 4: The CDF of all peers' task durations

the parameters (λ_0, τ) in Figure 3. In this figure, x-axis denotes λ_0 and y-axis denotes τ. We can find that τ has an upper bound (1000) except for the cold swarms with too small task arrival rate to make a good fit, which implies the task arrival rate fits a exponential decreasing function well.

Task duration depicts how long a peer remains in the swarm from a task view. We assume that the task duration follows Weibull distribution, since it is widely used in survival analysis, including duration analysis and modeling, and can approximate a wide range of classes of functions including exponential, normal and lognormal only with two parameters.

$$F(x) = 1 - \exp[-(\frac{x}{\mu})^k], \tag{2}$$

where $k > 0$ is the shape parameter and $\mu > 0$ is the scale parameter of the distribution. Different value of k can lead to different types of distribution. For cgbt trace, we collect the lengths of all tasks, each of which includes several online and offline states, and plot the cumulative distribution of task durations with their Weibull fitting in Figure 4 in log-linear scale. For most part of the distribution, the Weibull distribution is able to provide tight fitting except for the durations above 600 hours. The reason is that we cut off the tasks, which are still alive after our trace stop.

Task availability is the proportion of online states in the task duration, which indicate the peer online probability. Since the task duration includes online and

Table 1: Notations for our model

λ_0	the initial value of task arrival rate
τ	the attenuation parameter of task arrival rate
μ	the scale parameter for the distribution of task duration
k	the shape parameter for the distribution of task duration
a	the task availability (the proportion of online states in the task duration)
$N(\cdot)$	The evolution of a swarm (The average number of online tasks)
$HL(\cdot)$	The half-life of a swarm
L	The lifespan of a swarm

offline states, we use an alternative renewal process to model the behavior of each task. With the help of task availability, we can better understand how offline states contribute to the dynamics of the system, which is not considered by the fluid model. We define the task availability as

$$a = \frac{T_{on}}{T_{on} + T_{off}}. \tag{3}$$

where T_{on} is the mean of online state length and T_{off} is the mean of offline state length.

4 Task-Based Lifespan Model

In this section, we first present the swarm evolution model for the view of entire and then obtain a closed-form solution of lifespan through approximation. For the convenience of reference, we list the meanings of the parameters of our model in Table 1.

4.1 Swarm Evolution

As the lifespan is determined by the number of online tasks in the swarm, or the swarm evolution, we firstly model the swarm evolution in the view of task.

We define X as a random variables that represents the task duration of a peer. Suppose a task starts at t_0, with the help of task duration distribution, we can obtain the probability at time t that the task is still in the swarm by:

$$P_{alive}(t_0, t) = Pr(X > t - t_0) = 1 - F(t - t_0) = \exp[-(\frac{t - t_0}{\mu})^k]. \tag{4}$$

According to our task-based churn model, the number of new task arrival or task arrival rate at time t is $\lambda(t)dt$. And with the help of $P_{alive}(t_0, t)$, we collect all the tasks that join the swarm before t and can obtain the number of online tasks at time t by:

$$N(t) = a\int_0^t \lambda(x)P_{alive}(x, t)dx = a\lambda_0 \int_0^t \exp[-(\frac{t - x}{\mu})^k - \frac{x}{\tau}]dx. \tag{5}$$

4.2 The Closed-Form Expression of Lifespan

Although the real swarm lifespan is very hard to be determined in practice, swarm creator or system operator usually kills a swarm when it has very few peers. Keep this intuition in mind, we set a threshold on the number of online tasks as the criterion of swarm death. Without loss of generality, we select 1 as the threshold in our model. Consequently, the lifespan can be obtained by solving the equation $N(t) = 1$, called the *evolution equation*, to obtain the death point t. Unfortunately, $N(t)$ is a transcendental function, which prevents us from solving the equation analytically. Hence, we have to make necessary approximation to the evolution to obtain the closed-form solution of lifespan.

Approximation The first step of our approximation is to expand the exponential term $\exp[-(\frac{t-x}{\mu})^k - \frac{x}{\tau}]$ which is a transcendental function. According to the Taylor series, we can know that

$$\exp[-(\frac{t-x}{\mu})^k - \frac{x}{\tau}] = \exp(-\frac{x}{\tau})\sum_{j=0}^{\infty}\frac{(-1)^j}{j!}(\frac{t-x}{\mu})^{kj}. \tag{6}$$

Then we can calculate the integration of the Taylor expansion. If we let $y = t-x$, there is only one variable in the summation.

$$\int_0^t \exp[-(\frac{t-x}{\mu})^k - \frac{x}{\tau}]dx = \exp(-\frac{x}{\tau})\sum_{j=0}^{\infty}\frac{(-1)^j}{j!\mu^{kj}}\int_0^t y^{kj}\exp(\frac{y}{\tau})dy. \tag{7}$$

Note that the integration in Equation 7 can be decomposed with a Kummer function, which has a known approximation [13]:

$$\int_0^t y^{kj}\exp(\frac{y}{\tau})dy = \frac{t^{1+kj}}{1+kj}M(1+kj, 2+kj, \frac{t}{\tau}). \tag{8}$$

Here $M(\cdot)$ is a Kummer function. And when one of the three parameters $(1 + kj, 2 + kj, \frac{t}{\tau})$ is large, and the other two remaining modest in magnitude, this function has a special approximation as follows [13]:

$$M(1+kj, 2+kj, \frac{t}{\tau}) \approx \frac{\Gamma(2+kj)}{\Gamma(1+kj)}(\frac{t}{\tau})^{-1}\exp(\frac{t}{\tau}). \tag{9}$$

As τ is a constant during the swarm evolution, the above approximation condition can be met if t is large enough. Hence, we apply the approximation in the situation that t/τ is large enough and positive with $1 + kj \neq 0, -1, -2, \ldots$ [13]. In this way, the transcendental function can be simplified to a simple exponential function.

$$N(t) \approx a\lambda_0\tau\exp[-(\frac{t}{\mu})^k](t/\tau \text{ is large and positive}). \tag{10}$$

Finally, we can solve the transcendental equation and obtain the swarm lifespan with the start at 0 as:

$$L = \mu[\log(a\lambda_0\tau)]^{1/k}. \tag{11}$$

The result shows that μ, with a linear influence on lifespan, is the most significant parameter. And this equation suggests that the designers should try to encourage users to hold the task as long as possible.

4.3 Lifespan Prediction and Measurement

To predict and measure lifespan can be very difficult in reality, because the swarm death point is hard to be determined. In the fluid model, an explicit point that indicates the death of swarm is obtained to predict lifespan. However, this result relies on the single participation assumption, which is unrealistic. And it can be biased by the temporary or accidental leave of peers for a dying swarm with very few active peers. To address prediction, recalling the condition of our approximation, if t is large enough compared with τ, we can regard the evolution after t as a subexponential decay process. This implies us that if we choose a time point large enough in the swarm evolution as a start, we can use subexponential decay process to approximate the rest evolution process. Experientially, if $t/\tau > \pi$ can be met we can safely apply the approximation for predicting the swarm death point and lifespan.

To apply the above prediction, we have to estimate five parameters $(\lambda, \tau, \mu, k, a)$. To make it more efficient, we suggest to use half-life, which is the time that the number of online tasks decrease from the value at starting point to its nth half, to indirectly measure swarm lifespan based on our model. Half-life depicts the swarm decreasing speed and can be expressed as follows:

$$HL(n) = t_{\frac{1}{2^n}} - t_0 = \mu[(\frac{t_0}{\mu})^k + n\log(2)]^{1/k} - t_0, \tag{12}$$

where t_0 means the time of starting point and $t_{\frac{1}{2^n}}$ means the time when the evolution decreases by n folds. Therefore, the lifespan can be viewed as the function of $N(t_0)$ and $HL(n)$:

$$L = t_0 + HL(n). \tag{13}$$

By solving $2^n = N(t_0)$ with respect to n, we can derive the lifespan according to Equation 12 and Equation 13 with the start at 0. When predicting lifespan in real measurement, we firstly need to record the evolution of a swarm whose duration is long enough to include the decreasing part of evolution. Then we select a point in the decreasing part and label it as t_0. Since we have two unknown parameters (μ and k) in the Equation 12, we need to select at least two other point in the decreasing part to determine the unknown parameters (*e.g.* $t_{0.5}$ and $t_{0.25}$). In this way, we can obtain the prediction of lifespan according to Equation 13.

5 Trace Evaluation

In this section, we compare the accuracy of our evolution model with the fluid model on three tracker traces. Besides using the cgbt trace, we also introduce two other traces available on the Internet. One trace (alluvion trace) is statistics pages from two large trackers (www.alluvion.org and www.crapness.com), late 2003 to early 2004. After parsing these web pages, we found 96,339 swarms and 417,166 peers all together. The other one (filelist trace) was collected from Filelist.org by scraping its website during the period from Dec 14, 2005 until Apr 4, 2006, and it was collected by Roozenburg and *et al.*. This trace contains data collected from 3,000 swarms and 2,172,738 sessions.

5.1 Experimental Setup

In order to compare our model with real traces, we analyze the traces and obtain all parameters by fittings. Our experiment has four steps as follows.

1. We split the traces according to swarms and remove the cold swarms with peers less than 100.
2. For each swarm, we collect the task arrival rate, task duration and the task online/offline state length of each peer. To compare with the fluid model, we also collect peer arrival rate and peer online time following the definition in prior study [6].
3. For each trace, we select one swarm randomly. By counting active peers in each selected swarm, we plot the real evolution and the two predicted ones that are calculated according to the two models in a figure (Figure 5, 6 and 7).
4. To show the accuracy for all swarms, we calculate the MSE (Mean Square Error) between the modeled evolution and the real one in each swarm. As the real evolution can periodically fluctuate in one day (time-of-day effects [14]) that can bias the MSE aggressively, we use 24 hours moving average as the real evolution and plot the cumulative distribution of all swarms' MSE in a figure (Figure 8, 9 and 10).

5.2 Accuracy of Task-Based Evolution Model

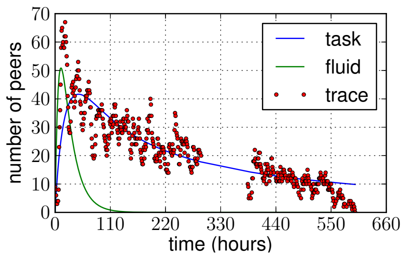

Fig. 5: The comparison between real evolution in cgbt trace and models

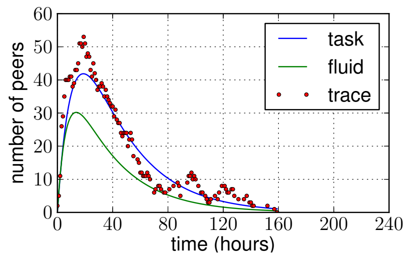

Fig. 6: The comparison between real evolution in alluvion trace and models

As shown in Figure5, we can see that the task-based model fits the real evolution very well, while the fluid model only captures the increasing part, which is only a short period after the swarm birth. The reason is the fluid model ignores the "extra" participations, causing significantly underestimation for the number of peers. Hence, the fluid model provides a shorter lifespan than the real one, which will be confirmed by our simulation later. In the other two traces, although the fluid model is more close to the real evolution than the it does in cgbt, task-based model is still better.

To evaluate task-based mode for all swarms, we show the cumulative distribution of the MSE for each swarm in Figure 8,9 and 10. The x-axis denotes the MSE between the model and real evolution of the corresponding swarm, while the y-axis denotes the fraction of swarms whose MSE less than or equal to the corresponding x value. Specifically, in the cgbt trace (Figure 8), the curve for the task-based model is on top of that for the fluid model, which means the MSEs of the task-base model in most swarms are smaller than those of the fluid model. In the alluvion trace and the filelist trace (Figure 9 and 10), our task-based model also yield s slightly better accuracy.

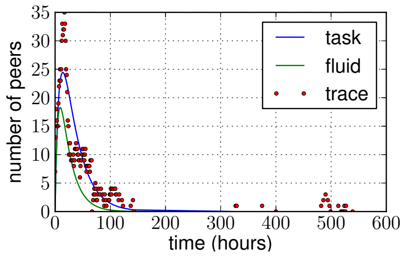

Fig. 7: The comparison between real evolution in filelist trace and models

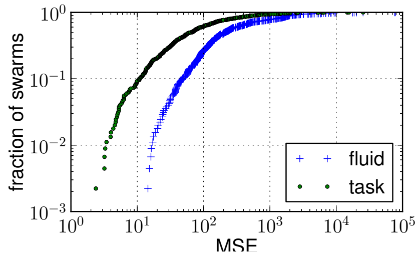

Fig. 8: The cumulative distribution of mean squared errors of models (cgbt trace)

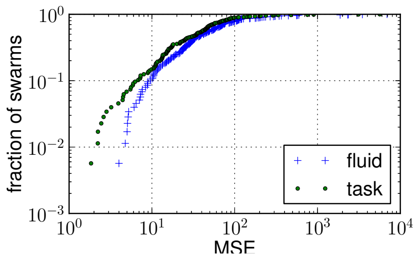

Fig. 9: The cumulative distribution of mean squared errors of models (alluvion trace)

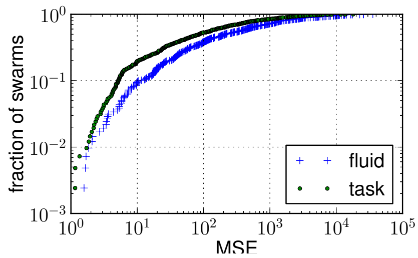

Fig. 10: The cumulative distribution of mean squared errors of models (filelist trace)

6 Simulation

In this section, we compare the accuracy of our task-based model with the fluid model by extensive simulations on a very famous simulator.

6.1 Simulator

As it is very difficult to collect a set of swarms with their whole lifespan recorded in a trace for time limitation, we will evaluate our task-based lifespan model through simulation. We choose OMNeT++ as the simulation platform, for it is an extensible, modular, component-based C++ simulation library and framework. As K. Katsaros *et al.* has developed a BitTorrent component for OMNeT++, which provided a set of modules that implemented a fully featured and extensible realization of the BitTorrent protocol [15]. The simulator creates a realistic simulation platform, which provides the simulation of underlying network structures. The architecture of the simulator includes full simulation of a BitTorrent system, such as the tracker, peer and the protocol [15]. But its churn model is borrowed from the fluid model [6]. We improve the BitTorrent component by making two important changes as follows:

1. In the original simulator, a peer participates in a swarm and then downloads files. When it ends the downloading, it behaves as a seed for a fixed time specified by a parameter. We add parameters such as μ, k, a, to simulate task behavior.
2. We extend the state machine of peer behavior so that peers could participate in a swarm repeatedly. When a participating peer finishes an online session, we make it sleep for the duration of an offline session and then wake it up for another online one. When the task duration is exhausted by online/offline sessions, we kill the peer.

6.2 Accuracy of Task-Based Lifespan Model

To compare the accuracy of task-based lifespan model with the fluid model, we perform an extensive simulation on real parameters, in which we analyze the swarms in real traces to obtain the parameters and their variation ranges related to task-based model and then we randomly generate parameters in the ranges. The results of 615 simulated swarms are plot in the Figure 11, showing the comparison of the swarm lifespan obtained from simulation with that from our model and the fluid model. In this figure, each point in x-axis denotes the real lifespan of a swarm, while each point in y-axis denotes the lifespan that obtained by models. The lifespan in the x-axis are sorted in non-descending order of the real lifespan. So the points lay on the line of $y = x$ mean the modeled lifespan equals to the real lifespan. As shown in the figure, our model fits the real lifespan very well, while the lifespan predicted by the fluid model is small than the real one.

7 Conclusion

Availability, or more specifically lifespan, is one of the most important issues in P2P systems. For lifespan model, existing studies are based on unrealistic assumptions of peer single participation. In this paper, to address multi-participation, which is more and more widespread in modern P2P systems, we

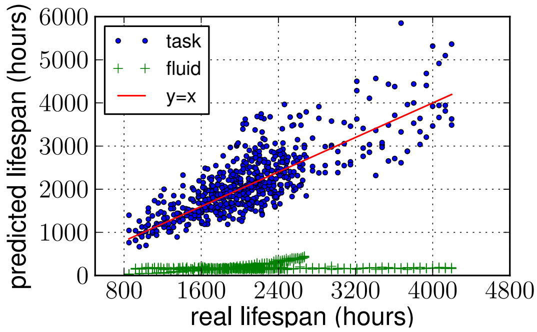

Fig. 11: The comparison of swarm lifespan: modeling and simulation

propose a novel task-based model, combining multiple peer participations into one task and regarding each task as an alternating renewal process that switch between online and offline, which is more general, insightful and practical. Based on the model, we derive a closed-form expression of swarm lifespan with the help of approximations and propose a new metric for lifespan measurement and prediction, half-life. The experimental evaluation based on real traces and extensive simulations verifies that o ur model is more accurate than the state-of-the-art fluid model.

Acknowledgement

We would like to thank the PDS group, TU Delft and the UMass Trace Repository for the alluvion trace and the flielist trace that were collected by them. This work is partly supported, by National Basic Research Program of China under grant No.2012CB315801, and by the National High-Tech Research and Development Plan 863 of China (Grant No.2011AA010705), and by the National Science Foundation of China (NSFC) under Grant 61070184, and by the Instrument Developing Project of the Chinese Academy of Sciences with grant No. YZ200926.

References

1. Yong Zhao, Zhibin Zhang, Yipeng Wang, Li Guo, and Binxing Fang. Performance evaluation of xunlei peer-to-peer network: A measurement study. In *Consumer Communications and Networking Conference (CCNC), 2011 IEEE*, pages 257 –261, jan. 2011.
2. Chao Zhang, P. Dhungel, Di Wu, Zhengye Liu, and K.W. Ross. Bittorrent darknets. In *INFOCOM, 2010 Proceedings IEEE*, pages 1 –9, 2010.
3. Klaus Mochalski Hendrik Schulze. Internet study 2008/2009. http://www.ipoque.com/resources/internet-studies/.
4. Ye Sun, Fangming Liu, Bo Li, Baochun Li, and Xinyan Zhang. Fs2you: Peer-assisted semi-persistent online storage at a large scale. In *INFOCOM 2009, IEEE*, pages 873 –881, april 2009.

5. Dongyu Qiu and R. Srikant. Modeling and performance analysis of bittorrent-like peer-to-peer networks. In *SIGCOMM '04: Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 367–378, New York, NY, USA, 2004. ACM.
6. Lei Guo, Songqing Chen, Zhen Xiao, Enhua Tan, Xiaoning Ding, and Xiaodong Zhang. A performance study of bittorrent-like peer-to-peer systems. *Selected Areas in Communications, IEEE Journal on*, 25(1):155 –169, jan. 2007.
7. Daniel Stutzbach and Reza Rejaie. Understanding churn in peer-to-peer networks. In *IMC '06: Proceedings of the 6th ACM SIGCOMM conference on Internet measurement*, pages 189–202, New York, NY, USA, 2006. ACM.
8. Daniel S. Menasche, Antonio A.A. Rocha, Bin Li, Don Towsley, and Arun Venkataramani. Content availability and bundling in swarming systems. In *CoNEXT '09: Proceedings of the 5th international conference on Emerging networking experiments and technologies*, pages 121–132, New York, NY, USA, 2009. ACM.
9. Xiaowei Chen, Yixin Jiang, and Xiaowen Chu. Measurements, analysis and modeling of private trackers. In *IEEE P2P 2010*, 2010.
10. S. Kaune, R.C. Rumi andn, G. Tyson, A. Mauthe, C. Guerrero, and R. Steinmetz. Unraveling bittorrent's file unavailability: Measurements and analysis. In *Peer-to-Peer Computing (P2P), 2010 IEEE Tenth International Conference on*, pages 1 –9, 2010.
11. S. Kaune, G. Tyson, K. Pussep, A. Mauthe, and R. Steinmetz. The seeder promotion problem: Measurements, analysis and solution space. In *Computer Communications and Networks (ICCCN), 2010 Proceedings of 19th International Conference on*, pages 1 –8, 2010.
12. Rameez Rahman, Tamás Vinkó, David Hales, Johan Pouwelse, and Henk Sips. Design space analysis for modeling incentives in distributed systems. In *Proceedings of the ACM SIGCOMM 2011 conference on SIGCOMM*, SIGCOMM '11, pages 182–193, New York, NY, USA, 2011. ACM.
13. Jerome Spanier and Keith B. Oldham. *An atlas of functions.* Taylor & Francis/Hemisphere, Bristol, PA, USA, 1987.
14. R. Bhagwan, S. Savage, and G. Voelker. Understanding availability. *Peer-to-Peer Systems II*, pages 256–267, 2003.
15. K. Katsaros, V.P. Kemerlis, C. Stais, and G. Xylomenos. A bittorrent module for the omnet++ simulator. In *Modeling, Analysis Simulation of Computer and Telecommunication Systems, 2009. MASCOTS '09. IEEE International Symposium on*, pages 1 –10, sept. 2009.