# Self-adaptive Overlay Networks*

Bogdan-Costel Mocanu, Florin Pop, Ciprian Dobre, Valentin Cristea
*University Politehnica of Bucharest, Romania,*
E-mail: bogdan.mocanu@hpc.pub.ro, florin.pop@cs.pub.ro,
ciprian.dobre@cs.pub.ro, valentin.cristea@cs.pub.ro

November 26, 2015

**Summary**

The highly increased volume of multimedia content from unreliable sources shared through Internet determines a growth in the development of multimedia applications for handling such data from data gathering to data manipulation and data analysis. The efficient management of this type of data must rely on uniform network overlays. One suitable network overlay for managing a high volume of data from unreliable and untrusted sources is the peer to peer overlay (Ghit et al. 2010) because peers might join or leave the system any time. For example, the project SIPTVMON (Wei et al. 2012) presents a secure application layer multicast overlay network for IPTV. The proposed solution for IPTV data assurance is based on the SIP protocol and uses cryptographic algorithms such as: elliptic curves, Diffie Hellman for key exchange and AES encryption for data confidentiality. Another project that focuses on security assurance of distributed architecture overlays is SecP2PSIP (Khan and Hasan 2014). Even though, the P2P is considered insecure, it might be possible for researchers to develop new security protocols for such network overlay. One challenging research topic in this filed might be the creation of an automated secure boot strapping for P2P nodes. Since we face a large variety of solutions for specific applications and platforms, a thorough and systematic analysis of existing solutions for Scheduling models, methods and algorithms used in Big Data processing and storage environments is needed. This chapter presents the state of art of existing solutions and creates an overview of current and near-future trends. The first chapter will focus on existing computing services tailored to sensed environments.

Keywords: Peer-to-Peer, Network overlays, Self-adaptive overlays, Structured overlays, Unstructured overlays, Bio-inspired overlays.

# 1 Introduction

Taking in consideration the high increase of the number of the intelligent devices interconnected such sensors and mobile devices a highly interest topic in the scientific community is trust and security assurance for such type of large scale networks that might be organized in different overlays with respect to different applications constraints.

*The work has been funded by the project *DataWay*: Real-time Data Processing Platform for Smart Cities: Making sense of Big Data, PN-II-RU-TE-2014-4-2731

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. The usage of Peer-to-Peer networks offers a great advantage from classical client-server networks through complexity, heterogeneity, mobility and dynamicity. Considering this criteria, in this paper will be presented a state of the art approach in a chronological manner of the structured and unstructured Peer-to-Peer overlays.

Taking in consideration 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.

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. The usage of Peer-to-Peer networks offers a great advantage from classical client-server networks through complexity, heterogeneity, mobility and dynamicity. Considering this criteria, in this paper will be presented a state of the art approach in a chronological manner of the structured and unstructured Peer-to-Peer overlays.

Another subject that will be presented in this paper is Pervasive computing, also known as Ubiquitous computing represent a challenging research field, because it implies the existence of computing resources anywhere, anytime in a dynamically and adaptive manner with low cost and high level of elasticity and availability. Ubiquitous networks can be created based on a distributed network with a high degree availability and adaptability.

Taking in consideration 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.

On top of the solutions analyzed in this paper the coexistence of multiple Peer-to-Peer overlay is presented. The organization of the peers in several overlays with different structures and the collaboration between them represent a major challenge in this field of research.

# 2 Background/Literature review/Context

## 2.1 Unstructured Peer-to-Peer

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 point of failure.

In this kind of networks, the participation of any node is open and this ensures a higher degree of resilience and robustness in cases of dynamicity and node movements and failures, while at the same time there are reduced maintenance costs. On the other hand, this type of networks is suffering from frequently changes in architecture because peers join and leave the system without any warning anytime, thus keeping the robustness of the network is considered to be the most challenging aspect.

The performance of lookups in unstructured Peer-to-Peer overlays is not so great because it is based on flooding the network, which creates a great overhead.

Furthermore, it will be analyzed several unstructured Peer-to-Peer overlays in chronological manner.

### 2.1.1 Freenet 2001

The Freenet unstructured Peer-to-Peer overlay was proposed in (Clarke et al. 2001). This free approach has allowed peers to exchange information in an anonymous manner. Moreover, the main goals of Freenet are reliability and security. Even though, Freenet is fully decentralized, and there are no rules concerning node joining and leaving, it has a proper mechanism to facilitate information retrieval without flooding. The structure of the freenet P2P overlay is illustrated in Figure .1

**Joining, leaving and Lookup in Freenet overlay** Joining the Freenet overlay is realized by simply finding an existing peer in the network and connecting to it. When the new nod has become part of the overlay it holds its own data-store along with a routing table to other peers together with an index of their stored resources. Each piece of resource is indexed by a unique identifier computed by all nodes with a hash function. Therefore, lookups in Freenet are realized through this unique keys. When a request for a key can be a satisfied by a remote node, the data will return to the node that originally requested it along the reverse path.

Leaving the Freenet overlay is realized asynchronously, without any announcement in the network. The resources stored by the node how has just left diaper until the peer joins again the overlay if there is no other peer in the system with that resources.

### 2.1.2 Gnutella 2002

The most widely known unstructured Pee-to-Peer overlay is Gnutella (Schollmeier et al. 2003). The purpose of this overlay is to allow peers to share resources in a fully decentralized, scalable, reliable and anonymous way.

**Joining, leaving and Lookup in Gnutella overlay** Similar to Freenet overlay, joining in Gnutella is realized by searching an existing node and contacting it. When a node joins Gnutella is able to realize 3 operations:

- Locate and connect to other nodes.
- Query and retrieve resources.

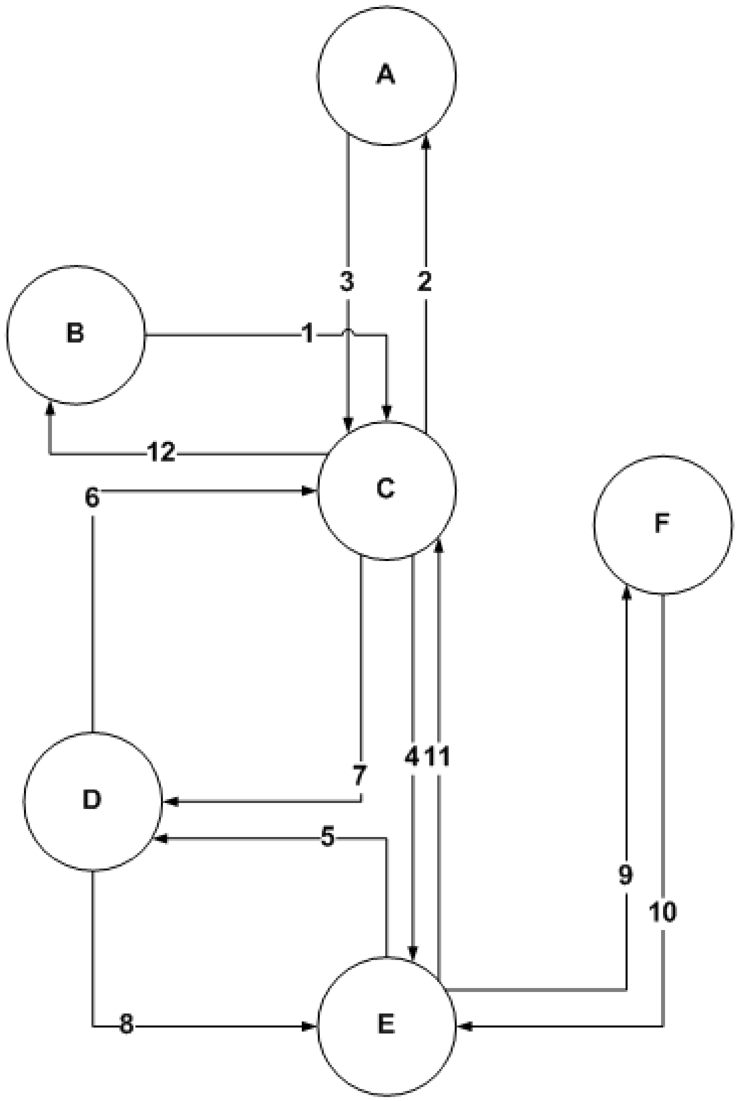


**Figure .1:** FREENET unstructured Peer-to-Peer overlay

- Push data resources.

Furthermore, in order to realize the operations mentioned above, Gnutella protocol has 5 type of messages:

1. PING  check if a node is available.
2. PONG  reply of the PING message if the node is available.
3. QUERY  search a particular resource.
4. QUERYHIT  reply from a node that has the queried resource.
5. PUSH  file downloads from nodes that are behind firewalls.

### 2.1.3 FastTrack 2006

FastTrack (Liang et al. 2006) is a fully decentralized unstructured Peer-to-Peer overlay with a hierarchical topology. In this overlay, there are 2 kinds of peers: super nodes and ordinary nodes. Thus FastTrack has a two-tier hierarchical topology.

**Joining, leaving and Lookup in FastTrack overlay** Concerning bootstrapping, every node has a local super-node list that also includes information about their workload. New nodes select the most appropriate super-node to connect to, based on the information about workload and on its location. As far as lookup is concerned, each node maintains a local resource index and in the case of ordinary nodes this is shared with its associated super-node. This guarantees that super-nodes can correctly reply about resources of all their underlying ordinary nodes. Thus, queries are propagated between super-nodes to enhance their coverage of the entire overlay and increase the possibility of getting a successful response.

### 2.1.4 Other unstructured P2P overlays

**BitTorrent 2003** BitTorrent Cohen (2003) is one of the most popular P2P file-sharing systems available nowadays with millions of active users.

**Gia 2004** Gia (Chawathe et al. 2003) is an unstructured Peer-to-Peer overlay based on Gnutella 0.6 and implements the concept of super-node proposed by FastTrack.

**UMM 2010** UMM (Unstructured Multisource Multicast (Ripeanu et al. 2010) is an unstructured Peer-to-Peer overlay designed for group communication using multicasting.

## 2.2 Structured Peer-to-Peer

Structured Peer-to-Peer overlays 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.

Furthermore, several structured Peer-to-Peer overlays will be analyzed in a chronological order.

### 2.2.1 CAN 2001

The most mature structured Peer-to-Peer overlays is CAN (Content Addressable Network). CAN overlay was proposed by Ratnasamy in article (Ratnasamy et al. 2001a). Based of DHT (Distributed Hash Tables) CAN overlays present the following features: self-organization, scalability and fault tolerance. The organization of nodes in a CAN overlays is in a n-dimensional coordinate space, where n represents the parameter of the CAN protocol.

In CAN overlays every node has an assigned a key/value pair. Based on this fact every node has the responsibility of a certain area in the CAN space, by storing additional information regarding it's neighbors such IP addresses or other coordinates. The key value stored by each node is generated based on a hash function because this functions has no proven collisions.

Furthermore, it will be analyzed the fundamental operations in Peer-to-Peer overlays: joining, leaving and lookup. The structure of the CAN P2P overlay is illustrated in Figure .2

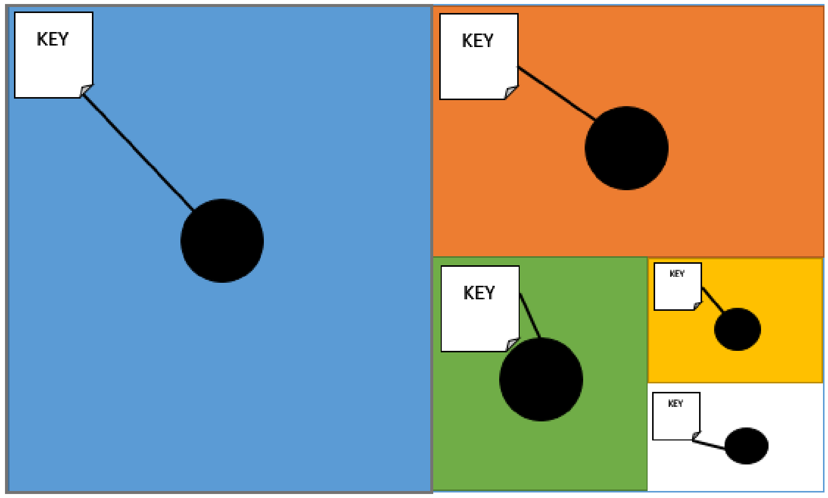


**Figure .2:** CAN STRUCTURED PEER-TO-PEER OVERLAY

**Joining in CAN overlays** When a node joins the CAN overlay needs to follow 3 fundamental steps: finding an existing node in the overlay, determine the zone witch will be allocated to that node and update its neighbor's sate tables. Therefore, when a node joins the overlay, randomly picks a point in the CAN space and sends a JOIN message through a known node in the network. When the targeted node receives a JOIN message, it will begin the process of zone allocation for the new node, by splitting its zone into half and assigning it to new arrived node. After the zone allocation the targeted node transfers the key/value pairs to the arrived node for which it has become responsible. Finally, neighbor's allocation tables are updated in order to be aware of the new peer through UPDATE messages.

**Leaving in CAN overlays** Leaving the CAN overlay deliberately by a node occurs when the node notifies the neighbor with the smallest zone and transfers all the content in order to keep de resources available.

Another common way of departure is node failure. When this occurs the space remained without owner must be taken over by an existing peer in the system. The detection of failing nodes is realized upon the update phase. When a node sends an UPDATE message and fails to receive an acknowledgement message from a node it triggers a self-timer and sends a TAKEOVER message. The timer for each node is dynamically allocated based on the size of their zone. If no response from the deserted node arrives, the elected node merges booth zones.

**Lookup in CAN overlays** Routing in CAN overlays is based on the coordinates stored by each node in its routing table. The information stored in this tables consists of the IP address and the unique identifier of each neighbor. Therefore, delivering a message to its destination is realized by forwarding it to the neighbor with the coordinates nearest the destination coordinates. This procedure is realized based on greedy mechanism. For instance, for a 2-dimension CAN space the test metric used is the Euclidian distance.

Moreover, concerning the performances of the CAN overlay, the average number of hops in a n dimension space with m nodes is given by the following equation:

$$Hops = O \times (n \times (m^{1 \div n})), \tag{.1}$$

Therefore, the CAN overlay can be considered reliable in respect of its construction.

**Applications of CAN overlays** An interesting application based on the CAN overlay is presented in (Ratnasamy et al. 2001b). The authors proposed a solution for an application-level multicast service. Their proposal has two major advantages: simplicity of the schema because of the architecture of CAN and scalability in terms of large groups without modifying the service architecture.

Another application based on CAN overlay is presented in (Elmas and Ozkasap 2004). This paper presents a prototype for document sharing and text classification over CAN. The authors proposed an algorithm for classification that produces a key value for each document found in the system.

### 2.2.2 Kademlia 2002

Kademlia overlay was proposed by Maymounkov and Mazieres in paper (Maymounkov and Mazieres 2002) This Peer-to-Peer overlay is a fully decentralized and structured. This overlay uses the XOR metric, that measures the distance between 2 nodes as the result of the bit-wise exclusive OR.

In Kademlia overlay each node and resource are assigned to a 160-bit value. The distance between 3 nodes is given by the following equation:

$$d(a, b) = a \oplus b, \tag{.2}$$

Each node in the overlay stores a table of nodes that has a distance from itself between $2^i$ and $2^{i+1}$, where $0i < 160$. This table is called a k-bucket, with a dynamic size given by parameter k.

**Joining and leaving in Kademlia overlays** Joining the Kademlia overlay practically implies that the new node will appear in a known nodes k-bucket and after that through lookup operations the new node populates its own list of node distances. When a new node, that has not yet participating the system before wants to join the overlay, simply computes a random number that has not been used by other peers and assign this as its ID. This value is kept by the peer until it leaves the system.

By using k-buckets, the Kademlia overlay is considered to be resistant to DoS (Denial of Service) attacks, because by its construction the k-bucket is not populated with the new node until an old node leaves the system.

**Lookup in Kademlia overlays** The most advantageous feature of Kademlia overlay is the fact that node lookup is realized in an asynchronous manner. When a node searches for an another one it initiates a $FIND - NODE$ request in its own k-bucket. Additional information like round trip timer is added when consulting other nodes in the system. The Kademlia architecture does not allow surpassing queries at the same time.

Concerning resource locating, there are located based on key value generated by a hash function. These values are stored on several nodes in order to gain asynchronous access to certain data. In terms of performance the Kademlia overlay has a value of $O \log N$ for resource discovery, where N represent the number of nodes in the overlay.

**Applications of Kademlia overlays** Kademlia overlay is one of the most popular structured overlays, that has a wide applicability in open-source implementations such as BitTorrent Pouwelse et al. (2005) or eMule (Kulbak et al. 2005).

### 2.2.3 Chord 2003

Even though Chord (Stoica et al. 2003) is not the most mature structured Peer-to-Peer overlay is considered the most popular overlay. The Chord protocol uses consistent hashing for identifying each node or resource in the system.

Moreover, based on its construction Chord is fully decentralized and distributed. Therefore, the structure of this overlay consists of a virtual ring, where every node has an identifier larger than its higher than its predecessor's one and has knowledge only on its successor neighbor. The structure of the Chord structured P2P overlay is illustrated in Figure .3

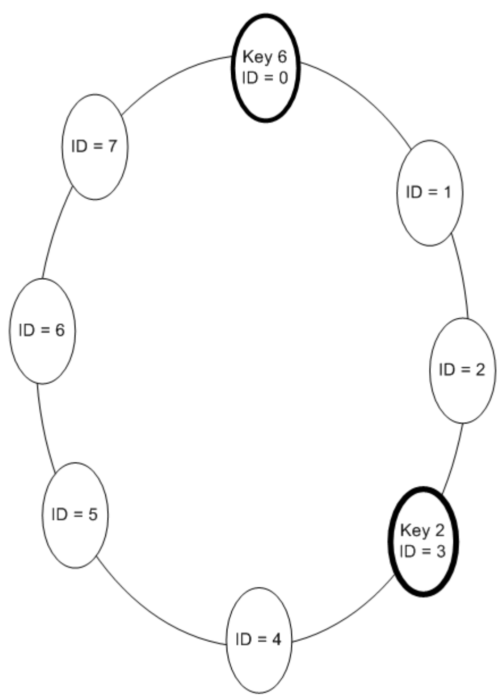


**Figure .3:** CHORD STRUCTURED PEER-TO-PEER OVERLAY

**Joining and leaving Chord P2P overlays** When a new node joins the system must be satisfied 3 conditions:

- Every node predecessor points correctly to its first successor.
- Each key is stored in successor (k).
- For each node the finger table is correct.

In order to fulfil the conditions mention above, when a node joins has done accomplish the following steps:

1. Initialization of the new node.
2. Notification of other nodes to update their tables.
3. The new node takes over the keys from its successor.

When a node leaves the overlay, the robustness of the system is not affected because each node keeps a list of successor nodes of a fixed size. If the immediately next successor fails to respond the node can modify the entry point of the second successor and become the first one.

**Lookup in Chord overlays** Data searching in Chord overlay is realized based on the successor relationships. Each node in the system stores a finger table of a maximum size of m and is constructed in such manner in which every node has on its $i^{th}$ position in the table the pointer to the successor node of $n + 2^{i+1}$ in the Chord virtual ring. Finger tables allow a node to have information about nodes in its vicinity as well as about a few remote nodes and balance the tradeoff between maintaining accurate information about a lot of nodes and only being aware of the successor in the Chord ring. Considering a stable $N - node$ Chord overlay the number of messages needed to locate a certain resource is given by $O \log N$.

**Applications of Chord overlays** Based on Chord overlay researchers have founded an interesting application for Wireless Mesh Chord (Canali et al. 2010). In this paper was presented a new approach called MeshChord, approach that exploits the features of wireless mesh networks.

#### 2.2.4 Other structured P2P overlays

In this subsection will be presented in summary form other structured Peer-to-Peer overlays.

**Viceroy 2002** Viceroy overlay (Malkhi et al. 2002) is similar to Chord overlay and uses hash values for mapping node resources. In Viceroy overlay every node is responsible for the resources with the identifier smaller than its own and larger than its immediate neighbor in the ring.

**SkipNet 2003** SkipNet (Harvey et al. 2003) is a scalable Peer-to-Peer overlay with good scalability properties, that offers secure and controlled data storage and guaranteed routing due to the fact that it organizes data primarily by string names.

**Coral -2004** This overlay (Freedman et al. 2004) shares principles with standard DHT-based, structured overlay networks by introducing distributed sloppy hash tables.

**Tapestry 2004** Tapestry overlay (Zhao et al. 2004) is a complete decentralized protocol for Peer-to-Peer systems. This overlay facilitates routing and efficient data localization without knowing the physical location of the data. Each node in Tapestry is assigned to a unique identifier formed by a number of 160 bits. An interesting feature of Tapestry is the fact that it is orientated to fault tolerance and high scalability. In order to achieve this features Tapestry uses redundancy of data.

**Cycloid 2006** This overlay (Shen et al. 2006) is similar to Viceroy. The topology of this overlay is that of a cube connected cycles graph. Each node has 7 connections to other nodes and the lookup complexity is $O(d)$.

**HyPeer 2011** This approach (Serbu et al. 2011) present an extended structured overlay of Chord based on a hypercube in order to offer flexible routing strategies due to the fact that it offers redundancy concerning the lookup paths.

# 3 Self-adaptive Overlays

## 3.1 Bio-inspired P2P overlays

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, it will be presented 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. Solutions based on swarm intelligence, namely based on the collective behavior of ant colonies or bees, have validated and guaranteed scalability due to the distributed intelligence and the reduced communication costs.

Moreover, a chronological analysis of several bio-inspired overlays will be presented.

### 3.1.1 SelfChord 2010

The Self-Chord overlay was proposed in paper (Forestiero et al. 2010) as solution 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 architecture of the overlay is based on the Chord structured overlay, thus the nodes in this Peer-to-Peer system are structured in a ring form. Each node is ordered according to its unique key, computed by using a hash function. The difference lies in the placement of resource identifiers. The main difference between Chord lies on the fact that resources obtain their identifiers from a different namespaces and are placed on nodes based on load balancing criteria, as well as semantic ones.

Furthermore, Self-Chord is considered a good proof of concept of a successful application of bio-inspired swarm intelligence algorithms in structured P2P overlays. Concerning resource discovery, Self-Chord is considered to be very efficient due to the fact that it uses finger tables to provide logarithmic guarantees in locating resources. The distribution of the resourced keys is realized in such manner it facilitates better load balancing. Moreover, the management of the of Self-Chord overlay is more efficient than the management of Chord because there is no need to re-arrange resource identifiers subject to node churn because.

### 3.1.2 P2PBA 2011

The P2PBA (Peer-to-Peer Bee Algorithm) (Dhurandher et al. 2011) is designed for efficient 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. The main idea of this algorithm is the fact that it and returns a list of zones called patches where high concentration of the results can be found by reducing the search area.

### 3.1.3 Self-CAN 2012

This overlay (Giordanelli et al. 2012) is similar to Self-Chord. 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. The rearrangement of resource identifiers aims at minimizing the centroid value of each node, thus essentially promoting the collection of similar resources at every node and speeding up lookup operations.

Further research will consider a better understanding of the bio-inspired Peer-to-Peer overlays and simulation of the performances of this overlays with existing tools. For example, in the next report will be presented an analysis of the Self-Chord overlay in terms of performance. Also in the next report will be presented the opportunity to improve the security of the self-chord overlay by adding a new layer for this purpose.

### 3.1.4 Honeycomb - 2014

This overlay (Pop et al. 2014) is a Peer-to-Peer overlay that is based on the hexagons of the honeycomb. The maim advantage of this structure is the minimal number of neighbors in the network which is 3.

### 3.1.5 SPIDER - 2015

Paper (Mocanu et al. 2015) presents an bio-inspired Peer-to-Peer network based in the spider web. In this type of overlay each node can have maximum 4 neighbors. The structure of this overlay consists of a fixed number of chains and a variable number of rings.

## 3.2 Multi layer Peer-to-Peer

In real life situations is more feasible to use several P2P overlays on the same hardware infrastructure due to the fact that there is no P2P overlay that satisfies all the pervasive network issues.

For instance, a node can be part of several domains or applications because it can facilitate data dissemination through file-sharing or VoIP (Hsu et al. 2010).

Figure .4 represents a multilayer P2P overlay that runs on the same physical network.

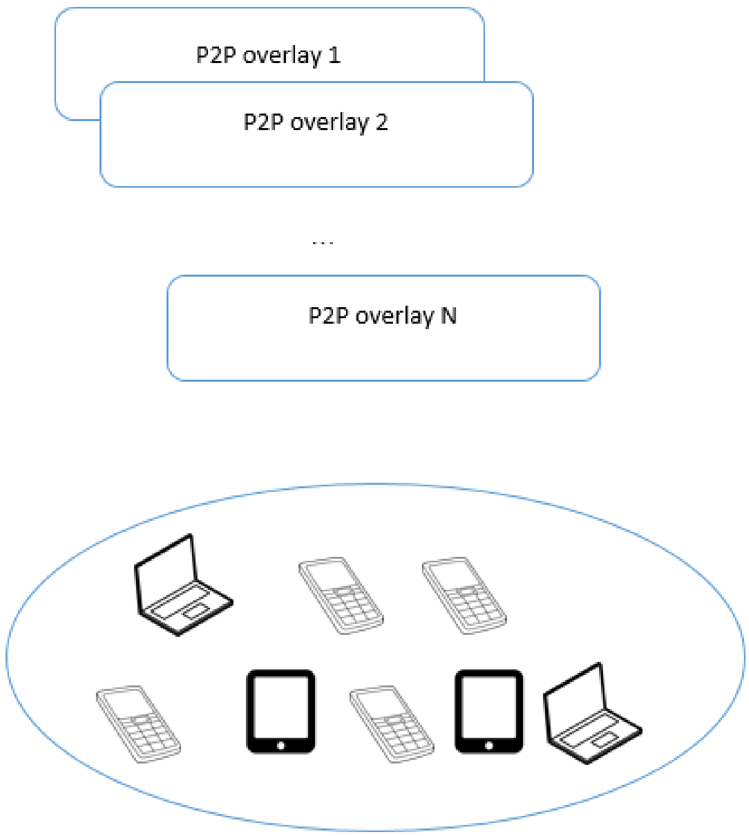


**Figure .4:** Multi-layer P2P overlay with N overlays situated on top of a single set of hardware devices

For this N layer P2P overlays the management must be realized simultaneously in order to have the same hardware infrastructure updated on each overlay. In order to realize such management is required the existence of a communication layer between each overlay called overlay coexistence.

Another strong point for the usage of the same hardware infrastructure on the bottom of several P2P overlays is presented in paper (Mao et al. 2012). This aim of this paper is to resent a vitalization platform for P2P overlays called MOSAIC. Visualization, in terms of P2P overlays can assure easily deployment of network protocols in real life networks.

The problems encountered when using multiple P2P overlays in the same time have been presented in paper (Jiang et al. 2005). Also a good reference for multiple P2P overlays is (Maniymaran et al. 2007). This paper presents the benefits of using booth structured and unstructured overlays o the same hardware infrastructure

The most important issue in multi-layer overlays is the manner in which the resources shared by the physical nodes in the physical network are distributed between the multi-layer overlays. In order to realize such coexistence it is introduced a middle-ware layer. In paper (Cooper 2006) is presented a priority based coexistence of multiple P2P overlays.

# 4 Hybrid Peer-to-Peer Systems

In this section two hybrid Peer-to-Peer overlays will be presented.

## 4.1 JXTA

The JXTA overlay was first presented in (Barolli and Xhafa 2011). This overlay represents a building block for the development of Peer-to-Peer applications and services. In JXTA overlay, the nodes are self-organized in a group manner called peergroup. A peergroup is defined a set of nodes that share common interests and respect the same politics and rules regarding look-ups, joining, leaving and naming. The JXTA overlay defines the Resolver service Protocol used for resolution operations such DNS or binding an IP address to a specific port. The resolver service is realized by the rendezvous nodes by reducing the communication overhead. In this overlay there are no restrictions for any peers to become a rendezvous peer. Each rendezvous peer in the overlay stores a list of advertisement messages delivered by each peer nodes through a service called Shared Resource Distributed Index. Figure .5 illustrates the structure of JXTA overlay.

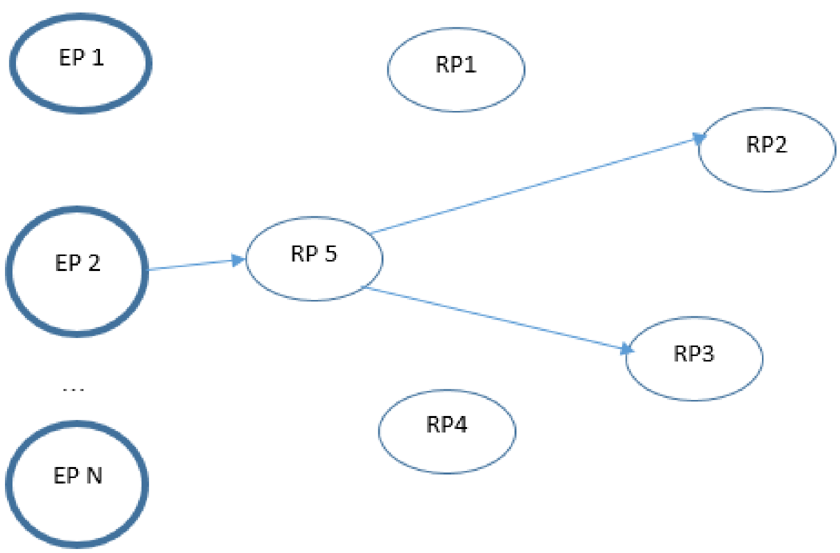


**Figure .5:** JXTA Peer-to-Peer overlay structure EP - edge peer, RP - rendezvou peer

**Table .1:** Table 1. Taxonomy of unstructured Peer-to-Peer overlays

| Overlay | Redundancy | Dynamicity | Security | Lookup | Queries |
|---|---|---|---|---|---|
| **Freenet** | Path redundancy | Distributed design | Anonymity and privacy | Constrained flooding | Any type of queries |
| **Gnutella** | No particular support | Periodic initiation of node discovery | Anonymity | Pure flooding | Any type of queries |
| **FastTrack** | Resource indices replication | No particular support | No | Flooding between super-nodes | Any type of queries |

## 4.2 MOPAR

MOPAR (Yu and Vuong 2005) is hybrid Peer-to-Peer overlay designed for Interest Management of Massively Multiplayer Online Games. This overlay is based on Network Virtual environment (NVE) in fully distributed manner. This approach uses booth structured P2P overlay such Distributed Hash Tables (DHT) and unstructured P2P overlay. The architecture of this overlay is based on hexagonal zoning approach. The main reason why hexagonal zoning approach is used mainly in game applications is the uniformity of booth the orientation and adjacency of the hexagons.

# 5 Discussion and Conclusions

This chapter presented a state of the art survey of the existing solutions for Peer-to-Peer overlays. The review was split in 3 categories of Peer-to-Peer overlays: unstructured, structured and bio-inspired overlays. For each overlay was presented the main idea and the basic operations. The conclusions drawn concerning the unstructured Peer-to-Peer overlays are ilustrated in Tablet1. As can be seen in terms of redundancy Freenet offers path redundancy and FastTrack only offers resource indices replication. For the Gnutella overlay there is no redundancy support. Concerning dynamicity booth Freenet and FastTrak overlays does not support any dynamical modifications on the fly. The security aspect is treated by the Freenet overlay in terms of assuring anonymity and privacy while for the Gnutella overlay that assures only anonymity. In all 3 analyzed unstructured P2P overlays the lookup operation is realized by flooding the network, therefore the overhead in the network is large.

Tablet2 presents the taxonomy of 3 structured Peer-to-Peer overlays. As it can be seen in the table in terms of redundancy CAN overlay supports routing redundancy and Kademlia overlay supports response replication and caching. Concerning the security issue Kademlia is the only structured P2P overlay that offers security facilities for ol nodes that are more trusted. In terms of speed booth chord and Kademlia overlays offer the same values for lookups which is higher in comparison with the CAN overlay.

Tablet3 illustrates the taxonomy of bio-inspired Peer-to-Peer overlays. Therefore, redundancy is a property of the mentioned overlays only for the BlatAnt and AntOM overlays while Self-Chord does not support such property. In terms of dinamicity the same overlays that have the redundancy property are satisfying this condition. Moreover, the security issue is not treated by any of the studied bio-inspired overlays. Blatant is a bio-inspired overlay that has a higher overhead for lookups in comparison with the other overlays due to the fact that is uses flooding and random

**Table .2:** Table 2. Taxonomy of structured Peer-to-Peer overlays

| Overlay | Redundancy | Dynamicity | Security | Lookup | Queries |
|---|---|---|---|---|---|
| **CAN** | Routing redundancy | No | No | (d/4) n1/d | Standard queries |
| **Kademlia** | Resource replication and caching | Node behavior for routes longevity | Old nodes more trusted | O (log N) | Standard queries |
| **Chord** | No | No | No | O (log N) | Standard queries |

**Table .3:** Table 2. Taxonomy of bio-inspired Peer-to-Peer overlays

| Overlay | Redundancy | Dynamicity | Security | Lookup | Queries |
|---|---|---|---|---|---|
| **Self-Chord** | No | No | No | O (log N) | Standard queries |
| **BlatAnt** | Yes | Yes | No | Flooding and random walks | Standard queries |
| **AntOM** | Yes | Yes | NO | O (log N) | Standard queries |

walks.

# Index of terms

**P2P** Peer-to-Peer

**UMM** Unstructured Multi-source Multi-cast

**DHT** Distributed hash tables

**CAN** Content addressable network

**P2PBA** Peer-to-Peer bee algorithm

# 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.

# References

Barolli, L. and F. Xhafa, 2011: Jxta-overlay: A p2p platform for distributed, collaborative, and ubiquitous computing. *Industrial Electronics, IEEE Transactions on*, **58 (6)**, 2163–2172.

Canali, C., M. E. Renda, P. Santi, and S. Burresi, 2010: Enabling efficient peer-to-peer resource sharing in wireless mesh networks. *Mobile Computing, IEEE Transactions on*, **9 (3)**, 333–347.

Chawathe, Y., S. Ratnasamy, L. Breslau, N. Lanham, and S. Shenker, 2003: Making gnutella-like p2p systems scalable. *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, ACM, 407–418.

Clarke, I., O. Sandberg, B. Wiley, and T. W. Hong, 2001: Freenet: A distributed anonymous information storage and retrieval system. *Designing Privacy Enhancing Technologies*, Springer, 46–66.

Cohen, B., 2003: Incentives build robustness in bittorrent. *Workshop on Economics of Peer-to-Peer systems*, Vol. 6, 68–72.

Cooper, B. F., 2006: Trading off resources between overlapping overlays. *Proceedings of the ACM/IFIP/USENIX 2006 International Conference on Middleware*, Springer-Verlag New York, Inc., 101–120.

Dhurandher, S. K., S. Misra, P. Pruthi, S. Singhal, S. Aggarwal, and I. Woungang, 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.

Elmas, T. and O. Ozkasap, 2004: Distributed document sharing with text classification over content-addressable network. *Content Computing*, Springer, 70–81.

Forestiero, A., E. Leonardi, C. Mastroianni, and M. Meo, 2010: Self-chord: a bio-inspired p2p framework for self-organizing distributed systems. *Networking, IEEE/ACM Transactions on*, **18 (5)**, 1651–1664.

Freedman, M. J., E. Freudenthal, and D. Mazieres, 2004: Democratizing content publication with coral. *NSDI*, Vol. 4, 18–18.

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

Giordanelli, R., C. Mastroianni, and M. Meo, 2012: Bio-inspired p2p systems: The case of multidimensional overlay. *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, **7 (4)**, 35.

Harvey, N. J., M. B. Jones, S. Saroiu, M. Theimer, and A. Wolman, 2003: Skipnet: A scalable overlay network with practical locality properties. *networks*, **34**, 38.

Hsu, C.-J., W.-C. Chung, K.-C. Lai, L. K.-C. Li, and Y.-C. Chung, 2010: A novel approach for cooperative overlay-maintenance in multi-overlay environments. *Cloud Computing Technology and Science (CloudCom), 2010 IEEE Second International Conference on*, IEEE, 81–88.

Jiang, W., D.-M. Chiu, and J. C. Lui, 2005: On the interaction of multiple overlay routing. *Performance Evaluation*, **62 (1)**, 229–246.

Khan, R. and R. Hasan, 2014: Secp2psip: A distributed overlay architecture for secure p2psip.

Kulbak, Y., D. Bickson, et al., 2005: The emule protocol specification. emule project.

Liang, J., R. Kumar, and K. W. Ross, 2006: The fasttrack overlay: A measurement study. *Computer Networks*, **50 (6)**, 842–858.

Malkhi, D., M. Naor, and D. Ratajczak, 2002: Viceroy: A scalable and dynamic emulation of the butterfly. *Proceedings of the twenty-first annual symposium on Principles of distributed computing*, ACM, 183–192.

Maniymaran, B., M. Bertier, and A.-M. Kermarrec, 2007: Build one, get one free: Leveraging the coexistence of multiple p2p overlay networks. *Distributed Computing Systems, 2007. ICDCS'07. 27th International Conference on*, IEEE, 33–33.

Mao, Y., B. T. Loo, Z. Ives, and J. M. Smith, 2012: Mosaic: Declarative platform for dynamic overlay composition. *Computer Networks*, **56 (1)**, 64–84.

Maymounkov, P. and D. Mazieres, 2002: Kademlia: A peer-to-peer information system based on the xor metric. *Peer-to-Peer Systems*, Springer, 53–65.

Mocanu, B., F. Pop, A. Mihaita, C. Dobre, and V. Cristea, 2015: Spider: A bio-inspired structured peer-to-peer overlay for data dissemination. *P2P, Parallel, GRIS, CLOUD and Internet Computing (3PGCIC), 2014 IEEE 10th International Conference on*, IEEE.

Pop, F., O.-M. Citoteanu, C. Dobre, and V. Cristea, 2014: Resource trust management in auto-adaptive overlay network for mobile cloud computing. *Parallel and Distributed Computing (ISPDC), 2014 IEEE 13th International Symposium on*, IEEE, 162–169.

Pouwelse, J., P. Garbacki, D. Epema, and H. Sips, 2005: The bittorrent p2p file-sharing system: Measurements and analysis. *Peer-to-Peer Systems IV*, Springer, 205–216.

Ratnasamy, S., P. Francis, M. Handley, R. Karp, and S. Shenker, 2001a: *A scalable content-addressable network*, Vol. 31. ACM.

Ratnasamy, S., M. Handley, R. Karp, and S. Shenker, 2001b: Application-level multicast using content-addressable networks. *Networked Group Communication*, Springer, 14–29.

Ripeanu, M., A. Iamnitchi, I. Foster, and A. Rogers, 2010: In search of simplicity: a self-organizing group communication overlay. *Concurrency and Computation: Practice and Experience*, **22 (7)**, 788–815.

Schollmeier, R., I. Gruber, and F. Niethammer, 2003: Protocol for peer-to-peer networking in mobile environments. *Computer Communications and Networks, 2003. ICCCN 2003. Proceedings. The 12th International Conference on*, IEEE, 121–127.

Serbu, S., P. Felber, and P. Kropf, 2011: Hypeer: Structured overlay with flexible-choice routing. *Computer Networks*, **55 (1)**, 300–313.

Shen, H., C.-Z. Xu, and G. Chen, 2006: Cycloid: a constant-degree and lookup-efficient p2p overlay network. *Performance Evaluation*, **63 (3)**, 195–216.

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

Wei, T.-T., C.-H. Wang, Y.-H. Chu, and R.-I. Chang, 2012: A secure and stable multicast overlay network with load balancing for scalable iptv services. *International Journal of Digital Multimedia Broadcasting*, **2012**.

Yu, A. P. and S. T. Vuong, 2005: Mopar: a mobile peer-to-peer overlay architecture for interest management of massively multiplayer online games. *Proceedings of the international workshop on Network and operating systems support for digital audio and video*, ACM, 99–104.

Zhao, B. Y., L. Huang, J. Stribling, S. C. Rhea, A. D. Joseph, and J. D. Kubiatowicz, 2004: Tapestry: A resilient global-scale overlay for service deployment. *Selected Areas in Communications, IEEE Journal on*, **22 (1)**, 41–53.