A Survey of Peer-to-Peer Simulators and Simulation Technology

[*1] He XU, [2] Suo-ping WANG, [3] Ru-chuan WANG, [4]Ping TAN
[*1]*College of Internet of Things, Nanjing University of Posts and Telecommunications, Nanjing 210003, China, xuhe2046@126.com*
[2]*College of Automation, Nanjing University of Posts and Telecommunications, Nanjing 210003, China, wangsp@njupt.edu.cn*
[3]*College of Computer, Nanjing University of Posts and Telecommunications, Nanjing 210003, China, wangrc@njupt.edu.cn*
[4]*School of Economics & Management, Zhejiang Sci-Tech University,Hangzhou,China*
doi:10.4156/jcit.vol6.issue5.30

Abstract

Because of the dynamic, diversity and complexity of Peer-to-Peer (P2P) network, this gives rise to a number of challenges to simulate P2P network. In this paper, we focus on various available P2P simulators and summarized them against a proposed set of attributes of each simulator by surveying a number of P2P simulators, such as scalability, architecture, programming language, etc. The survey shows that while there are a lot of simulators, there are many issues such as poor scalability, little or no documentation, no visualization, and no statistics limitations. The purpose of this paper is that we can select one of the existed P2P simulators as the foundation of simulation so as to carry out our research work properly on P2P system. In addition, we can design a new P2P simulator according to the criteria and requirements of P2P simulators.

Keywords: *P2P, Simulator, Network Simulation*

1. Introduction

Peer-to-Peer (P2P) network architecture is becoming one of the most popular service architectures in today's Internet[1, 2]. In a P2P network, nodes undertake the role of both client and server. This is in contrast with the client-server architecture, where client and server roles are separate. There are a variety of P2P protocols and systems have emerged. Existed P2P network can be grouped into the following three basic categories: (1) Pure: Pure P2P networks are those that have absolutely no centralized facilities. An example of such P2P architecture is the original Gnutella network. (2) Hybrid: Hybrid P2P networks incorporate some centralization into their architecture. For example, Napster uses a centralized server for searching for files. (3) Partially Centralized[3]: There are two types of partially centralized architectures. The first ones are similar to hybrid architectures, but instead of a single server there is a farm of servers with a P2P network at this level. The second ones are similar to pure architectures, but there are some nodes called supernode[4] that act as a central node. This supernode will perform the search for other supernode in order to find the requested file.

Simulation is nowadays most-widely used. By simulating P2P system, one is able to control every aspect of peer behavior. However, the P2P network is high dynamics[5, 6]: new peers arrive, existing peers can temporary disconnect or definitively abandon the network. Therefore, the simulators need to model the network dynamics. It is impossible to ask hundreds of thousands of P2P users to upgrade their software and participate in controlled tests. Literature [7] reports that out of 141 papers about P2P and based on simulation, 30.5% use a custom simulation tool while half of them do not even report which simulation tool was used. Inspired by the survey papers[7-11], we focus on various P2P simulators and summarize them against a proposed set of attributes such as scalability, architecture, programming language, etc.

2. Overview of P2P Simulators

In this section we analyze a set of existing simulators for P2P network.

2.1. P2Psim

P2PSim is part of the IRIS project[12]. P2PSim[13] is a discrete event packet level simulator that can simulate structured overlays only. P2PSim's main goal was to make the understanding of P2P protocols easier. P2PSim is written in C++ and supports Chord, Accordion, Koorde, Kelips, Tapestry, and Kademlia P2P protocols. P2PSim can simulate node failures and both iterative and recursive lookups are supported. Node IDs are generated by consistent 160-bit SHA-1 hashing. Distributed simulation, cross traffic and massive fluctuations of bandwidth are not supported. The C++ API documentation is poor, but implementation of other protocols can be built by extending certain base classes. Custom event generators can also be implemented by extending a base class. P2PSim does not provide any simulation visualizations or a GUI. However, P2PSim can be used in conjunction with a third party GTK application to provide a GUI. Perl scripts are provided for the generation of graphs. The P2PSim code suggests support for a wide range of underlying network topologies such as end-to-end time graph, G2 graph, GT-ITM, random graph and Euclidean graph, which is the most commonly used. P2PSim developers have tested its scalability with a 3,000-node Euclidean Constant Failure Model topology.

2.2. Peersim

Peersim[14, 15] is a Java-based overlay networks simulator for P2P, which is supporting for simulation of structured and unstructured P2P network. Peersim supports two modes of simulations – cycle-based and event-based. In cycle-based simulations, Peersim will sequentially execute all node protocols in one cycle. Between any two cycles, developers are allowed to add Control objects. These Control objects can be used to add or remove nodes, or monitor the values of specified variables. In event-based simulations, events are defined along a time axis. For each time tick, there may be zero or more events. Cycle-based engine is simplified by ignoring transport layer in protocol stack and lacks support for concurrency. Event driven engine supports dynamicity and is more realistic because it enables the simulation of protocol stacks. Peersim offers the best scalability among analyzed simulators as it can reach up to 10^6 nodes by using the cyclic model. PeerSim comes with few simple topologies and network observers, but it lacks any routing protocol. Moreover, the concept of node is extremely simplified and the concept of resource is never specified. The real advantage of Peersim is the engine that supports many extensible and pluggable components, with a flexible configuration mechanism. Many researchers realized their own protocols based on Peersim, such as Bandwidth management protocol, A fault-tolerant FSM, Pastry, Chord, Kademlia, Skipnet, BitTorrent, Aggregation, SG-1, Peer sampling service, T-Man, PdProtocol, Slacer, Resource World, SkillWorld, FirmNet, Cloudcast, Adam2 and so on.

2.3. PlanetSim

PlanetSim[16] is a discrete event-based simulation framework for P2P overlay networks and services, which supports structured (e.g., Chord, Symphony, SkipNet) and unstructured (e.g., Gnutella) P2P simulation. There are two mechanisms within PlanetSim to implement new P2P overlays: algorithm-based and behavior-based development methods. The former allows the implementation of all the P2P protocol into the Node entity, answering correspondingly to any incoming message. The latter encapsulates any individual part of the protocol into a Behavior entity, processing any incoming message as expected by the protocol, and where the Node is mainly a data container entity.

2.4. OverSim

OverSim[17] is an open-source overlay and P2P network simulation framework for the OMNeT++ simulation environment. The simulator contains several models for structured (e.g. Chord, Kademlia, Pastry) and unstructured (e.g. GIA) P2P systems and overlay protocols. OverSim was developed at the Institute of Telematics (research group Prof. Zitterbart), Karlsruhe Institute of Technology (KIT)

within the scope of the ScaleNet project funded by the German Federal Ministry of Education and Research. The simulator is actively developed and open to contributions.

2.5. Dnet

Dnet is the abbreviation of distributed simulation of P2P network[18]. Distributed simulation is to distribute the simulation over a number of computers. In order to reduce the resource-requirements (especially processing time), dnet is proposed by Wilkins A. Dnet is to be transparent to the application level, and be able to scale to simulations of tens to hundreds of thousands of nodes, using end-to-end, rather than point-to-point, packet-level simulation.

2.6. Optimal-sim

Haoyi Wan and Norihiro Ishikawa proposed Optimal-sim[19] to simulate the behavior of a P2P network on top of general Internet topology. Using Optimal-sim they evaluate the influence caused by the mismatch between the overlay topology and the underlying router-level topology.

2.7. P2PRealm

P2PRealm[20] is simulator for studying P2P networks. Its unique functionalities contain training methods for neural networks and optimized speed of execution. By combining P2PRealm with other tools developed in their project, the simulator can grow to a large-scale distributed P2P research environment.

2.8. P2PAM

P2PAM[21] is a PeerSim enhancement, providing an almost complete framework for P2P architectural modeling. The basic idea of P2PAM is to add another abstraction layer to PeerSim in order to let the developer focus only on routing protocol implementation and performance analysis. P2PAM comes with a set of network topologies defined across a protocol-level interface which simplifies the task of routing protocol development. The concept of node has been extended in order to make it more similar to a real peer node, with extensible data structures storing local resource descriptions, search queries, etc. A simulation automator has been introduced in order to simplify the task of simulating network dynamics (churn, publication and search) using a simple scripting language, rather than writing Java code and re-compiling each time. The effectiveness of P2PAM is demonstrated by showing how it has been used to develop simulations of two comparable routing protocols, JXTA-SRDI and HALO, with different unstructured network topologies. Due to its ease of use, P2PAM will be essential for study of complex P2P architectural models, considering different resource distributions but also different degrees of churn rate.

2.9. GPS

Yang et al in [22] presented the General P2P Simulator (GPS) which is an extensible framework for simulating P2P networks efficiently and accurately. Efficiency is accomplished by using message level simulation rather than packet level simulation. Accuracy is maintained by tracking the network infrastructure and using a flow model to accomplish accurate estimate of the message behavior. An important feature of GPS is that it supports modeling the download component which is commonly abstracted away in P2P simulators. The GPS framework provides all the infrastructures required for P2P simulation, so new protocols can be easily plugged in and even run on existing protocols.

2.10. SimGrid

The SimGrid[23] project was initiated in 1999 to allow the study of scheduling algorithms for heterogeneous platforms. SimGrid v3.3 is expected in 2008 and already available from the public SVN

repository, adds many new features with respect to the previous versions. It now supports dynamic resource availabilities and failures. Two user interfaces were added to allow the use of the software in new contexts. SimGrid has the following main components: two APIs for researchers who study algorithm and need to prototype simulations quickly, and two for developers who can develop applications in the comfort of the simulated world before deploying them seamlessly in the real world. SimGrid employs a modular simulation kernel that supports the addition and use of new resource models without changes in the user code. SimGrid consists of 30,000 lines of GPL'ed code. It is freely available from its Web page and comes with all relevant information as well as with several example programs and tutorials. SimGrid uses an extensive regression testing suite as well as set of automatic compilation daemons ensuring a reasonable level of software quality. It is ported to Linux, Windows, Mac OS X and AIX. SimGrid is a very active project, both in terms of research and development.

2.11. NeuroGrid

NeuroGrid[24] is an adaptive decentralized search system. NeuroGrid nodes support distributed search through semantic routing (forwarding of queries based on content), and a learning mechanism that dynamically adjusts metadata describing the contents of nodes and the files that make up those contents. NeuroGrid is an open-source project, and prototype software has been made available at [25].NeuroGrid presents users with an alternative to hierarchical, folder-based file organization, and in the process offers an alternative approach to distributed search. The NeuroGrid approach could be seen as an alternative to these P2P systems such as Chord and CAN if used to track file locations, or it could be seen as a discovery layer that operates on top of one of these P2P systems.

2.12. OPSS

OPSS[26, 27] is an Overlay P2P Streaming Simulator designed to simulate large scale (i.e. in the order of 100K nodes) P2P streaming systems. OPSS is a discrete-event fluid-flow simulator. It allows simulating the data distribution at the flow level, i.e. Neglecting transmissions of single packets and focusing on events, such as start/end of a file or a file chunk transmission, which lead to a variation in the rate of the connections among peers. OPSS is able to simulate a fair (i.e. "TCP-like") sharing of the uplink and downlink bandwidth among different connections, and it guarantees extensibility by allowing the implementation of different P2P streaming algorithms as separate modules. Source code of OPSS is available under the GPL license.

2.13. Overlay Weaver

Overlay Weaver[28] is a P2P overlay construction toolkit, which supports routing algorithm researchers in addition to application developers. It provides multiple routing (DHT) algorithms such as Chord, Kademlia, Koorde, Pastry and Tapestry and a distributed environment emulator. It is written in Java for structured overlays only. Whenever simulation performed using this, many pitfalls are there such as statistic gathering need a lot of work, documentation need to be written and the lack of scalability might need a fundamental redesign. A newly implemented algorithm can be tested, evaluated and compared on emulator, which can host tens of thousands of virtual nodes. It enables large-scale emulation and fair comparison between algorithms since simulations have to be run in real-time and there is no statistical output, its use as an overlay network simulator is very limited.

2.14. ChunkSim

P2P content distribution is a rapidly growing area and the current lack of good simulation tools severely limits the possibilities for research. ChunkSim[29] is an extensible simulation framework, specifically designed for investigating P2P content distribution systems. It incorporates all the key features of such systems: chunk selection, peer selection, peer arrivals and departures, as well as a simple but sufficient network connectivity model.

2.15. 3LS

3LS[30] is a time-stepped simulator that uses a central step-clock is used to simulate the timing. In 3LS the models for network, P2P protocol and user model are clearly separated. With the separation of the network, protocol and application model from each other, the simulation of various network topologies, for different protocol, applications, and user models becomes possible. Three levels have been defined: Network level (bottom), Protocol level (middle) and User level (top). Communication can only happen between the directly connected levels. The protocol-level, that is responsible for simulating the P2P protocol and application, and serves as the interface between the user-level and the network-level. Input information from the user is fed into the network level through a GUI interface or a file. Upon starting the simulator it is possible to either create the models for the three described levels or to choose among a library the ones most suited models/combination for the simulation run. As the simulation is running, the events are displayed on the command prompt screen. After the simulation has been completed, all simulation data is saved into a file for future analysis. Though simulation languages provide most of the features needed in programming a simulation model and the details of the simulation models can be easily changed, a general-purpose language was selected to provide "greater programming flexibility". Since Java is the preferred language of many P2P programmers it was chosen as the host-language for the 3LS simulator. Visualization of the network is done with the aid of the tool AiSee[31]. When screenshots of the p2p-network are to be visualized, files containing the information of the graph are created by 3LS using the Graph Description Language (GDL). Once the file is created a user can use AiSee to render an image of the graph.

2.16. OPNET

PNET is a well known commercial discrete event simulator. In literature [32], the authors examine suitability of OPNET as a tool for performance evaluation of P2P overlay networks. They created simple framework and used it to analyze the flooding search algorithm which is a popular technique for searching files in an unstructured P2P network. However, the size of simulated network was limited. When tries with simulation of network consisting of more than 5000 nodes lasted much longer and sometimes OPNET stopped responding at all.

2.17. Other P2P simulators

A number of simulators for P2P overlay networks have been proposed. To address a large number of needs and demands, most of these simulators only intend to be used by their own developers, and even when they are released to the public they target a very specific community, such as BitTorrent-like P2P protocol[33]. BitTorrent (BT) is a widely-used protocol that employs a novel algorithm for distributing the file-sharing load among peers in a way that ensures participation. The basic idea in BT is to divide a single large file into small pieces (a typical piece size is 256KB). Peers query a centralized server, which is called tracker, to retrieve peer list owning part or whole of the file they are interested in, and simultaneously download different pieces of the file from different peers. Chen et al in [34] presented Virtual BT which is lightweight for BitTorrent-like file sharing systems. BTSim Simulation Components in GPS[22] is the implementation of BT protocol specification. Vogeleer et al in [35] presented an implementation of a BT simulator intended for future use in investigating modifications to the BT system to provide streaming media capabilities.

There are also many researches on P2P simulation for personal project use, such as for study network security. Rice and Wright in [36] presented a useful simulator that models P2P networks using several Java classes to study how P2P networks resistant to the propagation of malicious code. Chen and Gray in [37] provided a workload-driven simulation framework to characterize non-scanning worms which allows P2P worms to evade many of today's detection methods aimed at random-scanning worms, and identify the parameters influencing their propagations. Aringhieri and Bonomi in [5] presented an agent-based simulation model for the evaluation of a generic trust and reputation systems (TRS) within a decentralized P2P network, and proposed a complete model of peers in which the behavior of both good and malicious peers is accurately defined.

There are some other P2P simulators as following description. Bakos B, et al [38] have proposed a JXTA based P2P resource management scheme for HLA[39] based distributed simulations and confirmed the feasibility of using JXTA for improving the reusability of simulation components. HLA [39] is one of the well-known distributed simulation standards that aim to provide the interoperability and reusability of simulation components. Antoniu G, et al [40] proposed the JUXMEM Project. Their contribution consisted in enhancing the JXTA Distributed Framework (JDF) to fulfill some of requirements for commandments which should be observed by a deployment and control tool to successfully support large-scale P2P experiments. This enhancement mainly includes a more precise and concise specification language describing the virtual network of JXTA peers, the ability to use various batch systems, and also to control the volatility conditions during large-scale tests. Some preliminary performance measurements for the basic operations are reported.

Merz et al [41] proposed NetSim, which is a lightweight, event-based simulation framework, dedicated to the simulation of P2P overlay construction and maintenance. It concentrates on the overlay itself and generally abstracts from lower layers except for the message delay (round-trip time, RTT) between two inter-connected nodes. By construction, it requires RTT information to be generally available for node-to-node communication delays. NetSim has been developed in the Java language for portability reasons, featuring a modular, object-oriented structure. Besides a concentration on the overlay level and the integration of network coordinates, a major design goal has been the provision of a simple, straightforward way of transforming the implemented protocol into a real application that performs actual work in a distributed fashion.

Amoretti et al [42] proposed a novel discrete event simulation tool, DEUS, which includes a powerful engine, a XML configuration parser and an extensible Java API for the development of complex system simulators. They presented the capabilities of their tool by showing a Chord simulator, and they implemented using the P2P-oriented extension package for DEUS, and they compared its features (coding requirements, support for simulation of dynamic scenarios, performance) with those of the same P2P system simulator developed with the PeerSim API.

PeerNS was proposed by Dhurandher et al in [43]. PeerNS was built to study problems of P2P networks and Gnutella, such as Gnutella suffers from serious problems of dead searches, complexity in network topology and network overloading. The simulator was developed on Fedora Core 5/Suse 10.0, but was targeted towards the Linux family of operating systems. The actual software development was carried out in a purely object-oriented fashion using C++ and templates. Scientific calculations were done using Gnu Scientific Library (GSL), based on industry-established standards. The C++ allowed a large amount of portability across multiple platforms. PeerNS works on actual P2P network statistics and, hence, it is very close to the real scenario.

Grid based distributed simulations are becoming more and more important with the increasing number of large-scale modeling and simulation applications. There are many researchers study on distributed simulations with the combination of P2P and Grid technology. Discrete Event System Specification (DEVS) [44] is a formal modeling and simulation formalism to facilitate the system design and verification. Boukerche in [45] proposed an architecture which is a multi-layered and contains a fully decentralized JXTA-core, which aims to address most of the difficulties in grid-based large-scale distributed simulations, including real-time capability, interoperability, fault-tolerance, dynamic load balancing, security, dynamic reconfiguration, scalability, and expandability. Harder et al in [46] presented a simulation of P2P network where nodes either require or offer resources, which can be thought as CPU time for example.

P2PStudio[47] has been developed as a monitoring, controlling and visualization tool for P2P networks. It uses a centralized architecture to gather events from a P2P network and can be used to visualize network topology and to send different commands to individual P2P nodes. The tool has been used with Chedar P2P network to study the behavior of different P2P resource discovery and topology management algorithms and for visualizing the results of NeuroSearch resource discovery algorithm produced by the P2PRealm[20] network simulator. P2PStudio is Java-based and it is divided into two separate programs: the user interface (UI) and the server. The graphical UI connects to the server program and uses it to carry out the commands entered by the user. The server program takes care of all of the communication between the UI and Chedar nodes. It also manages the data sent from Chedar nodes. Dividing the application into two programs allows mobility of the UI from the dedicated

hardware of the server. P2PStudio has been used for the following two P2P software: Chedar and P2PRealm for algorithm development. But the centralized architecture of P2PStudio is a potential bottleneck for scalability in the future when the size of the P2P networks being studied grows.

3. Criteria of P2P Simulators

In order to compare different simulators, some criteria and requirements have been proposed in [7, 8, 10, 17, 48-50] originally.

- Simulation Architecture: The operation and the design of the simulator. The criteria mainly include the simulation engines the simulator adopts, simulation level, whether the simulator supports distributed simulation and nodes churn.
- Usability: The criteria have to do with how easy the simulator is to learn and use, it include: whether the simulator has a clean API which allows protocol code to be easily understood, altered and ported to and from other simulators; how experiment scenarios are created, if there is a script language, how easy it is to learn and how expressive it is; what documentation exists and how easy it is to follow.
- Scalability: The simulator should be able to emulate thousands of peers in a reasonable amount of time and CPU occupancy. More peers should be supported by adding more computing and network resources.
- Underlying Network Simulation: P2P simulators take a number of different approaches to simulate from every packet to overlay layer. When simulating an underlying network, the variables associated with the network layer - such as link and packet latencies, link failures and packet loss - need to be dynamic.
- Statistics: Statistics is one of the most important aspects. The emulator should be able to collect statistical data such as sent, received network traffic per peer, relations of connecting peers, percentage of the file transferring, etc. Output should be organized in a format to post-process easily. The simulator should be convenient to deploy and perform data analysis. It includes whether the documents (manuals, documental API, user guideless.) are provided and how long the learning curve is.
- Visualization: The criteria mean whether there is a GUI support. In order to validate and debug new or existing overlay protocols, visualizes both the topology of the underlying network and the overlay topology in a customizable way is helpful.
- P2P protocol: The criteria concern about what kind of P2P protocol is supported by simulator.
- Programming language: The criteria concern about what kind of development language is used by simulator.

From the criteria above, we compare the surveyed simulators and the details are presented from Table 1 to Table 3.

4. Conclusions & Future Work

In this paper we have surveyed a lot of existing P2P simulators in recently years. Among in these surveyed simulators we conclude that OverSim may be the best for P2P researchers due to very high scalability, good documentation, visualization and dynamic in nature but it has also some disadvantages. We believe that there is a need for a P2P simulator that meets the requirements of P2P researchers, and we deduct that a decentralized framework will be dominant in the area of P2P simulations in the near future due to its flexibility, scalability, and the ease of reconfiguring simulation applications. We hope we can use the advantages from those P2P simulators to design a new generic P2P system simulator.

5. Acknowledgments

This work is supported by the Science & Technology Innovation Fund for Higher Education Institutions of Jiangsu Province (CX10B_196Z, CX10B_197Z, CX10B_198Z, CX10B_199Z).

6. References

[1]Park S, "A QoS-based P2P Streaming for Ubiquitous Personal IPTV", Advances in Information Sciences and Service Sciences, AICIT, vol. 2, no. 3, pp. 82-90, 2010.
[2]AlOmari S A, Sumari P, Al-Taweel S A, and Manasrah A M, " CUSTP: Custom Protocol for Audio and Video Conferencing System over P2P Networks", International Journal of Digital Content Technology and its Applications, AICIT, vol. 4, no. 3, pp. 61-74, 2010.
[3]Mauri J L, Fuster G, Santos J R D, and Domingo M E, "Analysis and Characterization of Peer-To-Peer Filesharing Networks", WSEAS Transactions on Systems, WSEAS, vol. 7, no. 3, pp. 2574-2579, 2004.
[4]Yu C, Gou X, Zhang C, and Ji Y, "Supernode Election Algorithm in P2P Network Based upon District Partition", International Journal of Digital Content Technology and its Applications, AICIT, vol. 5, no. 1, pp. 186-194, 2011.
[5]Aringhieri R and Bonomi D, "A Simulation Model for Trust and Reputation System Evaluation in a P2P Network", Computational Intelligence, Theory and Applications, pp. 169-180, 2006.
[6]Piccolo F L, Bianchi G and Cassella S, "Efficient simulation of bandwidth allocation dynamics in P2P Networks", In Proceedings of the IEEE Global Telecommunications Conference(GLOBECOM '06), San Francisco, CA, pp. 1-6, 2006.
[7]Naicken S, Livingston B, Basu A, Rodhetbhai S, Wakeman I, and Chalmers D, "The state of peer-to-peer simulators and simulations", ACM SIGCOMM Computer Communication Review, vol. 37, no. 2, pp. 95-98, 2007.
[8]Naicken S, Basu A, Livingston B, and Rodhetbhai S, "A survey of peer-to-peer network simulators", In Proceedings of the 7th Annual Postgraduate Symposium (PGNet '06), Liverpool,UK, pp. 1-8, 2006.
[9]Baker M and Lakhoo R, "Peer-to-peer simulators", Technical report, AMG. [2010-11-10]. http://acet.rdg.ac.uk/bin/projects/vre/reports/P2P_Simulators.pdf, 2007.
[10]Zhang D, Chen X and Yang H, "State of the art and challenges on peer-to-peer simulators", In Proceedings of the 5th International Conference on Wireless Communications, Networking and Mobile Computing(WiCom '09), Beijing,China, pp. 1-4, 2009.
[11]Bhardwaj R, Dixit V S and Upadhyay A K, "An Overview on Tools for Peer to Peer Network Simulation", International Journal of Computer Applications(IJCA), vol. 1, no. 19, pp. 70-76, 2010.
[12]IRIS. [2010-12-21]. http://iris.csail.mit.edu/
[13]P2PSim, a simulator for peer-to-peer protocols. [2010-11-21]. http://pdos.csail.mit.edu/p2psim/
[14]The Peersim simulator. [2010-11-19]. http://peersim.sourceforge.net/
[15]Lu E J L, Jang Y T and Hwang G W, "PeerSim Cooker: A GUI IDE for PeerSim", In Proceedings of the International Computer Symposium (ICS 2008), Tamkang University, Tamsui, Taiwan, vol. 2, pp. 267-272, 2008.
[16]Pujol-Ahullo J and García-López P, "PlanetSim: An extensible simulation tool for peer-to-peer networks and services", In Proceedings of the IEEE Ninth International Conference on Peer-to-Peer Computing(P2P '09), pp. 85-86, 2009.
[17]Baumgart I, Heep B and Krause S, "OverSim: A Flexible Overlay Network Simulation Framework", In Proceedings of the 10th IEEE Global Internet Symposium (GI '07) in conjunction with IEEE INFOCOM 2007, Anchorage, AK, USA, pp. 79-84, 2007.
[18]Wilkins A, "Distributed Simulation of Peer-to-Peer Networks", Technical Report. University of Western, Australia. [2010-11-10]. http://undergraduate.csse.uwa.edu.au/year4/Current/Students/Files/2005/AndrewWilkins/CorrectedDissertation.pdf, 2005.
[19]Wan H and Ishikawa N, "Design and implementation of a simulator for peer-to-peer networks: optimal-sim", In Proceedings of the IEEE Pacific Rim Conference on Communications, Computers and Signal Processing, Canada, pp. 105-108, 2005.
[20]Kotilainen N, Vapa M, Keltanen T, Auvinen A, and Vuori J, "P2PRealm-Peer-to-peer network simulator", In Proceedings of the 2006 11th International Workshop on Computer-Aided Modeling, Analysis and Design of Communication Links and Networks, pp. 93-99, 2006.

[21]Agosti M, Zanichelli F, Amoretti M, and Conte G, "P2PAM: a framework for peer-to-peer architectural modeling based on PeerSim", In Proceedings of the 1st International Conference on Simulation Tools and Techniques for Commuications, Networks and Systems & Workshops, Marseille, France, pp. 1-10, 2008.
[22]Yang W and Abu-Ghazaleh N, "GPS: a general peer-to-peer simulator and its use for modeling BitTorrent", In Proceedings of the 13th IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems, pp. 425-432, 2005.
[23]Casanova H, Legrand A and Quinson M, "SimGrid: A generic framework for large-scale distributed experiments", In Proceedings of the Tenth International Conference on Computer Modeling and Simulation(UKSIM 2008), Cambridge, UK, pp. 126-131, 2008.
[24]Joseph S, "NeuroGrid: Semantically routing queries in peer-to-peer networks", Web Engineering and Peer-to-Peer Computing, pp. 202-214, 2010.
[25]Neurogrid. http://www.neurogrid.net/
[26]Bracciale L, Piccolo F L, Luzzi D, and Salsano S, "OPSS: an Overlay Peer-to-peer Streaming Simulator for large-scale networks", ACM SIGMETRICS Performance Evaluation Review, vol. 35, no. 3, pp. 25-27, 2007.
[27]Bracciale L, Piccolo F L, Salsano S, and Luzzi D, "Simulation of Peer-to-peer streaming over large-scale networks using OPSS", In Proceedings of the 2nd international conference on Performance evaluation methodologies and tools, Nantes, France, pp. 1-10, 2007.
[28]Shudo K, Tanaka Y and Sekiguchi S, "Overlay weaver: An overlay construction toolkit", Computer Communications, vol. 31, no. 2, pp. 402-412, 2008.
[29]Kangasharju J, Schmidt U, Bradler D, and Schr Der-Bernhardi J, "ChunkSim: simulating peer-to-peer content distribution", In Proceedings of the 2007 spring simulaiton multiconference, Norfolk, Virginia, pp. 25-32, 2007.
[30]Ting N S and Deters R, "3LS- A Peer-to-Peer Network Simulator", In Proceedings of the third International Conference on Peer-to-Peer Computing (P2P 2003), pp. 212-213, 2003.
[31]AiSee. [2010-12-29]. http://www.aisee.com/
[32]Biernacki A, "Analysis of the flooding search algorithm with OPNET", Novel Algorithms and Techniques in Telecommunications and Networking, pp. 339-342, 2010.
[33]Eger K, Ho Feld T, Binzenh Fer A, and Kunzmann G, "Efficient simulation of large-scale p2p networks: packet-level vs. flow-level simulations", In Proceedings of the second workshop on Use of P2P, GRID and agents for the development of content networks(UPGRADE-CN'07), Monterey, California, USA, pp. 9-16, 2007.
[34]Chen X, Chu X and Liu J, "A Lightweight Emulator for BitTorrent-Like File Sharing Systems", In Proceedings of the 2010 IEEE International Conference on Communications (ICC), Cape Town, South Africa, pp. 1-5, 2010.
[35]De Vogeleer K, Erman D and Popescu A, "Simulating bittorrent", In Proceedings of the 1st international conference on Simulation tools and techniques for communications, networks and systems & workshops(QoSim 2008), Marseille, France, pp. 1-7, 2008.
[36]Rice D O and Wright G, "Peer-to-peer (P2P) Simulation for Network Security", In Proceedings of the 3rd Annual Symposium on Information Assurance(ASIA '08), pp. 26-33, 2008.
[37]Chen G and Gray R S, "Simulating non-scanning worms on peer-to-peer networks", In Proceedings of the 1st international conference on Scalable information systems, Hong Kong, pp. 29-41, 2006.
[38]EkloF M, Sparf M, Moradi F, and Ayani R, "Peer-to-Peer-Based Resource Management in Support of HLA-Based Distributed Simulations", Simulation, vol. 80, no. 4-5, pp. 181-190, 2004.
[39]"IEEE Standard for Modeling and Simulation High Level Architecture (HLA)Framework and Rules" in *1516-2010*: IEEE Standard, 2010,pp. 1-38.
[40]Antoniu G, Bougé L, Jan M, and Monnet S, "Large-scale deployment in P2P experiments using the JXTA distributed framework", In Proceedings of the Euro-Par 2004 Parallel Processing, pp. 1038-1047, 2004.
[41]Merz P, Priebe M and Wolf S, "A simulation framework for distributed super-peer topology construction using network coordinates", In Proceedings of the 16th Euromicro Conference on Parallel, Distributed and Network-Based Processing(PDP 2008), Toulouse, France, pp. 491-498,

2008.
[42]Amoretti M, Agosti M and Zanichelli F, "DEUS: a discrete event universal simulator", In Proceedings of the 2nd International Conference on Simulation Tools and Techniques, Rome, Italy, pp. 1-9, 2009.
[43]Dhurandher S K, Misra S, Obaidat M S, Singh I, Agarwal R, and Bhambhani B, "Simulating Peer-to-Peer networks", In Proceedings of the IEEE/ACS International Conference on Computer Systems and Applications(AICCSA 2009), Rabat, pp. 336-341, 2009.
[44]Syriani E, Vangheluwe H and Al-Mallah A, "Modelling and Simulation-based Design of a Distributed DEVS Simulator", Technical Report SOCS-TR-2010.3. McGill University, School of Computer Science. [2010-11-09]. http://www.cs.mcgill.ca/research/reports/2010/socs-tr-2010-3.pdf, 2010.
[45]Boukerche A and Zhang M, "Towards peer-to-peer based distributed simulations on a grid infrastructure", In Proceedings of the 41st Annual Simulation Symposium(ANSS 2008), Ottawa, Ont., pp. 212-219, 2008.
[46]Harder U and Martínez Ortu O F, "Simulation of a peer to peer market for Grid Computing", Analytical and Stochastic Modeling Techniques and Applications, pp. 234-248, 2008.
[47]Kotilainen N, Vapa M, Auvinen A, Weber M, and Vuori J, "P2PStudio: monitoring, controlling and visualization tool for peer-to-peer networks research ", In Proceedings of the ACM international workshop on Performance monitoring, measurement, and evaluation of heterogeneous wireless and wired networks, Terromolinos, Spain, pp. 9-12, 2006.
[48]Ting N S, "A generic peer-to-peer network simulator", In Proceedings of the 2003-2004 Grad Symposium, pp. 1-10, 2003.
[49]Joseph S, "An extendible open source P2P simulator", P2P Journal, vol. 1, pp. 1-15, 2003.
[50]Binzenhofer A, Hossfeld T, Kunzmann G, and Eger K, "Efficient simulation of large-scale P2P networks: Compact data structures", In Proceedings of the 15th EUROMICRO International Conference on Parallel, Distributed and Network-Based Processing(PDP '07), Napoli, pp. 467-474, 2007.

Table 1. Comparing with different P2P simulators

Simulator	P2PSim	Peersim	PlanetSim	OverSim	Dnet
Architecture	Discrete-event	Query-Cycle and Discrete-event	an object oriented simulation framework	a modular simulation framework, Discrete-event	a distributed simulator
Usability	Poor documentation	Only cycle based engine is documented.	Design and API thoroughly documented; detailed tutorial	Detailed tutorial	Document is not well given. The website can not be accessible.
Scalability max-nodes	3 000 nodes using Chord protocol	Very high 10^6 nodes using cycle based engine.	Medium (100 000)	Medium (100 000)	10 000
Underlying Network	a range, such as end-to-end time graph, G2,GT-ITM, random graph and Euclidean graph	Not modelled	Limited simulation of underlying network, but BRITE information could be used for more detail	Using INET Underlying and Simple Underlying	Using Stanford GraphBase for loading underlying network topologies from a file
Statistics	To summarize all logfiles into a single statistics file.	Components can be implemented to gather statistical data	There are no mechanism to collect statistics	The Global Observer module can be used as a statistics collector.	Take a strategy to calculating latency and hop-count values.
Visualization	Not provide any visualization or a GUI support. P2PSim can be used in conjunction with a third party GTK application to provide a GUI	Provides implementation method that support well known models such as random graph, lattice and BA-Graph.	has a visualize that makes use of pajek	Using OMNeT++	Using BRITE to generate topology representation
Support P2P Protocol	Chord, Accordion, Koorde, Kelips, Tapestry, and Kademlia.	Structured and unstructured overlays	Structured and unstructured overlays	Structured and unstructured overlays	focus on routing and content location for P2P.
Language	C++	Java	Java	C++	C, C++

Table 2. Comparing with different P2P simulators(Continue)

Simulator	Optimal-sim	P2PRealm	P2PAM	GPS	SimGrid
Architecture	Event-driven simulator	an efficient P2P network simulator for studying algo rithms based on neural networks.	network topologies defined across a protocol-level interface which simplifies the task of routing protocol development.	an extensible object oriented framework that allows modeling of alternative P2P protocols, network models, and flow-level models.	A simulation based framework for evaluating cluster, grid And P2P algorithms and heuristics.
Usability	Provide API to descript a P2P algorithm	Document is not well given.	ease of use	API is poorly documented.	Supplied documentation on web
Scalability max nodes	Medium (10 000)	Medium (100 000)	10 000	Very low (about 1000)	few 10 000
Underlying Network	Network topology is generated by BRITE as the underlying network	Not modelled	The topology package provides TopologyBuilder abstract class, which extends PeerSim's WireGraph class.	It complies with most common network topology generation tools	Not modelled
Statistics	Using the results managing tool	Provide files for statistics	Data can be stored in a file	provides option for using a separate logging thread to provide asynchronous logging	XBT implements data containers and logging mechanisms, support for statistics.
Visualization	Using BRITE' GUI	Using P2PStudio	provides specific classes, called observers, which access the graph structure of the network and analyze	Provide a GUI for visualization of simulation.	SimDag is designed for the investigation of scheduling heuristics for applications as task graphs.
Support P2P Protocol	To simulate the join, leave and failure behavior of peer nodes	P2P resource discovery algorithms: Breadth-First Search, Random Walker, Highest Degree Search and optimal path K-Steiner Tree approximation	JXTA and HALO routing	Structured and Unstructured overlays	The authors plan to improve usability in the P2P domain
Language	Java	Java	Java	Java	C or Java

Table 3. Comparing with different P2P simulators(Continue)

Simulator	Neuro-Grid	OPSS	Overlay Weaver	Chunk-Sim	3LS	OPNET
Architecture	Discrete-event for unstruct-ured networks	discrete-event fluid-flow simulator, and an Overlay P2P Streaming Simulator	distributed environment emulator	Discrete-event	time-stepped, using a central step-clocks sued to simulate the timing.	well known commercial discrete event simulator
Usability	Well supplied docume-ntation on web	Source code is available under the GPL license.	Documnetation is quite scattered. Statistics gathering need a lot of work.	Works on Windows, Linux, Solaris, and Mac OS platform.	The website is under construct-ion.	Not well document-ted for P2P simulator
Scalability	High (300000)	100K nodes	Low (4 000)	10 000	Very low (<1000)	5 000
Underlying Network	Not modelled	Overlay layer is to simulate the overlay network and interactions between peers.	Not modelled	can be extended to handle a more detailed model	Communi-cation can only happen between the directly connected levels.	Supported
Statistics	Mechanism for statistics gathering.	statistical functions included in Statistic class	there is no statistical output	Statistics are supported	result is stored to a file	statistics gathered at the global and local level
Visualization	Using Applet	a simple graphical interface done with NCURSES lib	There is a graphical real-time visualiser which helps in understanding the protocols operation.	There is the graphical interface	Using AiSee	Supported
Support P2P Protocol	Structured and Unstructur-ed overlays	simulate behavior of arbitrary tree-based or mesh-based approaches	Chord, Kademlia, Koorde, Pastry and Tapestry	P2P content distributio n systems	Unstruct-ured overlays	Can be extended to support structured and Unstructu red overlays
Language	Java	C++	Java	Java	Java	C