

# JXTA-Sim: Simulating the JXTA Search Algorithm

Sandra Garcia Esparza and René Meier

**Abstract** JXTA is a set of platform-independent, open source peer-to-peer protocols that has become popular for building services and applications based on peer-to-peer overlays. Existing work towards evaluating one of JXTA's core protocols, the JXTA search algorithm, has mainly focused on testbed-based experiments. Although such evaluation configurations offer accurate results based on real deployments, scaling experiments to a large number of distributed hosts is difficult and often prohibitively expensive. Furthermore, repeating experiments using different configuration parameters might yield distorted results due to the uncontrolled nature of testbeds. Simulators offer an alternative to testbeds for evaluating large-scale applications in a controlled environment. This paper presents JXTA-Sim, a simulator for studying and evaluating the JXTA search algorithm. JXTA-Sim enables researchers to study the behavior of JXTA's search algorithm using different configuration parameters and ultimately, to test JXTA-based peer-to-peer applications.

**Key words:** P2P computing, search algorithm, simulating JXTA applications

---

Sandra Garcia Esparza

CLARITY: Centre for Sensor Web Technologies, University College Dublin, Belfield, Dublin 4, Ireland, e-mail: [sandra.garcia-esparza@ucd.ie](mailto:sandra.garcia-esparza@ucd.ie)

René Meier

Distributed Systems Group, School of Computer Science and Statistics, Trinity College Dublin, Dublin 2, Ireland, e-mail: [rmeier@cs.tcd.ie](mailto:rmeier@cs.tcd.ie)

## 1 Introduction

Over the last years, features of peer-to-peer (P2P) networks, such as, self- organization, scalability and robustness, have captured the interest of researchers. Project JXTA[1] is an open-source platform originally conceived by Sun Microsystems Inc. in 2001 with the goal to standardize a set of protocols for building P2P applications. JXTA consists of a set of 6 language and platform independent protocols that allow developers to build interoperable and scalable P2P services and applications.

P2P networks are often used in applications where peers need to share information with other peers across a possibly highly-distributed network topology. Such peers are said to publish information, also called resources, and to discover resources respectively. This mechanism is key to any P2P application, needs to be efficient and scalable, and is referred to as a search algorithm for P2P networks. There are different algorithms for discovering resources depending on the structure of the underlying network. In structured networks, the graph of the network follows a particular structure and resources are located using the information about this structure. On the other hand, unstructured networks do not follow a particular structure and resources are randomly placed in the topology. The JXTA search algorithm supports a hybrid of these two approaches to locate resources. It follows a structured approach when the network is stable and an unstructured approach when the network is unstable, i.e., is exposed to frequent topology changes due to peers joining and leaving. Although JXTA has a widespread use in research and industry, its performance and scalability capabilities still remain unclear. Hence, we need to investigate ways to evaluate the JXTA platform. JXTA consists of a big and complex architecture and, as a result, an evaluation may assess individual JXTA components separately.

Past evaluations [4, 3] have studied different aspects of the platform by using testbeds. However, these evaluations consider only a small number of peers due to the high cost of deploying testbeds for large scenarios. Simulations offer a good alternative to testbeds for evaluating an algorithm in such scenarios. Simulators can model an algorithm in networks with a large number of nodes without the costs testbeds introduce. Furthermore, they offer a level of control and easy modification of the simulated algorithm since they are typically deployed on relatively few nodes. This paper presents JXTA-Sim, a simulator for studying and evaluating the JXTA search algorithm. JXTA-Sim has two main goals. The first goal is to allow researchers to understand, study and evaluate the JXTA search algorithm. The second is to allow P2P researchers to test their applications on top of JXTA's overlay network. This will enable them to evaluate the performance of the algorithm (and ultimately their application) and compare it with other overlay algorithms. JXTA-Sim is based on PlanetSim[8], a P2P framework for implementing P2P overlay networks. This paper is organized as follows. Section 2 introduces JXTA's search algorithm. Section 3 discusses related work in the field of simulation and JXTA evaluation.

---

©Springer-Verlag, 2010. This is the author's version of the work. The original publication is available at <http://www.springer.com/engineering/book/978-3-642-14882-8>. It is posted here by permission of Springer-Verlag for your personal use. Not for redistribution. The definitive version was published in *Advances in Intelligent and Soft Computing*, Volume 79/2010.

Section 4 presents the design and the architecture of JXTA-Sim. Section 5 presents the evaluation experiments we have performed with JXTA-Sim. Finally, Section 6 concludes this paper and discusses issues that remain for future research.

## 2 The JXTA Search Algorithm

The JXTA project does not specify how to search for advertisements (i.e. XML documents describing resources) but provides a generic resolver protocol framework with a default policy that can be overwritten [1]. JXTA combines the advantages of structured and unstructured networks [5] by using a hybrid approach that combines the use of a *loosely-consistent Distributed Hash Table (DHT)* with a *limited-range rendezvous walker* [6]. The DHT is used for networks with low peer churn rates while the rendezvous walker is used for networks that change frequently.

There are two type of peers (or nodes) involved in the JXTA search algorithm. *Edge peers* are the basic peers and are responsible for publishing and discovering advertisements. *Rendezvous peers*, on the other hand, have the ability to forward discovery requests and in order to do so, they store the indexes of the advertisements. Edge peers publish and discover advertisements by pushing their indexes to their rendezvous peers. They also store advertisements in their local cache. Rendezvous peers form a rendezvous network where they propagate the queries and responses generated by peers. To keep the connections with other rendezvous peers, each of them maintains a Rendezvous Peer View (RPV), which is an ordered list of IDs of the known rendezvous peers in the group. This list does not need to be globally consistent. A peer may not be aware of the existence of other rendezvous peers and RPVs may be different on different peers.

After storing an index locally, rendezvous peers compute a hash function to identify the rendezvous peer from the RPV that has a replica of the advertisement's index. The *replication distance* is a parameter that specifies the number of peers in each direction of the rendezvous peer (up and down) that holds a copy. When an edge peer needs to retrieve an advertisement, it forwards the request to its rendezvous peer. This will execute the same hash function as the one used for the publication, to identify the rendezvous peer from its RPV that stores the index (and therefore knows the edge peer that stores the advertisement). Because the same function is executed, if the network is stable and all rendezvous peers share the same RPV, the chosen rendezvous will be the one storing the index of the advertisement. On the other hand, if it is unstable and rendezvous peers have not been able to keep their tables updated, two things can happen. If the network has not experienced many changes, we may find a replica of the index in a rendezvous peer near the original one and continue the retrieval from there. However, if the index is not found in any of the replica peers, then a walker needs to be started in both up and down directions in the vicinity of the rendezvous peer chosen from the RPV. This walking continues until the advertisement is found or until a maximum number of hops are reached.

### 3 Related Work

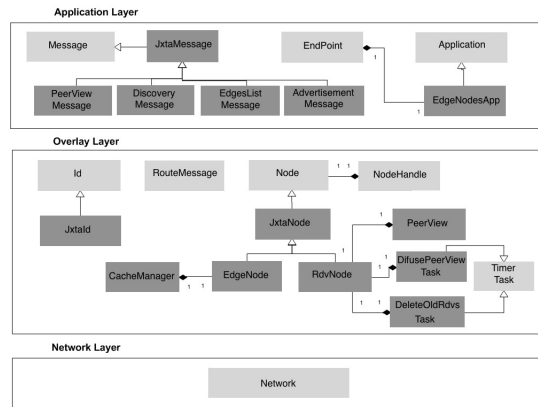
Past evaluations [4, 3] have studied different aspects of the JXTA platform by using testbeds. However, these evaluations consider only a small number of peers (up to 32). In [2] an experimental evaluation of the scalability of the JXTA search is presented based on using a testbed consisting of 5000 CPUs. This work focuses on measuring the time it takes for the rendezvous protocol to make the peerviews stable (all rendezvous have the same view) and the time it takes to retrieve an advertisement. These experiments concluded that with more than 45 rendezvous peers, the rendezvous protocol fails to make the peer views consistent. In these experiments, the rendezvous algorithm has been studied by installing the whole JXTA platform on every node. In contrast, our focus is on studying different aspects of the search algorithm in a controlled manner. We aim to provide researchers with a tool for assessing the algorithm in application-specific scenarios while allowing for changes to the set of configuration parameters. By using a simulator, we can facilitate changes to the algorithm and re-evaluate scenarios that may scale to a considerable number of nodes. [7] discusses desirable characteristics for P2P overlay simulators. PlanetSim [8] is a discrete, event-based overlay simulator that supports implementation and validation of overlay algorithms and which provides most of those characteristics by being object-oriented, extensible and customizable. As a result, PlanetSim has been chosen as the platform for building JXTA-Sim.

### 4 JXTA-Sim Design and Architecture

JXTA-Sim builds on PlanetSim extending the three architecture layers: application, overlay and network. In fig. 1 depicts the architecture of JXTA-Sim. Dark colored components represent JXTA-Sim extensions that build upon the light colored PlanetSim components.

**Application Layer.** This layer allows applications to be assessed over different overlay schemes. JXTA-Sim defines two classes in this layer. The *EdgeNodesApplication* component represents an application that runs on an edge node. This class contains the application to be tested. *Application* can also be extended to support multiple applications concurrently. The *JxtaMessage* class has been extended from *Message* to support the different types of messages sent by JXTA peers.

**Overlay Layer.** This layer represents the virtual layer on top of the physical layer. As PlanetSim allows for only *Node* type, JXTA-Sim defines an intermediate *JXTA Node* class that can be of two types: *EdgNode* or *RendezvousNode*. Each node contains the methods according to their behavior. While *EdgeNode* contains methods, including, "connect to rdv", "publish adv" and "find adv", *RendezvousNode* has methods such as "store adv", "retrieve adv", "update peer view" and "diffuse peer view". In addition, *EdgeNode* has a structure to store the published advertisements, which is the *Cache Manager*. *RendezvousNode* supports a structure to maintain the connections to other rendezvous peers, called the *Rendezvous Peer View (RPV)*.



**Fig. 1** The architecture of JXTA-Sim

*DiffusePeerView* and *DeleteOldRendezvous* are periodic tasks used for propagating RPVs and for removing expired entries in the RPV respectively.

**Network Layer.** This layer represents the physical network where nodes communicate with each other by sending *Route Messages*. Currently it does not support some networking aspects, such as, latencies or node mobility. However, future versions of the network layer may simulate such characteristics. For example, in [8], the authors of PlanetSim propose an extension that supports latencies.

### 4.1 Configuration Parameters and Gathering Results

Currently, JXTA does not support parameters, for example, for changing the replication distance or for adjusting the maximum number of hops. We believe that allowing users to modify these parameters may improve the effectiveness of the search algorithm for certain application scenarios; for instance, it might be beneficial to have a higher replication distance in networks with high churn compared to those with a low churn. The main parameters that can be configured in JXTA-Sim are the following: number of simulation steps, max. number of edge peers connected to a rendezvous, max. number of walker hops, size of the RPV, expiration time for a RPV entry, size of edge peers cache, replication distance, number of published advertisements during the simulation, number of advertisements to be discovered and number of rendezvous that will join and leave during simulation.

## 4.2 Using JXTA-Sim

JXTA-Sim uses a configuration file for defining both the simulation and the search algorithm parameters. Furthermore, an event file contains a list of time-event pairs representing the events that occur during the simulation. Alternatively, users can write a program that calls the desired events at appropriate times.

JXTA-Sim provides alternative means for capturing the results of a simulation. Using PlanetSim, JXTA-Sim allows creating a graph of the network at any time during a simulation. It also allows to see the different types of messages sent at every step of the simulation and introduces new aspects to show the number of hops of searches. Finally, JXTA-Sim captures information about the connections between peers. This enables users to observe how the edge peers connect to certain rendezvous peers and to explore a peer's RPV.

## 5 Evaluation

### 5.1 Assessing JXTA-Sim

In order to assess the simulation of the search algorithm, we perform a set of experiments that test the features of the algorithm, including adding peers, disconnecting peers, publishing advertisements and locating advertisements. These experiments generate traces and ultimately graphs for studying and verifying the behavior of the algorithm. We capture the sequence of events, such as peers joining and leaving as well as the messages sent between peers as a result of these events. This visualizes the various JXTA messages and provides a basis for verifying that they match the JXTA algorithm as described above in section 2. Due to space limitations, we only show one diagram of the results from these experiments. Fig. 2 depicts the messages sent in an experiment with 60 simulation steps in a network with 5 peers. In this experiment, a peer leaves at simulation step 28, while another peer publishes an advertisement at step 34 that is discovered by a third peer at step 53. Moreover, the messages captured in these experiments also verify that the DHT table is used (and not the walker) when the network is stable. As a result, these experiments show that our simulation of the search algorithm behaves according to the JXTA specification.

### 5.2 Assessing the JXTA Search Algorithm in JXTA-Sim

After assessing the simulator we now focus on evaluating the JXTA search algorithm. The performance of the algorithm under different configuration parameters is evaluated by means of the percentage of successful searches and the number of hops

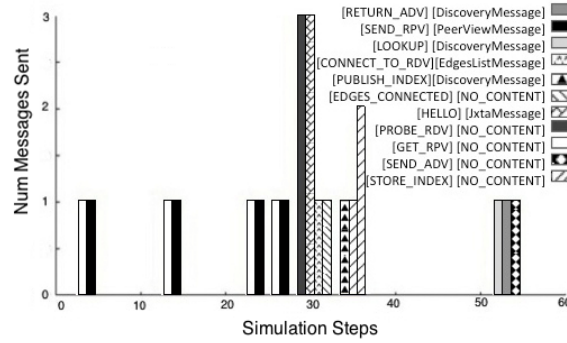
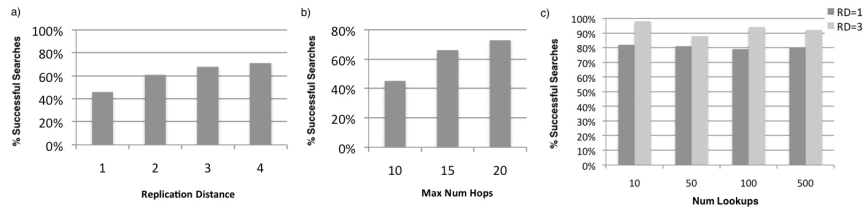


Fig. 2 Messages sent in a network with 5 rendezvous peers.

required to locate an advertisement. The initial scenario studies the performance on a static network, where peers do not join or leave the group, and uses the following parameters: 50 rendezvous peers, 150 edge peers, replication distance (RD) = 1, and maximum number of hops = 5. This scenario was repeated for lookup searches ranging from 10 to 500 along 1000 simulation steps and found that around 95% of the advertisements were located in each setting. The refined scenario studies how churn affects the performance of the search and, most importantly, how to alleviate the effects of churn by modifying configuration parameters. Fig. 3c, shows that churn reduces the number of successful searches and that increasing RD to 3 alleviates this effect. A churn of 20% (20% of the rendezvous peers leave and 20% join during the simulation) reduces successful searches to about 80% while increasing RD increases them to above 90%.

A second scenario assesses the impact of RD and the number of hops (NH) in a network with 4000 nodes (1000 rendezvous peers and 3000 edge peers), a churn of 10%, RD= 1 and maximum number of hops = 10. Fig. 3a shows that increasing RD can improve successful searches from 45% to 73%. This again demonstrates that increasing RD may compensate the effect of churn. Fig. 3b shows that increasing NH can also improve successful searches. It can be observed that for RD=1, NH needs to be doubled to achieve the same performance increase as for a RD=4. Doubling the number of hops implies that the walker performs up to 20 hops, which increases the network traffic considerably. Hence, increasing the replication distance is likely the preferable option for compensating for churn.

These experiments demonstrate that JXTA-Sim can support a significant number of peers and that being able to modify configuration parameters can help the JXTA search algorithm to adapt to the different requirements of application scenarios. While these parameters are currently fixed to a given default value, we argue that the JXTA platform would benefit from such an extension as it will likely improve overall application performance for varying network sizes and overlay churns.



**Fig. 3** Percentage of successful searches under different Replication Distance (a,c) and Max Num Hops (b) values.

## 6 Conclusions and Future Work

This paper has presented JXTA-Sim, a simulator for understanding, studying and evaluating the behavior and performance of the JXTA search algorithm. JXTA-Sim provides a means for researchers to test peer-to-peer applications on top of the JXTA virtual network and to evaluate (and compare) their applications under varying networking conditions, environment constraints, and user requirements. Our assessment of JXTA-Sim has demonstrated that the search algorithm behaves according to the protocols defined in Project JXTA and that a considerable number of peers can be simulated. Perhaps most importantly, we have found that a future version of JXTA might allow for the application-specific configuration of parameters that currently support default values only. This can improve the performance of certain JXTA applications by increasing the availability of network resources and reducing the traffic in the overlay network.

JXTA-Sim currently focuses on simulating the JXTA search algorithm and, as a result, the JXTA discovery and routing protocols are presently supported only. Future work will extend JXTA-Sim to feature further JXTA protocols. The Pipe Binding Protocol, which implements virtual communication channels, and until now has only been evaluated using testbeds, might be supported, with simulated latencies. The Peer Endpoint Protocol might be supported to evaluate JXTA's concept of relay peers used for indirect routing. Finally, supporting peer groups would allow for experiments with multiple search scopes where peers belong to one or more groups.

### Acknowledgments.

We would like to thank the PlanetSim team for their help and support during the development of JXTA-Sim.



## References

1. B. T. Ahkil, B. Traversat, A. Arora, M. Abdelaziz, M. Duigou, C. Haywood, J. Hugly, E. Pouyoul, and B. Yeager. Project jxta 2.0 super-peer virtual network. Technical report, Sun Microsystems, Inc., May 2003.
2. G. Antoniu, L. Cudennec, M. Jan, and M. Duigou. Performance scalability of the jxta p2p framework. In *Parallel and Distributed Processing Symposium, 2007. IPDPS 2007. IEEE International*, March 2007.
3. B. T. E. Halepovic, R. Deters. Performance evaluation of jxta rendezvous. In *International Symposium on Distributed Objects and Applications (DOA '04)*, Agia Napa, Cyprus, Oct 2004. Springer Verlag.
4. E. Halepovic and R. Deters. Jxta performance study. In *Communications, Computers and signal Processing, 2003. PACRIM. 2003 IEEE Pacific Rim Conference on*, Aug. 2003.
5. E. Lua, J. Crowcroft, M. Pias, R. Sharma, and S. Lim. A survey and comparison of peer-to-peer overlay network schemes. *IEEE Communications Surveys and Tutorials*, 7:72–93, 2005.
6. B. T. Mohamed, M. Abdelaziz, and E. Pouyoul. Project jxta: A loosely-consistent dht rendezvous walker, 2003.
7. S. Naicken, B. Livingston, A. Basu, S. Rodhetbhai, I. Wakeman, and D. Chalmers. The state of peer-to-peer simulators and simulations. *SIGCOMM Comput. Commun. Rev.*, April 2007.
8. J. Pujol-Ahulló, P. García-López, M. Sánchez-Artigas, and M. Arrufat-Arias. An extensible simulation tool for overlay networks and services. In *SAC '09: Proceedings of the 2009 ACM symposium on Applied Computing*, New York, NY, USA, 2009. ACM.