KKM+07] *Aleksandra Kovacevic, Sebastian Kaune, Patrick Mukherjee, Nicolas Liebau, Ralf Steinmetz*; ***Benchmarking Platform for Peer-to-Peer Systems***; it - Information Technology (Methods and Applications of Informatics and Information Technology), Septembc; 2007, S. 312-319

Benchmarking Platform for Peer-to-Peer Systems

Benchmarking Plattform für Peer-to-Peer Systeme

Aleksandra Kovačević, Sebastian Kaune, Nicolas Liebau, Ralf Steinmetz, KOM – Multimedia Communications Lab, Darmstadt
Patrick Mukherjee, ES – Real-Time Systems Lab, Darmstadt

Summary The benefits of the peer-to-peer paradigm have been proven through various applications besides file sharing. The requirements for the design of peer-to-peer overlay networks vary according to its purpose. In order to compare existing overlay networks and determine their suitability for specific purposes, requirements are defined with abstract quality attributes. Once the benchmarking set (quality attribute, metrics, and scenarios) is identified, experiments should be applied under the same circumstances on each overlay in order to obtain comparable results. This paper presents Peerfact-Sim.KOM, a simulator providing a benchmarking platform for peer-to-peer systems, especially for overlay networks. It supports defined benchmarking sets for all kinds of peer-to-peer overlays through an implemented catalogue of metrics and a simple but comprehensive scenario specification. Various peer distributions and churn rates are given which also supports geographical-location dependence. The platform is extensible due to its modular design and can scale up to around 10^6 peers for simple overlays such as Gnutella and 10^5 for more complex overlays like Kademlia. ►►► **Zusammenfassung** In vielen Anwendungen jenseits der Dateitauschbörsen zeigen sich die Vorteile des Peer-to-Peer Kommunikationsparadigmas. Abhängig vom Zweck variieren die Anforderungen an das Design eines geeigneten Peer-to-Peer-Overlay-Netzes. Um existierende Overlay-Netze vergleichen zu können und um zu entscheiden, ob sie für einen festgelegten Zweck geeignet sind, werden die Anforderungen mit abstrakten Qualitätsattributen beschrieben. Hat man einen Benchmarking-Satz (Qualitätsattribute, Metriken und Szenarios) identifiziert, sollte dieser, um vergleichbare Resultate zu erzielen, bei der Messung eines jeden Overlay-Netzes eingesetzt werden. Der Schwerpunkt dieser Arbeit ist Peerfact.KOM, ein Simulator, der eine Benchmarking-Plattform für Peer-to-Peer-Overlay-Netze bereitstellt. Für alle verschiedenen Arten von Peer-to-Peer-Overlay-Netzen bietet er festgelegte Benchmarking-Sätze, einen implementierten Katalog von Metriken und eine einfache, aber ausdrucksstarke Methode, um Szenarien zu spezifizieren. Verschiedene Churn-Raten und Peer-Verteilungen, die optional auch auf geographische Ortsangaben bezogen sind, werden geboten. Der Simulator ist durch sein modulares Design erweiterbar. Er skaliert bis zu 10^6 Peers in einfachen Overlay-Netzen wie Gnutella und 10^5 in komplexeren Overlay-Netzen wie Kademlia.

KEYWORDS C.2 [Computer Systems Organization: Computer-Communication Networks: Distributed Systems] benchmarking, evaluation, metrics, peer-to-peer, simulator

1 Introduction

The benefits of the peer-to-peer (p2p) paradigm have been proven in various applications besides file sharing. Depending on its purpose, the requirements for the design of appropriate p2p overlay networks vary widely in terms of user behavior, dynamics of the network, acceptable communication delay, or resources of the peers. Current p2p research offers different types of overlays, which are classified according to the relationship between the formed overlay topology and the stored content (structured and unstructured), grade of centralization (pure decentralized, partly centralized, or hybrid), and structure of the overlay (flat and hierarchical). Today, a complete understanding of their advantages and disadvantages as well as interdependencies is still lacking. This is necessary to find a suitable overlay for a specific purpose. For example, structured overlays are more scalable and retrievable than the unstructured, however unstructured overlays provide better support for highly transient peer populations. Superpeers in partially decentralized overlays facilitate overlay services and decrease protocol overhead. However, such an overlay is based on the assumption that some peers are able to take on more important roles and handle higher loads.

In order to compare overlay networks and determine their suitability for a given purpose, it is necessary to define the requirements through abstract *quality attributes.* Four groups of quality attributes are defined in [1], namely: adaptability, efficiency, validity, and trust. In order to quantify each quality attribute, appropriate *metrics* are needed. Furthermore, in order to systematically compare overlay networks, standardized *scenarios* are needed (user behavior, peer resources, churn, etc.). These three aspects identified make up a benchmarking set.

Evaluation methods for p2p overlay networks can be analytical, include testing prototypes in testbeds, or involve simulation. As p2p systems are very complex, an analytical approach requires far too many simplifications. Running large scale experiments in a testbed with prototypes is difficult due to a lack of sufficiently large testbeds. Only PlanetLab [2] is a possible alternative as a testbed with about 766 nodes (March 2007). However, it is still not sufficiently large [3] to provide a precise snapshot of a p2p system with its millions of participants. The approximations which simulations provide are much closer to reality than an analytical approach, and it is possible to simulate networks of hundred thousands of peers.

In this paper we will present PeerfactSim.KOM, a simulator which provides a benchmarking platform for p2p overlay networks. It supports the predefined benchmarking sets for all kinds of p2p overlays by an implemented catalogue of metrics and a simple but comprehensive scenario specification. The simulator is designed modularly, extensible, and scales for around 10^6 participants for simple overlays such as Gnutella and 10^5 for more complex overlays like Kademlia.

The paper is organized as follows. In Section 2, the requirements for benchmarking platform are identified. PeerfactSim.KOM is presented in Section 3 with its performance in Section 4. Other relevant existing simulators for p2p overlay networks are analyzed in Section 5. Finally, the paper concludes with Section 6.

2 Requirements for a Benchmarking Platform

Here we define the requirements necessary for a complete simulation framework which enables benchmarking of p2p systems:

Modularity. Each functional part of the simulator should be modularly designed to easily support exchange with different implementations. The distinction among the functional modules should be clear and the decomposition should identify all independent modules that could be exchanged in the future. For example, if a user wants to change the simulation engine from being event-driven to time-driven, this can be done without changing any part of the simulator. If the underlying network model is unnecessarily complex for a particular simulation, it can be changed as well. Therefore, the overlay model must be changeable, independent of other modules in order to support the simulation of different p2p protocols within the same environment.

Underlay network model. This requirement highly influences scalability of a simulator. All the impacts of an underlaying network, such as packet-loss, propagation delay, congestion, etc. on p2p overlays have to be identified and modeled.

User behavior. In a simulation scenario the following have to be defined: exactly when all relevant peer operations occur, whether on individual or group basis, standard churn rates, and location-dependent peer distribution. In order to measure the impact of particular aspects (overlay parameters, churn rate, etc.), it is necessary to support the repetition of simulation with the same events happening at the same time.

Resource model. Peer-to-peer networks are characterized by the heterogeneity of participants, especially regarding their resources such CPU power, memory, and bandwidth. The effect that heterogeneity has on the performance of p2p overlays is significant and should be captured.

Service model. The application areas of p2p networks vary greatly and are continuously expanding. Each application has its own requirements and the performance of the protocol is different for each application scenario. A p2p generic simulator must include an extendable model of the services which p2p networks offer, apart from common ones such as lookup and routing. The most used p2p service currently, like file-sharing, should be offered within the basic framework.

Easy experiment setup. A user should be able to easily define measurements with an extensive set of standard metrics relating to p2p overlays. The code of a simulator should not have to be modified if standard metrics like the number of hops, hit-rate, or response time are used. It should be possible to implement new metrics without changing parts of the code. The output obtained has to be clear with minimal preparation of results required to plot the graphs.

Scalability. As p2p systems are large in scale, scalability of a p2p simulator is a crucial requirement. Therefore, the trade-off between the scalability of a simulator and providing a realistic simulated environment plays an important role when p2p overlays are simulated.

Documentation. The same as any other software, a p2p simulator has to provide user, architecture, developer, and clear source code documentations.

3 PeerfactSim.KOM

The primary goal of PeerfactSim.KOM is to provide a general benchmarking platform for p2p systems, related to the tuple of quality attributes, metrics, and scenarios. It provides a catalogue of defined benchmarking sets (Section 3.5) for various quality attributes with the appropriate user behavior models and output statistics. Various peer distributions and churn rates are provided which also supports geographical-location dependence (Sections 3.4.1 and 4.3.2). The underlying network model takes into account geographical distances between peers, the processing delay of intermediate systems, signal propagation, congestions, retransmission, and packet loss (Section 3.1). The resource model includes a peer's bandwidth model and supports for message priorities with appropriate scheduling mechanisms.

PeerfactSim.KOM is a discrete-event based simulator, written in Java, modularly designed. It consists of five layers (Fig. 1). They are identified as the key components of the widely deployed p2p systems based on an analysis of their functionality

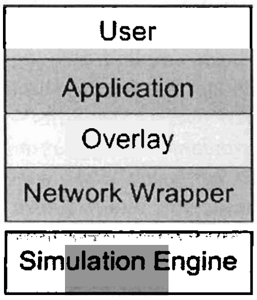


Figure 1 Functionality layers of the simulaton.

and supported services. Each layer encapsulates important aspects so as to model best a p2p system in its entirety. In the following, we will give a brief overview of each layer identified. Detailed documentation is provided in [4].

3.1 Network Wrapper

In order to reduce computation complexity and address scalability problems of common p2p simulation frameworks, there is evidently the need for abstraction models and different levels of details. As the focus of p2p system simulations is on the layers above the transport layer, we have developed a simple latency model to simulate message delivery times. This model takes into account the details of the four underlying OSI layers from end-to-end connections between peers by incorporating important network characteristics, like geographical distance between peers, the processing delay through intermediate systems, signal propagation, congestions, retransmission, and packet loss. Our latency model has been proven valid in [5] and is described in the formula

$$Latency = f \cdot \left[df + \frac{dist}{v} \right]$$

where *dist* describes the geographical distance between the start and end point of the transmission, *df* represents the delay in processing of the intermediate systems, v stands for the speed of the signal propagation through the transmission medium, and f is a variable part which encapsulates the retransmission, congestion, packet loss, etc.

Hence, the quotient $\frac{dist}{v}$ results in the propagation delay of a signal. Therefore, a message with a larger distance between start and end point also has a larger latency. The factor *df* represents the algorithmic processing delay through different intermediate systems whereas f dilates and compresses the total latency, depending on the current conditions of the network. It models network characteristics like retransmission (package loss, package damage or out-of-order packages) and congestion, depending on current network conditions.

3.2 Overlay

Due to wide variety and large design space of overlay networks, overlay layers play an important role in enabling the simulation of complex approaches, advanced concepts such as clustering mechanisms and hybrid architectures. It encapsulates details of overlay communication protocols and specific overlay related algorithms and operations (such as message routing or the maintenance of the network structure). Furthermore, in order to support the heterogeneity of participating peers, PeerfactSim.KOM assigns each peer a different role depending on their responsibilities within the network: router, maintainer, indexer, and cacher [6]. At the moment, it offers the full implementations of Chord [7], globase.KOM [8], CAN [9], Kademlia [10], and Omicron [11].

3.3 Application

This layer enables us to model p2p applications for content distribution, communication, and collaboration. This layer is separated from the overlay layer to allow for experiments with different applications on top of the same overlay. Furthermore, the application layer has to be separated from the user layer as user behavior obviously influences the performance of the entire system. Therefore it is important that simulations are supported by the same application and different user behavior models. Up-to-date, only a simple file sharing application has been implemented.

3.4 User

As previously described in Section 2, the modeling of user behavior is crucial in order to simulate p2p systems as realistic as possible. For example, it is necessary to model the dynamic participation of peers (churn) within an overlay network as this strongly affects the performance or stability of the overlay network. Besides this, peers may become free-riders or be willing to share their resources in order to help overloaded peers. All these characteristics directly influence the overall performance of the system and necessitates the introduction of a user layer which is able to capture the behavior of a specific user during a simulation scenario. In the following, we present several important functionalities which PeerfactSim.KOM supports such as generating peers based on a gray colored world map, the selection of different churn-rates, describing user behavior and simulation setup within an XML document.

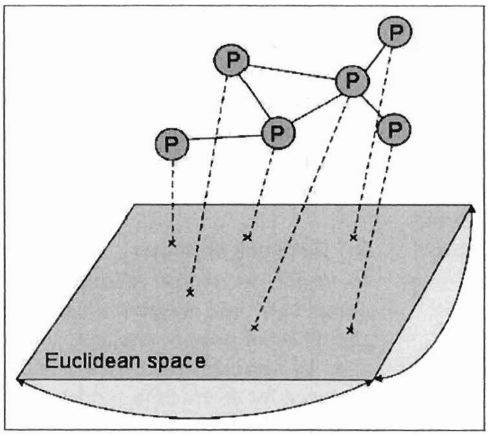
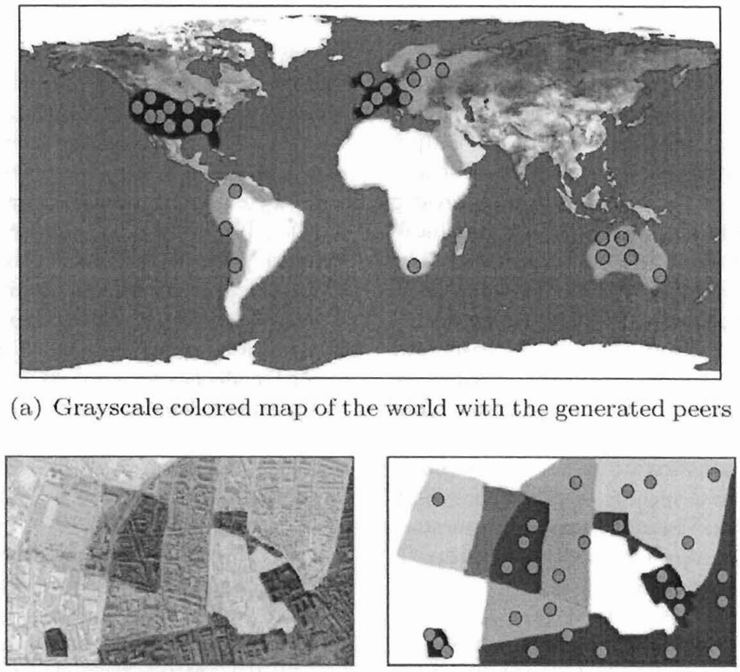


Figure 2 Mapping peers in an Euclidean space.

Peer distribution In our simulator, the virtual space where the peers are located is represented by an Euclidean plane. Each peer generated obtains a unique two-dimensional coordinate (see Fig. 2) according to its position. The distribution of peers on this plane significantly affects simulation.

Two variants for modeling a peer distribution have been realized, uniform random and bitmap-based random.

Uniform random. According to this distribution, peers in the network are distributed uniform-randomly on an Euclidean plane. The advantage of such distribution is its simplicity. However, in order to capturing the effects of distribution on the overlay, this model is not able to give the realistic picture.



(a) Grayscale colored map of the world with the generated peers

(b) Grayscale colored map of the center of Darmstadt

(c) Resulting grayscale-bitmap

Figure 3 Geographical maps colored with grayscale reflecting the concentration of users – the darker parts represent areas with more users relative to lighter parts on the map.

Bitmap-based random. The peers' distribution is not uniform across the world as peers form clusters concentrated at certain parts of the world whilst other vast areas parts are deserted. In order to simulate such non-uniform distribution, peers can be randomly distributed based on a grayscale colored bitmap. This bitmap can be a map of the world or a map of a smaller area, like the map of a city. Sparsely populated areas are represented by a lighter gray and darker areas represent the denser areas. Therefore, the darker an area is on the bitmap, the higher the probability that there a peer is mapped at this location (see Fig. 3).

Churn-rates An important characteristic of p2p networks that has to be taken into account during a simulation is the so-called *Churn* – the user-controlled participation dynamics in a network [12]. Currently two churn generators have been implemented in order to model the joining and leaving of p2p overlay networks – *the uniform-random churn generator* and *the mixed log-normal churn generator*. The uniform random churn model is a simple but unrealistic model which hides important effects of churn on the overlay. Therefore, PeerfactSim.KOM provides an additional churn model based on the mixed log-normal distribution. This distribution best fits the data from [13] and parameters are taken from [11].

3.5 Benchmarking Sets

PeerfactSim.KOM provides several benchmarking sets, defined as a tuple of quality attributes, appropriate metrics and scenarios:

Quality attributes. The nature of peer-to-peer networks introduces some common quality attributes each p2p application has to consider. Examples are *stability*, *scalability*, and *load balancing*, since the peers are autonomous entities and randomly leave, join, or perform overlay operations at any time. Furthermore, peers have variable connectivity and failures can appear at a random point in time, which introduces *robustness* as another important quality attributes. We considered these aspects for our benchmarking sets, with the following definitions (based on [1]): *Efficiency* is defined as the ratio of performance (performance of overlay operations and service provisioning) and costs (from the view of individual node, the whole p2p overlay, IP infrastructure support). *Scalability* is adaptability of the overlay to a changing number of participants or services in the overlay. *Stability* describes the ability of a peer-to-peer overlay to maintain the optimal performance when the frequency of overlay activities (joining or leaving the network, lookup/search) change. *Robustness* is defined here as the time an overlay needs to 'recover' from sudden failures. *Failures* stands for simple absence of peers from the network whereas a peer that *leaves* the network first notifies particular peers (according to a protocol). *Load balancing* represents the distribution of the traffic load on the individual peers. Sometimes peers can have a more significant role or offer more popular service and therefore bear more of the load.

Scenarios. The scenario defines what overlay operations of a certain peer or group of peers will be performed and at what point in time. The scenarios considered in our benchmarking sets are the following:

- *'Ideal'* stands for the scenario where peers first join the network and, when the assumed bootstrapping process is over, the peers start to perform the overlay operations. A new overlay operation will not take place before the appropriate stabilization phase is over, and churn is not expected.
- *'Scaling'* is a scenario where the size of the network is changing. All participants are divided into groups. After a group has joined, all peers publish their data and start random *get(key)* operations. Then the next group joins and repeats the same steps. This process repeats until all groups have joined and finished the described steps.
- In the *'unstable'* scenario, a significant number of peers leave in a short time interval. In another variant, these peers perform a large number of overlay operations in the same time interval.
- In the scenario *'failures'*, peers randomly fail so that messages get lost, contacts in routing table become outdated, etc.

Metrics. The catalogue of metrics offered by PeerfactSim.KOM includes the following: *Number of hops* is the number of peers contacted in order to resolve the lookup/search operation, while the *response time* shows the duration of the operation. The *overall success rate* is defined as the ratio of successfully resolved lookup/search operations to their overall number. Relative Delay Penalty (RDP) describes how well the overlay structure matches the underlying network topology. It is defined as the ratio of the measured latency introduced by sending a message from point A to B through the overlay structure to the corresponding latency when sending it directly through the underlay [14]). *Stale contact ratio* records each usage of stale contact information from the routing table. Peers join and leave the network and thereby their contact information in routing tables of other peers can be stale, influencing the overall performance of the protocol. Stale contact ratio is the share of messages sent to peers which already left the network over the total number of sent messages. *Message distribution* shows the exact portion of the total number of received messages for each peer. This metric can directly show the load balancing of the individual peers. *Message type distribution* sorts all received messages into different types, in order to depict protocol overhead or load balancing. Currently, the following five different types of messages (the list is certainly not complete) are identified: join, leave, maintenance, user message, and result transfer. *Stale message ratio* determines the percentage of lost messages caused by churn.

A benchmarking set for *stability* can be the 'unstable' scenario, with the metrics: number of hops, overall success rate, RDP, response time, and stale contact ratio. For *load balancing*, we use the 'ideal' scenario and the metric of message (type) distribution. RDP and message type distribution are crucial metrics for *efficiency* with the 'ideal' scenario.

3.6 XML-based Simulation Setup

All the information relevant for a simulation are described using an XML document which includes: *general settings for simulation* (observed overlay, number of participating peers, enabling network stabilization, random generators seed, etc.), *protocol-specific settings* (e.g., the number of successors in Chord [7] or the size of buckets in Kademlia [10]), *selection of churn rate* (currently available churn rates is given in Section 3.4.2), *selection of the benchmarking set used* (described in Section 3.5), *peer distribution* (described in Section 3.4.1), *peer behavior* (for individual peers, different groups of peers, and the default behavior of peers).

4 Performance of the Simulator

PeerfactSim.KOM supports extensibility due to its modular design. All functional components are clearly separated. Different design patterns [15] are used in order to cope with the complexity of numerous problems in a clear and efficient way. For example, memory consumption is optimized through the use of the singleton pattern (GOF 127 from [15]), object pooling and static objects. Extensibility is provided by using the composite pattern (GOF 163 from [15]).

4.1 Memory Usage

Both, memory usage and simulation runtime are influenced mainly by the overlay-specific parameters, the size of an experiment, the complexity of a scenario, and the level of detail in an underlay network model. To show how PeerfactSim.KOM is performing we take a look at the demand on resources for experiments in two overlay protocols, Chord and Kademlia. Both experiments were based on the same scenario.

All simulation runs were executed on an average desktop PC running at 1.8 GHz (AMD Athlon 3000+) with 1 GB RAM. We varied the number of peers simulated while all other parameters remained fixed. A simple scenario was used, containing only a few overlay operations. The node failure feature was deactivated, so that all nodes remained in the network until the end of the simulation. Fig. 4(b) and Fig. 4(c) depict the usage of memory during the simulations.

The resource demands of Kademlia increased considerably along with the number of nodes on the network and as time progressed. In contrast, memory consumption of the Chord protocol remained roughly consistent and only increased negligible as the experiment size increased. This was due to the dynamic growth of the routing tables along with the new contacts in the Kademlia protocol, which consumes more memory than the routing tables in Chord. In a network of 1000 peers, each node holds on average contact information of 113 contacts by the end of a simulation (see Fig. 4(a)).

4.2 Simulation Duration

Simulation time increases almost linearly with the number of peers (experiment size). Compared to Chord, Kademlia requires nearly twice as much time to simulate a simple scenario.

Fig. 5 depicts the simulation time in relation to experiment sizes. In a more complex scenario the difference is even more drastic. In a simulation with 10 000 nodes, with active failure rate and a bucket size of 20, it lasted 6 h 30 with the Kademlia protocol.

However, the same scenario in Chord simulation took only an hour. The reason behind this is the complex construction and management of the routing tables in Kademlia, which are calculated using an XOR-comparator.

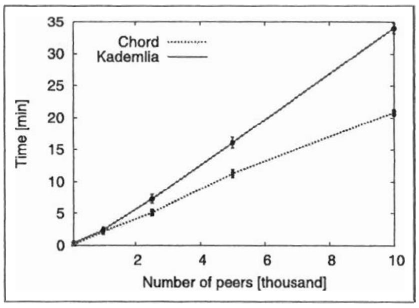
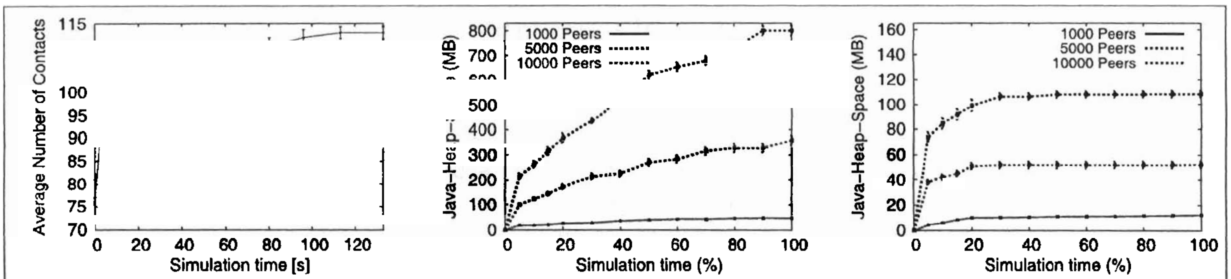


Figure 5 Duration of simulation Chord and Kademlia.

5 Related Work

Currently, most of the research papers on p2p overlays use simulators developed specifically for the purpose of that paper [16]. This makes it impossible to compare the evaluation from different papers, in an attempt to make a valid statement about the contribution of published research. The need for a general simulator for p2p overlays has been recognized and there are numerous solutions. Here we will consider a few of them as they share a similar aim – a general benchmarking platform for p2p systems. In this sense, P2PSim [17] despite its rich model of overlays, underlay network, and user behavior does not relate to our focus as it only supports structured p2p overlays. The aim of PeerSim [18] is to provide a scalable general simulation framework for p2p overlays.



(a) Average number of contacts during the simulation of Kademlia

(b) Kademlia ($k = 10$, $b = 5$, $\alpha = 3$)

(c) Chord

Figure 4 Memory consumption for simulating Kademlia and Chord.

However, only the cycle-based version of PeerSim scales well and is documented, but has no underlying network model like discrete-event based version. None of versions does model resource, service, or user-behavior model. The main focus of Overlay Weaver [19] is the support of both overlay design and application development. It can be used either as an emulator, by using real TCP/UDP messages, or as a simulator, using discrete event message passing without any underlying network model. It has poor scalability (4000 nodes) and lacks the other relevant models. The focus of NeuroGrid [20] is the simulation of file sharing p2p systems. Therefore, it offers a rich model for keyword-based search. It scales very well (300 000 nodes) but does not model underlay network, peer resources, nor churn. PlanetSim [21] has an additional objective which enables the easy transition from simulation code to experimentation code running on the Internet. Due to the lack of an underlying network model, PlanetSim scales up to 100 000 nodes of the simple Chord overlay. In addition, PlanetSim does not model peer resources nor peer behavior. OverSim [22] provides a scalable overlay network simulation framework (100 000 nodes) and has different levels of accuracy for its underlay network model. Similar to PlanetSim, it aims to achieve the reusability of simulation code for prototypes, but has no model of user behavior nor peer resources. All of the simulators mentioned (except Neurogrid) do not include a service model.

6 Conclusion

The need to benchmark p2p overlay networks has been increasing as the spectrum of peer-to-peer applications continues to grow. Once the benchmarking set (tuple of quality attributes, appropriate metrics, and scenarios) has been identified, an appropriate evaluation method has to be applied to each overlay in order to provide the same experiment conditions and obtain comparable results. Simulation, as the most appropriate evaluation method for peer-to-peer overlay networks, has to fulfill the requirements of usability, realistic models, and simulator performance. All these requirements have been identified in this paper and existing simulators for p2p overlay networks have been analyzed. In this paper, we presented PeerfactSim.KOM, a general simulation framework for p2p systems that best meets our requirements. Therefore, its primary goal is to provide a benchmarking platform – by offering catalogue of metrics and a simple but comprehensive scenario specification in an XML document for each quality attribute. It provides several models of location-dependent peer distribution and churn rates together with novel bitmap-based model. The underlying network model takes into account geographical distance between peers, the processing delay of intermediate systems, signal propagation, congestions, retransmission, and packet loss. The simulator includes a service model, which not only supports simulating basic overlay services (such as lookup or search) or file-sharing applications, but communication and collaboration applications as well. The overlay model includes the full implementation of Chord, Kademlia, CAN, Gnutella, globase.KOM, and Omicron. The simulator scales for around 10^6 participants for simple overlays such as Gnutella and 10^5 for more complex overlays like Kademlia. Future development of PeerfactSim.KOM will be focused on implementing various services and resource models, peer-to-peer overlays, metrics, peer distributions, and churn rates. A mobility model of the peers will be included to support the simulation of scenarios with nomadic peers like in rescue operations management. PeerfactSim.KOM is open source and it is published in [4].

References

[1] O. Heckmann et al. Qualitätsmerkmale von Peer-to-Peer-Systemen. Technical Report KOM-TR-2006-03, TU Darmstadt, May 2006.

[2] PlanetLab. An Open Platform for Developing, Deploying, and Accessing Planetary-Scale Services, 2006. www.planetlab.com.

[3] N. Spring, L. Peterson, A. Bavier, V. Pai. Using planetlab for network research: myths, realities, and best practices. *SIGOPS Oper. Syst. Rev.*, 40(1):17–24, 2006.

[4] PeerfactSim.KOM: A Simulator for Large-Scale Peer-to-Peer Networks. http://www.peerfactsim.com.

[5] O. Heckmann. *A System-oriented Approach to Efficiency and Quality of Service for Internet Service Providers.* PhD thesis, TU Darmstadt, 2004.

[6] V. Darlagiannis, A. Mauthe, N. Liebau, R. Steinmetz. An Adaptable, Role-based Simulator for P2P Networks. In *MSV'04*, pp. 52–59, 2004.

[7] I. Stoica et al. Chord: A Scalable Peer-To-Peer Lookup Service for Internet Applications. *IEEE/ACM Trans. Netw.*, 11(1):17–32, 2003.

[8] A. Kovačević, N. Liebau, R. Steinmetz. Globase.KOM – A P2P Overlay for Fully Retrievable Location-based Search. In *Proc. of the 7th IEEE Int'l Conf. on P2P Comp.*, 2007.

[9] S. Ratnasamy et al. A scalable content-addressable network. In *SIGCOMM '01*, 2001.

[10] P. Maymounkov, D. Mazieres. Kademlia: A Peer-to-Peer Information System Based on the XOR Metric. http://pdos.csail.mit.edu/~petar/papers/maymounkov-kademlia-lncs.ps.

[11] V. Darlagiannis. *Overlay Network Mechanisms for Peer-to-Peer Systems.* PhD thesis, TU Darmstadt, 2005.

[12] D. Stutzbach, R. Rejaie. Understanding churn in peer-to-peer networks. In *IMC '06*, 2006.

[13] F. E. Bustamante, Y. Qiao. Friendships that last: peer lifespan and its role in P2P protocols. In *Proc. of WCW8*, 2004.

[14] S. Jain, R. Mahajan, D. Wetherall. A Study of the Performance Potential of DHT-based Overlays. In *USENIX'03*, 2003.

[15] E. Gamma, R. Helm, R. Johnson, J. Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software.* Addison-Wesley Professional, 1995.
[16] S. Naicken, A. Basu, B. Livingston, S. Rodhetbhai, I. Wakeman. Towards Yet Another Peer-to-Peer Simulator. *HET-NETs,* 2006.
[17] P2Psim: A Simulator for Peer-to-Peer (P2P) Protocols. http://pdos.csail.mit.edu/p2psim.
[18] PeerSim. http://peersim.sourceforge.net.
[19] K. Shudo. Overlay Weaver. http://overlayweaver.sourceforge.net/.
[20] NeuroGrid. http://www.neurogrid.net/.
[21] P. Garcia et al. PlanetSim: A New Overlay Network Simulation Framework. In *LNCS, Software engineering and middleware. Int'l Workshop. No. 4, Linz,* vol. 3437, pp. 123–136, 2005.
[22] OverSim: A Flexible Overlay Network Simulation Framework. http://www.oversim.org/.


1


2


3


4


5

1 **Aleksandra Kovačević** is a PhD student at the Multimedia Communication Lab of Technische Universität Darmstadt. She graduated from the School of Electrical Engineering, University of Belgrade in 2004. Her research interests include peer-to-peer networks, particulary in location-based search, the investigation scalability and stability of overlay networks, as well as the simulation of large scale peer-to-peer networks.
Address: KOM – Multimedia Communications Lab, Merckstr. 25, 64283 Darmstadt, Germany, Tel.: +49-6151-166886,
Fax: +49-6151-166152,
E-Mail: sandra@KOM.tu-darmstadt.de

2 **Sebastian Kaune** is PhD student at the Multimedia Communication Lab of Technische Universität Darmstadt. Focus of his studies at Department of Computer Science was computer networks and distributed systems. He received his degree Dipl.-Inf. in 2006. His research interests include the development of novel distribution strategies in content distribution systems based on the P2P paradigm, simulation of peer-to-peer networks, and P2P applications.
Address: KOM – Multimedia Communications Lab, Merckstr. 25, 64283 Darmstadt, Germany, Tel.: +49-6151-166238,
Fax: +49-6151-166152,
E-Mail: kaune@KOM.tu-darmstadt.de

3 **Patrick Mukherjee** is a PhD student at Real-Time Systems Lab at the Technische Universität Darmstadt. He graduated in Computer Science at the Technical University of Berlin in October 2004. His research focuses on (semantic) correctness of versioned data objects distributed in peer-to-peer networks. He focuses in the building of peer-to-peer based platforms to support global project developement.
Address: ES – Real-Time Systems Lab, Merckstr. 25, 64283 Darmstadt, Germany, Tel.: +49-6151-163776,
Fax: +49-6151-166942, E-Mail: patrick.mukherjee@es.tu-darmstadt.de

4 **Nicolas Liebau** is PhD student at Multimedia Communication Lab of Technische Universität Darmstadt. He studied a joint degree in Electrical Engineering and Business Administration. He received his degree Dipl.-Wirtsch.-Ing. in 2002. Then he joined the Multimedia Communication Lab and was funded by a scholarship from DFG until 2005. Currently, he heads the Peer-to-Peer Networking Group at the Multimedia Communication Lab. His research focus is decentralized accounting and charging, trust mechanisms as well as decentralized pricing mechanisms in the context of Peer-to-Peer systems. Nicolas Liebau is member of the ACM and the IEEE.
Address: KOM – Multimedia Communications Lab, Merckstr. 25, 64283 Darmstadt, Germany, Tel.: +49-6151-165240,
Fax: +49-6151-166152,
E-Mail: liebau@KOM.tu-darmstadt.de

5 **Prof. Dr. Ralf Steinmetz** (F'99) worked for over nine years in industrial research and development of distributed multimedia systems and applications. He has been head, since 1996, of the Multimedia Communications Lab at Technische Universität Darmstadt, Germany. From 1997 to 2001 he directed the Fraunhofer (former GMD) Integrated Publishing Systems Institute IPSI in Darmstadt. In 1999 he founded the Hessian Telemedia Technology Competence Center (httc e. V.). His thematic focus in research and teaching is on multimedia communications with his vision of real "seamless multimedia communications". With over 200 refereed publications he has become ICCC Governor in 1999, was awarded the ranking of Fellow of both, the IEEE in 1999 and the ACM in 2002.
Address: KOM – Multimedia Communications Lab, Merckstr. 25, 64283 Darmstadt, Germany, Tel.: +49-6151-166151,
Fax: +49-6151-166152,
E-Mail: rst@KOM.tu-darmstadt.de