SPIDER: A Bio-Inspired Structured Peer-to-Peer Overlay for Data Dissemination

Bogdan Mocanu, Florin Pop, Alexandra Mihaita (Mocanu), Ciprian Dobre, Valentin Cristea
Computer Science Department, University Politehnica of Bucharest, Romania
Email: bogdan.mocanu@hpc.pub.ro, florin.pop@cs.pub.ro, alexandra.mocanu@hpc.pub.ro, ciprian.dobre@cs.pub.ro, valentin.cristea@cs.pub.ro

***Abstract*—There has been an exponential growth of the multimedia data shared through Internet, and managing this amount of information has become a great provocation. Classical Peer-to-peer systems has proven their great contribution in data distribution over the Internet. Bio-inspired Peer-to-Peer overlays have become a challenging research topic in the computer science field since the beginning of this decade. Starting from ant colonies to bee swarms or fish banks all this natural phenomenon has become an inspiring source for researchers. Data dissemination for crowd management is also a very actual research topic in the computer science community. In this paper we present SPIDER, a new Peer-to-Peer structured overlay based on a natural structure of a spider web and some possible uses cases for crowd management using data dissemination. The main goal of this paper is to offer a brief description of the main operations in this kind of overlay from joining to leaving the overlay. The performance of this overlay is presented as theoretical results.**

***Keywords*—Peer-to-Peer Networks, Adaptive Overlay, Structured Overlay, Bio-inspired Models, Data Dissemination**

I. INTRODUCTION

In the recent years, there has been noticed a highly increased volume of multimedia data over the Internet. The majority of this content relies on unreliable sources interconnected through different type of networks and organized in many forms or shapes. The management and dissemination of this amount of data has become a significant issue in the computer science research community. Taking in consideration the fact hat data there are no central servers, or single points of failure, the most appropriate topology for this king of networks are Peer-to-Peer systems. The usage of Peer-to-Peer networks offers a great advantage from classical client-server networks through complexity, heterogeneity, mobility and dynamic behavior. Based on the recent development in the infrastructure of the interconnected devices through Internet and the concept of the Internet of Things Peer-to-Peer overlays networks has shown an increasing interest in the computer science research community.

Considering the heterogeneity of mobile computing, building a Peer-to-Peer overlay on this kind of infrastructure implies many challenges. The most significant is the fact that mobile peers can join or leave the network any time without making any announce before. This fact is caused by short battery lifetime, week Service Provider signal strength or even leaving the system on purpose. In order to face such challenges researchers have developed a new concept of Peer-to-Peer overlays inspired from natural phenomena. This is called self-adaptive Peer-to-Peer Overlays.

The aim of Peer-to-Peer systems is to sustain large-scale systems [1] based on self-organization and decentralized management, thus there must be considered four assumptions. The amount of nodes that must be connected in a Peer-to-Peer system is always around thousands or millions. Traditional systems are based on a central server with represent a single point of failure. The main problem of traditional large scale systems are frequently bottlenecks. This problem is solved in Peer-to-Peer because it does not imply any centralized peer of single point of failure. Another important assumption is the communication cost, which is very important in local area networks. Due to the fact that the data types managed in Peer-to-Peer systems usually includes multimedia content, or Web content, query paradigms is retrieval-oriented.

Data management in Peer-to-Peer systems is realized in a top layer of the Peer-to-Peer overlay called data management layer. This is realized in an independent manner by separating the physical data in two flavors. The first one is based on traditional indexing structures such as has tables, binary trees and table scans. There are similarities between traditional database systems and structured Peer-to-Peer overlays. The second one called isolations from network dynamics handle such churn, network topology and performance.

There are two main types of Peer-to-Peer data management systems: homogeneous and heterogeneous [1]. On one hand, in homogenous systems the peers manages one database that is hosted in the distributed system. While on the other hand, in heterogeneous Peer-to-Peer systems, peers manages different databases witch are distributed in the network.

In this paper we present a bio-inspired approach [11] for a structured Peer-to-Peer overlay based on the spider web. The rest of the paper is structured in 4 sections. In Section II is presented a state of the art of the Peer-to-Peer systems with a focus on bio-inspired overlays. In Section III is presented a general overview of SPIDER overlay and the main operations in this kind of systems. Section IV concludes our work with critical comparison of the SPIDER overlay with Chord and Honeycomb overlays.

II. RELATED WORD

From the beginning of the year 2000 a new type of networks has emerged based on different paradigm than the traditional client-server. The begging of the 2000 has coincide with the development of broadband Internet infrastructure, thus Internet has become widely available. As a consequence, computer networks have started to organize themselves in overlays structures in order to collect and distribute content over the Internet through TCP and HTTP protocols in a fully decentralized manner without any central server or single point of failure. Structured Peer-to-Peer overlays [2] represent a type of network organization type where date is distributed in a deterministic manner, not randomly. The access of a certain data is realized in a finite number of steps based on an algorithm. The main advantage of structured Peer-to-Peer overlays consists of high performance in terms of resource discovery and resource access.

In the last years there has been shown a great interest for researchers for nature-inspired design of self-adaptive Peer-to-Peer overlays. Modeling a natural phenomenon and implement a distributed computing algorithm on its model is a non-trivial task. Furthermore, we present several bio-inspired protocol for Peer-to-Peer systems. These models were translated from a mathematical model of a natural phenomenon into a Peer-to-Peer protocol with a predictable behavior.

Bio-inspired solutions have shown a better efficiency for solutions in the domain of computer networks [3]. Solutions based on swarm intelligence, namely based on the collective behavior of ant colonies or bees, have validated and guaranteed availability due to the distributed intelligence and the reduced communication costs.

In paper [4] is presented a solution for for grid and cloud computing infrastructures. The construction of this overlay is based on swarm intelligence and ant colonies, using multiple independent mobile agents. The authors of [5] presented an efficient algorithm for resource discovery on mobile ad-hoc networks. This overlay is inspired from the foraging behavior of honey bees. Due to this fact it, is very lightweight in terms of exchanged messages. In paper [6] is presented a similar overlay to Self-Chord [1]. The topology of the underlay remains identical to Self-Chord, while the identifiers of the resources are rearranged by ant-inspired mobile agents to improve the performance of resource discovery.

Besides this state of the art we propose a distributed fault-tolerant and a high-availability solution for data dissemination in large sale distributed systems.

III. SPIDER PEER-TO-PEER OVERLAY

A SPIDER Peer-to-Peer overlay consists of a Super Peer that is positioned in the center of the structure. The Super Peer it is not a single point of failure due to the fact that it can be replaced by any other peer in the system. The main purpose of the Super Peer is make possible the joining of the first nodes in the overlay. We can imagine more functions for the Super Peer, which can be successfully applied in data dissemination techniques.

The construction of the overlay is bio-inspired, creating the network similar with the creation of the spider web. The first are created a fixed number of chains and after that the rings are build over the chains.

The SPIDER overlay consists of a n fixed number of chains labeled from $1 \dots n$ and a dynamical number of k rings depending of the number of peer in the system. If a ring k has n nodes (it is complete), we labeled the nodes from $(k-1)*n+1$ to $k*n$, with $k \geq 1$. Therefore, the structure of the overlay can be presented a two dimension matrix read on columns, where each row represent a chain and each column represents a ring. The graphical representation of SPIDER overlay is presented in Figure 1.

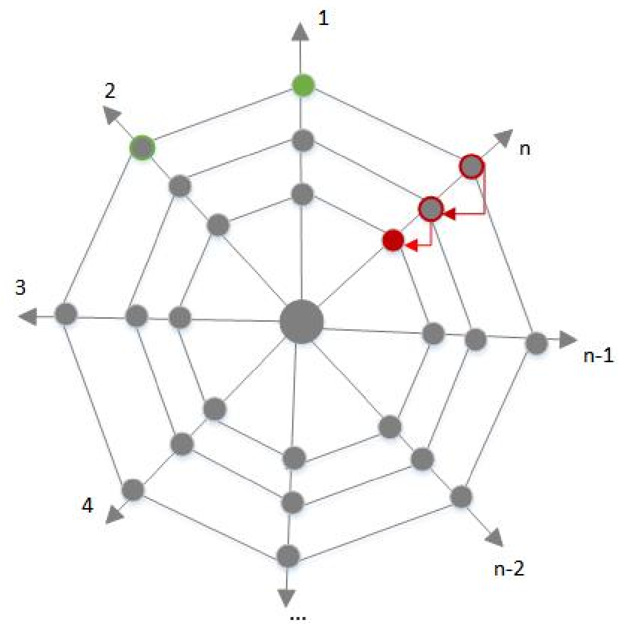

Fig. 1. SPIDER Peer-to-Peer Overlay. This figure presents two scenarios of actions of peers in SPIDER overlay: Joining and Leaving. *Joining*: Considering a new node (solid green node) on chain 1. It can join the system only if it knows an existing node in the system (green border node), and that node has left at least one neighbor position free. *Leaving*: Considering a node that leaves the system (solid red node) on chain n ring 1. When it leaves the system all the nodes on the same chain (red border nodes) advanced one position on the ring. Therefore, the node on chain n ring 2 replaces the left nodes and the node from ring 3 of the same chain will replace se second one. Thus ring 3 will become incomplete, but this does not affect the functionality of the system.

Each node n_i in the SPIDER overlay has maximum 4 neighbors. For instance, a $node[c, r]$, where c is the chain number and r is the ring number, has the following neighbors:

N1: $node[(c\%n), r-1]$ - is the below neighbor;

N2: $node[c, r+1]$ - is the upper neighbor;

N3: $node[(c+1)\%n, r]$ - is the right neighbor;

N4: $node[(c-1)\%n, r]$ - is the left neighbor.

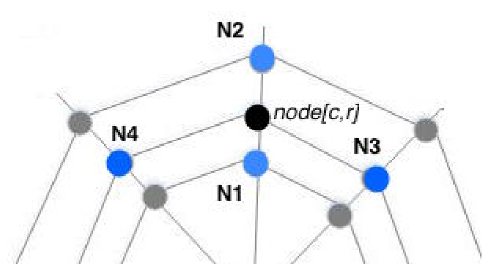

Fig. 2. Node neighbors. The neighbors for $node[c, r]$ are marked with blue.

A. Joining

When a peer wants to join the system it must know at least one peer that is already in the overlay. The existing peer in the system will check if it has occupied al 4 neighbors. If it has a right, or a left neighbor is available it places the new peer on that position and update the neighbor table, or the only position left is the upper neighbor, it places the new peer there. After this operation all the neighbors tables are updated.

As it can be seen each node in the overlay will always have at least one neighbor when joining the overlay that is the bellow neighbors.

B. Naming

The Super Peer id is 0 and it belongs to all chains in the network, and there is no ring containing it. Then, for each node we consider the chain c, with $1 \leq c \leq n$ and the ring r, with $r \geq 1$. The node id is computed as follow:

$$n_i = node[c, r] = n * (r - 1) + c.$$

The total number of nodes per chain is r and the total number of nodes in a complete network is $N = r * n$. If the last ring is incomplete:

$$(r - 1) * n + 1 \leq N \leq r * n.$$

The total number of links for a ring is equal with n, then we have maxim n links between the current ring and the upper ring. So, the total number of links in the network follow the inequality:

$$n * r + 1 \leq L \leq n * (r + 1).$$

For a node n_i we can compute its location as follow:

$$chain(n_i) = (n_i - 1)\%n + 1$$

and

$$ring(n_i) = \frac{n_i - chain(n_i)}{n} + 1.$$

Now, for node monitoring and data dissemination it is important to calculate the minimum number of hops between two nodes. Fist, we introduce a function that compute the number of hops between two nodes from the same ring:

- if $|chain(n_i) - chain(n_j)| \leq \left[\frac{n}{2}\right]$ then:

  $$ChainDIF(n_i, n_j) = |chain(n_i) - chain(n_j)|;$$

- $|chain(n_i) - chain(n_j)| > \left[\frac{n}{2}\right]$ then:

  $$ChainDIF(n_i, n_j) = n - |chain(n_i) - chain(n_j)|;$$

Second, we compute the difference between rings containing n_i and n_j as follow:

$$RingDIF(n_i, n_j) = \max\{ring(n_i), ring(n_j)\} - \min\{ring(n_i), ring(n_j)\},$$

And, after some calculations:

$$RingDIF(n_i, n_j) = |ring(n_i) - ring(n_j)|.$$

Now, the total number of hops between node n_i and node n_j are:

$$hops(n_i, n_j) = ChainDIF(n_i, n_j) + RingDIF(n_i, n_j).$$

The complexity of data dissemination for a node to the all nodes in the network is $O(L) = O(n * r)$.

Similar results were obtained for epidemic-style global load monitoring in large-scale overlay networks [9] and for decentralized trust management in Peer-to-Peer Systems [10] were the hops between nodes are used as metric in the collection process of messages between nodes. This approach is applicable also to evaluation of intragroup optimistic data replication in P2P groupware systems [12].

C. Look-up

Look-ups in the SPIDER Overlay are realized by queering each available neighbor for the desired resource piece. If the searched resource is available on one of the neighbors, it is send immediately. If the resource is not present at any neighbor, the look-up process begins again starting from each neighbor.

Another way of making look-up in the SPIDER overlay is a hierarchical manner. Considering all the peers on a chain having common interests. Therefore, resources can be disseminated only between the peers on the same chain by sending then from the lowest ring to the upper ring.

If a resource must be shared between chains, it is possible only if it is disseminated on the left or right neighbor on the same ring. Thus upper ring peers cannot share resources with other peers from different chain lower rings.

D. Leaving

When a peer leaves the Peer-to-Peer overlay, its place is taken easily by the upper neighbors, because it is unlikely suitable to break an existing ring. If there are none upper neighbors, the system will wait until a new peer joins the network. Taking in consideration the same algorithm, the Super Peer is replaced in the same manner if it leaves the system.

IV. COMPARISON BETWEEN DIFFERENT OVERLAYS

Table I presents a comparison between Chord, Self-Chord, Honeycomb and Spider overlays concerning 5 metrics. Redundancy represent the property of system to store the same information on two different nodes in order to assure availability 24/7 in case one the nodes leaves the system. Dynamicity represents the property of a system to auto adapt.

The Security property means the possibility of establishing a secure communication channel between nodes. Lookup resents the number of queries in the system until a resource is located. Queries column indicated the type of queries the system can handle.

As it can be seen in table, every overlay does not support redundancy features. Concerning dynamicity, Honeycomb

TABLE I. COMPARISON BETWEEN DIFFERENT OVERLAYS

Overlay	Redundancy	Dynamicity	Security	Lookup	Queries
Chord	NO	NO	NO	$\log N$	Standard queries
Self-Chord	NO	NO	NO	$\log N$	Any type of queries
Honey comb	NO	YES	YES	$\sqrt{N}$	Any type of queries
SPIDER	NO	YES	YES	$\sqrt{N}$	Any type of queries

overlay and spider overlay has this property. Also security features can be implemented on both Honeycomb and Spider overlay.

The lookups realized in Chord and Self-chord are the same $O(\log N)$ while on honeycomb and SPIDER is $O(\sqrt{N})$. The query types supported by Self-Chord, Honeycomb and SPIDER are any type of queries while Chord overlay supports only standard queries.

V. CONCLUSION

In this paper we presented an overview of a new bio-inspired Peer-to-Peer network overlay. The main goal was to emphasize the possible actions in this overlay: joining, leaving and lookups. The total number of links and the number of hops between two nodes were estimated in order to find the performance of data dissemination process in such type of overlay. A critical comparison of the SPIDER overlay with Honeycomb [7] and Chord [8] overlays was presented.

ACKNOWLEDGMENT

The work has been funded by the Sectoral Operational Programme Human Resources Development 2007-2013 of the Ministry of European Funds through the Financial Agreement POSDRU/187/1.5/S/155420 (PROSCIENCE). The research presented in this paper is supported also the project *DataWay*: Real-time Data Processing Platform for Smart Cities: Making sense of Big Data, PN-II-RU-TE-2014-4-2731.

We would like to thank the reviewers for their time and expertise, constructive comments and valuable insight.

REFERENCES

[1] Karl A., (2011) Peer-to-peer Data Management. Morgan & Claypool Publishers

[2] Meshkova, E., Riihijrvi, J., Petrova, M., & Mhnen, P. (2008). A survey on resource discovery mechanisms, peer-to-peer and service discovery frameworks. Computer networks, 52(11), 2097-2128.

[3] Bongard, J. (2009). Biologically inspired computing. Computer, (4), 95-98.

[4] Forestiero, A., Leonardi, E., Mastroianni, C., Meo, M. (2010). Self-chord: a bio-inspired P2P framework for self-organizing distributed systems. Networking, IEEE/ACM Transactions on, 18(5), 1651-1664.

[5] Dhurandher, S. K., Misra, S., Pruthi, P., Singhal, S., Aggarwal, S., Woungang, I. (2011). Using bee algorithm for peer-to-peer file searching in mobile ad hoc networks. Journal of Network and Computer Applications, 34(5), 1498-1508.

[6] Giordanelli, R., Mastroianni, C., Meo, M. (2012). Bio-inspired P2P systems: The case of multidimensional overlay. ACM Transactions on Autonomous and Adaptive Systems (TAAS), 7(4), 35.

[7] Pop, F., Citoteanu, O. M., Dobre, C., & Cristea, V. (2014, June). Resource Trust Management in Auto-Adaptive Overlay Network for Mobile Cloud Computing. In Parallel and Distributed Computing (ISPDC), 2014 IEEE 13th International Symposium on (pp. 162-169). IEEE.

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

[9] Ghit, B., Pop, F., Cristea, V. (2010, November). Epidemic-style global load monitoring in large-scale overlay networks. In P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC), 2010 International Conference on (pp. 393-398). IEEE.

[10] Visan, A., Pop, F., Cristea, V. (2011, July). Decentralized Trust Management in Peer-to-Peer Systems. In Parallel and Distributed Computing (ISPDC), 2011 10th International Symposium on (pp. 232-239). IEEE.

[11] Ghit, B., Pop, F., Cristea, V. (2011, October). Using bio-inspired models to design peer-to-peer overlays. In P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC), 2011 International Conference on (pp. 248-252). IEEE.

[12] Xhafa, F., Potlog, A. D., Spaho, E., Pop, F., Cristea, V., Barolli, L. (2012). Evaluation of intragroup optimistic data replication in P2P groupware systems. Concurrency and Computation: Practice and Experience.