

Impact of User Behavior Modeling on Evaluation of Peer-to-Peer Systems

Konstantin Pussep, Sebastian Kaune, Christof Leng¹, Aleksandra Kovacevic, Ralf Steinmetz



TECHNISCHE
UNIVERSITÄT
DARMSTADT



KOM – Multimedia
Communications Lab

Technische Universität Darmstadt
Department of Electrical Engineering and Information Technology
Department of Computer Science (Adjunct Professor)
Multimedia Communications Lab
Prof. Dr.-Ing. Ralf Steinmetz

Impact of User Behavior on Evaluation of Peer-to-Peer Systems
Technical Report KOM-TR-2008-07

By
Konstantin Pussep, Sebastian Kaune, Christof Leng¹,
Aleksandra Kovacevic, and Ralf Steinmetz

Contact
konstantin.pussep@kom.tu-darmstadt.de
<http://www.kom.tu-darmstadt.de>

First published: November 12, 2008
Last revision: January 26, 2009

For the most recent version of this report see
<ftp://ftp.kom.tu-darmstadt.de/pub/TR/KOM-TR-2008-07.pdf>

¹ The author is affiliated with:
Databases and Distributed Systems, Department of Computer Science, Technische Universität Darmstadt

Abstract

Simulations are a common mean to investigate the performance of communication systems. They offer a “middle course” between analytical studies and real application, therefore, offering a higher level of detail than analytical method but being more abstract than real systems.

Simulative evaluations require a mature model of user behavior to estimate the system performance. This is even more the case with peer-to-peer (p2p) systems where large number of users act simultaneously as consumers and providers.

The evaluation of a peer-to-peer system is a challenging enterprise. Despite that the large amount of users are necessary to simulate and a realistic underlay model is required to simulate a large-scale geographically distributed system the *user behavior* is an important parameter of a proper evaluation model.

The significance of a proper user behavior model and its impact on system performance are often neglected in simulative studies. Often simple or random models are used to represent the system’s workload and churn. However, this can distort the results and a system that performs well in a simulation may fail in a real environment with real users.

In this paper we present a modeling framework of user behavior for simulative evaluation of p2p systems, focusing on the users’ lifetime and system workload. We compare different modeling approaches and study their impact on system performance in representative scenarios.

Contents

1	Introduction	4
2	System Model	5
3	Modeling Churn	7
3.1	Lifetime Models	7
3.2	Session Models	8
4	Workload Model	10
5	User Properties Model	11
6	Use Case: Kademlia	12
7	Evaluation	14
7.1	Methodology	14
7.2	Setup 1: Fixed Network Size, Variable Session Models	14
7.3	Setup 2: Fixed Session Duration, Variable Intersession Duration	16
7.4	Setup 3: Fixed Network Size, Variable Event Rate	16
7.5	Setup 4: Variable Arrival Processes	16
8	Related Work	19
9	Conclusions	20

1 Introduction

Peer-to-peer (p2p) systems are very popular on the today's Internet and the research in this area is substantial [?]. As there are plenty of approaches based on the paradigm a proper understanding of their benefits and drawbacks is important to estimate their applicability in real systems.

Therefore, the evaluation of peer-to-peer systems is crucial for understanding their performance and therefore their feasibility in the real world. Different techniques, such as testbed evaluations, analytical analysis, and simulations, can be used to evaluate system performance. Due to the inherent complexity and the high number of participants simulations are a very popular approach to simulate peer-to-peer (p2p) systems [?].

While simulations are a common technique for estimating the performance of large scale p2p systems with an acceptable level of detail, the modeling of the system is crucial to obtain realistic results. Aside from a proper model of the underlying network, a proper evaluation has to take user behavior into account, since it is a critical factor of the system dynamics. The modeling of user behavior is already important in traditional client/server environments, because it heavily influences the workload. In addition to their role as consumers, users become providers in p2p systems which aggravates their importance to system performance.

In general, a good model of user behavior for p2p systems has to capture both the consumer and provider nature of the participating peers. For example, churn, which describes peers connecting to and disconnecting from the system, has a severe impact on data availability and consistency, while resource distribution and popularity have a significant impact on load balancing. Therefore, a user behavior model must be abstract enough to allow efficient implementations but at the same time capture the most relevant components.

In this paper we describe our efforts to create a general user behavior model for simulations of p2p systems, and show how it can be applied to Distributed Hash Tables (DHTs), using Kademlia [MM02] as an example. We discuss a general model and the alternative approaches for individual components and present their impact on evaluation results. We hope to encourage the discussion of p2p user behavior modeling and its use in simulation. The goal is to improve the performance evaluation of p2p systems and thus the significance of research results. We reckon that proper user modeling will help researchers to discover shortcomings in their systems that would have been overlooked by less sophisticated evaluation.

Section 2 presents the modeling framework of user behavior in a peer-to-peer system. Churn modeling is presented in detail in Section 3. In Sections 4 we discuss the modeling of workload in p2p systems, while Section 5 shortly presents the modeling of user properties. The application of the modeling framework to a concrete system is shown in Section 6 together with the evaluation results in 7. Finally, the related work is presented in Section 8 followed by conclusions in Section 9.

2 System Model

Investigation of the p2p system performance requires, beside the actual system under test, modeling of two external components: underlying Internet used to model transmission times, loss rates etc. (see [?] for more details) and the behavior of the system users, who in fact create the load on the system.

The behavior of p2p users is rather complex. We aim to break it down into understandable and independent components. The main categories shown in Figure 2.1 are churn, workload, and strategy, which can be separated further into lifetime, sessions, requests, resources, and strategy.

Churn describes the participation dynamics of p2p nodes. Users join the network, leave, and come back several times. Sometimes they even leave forever because they do not use the system anymore or have reinstalled their computer. For example, in systems such as BitTorrent [Coh03] the user is interested in exactly one download per torrent and will normally not join a distribution overlay in which he already completed downloading.

Churn consists of the *lifetime* of a node which starts with its initial creation and ends with its permanent departure. During its lifetime, a node goes through several online (i.e. participating in the overlay) and offline cycles (see Figure 3.1). The time span a node is online is called a *session*. The time between sessions is accordingly called *intersession*.

Joining and *leaving* are system-specific operations. In most overlays joining includes finding neighbors, initializing routing tables, replicating data, and other setup operations. Leaving cleans up the session state. A regular leave normally includes the notification of neighbors to help them reorganizing their state. Not all nodes leave the network orderly, some simply *crash*, they disappear from the overlay without any notification. This can be caused by software crashes, hardware crashes, or loss of connectivity. A crash is simply a special form of leaving.

A node in a p2p system is both a server and a client. As such it provides *resources* and executes *requests* for available resources. The provision and consumption of resources specifies the *workload* of the system. A realistic workload model is crucial for sensible performance evaluation. For example, resources include shared files in filesharing systems, a request would stand for downloading a file. As previous studies [GDS⁺03] have shown, the workload is not uniform over all resources. Some resources are abundant, many are scarce. The same is true for requests, some resources are far more *popular* than others.

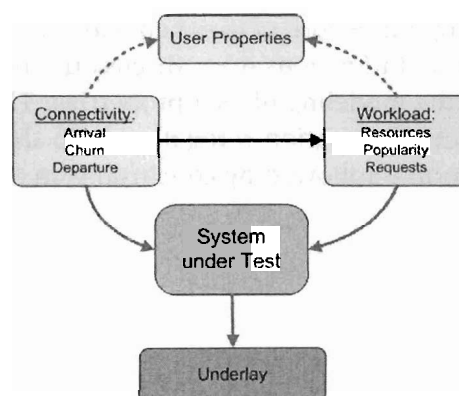


Figure 2.1: System view

Both churn and workload operate directly on the simulated system which itself relies on an underlay model of the Internet in the simulator framework (see Figure 2.1). The *strategy* of a user interacts with

churn and workload. The term strategy here denotes the goals and interests of a user. Similar to the global popularity of resources in the workload, a user has no uniform interest in all resources, but consumes only certain types or even very specific resources. This interest clustering is especially important for reputation systems, but has also been used to build specialized content-clustered overlays [CGM03]. Also part of the strategy is the user's willingness to cooperate. Some users try to maximize their benefit, e.g. by not uploading at all or tricking a reputation system. On the other hand, many users do not leave the network immediately after a completed download [LCC⁺06]. Thus, the user strategy does influence both workload and churn.

The *underlay*, the network infrastructure the overlay operates on, is critical for a reasonable performance evaluation as well. Bandwidth, latency, message loss and other properties of the underlay links have to reflect the conditions of the target real-world environment. However, the modeling of underlays is out of scope for this paper as it is mostly orthogonal to the user behavior.

In the next three sections we discuss the components in more detail.

3 Modeling Churn

One of the most well-studied and analyzed component of user behavior in p2p systems is churn, i.e. the property of peers in the system to leave it and come again at will. Studies such as [SR06b] by Stutzbach et al. measured and analyzed the churn behavior in different p2p systems, focusing on session and intersession times. They found churn characteristics being significantly different across systems.

Our modeling framework captures both online and offline peers by the modeling of lifetime and sessions. Typically, a user participates in the system over a certain time, the *user lifetime* in the system. During this time a user may connect and disconnect from the system many times, therefore creating *online sessions*. Figure 3.1 shows an example lifetime of a peer in the system together with three sessions.

We divide the modeling of churn into *lifetime* and *session* models. The first model describes when a peer appears in the system for the first and the last time. The latter describes when the peer's online sessions during its lifetime in the system.

In a simulation run, there is a number of N peers instantiated and available for simulations. At any point of time a subset of these peers is online resulting in the actual network size. This network size depends on four factors: arrival rate, join rate, leave rate, and departure rate.

Because scalability is an important property of p2p systems, simulations have to show how the system performs for a given network size. Here the number of *online* hosts is relevant. We will see how the network size can be predicted and how long it might take to reach a steady network state.

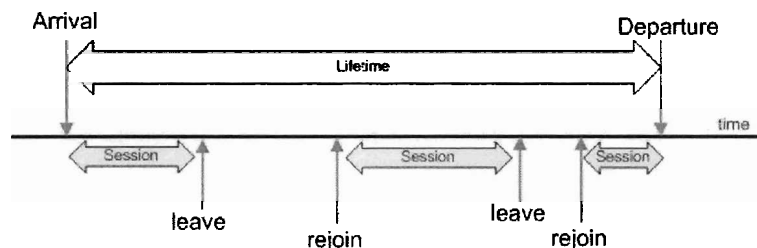


Figure 3.1: Peer's lifetime.

3.1 Lifetime Models

The lifetime of a peer is determined by two events: *arrival* and *departure*. The peers' arrivals in a system can be influenced by many factors, such as the application deployment process or the dynamics of social networks where people start using the system recommended by their friends. These factors are difficult to capture exactly and therefore a stochastic model appears reasonable to describe this behavior.

Similar arguments can apply to peers' departures, because users might stop using an obsolete system and switch to a new one. However, this process is typically of less interest for a simulation scenario.

Moreover, in a real systems the lifetime of a peer in the system is much longer than the session time. The lifetime can be several days or even months long while the sessions are few hours long on average [BSV03]. As the simulated time often ranges in the order of hours only, the rate of departures is so low that they can be ignored.

However, this is different for systems such as BitTorrent where peers are only interested in downloading one single file and depart from the system permanently afterwards. This applies because for each file a distinct short-living overlay is created.

In the following we consider two possibilities of modeling the arrival process:

Deterministic Arrival Process

Here all peers arrive with a constant inter-arrival interval τ . The complete network is built up after the fixed time $N \cdot \tau$.

A possible downside of this approach is the lack of randomness, often observed in real system. Furthermore, the fixed inter-arrival intervals can cause a synchronization of periodic events.

Poisson Arrival Process

If peers are assumed to arrive independently from each other a Poisson arrival process can be applied [Hav98]. Given a peer arrival rate λ , peer arrivals are distributed exponentially with the same rate λ resulting in the mean inter-arrival rate $\frac{1}{\lambda}$. For each arrival event an absent peer is selected and connected to the system. Therefore, for N peers the network will be completely built up after the time $\frac{N}{\lambda}$.

This is a more realistic model because arrival events happen with different intervals.

3.2 Session Models

An online session model describes peers going online and offline after they joined the system for the first time. This model contains two events: *join* and *leave*. The reason why it can be important to distinguish *join* and *arrival* events in a system is that peers coming back to the system can reuse contact information of other peers and the data items stored locally.

The modeling of online sessions can be done either *globally* or *per peer*. In the first case, there is a global rate of join and leave events. Here, when a global join event occurs an offline peer is selected to join the network. The same can apply to offline events. In a simulation environment this method requires only one periodic join and leave event to be scheduled.

The second method schedules one session start/end event per peer. Once a peer has arrived to the system, its session length is computed and the respective *leave event* is scheduled. After this time is reached and the event is fired, the intersession length is computed and the *join event* is scheduled. While this method requires more events to be scheduled, there is a benefit of a high flexibility: the session and intersession durations can follow any desired distribution: exponential, Weibull etc. Furthermore, the peers can be divided into groups, where some peers are long-lasting while others connect and disconnect frequently.

In the following we consider examples for each of both methods.

Exponential Session and Inter-Session Durations

In this model both the session and intersession time of peers follow an exponential distribution with the (possibly different) rates λ_s and λ_i respectively.

The mean session duration of a single peer is $mean_s = \frac{1}{\lambda_s}$ and the mean intersession duration is $mean_i = \frac{1}{\lambda_i}$. Then the expected online probability of a peer being online at a given time is the *connectivity factor* $c = \frac{mean_s}{mean_s + mean_i}$. Therefore, for N available peers only approximately $c \cdot N$ peers will be online for any given time.

This rises an interesting trade-off: If we want to simulate with a large number of *online* peers and have realistic session and intersession times, then we might end up with a much larger number of peers that

must be *available* for the algorithm. This can be a problem if memory is the limiting resource for the simulation run.

Global Leave Rate with Peer Replacement

In order to keep the network size (i.e. number of online peers) constant a *global leave rate* can be applied, as used e.g. in [RGRK04]. Here we can use only one periodic event that dictates when a peer will leave the network. Each time the event is triggered one randomly selected online peer leaves the system.

For the join events no additional event is used. Instead an offline peer is randomly selected and joins the network to replace the peer going offline. Therefore, the network size is constant.

In more detail, when the churn process starts, there is a Poisson process deciding when the next *leave* event should occur. If the leave process is timed by an exponential distribution with the rate of λ_{leave} , then the expected inter-event interval is $1/\lambda_{leave}$ and the corresponding session time is expected to be $t_{mean} = \frac{N_0}{\lambda_{leave}}$ where N_0 is the desired session duration.

A benefit of this model is that the memory consumption can be easily limited, e.g we can keep 10,000 peers online from the set of 11,000 available peers, while still keeping the desired mean session duration. For example, if the desired session duration is 50 minutes than the leave rate is set to $10,000/50 = 200$ events per minute.

4 Workload Model

This model comprises two parts: resource distribution that describes the resources offered and used in the system and the user requests specifying how this resources are requested and consumed.

Since in a p2p application the users are connected to exchange content, the question arises how the information is distributed among the users. The content items are spread in the system according to some distribution and again depending on the application type and the user structure.

The resource distribution defines how many resources are present in the system. The replication distribution specifies how many replicas of each file are present in the system. Finally, the popularity specifies how often single resources are requested by users.

Several works consider the specifics of p2p file sharing systems regarding the user request patterns, e.g. [GDS⁺03] and [SGD⁺02]. One interesting result is that systems where users consume multimedia content (so most file sharing systems, too) show a download-once behavior, e.g. the user almost never request the same content twice. This leads to a different popularity distribution compared to classical client-server systems, such as web servers.

The actual requests to the system are issued depending on offered resources, resource popularity and user's interests. In a simple model a user issues requests on a regular basis, e.g. each m time units. In a more realistic approach the users issue queries following a distribution, e.g. an exponential distribution.

For systems dealing with content (file sharing, video streaming) the skewed popularity of content can be expressed by heavy-tailed distributions of content replicas and queries. In real p2p systems Zipf or Zipf-Mandelbrot were found to fit the content and query distributions well [GDS⁺03] .

5 User Properties Model

This part captures user specific properties, e.g. interests, strategy. The interests define which resources the user is interested in, this can be captured by a resource category the user is interested in and results in different request rates and targets per peer. Precisely, for user u it defines the interesting content $R_u \subseteq R$ as a subset of all available resources R .

The strategy typically describes the cooperativeness of the peer, as some users are willing to contribute more resources than others. For this reason, *altruistic users* will stay online longer and offer more resources, e.g. files, to the network, while other *strategic users* will try to minimize their contribution. A model can further include *malicious users* who try to break the systems specification.

User properties are very specific to the used application, which we cannot discuss them here in a general way. Nonetheless, in most cases user properties simply influence other aspects of the user's behavior, e.g. online time, resources, and requests. Thus, it is generally a good idea to model user properties on top of the churn and workload models.

6 Use Case: Kademlia

In this subsection we analyze how to model realistic user behavior for distributed hash tables in order to evaluate the performance of Kademlia overlay routing protocol.

As a generic DHT Kademlia has many use-cases. We consider a file-sharing application in which the DHT is used to locate files offered by peers using a file ID. That results in rare lookup and store requests because users only use the DHT while searching for file sources but not within the actual file transfer. For example, in eMule KAD [SR06a], a variant of Kademlia protocol, is used.

In the BitTorrent [Coh03] derivative Azureus¹, Kademlia is used as a decentralized alternative to a centralized tracker. Here the peers can obtain the addresses of other peers downloading the same file and publish their own addresses.

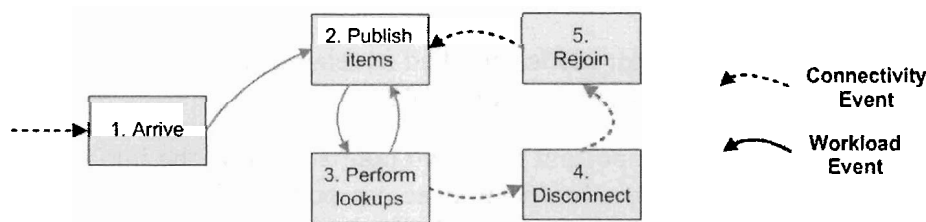


Figure 6.1: User activities in a DHT. The events that cause the change the current activities are differentiated by the responsible component.

The general activity diagram of a Kademlia user is shown in Figure 6.1. At peer's arrival the Kademlia join procedure takes place. In the *perform Lookups* phase, lookups and store operations are executed. Here especially the lookup rate and distribution are timed by the applied workload.

The arrival, leave and join events are timed by the churn component. If a peer goes offline then she cancels all running queries. Upon a peers return to the system the workload component takes over the control of the peer and issues new lookups. It further specifies the objects (representing any kind of

¹ www.azureus.com

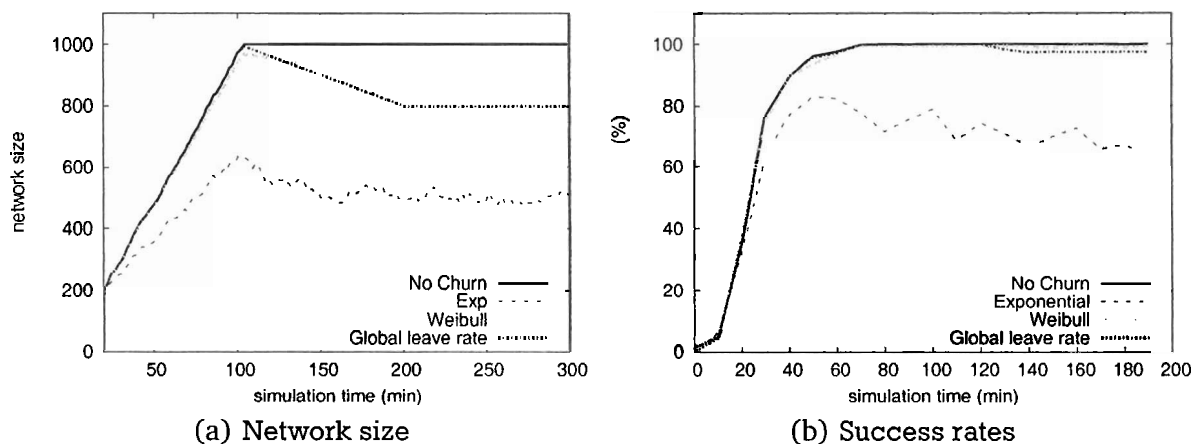


Figure 6.2: Impact of churn model on performance.

Table 6.1: Kademlia modeling parameters

Component	Model
resources model	Zipf distributed popularity
request model	Zipf distributed popularity
lifetime model	constant arrival intervals exponential arrival intervals
session model	exponentially distributed sessions and intersessions Weibull distributed sessions/intersessions based Global (Poisson) leave rate with peer replacement no churn (for comparison)
departure rate	no permanent departure
user type model	simple user model

resources to be published and looked up) available in the system together with their popularity and replication.

We analyzed different user behavior models applied in related work and identified the following most useful characteristics for modeling user behavior in DHTs:

- Peer arrival rate: this one is important from two points of view: the initial creation of the network and the avoidance of event synchronization across peers. The latter might happen because of periodic maintenance operations running in the background.
- Join and leave events: Performance of a DHT depends on the quality of the routing tables. Stale contacts in the table might cause expensive lookup timeouts and even the lookup success rate can degrade.
- Lookup/store activity: Bulk requests can overload peers responsible for popular content. Furthermore, Kademlia tries to minimize the maintenance overhead, but if there are no user requests over certain time intervals, maintenance overhead will increase.
- Resources distribution, replication and popularity: Different popularity of content will distribute the load in the system unevenly. Also, resources with a low replication factor might become unavailable under churn and result in failed lookups.

Table 7.1: Kademlia setup

Parameter	Value	Description
ID-Length	80	length of Kademlia id space
b	2	order of the routing tree
k	10	number of contacts per bucket
refresh-interval	1 hour	routing table refresh interval
α	3	number of concurrent messages
republish-interval	1 hour	how often the items are republished

7 Evaluation

In this section we analyze the impact of user behavior on performance of Kademlia protocol. Table 6.1 lists the different components of user behavior under consideration. We focus on the impact of churn being the main issue for the performance of search overlays. In detail, we compare the impact of varying churn setups: different churn models as introduced in Section 3 and the impact of single parameters: network size, session and intersession durations.

7.1 Methodology

Our simulation platform for the experiments is the discrete event-based simulator PeerfactSim.KOM [?]. The simulator offers a generic framework for different overlays and network models. We use the implementation of the Kademlia routing protocol with the setup parameters as shown in Table 7.1. These are standard Kademlia parameters, except the ID that was set to a smaller size due the known maximum amount of available peers.

In order to estimate the impact of user behavior on overlay routing the following metrics are used:

- **Routing table quality** measured as the ratio of fresh contacts, i.e. contacts that are online at the given time. This metric is measured globally for all peers in periodic intervals of 5 minutes.
- **Success rate** determines the ratio of lookups being able to find the desired objects. The success rate is aggregated over intervals of 5 minutes. A lookup can fail either if the object is not available at online peers or the peers holding the object are not found by the routing protocol due to the inconsistency of routing tables.
- **Dropped messages per peer** reflects the impact of stale (offline) contacts in the routing tables. Overlay messages in Kademlia are sent via UDP and, hence, get dropped if the destination peer is offline.
- **Received messages per peer** reflects the load of single peers in the overlay.

7.2 Setup 1: Fixed Network Size, Variable Session Models

At first we evaluate the impact of the session model used on the system performance. The following models are considered:

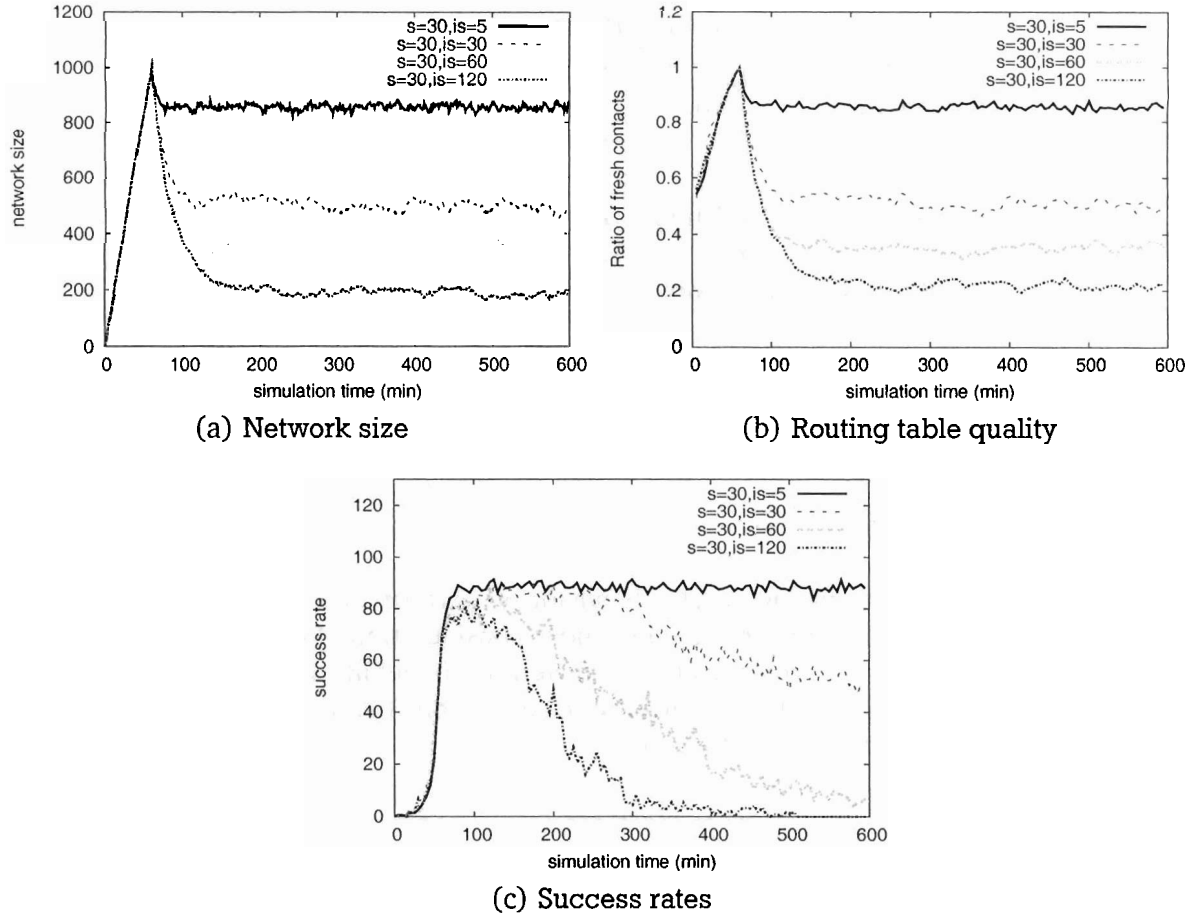


Figure 7.1: Impact of varying intersession durations ($is = \{5, 30, 60, 120\}$ minutes) with the fixed session duration ($s = 30$ minutes).

- Exponentially distributed session and intersession durations (see Section 3.2). Both the session and intersession rates are set to 0.05 events per minute (accordingly the means are 20 minutes each).
- Weibull distributed session and intersession durations (following the measurements of Steiner et al. in [SEN07]). The Weibull distribution parameters scale and shape are set to 169.5385 and 0.61511 for session durations. The parameters for the intersession durations are set to 413.6765 and 0.47648 respectively.
- Global leave rate with peer replacement (see Section 3.2). Here the rate of leave events is set to 10 events per minute and the target network size to 800 peers.
- No churn for comparison. Here all peers are online once they arrived in the system.

The network sizes of each model during the simulation run are shown in Figure 6.2(a). As we can see here, depending on the distribution size the actual number of online peers is different. Most of them are similar, ranging between 800 and 1000 peers once the network is built up. Only the exponential churn model has different network sizes. For the exponential model with short session durations (both session and intersession means are set to 20 minutes) the network size oscillates around the expected mean of $\frac{N \cdot \text{mean}_s}{\text{mean}_s + \text{mean}_i} = \frac{1000 \cdot 20}{20 + 20} = 500$ peers.

For the model with a global leave rate the network size reaches the target size of 800 peers after 200 minutes. Here 100 minutes are required for all 1000 peers to arrive to the system and 100 more minutes

to reach the desired network size of 800 peers. Later the network size stays unchanged, because each leaving peer is immediately replaced by an offline peer.

Figure 6.2(b) presents the impact of the churn model used on the success rate. For most of the models no significant difference is visible except the exponential session model, where the success rate degrades dramatically. The explanation is that because of the high churn rate, some objects become unavailable and therefore the lookups for these objects fail. We conclude that the impact of the network size is more relevant than the actual model being used.

7.3 Setup 2: Fixed Session Duration, Variable Intersession Duration

In order to evaluate the impact of intersession duration times on the overlay performance we fix the model to the exponential session model and vary the intersession durations. Four different values are used: 5, 30, 60, and 120 minutes. Figure 7.1(a) shows the network sizes for each of them. We can see that after the arrival process is finished the system reaches the expected size and the curve oscillates slightly. The quality of routing tables (shown in Figure 7.1(b)) reflects the network size. For example, the network size of 500 peers out of 1000 available (curve $s=30$, $i_s=30$) results in the fresh contact ratio of roughly 50%.

As next we analyze the success rate plotted in Figure 7.1(c). Interestingly, the success rate changes very dramatically if intersession intervals increase. For example, the change of the intersession interval from 5 to 30 minutes results in a dramatic fall of the success rate. Another observation is that for large intersession lengths has negative impact on the success rate over time.

7.4 Setup 3: Fixed Network Size, Variable Event Rate

Furthermore, we analyzed the impact of varying online and offline event rates for the constant connectivity factor. Because the equation for connectivity $c = \frac{mean_s}{mean_s + mean_{is}}$ applies we obtain $mean_s = \frac{mean_{is}}{1-c}$. We fix the connectivity factor to 0.5 and therefore obtain equal durations for the session and intersession intervals.

The experiments conducted have expected interval durations of 10, 30, 60, and 120 minutes for both sessions and intersessions. Hence, the expected network size is $0.5 \cdot 1000 = 500$ peers.

As shown in Figure 7.2(a) the network sizes are very similar. Similarly, the quality of routing tables is around 50%. (see Figure 7.2(b)). However, the success rate is very differently as shown in Figure 7.2(c). It drops significantly if the intersession durations are cut to 5 minutes. Here the the global leave rate is $500 \cdot \frac{1}{5} = 100$ events per minute reduces the success rate from 80 to 60%.

Additionally, Figures 7.2(d) and 7.2(e) present the distributions of received and dropped messages in the network per peer for two setups. Again we observe that both metrics suffer under high churn rates.

The direct implication of these results is that the replication rate has to be adjusted to the churn rate in order to assure successful lookups. In a real system where, the churn cannot be easily known in advance the replication mechanism has to be adaptive. For example, in Kademlia an adaptive bucket size k can result in a high recall rate without too much overhead.

7.5 Setup 4: Variable Arrival Processes

We further analyzed the impact of different peer arrival rates on Kademlia but found them having a negligible effect on the system performance. The alternatives compared were the deterministic and the stochastic processes as described in Section 3.1 and 3.1 respectively.

Even if during the arrival process a part of objects is unavailable, once their holders arrive, the routing tables get repaired quickly. Hence, in the steady state we measured almost the same success rates despite

the arrival process applied. Especially, we found no difference between a Poisson process and peer arrivals with the constant inter-arrival interval. However, this effect, even if not relevant for Kademlia, is expected to have higher impact on systems such as BitTorrent [Coh03].

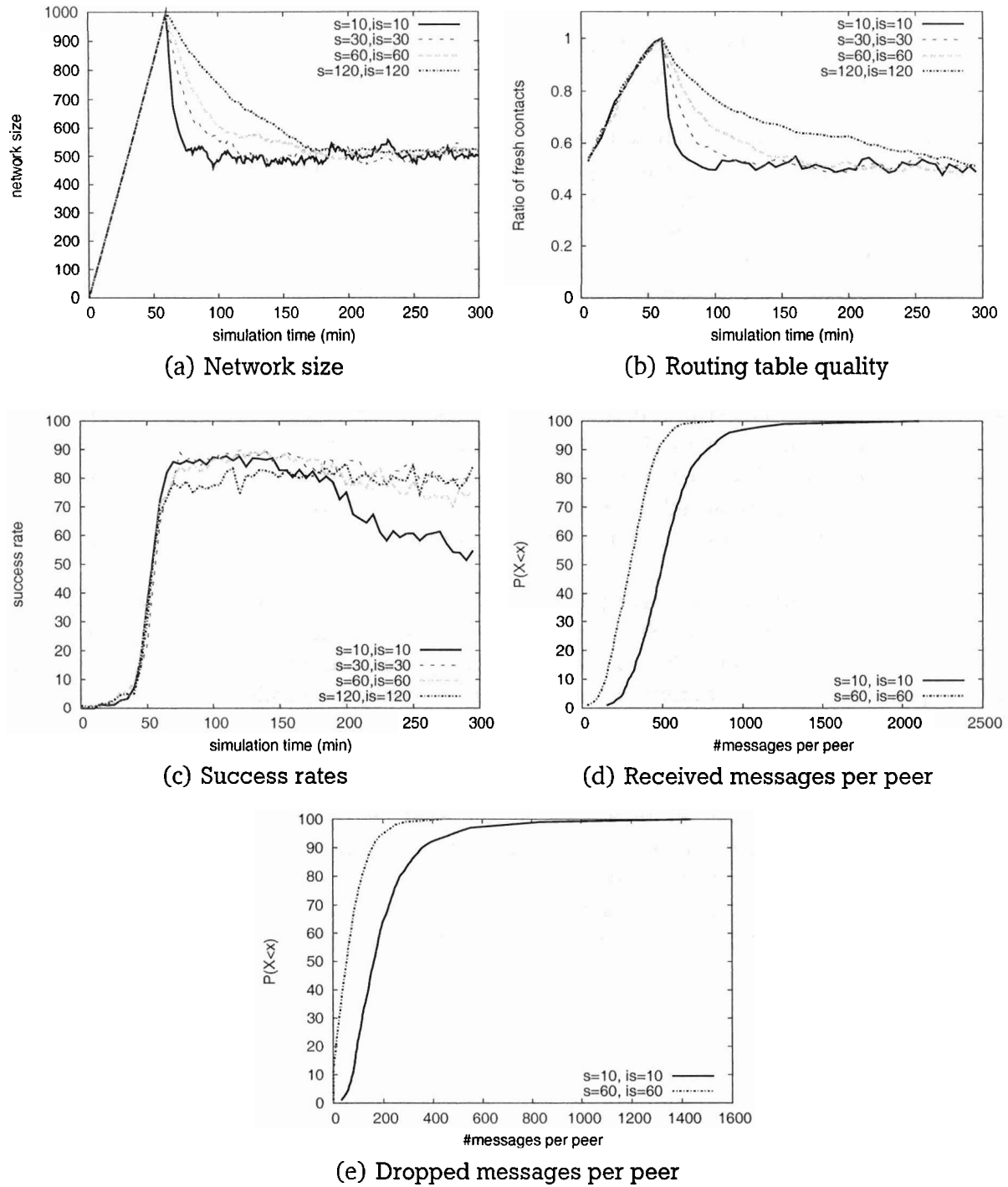


Figure 7.2: Impact of varying intersession (is) and session duration (s) with the fixed network size of 500 peers.

8 Related Work

There is substantial related work on modeling of user behavior for single p2p overlays, such as Gnutella and DHTs. We studied them in order to identify the most important properties of user behavior being worth to be represented in our model.

Herrera et al. propose a group-based model [HZ07] to model churn in a p2p system, where peers are divided into three groups, benefactors, peers and peepers, showing different characteristics regarding their session and intersession lengths. The modeling is based on the *connectivity factor*, that specifies the fraction of all available peers being online at any given point of time. This value is probabilistic, because the session and intersession times of peers are probabilistic and therefore fluctuate around the desired network size defined by the connectivity factor. This connectivity factor is computed by $C_f = \frac{OS}{OS+IS}$ where OS is the average session length and IS is the average intersession length.

The performance of different Distributed Hash Tables (DHTs) under churn is analyzed by Rhea et al. under churn [RGRK04]. Here additionally to the churn the arrival process, i.e. with which rate and distribution the peers connect to the system is considered. This model is suitable especially for search overlays and we use it for comparison. The interesting property of the churn model applied, is that it allows a stable size of the network, i.e. the number of peers being online is constant during the simulation time. This is achieved by replacing each peer going offline by another peer connecting to the network immediately. This is different to pure probabilistic models and allows more control of the simulation parameters.

A work on DHTs with mobile participants [ZDK08] distinguishes churn behavior for heterogeneous networks, including mobile peers. Such peers might use mobile phones for Internet access and therefore show much higher churn rates due to expensive online access and failure rate. The modeling here is based on peer classes with different mean online times, failure probabilities, number of shared objects, average query rate.

Specific models can be applied for applications such as Gnutella, where the query circle model was introduced by Schlosser et al. [SCK03]. They focus mostly on the query inter-arrival times and content popularity. Another work on Gnutella from Aggarwal et al. [AAF08] concentrates on the impact of realistic and extreme user behavior types on the system. In their model, content replication, session length and query strings are considered.

Andreolini et al. characterize the resources distributed in Gnutella network [ALY04], especially the file types (video, audio, archives), their sizes and popularity distributions.

9 Conclusions

In this paper, we have studied the modeling of user behavior when simulating p2p systems. We have presented the framework used in our simulator to model user behavior in p2p systems. As an example, we showed how it can be applied to Distributed Hash Tables.

We studied and demonstrated the impact of different modeling approaches using Kademlia as use case. In particular, arrival rates and churn have significant impact on the system performance resulting in diverging latencies and success rates. This again emphasizes the importance of realistic user loads for evaluation of p2p overlays. On the other hand, given a fixed arrival rate the actual distribution of the arrival events did not show a significant impact on the performance of the evaluated system. We believe that this effect is common for many DHTs as long as the processing power of the peers is high enough to handle events with high rates. This might be, however, different for systems built on mobile nodes with low connectivity and low processing power.

Our future work will concentrate on the impact of user behavior modeling on BitTorrent-like content distribution overlays. There, we expect a higher impact of the arrival rate and some specific modeling parameters, e.g. the seeding time after a peer finishes the file download.

List of Figures

2.1	System view	5
3.1	Peer's lifetime.	7
6.1	User activities in a DHT. The events that cause the change the current activities are differentiated by the responsible component.	12
6.2	Impact of churn model on performance.	12
7.1	Impact of varying intersession durations ($is = \{5, 30, 60, 120\}$ minutes) with the fixed session duration ($s = 30$ minutes).	15
7.2	Impact of varying intersession (is) and session duration (s) with the fixed network size of 500 peers.	18

Bibliography

- [AAF08] Vinay Aggarwal, Obi Akonjang, and Anja Feldmann. Improving user and isp experience through isp-aided p2p locality. In *Proceedings of 11th IEEE Global Internet Symposium 2008 (GI'08)*, Washington, DC, USA, April 2008. IEEE Computer Society.
- [ALY04] M. Andreolini, R. Lancellotti, and PS Yu. Analysis of peer-to-peer systems: workload characterization and effects on traffic cacheability. In *Modeling, Analysis, and Simulation of Computer and Telecommunications Systems, 2004.(MASCOTS 2004)*, pages 95–104, 2004.
- [BSV03] Bhagwan, Savage, and Voelker. Understanding availability. In *International Workshop on Peer-to-Peer Systems (IPTPS)*, LNCS, volume 2, 2003.
- [CGM03] Arturo Crespo and Hector Garcia-Molina. Semantic overlay networks for P2P systems. Technical report, Stanford University, 2003.
- [Coh03] B. Cohen. Incentives build robustness in bittorrent. In *Proceedings of the Workshop on Economics of Peer-to-Peer Systems*, Berkeley, CA, USA, 2003.
- [GDS⁺03] Krishna P Gummadi, Richard J. Dunn, Stefan Saroiu, Steven D. Gribble, Henry M. Levy, and John Zahorjan. Measurement, modeling, and analysis of a peer-to-peer file-sharing workload. In *SOSP '03: Proceedings of the nineteenth ACM symposium on Operating systems principles*, pages 314–329, New York, NY, USA, 2003. ACM.
- [Hav98] B.R. Haverkort. *Performance of Computer Communication Systems: A Model-Based Approach*. John Wiley & Sons, Inc. New York, NY, USA, 1998.
- [HZ07] Octavio Herrera and Taieb Znati. Modeling churn in P2P networks. In *Annual Simulation Symposium*, pages 33–40. IEEE Computer Society, 2007.
- [LCC⁺06] Uichin Lee, Min Choi, Junghoo Cho, M. Y. Sanadidi, and Mario Gerla. Understanding pollution dynamics in p2p file sharing. In *5th International Workshop on Peer-toPeer Systems (IPTPS'06)*, 2006.
- [MM02] Petar Maymounkov and David Mazières. Kademlia: A peer-to-peeiptr information system based on the XOR metric. In *International Workshop on Peer-to-Peer Systems, (IPTPS)*, 2002.
- [RGRK04] Sean C. Rhea, Dennis Geels, Timothy Roscoe, and John Kubiatowicz. Handling churn in a DHT (awarded best paper!). In *USENIX Annual Technical Conference, General Track*, pages 127–140. USENIX, 2004.
- [SCK03] M.T. Schlosser, T.E. Condie, and S.D. Kamvar. Simulating a file-sharing p2p network. In *Workshop on Semantics in Peer-to-Peer and Grid Computing*, 2003.
- [SENB07] M. Steiner, T. En-Najjary, and E.W. Biersack. A global view of kad. In *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, pages 117–122. ACM New York, NY, USA, 2007.
- [SGD⁺02] Stefan Saroiu, P. Krishna Gummadi, Richard J. Dunn, Steven D. Gribble, and Henry M. Levy. An analysis of internet content delivery systems. In *OSDI*, 2002.
-

-
- [SR06a] D. Stutzbach and R. Rejaie. Improving lookup performance over a widely-deployed dht. In *Infocom*, volume 6, 2006.
- [SR06b] D. Stutzbach and R. Rejaie. Understanding churn in peer-to-peer networks. In *Proceedings of the 6th ACM SIGCOMM on Internet measurement*, pages 189–202. ACM Press New York, NY, USA, 2006.
- [ZDK08] Stefan Zöls, Zoran Despotovic, and Wolfgang Kellerer. On hierarchical DHT systems - an analytical approach for optimal designs. *Computer Communications*, 31(3):576–590, 2008.