

An Overview on Tools for Peer to Peer Network Simulation

Rupali Bhardwaj
Dept of MCA,
KIET
Ghaziabad, India

V.S. Dixit
Dept of CS & IT
ARSD College
Delhi, India

Anil Kr. Upadhyay
Dept of IT
IMR
Ghaziabad, India

ABSTRACT

A simulation is an attempt to model a system in order to study it scientifically. Simulations are the most popular tool for examine peer-to-peer (P2P) applications. The cost of implementation of simulated model is less than that of large-scale experiments and, if carefully constructed, the simulated model can be more realistic than any tractable mathematical model. Simulating P2P overlay networks is a common problem for researchers and developers because P2P systems can consist of million of nodes and dynamic in nature. So that simulation for such a large dynamic network is difficult due to technical constraints even on the most powerful machines. In this paper we focus on various available P2P simulators and summarized them against a proposed set of attributes such as scalability, architecture language and pros and cons of each simulator.

Keywords

Peer to Peer, Simulator, Packet based, Flow based, structured, unstructured

1. INTRODUCTION

A P2P system in which every participating node acts both as a client and as a sever (servent) and share a part of their own hardware resources such as processing power, storage capacity or network bandwidth. These resources accessible by other peers directly without passing intermediate entities. The participants of such a network are thus resource providers as well as resource requestors. Peer to Peer systems have a no of advantages over client server system such as scalability, fault tolerance, dynamicity, and performance. Scientists/ Researchers conducted a large amount of research in some challenging areas such as security, Reliability, Flexibility, Load balancing, Searching etc. Experiments of their research must be shown to be valid and others must be able to reproduce results of them for comparison with their own solutions. According to Pawlikowski [25], there are two main possible experimentation streams: experimentation with the actual system and experimentation with a model of the system (see Figure 1).

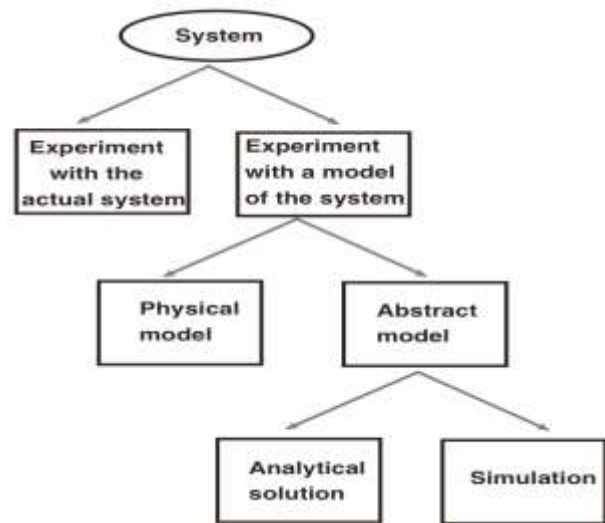


Figure 1-Experimentation Stream

Analytical solution requires mathematical modeling of P2P system. Real world complexity often prevents directs mathematical analysis of model. Another approach is to run experiments with actual system. P2P system consists of million of nodes, performing experiments with actual system require significant resources which could be very costly in hardware and administration. Simulators do not suffer from problems given above with analytical methods and experimenting with system itself.

Paper is structured as follows: in section 2, presented an overview on various P2P simulators .After that in last, paper concludes with a summary and conclusion.

2. P2P NETWORK SIMULATORS

Simulators can be classified into two categories packet based and flow based. According to types of simulators, a study of P2P simulators is split into two parts- Flow based simulators and Packet based simulators.

2.1 Flow Based Network Simulators

Flow based simulators or application level simulators,

means they work at application level and disregard

parts of TCP/network stack. Flow based simulators calculate bandwidth and delay to/ from network end points [12].

2.11 Peersim

Peersim has been designed to support dynamicity and scalability and written in Java language, offer predefined models for P2P simulation as overstat, SG-1 and T-man. It partly developed in BISON project [3] and is used by DELIS project. The engine consists of components which may be 'plugged in' and use a simple ASCII file based configuration mechanism which helps reduce the overhead. Two models of simulation are supported by peersim, cycle-based and event-based. 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 but decreases scalability of simulation [1, 26, 32].

Programming Language: Java

Scale: 10⁶ nodes using cycle based engine.

Support: Structured and unstructured overlays

Status: Active, latest release Peersim 1.0.4, 09-11-2008

Pros: 1. Very high scalability.
2. Cycle based and Event based simulation model
3. Support some well known models.
4. Supports dynamic network.

Cons: 1. Not having details of underlying communication network.
2. Not support distributed simulation.
3. Only cycle based engine is documented.

URL: <http://peersim.sourceforge.net>

2.12 P2PSim

P2P is C++ based discrete event simulator that can simulate structured overlay only. This multithreaded, discrete event simulator runs in several UNIX like operating system, does not provide any simulation visualization or a GUI support. P2PSim already supports Chord [9], Accordion [10], Koorde [13], Kelips [8], Tapestry [4] and Kademia [28] P2P Protocols. P2PSim can be used in conjunction with a third party GTK application to provide a GUI. Perl scripts are provided for generation of graphs. [1, 10, 22, 30]

Programming Language: C++

Scale: 3000 nodes using Chord protocol.

Support: Structured overlays

Status: Active, latest release P2PSim-0.3, 18-04-2005

Pros: 1. Simulate node failures.
2. Implementation of some well known protocols.
3. Support for a wide range of underlying network topologies.
4. Support Dynamic network.

Cons: 1. Not support distributed simulation
2. Documentation is poor.

URL: <http://pdos.csail.mit.edu/p2psim/>

2.13 3LS (3 Level Simulator)

3LS is a Java based time-stepped simulator that uses a central step-clock issued 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 protocols, applications, and user models becomes possible. Hence, three levels have been defined:

- Network level (describes distance between nodes)
- Protocol level (describes P2P protocol used)
- User level (user input via GUI or file).

Communication can only happen between the directly connected levels. The protocol-level, that is responsible for simulating the p2p-protocols and applications. Input information from the user is fed into the network level through a GUI interface or a file [1,18]. When simulation is completed, result is stored to a file and this is accessed by a visualization tool named Aisee [2].

Programming Language: Java

Scale: Fewer than 20 nodes using Gnutella 0.4

Support: Unstructured overlays

Status: NA

Cons: 1. Not support dynamic network.
2. Performance is not good enough.
3. Very low scalability.

URL: www.3ls.com or mail to

nyt431@mail.usask.ca, deters@cs.usask.ca

2.14 Neurogrid

The simulator is single threaded, Java based, and discrete event simulator. It is designed to work with distributed and decentralized networks. The distinctive feature of NeuroGrid is that it gradually adapts its own heuristic search over the web of computers. Consequently, it offers more efficient search results based on query-document relationships that reflect the way people search for documents. NeuroGrid monitors how frequently search terms lead to successful searches, personalizing search results to the way you think. It uses a configuration file to define the protocol which is to be simulated and network properties. [1, 16, 31, 32].

Programming Language: Java

Scale: Up to 3 lakh nodes on a 4GB PC

Support: Structured and Unstructured overlays

Status: Inactive, latest release Neurogrid 0.0.5 25-02-2002

Pros: 1. Mechanism for statistics gathering.
2. Well supplied documentation.
3. High scalability.
4. Supports dynamic network.

Cons: 1. Node failure is not simulated.
2. Not simulate the underlying network.

URL: www.neurogrid.net

2.15 Narses

It is a scalable, discrete event, flow based network simulator designed to avoid overhead of packet level simulator. Narses is written in Java targeted towards large distributed applications. It provides several network models that can be easily interchanged to provide best tradeoff between speed and accuracy. Narses reduces complexity of simulations by approximating behavior of physical, link, network and transport layer. It reduces memory and computational requirements by simulating a chunk of bytes not individual packets. To further reduce memory requirements, narses assumes that transfer between two end users is limited to the bandwidth allocated on the first link to the network. This ensures that there is no need for intermediate routers and reassessment of link bandwidth, hence further reducing the number of events in the simulation. Narses was reported to be 45 times faster and consumed 28% of memory of NS2. Simulation completion time maintains an average accuracy of 8%. Narses does not contain implementation of any overlay protocols [1, 15, 32, 34].

Programming Language: Java

Scale: 600 nodes

Support: Unstructured overlays

Status: Inactive, latest release Narses 0.1 a, 08-10-2003

Pros: 1. Reduction in memory requirements.
2. No need for intermediate routers
3. Faster as compare to NS2.

Cons: 1. Not contains any implementation of protocols.

URL: <http://sourceforge.net/projects/narses>

2.16 Query cycle

Query-Cycle Simulator is a file-sharing P2P network simulator that is standardized, extensible, and modeled after measurements in this model, both good and malicious peers form an unstructured P2P network. It has realistic models for content distribution, query activity, download behavior etc. Content distribution is based on a model where each file belongs to one category and that category is defined by popularity of file. Simulations proceed in query cycles representing time period between issuing a query and receiving a response. Generated queries are passed into a queue and handled on a FIFO basis. Simulator makes use of a GUI from which user can specify parameters for network characteristics, content distribution and peer behavior. Once these are committed and simulation is in a running mode, properties can not be modified but it is possible to halt, restart and save simulation for later execution [14, 29, 32]

Programming Language: Java

Scale: NA

Support: Unstructured overlays

Status: Inactive, 2003

Pros: 1. Used to demonstrate the adaptive topologies algorithm.

Cons: 1. Properties of simulation can not be modified during running mode of simulator.

URL: <http://p2p.stanford.edu/>

2.17 GPS

General purpose P2P simulator is written in Java and allows simulation of both structured and unstructured overlays. GPS described as a message level simulator which works at the application level of network stack. GPS provides a GUI for visualization of simulation. It models the performance of the network with more accuracy by allowing the users to use macroscopic models that account for bandwidth, delay and the effects of TCP. Thus, it provides the efficiency of message-level simulation with significantly better accuracy than existing simulators. GPS is distinguished from other P2P simulators by the following properties [7, 32, 35]-

1. It provides an extensible object oriented framework that allows modeling of alternative P2P protocols, alternative network models, and alternative flow-level models.
2. It complies with most common network topology generation tools and provides visualization support.
3. It provides simulation models for BitTorrent, which, to the best of our knowledge has not been modeled functionally before.

Programming Language: Java

Scale: NA

Support: Structured and Unstructured overlays

Pros: 1. Provides GUI for visualization of simulation.
2. Provide simulation model for Bittorrent.
3. Supports dynamic network.

Cons: 1. Not support distributed simulation.

2. API is poorly documented.

3. Support for extending the simulator to support protocols other than BitTorrent is limited.

Status: Inactive, [gps 1.2](#) 2005

URL: www.cs.binghamton.edu/~wyang/gps/

2.18 DHTSim

It is a desirable event simulator for structured overlays, specifically DHTs. It does not include much functionality for extracting statistics. It does not support distributed simulation, nor does it allow for nodes to fail. Churn of bytes can be simulated with two script commands which allow a number of nodes to join over a period of time or a number of randomly selected nodes to leave over a period of time. It is a basis for teaching the implementation of DHT protocols. API is fairly straightforward and documentation is limited. [5, 32]

Programming Language: Java

Scale: NA

Support: Structured overlays

Status: Active, dht-sim-0.3 2003

Pros: 1. Event simulator specifically for DHTs. .
2. API is fairly straightforward.

Cons: 1. Node failure is not simulated.

2. Not include much functionality for extracting statistics.

3. Not support distributed simulation.

URL:

<http://www.infomatics.sussex.ac.uk/users/ianw/teach/di-st-sys/dht-sim-0.3>

2.19 Overlay Weaver

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. [11,32].

Programming Language: Java

Scale: 4000 nodes

Support: Structured overlays

Status: active, overlay weaver 0.9.4, 17-05-2009

Pros: 1. Capabilities for distributed simulation.
2. API is clean and well designed so that source is quite readable.
3. Provides multiple routing algorithms as Chord, Kademlia, Koorde, Pastry and Tapestry.

Cons: 1. Statistics gathering need a lot of work.
2. Documentation is quite scattered.

URL: <http://overlayweaver.sourceforge.net/>

2.2 PACKET LEVEL SIMULATORS

Packet based simulators calculate delay, bandwidth and routing for each packet generated or used by simulation.

2.21 Gnutellasim

Gnutellasim is a scalable packet-level simulator that enables the complete evaluation of the Gnutella system with a detailed network model. GnutellaSim is based on a framework we designed for packet-level peer-to-peer system simulation, which features functional isolation and a protocol-centric structure, among other characteristics. The framework is designed to be extensible to incorporate different implementation alternatives for a specific peer-to-peer system and is portable to different network simulators. To support GnutellaSim and to facilitate the simulation of other applications on ns2, we also extended the TCP implementation in ns2 to make it closer to real TCP. The additional features include: receiver advertised window, sender buffer, Socket-like APIs, dynamic connection establishment of TCP, and real payload transfer. This part of the code could be independently used [6, 33].

Programming Language: C++

Scale: 600 nodes

Support: Structured and Unstructured overlays

Status: Inactive, latest release Gnutella 0.4, 2003

Pros: 1. Framework is portable to different network simulators.

Cons: 1. poor scalability.

URL:

<http://www.cc.gatech.edu/computing/compas/gnutella/>

2.22 OMNeT++

OMNeT++ is a discrete event simulation environment. Although it is not strictly a network simulator but gaining popularity in this domain. OMNET++ itself does not provide components for simulations, instead there are simulation models and frameworks which are used with OMNET++. Components (*modules*) are programmed in C++, and then assembled into larger components and models using a high-level language (*NED*). Reusability of models comes for free. OMNeT++ has extensive GUI support, and due to its modular architecture, the simulation kernel (and models) can be embedded easily into your applications. It has been successfully used in other areas like the simulation of IT systems, queuing networks, hardware architectures and business processes as well. Omnet++ also has a number of contributed models, which provide P2P protocols or simulations. A 1000 node swarming simulation was achieved with OMNet ++ and the P2P swarming [23] add-on by the developer. Multi-tier topologies are supported by OMNet++. Java interoperability is provided by the JSimpleModule [20], which is an extension that allows OMNet++ modules to be written in Java [1, 12, 19].

Programming Language: C++

Scale: 1000

Support: Structured overlay

Status: active, omnet++ 4.0, 27-02-2009

Pros: 1. Vast GUI support.
2. Reusability of simulation models.
3. Multi tier topologies are supported.

Cons: 1. Low scalability.

URL: <http://www.omnetpp.org>

2.23 Planetsim

PlanetSim is an object oriented simulation framework for overlay networks and services. In PlanetSim, developers can work at two main levels: creating and testing new overlay algorithms like Chord or Pastry, or creating and testing new services (DHT, CAST, DOLR, etc) on top of existing overlays. PlanetSim also aims to enable a smooth transition from simulation code to experimentation code running in the Internet. Because of this, we provide wrapper code that takes care of network communication and permits us to run the same code in network testbeds such as PlanetLab. Moreover, distributed services in the simulator use the Common API for Structured Overlays. This enables complete transparency to services running either against the simulator or the network. There are no mechanism to collect statistics

from simulation runs, simulator supports churn and node failure, has a visualize that makes use of pajek [24] /GML and simulation can be saved to disk for reuse [12, 27, 30, 32].

Programming Language: Java

Scale: 100,000 nodes

Support: Structured and unstructured overlays

Status: active, planetsim v3.0, 27-07-06

Pros: 1. Output network topology graph in GML and pajek formats.
2. API is clean and well designed so that source is quite readable.

Cons: 1. No mechanism for statistics gathering.
2. Only simple underlying networks are provided.

URL: <http://projects-deim.urv.cat/trac/planetsim/>

2.24 OverSim

OverSim is an open-source overlay network simulation framework for the OMNeT++/OMNEST simulation environment. The peer-to-peer simulator contains several models for structured (e.g. Chord [9], Kademia [28], Pastry) and unstructured (e.g. GIA) P2P protocols. Oversim shares many characteristics with OMNeT++ such as GUI interface for validation and debugging new or existing overlay protocols [12,21].

Programming Language: C++

Scale: 100,000

Support: Structured and unstructured overlays

Status: active, [oversim-20090908](http://www.oversim.org/), 08-09-2009

URL: <http://www.oversim.org/>

2.25 NS-2

NS-2 is a discrete event simulator targeted at networking research. It provides substantial support for simulation of TCP, routing and multicast protocol over wired and wireless network. Currently there is only one p2p simulation available for NS-2(Gnutella).NS-2 performed simulation using a mixture of C++ and OTCL (object oriented version of TCL).The TCL script define the nodes and characteristics of communication links, while protocols are implemented in C++.NS-2 uses NAM(network animator) to provide visualization. NAM also allows users to arrange network graphs to aid designing and debugging of network protocols. GT-IT provides three types of topology, flat, N-level hierarchal and transit-stub can be used to generate topologies for NS-2. Due to NS-2 realistic nature scalability is a major issue. However like most packet level simulators NS-2 can run in parallel with a number of other machines [1,12,17].

Programming Language: C++

Scale: NA

Support: Structured and unstructured overlay

Status: active, NS-2.34, 17-06-2009

Pros: 1. Support visualization using NAM.
2. Run parallel with other machines.
3. Provide extensive documentation.

Cons: 1. Very low scalability.
2. Not dynamic support.

URL: www.isi.edu/nsnam/ns

3 SUMMARY & CONCLUSION

P2P systems became an interesting research area since beginning of twenty first century. Scientists/ Researchers conducted a large amount of research in some challenging areas such as security, Reliability, Flexibility, Load balancing, Searching etc. Experiments of their research must be shown to be valid and others must be able to reproduce results of them for comparison with their own solutions. So, for that purpose concept of simulator was introduced. Many simulators are developed from last few years as shown in figure 2. We survey many simulators and find that many of them have no mechanism to allow a user to gather statistics of a simulation run does not support distributed simulation, not support dynamicity. We believe that the poor state of existing P2P simulators is the reason that much published research work makes use of custom built simulators, which complicates the task of validating research and reproducing results, as these simulators are often not released publicly. Among in these surveyed simulators peersim is best for p2p researchers due to very high scalability, good documentation 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.

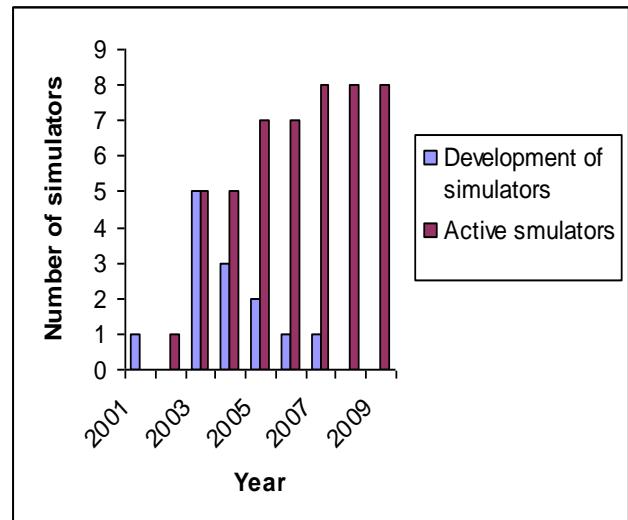


Figure 2 –History of simulators

References

[1] A. Brown Stirling, M. Kolberg Stirling, "Tools for Peer to Peer Netwrk Simulation", IRTF P2pRG

[2] AbsInt. Aisee homepage.
<http://www.aisee.com>

- [3]“BISON: Biology-Inspired Techniques for Self-Organization in Dynamic Networks.” Available: <http://www.cs.unibo.it/bison/>
- [4] B. Zhao, J. Kubiatowicz, and A. Joseph, “Tapestry: An Infrastructure for Fault-Tolerant Wide-Area Location and Routing,” *Computer*, 2001.
- [5]“DHTSim”, Available: <http://www.informatics.sussex.ac.uk/users/ianw/teach/dist-sys/>
- [6]Gnutellasim, Available: <http://www.cc.gatech.edu/computing/compass/gnutella/>
- [7]GPS, Available: www.cs.binghamton.edu/~wyang/gps/
- [8] I. Gupta, K. Birman, P. Linga, A. Demers, and R. van Renesse, “Kelips: Building an Efficient and Stable P2P DHT Through Increased Memory and Background Overhead,” *Group*, vol. 30, no. 1490, p. 23ms.
- [9] I. Stoica, R. Morris, D. Liben-Nowell, D. Karger, M. Kaashoek, F. Dabek, and H. Balakrishnan, “Chord: a scalable peer-to-peer lookup protocol for internet applications,” *IEEE/ACM Transactions on Networking*, vol. 11, no. 1, pp. 17–32, Feb. 2003.
- [10] J. Li, J. Stribling, R. Morris, and M. Kaashoek, “Bandwidth-Efficient Management of DHT Routing Tables,” *Proceedings of the 2nd NSDI*, 2005.
- [11] K. Shudo, “Overlay Weaver,” Available: <http://overlayweaver.sourceforge.net/>
- [12] M. Baker, Rahim Lakhoo, “Peer to Peer Simulators”
- [13] M. Kaashoek and D. Karger, “Koorde: A Simple Degree-Optimal Distributed Hash Table,” *Proceedings of the 2nd International Workshop on Peer-to-Peer Systems (IPTPS '03)*, 2003.
- [14] M.T. Schlosser and S.D. Kamvar, “Simulating a File Sharing P2P Network”, Stanford University, Tech. Rep. 2002.
- [15] “Narses Network Simulator,” Available: <http://sourceforge.net/projects/narses>
- [16]“NeuroGrid,” Available: <http://www.neurogrid.net/>
- [17] NS2, Available: www.isi.edu/nsnam/ns
- [18] Nyik San Ting, Ralph Deters, “3LS – A Peer to Peer Network Simulator”, P2P 2003.
- [19] OMNeT++, Available: <http://www.omnetpp.org/>
- [20] OMNeT++, JsimpleModule, <http://www.omnetpp.org/pmwiki/index.php?n=Main.JsimpleModule>
- [21] OverSim: The Overlay Simulation Framework. <http://www.oversim.org>
- [22] “P2Psim: A Simulator for Peer-to-Peer (P2P) Protocols,” Available: <http://pdos.csail.mit.edu/p2psim/>
- [23] P2P Swarming Protocol Simulation, <http://me55enger.net/index.cgi/software/swarm>
- [24] “Pajek - Program for Large Network Analysis.” [Online]. Available: <http://vlado.fmf.uni-lj.si/pub/networks/pajek/>
- [25] Pawlikowski, K, “Simulation Modeling and Analysis with Peer Computing “, (*P2P 2004*). Zurich, Switzerland: IEEE, Aug. 2004,
- [26] “PeerSim P2P Simulator,” Available: <http://peersim.sourceforge.net/>
- [27] planetsim, Available: <http://planet.urv.es/planetsim/> pp. 202–209. *Proc. 19th IEEE ASE*, 2004.
- [28] P. Maymounkov and D. Mazières, “Kademlia: A peer-to-peer information system based on the xor metric,” in *Peer-to-Peer Systems: First International Workshop, IPTPS 2002 Cambridge, MA, USA, March 7-8, 2002. Revised Papers*, vol. Volume 2429/2002, 2002, pp. 53–65.
- [29] Querycycle, Available: <http://p2p.stanford.edu/>

[30] R. RALLO, “PlanetSim: A New Overlay Network Simulation Framework,” Available: <http://planet.urv.es/planetsim/>

[31] S. Joseph, “NeuroGrid: Semantically Routing Queries in Peer-to-Peer telecommunication networks.”, Proceedings of the International Workshop on Peer To Peer Computing, 2002.
Simulators,” Proceedings of The Seventh Annual Postgraduate Symposium, Liverpool, UK, 2006.

[33] “The Annotated Gnutella Protocol Specification v0.4.”, Available <http://rfc-gnutella.sourceforge.net/developer/stable/index.html>

[34] TJ Guili and Mary Barker, “ Narses : Scalable flow based network simulator”, arXiv:cs.PF/0211024 vl 20 Nov 2002

[35] W. Yang and N. Abu-Ghazaleh, “GPS: A General Peer-to-Peer Simulator and its use for Modeling BitTorrent”, *Mascots*, vol. 00, pp. 425-434, 2005

[32] S. Naicken, A. Basu, B. Livingston, and S. Rodhetbhai, “A Survey of Peer-to-Peer Network

Appendix

Table1-Summary Report of P2P Simulators

Simulator	Language	Support	Scalability(max nodes)	Year	Status	URL
Peersim	Java	Structured and unstructured overlays	10 ⁶ nodes	2004	Active	http://peersim.sourceforge.net
P2PSim	C++	Structured overlays	3000 nodes	2004	Active	http://pdos.csail.mit.edu/P2PSim
3LS	Java	Unstructured overlays	Fewer than 20 nodes	NA	NA	www.3ls.com
Neurogrid	Java	Structured and Unstructured overlays	Up to 3 lakh nodes	2001	Inactive	www.neurogrid.net
Narses	Java	Unstructured overlays	600 nodes	2003	Inactive	http://sourceforge.net/projects/narses
Querycycle	Java	Unstructured overlays	NA	2003	Inactive	http://p2p.stanford.edu/
GPS	Java	Structured and Unstructured overlays	NA	2005	Inactive	www.cs.binghamton.edu/~wyang/gps/
DHTSim	Java	Structured overlays	NA	2003	Active	http://www.informatics.sussex.ac.uk/users/ianw/teach/dist-sys/dht-sim-0.3
Overlay Weaver	Java	Structured overlays	4000 nodes	2006	Active	http://overlayweaver.sourceforge.net/
Gnutellsim	C++	Structured and Unstructured overlays	600 nodes	2003	Inactive	http://www.cc.gatech.edu/computing/compass/gnutella/
OMNeT++	C++	?	1000 nodes	2003	Active	http://www.omnetpp.org/filemgmt/
Planetsim	Java	Structured overlays	100,000 nodes	2004	Active	http://projects-deim.urv.cat/trac/planetsim/
OverSim	C++	Structured and Unstructured overlays	100,000 nodes	2007	Active	http://www.oversim.org/
NS-2	C++	Structured and Unstructured overlays	NA	2005	Active	www.isi.edu/nsnam/ns