Journal of Network and Computer Applications 55 (2015) 1–23

Contents lists available at ScienceDirect

Journal of Network and Computer Applications

journal homepage: www.elsevier.com/locate/jnca

Review

State-of-the-art survey on P2P overlay networks in pervasive computing environments

Apostolos Malatras

European Commission, Joint Research Centre (JRC), Institute for the Protection and Security of the Citizen (IPSC), via Enrico Fermi 2749, 21027 Ispra VA, Italy

ARTICLE INFO

Article history:
Received 16 July 2014
Received in revised form
17 March 2015
Accepted 27 April 2015
Available online 9 May 2015

Keywords:
P2P
Overlay networks
Coexisting overlays
Bio-inspired overlays
Structured
Unstructured

ABSTRACT

P2P overlay networks have attracted significant research interest due to their utility as virtualized abstractions of complex network infrastructures, optimized to satisfy specific criteria, e.g. minimum delay or shortest diameter. In the context of modern pervasive environments, which are characterized by complexity, heterogeneity, dynamicity, and mobility in terms of the underlying networks, the utilization of P2P overlays accordingly offers a series of advantages by countering the aforementioned adverse characteristics. The widespread deployment of pervasive environments and the plethora of proposed P2P systems, both call for a systematic way to study related research works. In this paper we review related research on P2P (peer-to-peer) overlay networks in pervasive environments. In this respect, we therefore analyze relevant requirements and discuss the application and deployment of P2P overlays and systems on top of the networking infrastructures that are supported by pervasive environments. Aspects such as scalability, resource discovery, algorithmic complexity, security, and support for dynamicity are examined for existing research works, in an effort to identify the most suitable P2P overlay for the requirements set by the nature of pervasive environments. We also taxonomize P2P overlays using the well-established classification scheme in regard to their structure or lack of it. Furthermore, we study the notions of multi-layer and bio-inspired P2P overlays that have great synergies with pervasive environments due to their inherent characteristics, especially in terms of flexibility and robustness. By describing and critically analyzing existing systems and discussing current research and open issues, we aim to instigate further research in this domain, while at the same time this work should serve as a point of reference for state-of-the-art P2P overlays in pervasive environments.

Contents

http://dx.doi.org/10.1016/j.jnca.2015.04.014
1084-8045/

1. Introduction

By providing end users with seamless, customized and unobtrusive services over heterogeneous infrastructures pervasive computing environments have attracted significant research interest and have found great applicability in commercial settings (Abowd and Mynatt, 2000). The notion of pervasive computing (also known as ubiquitous computing) refers to anywhere and anytime user-centric provisioning of value-added services and applications (Weiser, 1991) that are adaptive to user preferences and monitored conditions, i.e. context information (Dey, 2001). Pervasive environments are built on top of high capacity, distributed networking infrastructures, and therefore they inherently promote flexibility, availability and adaptability, as well as support mobility.

In the last years we have experienced a proliferation of wireless, mobile and wired networking technologies that is spurred by the need to support and optimize pervasive computing scenarios (Conti and Giordano, 2007; Akyildiz et al., 2005). In this respect, the underlying networks should be flexible and extensible enough to accommodate such a level of diversity and personalization concerning both the users and the applications running on top of pervasive environments (Schonwalder et al., 2009). Therefore, the fulfillment of the vision set by pervasive computing comes at a high cost when considering relevant concerns such as the dynamicity, scale, complexity and heterogeneity. Accordingly, network virtualization techniques such as P2P overlays are a prominent solution to alleviate the aforementioned concerns, since they allow for multiple virtual network topologies to be built on top of the actual physical networks (Niebert et al., 2008). Complex, volatile topologies are abstracted with the use of P2P overlays into more manageable topologies. P2P overlays can be constructed to satisfy certain criteria, e.g. bounded delay or hop count (Zhang et al., 2005) or load balancing (Forestiero et al., 2010).

P2P overlays in pervasive environments is a topic that has attracted significant research interest with a number of relevant works proposing solutions to address the adverse properties of these environments that were previously mentioned. Besides mechanisms and algorithms considered for traditional structured and unstructured P2P overlays to enhance their operation and resource discovery mechanisms accordingly, there has recently been a growing interest in the research community towards overlay cooperation. The latter emerges as a necessity due to the large scale of pervasive environments and their inherent heterogeneity, which combined with the diversity of applications' requirements will undoubtedly require multiple overlays coexisting. Furthermore, the need for adaptive overlays in these dynamic and unreliable environments becomes evident. Accordingly, bio-inspired, highly reactive solutions for building and maintaining overlays have been proposed and exhibit efficient operation.

We review in the following the state-of-the-art in P2P overlays based on a standard classification scheme of structured vs. unstructured overlays. The classification is meant to highlight the differences between the topological properties that characterize each of the overlays. We also extend our review to cover bio-inspired P2P overlays, namely ones the topology and lookup operations of which are inspired by processes found in nature and biological organisms. Lastly, we also survey multi-layer overlays that consider multiple concurrent overlays cooperating in parallel and that benefit from the established synergies between the latter. Figure 1 presents an overview of the reviewed systems under the adopted classification scheme. Four distinct categories are identified, namely structured P2P overlays, unstructured, bio-inspired and multi-layer P2P. For each of these categories, Fig. 1 lists their main qualitative characteristics and features, as well as the main research works that fall under each category.

The contribution of this work is that it provides a systematic taxonomy and reference for the research work on P2P overlays in pervasive environments. It aims to group together all relevant

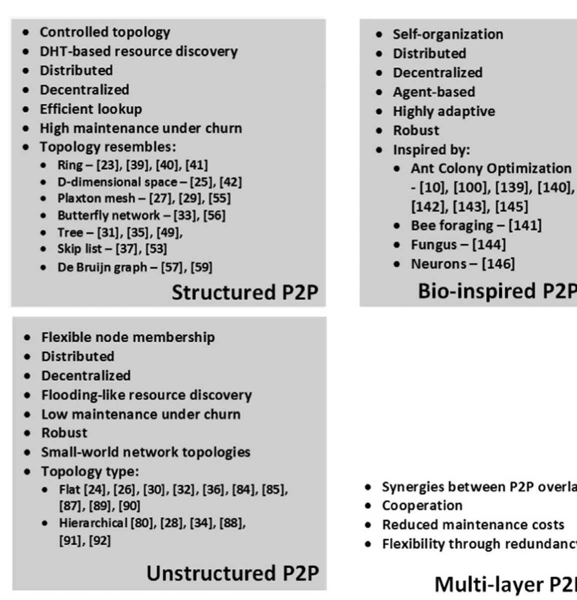

Fig. 1. Classification of P2P overlays and main features of each category, as well as typical examples.

research and expose open challenges and issues and thus to instigate further research in the domain. In addition, by presenting existing work in a systematic manner, prospective researchers can easily classify their work and validate their innovations, while also comparing the performance and other qualitative features of their work with related research efforts. Furthermore, a novelty in itself is the review of multi-layer and bio-inspired P2P overlays. Our methodology involves describing the different overlays and in particular their operation, but also qualititative features such as performance, lookup operations, security, load balancing, topology and handling of heterogeneity and dynamicity. In this way, we highlight how each of the considered overlays handles the adverse characteristics of pervasive environments that were previously mentioned, namely dynamicity, large scale, complexity and heterogeneity.

In this paper we present a detailed review of research on P2P overlays in the context of pervasive networking. We aim at highlighting their contributions and shortcomings and therefore facilitate their comparison and comparative analysis in a holistic context. After this brief introduction, Section 2 introduces the notion of P2P overlays, describes their general features and classifies them in structured and unstructured categories, typical cases of which are given in Section 3 and 4 respectively. Review of related research on multi-layer overlays and coexisting overlays is the focus of Section 5, while Section 6 discusses P2P overlays based on bio-inspired optimization techniques. Finally, the paper concludes in Section 7 with pointers to open issues and future research directions.

2. P2P overlays

The connectivity between nodes in a physical network is subject to the network infrastructure that is in place and it is cumbersome to modify it. Conversely, P2P overlays are essentially virtual, i.e. logical, structures/topologies that are built on top of physical networks. They do not suffer the rigidness of physical network topologies, since they are logical in nature, whereas this increased flexibility allows for extensibility and adaptive reconfiguration. The nodes (also known as peers) that participate in the P2P overlay represent their physical network counterparts, but the connectivity between them differs. They are organized in a distributed manner without any hierarchy or centralized control (Eng Keong et al., 2005). This implies that peers communicate with each other to establish self-organizing overlay structures on top of the underlying physical networks. The fact that P2P overlays can be built dynamically and reflect desired optimization criteria allows them to support a variety of application level services, which is the main motivation for their widespread use and utility. By design, P2P networks have a high degree of decentralization, inherent support for heterogeneity, self-organized behavior, they are easily deployed at the application-level, while being highly scalable and resilient to failures (Rodrigues and Druschel, 2010).

The flexibility and extensibility brought by P2P systems has spurred numerous research works in the area, with diverse P2P systems being proposed in order to cater for various optimization criteria and application requirements. It is therefore evident that a taxonomy to classify P2P overlay systems is necessary, in order to promote its systematic study and examination from the research community. The focal point and contribution of this paper is such a classification scheme for P2P overlays and systems, categorizing them in 4 distinct categories and presenting the main research works that fall under each of these categories.

One of the most common ways to classify P2P systems has been based on the structure of the topology of the overlay. Accordingly, P2P systems can be categorized as being either structured or unstructured. In structured approaches the topology of the overlay is tightly regulated and there are specific rules concerning the placement of peers in the topology, namely Distributed Hash Tables (DHTs) are usually exploited to determine where a peer will be located in the topology and with which other peers it will be connected. In unstructured approaches the topology of the overlay is unknown to peers and there are no rules to control it. Flooding mechanisms are used for resource discovery and peer locations are random (Eng Keong et al., 2005). While structured approaches allow for quick and efficient resource discovery, they have much higher management overhead for the overlay's topology maintenance (Meshkova et al., 2008).

When considering pervasive environments, the underlying physical network infrastructures are commonly comprised of heterogeneous wired and wireless networks, while additionally issues of large scale, dynamicity and mobility arise. Mitigating or addressing these adverse characteristics has motivates the need to build P2P overlay systems on top of pervasive environments. Moreover, the dynamic nature of pervasive environments where the physical connections change over time makes it difficult to perform resource discovery, e.g. look for files or services. P2P overlays could potentially assist in this respect by providing more scalable, robust lookup topologies that are not influenced by dynamicity. P2P overlay networks are therefore employed to produce more manageable networks and optimize resource monitoring, by exposing a (possibly) smaller scale abstraction of the underlying network infrastructure that adapts rapidly to the volatility and dynamicity of the underlay's topology.

Nonetheless, it should be noted that P2P overlay topologies are subject to changes too in order to retain their desired features (e.g. structured topologies) or to adapt to overlay network partitions and failures/mobility of nodes. However, carefully designed topology management algorithms should minimize these changes and accordingly adapt the topologies rapidly. Such topology management algorithms are at the core of P2P overlays, as will be discussed in the following, whereas there have also been dedicated research efforts towards promoting overlay reconfiguration in a more generic manner. Adaptive P2P overlay topologies have been studied in Condie et al. (2004) to improve resource discovery, while an adaptive topology control algorithm designed specifically

Table 1
Typical examples of structured and unstructured P2P overlays.

Structured	Unstructured
Chord (Stoica et al., 2003)	Freenet (Clarke et al., 2001)
CAN (Ratnasamy et al., 2001)	Gnutella (Gnutella, 2002)
Pastry (Rowstron and Druschel, 2001)	FastTrack (Liang et al., 2006)
Tapestry (Zhao et al., 2004)	BitTorrent (Cohen, 2003)
Kademlia (Maymounkov and Maziéres, 2002)	UMM (Ripeanu et al., 2010)
Viceroy (Malkhi et al., 2002)	Gia (Chawathe et al., 2003)
P-Grid (Aberer et al., 2003)	Phenix (Wouhaybi and Campbell, 2004)
SkipNet (Harvey et al., 2003)	

for P2P overlays running on top of mobile ad hoc networks has been proposed in Mawji et al. (2011) with the goal of reducing the stretch factor between the overlay and the underlay networks and eventually minimizing energy consumption. Similarly, the X-BOT protocol discussed in Leitao et al. (2012) retains desired features of existing unstructured P2P overlays such as node degree and connectivity, but adapts the topology in order to optimize the stretch factor in a completely decentralized manner and thus lead to more efficient overlay routing. In handling and proactively responding to partitions in the overlay the work by Qiu et al. (2007) provides an interesting theoretical framework to address this problem in a generic way. Lastly, Fan and Ammar (2006) discuss high-level reconfiguration policies for the optimization of service overlay topologies in terms of robustness.

Table 1 presents the most indicative cases of structured and unstructured P2P systems that we review in the following. The interested reader is also referred to Androutsellis-Theotokis and Spinellis (2004) and Eng Keong et al. (2005) for slightly outdated yet related reviews on P2P overlays. We should also highlight the extensive surveys on search techniques in P2P overlays reported in Risson and Moors (2006) and Ahmed and Boutaba (2011). Due to the ever increasing trend towards green IT there has been a number of research efforts aimed at reducing energy consumption of P2P overlay networks, for a detailed survey of which we refer to Malatras et al. (2012a).

3. Structured P2P overlays

The notion of structure in this category of P2P overlays refers to the tightly controlled topology that they maintain over the network graph and the fact that resources are not placed randomly on nodes but in a deterministic manner, i.e. using DHTs. These features allow structured protocols to have very good performance in terms of resource discovery, since resource queries can be satisfied in bounded number of steps even in large-scale distributed architectures. In most cases, nodes and resources share the same identifier space and are mapped on that space by means of consistent hashing functions. In the following we analyze the most prominent structured P2P overlays with the goal of describing their topologies (construction, maintenance under churn), their lookup operations (discover and retrieve resources from other nodes) and their performance (lookup efficiency, tradeoffs and applicability).

3.1. Chord

Chord (Stoica et al., 2003) is one of the first and most popular protocols for structured P2P overlays. It addresses the issue of locating nodes in P2P overlays that hold specific data items, i.e. keys, by using DHTs. In particular, Chord utilizes consistent hashing (Karger et al., 1997) for the assignment of keys to nodes and data in order to balance related workload of nodes, since it leads to nodes holding approximately equal number of keys. Moreover, Chord is resilient to churn and is fully decentralized and distributed.

Topology: Nodes in a Chord overlay are organized in a virtual ring where a node always has a higher identifier than its predecessor and is aware of its successor as its next hop neighbor. The consistent hashing technique unilaterally assigns a key to each node identifier (by hashing the node's IP address), whereas a key is also generated for every data item available on the overlay (by hashing the data). If the length of the identifier is m, then the identifiers are ordered in a Chord ring of modulo $2m$ size (from 0 to $2m-1$). A key k is handled by the node in the Chord ring whose identifier is either k or is the first to come in a clockwise order after k. When a new peer joins the overlay its IP is hashed and its position in the Chord ring is thus deduced based on its assigned key. Such an action will nonetheless undoubtedly require a rearrangement of the key-node assignments. In particular, some keys that were previously handled by the successor of the new node in the Chord ring will now need to be assigned to the new node itself, based on the previous principle of key-node assignment. Similarly, when a node leaves the overlay all the keys that were assigned to it will have to be reassigned to its successor in the ring. It is clear that node churn influences only the neighboring nodes the joining/leaving nodes and not the whole overlay and this is an important feature of Chord's efficiency.

Lookup: To search for data items on the Chord overlay (since a sequential traversal of the Chord ring using the successor relationships is clearly not efficient), each node maintains a routing table called the finger table. The finger table has a maximum size of m and is constructed so that for a node n its ith entry has a pointer to the successor of node $n+2^{i-1}$ in the Chord ring, where $1 \le \bar{i} \le \overline{m}$. 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. Evidently node churn influences the finger tables, which should thus be updated constantly to ensure accurate lookups. The same applies to the successor relationships between nodes in the Chord ring.

Performance: According to Stoica et al. (2003) a lookup procedure in a stable N-node Chord overlay requires $O(\log N)$ messages, while Eng Keong et al. (2005) indicate a $(\log N)^2$ performance under churn. Is it however noted that Chord suffers from massive churn values that diminish its performance and that it has a high maintenance cost especially under churn, as is the case with all structured P2P overlays. This has led to numerous approaches to improve the latter aspects, one of the most prominent of which is the Chord2 protocol (Joung and Wang, 2007) that considers a hierarchical structure with super-peers acting as index servers for lookup queries and facilitating message routing. Additionally, redundancy by means of concurrent maintenance of multiple Chord rings has been studied in Flocchini et al. (2007) as a means to improve lookup and routing efficiency with related results indicating up to 50% less hops to successfully satisfy lookup queries. Recently, a specialization of Chord for wireless mesh networks called MeshChord was proposed (Canali et al., 2010) that extends the algorithm to consider the particularities of the considered networks such as 1-hop broadcast and the existence of a wireless backbone infrastructure.

3.2. CAN

The main design goals of the distributed and decentralized structured CAN (Content Addressable Network) overlay (Ratnasamy et al., 2001) include self-organization, fault tolerance and scalability. It is also based on the DHT concept, but instead of organizing the overlay nodes in a ring like Chord it considers them as points in a d-dimensional coordinate space, where d is a parameter of the CAN protocol. Every node of the CAN P2P overlay is in charge of a particular area of the d-dimensional coordinate space, which is distinct and non-overlapping to that of other nodes.

Topology: The CAN coordinate space is used to store key/value pairs. In CAN every node is responsible for a particular area of the CAN space and has routing information, e.g. IP address and coordinates, of its neighbors in the coordinate space. To assign key/value pairs to specific nodes, a hash function is applied to the key to map it to a point on the CAN space. The node in the area of which that point belongs becomes responsible for that key/value pair. In terms of nodes joining or leaving the overlay, CAN assumes the existence of well-known bootstrap nodes that can be contacted in order to discover further CAN nodes. When a new node joins the overlay, it randomly picks a point in the CAN space and sends a dedicated JOIN message to that point. The existing CAN node in charge of that point (i.e. its area contains that point) divides its area in two, keeping one half for itself and assigning the other half to the node. It is clear that key/value reassignment occurs in accordance to the division of the area. Upon node departure, one of the departing node's neighbors assumes responsibility of its dedicated area and that of its associated key/value peers. Accordingly, all nodes continuously monitor their neighbors to update their routing information.

Lookup: Assuming the initiating node a lookup query is aware of the key associated to a data item, it applies the CAN hashing function to locate the point to which that key has been associated with and consequently the node that is in charge of the space containing that point. The CAN routing mechanism then takes over to route the request (and inversely the reply) to the discovered node. In doing so, CAN utilizes a greedy forwarding routing mechanism, namely at each node forwarding the request to the neighbor that is closest to the neighbor, e.g. utilizing Euclidean distance as a metric for a 2-dimensional CAN overlay.

Performance: As stated in Ratnasamy et al. (2001), every node of the CAN overlay maintains a list of $2*d$ neighbors, so assuming n equal zones in the d-dimensional space, the average routing path is computed to be equal to $(d/4)*n^{1/d}$. CAN is reliable since by its construction there are alternative routing paths to connect nodes. However, it is subject to failure in case of partitions and it does not guarantee load balancing as Chord, because there can be areas (and hence nodes) that contain much larger number of key/value pairs than others. Resilience under churn can be greatly improved by considering greater numbers of neighbors in the CAN overlay. In a typical such example, Wang et al. (2005) consider additional nodes in CAN routing, following small-world principles and the notion of finger table present in Chord.

3.3. Pastry

With a lot of similarities to Chord, the Pastry (Rowstron and Druschel, 2001) P2P overlay is a structured, self-organized and fully decentralized overlay aiming at scalable resource discovery and routing, while at the same time taking into account network locality to reduce average path length. Pastry exploits prefix routing as suggested in Plaxton et al. (1997) to route messages between nodes. Nodes and keys have unique, uniformly assigned identifiers of 128 bits with base B, where $B=2^b$ is a configuration parameter of the algorithm (typical value of $b=4$).

Topology: The topology of Pastry is similar to that of Chord in that the nodes are organized in a virtual ring in an ascending order according to their node identifiers ($0 \leq \overline{id} \leq 2^{128-1}$). Every node of the Pastry overlay maintains three structures, namely a routing table, a neighborhood set and a leaf set. The routing table has $\log_B N$ rows and $B-1$ columns, where every cell of row n contains routing information, e.g. IP addresses, of all nodes whose identifier has the same n first digits as the current node. The neighborhood set contains all nodes that are of close proximity to the current node (typical value of set is $2*B$). Lastly, the leaf set L contains the $|L|/2$ nodes with numerically closest larger identifiers than the current node and respectively the $|L|/2$ nodes with numerically closest smaller identifiers (typical value for $|L|$ is B). Upon arrival of a new node in the Pastry overlay it contacts a well known bootstrap node in the existing overlay network. The bootstrap node returns to the new node a list of close nodes in order to choose one at random to contact and send it its join request with its identifier as the key (the new node also initializes its neighborhood set from that of the closely located node). The node of the Pastry ring whose identifier is numerically closest to that of the new node receives this join request and replies by sending its routing table (the same for all other nodes in the routing path of the join request). Finally, the new node initializes its routing table based on the received information and notifies all interested nodes of its arrival to accordingly update their 3 main Pastry structures. When a node leaves the Pastry overlay its neighbors will contact the remaining members of their neighborhood sets and proceed in exchanging these sets with each other in order to find alternative closely located nodes to replace the departed one with.

Lookup: Data items and resources in Pastry are also assumed to have a unique identifier drawn from the same identifier space as that for the nodes, i.e. with base B. Keys are handled by nodes whose identifiers are numerically close to each other. In order to find the node that is responsible for a specific key, Pastry nodes forward lookup queries to nodes whose identifier shares with the key a prefix of at least 1 bit longer than with that of the current node. Thus, the node whose identifier is closest numerically to the key is located in a progressive manner.

Performance: In a Pastry overlay according to Rowstron and Druschel (2001) a key can be located in up to $\log_B N$ overlay hops (equal to the number of rows in the routing table). Support for scalability is a major advantage of Pastry, with relevant experiments showing that it works efficiently for networks of even 100,000 nodes. In terms of fault tolerance, Pastry behaves relatively well since single or small group node departures do not affect its operation. Nonetheless, when groups of at least $|L|/2$ nodes depart the overlay routing functionality is subject to failure.

3.4. Tapestry

The Tapestry (Zhao et al., 2004) P2P overlay bears many similarities with Pastry mainly in terms of their common routing infrastructure that is based in both cases on the Plaxton mesh (Plaxton et al., 1997). Tapestry is completely decentralized and allows for efficient location of objects and routing amongst a collection of nodes independently of their physical location. Moreover, Tapestry has a clear focus on scalability and fault tolerance under dynamic network conditions. In Tapestry nodes and objects are assigned unique identifiers of base β from the same 160-bit identifier space using a uniform distribution. Tapestry differentiates itself from Pastry in that it replicates data objects for redundancy and in the definition of network locality.

Topology: Tapestry overlays aim at retaining a low network stretch. Every node has information, e.g. node identifiers and IP addresses, only about its direct neighbors in the form of neighbor maps, i.e. local tables that are formed in a manner similar to longest prefix routing also used by Pastry. Each row of a neighbor map contains information about the node's neighbors that share a prefix with that node of length equal to the row's level in the map. Routing occurs in Tapestry on a digit by digit manner, up until the exact identifier of the destination is found. Since in dynamic environments connectivity between nodes cannot be ensured, a mechanism is needed to augment in terms of reliability this process in the case of node churn. In Tapestry this is achieved by means of redundant routing, where every node maintains for every entry in its neighbor map two additional backups with the same properties, i.e. sharing the same amount of digits in their identifier prefix, as the original entry. Every node also maintains a list of backpointers, namely nodes that point at it through their neighbor maps. When a new node joins a Tapestry overlay the so-called *need-to-know* nodes are informed of the new arrival, since this can complete an entry in their neighbor maps. Moreover, the new node based on its identifier might need to become responsible, i.e. root node, for existing identifiers and therefore corresponding object references need to be moved to the new node. Lastly, the Tapestry algorithm is required to complete a new neighbor map for the newly arrived node, while nearby nodes might consider the new node in their neighbor maps as an optimization since its identifier might better fit their needs. Similar procedures are followed in case of node departure.

Lookup: To publish objects, the node whose identifier is the same or the one closest to that of an object's identifier is the node who is assigned to be the root node of that object (the owner sends a dedicated Tapestry publish object message). The root node holds information about the original owner of the object, in order to be able to return it to any requesting nodes. Nodes between the owner of the object and the root of that object that receive the publish message, retain this information to promote faster lookups. Resource discovery therefore happens either by progressively (digit by digit) contacting the root node of the resource object or by encountering a node that holds information about a particular resource's owner.

Performance: Routing (and thus lookup) in Tapestry occurs in at most $\log_{\beta} N$ hops, where N is the size of the identifier space. According to the extensive simulation results presented in Zhao et al. (2004), Tapestry performs efficiently even under dynamic conditions, e.g. node churn, with only around 5% of lookup queries failing. Additionally, routing performs really well since the stretch factor for locating resources remains low, starting from 3 to eventually approach 1. Tapestry's significant popularity has been spurred by its lightweight, cross-platform implementation called Chimera.[1]

3.5. Kademlia

Kademlia (Maymounkov and Maziéres, 2002) is a structured, fully decentralized P2P overlay. While it also utilizes consistent hashing to map identifiers to keys and nodes, it nonetheless exploits a XOR-based metric to compute the distance, i.e. the closeness, between identifiers. This facilitates formal proof of the properties of the overlay algorithm compared to other DHT-based overlays. Moreover, Kademlia was designed based on observations regarding existing P2P overlays' operation, namely in Saroiu et al. (2001) it was observed that the longer a node remains connected to an overlay the higher the probability that it will remain connected for at least another hour. Accordingly, Kademlia promotes the use of "old" nodes in the overlay as neighbors by other nodes in order to ensure the overlay's connectivity with higher probability.

Topology: Both nodes and resources are assigned an identifier from a 160-bit namespace. The distance between two identifiers is defined as the integer value of their XOR. Each node maintains lists of nodes of distance between 2^i and 2^{i+1} from itself, where $0 \preceq i \prec 160$, called k-buckets because their size can grow up to k (typically 20). Each k-bucket is actually a sorted list of nodes satisfying the distance criterion, where the least seen node is at the top of the list. k-buckets are updated upon receipt of messages from other nodes: in case of an already seen node its order in the k-bucket is updated, i.e. it is moved to the end of the list, whereas in case of a previously unknown node the k-bucket is checked to see whether there is space to add an additional identifier. Upon positive confirmation the new node's information is stored at the end of the relevant k-bucket, while when the k-bucket is full the least seen node is pinged to check its status. If it responds the new node's information is discarded, conversely the new node is entered at the end of the list and the node that did not respond to the ping is discarded. This way of maintaining k-buckets promotes "old" nodes as neighbors in the overlay to enhance stability. A new node joining the overlay is entered in the appropriate k-bucket of a bootstrap node and it performs a lookup procedure for its own identifier to collect information on other nodes and thus populates its k-buckets accordingly, as well as enters itself into other nodes' k-buckets when appropriate.

Lookup: When a certain node identifier is requested, the protocol actually locates the k closest nodes to the given identifier. The initiator of the lookup first checks in its own corresponding k-bucket for the closest α nodes (typical value for α is 3). Then it sends similar requests to these α nodes in parallel so that they can return nodes that are close to the requested identifier according to the entries in their k-buckets. Qualifying nodes are recursively contacted to collect further nodes that are close to the requested one up until k nodes have been collected at the initiator. Kademlia allows for redundancy and caching to promote lookup efficiency (keys about resources can be stored to up to k different nodes in the overlay) and proper operation even under dynamic conditions, e.g. node failures or churn.

Performance: Similar to other DHT-based structured P2P overlays, Kademlia has a performance of $O(\log N)$ in terms of resource discovery, where N is the number of nodes in the overlay. Kademlia is the most widely deployed and most popular structured P2P overlay (Steiner et al., 2007), which has found applicability in commercial and open-source P2P implementations such as eMule/eDonkey and Bittorrent.

3.6. Viceroy

In the decentralized Viceroy P2P overlay network (Malkhi et al., 2002) the focus is on the topology in order to maintain a connection graph between participating nodes that has beneficial properties such as constant degree (to avoid hubs and thus bottlenecks in the overlay), low maintenance costs (small number of messages exchanges when nodes join or leave the overlay) and bounded path length for lookup operations (to promote efficiency of resource discovery). Accordingly, the Viceroy topology has a logarithmic diameter and approximates a butterfly network.

Topology: Similar to Chord and other DHT-based approaches, Viceroy too uses consistent hashing (Karger et al., 1997) to map node and resource identifiers to the same identifier space, this space being the unit ring $[0, 1)$. A node is responsible for all resources whose identifier is smaller than its own and larger than its immediate neighbor in the ring (resources mapped to their successor in the ring as in Chord). In order to achieve the aforementioned properties the ring is augmented with additional connections between nodes. First, all nodes are connected to their successor and predecessor in the ring. Moreover, inspired by previous work on small world networks (Kleinberg, 2000) and long range network contacts (Barriére et al., 2001), each node first selects a level l uniformly at random from a possible $\log N$ levels, where N is the total number of nodes, and then creates additional links to five nodes as follows. Two downward links are created to two nodes of level $l+1$, one of which to a node at a distance approximately $1/2^l$ away (right-connection) and one to a node with the closest distance (left connection). Additionally, one upward link is created to a node of level $l-1$ that is at the closest distance (except when $l=1$). Lastly, two more links are created to the previous and following nodes of level l. With this construction every node in the overlay topology has always 7 neighbors, thus achieving the goal of constant degree and minimizing the modifications needed when nodes join or leave the overlay. When a node leaves the overlay, its successor in the ring takes over its resources and all interested nodes are informed to update their connections accordingly.

Lookup: To locate a resource in Viceroy the node requesting a resource first traverses the level hierarchy in an upward direction,

[1] http://current.cs.ucsb.edu/projects/chimera/

namely it uses the upward connections to reach a node at level 1. In the next step the node of level 1 (and progressively nodes of lower levels) makes use of one of its downward links according to its vicinity to the requested resource's identifier. If the latter is greater than $1/2^l$ then the downward left connection is used, otherwise the right one. This procedure continues up until a node is found that has no downward links and its identifier is close to that of the request resource. In this case, a so-called vicinity search is performed using the ring connections and the ones to the nodes of the same level until the owner of the resource is located.

Performance: In Viceroy the lookup and routing procedures have a complexity of $O(\log N)$ similar to other DHT-based overlays. The great benefit of Viceroy is the fact that it avoids congestions by means of hubs, by ensuring load balancing in terms of connections to other nodes. The upper bound of 7 in the node degree in Viceroy overlay graph also assists in maintenance operations, by limiting the number of changes needed in the topology upon node churn. The Georoy algorithm proposed in Galluccio et al. (2007) is based on Viceroy, but extends the topology by adopting a two-tier, hierarchical architecture where high-level nodes serve as resource indices, and utilizes location information to map resources to nodes to promote lookup efficiency.

3.7. P-Grid

The major design goal of the P-Grid (Peer-Grid) P2P overlay (Aberer et al., 2003, 2002) was to facilitate lookup operation across large-scale distributed systems, while provisioning for resilience under dynamic conditions by means of resources' replication. P-Grid is completely decentralized and promotes load balancing of nodes in regard to the resources they are in charge of and therefore the queries they need to respond to.

Topology: The topology of the P-Grid overlay is essentially a virtual binary tree, where nodes of the P2P overlay are represented by leafs of the tree and the root and other parent nodes of the tree are used only for traversal, organization and maintenance of the overlay, i.e. they do not represent active nodes holding resources. In P-Grid each node assumes a position in the tree the path of which (binary string from root to leaf) reflects the information/resources that node is in charge of. In Aberer et al. (2002) a method for mapping filenames to unique and uniformly distributed keys is presented, while it should be noted that for redundancy reasons and to promote more efficient resource retrievals it is possible for more than one nodes to share the same path in the tree. Construction and maintenance of the P-Grid overlay relies on nodes' interactions. Initially, each node is responsible for all resources' identifiers. When two nodes meet they split the identifier space between themselves and each of them records a pointer to the other node in order to be able to satisfy requests for resources. This is how the tree grows and expands to progressively cover more specialized paths, i.e. binary string identifiers of resources. Therefore, for every bit in its path each peer holds references to the nodes that are responsible for the other side of the binary tree at the same level. Since this process is decentralized it is possible that two nodes that share the same path might meet, in which case once again the identifier space is divided between the two as before. There are also procedures to help balance the overlay tree and thus ensure load balancing in the P-Grid overlay.

Lookup: When a node receives a request for a resource it first computes the binary string identifier of the resource and if this request cannot be successfully satisfied by the node then it is forwarded to one of the nodes that the current node holds references to. In this way the request is being progressively forwarded to nodes that are closer to the resource, due to the construction algorithm of the P-Grid virtual binary tree. When more than one nodes share the same prefix then the selection of the next node to route the request to is done randomly. When no other node closer to the request resource's identifier can be found then the last node in the lookup path checks its local archive of resources in order to either satisfy or stop the request.

Performance: Similar to other DHT-based structured overlays, P-Grid has also a $O(\log_2 N)$ complexity when it comes to lookup of resources, where N is the number of resources. P-Grid performs well under dynamic conditions and generates far fewer messages than similar systems. While its tree based structure shares commonalities with Plaxton-like structures as seen in Pastry and Tapestry, P-Grid differs in that it allows for advanced types of queries (range, Datta et al., 2005 or similarity, Karnstedt et al., 2006, queries are possible) and that less strict assumptions over participating nodes are required (higher degree of autonomy and self-organization).

3.8. SkipNet

The SkipNet overlay network (Harvey et al., 2003) is a distributed, decentralized structured P2P overlay that draws inspiration from the Skip Lists data structure (Pugh, 1990), similar to the work discussed in Aspnes and Shah (2007). SkipNet, conversely to other DHT-based approaches, allows for controlled data placement. Moreover, it promotes data and routing path locality and thus improves availability, performance and security. It supports two identifier namespaces that are used in parallel, i.e. a string namespace for nodes and resources and a numeric identifier namespace that is derived by hashing the corresponding identifiers from the string namespace.

Topology: The topology of SkipNet is that of a ring where the string names of the participating nodes are placed in a sorted ascending order according to their string identifiers; the ring is doubly linked and each node maintains pointers to the previous and next nodes in the ring, as well as $2*\log N$ (N is the number of nodes) additional pointers to other remote nodes on the ring stored in the the node's routing table. The latter has $\log N$ levels and two directions since the list is doubly linked (hence the $2*\log N$ numbers of pointers per node) and is constructed as follows. For a given level h of the routing table, its two entries contain the nodes that are approximately $2*h$ nodes to the left or right of the current node respectively. At level h the ring is essentially split into two disjoint rings of level $h+1$ each one of which is formed by every second node of the corresponding h level ring. To maintain such a structure would require a lot of management messages being passed around, a drawback which motivated the need to relax this topology construction by allowing nodes of level h to uniformly and randomly choose in which of the two disjoint rings of level $h+1$ they will be assigned to. With such an approach nodes joining or leaving the overlay influence the routing tables of only two other nodes in each of the rings that they belong to. When nodes join the SkipNet overlay they first have to find the highest level ring to which they belong based on their numeric identifier. Then the new node discovers the other nodes that are part of the same ring by looking for them by their string identifier and starting from one of these neighbors it progressively follows the path to the ring of level 0 and in the process it inserts itself to every qualifying ring.

Lookup: When attempting to discover resources by their string name, the SkipNet is traversed to the direction closer to the requested identifier. This is a rather simple process because the SkipNet ring is constructed to be sorted according the string namespace. To look for resources by their numeric identifier, the first node in level 0 whose numeric identifier matches the first digit of the one requested is located and then the process moves on to this node's level 1 ring. The process is iteratively repeated until no more progress can be made, in which case one of the nodes in the last visited ring is the one that the request was made for.

Performance: SkipNet, as most of structured P2P overlays, has a complexity of $O(\log N)$ when it comes to locating resources and routing messages in an overlay of N nodes. It should be highlighted that the controlled data placement of SkipNet actually contradicts the beneficial load balancing features that are inherent of DHTs, a constraint that has been pointed out in Harvey et al. (2003). It nonetheless has very good performance in dynamic situations and addresses a security concern that is present in most DHT-based systems and that involves placing resources on untrustworthy nodes.

3.9. Other approaches

For the sake of reference, we briefly summarize a few other indicative research efforts towards building and maintaining structured P2P overlays, while it should be noted that this list is far from exhaustive.

- *Cycloid* (Shen et al., 2006): Similar to Viceroy, Cycloid is a constant degree DHT-based P2P overlay, the topology of which is that of a cube connected cycles graph. Each node has 7 connections to other nodes and the lookup complexity is $O(d)$ where d is a constant such that for the N nodes in the overlay the following stands $N = d * 2^d$.
- *Skip Graphs* (Aspnes and Shah, 2007): Structured P2P overlay that does not consider consistent hashing and operates with logarithmic complexity, as well as supporting complex queries and fault tolerance for even large groups of nodes.
- *Z-Ring* (Lian et al., 2005): It is an extension of Pastry (using a base of 4096) to accommodate 1-hop and 2-hop information retrieval by increasing the number of connections of nodes, while at the same time retaining a low overhead in terms of maintenance operations. The latter is achieved by means of efficient membership protocols that guarantee message exchanges between group members alone.
- *Ulysses* (Kumar et al., 2004): A DHT-based, structured P2P overlay that considers the butterfly topology with some modifications in order to limit the diameter of the overlay to $\log N/\log(\log N)$ thus simultaneously improving the performance of resource discovery without sacrificing robustness.
- *D2B* (Fraigniaud and Gauron, 2006): Exploiting de Bruijn graph topologies, this structured P2P overlay exhibits beneficial properties in handling node churn under dynamic conditions. The D2B overlay has a constant degree and similar to other DHT-based overlays its complexity is $O(\log N)$.
- *D1HT* (Monnerat and Amorim, 2006): Specialization of DHT-based P2P overlays that exploits 1-hop DHTs to locate resources, thus greatly increasing resource discovery performance, albeit at a reasonable cost in terms of traffic overhead.
- *Koorde* (Kaashoek and Karger, 2003): This DHT-based P2P overlay is built on Chord and makes use of de Bruijn graphs to form a robust topology that allows for efficient and flexible information retrieval. When each node is connected to only 2 neighbors, lookup procedures have a complexity of $O(\log N)$, whereas when the number of neighbors grows to $O(\log N)$ as in typical overlays like Chord, lookup procedures take approximately $O(\log N/\log(\log N))$ hops.
- *HyPeer* (Serbu et al., 2011): Structured P2P overlay that extends the traditional ring structure of Chord-like overlays to that of a hybercube in order to provide redundant paths for lookup and routing and eventually to support flexible routing strategies.
- *BATON* (Jagadish et al., 2005): Structured P2P overlay following a balanced binary tree topology and having a complexity of $O(\log_N)$ (N is the number of nodes), as well as supporting both exact match and range lookup queries.
- *Coral* (Freedman et al., 2004): P2P content distribution network that shares principles with standard DHT-based, structured overlay networks taking into account clusters of well-connected nodes and relaxing the notion of consistency in hashing by introducing distributed sloppy hash tables.

3.10. Discussion

The majority of the structured solutions that have been introduced and were summarized before have a logarithmic performance ($\log N$ for N nodes in the overlay) in terms of lookup requests and routing, namely they require a logarithmic number of steps to locate the node where a requested resource or service resides. This parameter, i.e. network diameter, is directly influenced by the routing table of participating nodes as discussed in Xu et al. (2006) and explained in the preceding description of the various structured P2P overlays. In general, a small diameter (maximum number of hops), which is common in most structured P2P overlays, allows for efficient resource discovery. Chord, Kademlia, Viceroy, P-Grid and SkipNet all exhibit a $\log N$ lookup performance, whereas Pastry and Tapestry also have logarithmic performance of $\log_B N$ that is however linked to the base B of the considered identifier space. CAN's performance is not linked to the number of nodes in the overlay ($(d/4) * n^{1/d}$), but instead to the number of neighbors of a node (d) and the total number of zones in the d-dimensional zones (n). The relatively small values for lookup that are inherent in structured overlays constitute a desirable feature in pervasive environments, since they promote efficient and quick access to resources. Due to the dynamicity of pervasive environments that translates into volatility regarding network connections, fast resource discovery becomes paramount. Therefore, structured overlays are very promising in handling such adverse behavior effectively.

One of the biggest advantages of such overlays can be found in the DHT consistent hashing algorithms used to map resources/services on specific nodes. In particular, these algorithms allow for uniform distribution of resources across the nodes that participate in the P2P overlay and thus promote load balancing in managing resources and handling lookup and routing requests. If only a few nodes are tasked with hosting large numbers of resources then they would automatically be construed as being possible single points of failures and prone to attacks by malicious users. Load balancing is a very desirable feature since it promotes availability, accessibility, scalability and reduces the occurrence of bottlenecks in the overlay. Such features are quite beneficial especially in the case of pervasive environments with their particular characteristics in terms of heterogeneity, scalability and dynamicity.

Nevertheless, hashing algorithms suffer from the need to reorganize resources when changes in the overlay occur. Reorganization operations are quite costly in terms of processing and communication overhead, especially in Chord and CAN that have very rigid structures and rules. In Kademlia, the notion of redundancy in terms of resource placement further improves the performance of the hashing algorithms and of lookup operations, albeit at the cost of a cumbersome process. The particular design of Viceroy is noteworthy in that it inherently avoids the generation of hubs, while additionally limiting the number of changes in the topology to 7 when the need for reconfiguration arises. Pervasive environments with their dynamicity and volatility could thus greatly benefit from this behavior. Conversely, SkipNet is a structured overlay that does not make use of hashing algorithms to place resources on nodes, but instead is based on controlled data placement. This can lead to load balancing issues where few

Table 2
Taxonomy of structured P2P overlays for pervasive environments.

P2P	Lookup	Dynamicity	Redundancy	Churn	Security	Queries	Load balance
Chord	$O(\log N)$	No particular support	N/A	Performance drops and high overhead	N/A	Standard queries	Consistent hashing
CAN	$(d/4)*n^{1/d}$	Not failsafe from partitions	Routing redundancy	Resilience is a factor of node degree	N/A	Standard queries	N/A
Pastry	$O(\log BN)$	No particular support	N/A	Robust for few departures	N/A	Standard queries	Consistent hashing
Tapestry	$O(\log \beta N)$	Routing redundancy	Resource replication	Resource replication and routing redundancy	N/A	Standard queries	Consistent hashing
Kademlia	$O(\log N)$	Considers previous node behavior for routes' longevity	Resource replication and caching	Good support through redundancy	Old nodes more trusted	Standard and range queries	Consistent hashing
Viceroy	$O(\log N)$	Bounded degree and path length	No particular support	Low maintenance costs	N/A	Standard queries	Constant degree and consistent hashing
P-Grid	$O(\log N)$	Routing redundancy	Resource and routing replication	Self-organization of nodes	N/A	Standard, range and similarity queries	Consistent hashing and tree balancing
SkipNet	$O(\log N)$	Data and routing locality	Routing redundancy	Multiple alternate paths	Controlled data placement	Standard queries	N/A

nodes are in charge of many resources and can thus become overloaded in serving relevant requests. This adverse characteristic is also shared by CAN that due to its construction can have areas of different size and different resource count.

However, structured P2P overlays are broadly speaking not targeted for very dynamic networks where churn rates are extremely high. The reason for this can be found in the construction of such overlays that requires nodes and resources to be specifically placed in the overlay's topology by creating or updating existing connections of themselves and of other neighboring nodes. Therefore, when a new node joins the overlay or when an existing one departs, other nodes are also influenced, which in turn involves message exchanges and updates in overlay connectivity that require both time and overhead traffic. P2P overlays built on top of pervasive environments that are plagued with high rates of node arrivals/departures will thus have sub-optimal performance due to the constant updates in the overlay topology and in the relocation of resources across nodes in order to satisfy the rigid rules of the structured P2P algorithms (Rhea et al., 2004).

Moreover, when taking into account mobility of nodes and the instability of the underlying networking infrastructures of pervasive environments, it becomes clear that any P2P overlay built over such networks should take these issues into account. Structured P2P overlays generally do not consider the real network latency when building overlay links and this can therefore hinder efficient resource discovery and routing. However, the importance of this aspect has been considered by a number of P2P overlays, especially the ones based on Plaxton routing, e.g. Pastry, Tapestry and there have also been approaches adopting redundancy in overlay links and node connectivity to promote robustness and resilience even under dynamicity (Zhao et al., 2003). An in depth analysis of the influence of churn on both structured and unstructured P2P overlays can be found in Stutzbach and Rejaie (2006) where design implications for prospective churn-aware P2P overlays are elicited.

Structured P2P overlays based on DHT solutions are very efficient in locating specific resources, i.e. exact matching, since the consistent hashing methods that are applied ensure that the identifiers of the resources/services that have been requested can be uniquely mapped back to the hosting node. They nonetheless are not well-suited for range queries, e.g. all nodes whose remaining battery is above level x. These are typical queries in regard to pervasive environments, where it is most often the case that the most suitable service/resource is selected among the many available ones. Solutions like SkipNet allow for such queries in structured overlays, whereas further research has also proposed ways to enable range queries in DHT-based overlays, such as Huebsch et al. (2003), Li et al. (2005) and Bharambe et al. (2004), as well as multi-dimensional queries, e.g. the work in Ganesan et al. (2004). Of particular interest is the Armada system (Li et al., 2009) that allows for single- and multi-attribute range queries on constant degree DHT-based P2P overlays with a bounded delay, hence proving to be a quite efficient approach. P-Grid also allows for advanced types of queries that involve similarity or range. Nevertheless, unstructured protocols with their flooding-based discovery mechanisms are inherently more suitable for such queries.

Because of the dynamic nature of pervasive environments (examples), paths in the underlying networks are subject to changes. Any P2P overlay built on top of such environments should therefore take into account these changes. Structured overlays are in general in need of reconfiguration when changes have been detected and this could impose delays and hence diminish their performance. Accordingly, redundancy is the mechanism that is employed by structured overlays to ensure efficient operation even under dynamic conditions. Chord does not support redundancy as such, but there have been proposals to extend Chord by concurrently maintaining multiple rings to address this need (Flocchini et al., 2007). CAN instead does not explicitly support the notion of redundancy, or do Pastry, Viceroy, SkipNet and Tapestry. In Kademlia, redundancy is focal in placing resources in more than one node, hence allowing for failsafe operation on one hand and increasing lookup efficiency on the other hand by having multiple copies of a resource around the overlay. A similar approach is adopted by P-Grid by replicating resources across nodes in order to cope with dynamicity and increase load balancing and lookup efficiency.

Lastly, similar to every distributed system DHT-based structured overlays also suffer from security threats, such as the Sybil attack, the Eclipse attack and routing and storage attacks (Urdaneta et al., 2011). One of the main problems is the placement of resources on nodes according to hashing algorithms and random identifiers, hence risking that resources will be hosted by non-trusted nodes. SkipNet manages to alleviate such concerns by selectively placing resources, but at the cost of lack of load balancing as mentioned before. Since this is a topic that remains outside the scope of this work, we refer the interested reader to the aforementioned survey, as well as Wallach (2003) for further reading.

Summing up, in Table 2 we present (based on the detailed discussion that preceded) a comparison between the different structured P2P algorithms in regard to various features that are characteristic of pervasive environments, namely lookup performance, dynamicity, redundancy, churn, security, complex queries, load balancing.

4. Unstructured P2P overlays

In contrast to the rigid and tightly controlled topologies and placement of resources that characterize structured P2P overlays, unstructured ones are much more flexible in terms of node relationships and lookup operations. Node membership 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, i.e. exchanged messages overhead, as far as the topology is concerned. However, performance of lookup operations is not as efficient as in structured solutions because it is mainly based on flooding. Since such an approach clearly does not scale, optimized solutions for resource discovery in unstructured overlays have been proposed (Barjini et al., 2012). In what follows we discuss widely deployed unstructured P2P overlays with the goal of describing their topologies, lookup operations and their performance.

4.1. Freenet

Freenet (Clarke et al., 2001) (https://freenetproject.org/) is a distributed, unstructured P2P overlay network that was built in order to allow anonymous access to and handling of data (publication, replication and retrieval) over wide network infrastructures. The main design goal of protecting the privacy of data publishers is what drove the decision towards a completely decentralized and distributed approach, with no single point of failures. Additional design goals of Freenet include security and reliability that emerges as a necessity due to the dynamicity of the participating nodes, i.e. churn and failures. Despite the distributed nature of Freenet, appropriate mechanisms have been proposed to facilitate information retrieval in an efficient manner without resorting to flooding or the use of central resource indices.

Topology and Lookup: When new nodes join the overlay they just have to locate at least one existing node to connect to and thus become part of the topology. This inherently leads to a continuously evolving and changing topology. Each node in Freenet holds its own datastore, as well as a routing table to other Freenet nodes together with an index of their data. In terms of lookup, each data item has a unique key that can be computed by all nodes, e.g. by hashing some related descriptive text thus allowing for semantic similarities to be preserved. 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. Moreover, at every intermediate node of that path the data and the key will be replicated and the routing tables will be updated to reflect the node that responded positively to that request. This procedure ensures that progressively more information about the location of similar files will be propagated across the Freenet overlay thus eventually ensuring faster response times that span only few hops. This replication process also increases resilience in light of node failures. Broadcasting a request on the Freenet overlay is done in a selective manner by avoiding flooding and checking for duplicate data requests. Additionally, direct links are created between the requested and the responder, thus further promoting prospective lookup operations. A similar mechanism is used for inserting data, whereas there are also mechanisms to remove stale entries in routing tables and datastores.

Performance: Simulation results reported in Clarke et al. (2001) validate the performance of Freenet. In particular, average path length to satisfy resource requests relatively quickly drops to around 6 for a 1000 node overlay, thus proving the benefits of the replication mechanisms. In terms of scalability, simulation showed that this average path length approximately grows in a logarithmic manner, while additionally it remains in reasonable values even under high rates of node failures (up to 30% of the node population). This fault tolerance is mostly attributed to the fact that the Freenet overlay's topology closely resembles small-world network topologies with all their inherent robustness features due to the power law distribution of node links, as studied in Milgram (1967), Watts and Strogatz (1998), and Albert et al. (2000).

4.2. Gnutella

Gnutella (2002) is one of the most widely known and popular peer-to-peer systems under the unstructured category. Gnutella was designed to be a file sharing system based on an unstructured P2P overlay that allows for decentralized, scalable, reliable and anonymous sharing of files between participating nodes. Gnutella bases its popularity on its simplicity and its open and flexible membership, which however have both led to serious performance degradation issues (Ripeanu, 2001).

Topology and Lookup: To join Gnutella a node needs to contact an existing node of the overlay. A node in the overlay node can undertake three main operations, namely to locate and to connect to other nodes, to query for and to retrieve files, as well as to push files to other nodes that lie behind firewalls. These operations are supported by 5 different types of messages: the *Ping* message is sent from a node to indicate its availability and to collect information about other nodes, the *Pong* message is a reply to the *Ping* message, the *Query* message refers to search requests and related parameters, the *QueryHit* message is a positive reply to a received *Query* message and lastly the *Push* message is used to allow for file downloads from nodes that are behind firewalls. Actual file downloads are outside the scope of the Gnutella protocol and take place using standard HTTP GET requests. A node discovers other nodes in the overlay topology using combinations of *Ping–Pong* messages; the former are flooded across the network by a node, while the latter are routed back to the node that originally issued the *Ping* message via the same path. Due to network dynamicity there is a need for periodic initiation of the node discovery process to update node connections. Similarly, resource lookup requests are based on the exchange of *Query–QueryHit* messages. Lookup operations on this very flexible topology are thus based on pure flooding.

Performance: Scalability is the major concern in Gnutella, since it does not perform well in this regard as observed in Ritter (2001). Gnutella overlays exhibit properties that are consistent to power-law networks, similar to Freenet (Ripeanu, 2001). These properties promote the robustness and resilience of the Gnutella overlay in the case of node failures. However, if these node failures refer to core nodes of the power-law graph, then the results can be detrimental. This is an aspect that affects all such network topologies and can be adversely exploited by malicious attackers. Additionally, in terms of traffic overhead (Ripeanu, 2001) indicates high overhead in terms of *Ping–Pong* messages that can be up to 50% of the overall traffic attributed to Gnutella operations. In general, Gnutella generates high network traffic overhead due to its simplistic flooding operations. Therefore, there have been quite a few enhancements proposed mostly in the direction of imposing a hierarchy of nodes/supernodes, the latter being in charge of message forwarding and thus collectively playing the role of the network's backbone, e.g. ultrapeers introduced in version 0.6 of Gnutella. In the same direction, in Castro et al. (2004) a structured topology for node connections was considered, while retaining standard Gnutella functionalities for file placement and lookups. This approach managed to achieve lower traffic overhead compared to standard Gnutella. This result is counter-intuitive, since it is generally accepted that structured overlays incur more overhead for their maintenance compared to unstructured ones. Nonetheless, it appears that the randomness involved in flooding operations counters this effect.

4.3. FastTrack

The FastTrack unstructured overlay has been used by a number of file sharing applications such as KaZaa and iMesh. Whereas its internal operations have not been formally documented, Liang et al. (2006) provided a solid understanding of the actual operation of the FastTrack overlay by means of a thorough measurement study. This is to date the main source of reference for FastTrack together with the efforts of an open-source project that reverse engineered part of the protocol.[2]

Topology and Lookup: FastTrack is fully decentralized and its topology is hierarchical. Taking into account the inherent heterogeneity of nodes in terms of computing capabilities (memory, processing, networking, etc.), FastTrack makes a distinction between highly capable supernodes and ordinary nodes. The latter are connected to just one supernode at a time and it is this supernode that gives the ordinary nodes access to the overlay. Supernodes are connected between them, but this is far from a full graph; instead as reported through measurements in Liang et al. (2006) supernode connections are very sparse. The overlay thus has a decentralized two-tier hierarchical topology and there are no reported mechanisms for its maintenance in case of node failures. In terms of bootstrapping, each node is assumed to have a local supernode list that also includes information about their workload. Using this information a new node selects the most appropriate supernode to connect to, based on its locality (preference given to nearby neighbors) and its workload (for load balancing reasons, less loaded nodes are preferred). Periodically this selection changes to reflect changes in the overlay and hence the supernode-to-overnode associations and subsequently the topology of FastTrack is subject to very high dynamics. 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 supernode. This ensures that supernodes can collectively answer queries about resources of all their underlying ordinary nodes. Queries are propagated between supernodes to enhance their coverage of the entire overlay and increase the possibility of getting a successful response.

Performance: In the “live” FastTrack overlay there seems to be a convergence in the number of active supernode-to-supernode connections to around 40–50 and each supernode has on average between 100 and 160 associated ordinary nodes. This workload would definitely undermine the performance and longevity of supernodes, which is probably the reason for their relatively small lifetime (around 2.5 h). Moreover, in terms of lookup performance Liang et al. (2006) imply that this is not quite satisfactory and proposes enhancements, such as allowing supernodes to exchange their resource indices among each other to increase query coverage (the selection of supernodes according to the locality criterion leads to localized search results) and speed up query response times.

4.4. BitTorrent

BitTorrent (Cohen, 2003) is one of the most popular P2P file-sharing systems available nowadays with millions of active users. While the BitTorrent protocol specification[3] and Cohen (2003) do not classify BitTorrent as an unstructured P2P overlay, we assert that the connections between peers and their adaptive management does indeed create a P2P overlay network topology. Since the latter is not built on any strict rules, but is rather subject to random and uncoordinated nodes’ activities, we classify the BitTorrent P2P overlay as an unstructured one.

Topology and Lookup: BitTorrent is a centralized unstructured P2P overlay, since a node needs to connect to a specific entry point to join the overlay. In particular, when a user wishes to download/upload a file (at least one other node needs to upload the complete file at all times) it launches the BitTorrent application by executing the corresponding *.torrent* file, which holds information about the file and a URL to its associated tracker. The tracker is essentially the central point of entry to the P2P overlay concerning the specific file, since it contains information about all the other nodes that are currently downloading or uploading that file. The tracker sends a random list of other nodes to a downloader, to which the latter connects to. It should be noted that a downloader connects to more than one other nodes, since files are partitioned in smaller parts that can be retrieved in parallel from different sources to facilitate and speed up file downloads. These node connections for all the available files built up the topology of the BitTorrent overlay. No management operations are explicitly mentioned in the protocol's specification other than the joining of the overlay; when nodes depart the overlay their active connections are cancelled and other nodes that were connected to them find – by accessing the trackers – other nodes to connect to in order to complete their downloads. BitTorrent implements, in the context of maintaining fairness between uploading and downloading, a choking algorithm to temporarily block uploading operations based on a tit-for-tat analogy. These chocking operations nonetheless do not generally affect connectivity. Lookup procedures are centralized and occur via the tracker. When nodes with the complete file start uploading it, then they get registered to the tracker and are thus accessible to other nodes. When nodes start downloading pieces of a file, then they become immediately available to other nodes as uploaders of these pieces of files and they therefore are registered in the trackers accordingly. This process promotes the efficiency of lookup procedures.

Performance: BitTorrent, being a “live” P2P overlay with no formal specification, suffers similar to FastTrack from a lack of analytical measurements regarding its performance. Nonetheless, there have been some studies to analyze the performance of BitTorrent-like P2P overlays, e.g. (Qiu and Srikant, 2004). In the latter study a probablistic fluid model for BitTorrent overlays was proposed and based on it the scalability of BitTorrent and its efficient operation in terms of file downloading in the context of diverse uploading/downloading capabilities of nodes was validated.

4.5. UMM

UMM (Unstructured Multisource Multicast) (Ripeanu et al., 2010) is an unstructured P2P overlay that aims at addressing the problem of group communications by means of multicasting. UMM is fully distributed and is targeted at dynamic environments.

Topology and Lookup: The bootstrap procedure of UMM is similar to that of Gnutella, namely a new node contacts an existing one that provides the former with additional identifiers of other overlay nodes to connect to. Once a node is part of the overlay, it can partake in the topology maintenance and lookup operations. UMM adheres to a self-organized principle in that every node

[2] http://developer.berlios.de/projects/gift-fasttrack/ [accessed May 2013].

[3] Protocol specification: http://www.bittorrent.org/beps/bep_0003.html [accessed May 2013].

makes localized decisions on the overlay topology. Collectively these local decisions on topology reconfiguration lead to a progressive optimization of the entire overlay. In that aspect UMM is very similar to bio-inspired approaches for P2P overlays that will be presented later. In particular, nodes in UMM maintain two types of connections to other nodes, i.e. short and long ones, which have a 1–1 ratio and are optimized in terms of latency and bandwidth respectively. Each node manages its connections independently of other nodes and continuously optimizes them in order to reflect potential changes in the underlay networks. Lookup operations in UMM correspond to dissemination over multicast trees. The latter are created and maintained using implicit information that has been derived from redundant flooding messages sent around the overlay. UMM utilizes such information to infer the optimal paths in the overlay and thus generate the corresponding multicast trees. Lastly, UMM considers partitions and node failures by introducing some basic mechanisms to address these adversities. To this end, some special nodes in the overlay are considered that emit heartbeat signals to be received by all nodes in the overlay at a certain period. If a node does not receive any heartbeat over a course of time, then it enters the repair mode of operation and tries to re-connect to the overlay by contacting the heartbeat nodes from which no signal has been received.

Performance: UMM has been extensively evaluated in comparison to other unstructured and P2P overlays using both simulations and PlanetLab emulations, with relevant results appearing in Ripeanu et al. (2010). The results support the initial design goals set by the designers of UMM, indicating good performance in terms of multicast information dissemination efficiency and adaptation, while retaining a relatively low overhead. The main strength of UMM lies in its simplicity that is attributed to its self-organized design. Nonetheless, UMM does not bode well as far as scalability is concerned with experiments considering networks of only up to 1024 nodes.

4.6. Gia

Building on Gnutella 0.6 and utilizing the concept of supernodes introduced in FastTrack, Gia (Chawathe et al., 2003) is a decentralized, unstructured P2P overlay with a clear aim at supporting scalability and thus addressing the relevant shortcomings of Gnutella.

Topology and Lookup: A new node needs to contact an existing node of the overlay to connect to and thus join the overlay. Nodes in Gia are not all considered to be equal, but instead are classified according to their capacity levels taking into account a node's computing capabilities, access bandwidth, etc. Gia works on an adaptive topology adaptation algorithm that ensures that high capacity nodes become central to the topology by having higher peer degree compared to low capacity nodes. In this manner, the heterogeneity of P2P overlays is efficiently accommodated; Gnutella-like protocols that exhibit power law behavior have a number of well-connected nodes in their topologies, which inadvertently become overloaded. By promoting high-capacity nodes in this role, Gia ensures that the effects of node overload are minimized. The topology adaptation algorithm allows for continuous optimization: all nodes periodically monitor their neighbor connections and individually decide on whether all of them are going to be retained or some of them will be replaced with other neighbors to improve the quality of the neighbors set. This localized decision-making process resembles self-organization principles found in bio-inspired approaches. As far as lookup operations are concerned, the notion of flow control is introduced (proactive mechanism to avoid node overloading by allowing nodes to send a message only when its intended receiver has authorized this action by means of a token), as well as the replication of content indices between neighbors. The latter mechanism is exploited by the lookup protocol that utilizes biased random walks, i.e. lookup queries are not flooded in the overlay, they are rather forwarded to the highest capacity neighbor of a node. Since nodes with high capacity have by design a higher degree, they will have a higher probability to respond to a query since they will have access to the content indices of a large number of other nodes aside to their own. This mechanism is inspired by prior work (Lv et al., 2002; Adamic et al., 2001) that suggested the use of random walks and biased random walks towards high-degree nodes respectively.

Performance: The performance of Gia compared to Gnutella according to simulation studies reported in Chawathe et al. (2003) consist in improvements of 3–5 orders of magnitude in regards to the total capacity of the system. Moreover, lookup operations are also greatly improved and the proposed design is quite robust to failures. This is extremely important since power-law networks suffer from such a problem; however, this does not seem to be an issue with Gia, due to its continuous optimization topology adaptation algorithm. It is a matter of discussion whether this design can indeed promote the longevity and optimal performance of the overlay's topology, since even high capacity nodes can become overloaded if they are constantly over-utilized.

4.7. Phenix

Phenix (Wouhaybi and Campbell, 2004) is a fully decentralized, unstructured P2P overlay with the goal of maintaining low-diameter topologies. The motivation behind Phenix lies in the construction of topologies, the degree distribution of which is a power-law one. In this respect, the resulting overlay topologies emerge as being of low diameter, while the potential reliability concerns of such topologies (e.g. attacks on popular, high-degree nodes) are alleviated by means of a series of mechanisms.

Topology and Lookup: Similar to UMM, Phenix operates on localized decision making in determining high degree nodes in the topology. A new node acquires access to the overlay by contacting some of its existing nodes. The algorithm is based on the principle that every node maintains its own neighbor list and the process of doing so ensures the creation of power-law topologies as proven in Wouhaybi and Campbell (2004). In particular, a node that joins the overlay acquires through out-of-bands means a list of potential neighbors, which it splits in two, namely the *random* and *friends* subsets. It then contacts the nodes in the *friends* subset and expects a reply from each one of them containing a list of their own neighbors, whereas they also contact these neighbors on behalf of the original node in order to have the latter added to a special list called Γ that is maintained locally on each of those nodes. The original node, upon receiving the lists of neighbors from the nodes in the *friends* subset it merges them and sorts the output according to the number of times a node appears, i.e. most well-connected nodes will appear first. It then produces the *preferred* subset from the highest ranked nodes in the previous list. The combination of nodes in the *random* and *preferred* subsets yields the neighbors list of the newly arrived node. The neighbors are continuously maintained in order to ensure that the network overlay remains operational and well-functioning even in the case of node failures. Additionally, to further enhance resiliency of the overlay, nodes also maintain a *backward* subset of nodes, which for a given node are composed of its nodes in the *preferred* subset that have the given node in their Γ list. This mechanism ensures bidirectional connectivity between these pairs of nodes and hence more failsafe operation. The Phenix overlay does not consider lookup operations per se.

Performance: Phenix is expected to have a larger overhead in terms of traffic compared to Gnutella, which also exhibits power-law graph properties while having a simpler design than Phenix. Nonetheless, the merits of Phenix appear in regards to its behavior under attacks as reported in Wouhaybi and Campbell (2004): its resilience is very good in terms of both node mobility and node failures. The latter is mainly attributed to the special mechanisms introduced in Phenix to handle the adverse performance of power-law graphs under dynamicity. Moreover, Phenix exhibited remarkable connectivity (i.e. lack of overlay partitions) even under high rates of node failures in PlanetLab emulations.

4.8. Other approaches

For the sake of reference, we briefly summarize a few other indicative research efforts towards building and maintaining unstructured P2P overlays, while it should be noted that this list is far from exhaustive.

- *Newscast* (Jelasity and van Steen, 2002): This overlay is used for large-scale and multicast information dissemination across the Internet. The main principle is retaining simplicity and reducing message exchanges by lazily propagating node membership and connectivity information, along disseminated information. It is based on evolutionary computation and it utilizes agents to do computations and assist in communications, which are both the reasons why this approach could also be classified as bio-inspired one. Additionally, the notion of probabilistic behavior is prominent, with the plethora of agents ensuring failsafe and performant operation.
- *mOverlay* (Zhang et al., 2004): A feature usually ignored by most P2P overlays, i.e. node locality, is paramount in the design of topology of the mOverlay unstructured overlay. The experimental results presented in the related paper validate the performance improvements compared to similar overlays, mainly attributed to the limited overhead that is achieved due the adopted principle of neighbors' locality that is applied by grouping closely located nodes in clusters. A similar unstructured P2P overlay is the focus of the more recent study (Hsiao et al., 2009).
- *Saxons* (Shen, 2004): It is an application layer unstructured P2P overlay that was built with the threefold goal of achieving small average path length, low path latency and high bandwidth for overlay links. Saxons can also operate on top of other P2P overlays and it is based on the observation that not all links between nodes in the overlay need to be managed at all times. In this respect, it allows us to dynamically adjust the set of active overlay connections to achieve the desired performance.
- *PROSA* (Carchiolo et al., 2010): Inspired by the way people behavior in social dynamic situations, PROSA is a semantic unstructured overlay the degree distribution of which resembles a power-law one. Node connections are established based on their common preferences thus creating clusters of nodes with common interests.
- *SLUP* (Sun et al., 2008): In SLUP both topological properties and semantic relationships between nodes and resources are taken into account in constructing the overlay topology, i.e. building and maintaining neighborhood relationships.
- *Foreseer* (Cai and Wang, 2004): To ameliorate efficiency of resource queries and overall system scalability, this overlay protocol utilizes locality information in both geographical and temporal contexts to build and maintain neighboring relationships between nodes in the overlay's topology. Moreover, distributed resource indices are exploited to improve lookup operations.
- *UDHT* (Puttaswamy and Zhao, 2007): In this work the interesting concept of applying a DHT structure on top of an unstructured overlay is studied. UDHT supports both complex, multi-range queries as well as ones for rare objects, thus leveraging the benefits of both main categories of P2P overlays.
- *PDG Superpeers* (Li and Chao, 2010): The two-tier hierarchical topology of superpeers and ordinary nodes found in this overlay is very similar to that of FastTrack and other hierarchical unstructured overlay. In this work however, the connections between superpeers are not random, but they rather form a perfect difference graph that has desired properties such as graph diameter of just 2 (see Parhami and Rakov, 2005 for further information). The experimental results report reduced message overhead and constrained overlay diameter.

4.9. Discussion

Contrary to the rigid rules set by structured overlays on topology maintenance and on the advertisement of resources, unstructured P2P overlays offer a much more flexible and open alternative. Each node that participates in the overlay is in charge of its own resources, whereas the topology is not tightly controlled and instead emerges dynamically as a result of the interactions between nodes, i.e. connections between them being constructed or dismantled according to localized node decision-making. This mostly distributed notion of management of the overlays' topologies and the general flexible node membership that is observed in unstructured P2P overlays, both constitute such approaches as highly applicable to pervasive environments. The reason for this lies in the fact that the latter environments are characterized by node dynamicity, heterogeneity and are subject to high degree of failures, thus inherently making them a good application domain for unstructured overlays. In terms of dynamicity, an analysis and evaluation of the performance of unstructured overlays under the influence of churn can be found in Baldoni et al. (2006), where the authors determine the breaking point of such overlays after which partitions occur consistently.

The majority of unstructured P2P overlays does not consider any particular mechanism for handling dynamicity, instead relying on the distributed nature of the topology management algorithms. Freenet and Gnutella both share common principles of operation to small-world network topologies and therefore the topologies' links follow a power law distribution with a few well-connected hubs. The distributed and decentralized construction and maintenance of Freenet and Gnutella allow for flexible membership and thus dynamicity is handled well. When nodes arrive or depart the overlay there is no particular need for a reconfiguration of the entire topology. Nonetheless, if the departing nodes are among the hubs then dynamicity will greatly affect the performance of both these P2P overlays. A similar behavior can be observed for FastTrack. However, in this case the hierarchical architecture with the connected supernodes and the replication of resources indices from the nodes to their corresponding supernodes facilitates proper handling of the adverse effects of dynamicity. In BitTorrent conversely, dynamicity is implicitly addressed by means of redundant paths for accessing resources and the fact that the latter are fragmented to retrieve them in a more resilient manner. Nevertheless, the centralized repository for resource lookup limits deployment of BitTorrent overlays in highly dynamic pervasive environments. Gia proposes a continuous topology optimization algorithm to proactively cater for problems that might occur due to dynamicity, whereas UMM makes use of multicast trees and self-organization of nodes and Phenix explicitly tackles dynamicity by means of bidirectional connectivity of nodes and redundant routing paths.

Table 3
Taxonomy of unstructured P2P overlays for pervasive environments.

P2P	Lookup	Dynamicity	Redundancy	Churn	Security	Queries	Load balance
Freenet	Constrained flooding	Distributed design	Path redundancy	Routing redundancy and small-world topology	Anonymity and privacy	Any type of queries	Power law distribution of links
Gnutella	Pure flooding	Periodic initiation of node discovery	No particular support	Small world topology	Anonymity	Any type of queries	Power law distribution of links
FastTrack	Flooding between supernodes	No particular support	Resource indices replication	Periodic update of supernodes	N/A	Any type of queries	Connections based on locality and supernode workload
BitTorrent	Central repository	Centralized single-point-of-failure	Fragmented, replicated resources	No particular consideration	N/A	Any type of queries	Choking algorithm
UMM	Flooding over multicast trees	Node self-organization	No particular support	Heartbeat monitoring mechanism	N/A	Any type of queries	Power law distribution of links
Gia	Biased random walks	Node self-organization	Content indices replication	Topology adaptation algorithm	N/A	Any type of queries	Power law distribution of links with advanced nodes promoted
Phenix	Not considered	Bidirectional node connectivity	No particular support	Continuous optimization of neighbor lists	N/A	Any type of queries	Power law distribution of links with advanced nodes promoted

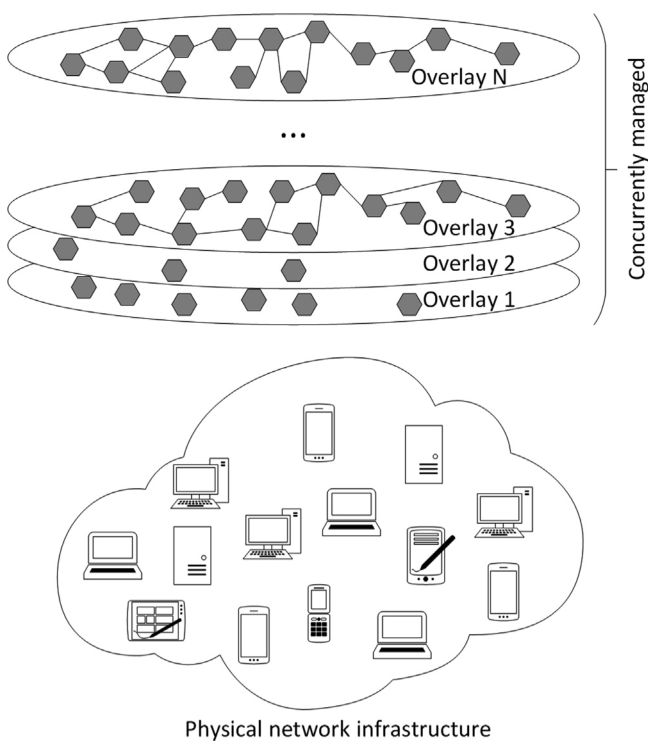

Fig. 2. Typical deployment of multiple overlays operating on top of the same physical network.

Furthermore, it becomes evident that quite a few of the unstructured topologies have features that classify them as small world networks (Milgram, 1967), namely graphs the degree distribution of which adheres to a power-law distribution. These overlays have a relatively small diameter, since some of their nodes are well-connected and have high degrees and therefore a lot of the other nodes are connected to them. Clearly, there is a great performance and security risk in this design, attributed to the semi-centralized nature of having hubs in the overlays' topologies. In this respect, these algorithms do not promote uniform distribution of resources across the nodes that participate in the P2P overlay and thus it becomes difficult to achieve load balancing in managing resources and handling lookup and routing requests. In particular, as far as the hub nodes of the small world graphs are concerned they can become heavily loaded and this can cause failures or bottlenecks (Albert et al., 2000).

Freenet, Gnutella and Gia all exhibit a small-world topology with few nodes emerging as hubs. Gia in particular proposes an enhancement over traditional small-world networks by promoting computationally "powerful" nodes as supernodes, i.e. hubs (in Gnutella and Freenet hubs emerge randomly). Similarly, FastTrack creates its two-tier hierarchical topology by promoting nodes with advanced computing capabilities as supernodes. BitTorrent's topology is classified as unstructured since it is created and modified according to requests for resources, therefore no small-world properties exist in this case. UMM instead follows along the line of Gnutella to build a power law topology, albeit with additional robustness-enhancing mechanisms considered. Phenix has a dual design goal, namely of retaining small-world network properties and low diameters at the same time.

Moreover, small overlay diameter implies more efficient resource discovery operations, due to the fact that such operations in unstructured overlay are based on either differentiations of standard flooding mechanisms, e.g. constrained flooding, expanding ring, or random walks (Lv et al., 2002). As long as the network diameter is constrained, then the messages do not get propagated along long paths and therefore consume less bandwidth on one hand, while on the other hand queries have better chances to be answered (Dimakopoulos and Pitoura, 2006). While this is the case for Freenet and Gnutella, i.e. constrained flooding, Gia and FastTrack utilize the concept of supernodes to augment the lookup procedure: nodes share their resource indices with the supernodes to which they are connected to and lookup operations take place at the supernode level of the Gia hierarchy. Conversely, BitTorrent and UMM use a centralized index of resources and multicast trees for lookup, respectively.

In addition, as mentioned before unstructured protocols with their flooding-based discovery mechanisms are inherently more suitable for single- and multi-attribute range queries. There have also been proposals to enhance unstructured protocols in order to support semantic queries (Nakauchi et al., 2004) and to improve lookup efficiency by exploiting DHT-like features such as hashing and content replication (Morselli et al., 2007). In the same direction, the noteworthy approach reported in Yang and Garcia-Molina (2002) and Crespo and Garcia-Molina (2002) is of great interest. In this approach the notion of search indices is utilized, namely at each node of the overlay there are pointers who indicate with great probability the direction that queries following random walks should take in order to become satisfied. A similar improvement in resource discovery mechanisms for a bio-inspired, self-organized unstructured P2P overlay, i.e. BlatAnt, was discussed in Brocco et al. (2010), where proactive caching of locators for resources across the overlay is propagated to expedite lookup.

Due to their indexing operations involving advertised resources, structured overlays have a very good performance in locating and retrieving rare resources. Conversely, unstructured protocols that rely on flooding operate better in finding highly replicated, popular resources. This observation has led researchers to devise approaches to improve the efficiency of unstructured P2P overlays in locating rare resources by utilizing structured approaches of indexing on top of unstructured overlays. A typical case of such an approach is the work presented in Loo et al. (2004). These observations regarding the behavior of structured and unstructured overlays in terms of different types of queries and the involved complexity have been tested and evaluated in Castro et al. (2005), thus validating some claims as far as resource discovery is concerned, but also negating some other claims according to which the complexity and overhead of structured protocols is higher than that of unstructured ones. It should nevertheless be noted that this work, while providing good indications of the performance of both types of P2P protocols, should be taken with a grain of salt since it cannot be generalized for all specific P2P protocols. A similar study comparing performance and resilience features of Gnutella and the DHT-based Overnet (Kutzner and Fuhrmann, 2005) was presented in Qiao and Bustamante (2006).

Lastly, similar to DHT-based structured overlays, unstructured ones also suffer from related security threats, but since this topic lies outside the research described in this work we refer the interested reader to literature such as Wallach (2003) for a comprehensive review of the subject. It is worth noting that Freenet was designed with the goal of preserving privacy and anonymity of participating nodes, whereas additionally Gnutella supports nodes' privacy.

Summing up, in Table 3 we present (based on the detailed discussion that preceded) a comparison between the different unstructured P2P algorithms in regard to various features that are characteristic of pervasive environments, namely lookup performance, dynamicity, redundancy, churn, security, complex queries, load balancing. The same features were also considered in the case of structured P2P overlays in Section 3, thus easing comparisons between the two approaches in terms of how well they satisfy requirements raised by the nature of pervasive environments.

5. Multi-layer P2P overlays

The previous discussion on various existing P2P overlays, structured and unstructured, highlighted their particular benefits and shortcomings regarding their operational context and performance. It is evident that there is no panacea in regard to P2P overlays, namely a solution that can be utilized to address the totality of issues pertaining to pervasive networks (Cooper, 2006). In this respect, we can safely assume that multiple overlays will be concurrently operating on top of the same physical network infrastructures, an assumption which is valid and can actually be witnessed in current network deployments (Lin et al., 2009; Maniymaran et al., 2007). A further reason for this is the fact that most P2P overlays are domain- or application-specific, e.g. VoIP or file sharing, and since it is to be expected that nowadays nodes will be active in more than one domains or activities, the motivation for the coexistence between multiple overlays becomes more prominent (Hsu et al., 2010). Figure 2 presents a typical configuration of concurrent, multiple P2P overlays running on top of the same physical network infrastructure. P2P overlays 1 to *N* operate concurrently over the same physical network indicated at the lower part of Fig. 2. This implies that the topologies of these P2P overlays are managed at the same time and reflect both the changes in the physical network, as well as the optimization criteria set by the design of the P2P overlay. Depending on the algorithm involved to build and maintain the multi-layer P2P overlay, the coexisting P2P overlays can communicate with each other or not.

Moreover, the need for multiple overlays is spurred by their inherent benefit in terms of virtualization (Mao et al., 2012). Network virtualization in the view of P2P overlays enables the utilization of the same physical resources by many different applications. However, simulation studies might potentially not yield accurate results (Shrestha et al., 2008; Chowdhury et al., 2009) since they make many assumptions regarding realistic deployments and should therefore act only as complementary to real experiments. Network virtualization through P2P overlay networks can thus greatly facilitate the development of novel network protocols and their deployment in real network conditions, with platforms such as PlanetLab (Peterson et al., 2003), GENI (Anderson and Reiter, 2006), MOSAIC (Mao et al., 2012) and VINI (Bavier et al., 2006) as typical examples. With the growing need for "live" experiments for novel Internet architectures and protocols, P2P overlays allow for multiple testbed experimentations to take place at the same time, over the same network resources and under realistic settings (Anderson et al., 2005). The cost savings achieved using such an approach are noteworthy, due to economies of scale as well as the reduction of potential runtime errors that simulation studies cannot foresee.

In the following we discuss the emerging benefits attributed to the coexistence of multiple overlays and the potentials for their cooperative maintenance. We also present typical examples of multi-layer overlays that share the same logic in creating and maintaining multiple P2P overlay topologies in parallel. In addition, we also highlight frameworks and architectures that facilitate inter-overlay cooperation.

5.1. Coexistence of multiple overlays

We argue that just by supporting multiple overlays to run on the same underlying networks, the aforementioned benefits and motivations would quickly become diminished due to the complexity of handling many overlays and their inter-operation protocols. There is thus the emergent requirement of concurrently maintaining these overlays in a coordinated manner that would simplify their management and would reduce their collective load on participating nodes (Anderson et al., 2005). The coexistence of more than one P2P overlay layers over the same underlying networks and its impact on network performance has not received significant research attention to date. Lin et al. (2009) experimented with the synergies that arise between coexisting overlays and argued that when carefully designed they can yield significant performance improvements. This work sets the foundations for coexisting overlays by proposing a well-defined classification scheme for potential synergies among such overlays. The classification scheme considers temporal synergies and dynamicity, as well as communication, state and service interactions between the coexisting overlays and their pattern of interaction, namely horizontal (parallel operation of overlays) or vertical (high-level overlay exploiting the functionality of lower-level overlay).

In Jiang et al. (2005) the effect of using concurrent multiple overlays for routing is explored, highlighting the problems that arise when these layers are not interacting with each other. The potential of combining the benefits of both structured and unstructured P2P overlays are reported in Maniymaran et al. (2007), where instances of both overlay types – a Pastry-like structured and a gossip-based unstructured overlay – on the same network were employed and their complementary operation was promoted. In particular, the authors propose to distinguish the operations of P2P overlays into primary and secondary, where the latter can be replaced by similar operations found in other overlays and the former are strictly maintained. While interesting, this work suffers from loss of generality since the example overlays discussed in the paper are not representative cases and cannot be used to extrapolate synergies between other P2P overlays.

Another important issue regarding overlay coexistence deals with how the resources of the underlying physical networks are to be distributed between the multitude of P2P overlays. In Demirci and Ammar (2010) the problem of how to best allocate bandwidth among competing and coexisting overlay networks running on the same physical network infrastructure is examined. A fairness metric is introduced in order to regulate bandwidth sharing with quite promising results. The approach followed by Cooper (2006) is also interesting; a middleware infrastructure is introduced to group the runtime functionalities of the different P2P overlays and specific filters are applied to control the messages that correspond to each of these overlays in order to efficiently handle the allocation of physical network resources, e.g. available bandwidth. Cooper (2006) also suggests the assignment of priorities to the different coexisting overlays, according to which the highest priority overlay would have more resources available without completely starving the other overlays. Lastly, Braun et al. (2008) proposed the UP2P overlay for ubiquitous communications support, in which the notion of federated overlay organization is introduced. According to the latter work, distinct overlays can coexist by cooperating through the use of a super-overlay that connects all of them together and is in charge of routing messages from one overlay to another. This super-overlay is based on DHTs to optimize lookup operations.

5.2. Representative cases of multi-layer overlays

The work on the MOMO[4] (Multi-Overlays for Multi-hOming) research project is to date the most typical case of multi-layer P2P overlays, considering Wireless Mesh Networks (WMNs) in line with network virtualization, i.e. overlays. The goal is to allow the creation of multiple context-based overlays, each of which is build in such a manner as to satisfy specific user context requirements, e.g. related to security, mobility, or service. An analytical model of the proposed approach on multi-layer context-based overlays is described in Matos et al. (2011a), while examples of how to model context information to facilitate the overlay creation and maintenance are presented in Matos et al. (2011b), where additionally the importance of context prediction for the stability of the considered approach is noted. In this work structured P2P systems are exploited by the authors, since efficient resource discovery, namely lookup regarding context information, is the most important design requirement. Nonetheless, specifics about the overlay construction and maintenance algorithms are not discussed, with the exception of the P2P control overlay, an overlay that is comprised of a node from each of the available context-based overlays and that serves to associate user requirements with the most appropriate context-based overlay (Matos et al., 2011b).

Initial work considered only structured P2P overlays and suffered from poor performance in terms of processing time per overlay link and node that was proportional to the number of context-based overlays that were concurrently maintained. Moreover, it was not clear how to control such an architecture, for example to collect and disseminate global context information and to manage the P2P control overlay. For these reasons, an extension of this work was presented in Matos et al. (2012) considering a distributed framework to manage the aforementioned multi-layer overlay architecture. The framework promotes cooperation between the different context-based overlays by supporting both unstructured, i.e. flooding, and structured, i.e. DHT-based, resource discovery mechanisms across all overlays. While the authors favor a distributed approach to discover available context-based overlays and share and retrieve information from them, their solution nevertheless suffers from a single point of failure, namely the *Global WMN Manager* entity that manages the control overlay, stores global context information and schedules the physical network resources among the number of overlays. This very promising approach focuses on the user association to a context-based overlay and the overhead of maintaining such a framework, whereas the actual construction and maintenance of the concurrently running overlays is sidestepped.

In Hsu et al. (2010) the case is made for cooperative maintenance of multi-overlay environments based on the premise that the multitude of considered overlays share common maintenance operations, e.g.failure detection or network proximity estimation. By adopting a master–slave model, whereby the master takes care of the common functionality thus alleviating the burden from its associated slaves, the overall maintenance costs across all overlays are minimized with reductions up to 60% as reported in Hsu et al. (2010). Furthermore, Serbu et al. (2011) propose a novel P2P DHT-based overlay that allows for flexible routing choices by supporting multiple, redundant paths in the overlay, which are constructed and maintained based on different objectives, such as fault tolerance, low latency and load balancing. The authors propose a hybercube approximation DHT overlay structure to support the diverse paths.

5.3. Frameworks for inter-overlay cooperation

Orthogonal to the issue of concurrently building and managing multiple P2P overlays that we are focusing on, there has also been a significant amount of research work on how to inter-connect heterogeneous overlays, i.e. frameworks and architectures to support the inter-operation of existing overlays. For the sake of completeness we indicatively refer the interested reader to related works such as Tan and Jarvis (2006), Wu and Li (2007), and Ciancaglini et al. (2011), but we analyze them no further since they are outside the scope of this article. It is nonetheless noteworthy to highlight the research from the University of Osaka on the bio-inspired analysis of co-existing overlays that was built on the premise of symbiosis studies that were performed in the field of biology (Wakamiya and Murata, 2006; Morimoto et al., 2009). In the latter work extensive numerical analytical modeling is used to study the symbiosis of multiple overlays and their competition for resources and cost minimization.

The use of a policy-based, context-aware architecture to handle the inter-operation between and the dynamic adaptation of different service-specific, semantic overlay networks was reported in Al-Oqily and Karmouch (2011). This agent-based architecture utilizes context information from a variety of sources to trigger policies that reconfigure the overlay's operation, e.g. routing, create new overlays or terminate existing ones. Policies are automatically generated by gathered context information, hence not being in alignment with business objectives but rather being very tightly intertwined with the P2P algorithm. Moreover, issues such as synchronization of policy execution have not not considered. A similar approach has been undertaken by the BioMPE research project and its proposed multi-layer overlay architecture, in which policy-based management principles and context information are utilized to select the most appropriate P2P overlay to activate based on monitored context information and user/application requirements (Malatras et al., 2012b).

Finally, of interest are architectures such as OverMesh (Ding et al., 2006), Open Overlays (Grace et al., 2008), ODIN-S (Cooper,

[4] http://momo.ani.univie.ac.at/index.html

2006) and Synergy (Kwon and Fahmy, 2005), which facilitate the development and deployment of concurrently running P2P overlays. These architectures have greatly assisted in studying the effects of concurrent overlay operation and examined the potentials for cross-overlay cooperation, further allowing the rapid evaluation of inter-overlay deployments.

5.4. Discussion

The aforementioned discussion has highlighted the potential benefits from exploiting multiple concurrently running overlays, i.e. reduced costs and greater flexibility. It is clear that when the need arises to employ two or more overlays in parallel, it is sensible to make use of the potential synergies between them in order to minimize as much as possible the number of management overhead, i.e. exchanged messages. Moreover, the concurrent presence of P2P overlays can satisfy the diversity of application requirements from such overlays. Whereas the efficient resource discovery of structured P2P overlays might be the goal of some applications, the flexible membership supported by unstructured overlays might be preferred by others. This was clearly exhibited by Lin et al. (2009) and Maniymaran et al. (2007), where the key term is that of synergy. Lin et al. (2009) in particular explored the different options available for collaboration between overlays and set the theoretical grounds for employing relevant solutions.

Undoubtedly, these benefits can be greatly improved when the considered overlays operate in a cooperative manner, namely when the synergies between them are taken into account. Unless carefully designed and deployed, the P2P overlays might end up competing for network and computational resources and thus lead to their depletion. Anderson et al. (2005) highlighted this problem by calling for novel solutions to support multiple simultaneous architectures running on the Internet. Such an approach would lower the barriers to deploying and running numerous concurrent overlays as the authors argue and would optimize their performance over the sole physical infrastructure, namely the Internet. Similar problems were identified in Jiang et al. (2005) for the case of routing. To alleviate concerns regarding fair resource utilization, approaches such as Demirci and Ammar (2010) (proposing a fairness metric to regulate shared access to resources) and Cooper (2006) (introduces a priority-based mechanism for accessing resources) are quite promising.

In this respect, solutions such as the ones discussed offer great benefits in virtualizing current network infrastructures and offering numerous alternate P2P substrates for applications and services to operate upon. Maniymaran et al. (2007) for example combined structured and unstructured overlays and in doing so exploited the benefits of both approaches. In UP2P (Braun et al., 2008), a DHT-based super-overlay was proposed to manage the federation of multiple overlays and a related approach is adopted by MOMO (Matos et al., 2011b) in building a higher level control overlay.

In general, the potentials and opportunities from the synchronized cooperation of overlays are significant. Accordingly, this line of research is consistent with other network research work on exploiting overlapping functionalities and operations in networks to better utilize the scarce available resources, such as Saha et al. (2015) and the Synapses framework that targets P2P overlays (Ciancaglini et al., 2013).

6. Bio-inspired P2P overlays

Algorithms and techniques inspired from the field of biology and the observation of natural phenomena have been utilized for years in solving complex optimization problems. Lately, similar solutions have started to be applied in the management of complex systems in particular in distributed environments, since these solutions do not suffer from single-point-of-failure concerns and are highly reliable. Considering bio-inspired approaches to address problems and performance issues in computing environments is progressively gaining wide applicability, mostly attributed to the beneficial properties of such approaches. In particular, bio-inspired solutions are characterized by their highly adaptive and reactive behavior, inherent support for heterogeneity, distributed operation, resilience to failure of components and self-organization (Babaoglu et al., 2006). It thus becomes evident that such approaches are very promising candidates to address and handle the dynamicity of pervasive networks and the management of P2P overlay networks that are constructed on top of the latter. Therefore, bio-inspired computing is steadily emerging as a prominent solution for distributed and pervasive computing (Bongard, 2009), as well as networking (Dressler and Akan, 2010).

Bio-inspired solutions have proven to be quite effective in providing efficient solutions in the domain of computer networks (Balasubramaniam et al., 2011). 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 that are achieved through the concept of stigmergy that enables indirect communication means for cooperating entities. In particular, the Ant Colony Optimization (ACO) theory (Dorigo et al., 1996), which was originally studied by Dorigo, has been successfully utilized to satisfy at a low cost routing, scheduling and allocation problems. ACO has also been used to construct and maintain P2P overlays, with the BlatAnt overlay being the most prominent instance of such work. Moreover, the adaptability of bio-inspired solutions through their self-organization and reactive nature is also of great benefit in addressing distributed networking concerns, as well as to support the inherent high-degree of heterogeneity.

Furthermore, as far as P2P overlays are concerned, there have been some approaches to construct and maintain such overlays using bio-inspired approaches. Inspiration from nature has also been taken into account for the provision of efficient resource discovery mechanisms built on top of P2P overlay topologies. Based on the aforementioned discussion on characteristics of bio-inspired solutions, i.e. distributed, support for heterogeneity and continuous adaptation, it is evident that the majority of P2P overlays constructed using such solutions belongs to the unstructured category. Nonetheless, there has also been research in applying bio-inspired approaches on structured P2P overlay topologies. In the following we describe the most typical cases of bio-inspired P2P overlay protocols.

6.1. BlatAnt

The BlatAnt algorithm (Brocco et al., 2010) aims at the construction and maintenance of bounded diameter, unstructured P2P overlays over grids. ACO principles are exploited to ensure optimization of node connections in the P2P overlay topology in a fully decentralized and distributed manner. The use of ant colonies also promotes adaptability and robustness of the constructed overlays, two features which are highly important especially when the underlying networks are subject to dynamic conditions. Optimization of topologies in BlatAnt is based on two rules, i.e. the connection and disconnection rule, that collectively limit the diameter between D and $2D-1$ with D being a parameter of the algorithm. According to the connection rule, two nodes will become connected if their current minimal distance is $\geq 2D-1$. Conversely, the disconnection rule disconnects two nodes if their maximum distance in the overlay topology is $\leq D-1$. These two rules are continuously evaluated and applied when needed. BlatAnt has dedicated mechanisms to recover from overlay partitions occurring both when nodes gracefully depart the overlay, as well as when they do so abruptly.

BlatAnt's proper operation is dependent on the timely and accurate dissemination of information regarding the network topology. This operation is the responsibility of ant-inspired agents that roam the overlay and during their migration they collect information from nodes and distribute them to other nodes. Moreover, to ensure that these ant agents spread throughout the overlay and to guarantee its full coverage, another bio-inspired technique is used, namely that of pheronomone deposits. The latter are used by ant agents to indicate whether particular paths in the overlay topology have been explored or not and to what extent (how popular they are). Accordingly, ant agents tend to prefer less popular paths to promote network coverage. Path pheromones evaporate over time to allow for changes in the overlay's topology to be reflected by the deposit of new pheromones. Consequently, network topology optimization according to the two previous rules occurs for each node independently. After a reasonable amount of time, a relatively stable state can be reached upon which the topology changes are minimal and subject to the underlying network dynamics.

BlatAnt has been extensively studied and its operation and effectiveness in constructing bounded P2P overlays has been validated through simulation studies performed in a custom build simulator that did not assess overhead traffic. It has proved to be very performant in quickly converging to overlays with bounded diameter even under extreme dynamic conditions. In addition, efficient resource discovery mechanisms using random walks and proactive caching of resource indices have been proposed (Brocco et al., 2010). BlatAnt was solely focused on bounded diameter optimization because it was targeted specifically at grids, where resource discovery is of outmost importance. Pervasive environments conversely have plentiful and variable optimization requirements and hence overlays targeted for them should be more flexible.

6.2. AntOM

AntOM (Ant-inspired Overlay Maintenance) (Peng et al., 2012) is a topology optimisation algorithm for multi-layer P2P overlays on top of pervasive environments. The algorithm is based on the BlatAnt algorithm and utilizes ACO in regard to the network exploration and neighborhood optimisation. By exploiting different ant families, multiple overlay layers, each one optimised for a different property, are concurrently maintained at a low cost as proven by the experimental results reported in Peng et al. (2012). AntOM therefore provides concurrent support for applications with diverse requirements that utilize the same networking infrastructure.

The algorithm constructs multiple overlay network layers on top of the same underlay. Each layer is optimised for a specific node property, e.g. battery levels, Quality-of-Service level. ACO principles are used for the continuous exchange of information among nodes in order to promote continuous optimization and adaptation. Contrary to BlatAnt, ants are responsible for disseminating information by utilising two types of pheromones, namely one that indicates the quality and validity of node properties and guides neighbours selection, and one that guides ants to evenly explore the entire overlay (linked to time since a neighbour was last visited). Furthermore, ants disseminate information in the overlay by stochastically migrating from one node to another following inverse pheromone concentrations. Due to the uncertainty in pervasive environments and the underlying dynamicity, there is a major concern regarding information delay that could be detrimental for the overlay's performance. To alleviate this concern an additional family of ants is proposed to rapidly collect information in the vicinity of a node. There are thus two ant families that co-exist in AntOM. Topology optimization occurs by adding and removing links locally at every node in order to achieve local maxima or minima (depending on the optimization criteria).

The algorithm was evaluated in terms of convergence of properties as well as overhead under different dynamicity levels and number of concurrently maintained layers. Moreover, it created optimal overlay layers for the considered node properties, such as node workload, thus supporting load balancing. It is encouraging to highlight that network dynamicity influenced the traffic overhead in a controlled and limited way. Additionally, even under highly dynamic conditions the algorithm converged quickly to stable overlays.

6.3. Self-Chord

Self-Chord was presented in Forestiero et al. (2010) as a bio-inspired P2P overlay targeted at grid and cloud infrastructures. This overlay construction and maintenance algorithm is inspired by swarm intelligence and ant colonies, where multiple simple and independent mobile agents roam the overlay network and collectively re-arrange resource identifiers to promote lookup efficiency.

As its name implies, Self-Chord is based on Chord and actually as far as the topology is concerned it is exactly the same as in Chord, namely Self-Chord utilizes a logical ring topology to organize nodes in the P2P overlay. The nodes are ordered according to their identifier that is derived through a hashing function similarly to Chord. The difference lies in the placement of resource identifiers. Whereas in Chord resources and nodes shared the same identifiers' numerical space and accordingly resources were placed on nodes whose identifiers were close to each other, in Self-Chord resources obtain their identifiers from a different namespaces and are placed on nodes based on load balancing criteria, as well as semantic ones, i.e. closely related resources are clustered on neighboring nodes. It is important to note that as stated by the authors in Forestiero et al. (2010), the methodology to enhance structured P2P overlays that was employed in Self-Chord can serve as a guideline for analogous mechanisms being applied to other structured protocols such as CAN and Pastry, e.g. Self-CAN overlay (Giordanelli et al., 2012). Moreover, Self-Chord serves as a case-study of successful application of bio-inspired swarm intelligence mechanisms in structured P2P overlays and in doing so it also provides a proper mathematical analysis of these mechanisms that can be exploited to study its performance and scalability.

Self-Chord is very efficient in terms of resource discovery since it is utilizing the fingers table of the standard Chord overlay to provide logarithmic guarantees in locating resources. Moreover, through the operations of the swarm agents that re-arrange resource identifiers and place them on nodes according to their similarity (using a custom metric called *centroid*), clusters of similar resources are inherently constructed. This allows users to perform range queries easily, since it is only needed to look for a particular resource and then access the resources that are located on the same node to ensure that one discovers other semantically close resources. Furthermore, the statistical distribution of resource identifiers on nodes leads to better load balancing for the nodes that need to serve requests for resources. In addition, management operations of Self-Chord are significantly less than those of Chord since there is no need to re-arrange resource identifiers subject to node churn because of the continuous optimization performed by the swarm agents.

6.4. Other approaches

For the sake of reference, we briefly summarize a few other indicative research efforts towards building and maintaining P2P overlays using inspiration from nature and biological primitives, while it should be noted that this list is far from exhaustive.

- *P2PBA* (Dhurandher et al., 2011): The P2PBA (Peer-to-Peer Bee Algorithm) has a clear focus on providing efficient resource discovery mechanisms on mobile ad hoc networks. It is inspired from the foraging behavior of honey bees and in this respect it is very lightweight in terms of exchanged messages and also returns an ordered list of areas called patches where high concentration of the results can be found instead of individual nodes holding the requested resources. This process greatly reduces the amount of time to locate resources since it guides directly to the patch that is most close to the results and hence minimizes the search area. Reported simulation results validate the reduced traffic overhead for this algorithm compared to related ACO-based algorithms, as well as achieving lookup efficiency.
- *Antares* (Forestiero and Mastroianni, 2009): The Antares (Ant-Based Algorithm for Resource management in grids) overlay is characterized as self-structured and is targeted at grids. It uses ant agents that roam the P2P overlay topology and re-organize resources by moving them from one node to another in order to ensure that similar resources are eventually clustered in the same neighborhood of the overlay's topology and thus promote the efficiency of resource discovery. The algorithm leads to an unstructured overlay, yet the positioning of resources according to their similarity bears semblance to relevant operations of structured overlays, e.g. Self-Chord.
- *AntCAN* (Apel and Buchmann, 2005): Utilizing mechanisms from the ACO theory and swarm intelligence, the AntCAN P2P overlay is essentially a standard, structured CAN overlay enhanced with ant-inspired resource discovery mechanisms. Of interest is the notion of organic P2P overlay networks that is proposed by Apel and Buchmann (2005): they argue towards a framework based on biology-inspired principles to augment P2P overlay structures by exploiting the potential of self-organizing P2P overlays, in a manner similar to BlatAnt and AntOM.
- *Myconet* (Snyder et al., 2009): This algorithm is inspired by the growth mechanisms found in fungal hyphae and leads to an unstructured P2P overlay, but with the presence of superpeers that serve as a backbone to facilitate resource discovery and overlay maintenance operations. The algorithm has a low resilience to node churn, with a tolerance of up to 5% churn rate, but is highly scalable. Moreover, its merits lie on its adaptability properties in that the algorithm quickly recovers the network topology from partitions or failures of large node populations (up to 50%).
- *Self-CAN* (Giordanelli et al., 2012): It works similarly to Self-Chord, i.e. the topology of the underlying standard, structured P2P overlay remains the same, 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.
- *P2PSI* (Hoh and Hwang, 2007): Targeted at mobile ad hoc networks, the P2PSI P2P overlay has a clear focus on supporting file sharing. Accordingly, it does not consider construction and maintenance of an overlay topology, but instead proposes an ACO-inspired mechanism for efficiently locating files on an existing topology. The behavior of ants is emulated by query messages, whereas the associated pheromones to guide query-reply and prospective query messages are inversely proportional to the number of hops a query-reply message needs to travel on the overlay to reach the initiator of the query, thus promoting file holders in the vicinity of the initiator.
- *SCAN* (Ghanea-Hercock et al., 2006): It is targeted at very dynamic networks with a high degree of node churn and its goal is to be adaptive and reliable to node failures, as well as targeted attacks on core nodes of the overlay (a common type of attack on power-law graphs). Inspired by the behavior of biological neurons, the SCAN protocol aims at retaining at each node a minimum number of active connections that maximize its perceived utility. The resulting unstructured P2P overlay exhibits improved performance in terms of lookup efficiency and robustness against node failures.

6.5. Discussion

Utilizing inspiration from biological processes to construct and maintain P2P overlays has attracted some research interest to date and new research projects such as BioMPE aim to extend this research and instigate further work in this direction. From the aforementioned discussion it becomes clear that the majority of related solutions focus on providing efficient resource discovery mechanisms using swarm intelligence techniques, e.g. Michlmayr (2006), Wu et al. (2006), and Deng et al. (2009). This is mainly attributed to the fact that such techniques have proven performance benefits in regard to routing and scheduling in dynamic networks, while they have also inherent support for adaptability and robustness in light of node failures (Ko et al., 2008). Conversely, using such techniques for topology management has not really been exploited with the exception of solutions such as BlatAnt and AntOM.

Despite their merits, bio-inspired solutions have a series of disadvantages that have limited their applicability and wide deployment. In particular, it is computationally difficult to implement such solutions since they are based on principles of self-organization and the plethora of independent agents interact with each other using indirect means, an aspect that is hard to model and realize in practical information systems (Biskupski et al., 2007). For this purpose simulation systems such as Anthill (Babaoglu et al., 2002), SADMAS (Michal Pechoucek et al., 2012) or OverSwarm (Brocco and Baumgart, 2012) are of great value since they support the development and evaluation of bio-inspired P2P overlays thus facilitating the resolution of deployment and practical execution of such multi-agent systems (Michal Pechoucek et al., 2012). OverSwarm in particular has been developed on top of a powerful network simulator, namely OMNET++/OverSim (Baumgart et al., 2007).

Another problem usually attributed to bio-inspired algorithms and in particular those based on ACO is that they induce significant overhead due to the big number of exchanged messages, i.e. ants and pheromone updates. Lastly, since swarm intelligence algorithms are inherently probabilistic it is very difficult to guarantee quality of service levels as far as they are concerned, whereas for the same reason issues such as stagnation might occur (Bonabeau et al., 2000).

Summing up, in Table 4 we present a comparison between the different unstructured P2P algorithms in regard to various features that are characteristic of pervasive environments, namely lookup performance, dynamicity, redundancy, churn, security, complex queries, load balancing. The same features were also considered in the case of structured and unstructured P2P overlays in Sections 3 and 4 respectively, thus easing comparisons between all

Table 4
Taxonomy of unstructured P2P overlays for pervasive environments.

P2P	Lookup	Dynamicity	Redundancy	Churn	Security	Queries	Load balance
BlatAnt	Use of resource indices to guide "flooding" and random walks	ACO principles for self-organization	Resource indices shared between nodes	Partition recovery mechanisms and ACO continuous optimization	N/A	Any type of queries	Bounded diameter and low degree
AntOM	Random walks and constrained flooding	ACO principles for self-organization and use of short-range ants	Path redundancy by means of multiple overlays	Partition recovery mechanisms and ACO continuous optimization	N/A	Any type of queries	Bounded diameter and low degree. Aggregation of operations among multiple overlays
Self-Chord	$O(\log N)$ like Chord	No particular support	No particular support	Periodic update resource placement by ant agents	N/A	Any type of queries (range queries supported via semantically close placement of resources by ant agents)	Resources placed on nodes using load balancing criteria

considered approaches in terms of how well they satisfy requirements raised by the nature of pervasive environments. Security is a major concern when bio-inspired P2P overlays are considered, since due to the self-organized nature of the agents, the latter can be manipulated by malicious attackers (Zhong and Evans, 2002). Pheromone biasing mechanisms are foremost among the related threats and in this respect trust-based protection approaches such as Mrmol and Prez (2010) are quite promising.

7. Conclusions

We systematically reviewed P2P overlays in pervasive environments by considering their operation in light of the dynamicity, mobility and heterogeneity of such environments. These adverse features negatively influence the performance and proper operation of traditional P2P overlays and therefore mechanisms and techniques to alleviate their effects have been proposed. We have discussed such techniques for both structured and unstructured P2P overlays in order to provide extensive coverage of the particular research domain. Moreover, we have presented a review of research on approaches towards P2P overlay coexistence and cooperation. Such approaches are very significant for the context of pervasive environments, where their inherent complexity and heterogeneity undoubtedly necessitate the deployment of multiple overlays, each one with diverse characteristics. The plethora of applications and services that operate on P2P overlays in pervasive environments and their specialized requirements further motivate research in the direction of overlay cooperation.

Furthermore, there is a prominent need for constant optimization and adaptation to fit the evolving dynamic requirements of pervasive environments. In this respect, we systematically reviewed related research and focused on a recently emerging trend, namely that of bio-inspired overlays. The desired features of such approaches, i.e. robustness, self-organization, adaptation, constitute them prominent candidates to address the relevant concerns found in pervasive environments. This review on P2P overlays also highlighted that there is a necessity to devise uniform and realistic testing environments for the latter, in order to allow for comparisons and proper evaluation prior to actual deployment. In addition to that, in depth analysis of the costs involved in realizing and deploying P2P overlays would greatly benefit their adoption from the user community, which by and large is to date rather agnostic in this respect.

By describing and critically analyzing existing systems and discussing current research and open issues, we aspire to instigate further research in a domain that has great potential to grow. Issues such as energy efficiency, synergies between coexisting overlays and bio-inspired solutions at low maintenance costs and with flexibility are some of the concerns that have arisen from the state-of-the-art review. We assert that the latter concerns will serve as functional requirements for the design and development of novel P2P overlay solutions in pervasive environments in the future.

References

Aberer K, Punceva M, Hauswirth M, Schmidt R. Improving data access in p2p systems. Internet Comput IEEE 2002;6(1):58–67.
Aberer K, Cudré-Mauroux P, Datta A, Despotovic Z, Hauswirth M, Punceva M, et al. P-grid: a self-organizing structured p2p system. SIGMOD Rec 2003;32(September (3)):29–33.
Abowd GD, Mynatt ED. Charting past, present, and future research in ubiquitous computing. ACM Trans Comput–Hum Interact 2000;7(March (1)):29–58.
Adamic LA, Lukose RM, Puniyani AR, Huberman BA. Search in power-law networks. Phys Rev E 2001;64(September).
Ahmed R, Boutaba R. A survey of distributed search techniques in large scale distributed systems. Commun Surv Tutor IEEE 2011;13(2):150–67.
Akyildiz IF, Wang X, Wang W. Wireless mesh networks: a survey. Comput Netw 2005;47(4):445–87.
Albert R, Jeong H, Barabási A-L. Error and attack tolerance of complex networks. Nature 2000;406(6794):378–82.
Al-Oqily I, Karmouch A. A decentralized self-organizing service composition for autonomic entities. ACM Trans Auton Adapt Syst 2011;6(February (1)):7:1–18.
Anderson T, Reiter MK. Geni: Global Environment for Network Innovations Distributed Services Working Group; 2006.
Anderson T, Peterson L, Shenker S, Turner J. Overcoming the Internet impasse through virtualization. Computer 2005;38(April (4)):34–41.
Androutsellis-Theotokis S, Spinellis D. A survey of peer-to-peer content distribution technologies. ACM Comput Surv 2004;36(December (4)):335–71.
Apel S, Buchmann E. Biology-inspired optimizations of peer-to-peer overlay networks. Prax Inf Kommun 2005;28(4):199–205.
Aspnes J, Shah G. Skip graphs. ACM Trans Algorithms 2007;3(November (4)).
Babaoglu O, Meling H, Montresor A. Anthill: a framework for the development of agent-based peer-to-peer systems. In: Proceedings of the 22nd international conference on distributed computing systems; 2002. p. 15–22.
Babaoglu O, Canright G, Deutsch A, di Caro GA, Ducatelle F, Gambardella LM, et al. Design patterns from biology for distributed computing. ACM Trans Auton Adapt Syst 2006;1(September (1)):26–66.
Balasubramaniam S, Leibnitz K, Lio P, Botvich D, Murata M. Biological principles for future Internet architecture design. Commun Mag IEEE 2011;49(7):44–52.
Baldoni R, Bonomi S, Rippa A, Querzoni L, Piergiovanni ST, Virgillito A. Evaluation of unstructured overlay maintenance protocols under churn. In: 26th IEEE international conference on distributed computing systems (ICDCS) workshops; 2006.
Barjini H, Othman M, Ibrahim H, Udzir N. Shortcoming, problems and analytical comparison for flooding-based search techniques in unstructured p2p networks. Peer-to-Peer Netw Appl 2012;5:1–13.
Barriére L, Fraigniaud P, Kranakis E, Krizanc D. Efficient routing in networks with long range contacts. In: Proceedings of the 15th international conference on distributed computing, DISC '01. Springer, Berlin Heidelberg; 2001. p. 270–84.
Baumgart I, Heep B, Krause S. Over Sim: a flexible overlay network simulation framework. In: Proceedings of the 10th IEEE global Internet symposium (GI ' 07) with IEEE INFOCOM; May 2007. p. 79–84.
Bavier A, Feamster N, Huang M, Peterson L, Rexford J. In vini veritas: realistic and controlled network experimentation. In: Proceedings of the 2006 conference on applications, technologies, architectures, and protocols for computer communications, SIGCOMM ' 06. ACM, NY, USA; 2006. p. 3–14.
Bharambe AR, Agrawal M, Seshan S. Mercury: supporting scalable multi-attribute range queries. SIGCOMM Comput Commun Rev 2004;34(August (4)):353–66.

Biskupski B, Dowling J, Sacha J. Properties and mechanisms of self-organizing manet and p2p systems. ACM Trans Auton Adapt Syst 2007;2(March (1)).
Bonabeau E, Dorigo M, Theraulaz G. Inspiration for optimization from social insect behaviour. Nature 2000;406(July):39–42.
Bongard J. Biologically inspired computing. Computer 2009;42(April (4)):95–8.
Braun D, Buford JF, Fish RS, Gelman AD, Kaplan A, Khandelwal R, et al. Up2p: a peer-to-peer overlay architecture for ubiquitous communications and networking. Commun Mag IEEE 2008;46(12):32–9.
Brocco A, Baumgart I. A framework for a comprehensive evaluation of ant-inspired peer-to-peer protocols. In: 20th Euromicro international conference on parallel, distributed and network-based processing (PDP); 2012. p. 303–310.
Brocco A, Malatras A, Hirsbrunner B. Enabling efficient information discovery in a self-structured grid. Fut Gener Comput Syst 2010;26(6):838–46.
Cai H, Wang J. Foreseer: a novel, locality-aware peer-to-peer system architecture for keyword searches. In: Proceedings of the fifth ACM/IFIP/USENIX international conference on Middleware. Springer, Berlin, Heidelberg; 2004. p. 38–58.
Canali C, Renda ME, Santi P, Burresi S. Enabling efficient peer-to-peer resource sharing in wireless mesh networks. IEEE Trans Mob Comput 2010;9(3):333–47.
Carchiolo V, Malgeri M, Mangioni G, Nicosia V. An adaptive overlay network inspired by social behaviour. J Par Distr Comput 2010;70(3):282–95.
Castro M, Costa M, Rowstron A. Should we build gnutella on a structured overlay? SIGCOMM Comput Commun Rev 2004;34(January (1)):131–6.
Castro M, Costa M, Rowstron A. Debunking some myths about structured and unstructured overlays. In: Proceedings of the second symposium on networked systems design & implementation, NSDI'05. USENIX, Berkeley, CA, USA; 2005. p. 85–98.
Chawathe Y, Ratnasamy S, Breslau L, Lanham N, Shenker S. Making gnutella-like p2p systems scalable. In: Proceedings of the 2003 conference on applications, technologies, architectures, and protocols for computer communications, SIGCOMM '03. ACM, NY, US; 2003. p. 407–418.
Chowdhury NM, Mosharaf K, Boutaba R. Network virtualization: state of the art and research challenges. Commun Mag IEEE 2009;47(July (7)):20–6.
Ciancaglini V, Liquori L, Hoang G. Towards a common architecture to interconnect heterogeneous overlay networks. In: IEEE 17th international conference on parallel and distributed systems; December 2011. p. 817–22.
Ciancaglini Vincenzo, Gaeta Rossano, Loti Riccardo, Liquori Luigi. Interconnection of large scale unstructured p2p networks: modeling and analysis. In: Dudin Alexander, De Turck Koen, editors. Lecture notes in computer science. Analytical and stochastic modeling techniques and applications, vol. 7984. Berlin, Heidelberg: Springer; 2013. p. 183–97.
Clarke I, Sandberg O, Wiley B, Hong TW. Freenet: a distributed anonymous information storage and retrieval system. In: International workshop on designing privacy enhancing technologies: design issues in anonymity and unobservability. Springer, Berlin Heidelberg, Germany; 2001. p. 46–66.
Cohen B. Incentives build robustness in BitTorrent; 2003.
Condie T, Kamvar SD, Garcia-Molina H. Adaptive peer-to-peer topologies. In: Proceedings of the fourth international conference on peer-to-peer computing; 2004. p. 53–62.
Conti M, Giordano S. Multihop ad hoc networking: the theory. Commun Mag IEEE 2007;45(April (4)):78–86.
Cooper BF. Trading off resources between overlapping overlays. In: Proceedings of the ACM/IFIP/USENIX 2006 international conference on Middleware, Middleware '. Springer Verlag New York, NY, USA; 2006. p. 101–20.
Crespo A, Garcia-Molina H. Routing indices for peer-to-peer systems. In: Proceedings of the 22nd international conference on distributed computing systems; 2002. p. 23–32.
Datta A, Hauswirth M, John R, Schmidt R, Aberer K. Range queries in trie-structured overlays. In: Proceedings of the fifth IEEE international conference on peer-to-peer computing, P2P ' 05. IEEE, Konstanz, Germany; 2005. p. 57–66.
Demirci M, Ammar M. Fair allocation of substrate resources among multiple overlay networks. In: Proceedings of the 2010 IEEE international symposium on modeling, analysis and simulation of computer and telecommunication systems, MASCOTS '10. IEEE; 2010. p. 121–30.
Deng Y, Wang F, Ciura A. Ant colony optimization inspired resource discovery inp2p grid systems. J Supercomput 2009;49(1):4–21.
Dey Anind K. Understanding and using context. Pers Ubiq Comput 2001;5(January (1)):4–7.
Dhurandher SK, Misra S, Pruthi P, Singhal S, Aggarwal S, Woungang I. Using bee algorithm for peer-to-peer file searching in mobile ad hoc networks. J Netw Comput Appl 2011;34(5):1498–508.
Dimakopoulos VV, Pitoura E. On the performance of flooding-based resource discovery. IEEE Trans Par Distr Syst 2006;17(11):1242–52.
Ding G, Vicente J, Rungta S, Krishnaswamy D, Chan W, Miao K. Overlays on wireless mesh networks: implementation and cross-layer searching. In: Proceedings of the ninth IFIP/IEEE international conference on management of multimedia and mobile networks and services, MMNS'06. Springer; 2006. p. 171–82.
Dorigo M, Maniezzo V, Colorni A. Ant system: optimization by a colony of cooperating agents. IEEE Trans Syst Man Cybern Part B: Cybern 1996;26 (1):29–41.
Dressler F, Akan OB. Bio-inspired networking: from theory to practice. Commun Mag IEEE 2010;48(11):176–83.
Eng Keong L, Crowcroft J, Pias M, Sharma R, Lim S. A survey and comparison of peer-to-peer overlay network schemes. Commun Surv Tutor IEEE 2005;7 (2):72–93 Quarter.
Fan J, Ammar MH. Dynamic topology configuration in service overlay networks: a study of reconfiguration policies. In: 25th IEEE international conference on computer communications (INFOCOM); 2006. p. 1–12.
Flocchini P, Nayak A, Xie M. Enhancing peer-to-peer systems through redundancy. IEEE J Select Areas Commun 2007;25(January (1)):15–24.
Forestiero A, Mastroianni C. A swarm algorithm for a self-structured p2p information system. IEEE Trans Evol Comput 2009;13(4):681–94.
Forestiero A, Leonardi E, Mastroianni C, Meo M. Self-chord: a bio-inspired p2p framework for self-organizing distributed systems. IEEE/ACM Trans Netw 2010;18(5):1651–64.
Fraigniaud P, Gauron P. D2b: a de Bruijn based content-addressable network. Theor Comput Sci 2006;355(April (1)):65–79.
Freedman MJ, Freudenthal E, Maziéres D. Democratizing content publication with coral. In: Proceedings of first symposium on networked systems design and implementation, NSDI'04. USENIX, Berkeley, CA, USA; 2004.
Galluccio L, Morabito G, Palazzo S, Pellegrini M, Renda ME, Santi P. Georoy: a location-aware enhancement to viceroy peer-to-peer algorithm. Comput Netw 2007;51(8):1998–2014.
Ganesan P, Yang B, Garcia-Molina H. One torus to rule them all: multi-dimensional queries in p2p systems. In: Proceedings of the seventh international workshop on the web and databases, WebDB '04. ACM, NY, USA; 2004. p. 19–24.
Ghanea-Hercock RA, Wang F, Sun Y. Self-organizing and adaptive peer-to-peer network. IEEE Trans Syst Man Cybern Part B: Cybern 2006;36(6):1230–6.
Giordanelli R, Mastroianni C, Meo M. Bio-inspired p2p systems: the case of multidimensional overlay. ACM Trans Auton Adapt Syst 2012;7(December (4)):35:1–28.
Gnutella. Gnutella protocol specification; 2002. 0.6.
Grace P, Hughes D, Porter B, Blair GS, Coulson G, Taiani F. Experiences with open overlays: a middleware approach to network heterogeneity. SIGOPS Oper Syst Rev 2008;42(April (4)):123–36.
Harvey N, Jones M, Saroiu S, Theimer M, Wolman A. Skipnet: a scalable overlay network with practical locality properties. In: Fourth USENIX symposium on Internet technologies and systems, USITS'03; 2003.
Hoh C-C, Hwang R-H. P2p file sharing system over manet based on swarm intelligence: a cross-layer design. In: IEEE wireless communications and networking conference (WCNC); 2007. p. 2674–9.
Hsiao H-C, Liao H, Huang C-C. Resolving the topology mismatch problem in unstructured peer-to-peer networks. IEEE Trans Par Distr Syst 2009;20 (11):1668–81.
Hsu C-J, Chung W-C, Lai K-C, Li K-C, Chung Y-C. A novel approach for cooperative overlay-maintenance in multi-overlay environments. In: 2010 IEEE second international conference on cloud computing technology and science (CloudCom); December 2010. p. 81–8.
Huebsch R, Hellerstein JM, Lanham N, Loo BT, Shenker S, Stoica I. Querying the internet with pier. In: Proceedings of the 29th international conference on very large data bases, VLDB '03. VLDB Endowment; 2003. p. 321–32.
Jagadish HV, Ooi BC, Vu QH. Baton: a balanced tree structure for peer-to-peer networks. In: Proceedings of 31st international conference on very large data bases, VLDB '05. VLDB Endowment, Trondheim, Norway; 2005. p. 661–72.
Jelasity M, van Steen M. Large-scale newscast computing on the Internet; October 2002.
Jiang W, Chiu D-M, Lui JCS. On the interaction of multiple overlay routing. Perform Eval 2005;62(October):229–46.
Joung Y-J, Wang J-C. Chord2: a two-layer chord for reducing maintenance overhead via heterogeneity. Comput Netw 2007;51(3):712–31.
Kaashoek MF, Karger D. Koorde: a simple degree-optimal distributed hash table. In: Kaashoek MF, Stoica I, editors. Peer-to-peer systems II. Lecture notes on computer science, vol. 2735. Berlin Heidelberg: Springer; 2003. p. 98–107.
Karger D, Lehman E, Leighton T, Panigrahy R, Levine M, Lewin D. Consistent hashing and random trees: distributed caching protocols for relieving hot spots on the world wide web. In: Proceedings of the 29th ACM symposium on theory of computing, STOC '97. ACM, NY, USA ; 1997. p. 654–63.
Karnstedt M, Sattler K-U, Hauswirth M, Schmidt R. Cost-aware processing of similarity queries in structured overlays. In: Sixth IEEE international conference on peer-to-peer computing; 2006. p. 81–9.
Kleinberg J. The small-world phenomenon: an algorithm perspective. In: Proceedings of the 32nd annual ACM symposium on theory of computing, STOC '00. ACM, NY, USA; 2000. p. 163–70.
Ko SY, Gupta I, Jo Y. A new class of nature-inspired algorithms for self-adaptive peer-to-peer computing. ACM Trans Auton Adapt Syst 2008;3(August (3)):111–134.
Kumar A, Merugu S, Xu J, Zegura EW, Yu X. Ulysses: a robust, low-diameter, low-latency peer-to-peer network. Eur Trans Telecommun 2004;15(6):571–87.
Kutzner K, Fuhrmann T. Measuring large overlay networks the overnet example. In: Mller P, Gotzhein R, Schmitt J, editors. Kommunikation in Verteilten Systemen (KiVS), Informatik aktuell. Berlin, Heidelberg: Springer; 2005. p. 193–204.
Kwon M, Fahmy S. Synergy: an overlay internetworking architecture. In: Fourteenth international conference on computer communications and networks, 2005. ICCCN 2005. Proceedings; October 2005. p. 401–6.
Leitao J, Marques JP, Pereira J, Rodrigues L. X-bot: a protocol for resilient optimization of unstructured overlay networks. IEEE Trans Par Distr Syst 2012;23(11):2175–88.
Li J-S, Chao C-H. An efficient superpeer overlay construction and broadcasting scheme based on perfect difference graph. IEEE Trans Par Distr Syst 2010;21 (5):594–606.

Lian Q, Zhang Z, Wu S, Zhao BY. Z-ring: fast prefix routing via a low maintenance membership protocol. In Proceedings of 13th IEEE international conference on network protocols, ICNP '05; 2005. p. 132–46.
Liang J, Kumar R, Ross KW. The fasttrack overlay: a measurement study. Comput Netw 2006;50(April (6)):842–58.
Li J, Sollins K, Lim D-Y. Implementing aggregation and broadcast over distributed hash tables. SIGCOMM Comput Commun Rev 2005;35(January (1)):81–92.
Li D, Cao J, Lu X, Chen K. Efficient range query processing in peer-to-peer systems. IEEE Trans Knowl Data Eng 2009;21(1):78–91.
Lin S, Taïani F, Blair G. Exploiting synergies between coexisting overlays. In: Proceedings of the ninth IFIP WG 6.1 international conference on distributed applications and interoperable systems, DAIS ' 09. Springer, Berlin, Heidelberg; 2009. p. 1–15.
Loo BT, Huebsch R, Stoica I, Hellerstein JM. The case for a hybrid p2p search infrastructure. In Proceedings of the third international conference on peer-to-peer systems, IPTPS'. Springer, Berlin, Heidelberg; 2004. p. 141–50.
Lv Q, Cao P, Cohen E, Li K, Shenker S. Search and replication in unstructured peer-to-peer networks. In: Proceedings of the 16th international conference on supercomputing, ICS ' 02. ACM, NY, USA; 2002. p. 84–95.
Malatras A, Peng F, Hirsbrunner B. Energy-efficient peer-to-peer networking and overlays. In: Obaidat MS, Anpalagan A, Woungang I, editors. Handbook on green information and communication systems. MA, USA; Oxford, UK: Academic Press, Elsevier; 2012a. p. 513–40.
Malatras A, Peng F, Hirsbrunner B. A framework for the autonomic management of multi-layer p2p overlays. In: Global information infrastructure and networking symposium (GIIS); 2012. p. 1–8.
Malkhi D, Naor M, Ratajczak D. Viceroy: a scalable and dynamic emulation of the butterfly. In: Proceedings of the 21st symposium on principles of distributed computing , PODC ', 02; ACM, NY, USA; 2002. p. 183–92.
Maniymaran B, Bertier M, Kermarrec A-M. Build one, get one free: Leveraging the coexistence of multiple p2p overlay networks. International conference on distributed computing systems; 2007.
Mao Y, Loo BT, Ives Z, Smith JM. Mosaic: declarative platform for dynamic overlay composition. Comput Netw 2012;56(January (1)):64–84.
Matos R, Marques C, Sargento S, Hummel KA, Meyer H. Analytical modeling of context-based multi-virtual wireless mesh networks. Ad Hoc Netw, 2011.
Matos R, Sargento S, Hummel K, Hess A, Tutschku K, Meer H. Context-based wireless mesh networks: a case for network virtualization. Telecommun Syst 2011:1–14.
Matos R, Marques C, Sargento S. Distributed approach to control and manage context-based multi-virtual networks. Mob Netw Appl 2012;17(August (4)):447–62.
Mawji A, Hassanein H, Zhang X. Peer-to-peer overlay topology control for mobile ad hoc networks. Pervasive Mob Comput 2011;7(4):467–78.
Maymounkov P, Maziéres D. Kademlia: a peer-to-peer information system based on the xor metric. In: Proceedings of first international workshop on peer-to-peer systems, IPTPS '01. Springer, London, UK ; 2002. p. 53–65.
Meshkova E, Riihijärvi J, Petrova M, Mähönen P. A survey on resource discovery mechanisms, peer-to-peer and service discovery frameworks. Comput Netw 2008;52(August (11)):2097–128.
Pechoucek Michal, Jakob Michal, Novak Peter. Towards simulation-aided design of multi-agent systems. In: rem Collier, Dix Jurgen, Novak Peter, editors. Programming multi-agent systems. Lecture notes in computer science, vol. 6599. Berlin, Heidelberg: Springer; 2012. p. 3–21.
Michlmayr E. Ant algorithms for search in unstructured peer-to-peer networks. In: Twenty-second international conference on data engineering workshops; 2006.
Milgram S. The small world problem. Psychol Today 1967;2:60–7.
Monnerat LR, Amorim CL. D1ht: a distributed one hop hash table. In: 20th international parallel and distributed processing symposium (IPDPS 2006); 2006.
Morimoto T, Wakamiya N, Murata M. Proposal and analysis of biologically-inspired symbiotic p2p file-sharing networks. In: 10th WSEAS international conference on automation & information, ICAI'09; 2009. p. 111–7.
Morselli R, Bhattacharjee B, Marsh MA, Srinivasan A. Efficient lookup on unstructured topologies. IEEE J Select Areas Commun 2007;25(1):62–72.
Mrmol FlixGmez, Prez GregorioMartnez. State of the art in trust and reputation models in p2p networks. In: Shen Xuemin, Yu Heather, Buford John, Akon Mursalin, editors. Handbook of peer-to-peer networking. US: Springer; 2010. p. 761–84.
Nakauchi K, Ishikawa Y, Morikawa H, Aoyama T. Exploiting semantics in unstructured peer-to-peer networks. IEICE Trans Commun 2004;7:1806–17.
Niebert N, El-Khayat I, Baucke S, Keller R, Rembarz R, Sachs J. Network virtualization: a viable path towards the future Internet. Wireless Pers Commun 2008;45:511–20.
Parhami B, Rakov MA. Performance, algorithmic, and robustness attributes of perfect difference networks. IEEE Trans Par Distr Syst 2005;16(8):725–36.
Peng F, Malatras A, Hirsbrunner B, Courant M. Antom: constructing multi-layer overlays for pervasive environments. In: Eighth international workshop on mobile P2P computing (MP2P'12). IEEE; 2012. p. 1–6.
Peterson L, Anderson T, Culler D, Roscoe T. A blueprint for introducing disruptive technology into the internet. SIGCOMM Comput Commun Rev 2003;33(January (1)):59–64.
Plaxton CG, Rajaraman R, Richa A. Accessing nearby copies of replicated objects in a distributed environment. In: Ninth ACM symposium on parallel algorithms and architectures, SPAA'97; 1997. p. 311–20.
Pugh W. Skip lists: a probabilistic alternative to balanced trees. Commun ACM 1990;33(6):668–76.
Puttaswamy KPN, Zhao BY. 2007. A case for unstructured distributed hash tables. In: IEEE global Internet symposium. p. 7–12.
Qiao Y, Bustamante FE. Structured and unstructured overlays under the microscope: a measurement-based view of two p2p systems that people use. In: Proceedings of USENIX ' 06 annual technical conference, ATEC '06. USENIX, Berkeley, CA, USA; 2006.
Qiu D, Srikant R. Modeling and performance analysis of bittorrent-like peer-to-peer networks. SIGCOMM Comput Commun Rev 2004;34(August (4)):367–78.
Qiu T, Chan E, Chen G. Overlay partition: iterative detection and proactive recovery. In: IEEE international conference on communications, 2007. ICC' 07; 2007. p. 1854–9.
Ratnasamy S, Francis P, Handley M, Karp R, Shenker S. A scalable content-addressable network. SIGCOMM Comput Commun Rev 2001;31(August 4):161–72.
Rhea S, Geels D, Roscoe T, Kubiatowicz J. Handling churn in a dht. In: Proceedings of USENIX annual technical conference, ATEC ' 04. USENIX, Berkeley, CA, USA; 2004.
Ripeanu M. Peer-to-peer architecture case study: gnutella network. In: First international conference on P2P computing; 2001. p. 99–100.
Ripeanu M, Iamnitchi A, Foster I, Rogers A. In search of simplicity: a self-organizing group communication overlay. Concurr Comput Pract Exp 2010;22(May (7)):788–815.
Risson J, Moors T. Survey of research towards robust peer-to-peer networks: search methods. Comput Netw 2006;50(17):3485–521.
Ritter J. Why gnutella can't scale. No, really. Technical report; February 2001.
Rodrigues R, Druschel P. Peer-to-peer systems. Commun ACM 2010;53(October (10)):72–82.
Rowstron AIT, Druschel P. Pastry: scalable, decentralized object location, and routing for large-scale peer-to-peer systems. In: Proceedings of the IFIP/ACM international conference on distributed systems platforms, Middleware '01. Springer, London, UK; 2001. p. 329–50.
Saha D, Dutta A, Grunwald D, Sicker D. Gratis: free bits in the network. IEEE Trans Mob Comput 2015;14(January (1)):72–85.
Saroiu S, Gummadi PK, Gribble SD. Measurement study of peer-to-peer file sharing systems. In: Society of photo-optical instrumentation engineers (SPIE) conference series, vol. 4673; December 2001. p. 156–70.
Schonwalder J, Fouquet M, Rodosek G, Hochstatter I. Future internet = content + services + management. Commun Mag IEEE 2009;47(July (7)):27–33.
Serbu S, Felber P, Kropf P. Hypeer: structured overlay with flexible-choice routing. Comput Netw 2011;55(1):300–13.
Shen K. Structure management for scalable overlay service construction. In: Proceedings of the first symposium on networked systems design and implementation, NSDI'04. USENIX, Berkeley, CA, USA; 2004.
Shen H, Xu C-Z, Chen G. Cycloid: a constant-degree and lookup-efficient p2p overlay network. Perform Eval 2006;63(March (3)):195–216.
Shrestha SL, Lee J, Chong S. Virtualization and slicing of wireless mesh network. In: International conference on future Internet technologies (CFI); June 2008. p. 1–4.
Snyder P, Greenstadt R, Valetto G. Myconet: a fungi-inspired model for superpeer-based peer-to-peer overlay topologies. In: Third international conference on self-adaptive and self-organizing systems; 2009. p. 40–50.
Steiner M, En-Najjary T, Biersack EW. Exploiting kad: possible uses and misuses. SIGCOMM Comput Commun Rev 2007;37(October (5)):65–70.
Stoica I, Morris R, Liben-Nowell D, Karger DR, Kaashoek MF, Dabek F, et al. Chord: a scalable peer-to-peer lookup protocol for internet applications. IEEE/ACM Trans Netw 2003;11(February (1)):17–32.
Stutzbach D, Rejaie R. Understanding churn in peer-to-peer networks. In: Proceedings of the sixth ACM SIGCOMM conference on Internet measurement, IMC '06. ACM, NY, USA, 2006. p. 189–202.
Sun X, Li K, Liu Y, Tian Y. Slup: a semantic-based and location-aware unstructured p2p network. In: 10th IEEE international conference on high performance computing and communications; 2008. p. 288–95.
Tan G, Jarvis SA. Inter-overlay cooperation in high-bandwidth overlay multicast. In: International conference on parallel processing, 2006. ICPP 2006; August 2006. p. 417–24.
Urdaneta G, Pierre G, van Steen M. A survey of dht security techniques. ACM Comput Surv 2011;43(February (2)):8:1–49.
Wakamiya N, Murata M. Overlay network symbiosis: evolution and cooperation. In: First international conference on bioinspired models of network, information and computing systems, BIONETICS'06. ACM; 2006.
Wallach D. Software security theories and systems. In: Okada M, Pierce B, Scedrov A, Tokuda H, Yonezawa A, editors. A survey of peer-to-peer security issues. Lecture notes on computer science, vol. 2609. Berlin, Heidelberg: Springer; 2003. p. 42–57.
Wang S, Xuan D, Zhao W. Analyzing and enhancing the resilience of structured peer-to-peer systems. J Par Distr Comput 2005;65(2):207–19.
Watts D, Strogatz S. Collective dynamics of 'small-world' networks. Nature 1998;393:440–2.
Weiser M. The computer for the 21st century. Sci Am 1991;265(September (3)):94–104.
Wouhaybi RH, Campbell AT. Phenix: supporting resilient low-diameter peer-to-peer topologies. In: 23rd joint conference of the IEEE computer and communications societies (INFOCOM 2004); March 2004.

Wu C, Li B. Strategies of conflict in coexisting streaming overlays. In: INFOCOM 2007. 26th IEEE international conference on computer communications. IEEE; May 2007. p. 481–9.

Wu C-J, Yang K-H, Ho J-M. Antsearch: an ant search algorithm in unstructured peer-to-peer networks. In: Proceedings of the 11th IEEE symposium on computers and communications (ISCC); 2006. p. 429–34.

Xu J, Kumar A, Yu X. On the fundamental tradeoffs between routing table size and network diameter in peer-to-peer networks. IEEE J Select Areas Commun 2006;22(September (1)):151–63.

Yang B, Garcia-Molina H. Improving search in peer-to-peer networks. In: Proceedings of the 22nd international conference on distributed computing systems; 2002. p. 5–14.

Zhang X, Zhang Q, Zhang Z, Song G, Zhu W. A construction of locality-aware overlay network: moverlay and its performance. IEEE J Select Areas Commun 2004;22 (1):18–28.

Zhang X, Liu J, Li B, Yum Y-SP. Coolstreaming/donet: a data-driven overlay network for peer-to-peer live media streaming. In: Proceedings of the IEEE INFOCOM, vol. 3; March 2005. p. 2102–111.

Zhao BY, Ling Huang, Stribling J, Joseph AD, Kubiatowicz JD. Exploiting routing redundancy via structured peer-to-peer overlays. In: Proceedings of 11th IEEE international conference on network protocols; 2003. p. 246–57.

Zhao BY, Huang Ling, Stribling J, Rhea SC, Joseph AD, Kubiatowicz JD. Tapestry: a resilient global-scale overlay for service deployment. IEEE J Select Areas Commun 2004;22(January (1)):41–53.

Zhong Weilin, Evans David. When ants attack: security issues for stigmergic systems. Technical report. Charlottesville, VA, USA; 2002.