The State of Peer-to-Peer Simulators and Simulations

S. Naicken B. Livingston A. Basu S. Rodhetbhai I. Wakeman D. Chalmers
Department of Informatics, University of Sussex, Brighton, U.K.
{stephenn, b.j.livingston, a.basu, s.rodhetbhai, ianw, d.chalmers}@sussex.ac.uk

This article is an editorial note submitted to CCR. It has NOT been peer reviewed. Authors take full responsibility for this article's technical content. Comments can be posted through CCR Online.

ABSTRACT

In this paper, we discuss the current situation with respect to simulation usage in P2P research, testing the available P2P simulators against a proposed set of requirements, and surveying over 280 papers to discover what simulators are already being used. We found that no simulator currently meets all our requirements, and that simulation results are generally reported in the literature in a fashion that precludes any reproduction of results. We hope that this paper will give rise to further discussion and knowledge sharing among those of the P2P and network simulation research communities, so that a simulator that meets the needs of rigorous P2P research can be developed.

Categories and Subject Descriptors

I.6.7 [**Simulation and Modeling**]: Simulation Support Systems

General Terms

Design, Experimentation, Measurement, Performance

Keywords

Peer-to-Peer, Simulator Evaluation, Simulator Usage

1. INTRODUCTION

Simulations are the most popular tool for investigating overlay networks and peer-to-peer (P2P) applications. The cost of implementation is less than that of a large-scale experiment as significantly less computational resources are required and, if carefully constructed, the simulated model can be more realistic than any tractable mathematical model. The normal trajectory of P2P research is thus to first simulate and then experiment in the real world, where the re-use of simulation code is a practical advantage. There is no reason why simulation results should not be reproducible. However during the course of our research we have found that not only are there many different simulators in use, but reporting of which simulators are being used, and under what conditions, is sparse at best. Many of the simulators we investigated did not have functionality included to produce meaningful data for analysis. A lack of standard evidence gathering makes any claims on results hard to make with confidence and even harder for other researchers to verify. In addition, most had such poor documentation that implementing well-known overlay algorithms, such as Chord, was very difficult.

In this paper, we investigate the quality of P2P simulators and their use. We report on an evaluation of the available P2P simulators, and then investigate over 280 papers to determine how simulation use has been reported.

2. PEER-TO-PEER SIMULATOR SURVEY

This research was prompted by our own need to use P2P simulation to examine systems of our own design, and rather than re-inventing the wheel, we looked at existing P2P simulators. From our knowledge of P2P simulation research and research on the web we identified nine candidate simulators, described in table 1. In table 2 we evaluate these simulators with respect to the following criteria:

Simulation Architecture This relates to the design and functioning of the simulator, what features it includes and how they are implemented e.g. is it a discrete-event simulator and what node behaviours are simulated and how these are implemented, for example whether churn can be specified (nodes joining and leaving the DHT).

Usability These criteria have to do with how easy the simulator is to learn and use. They include: whether the simulator has a clean API, such as the Common API [2], which allows protocol code to be easily implemented, 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 P2P protocols are generally designed to be scalable and are often designed to solve problems of scalability. So one of the most important tests a simulator can provide is how the protocol scales to thousands of nodes or more, especially as it would be hard to conduct ongoing experiments on this number of machines. How well does the simulator make use of the available computing resources? If it is inefficient in its use of these resources then this will reduce its ability to scale.

Statistics Another key aspect of a simulator is the results it produces. The results need to be expressive and easy to manipulate in order to carry out statistical analysis. Mechanisms should exist that allow for the repeatability of experiments such as saving simulator state so that the reproducibility of results can be verified.

Underlying Network Simulation P2P simulators take a number of different approaches to simulating the underlying network, from simulating packet and links in detail to completely abstracting away everything below the overlay layer. The criteria used to assess this aspect of the simulators include: which properties of the network layer can be simulated; whether cross-traffic (other network traffic unrelated to the overlay) can be simulated; whether it can simulate differences in link latency; how realistic the underlying topology is.

The evaluation methodology was simple. Using resources such as manuals, source code and research papers, each simulator was evaluated with respect to the criteria above. In addition to this, attempts were made at implementing the Chord algorithm [10] for each simulator so that usage experiences could also be considered in the evaluation. We have omitted some of the simulators from the table 2 as they are not of importance with respect to this paper (e.g. one is a teaching tool unsuitable for use as simulator for research) and due to space considerations.

Not one simulator fully satisfied the evaluation criteria, with some having greater deficiencies than others. Of most concern was the lack of support for collecting statistics from simulation runs. Some simulators, such as PlanetSim [4], have no mechanism to facilitate collecting statistics, and of those that do, all require some degree of hacking so that the user can measure custom or non-preassigned variables.

Usability and in particular poor documentation is a problem with many of the simulators surveyed. Chord was already available on PlanetSim, OverlayWeaver, P2PSim and DHTSim. On only one of the other simulators were we able to successfully implement Chord, PeerSim[1], and even this required significant changes to the simulator code. This poor success rate can be attributed largely to the poor documentation of some simulators.

Authors often made claims to the scalability of their simulators, but even if achievable, these are significantly less than the numbers that can be achieved in real P2P systems. While it is unrealistic to expect levels of scalability in the order of millions of nodes, for some simulators improvements can be made. For example, with Overlay Weaver only a maximum of 2700 nodes could be simulated using a dual-Xeon 1.8Ghz computer running Linux with 1GB RAM and 2GB swap space. At 2700, the operating system thread limit was reached, thus limiting the scalability.

3. SURVEYING THE USE OF SIMULATION IN P2P RESEARCH

In this section, we investigate how simulations are used and presented within the literature. We examined 287 papers on P2P networking. The papers were identified by performing appropriate keyword searches on the ACM Digital Library[2] and other portals. The papers were then evaluated with respect to a common template, which was developed through in-depth reading of selected papers. Checks were done across experimenters to ensure inter-reader consistency. The papers were surveyed to find out which simulator, if any, they use for their simulations. Results are shown in figure 1.

"None" includes papers which involve no simulation in justifying the validity of the ideas presented. These papers are a combination of off-topic papers identified by the keyword search, position papers, and those using the other validation approaches described above. "Unspecified" are papers that discuss simulations and provide results, but do not state the simulator used. The paper's authors might have created one themselves or used an existing simulator, but it has not been named or referenced. "Custom" is all papers which state that a simulator was created specifically for simulating the algorithms and systems described in the paper. Some of these papers go on to describe the details of the simulator at length. "NS-2" includes all papers which used the general, packet-level network simulator NS-2. This program provides extensive low-level simulation of networks and as such is not specifically a P2P network simulator. The papers categorised as "Chord" used a modified version of the SFS[3] based simulator used in Chord [10].

Of the papers which did use simulation, more than half failed to specify which simulator was used. This could make it difficult to fully realise the significance of the results and harder to reliably reproduce them.

We found that of the 70 papers which stated which simulator they used, 62% of them used a specially created simulator. Some of these simulators might possibly be the same, reused within research groups. However, even taking this into account, the number of custom-made simulators far outnumbers the use of known simulators. This is not an ideal state of affairs, both in terms of duplication of effort and for ease of comparison and replication of results.

There was some concern that the inclusion of short papers may have been distorting the results as these were thought to be more likely to omit detailed information due to space constraints. There were similar numbers of 2 and 10 page papers, so in table 3 we sample these to see if omission due to space is a significant effect. However, we see that there is little difference in simulator usage between results reported in short and longer papers.

	Usage count in papers of ...	
Simulator	**2-pages**	**10-pages**
None	24	19
Unspecified	10	15
Custom	5	6
ns-2	0	3
Javasim	1	0
Nab	1	0
CSIM 19	0	1
Total	41	44

Table 3: Comparing Simulator Usage in Papers of Different Length

NS-2 is a popular network simulator with extensive documentation but has a steep learning curve. Some authors may already be familiar with NS-2 and so opt to use it in their P2P research to avoid wasting the great deal of time spent learning how to use another simulator. NS-2 is used in 11% papers which is surprising as NS-2 does not lend itself well to P2P simulations. It is designed for performing simulations at the network layer whereas most P2P research

[1] http://peersim.sourceforge.net/

[2] http://portal.acm.org/dl.cfm

[3] http://www.fs.net/sfswww/

Simulator	Language	Status	License	Notes & URL
P2PSim	C++	Active	GPL	Developed in the IRIS project at the Massachusetts Institute of Technology, USA. http://pdos.csail.mit.edu/p2psim/
PeerSim	Java	Active	LGPL	Developed in the BISON project at University of Bologna, Italy. http://peersim.sourceforge.net/
Query-Cycle Simulator	Java	Inactive	Apache	Developed in the P2P Sociology Project at Stanford University, USA. http://p2p.stanford.edu/
Narses	Java	Inactive	GPL-like	Stanford University, USA. http://sourceforge.net/projects/narses
Neurogrid	Java	Inactive	GPL	Developed by Dr. Sam Joseph (http://www2.hawaii.edu/~srjoseph/). http://www.neurogrid.net/
GPS	Java	Inactive	Open-Source, No License	University of Binghamton, USA. http://www.cs.binghamton.edu/wyang/gps/
Overlay Weaver	Java	Active	Apache	Developed by Dr. Kazuyuki Shudo (http://www.shudo.net/). http://overlayweaver.sourceforge.net/
DHTSim	Java	Active	GPL	Originally developed at the University of Pennsylvania, USA. Modified and used in teaching at University of Sussex, UK. http://www.informatics.sussex.ac.uk/users/ianw/teach/dist-sys/
PlanetSim	Java	Active	LGPL	Developed in the Planet project at the University Rovira i Virgili, Spain. http://planet.urv.es/planetsim/

Table 1: Project details of surveyed simulators.

Simulator	Architecture	Usability	Scalability (max nodes)	Statistics	Underlying Network
P2PSim	Discrete-event for structured P2P networks	Poor documentation	3000 nodes	Limited set of statistics can be collected before coding required	A range, including: end-to-end time graph, G2 graph, GT-ITM, random, and Euclidean
PeerSim	Query-Cycle or Discrete-event for unstructured networks. Can simulate nodes joining, departing and failing.	Only Query-Cycle simulator is documented	10^6 (Query-Cycle)	Components can be implemented to gather statistical data	Not modelled
Narses	Discrete-event, flow-based for tunable topologies	No documentation, source code is difficult to understand	600 nodes, depending on the underlying topology, untested by us	Yes, but requires modification of source	A number of underlying topologies, balancing execution speed and accuracy
Overlay Weaver	Distributed Emulation and a number of structured overlay algorithms	API and source code well documented, but some documentation missing	4000 nodes, we obtained only 2700 due to kernel & GLibC thread limits	Not possible to gather statistics	Not modelled
PlanetSim	Discrete-event simulator; uses Common API; distinct separation of services and overlay	Design and API throughly documented; detailed tutorial	100,000 nodes	No mechanism to gather statistics, but visualiser is available	Limited simulation of underlying network, but BRITE information could be used for more detail
Neurogrid	Discrete-event for unstructured networks, can be modified for use with structured networks.	Extensive documentation on web	300,000 nodes claimed, but replication fails due to thread limits	For pre-determined variables, but code would have to be modified for others	Not modelled

Table 2: Properties of surveyed simulators.

Simulator	Count of papers using
None	146
Unspecified	71
Custom	43
NS-2	8
Chord (SFS)	7
Javasim	2
Peersim	2
Aurora	1
CSIM 19	1
Modelnet	1
Nab	1
Narses	1
Neurogrid	1
P2PSim	1
SOSS	1

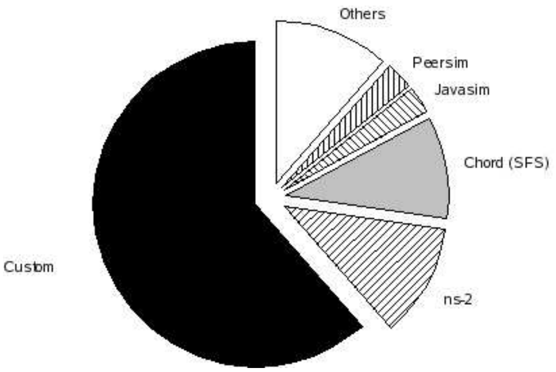

Figure 1: Quantitative survey on the use of simulators

is concerned with the application layer. As NS-2 models the network, link and physical layers in some considerable detail, its scalability, an important property for P2P network simulation, is limited. Further investigation showed that of the 8 papers using NS-2, four papers are concerned with Mobile Ad-Hoc Networks (MANETs) [7, 11–13]. This is to be expected as when conducting MANETs routing research, it is the network layer that is under testing [1] and NS-2 is a popular tool in this research area [6]. In the Geographical Hash Table paper [7], justification is given of the use of NS-2 in addition to the simulation model and results. Of the remaining four, two papers investigated streaming over P2P networks [8, 9], one discussed P2P incentives and disincentives [3] and the final paper discussed the implementation of [5]. Two papers simulated a maximum of 1000 nodes, significantly less than what could be expected in actual P2P systems and less than some of the scalability claims made by Neurogrid and PeerSim, 300000 and 10^6 nodes respectively.

Given the issues with simulators surveyed in section 2, there is clearly some work to be done if the surveyed simulator packages are to achieve acceptance within the research community. Our survey of simulator usage in the area of P2P shows that for the 141 research papers where simulation was used only four of the nine simulators surveyed given in table 2 were used, and this amounted to a total of 5 papers. Two of the papers, using PeerSim, were authored by researchers who developed the simulator. One paper using Narses originated from the same University as the simulator. There was no simulator omitted from our survey which was found to be popular. It is not possible to conclude with certainty that the weaknesses found in the surveyed simulators are responsible for their unpopularity without contacting all the authors of the surveyed papers to enquire about their knowledge regarding the existence of these tools. However, it should not be a surprise if shown to be the case.

4. CONCLUSION

All of the P2P simulators surveyed have functionality missing which we believe is of importance. While poor documentation is a hindrance, it is a problem that can be overcome, but it is entirely unacceptable that many of the simulators have no mechanism to allow a user to gather statistics of a simulation run. We believe that the poor state of existing P2P simulators is the reason that much published research makes use of custom built simulators. Surprisingly, 10% of papers making use of simulation from our survey used NS-2, a tool that is often inappropriate for simulating P2P networks as it lacks scalability due to its detailed modelling of the lower network layers that are often of little interest to P2P researchers. With custom simulators being so popular, it complicates the task of validating research and reproducing results, as these simulators are often not released publicly. Even if they are available comparing similar work, such as overlay algorithms, can be complicated.

Given the current state of simulation use in P2P, we believe that there is a need for a P2P simulator that meets the requirements of P2P researchers. We hope in writing this paper to provoke discussion and to help build a consensus on the common platform for P2P research, in the way that researchers in other areas have converged upon NS-2.

5. REFERENCES

[1] T. R. Andel and A. Yasinac. On the credibility of manet simulations. *Computer*, 39(7):48–54, 2006.

[2] F. Dabek, B. Zhao, P. Druschel, J. Kubiatowicz, and I. Stoica. Towards a common API for structured peer-to-peer overlays. In *Proc. IPTPS*, volume 58. Springer, 2003.

[3] M. Feldman, K. Lai, J. Chaung, and I. Stoica. Quantifying disincentives in peer-to-peer networks. In *1st Workshop on Economics of Peer-to-Peer Systems*, June 2003.

[4] P. García, C. Pairot, R. Mondéjar, J. Pujol, H. Tejedor, and R. Rallo. Planetsim: A new overlay network simulation framework. In *Proc. 4th Intl. Workshop on Software Engineering and Middleware (SEM 2004)*, volume LNCS 3437, pages 123–137, Linz, Austria, 2005. Springer.

[5] A. Ito, T. Ohta, K. Mitsukawa, and Y. Kakuda. An adaptive index allocation scheme for reliable data retrieval and provision in peer-to-peer networks. In *SAC '06: Proc. 2006 ACM symposium on Applied computing*, pages 697–704, New York, NY, USA, 2006. ACM.

[6] S. Kurkowski, T. Camp, and M. Colagrosso. Manet simulation studies: the incredibles. *SIGMOBILE Mob. Comput. Commun. Rev.*, 9(4):50–61, 2005.

[7] S. Ratnasamy, B. Karp, S. Shenker, D. Estrin, R. Govindan, L. Yin, and F. Yu. Data-centric storage in sensornets with ght, a geographic hash table. *Mobile Networks and Applications*, 8(4):427–442, 2003.

[8] R. Rejaie and A. Ortega. Pals: peer-to-peer adaptive layered streaming. In *NOSSDAV '03: Proc. 13th intl. workshop on Network and operating systems support for digital audio and video*, pages 153–161, New York, NY, USA, 2003. ACM.

[9] E. Setton, J. Noh, and B. Girod. Rate-distortion optimized video peer-to-peer multicast streaming. In *P2PMMS'05: Proc. ACM workshop on Advances in peer-to-peer multimedia streaming*, pages 39–48, New York, NY, USA, 2005. ACM.

[10] I. Stoica, R. Morris, D. Liben-Nowell, D. R. Karger, M. F. Kaashoek, F. Dabek, and H. Balakrishnan. Chord: a scalable peer-to-peer lookup protocol for internet applications. *IEEE/ACM Trans. Netw.*, 11(1):17–32, 2003.

[11] C.-Y. Tseng, P. Balasubramanyam, C. Ko, R. Limprasittiporn, J. Rowe, and K. Levitt. A specification-based intrusion detection system for aodv. In *SASN '03: Proc. 1st ACM workshop on Security of ad hoc and sensor networks*, pages 125–134, New York, NY, USA, 2003. ACM.

[12] J. van der Merwe, D. Dawoud, and S. McDonald. Fully self-organized peer-to-peer key management for mobile ad hoc networks. In *WiSe '05: Proc. 4th ACM workshop on Wireless security*, pages 21–30, New York, NY, USA, 2005. ACM.

[13] P. Vellore, P. Gillard, and R. Venkatesan. Delivery analysis of multicasting in bittorrent enabled ad hoc network (mbean) routing. In *IWCMC '06: Proc. 2006 Intl. Conference on Communications and Mobile Computing*, pages 1135–1140, New York, NY, USA, 2006. ACM.