Alleviating the Topology Mismatch Problem in Unstructured and Structured Overlay Networks: A Survey

Vassilis Moustakas[1], Hüseyin Akcan[2], Mema Roussopoulou[1] and Alex Delis[1]
[1]Department of Informatics and Telecommunications,
National and Kapodistrian University of Athens
{b.moystakas, mema, ad}@di.uoa.gr
[2]Department of Software Engineering,
Izmir University of Economics, Izmir, Turkey
huseyin.akcan@ieu.edu.tr

1. INTRODUCTION

This survey begins with a quick review of the conception of the overlay network and the available architectures of p2p systems. It continues with the identification of the phenomenon of the topology mismatch between overlay and underlying networks and its impact on the utilization of network resources. Ultimately, it reviews the most important resources found in the literature that attempt to tackle the problem as it proposes a taxonomic scheme, using a set of supertype-subtype relationships based on unique characteristics they have and/or specific goals they target.

2. OVERLAY NETWORKS AND P2P SYSTEMS

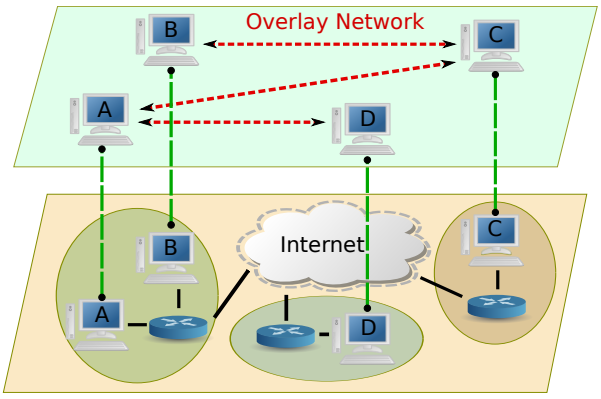

Fig. 1. An example overlay network.

An *overlay network* is an abstract, logical interconnection of entities formed on top of another network. It consists of a set of nodes, the computing elements, that are being connected by virtual (logical) point-to-point links. For example, the *Asymmetric Digital Subscriber Line (ADSL)* is, actually, an overlay network on top

of the *Public Switched Telephone Network (PSTN)*. Figure 1 illustrates an example physical ethernet network and the corresponding overlay network formed by four nodes. Nodes A and B are on the same local network, while C and D are in different networks. The top layer represents the overlay network formed by these nodes, which is a reflection of the application layer connections among these nodes. As the topology of the network can change based on the application, in this example, it is easy to observe that even though nodes A and B are in the same network, their communication goes through node C on another network, demonstrating a non-optimal mapping between the physical and the overlay networks. An overlay network is generally useful when abstraction of the network layer from the application layer is important for applications, particularly when the underlying network structure is subject to frequent changes.

The abstraction of overlay networks has been proposed as a way to implement efficient, fully distributed, application layer services on top of a best-effort IP layer forming a widely ranged family of protocols collectively known as *peer-to-peer* or *p2p* for short. P2p network systems gained significant attention, in recent years, due to their plethora of unique features in supporting file sharing among huge numbers of (inter-) networked computers called *peers*, where each, such, *peer* could act both as a resource provider and a resource consumer. This changed the traditional *client-server* model dominating the internet and lead to the introduction of the *servent*-concept [Gnutella], a portmanteau that blends the notions of *server* and *client* to denote the twofold role of the participants in a p2p network. Such serverless systems, proved to be able to achieve outstanding aggregate resource capacities as participants join the system without requiring additional expenditure for infrastructure.

As p2p systems evolved, three main architectures have been emerged, namely:

centralized
decentralized unstructured
decentralized structured

Centralized architectures were the first to recognize that requests for popular content need not be sent to a central server but instead could be handled by the many hosts that already posses the content. *Napster*, for example, maintained a *central index server* based on file lists provided by participating peers. The central index server was queried by the users and it returned pointers to the actual content. Thus, by centralizing search while distributing downloads, Napster achieved a highly functional design that was widely acknowledged, at the time, as "the fastest growing Internet application ever". This centralized search facility, though, renders this scheme not fully distributed and can be proven its "Achilles' heel" in terms of being a scalability bottleneck, a central point of failure and vulnerable to malicious acts (e.g. *Denial-of-Service (DoS) attacks*).

The centralized approach has been eventually replaced by architectures that distributed both the search and the download capabilities. In these *decentralized unstructured* architectures, file placement is random, which means there is no correlation with the network topology whatsoever [Yang and Garcia-Molina 2002]. The most important properties of such systems are that they support inherent heterogeneity of peers, are highly resilient to peer failures, and incur low maintenance

overhead at handling the dynamics of peer participation [Stutzbach and Rejaie 2006]. In the literature, they are also known as *broadcast-based* systems, because they use *message flooding* among peers (i.e. Gnutella[Gnutella]) or among super-peers (i.e KaZaA[Kazaa]) to propagate queries. Even though the unstructured approach became highly popular among file sharing applications, they do have a certain disadvantage in locating rare objects due to the unstructured nature and the limitations of the flooding approach.

Recently, *decentralized structured* schemes have been proposed in order to provide a self-organising infrastructure for large-scale p2p applications [Ratnasamy et al. 2001; Stoica et al. 2001; Rowstron and Druschel 2001; Zhao et al. 2001; Maymounkov and Mazières 2002]. They implement a *Distributed Hash Table (DHT)* that maps objects to nodes through a deterministic mechanism. The main advantage of the DHT approach of decentralized structured p2p networks over decentralized unstructured ones, is that they provide a guaranteed bound on the number of overlay routing hops that have to be taken in order for an object to be located even in the case where only a single copy exists in the system. This is $O(logn)$ compared to Gnutella, that requires $O(n)$ to reliably locate a specific object.

Initial efforts targeted on the implementation of one-to-many addressing schemes that could replace *IP multicast* in providing higher quality of streaming media through *quality of service (QoS)* guarantees. IP multicast as well as other proposals such as *IntServ* and *DiffServ* [Eisner 2005] have not seen wide acceptance (yet?), largely because they require changes to all routers of the network. On the other hand, an overlay network can be incrementally deployed on end-systems running the overlay protocol software, without cooperation from the ISPs. Academic research includes [Chu et al. 2000; Jannotti et al. 2000; Kwon and Fahmy 2002] among others. Overlay networks were also implemented in order to back the routing of messages to destinations whose address is not known in advance. This resulted in the appearance of well known peer-to-peer protocols such as [Gnutella] and [Maymounkov and Mazières 2002] widely used for digital content sharing among their network nodes. Other special purpose overlay networks include [Andersen et al. 2001] for resilient routing; [Subramanian et al. 2004] for quality of service guarantees; and [Clarke et al. 2001] for anonymity; to name just a few.

The main focus of this paper is the topology mismatch problem, therefore the available approaches in overlay networks are briefly summarized, and interested readers are pointed to the available surveys on overlay networks [Androutsellis-Theotokis and Spinellis 2004; Lua et al. 2005].

3. THE TOPOLOGY MISMATCH PROBLEM

One of the major issues that defines the efficiency of the overlay network is the mapping of the physical links and the overlay paths. Ideally, overlay networks are expected to achieve an optimal mapping among the overlay paths and the underlying physical links, and avoid the inefficient mapping that is shown in Figure 1. The problem of constructing an optimal overlay is referred to as the *topology mismatch problem*, and formally defined as follows:

DEFINITION 3.1. *Let* $V = \{v_1, ..., v_n\}$ *be a set of points denoting the network nodes,* $\{v_i, v_j\} \in E$ *be the set of unicast distances between nodes* v_i *and* v_j*,* $G =$

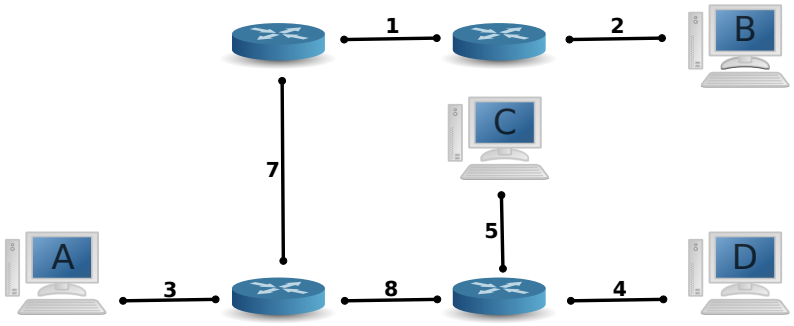

Fig. 2. Interconnection of nodes in the physical level.

(V, E) be a complete distance graph over V. The topology mismatch problem is to construct a minimal spanning tree, where node degree is restricted to a constant $(k \geq 2)$ by the bandwidth of each node v_i.

Constructing degree constrained spanning tree is known to be NP-Hard [Garey and Johnson 1979], as well as the topology mismatch problem [Chawathe 2000]. Moreover, the internet is an interesting environment where end-to-end latencies demonstrate triangle inequality violations (TIVs), which further complicates the problem. These delay space TIV is a consequence of the Internet's structure and routing policies and thus will remain a property of the Internet for the foreseeable future [Zheng et al. 2005]. TIVs affect network coordinate [Cox et al. 2004; Wong et al. 2005] and positioning [Ng and Zhang 2001] and makes difficult the construction of delay aware overlays. A number of heuristic approaches have been proposed in an effort to finding an algorithm that performs reasonably and consistently well.

A topology unaware overlay network is able to control the sequence of peers a message traverses before reaching its destination, but it is completely unaware of how the actual packets are switched at the underlying infrastructure along the overlay path. For example, a single logical point-to-point link on the overlay, most of the time, corresponds to multiple physical links in the underlying layer. Additionally, a link in the underlying network often serves the mapping of several overlay paths causing increase of the traffic on the physical link, which is also called the link's *stress*[Chu et al. 2002]. Furthermore, the stochastic behaviour of peers randomly joining and leaving the peer-to-peer network all cause the overlay to physical mapping to create such unnecessary, redundant maintenance traffic that impacts the efficiency of the network in terms of average response time.

For example, assume nodes A, B, C and D are connected through the physical network shown in Figure 2, where the network costs are given in milliseconds, and peer A sends a message to peer D. If these peers participate in an overlay network according to one of the setups of Figures 3 (left) and 3 (right) then users will yield different performances. In the first overlay shown in Figure 3 (left), the message will traverse the following sequence of links in the physical layer (marked with their costs): $3 \rightarrow 7 \rightarrow 1 \rightarrow 2 \rightarrow 2 \rightarrow 1 \rightarrow 7 \rightarrow 8 \rightarrow 5 \rightarrow 5 \rightarrow 4$. In the second overlay shown in Figure 3 (right), the path will be: $3 \rightarrow 8 \rightarrow 4$. The total cost of the first path is $45ms$, while the second only costs a mere $15ms$. Therefore, we can

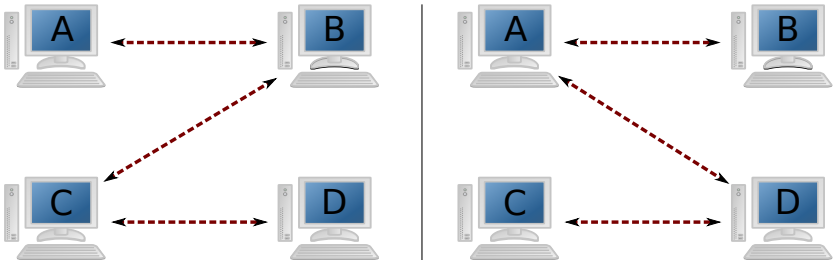

Fig. 3. HA: Put subfigure, and a new caption here

conclude that the second overlay is more congruent with the underlying physical network than the first one and thus more efficient.

Early incarnations of the overlay protocols, however, did not make use of the optimal mapping with the physical network topology. Early Gnutella protocol, though were considered far from scalable [Ritter 2001], randomly choose logical neighbours without knowledge of the underlying network, effectively causing mismatch between physical and application-level topologies. Additionally, queries may be flooded to multiple paths merging in one peer while in such case one of the paths would have been enough.

Similar problems are observed in decentralized structured schemes also. Typically, node IDs are assigned *randomly*, resulting in excellent load balancing, scalability and robustness. Unfortunately, this randomness has a negative impact on the *routing locality* of the network. This means that even though the target node can be reached with logarithmic overlay hops, the distance traveled in the physical underlying network, during the overlay routing process, can be far from optimal. The ratio of the physical distance, a query travels through the overlay to an object, and the minimal distance to that object (i.e. though IP) is known in the literature as *stretch* or *Relative Delay Penalty (RDP)* [Chu et al. 2000].

Studies on peer-to-peer traffic reveals the effect of the peer-to-peer traffic on the overall internet. Measurements [Saroiu et al. 2002; Sen and Wang 2004] on some popular systems, such as FastTrack, Gnutella and DirectConnect have shown that the peer-to-peer traffic contributes the largest portion of the overall Internet traffic. [Ripeanu et al. 2002] showed that, even given that 95% of any two nodes are less than 7 hops away from each other, a flooding-based algorithm, like the one used by Gnutella, can generate 330TB/month in a 50.000 node network! Therefore, the efficiency of the overlay networks has serious consequences on the overall well-being of the modern internet.

Having understood the nature of the problem that peer-to-peer architectures face, Section 4 and 5 discuss the recent academic work that has been conducted in the field. What follows does not claim to be a thorough citation of all known protocols that are available out there, but instead, a careful selection of those that left a distinctive fingerprint contribution in the efforts of the research community to alleviate the topology mismatch problem.

4. UNSTRUCTURED DECENTRALIZED ALGORITHMS

In this section, the unstructured decentralized P2P algorithms are analysed based on their architecture and base methodology to tackle the topology mismatch problem. The unstructured decentralized algorithms are visited and categorized based on their use of the overlay structure, messaging between nodes, and on the techniques proposed to detect proximity to optimize the topology. The base methods that unstructured P2P algorithms use to tackle the topology mismatch problem can be categorized as Broadcast, Cache, Overlay Optimization, and Landmark Based. In this section, these methodologies are discussed in detail and the algorithms in each category are highlighted.

4.1 Broadcast Based

In the naive Gnutella protocol, each received query request is forwarded to all the neighbours, which clearly generates unnecessary traffic over the network. The algorithms categorized in the *broadcast based* methodology prefers re-broadcasting the query messages to a selected subset of their neighbours, where the selection criteria varies between algorithms. The selection is made using one or a combination of various statistical metrics, such as the reliability or the latency of the link between nodes, etc. The forwarding based approach enhances search efficiency but has also the drawbacks of having drastically reduced search scope due to the selective re-broadcasting policy, and scalability problems on large P2P networks.

The forwarding based P2P algorithms and their details are presented below.

Improving search in peer-to-peer networks. [Yang and Garcia-Molina 2002] proposes an easy to implement and practical solution to the inefficiency problem caused by blind flooding in Gnutella like P2P file sharing protocols. The paper replaces blind flooding with three approaches, namely *iterative deepening*, *directed BFS*, and *local indices*. In *iterative deepening*, the search is performed on a BFS tree with multiple preset depths. The depth limit is iteratively increased by the source node for each query based on the quality of the results. The source node, after examining the results, may issue a new request by increasing the depth limit, which will trigger the nodes at the last depth level to resume the search. The iterative approach avoids to start the search from scratch at each iteration and reduces the load of the nodes on the upper levels of the tree. The major drawback of *iterative deepening* is the delay between successive iterations, as the source node needs to examine the results at each iteration before deciding to quit or resume the query. The *directed BFS* tries to avoid this delay by forwarding the query messages only to a selected set of neighbours, in which the selection criteria varies from the number of results received previously, distance in terms of hops, bandwidth, or the query load of the neighbour. In *local indices*, each node sustains a local data index of the nodes within a radius of r hops of itself and uses this local index to remotely query the neighbour nodes without generating additional traffic. *Local indices* greatly reduces the aggregate bandwidth usage of the network and improves query efficiency, however, updating the indices in cases with frequent node joins and leaves introduces a serious overhead to the system if the radius is kept broad.

Gia. *Gia* [Chawathe et al. 2003] is an effort to solve the scalability problem of the unstructured P2P file sharing systems, in particular Gnutella. The main novelty in the design of Gia is the replacement of blind flooding approach of Gnutella with random walks [Lv et al. 2002]. Although random walks is a step in the right direction, issuing only a single copy of the query within the network reduces the search scope, thus affects the success rate of the query. In order to overcome this limitation, Gia introduces a token-based flow control algorithm, which is essentially an intelligent flow control algorithm that gradually redirects the queries to nodes with more chance of answering. Gia also acknowledges the heterogenity in peer bandwidth, processing power, disk speed, etc, of the nodes in P2P networks and uses this heterogenity when connecting nodes to each other, such that by using a topology adaptation algorithm, Gia ensures that high capacity nodes have high degrees and low capacity nodes are within short proximity of high capacity nodes. In order to prevent overloading of nodes with query requests, Gia uses a token-based flow control algorithm in which each node announces the number of query requests it can handle in terms of tokens to its neighbours, so that neighbours only forward query requests to nodes that they previously received tokens from. Although the topology adaptation algorithm Gia uses improves the scalability of the network, it does not help much in solving the topology mismatch problem, since it does not consider the underlying physical topology.

Distributed Cycle Minimization Protocol. The flooding based query approach of Gnutella creates many duplicate packets in the network. Nodes receive the same query request from various neighbours, which they generally discard when identified as duplicate. One other reason for duplicate messages is the cycles in the forwarding paths. These duplicate packets affect the performance of the P2P network negatively by increasing the overall resource usage. Moreover, the more active nodes, that have high capacity, high bandwidth, or contribute more to the network, suffer more from the overhead caused by the duplicate packets. Therefore, *Distributed Cycle Minimization Protocol (DCMP)* [Zhu et al. 2008] is introduced as a method to remove the cycles and eliminate the duplicate packets, without sacrificing the connectivity. In DCMP, once a cycle is detected, the most powerful node in the cycle is elected as the *Gate Peer* and the cycle is cut from a special link such that the distance of all the nodes within the cycle to the *Gate Peer* is minimized. The process is managed by using two special message types, *Information Collection (IC)* and *Cut Message (CM)*. One disadvantage of DCMP is that since the distance a forwarded message can travel is limited with the TTL value, which is practically 7 in Gnutella, the DCMP cannot detect cycles of length larger than 7. Even though cycle elimination improves the network performance, it does not solve the topology mismatch problem.

4.2 Cache Based

Caching is a widely used methodology to exploit the locality to minimize the redundant transfer of data, which is successfully used in web servers and file servers. As peers in a P2P system also work as servers, intuitively it is expected that P2P file sharing systems can also benefit from caching. However, one of the most important characteristic properties of P2P systems is that nodes join and leave frequently, and

the lifetime of a query is relatively short compared to web servers, which makes caching in P2P networks nontrivial. There are usually two levels of caching possible in file sharing systems, the indices or pointers to data can be cached, or the data itself can be cached. Caching is already implemented and successfully used in some commercial P2P systems. One popular example is KazaA, which uses caching of indices in its super peers. We present the other protocols that use caching and their details below.

Replication Strategies in Unstructured Peer-to-Peer Networks. [Cohen and Shenker 2002] aim to improve the inefficient blind search algorithm by replicating the data in a peer to peer network. The main intuition behind the idea of replication, or using cached copies, is that as the number of copies for each item increases in the network, it would be easier for a search algorithm, even a random one, to locate these items. In order to analyse the feasibility of such replication approach, the authors investigate three different replication strategies, namely uniform, proportional, and square root allocation. In the uniform model, the copies of items are uniformly replicated in the network, while in the proportional model, the items are replicated based on their query rate, so that frequently queried items are replicated more. The square root allocation is a strategy proposed by the authors, which is a model between the uniform and proportional allocation. In order to evaluate the outcomes of the replication models, the authors compare the overall costs of successful and unsuccessful searches in the network. The results can be summarized as follows; the uniform allocation model minimizes the maximum search size, therefore reduces the time spent on an unsuccessful search. The proportional model, on the other hand, promotes the more frequently queried items by replicating them more, therefore decreases the search time for popular items, but suffers when locating the rare items. The authors also claim that the expected successful search size is the same for uniform and proportional models, and any approach between them would behave much better. Therefore they propose the square root allocation approach, which is a replication strategy that minimizes the expected search size of successful queries in P2P networks.

Tracing a large-scale Peer to Peer System: an hour in the life of Gnutella. [Markatos 2002] analysis the network traces of the Gnutella network and detects that query requests in Gnutella has locality, and proposes a caching algorithm that exploits this locality. The analysis of the trace data reveals other important facts about the structure of the Gnutella network and the query data. One significant observation is that the geographic locations of clients do not have a correlation with the number of query requests they receive. This is an obvious result of the topology mismatch problem caused by the overlay structure of the Gnutella network. Gnutella traffic is observed to be bursty both for query requests and query responses, even in longer intervals, and nine out of ten queries do not generate any response due to the inefficient design of the Gnutella network. When developing a caching system to exploit locality, applying an approach similar to web caching does not fit well with P2P systems. The caches in P2P systems not only have to consider the query string, but also the TTL value, the source of the query, and the time of the query as well. In general, even though optimum caching

is hard to achieve, it is reported that caching improves the overall performance of the Gnutella system.

4.3 Overlay Optimization Based

The overlay optimization based protocols modify the topology of the P2P network using various techniques. The two commonly used approach that is investigated in this section are creating spanning trees using connection graphs, and creating cluster of physically close nodes. Spanning tree based approaches construct rich graphs based on the network connections and build minimum spanning trees on the graphs. Even though the spanning tree provides efficient query performance, the construction and the update costs of the spanning tree, especially in dynamic environments with many nodes joining and leaving the network, cause large traffic overhead to the system [Chu et al. 2000; Chu et al. 2002].

The cluster based approaches on the other hand select to link physically closer nodes with each other, therefore shrink the search scope significantly. However, commonly used methods for determining the correct physical positions of nodes over the internet does not always return reliable results, therefore mapping accuracy is not always guaranteed.

Below, the details of the algorithms that use the spanning tree based or cluster based topology optimization methods are presented.

Narada. Narada [Chu et al. 2000; Chu et al. 2001; 2002] is a generic protocol to design self adapting overlay networks, that can achieve application layer multicast without requiring IP multicast at the network layer. Although Narada is not originally designed as a peer-to-peer file sharing application protocol, it has become one of the pioneering works in the overlay networks area. Even though the design of the IP multicast protocol is finalized a while ago, and some of the routers on the internet today already support this protocol, the IP multicast idea did not take off as anticipated. One of the main reasons for this delay is that the IP multicast violates the stateless design of the current internet. Although the multicast is not widely used, and the future of the protocol is not bright, there are multiple applications that need this kind of service, such as IP TVs, P2P networks, streaming services, etc. The application layer multicast is then a valid alternative for these applications. The authors claim that, although Narada is not as efficient as IP multicast, still it provides reasonable performance, without the additional cost of implementing multicast services on the current internet infrastructure. The main reason for the inefficiency is due to the Topology mismatch problem. Narada tries to solve this problem by building a richer connected graph[1], called a mesh, and building per source minimum spanning trees. Narada also maintains the graph and the trees dynamically updated on node joins and leaves. Moreover, Narada tries to optimize the physical link stress, the overall resource usage and the relative delay among end systems. The main limitation of Narada is that although it works reasonably well for small groups, it does not scale well for larger networks, therefore it is not suitable for P2P file sharing applications which can easily scale to millions of nodes.

[1]ENarada [Xing-feng et al. 2008] used Gossip protocol for the construction

Adaptive Overlay Topology Optimization and Adaptive Connection Establishment. *Adaptive Overlay Topology Optimization (AOTO)* [Liu et al. 2003] is one of the first attempts, along with Narada, from the P2P research community to address the topology mismatch problem. AOTO is a distributed algorithm that uses *Selective Flooding* and *Active Topology* to optimize the overlay network topology. The *Selective Flooding* algorithm creates a minimum spanning tree over the overlay and uses this tree instead of flooding the whole network, without shrinking the search scope. In *Active Topology*, the physical locations of nodes are estimated based on the network delay among them and physically close nodes are connected as neighbours to revise the overlay topology as close as possible to the physical network topology. Overall, AOTO reports a 55% performance improvement in terms of the traffic load. In *Adaptive Connection Establishment (ACE)* [Liu et al. 2004], the authors extend the idea by introducing optimizations based on the depths of the minimum spanning trees. As the network delay is not always a reliable estimation method to detect physical locations of the nodes, the algorithm still suffers from the discrepancies caused by mislocated nodes.

Location-aware Topology Matching. Peer-to-peer algorithms designed without considering the topology mismatch problem generates unnecessary heavy traffic on the internet infrastructure, due to the popular use of these P2P programs for file sharing. *Location-aware topology matching (LTM)* [Liu et al. 2004] proposes a method to optimize the overlay structure of the P2P network based on the physical topology. In doing so, LTM issues a special message called *TTL2-detector* to detect the latency of N^2 neighbours around each node. The latency information gathered is later used to evolve the overlay network into a more efficient one, without reducing the search scope. Each node examines the latency information with the neighbours and low productive connections are dropped and replaced by more efficient ones, thus reducing the latency on the overall network, and eliminating some potential inefficiencies, such as unnecessary crossing of messages over autonomous system (AS) boundaries, etc. Although the LTM approach improves the overall efficiency of the P2P network, since it uses the latency to detect close by nodes, does not use the real physical topology information, therefore does not offer a guaranteed bound to the topology mismatch problem.

Scalable Bipartite Overlay. *Scalable Bipartite Overlay (SBO)* [Liu et al. 2004; 2007] improves the overhead of creating and maintaining a minimum spanning tree cost by randomly dividing the nodes into two groups, red and white, and assigning different tasks to different groups. The white peers measure distances to neighbours by using the network delay as the metric and reports the results to red peers. The red peers, equipped with the information of all two hop (N^2) neighbours, creates a minimum spanning tree of these neighbours and assigns efficient forwarding paths. The white peers can further optimize their positions within the overlay if need be.

Peer-exchange Routing Optimization Protocols. [Qiu et al. 2007] introduces two protocols called Peer-exchange Routing Optimization Protocols (PROP) to adjust the neighbourhood graph of the overlay network in order to reduce the overall link latency of the network. The PROP algorithms are based on the exchange of neighbours among peers, which is triggered by the mutual benefit of both

peers to reduce the network delay. In PROP-G (Generic), peers exchange all their neighbours with another peer, while in PROP-O (Optimized) only selected number of neighbours are exchanged among peers. The PROP-G is a generic protocol that guarantees the connectivity of the overlay graph during exchanges, therefore can be applied to both unstructured and structured overlay networks.

T2MC. T2MC [Shi et al. 2008] uses traceroute logs to detect the AS boundaries and cluster close by peers with each other to reduce the redundant multiple message passes between the AS boundaries. T2MC uses a customized k-mean classification algorithm with $k = 2$ to perform the classification, and exploits the stable structure of the internet routers to guide clustering. Even though traceroute provides detailed information about the network structure, use of traceroute creates overhead to the overall network structure. For this same reason, it is not unusual for network administrators to disable this support on their network routers, which may affect the performance of the T2MC algorithm.

Unnamed Unstructured!!

.

4.3.0.1 . In [Hsiao et al. 2009], Hsiao et al, claim to construct topology-aware unstructured overlays that *guarantee* performance qualities in terms of *i*) the expected communication latency among any two overlay peers regardless of the network size, and *ii*) the broadcasting scope of each participating peer.

The algorithm constructs an undirected graph $G = (V, E)$ comprised by two subgraphs. The first, namely $G^{(red)} = \left(V^{(red)}, E^{(red)}\right)$ in the paper's context, includes all vertices of G and ensures the connectivity of the graph by securing at least one path between any two nodes. In contrast, $G^{(blue)} = \left(V^{(blue)}, E^{(blue)}\right)$, contains those vertices of G that have free edges to link to other nodes and because these are fully utilized, the following also stands $E = E^{(red)} \cup E^{(blue)}$.

A joining peer u, partitions its neighbours into two subsets, the $B_u^{(red)}$ and $B_u^{(blue)}$. In order to populate the $B_u^{(red)}$ subset, peer u samples peers uniformly and at random. Then, each of these selected peers discovers a routing path starting from itself towards the node with the smallest (or the largest) ID in the system.

Distributed Domain Name Order. *Distributed Domain Name Order (DDNO)* [Zeinalipour-Yazti and Kalogeraki 2006] uses the domain names to detect topologically close nodes based on the assumption that nodes within the same domain are topologically close to each other. Half of the possible connections of a node is used to connect to these local peers, and the other half is used to randomly connect to the peers anywhere on the network. *DDNO* is a heuristic approach to solve the topology mismatch problem, with local connections to improve the efficiency and reduce the delay on the network. The random connections on the other hand help ensure the connectivity in the network and avoid partitioning. *DDNO* uses *Split-Hash* and *dnMatch* algorithms to detect locality by using domain names.

4.4 Landmark Based Proximity

Detecting the proximity of a node using landmarks, also known as *Landmark Clustering*, is based on the view that nodes with similar distances to a set of predefined

well-known landmark nodes are pretty likely of being close to each other. But this approach has its weaknesses as well, such as the fact that is a rather coarse grained approximation, therefore not particularly well suited for detecting the correct positions of nodes within close distance to each other.

Landmark Binning. Landmark Binning [Ratnasamy et al. 2002] is an approach to partition close by nodes into bins based on their distance to well known anchor nodes within the internet. In order to detect locality, nodes use network latency (round trip time) as a measurement technique. The network latency, even though not always accurate, is selected in this work because its non-intrusive, transparent and easy to apply structure. In order for the binning to work, authors assume that there are a few well known anchor servers with known physical locations on the internet. The authors estimate that around 12 servers will do the job. The nodes measure their distance to all these servers and record the latency values for each. Later, the tuple of these latency values represents the bin of each node. Nodes with similar latency values are assumed to be close to each other, or at least within the same AS. The landmark binning is designed as a generic model that can work both on structured and unstructured P2P networks, and [Ratnasamy et al. 2002] has the detailed description of the algorithm for both topologies. As the main operation of the algorithm is independent from the underlying model, and does not change for structured and unstructured networks, we classify and give the details of the algorithm in this decentralized unstructured section, and omit the duplicate work in the decentralized structured section. The landmark binning is also proposed as a good candidate to work on content distribution networks. The major disadvantage of landmark binning is to install and maintain landmark servers on different AS, all over the world. A typical P2P network usually has a couple of million nodes connected at any time, which introduces possible scalability problems in terms of the landmark servers. The authors claim that latency estimation does not use much network resources and in order to avoid the possible scalability problems the authors propose to replace single landmark servers with clusters of servers within the same physical area. However, this approach does not reduce the possible high network traffic flow through these landmark servers. One other possible problem is the incorrect binning caused by inaccuracy in delay measurements methods, as the network latency is not a proven accurate method for location estimation.

mOverlay. mOverlay [Zhang et al. 2004] addresses the scalability problems that affect static landmark servers by introducing the dynamic landmarks. The nodes are divided into groups based on their distances to these groups and the neighbour nodes in these groups behave as dynamic landmarks. The mOverlay protocol is composed of two main parts, initially adding a new node to the overlay by selecting the closest group, and maintaining the overlay afterwards. Finding the correct closest group is the most important part of the overlay construction. The authors also formally prove that any new node can reach its group by performing at most $O(logN)$ communications within the network.

Table I: Decentralized Unstructured Algorithms

Algorithm	Arch.	Overlay structure	Base protocol	Dynamic update	Runtime	Scalability	cites
Narada	Decentralized unstructured	**Overlay optimization based**. Creates a mesh (richer connected graph) and builds minimum spanning trees on this mesh		Yes. Supports incremental evolving		Small and sparse groups	2000
Gia	Decentralized unstructured	**Forwarding based** Replaces Gnutella flooding with random walk, and introduces KaZaA style supernodes. Uses dynamic topology adaptation protocol	Gnutella	Yes		Better than Gnutella	974
Adaptive Overlay Topology Optimization	Decentralized unstructured	**Overlay optimization based**. Creates overlay multicast tree with Selective Flooding protocol	Gnutella	Yes, using Active Topology protocol		Better than Gnutella	36
Location-aware Topology Matching	Decentralized unstructured	**Overlay Optimization Based**. Uses *TTL2-detector flooding*, *low productive connection cutting*, and *source peer probing*.	Gnutella	Yes		Better than Gnutella	204
Replication Strategies in Unstructured P2P Networks	Decentralized unstructured	**Cache Based**. Uses uniform, proportional and square root allocation strategies to replicate data.	Gnutella	Yes		Better than Gnutella	580

Continued on next page

Table I – *Continued from previous page*

Algorithm	Arch.	Overlay structure	Base protocol	Dynamic update	Runtime	Scalability	cites
Tracing a large-scale Peer to Peer System: an hour in the life of Gnutella.	Decentralized unstructured	**Cache Based**. Proposes a caching algorithm based on the traces of the Gnutella traffic	Gnutella	Yes		Better than Gnutella	199
Improving search in P2P networks	Decentralized unstructured	**Forwarding Based**. Uses *iterative deepening*, *directed BFS*, and *local indices* to improve efficiency.	Gnutella	Yes		Better than Gnutella	380
Distributed Cycle Minimization Protocol	Decentralized unstructured	**Forwarding based** Uses a decentralized cycle elimination protocol		Yes			9
Scalable Bipartite Overlay	Decentralized unstructured	**Overlay optimization based** Uses bipartite partition graph and builds local minimum spanning trees	Gnutella	Yes		Better than Gnutella	77
Adaptive Connection Establishment	Decentralized unstructured	**Overlay optimization based** Forms Neighbour Cost Tables, builds local minimum spanning trees and perform local optimizations	Adaptive Overlay Topology Optimization (AOTO), Gnutella	Yes		Better than Gnutella	110
Hops Adaptive Neighbour Discovery							1

Continued on next page

Table I – *Continued from previous page*

Algorithm	Arch.	Overlay structure	Base protocol	Dynamic update	Runtime	Scalability	cites
Two-Hop-Away Neighbour Comparison and Selection (THANCS)	Decentralized unstructured	**Overlay optimization based** Uses piggybacking to discover neighbour distances and selects neighbours	Gnutella	Yes			36
mOverlay	Decentralized unstructured	**Overlay optimization based** Uses dynamic landmarks to find node locality		Yes		Due to dynamic landmarks and grouping, more scalable than tree-based or mesh-based protocols	123
Distributed Domain Name Order (DDNO)	Decentralized unstructured	**Overlay optimization based** Connects half of the nodes connections to the nodes in the same domain and the other half to random nodes, therefore supports locality and topological connection		Yes		Yes, by using super peers	5
Peer-exchange Routing Optimization Protocols	Works with both decentralized structured and unstructured	**Overlay optimization based** Optimizes overlay by the exchange of neighbors among peers		Yes		Yes	13

Continued on next page

Table I – *Continued from previous page*

Algorithm	Arch.	Overlay structure	Base protocol	Dynamic update	Runtime	Scalability	cites
MAY OMIT-T2MC	Decentralized unstructured	**Cluster based— IS IT ALSO PROXIMITY BASED?** Uses traceroute results for clustering the nodes					1
Unnamed-unstructured	Decentralized unstructured	**Overlay optimization based** Minimizes the communication delay and maximizes the broadcasting range		Yes		Better than THANCS and mOverlay	7
Landmark Binning	Can work with both decentralized structured and unstructured	**Proximity based, landmark binning** Uses network latency to partition nodes into bins		Yes			817

4.5 Discussion on Unstructured Decentralized Algorithms

Earlier in section 4 we presented the state of the art unstructured decentralized P2P algorithms in four different categories based on their architecture; broadcast based, cache based, overlay optimization based, and landmark based. In this section, we present a final discussion of all the methods including the advantages, disadvantages, and novelties presented in order to tackle the topology mismatch problem. Table I presents an overview of all the unstructured decentralized algorithms described earlier in this section.

The broadcast based approaches in general propose intelligent neighbour selection algorithms in order to replace the inefficient blind flooding algorithm of Gnutella. Instead of forwarding the queries to all the neighbours, intelligently selecting neighbours with high probability of answering the query, or neighbours that have some specific features such as high bandwidth, high capacity, or low latency in practice reduces the aggregate resource usage of the P2P network and improves the overall performance of the system. However, since the broadcast based algorithms proposed do not consider the underlying physical topology when selecting the neighbours, or optimizing the overlay structure, the algorithms do not offer a solution to the topology mismatch problem. Although in practice the algorithms work, none can guarantee that a single link is not used by more than necessary, or the created topology resembles the underlying physical topology. Broadcast based methods can be used in conjunction with other approaches to improve the quality of the P2P systems.

The success of caching, as a well studied method in the client server internet applications, also made it a good candidate for the peer-to-peer networks. Even though caching improves the performance, and reduces the overall resource usage of P2P systems, the design of caches is non trivial compared to the web caching systems. Due to the unique structure of the P2P overlays, each node being a server and a client simultaneously, two important problems has to be solved when designing caches. First, the lifetime of a query is short, as the nodes join and leave frequently. Second, the result of a single query string is not always the same, and changes based on the TTL value, source of the query, the lifetime of the query, etc. So, in order to develop a successful caching system in P2P systems, these parameters also have to be considered. Even though the state of the art P2P algorithms using caching methods reduce the resource usage of the network, since the physical network topology is not considered by any algorithm, the proposed systems do not provide direct ways to overcome the topology mismatch problem caused by the overlay network.

The overlay optimization based algorithms can be analyzed in two main topics, the minimum spanning tree based ones, and the clustering based ones. Even though the standardization of the IP multicast protocol is completed a while ago, and some vendors deployed routers supporting the protocol, in practice the standard is not widely embraced as predicted, mostly because the protocol violates the stateless design of the internet routers. The research community, as an alternative proposed application layer multicast frameworks, such as Narada, to simulate the same multicast protocol without requiring IP multicast services. The application layer multicast, however, is not as efficient as IP multicast, and suffers from the

topology mismatch problem. Narada, and its followers (AOTO, LTM, SBO) try to solve the topology mismatch problem by building a richer connected graph and forming minimum spanning trees over this graph that can efficiently route messages among peers. The application level multicast protocol, initiated by Narada, is a generic protocol that can be applied to P2P file sharing, as well as content distribution networks. The original Narada algorithm is designed for small groups therefore it is not scalable. The followers try to solve the scalability problem by introducing various methods including forming minimum spanning trees for each nodes' N^2 neighbours, partitioning the graph into two random groups where each group is responsible of different tasks, or performing local optimizations dynamically on the overlay graph. The advantage of the minimum spanning tree based approaches is that they maintain the connectivity on the network in an efficient way, while still not shrinking the search space. However, building and maintaining a minimum spanning tree creates a huge overhead for the network. One other popular approach used in overlay optimization is the cluster based approaches. In cluster based approach, nodes use various methods to detect physically close nodes to form clusters. T2MC uses traceroute logs and DDNO uses domain names to cluster close by nodes. However, the accuracy of the methods used to form clusters directly affect the success rate of the algorithm. Traceroute for example is a heavy weight protocol to frequently use on the network, and most of the network vendors for this reason do not allow traceroute calls. The major problem with cluster based approaches is that the limited connectivity within the local domains shrink the search scope dramatically, which negatively affects the search performance of the P2P system. DDNO addresses this limitation by allowing half of each nodes connections to be to random nodes over the network, which balances the efficiency of the clustering approach with improved connectivity.

In landmark based proximity method, nodes use network delay (round trip time) as a distance measurement method to position themselves based on distance measurements to apriori known servers on the internet. The servers behave as landmark points, and nodes use an estimation method to discover their positions. The landmark servers are either used by nodes to directly calculate their positions, or indirectly used to cluster nodes into groups or bins based on their estimated distances to well known landmark servers, due to the assumption that nodes with similar distances to a set of landmarks are physically close to each other over the network. The landmark based protocol has two important drawbacks. Firstly the network delay is not a reliable distance estimation method. For example, based on the load on the network the delay to certain nodes or networks can change from time to time, which will eventually affect the distance measurements and wrong measurements will lead to wrong estimated positions for the nodes, or incorrect and non optimal clusterings of the nodes. The second drawback of using landmark servers is the cost of installing and maintaining many landmark servers over the internet for various AS domains. As popular P2P file sharing applications usually have millions of peers connected at any time, it is not false to assume that the hardware and the network costs of maintaining these landmark servers will be quite high. A possible solution to the scalability problem of the static landmark servers is to use ordinary nodes as dynamic landmarks once they estimate their own positions. Even though this

approach scales much better than static landmark servers, still the measurement accuracy problem affects the overall performance of the system.

5. STRUCTURED DECENTRALIZED ALGORITHMS

In this section, the algorithms proposed for the structured decentralized architectures are analysed and categorized based on their use of the proximity information to optimize the underlying structure. In structured P2P algorithms, the construction of the routing tables among nodes determines the efficiency of the algorithms. Therefore, routing tables that represent the underlying physical structure well can achieve much better performance. For this reason, the proposed methods for topology mismatch problem generally use different levels of proximity information to optimize the routing table close to the physical network. We analyze the structured decentralized algorithms based on the categories presented in [Castro et al. 2002; Castro et al. 2002a; Ratnasamy et al. 2002]. In this section, the details of the methodologies used and the algorithms for each methodology are discussed in details.

5.1 Geographic Layout

Geographic layout is used as a method to optimize the routing table of the structured algorithms to represent the physical network topology as close as possible. In this method, physically close nodes are detected, and they are positioned closely in the overlay structure. In order for this method to work, there should be means to detect physical proximity of internet nodes. One popular method is to use the well known internet servers, or servers dedicated for this sole purpose, as landmark nodes, similarly as discussed previously in Section 4.4 (*Landmark Based Proximity*). The inaccuracy in the positioning using landmarks, and the cost of deploying and maintaining these landmarks are the major disadvantages of these systems. Even though managing the overlay structure based on the geographic layout of the nodes improves the query efficiency of the system, on the other hand, it tends to create hotspots, and the needed failure resilience is undermined by the fact that close by nodes are more likely to suffer collective failures.

Global Soft-State. *Global Soft-State* [Xu et al. 2003] builds a global map to help choose shorter routing paths, combining the landmark binning method and small scale distance probes to reveal the proximity properties of the underlying network to the overlay. This global view of the state is made available to all nodes in order to help them find the best way to route their messages. *Global Soft-State* operates in two main stages, generation and using of the proximity information. For generating the proximity information a hybrid approach is proposed, which uses landmark clustering as a preprocessing step in order to select a number of potential nearest neighbour candidates and then refine the selection by incorporating an RTT scheme to ultimately choose the closest node. For using the proximity information, the algorithm chooses a different path from the classic gossiping approaches for constructing and maintaining the overlay. It is based on landmark clustering based strategic placement of proximity information on the overlay enabling any node to access such information using a landmark number that reflects its physical position

in the network. For various logical regions[2] maps of physical information are built and published where each node may appear in a maximum of $log\,(N)$ such maps. To dynamically adapt to changing network conditions, a node subscribes to relevant *soft states* that utilize a notification system in order to initialize any necessary neighbour re-selection.

Maintaining several host states at different layers, makes any content migration costly. Additionally, the method does not make any continuing effort to remap the overlay structure after a node successfully joins, in order to adapt its state to any occurrence of condition change. Although this approach greatly reduces the routing latency to far nodes, it is unable to dynamically identify nodes that are close to routers and gateways in order to construct the secondary overlay. Nevertheless, static recognition of such nodes is currently done based on BGP reports and pre-chosen landmarks, sacrificing the self-organising attribute of traditional DHTs.

Mithos. *Mithos* [Waldvogel and Rinaldi 2002] is a P2P protocol which incorporates a directed incremental probing to find near optimal node placement, and classified by its authors as an integration of geographic layout and proximity routing overlay optimization methods.

The bootstrap phase of *Mithos* starts with a subset of existing members as the first set of candidates, and while the iteratively closest nodes are detected by probing the neighborhood, the candidate neighbor list is updated. In order to avoid a local minima *Mithos* probes all the neighbours within a two hop distance from the current minimum before concluding the process.

After finding its first neighbour, the newcomming peer is assigned an ID using information gathered during the iterative neighbour selection phase. Virtual coordinates are assigned to the newcomming peer by using the distances of two closest nodes and their neighbours, so that Euclidean distances between the node and all known hosts predict the network latency between them[Cox et al. 2004]. The benefit of these synthetic coordinates are that they are explicitly used as the node's ID, and distance computations among nodes can be done in ID space without requiring physical measurements.

The last step of the algorithm is the interconnection among neighbours. *Mithos* uses a *quadrant*-based mechanism according to which each node establishes a link to the closest neighbour in each quadrant. During forwarding, the next hop is performed towards a neighbour in the same quadrant as the final destination. The newcoming node may not know of other neighbours in all quadrants, therefore, the node first identifies neighbours in all quadrants using a mechanism based on ideas similar to a perimeter walk[3] and then improves the results using parallel path processing by taking into account further geometric properties of node relationships.

One major limitation of *Mithos* is that the protocol cannot handle dynamic arrivals and removals from the network, as is happens in mobile networks.

LAPTOP. *LAPTOP* [Wu et al. 2007], organises the overlay into a tree based hierarchy with main focus on reducing hops during message routing as well as min-

[2]This might be a high-order zone in the eCAN[Xu and Zhang 2002] context or a set of nodes sharing a particular prefix in overlays such as Pastry.
[3]Used in Greedy Perimeter Stateless Routing (GPSR) protocol.

imizing maintenance overhead. Additionally, a caching scheme is also incorporated so as to further reduce routing table update costs. The authors theoretically show that in *LAPTOP* routing path length is bounded by $O(log_d N)$ and node joining and leaving in the overlay network is bounded by $O(dlog_d N)$ hops in a balanced overlay tree, where N is the number of nodes, and d is the maximum degree of each node. *LAPTOP* implements the geographical layout approach and constructs a geographical layout in a self-organizing and efficient fashion, by estimating the round trip time (RTT) to a small number of nodes in the overlay network in order to make them roughly aware of their physical distances among them.

Each node is assigned an address in a dotted format (e.g 1.3.4). Each octet ranges from 1 to d, where d is the maximum degree of the nodes. The assignment process is done by appending a unique octet to the address of each nodes parent, while root node is assigned address 1. The routing scheme is similar to the longest-prefix IP matching scheme. At each forwarding hop, any message travels up the tree until the first common ancestor of source and destination node is reached and then starts descending to arrive to its target. During tree traversals, special entries in the routing tables, called *routing cache*, are maintained in order to increase routing efficiency and achieve finer load balance. Caching enables a node to forward a message to a better longest-prefix match than that of its direct ancestor making a large, quicker and more cost effective step through the overlay and toward the destination. To improve scalability, the number of children nodes and the size of the routing cache are limited. In terms of overlay maintenance, *LAPTOP* incorporates a simple *heartbeat*-based technique where each parent node is responsible for monitoring its children. At join process the newcomming node is assigned its level label as well as its address by its parent node. Additionally it initializes its routing table (with normal and caching entries) as it traverses the overlay in search for its parents node. During a graceful departure, the referred node checks for children in the overlay. If it does not have any, it simply notifies its parent and leaves. If it has, it selects the child node with the lowest RTT to it in order to take its place so that the locality property is preserved.

5.2 Proximity Routing

Proximity routing does not require routing tables to be built using any knowledge about network proximity. On the other hand it exploits such knowledge in order to choose the best next hop during routing a message. This approach balances between choosing the node that will further progress the routing towards the destination and choosing the closest entry in the routing table, in terms of network proximity. Thus, it is relatively less effective than geographical layout when applied to CAN(-like) implementations. Moreover, the technique has been incorporated into a version of Chord causing an increase on the overhead of node joins and the size as well as maintenance cost of finger tables. The methods that use proximity routing and their details are listed below.

Proximity in Kademlia. Before discussing *Proximity in Kademlia*, it would be helpful to briefly summarize the highlights about the *Kademlia* protocol. *Kademlia* [Maymounkov and Mazières 2002] is a distributed hash table (DHT) based protocol for P2P networks, which uses an iteration based lookup algorithm. The protocol

uses the standard 160 bit ID system for nodes and locates the nodes in a prefix binary tree, where IDs are used as prefixes. The iterative lookup operations are done over this prefix binary tree, which converges to logarithmic lookup times. The ID's are assigned randomly, therefore in Kademlia, there is no proximity control, which results in inefficient use of the underlying network during lookup and retrieve operations.

Proximity in Kademlia [Kaune et al. 2008] introduces proximity controls over the base Kademlia protocol to optimize the underlying network usage of the protocol. The authors define an abstract *underlay metric* that calculates the suitability of establishing a communication link between peers as a cost function, based on the used proximity criteria (RTT, ISP locality, etc.). *Proximity in Kademlia* adapted both *proximity routing* and *proximity neighbour selection* overlay optimization algorithms for controlling proximity. Therefore, we categorize the protocol only in this section, and skip the discussion in the further *proximity neighbour selection* section (Sec.5.3). *Proximity in Kademlia* also uses the MaxMind GeoIP database[4] to detect the proximity information of a given peer based on its region, country, or ISP location, and use this information to form clusters of nearby peers. One other method *Proximity in Kademlia* uses is the Vivaldi [Cox et al. 2004] protocol. However, authors report that clustering approach performs better than Vivaldi protocol based on their experiments. Authors also report that *Proximity routing* protocol successfully worked with the Kademlia protocol and improved the locality of the connections over peers.

CHOord considering Proximity on IPv6 (CHOP6). *CHOP6* [Morimoto and Teraoka 2007] is designed based on the *Chord* protocol. *CHOP6* roughly estimates the proximity among nodes by exploiting the IPv6 address format and RTT information if available. The proximity estimation is achieved by introducing a 64-bit ID scheme in which the least significant bit part is the IPv6 global routing prefix and thus enabling a longest prefix match scheme. The protocol is designed based on the observation that it is possible to estimate a node's geographical location by simply examining the upper 32-bits of its IPv6 address. Moreover, similar to *Chord*, *CHOP6* uses a finger table, whose entries hold more than one candidate node.

Chord6. *Chord6* [Xiong et al. 2005] is another *Chord* variant that tries to exploit the hierarchical features of IPv6 in order to create a substrate that reduces inter-domain traffic between service providers. *Chord6* is based on the original Chord protocol, and the main difference from the original protocol is in the identifier definition. Therefore, the approach can be easily portable to other DHTs such as CAN, Pastry and Tapestry. In Chord6 the identifier contains two parts: the higher bits are obtained by hashing the node's IPv6 address prefix of specific length, while the remaining lower bits are the hash value of the rest of that IPv6 address. As a result of this assignment, nodes in a domain will be mapped onto a continuous key space on the overlay network, which avoids unnecessary message forwarding across different service providers, thus minimizing overall routing cost.

[4]MaxMind Geolocation Technology. http://www.maxmind.com

5.3 Proximity Neighbour Selection

Proximity Neighbour Selection constructs the routing tables using proximity knowledge. The proximity information used in this method is different than the landmark based systems described in the *Geographic Layout* section, as TTL values between nodes, or directly node ID prefixes are used to detect proximity. Tapestry, Pastry, and CAN successfully implemented the proximity into their algorithms by using this approach. The routing protocol in Pastry is based on longest node ID prefix matching, while CAN uses RTT values to detect close by nodes. [Castro et al. 2002b] reports that proximity neighbour selection as an effective proximity based method. Proximity Neighbor Selection based algorithms are described in detail below.

DHT-PNS. Chord-DHT-PNS [Duan et al. 2006] implements proximity neighbour selection on top of the Chord DHT. The main purpose is to use proximity information to group physically close by nodes as neighbours in the DHT table. In order to detect proximity, virtual network coordinates of peers are used, by using the Vivaldi protocol [Cox et al. 2004]. The virtual coordinates are then used by the nodes to map to identifier space in the DHT. The space is partitioned using a *concentric circle clustering scheme* where successive cycles of radiuses ρ, 2ρ, 3ρ and so on, are constructed. Then the formed annuluses are divided into $2\chi - 1$ *sectors*, where χ denotes the sequence number of the annulus starting from $\chi = 1$ for the centre cycle. It is proved in the paper, that this way each sector occupies the same area as does the center cycle. Assuming uniform node distribution, this characteristic, favours a more load balanced clustering operation. Every sector in this $2D$ coordinate space is mapped to a unique *region* in the DHT space forming a multi-layer node identifier space. Thus, any nodes that belong to the same sector, are mapped to the same region as well, preserving their proximity relationship unveiled by the use of the Vivaldi protocol. The individual pieces are mapped to the identifier space uniquely, allowing logarithmic lookup operations with high probability on Chord.

IP-based Clustering (IPBC). Proximity neighbour selection algorithms use probing and other measures to detect proximity. However, such methods are either are not precise enough, or creates overload in the network. *IP-based clustering* [Karwaczynski and Mocnik 2007] is a proximity neighbour selection based algorithm in which the authors propose to use the IP address prefixes (16 bit for IPv4) in order to detect proximity. [Freedman et al. 2005] states that 97% of prefixes larger than 24 bits belong to a single geographical location. However, using less number of bits creates less precise results and more number of bits increase the burden on the network and reduce the possible number of neighbours. Therefore, a careful selection is required in terms of performance/accuracy tradeoff. The *IP-based Clustering* generates a key and its prefix and stores its prefix in the DHT itself, so that any newly joining nodes with the same IP prefix can query the prefix and identify all the neighbours with the same prefix relatively easily. Nodes periodically update their entries in the DHT and remove their entry when they leave, or the entry is timed out with no further activity detected from the node.

Cone. *Cone*[Huijin and Yongting 2007] extends the Chord using proximity neighbour selection topology optimization algorithm. The proximity information is generated using landmarks and RTT based distances to landmarks. *Cone* uses a two-layered identifier space. The first, named Chord-layer identifier, denoted as Id_{Chord}, is the same as in Chord. The second is the Cone-layer identifier, Id_{Cone} which is constructed by two component identifiers. The first, known as *group identifier (gid)* denotes a relevant group the node belongs to while the second, namely *local identifier (lid)* indicates the local identifier within the group. The group concept, which is introduced here, is a way of dividing nodes according to a common Id_{Chord} prefix. The structure of a Cone overlay, retains the Chord's circular topology. The difference lies on the fact that, now, two rings are created. A big ring, where nodes with the same *gid* are arranged at each position. Each of these positions are a smaller ring for the particular group's *lid*s. The routing is achieved in both clockwise and counter-clockwise directions in the big-ring, for which two routing tables are maintained, namely *front* and *back* finger tables. Entries in these tables, display physical network proximity with the current node. Moreover, a third table called *group* table maintains information about other online peers within the current node's group in a way that entries are now close in the ID space.

SAT-Match: Self-Adaptive Topology Matching. *SAT-Match* [Ren et al. 2004] is a protocol that specifically tries to solve the topology mismatch problem by mapping the overlay network as close as possible to the physical one. Even though no theoretical proof is given, valid results are obtained and demonstrated through experimentations. Similar to the other approaches in the proximity neighbour selection, *SAT-Match* uses probing to detect close by peers and obtain proximity information. However, as an additional feature, *SAT-Match* peers can do selective jumps to adjust their locations in the DHT, if it reduces the stretch of its one-hop neighbourhood. The stretch is defined as the ratio between the average logical and physical latency of a link. It is reported that, due to the selective jumps, the *SAT-Match* achieves 40% reduction in link stretch, and when used with the *Landmark Binning* approach (see Sec. 4.4), the reduction rate increases up to 60%. For dynamic environments, with frequent node arrival and leave, *SAT-Match* scales much better than *Mithos*, due to its self adaptation mechanism and selective jumps.

SAT-Match uses a small TTL value for the probing messages in order to reduce redundancy[Jiang et al. 2008]. This process begins as soon as the node joins the network using a DHT mechanism. Each probing message contains information about the source and a small TTL value. The recipient of such a message returns information about itself to the source and forwards the probing message to its neighbours if the TTL is non-zero. The discovered nodes are referred to as $TTL-k$ neighbourhood of the source node based on the TTL distance to the source node.

Blindly selecting the peer with the smallest RTT as neighbour is, generally, not the optimal decision to make in order to achieve global *stretch* reduction. In a structured scheme, when a node jumps to connect to a physically close node, it may need to connect to other distant nodes to maintain the structure's integrity, thus creating an overall increase in the overlay's *stretch*. The two nodes with the smallest RTT is then used in order to select one zone to jump in this phase. The algorithm is as follows: The source node S calculates the stretch change of its

$TTL-1$ neighbourhood and that of the $TTL-1$ neighbourhood of the first of the previously selected peers. These calculations are made as if the jump has been made. If the stretch reduction is over a predefined threshold the jump is performed, otherwise the second selected candidate is picked and the same computations are performed. If again, the threshold is not met, then no jump is ultimately done. In case of a jump, this is performed as a combination of *leave* and *join* operations, in the CAN context.

Table II: Decentralized Structured Algorithms

Algorithm	Arch.	Overlay structure	Base protocol	Dynamic update	Runtime	Scalability	cites
Global Soft-state	decentralized structured	**Proximity based, landmark clustering** Uses first landmark clustering then measures RTTs to identify close nodes		Yes			195
Mithos	decentralized structured	**Proximity based** Uses nodes as topology landmarks and directed incremental probing to optimize topology		Yes		Scales well as all operations are local ???	172
Self-Adaptive Topology Matching	decentralized structured	**Proximity based** Uses lightweight probing and selective jumps to optimize the topology	CAN	Yes		Better than Mithos	40
Delay Aware P2P System							1
VERSION OF CHORD - DHT-PNS	decentralized structured	**Proximity based** Uses Proximity Neighbour Selection and the Vivaldi system	Chord	Yes			5
MAY OMIT-VERSION OF CHORD - Quasi-Chord	decentralized structured	**Proximity based**	Chord	Yes			0

Continued on next page

Table II – *Continued from previous page*

Algorithm	Arch.	Overlay structure	Base protocol	Dynamic update	Runtime	Scalability	cites
LAPTOP	decentralized structured	**Geographic layout based** Hierarchical overlay structure		Yes		routing path length $\log_d N$, join/leave overhead $d \log_d N$	4
IP-Based Clustering	decentralized structured	**Proximity based** Proximity neighbour selection based on longest common prefix of IP addresses		Yes			1
CHOord considering Proximity on IPv6	decentralized structured	**Proximity based** Uses IPv6 address format to provide proximity	Chord	Yes		Better than Chord	1
Proximity in Kademlia	decentralized structured	**Proximity based** Applies proximity neighbour selection (PNS) and proximity route selection (PRS) to Kademlia	Kademlia	Yes			13
Cone	decentralized structured	**Proximity based** Uses proximity neighbour selection (PNS)	Chord	Yes		Better than Chord	3
Dynamo							3
MAY OMIT-BADLY WRITTEN-PChord	decentralized structured	**Proximity based**	Chord	Yes			16
AChord							7

Continued on next page

Table II – *Continued from previous page*

Algorithm	Arch.	Overlay structure	Base protocol	Dynamic update	Runtime	Scalability	cites
Chord6	decentralized structured	**Proximity based** Uses IPv6 hierarchical address format to cluster topologically close nodes	Chord	Yes			14

5.4 Discussion on Structured Decentralized Algorithms

In section 5, we presented the state of the art structured decentralized P2P algorithms in three different categories based on their handling of the network structure to tackle the topology mismatch problem: geographic layout based, proximity routing based, and proximity neighbour selection based. In this section, we present a final discussion of all the methods including the advantages, disadvantages, and novelties related to solve the topology mismatch problem. Table II presents an overview of all the structured decentralized algorithms described earlier in this section.

Structured P2P network algorithms use a global distributed hash table or a prefix tree structure to uniquely lookup peers or their data in the overlay network. As all the data is kept within the overlay, each node behaves as a client and a server, therefore nodes join and leave according to rules determined by the integrity of the global data structure. The main advantage of the structured P2P topology is that by the help of the global data structure, peers or their data can be found within the network even if there is only a single copy of that item present. However, each node join and leave creates maintenance overhead for the network due to updates required by the global data structure, and for networks with frequent node arrivals and departures the topology uses valuable network resources just to update the global structure. Nodes join the network by using a key value, which determines the location and the neighbourhood of the new node within the network. However, assigning random key values to the newly inserted nodes creates non-optimal matching with the underlying physical network topology, therefore, increasing the overhead of the network even more. One solution for handling the topology mismatch problem is to consider the proximity of the peers when generating the key and joining the node to the network, so that nodes within the same network domains are selected as peers, or neighbours, during the overlay topology construction. In this chapter, we have described three such approaches to optimize the topology matching problem: geographic layout, proximity routing and proximity neighbour selection.

In geographic layout, nodes try to estimate the geographic positions of the peers, and construct the DHT considering the proximity of the peers. Landmark servers and RTT measurements are two popular methods, which can also be used in conjunction, to discover physically close by peers over the network. However, as discussed in Section 4.4, these methods do not always give reliable estimates for the node positions over the internet. The landmark servers are not self-organizing and have maintenance overheads. To serve a P2P network with millions of peers, multiple landmark servers distributed over the whole world is required, which is hard to manage. The RTT measurements also can measure the delay between peers, but it is a greedy method, which can result non-optimal overlay topologies especially if close by nodes have low bandwidth connections among themselves.

Proximity routing does not discover or store proximity information for the peers, however, during package forwarding, nodes that have lower latency to the destination, or with a closer key to the destination is selected. As no proximity information is used, the system is based on a greedy approach and it usually selects low latency paths over the overlay, which maps to suboptimal longer paths on the physical

topology. For these obvious reasons, proximity neighbour selection is generally considered as a superior method to proximity routing, however, joint uses of these two protocols are also possible.

Proximity neighbour selection approach is similar to proximity routing, with an exception that during forwarding, the proximity information of the peers are also considered. Therefore, for each application, proximity information is extracted and used during routing. Depending on the application, the proximity information can be the node sharing the same prefix with the current node, or TTL to detect close by neighbours. Tapstry and Pastry are two popular methods implementing the proximity neighbour selection among others mentioned above in the proximity neighbour selection section.

6. CONCLUSIONS

Peer-to-peer architecture has been in the center of research attention in the last decade. Especially the decentralized unstructured genre exploits the advantages of loose coupling and self organization of computing nodes to form application-layer networks on top of the physical, best-effort infrastructure of the Internet that exhibit interesting properties. Scalability problems arouse quickly, though, because of the inefficient construction of this overlay that was built with no concern for the underlying physical network that causes a great deal of redundant traffic. The problem was identified by the research community as the topology mismatch problem between the overlay and the corresponding underlying physical network and a great deal of effort has been set towards alleviating it. Some fruits of this effort have been gathered and presented in this survey. Measurement of link cost through latency or RTT and deletion of inefficient established connections are the key concepts of almost all approaches. Others, manage to address the problem through hierarchical peer clustering (e.g. best IP matching). Additionally, some protocols focus on specific problems that furtherly arise, such as overlay partitioning, search scope reduction or convergence speed, to name just a few. Another desire is to form a protocol that could be applied to both decentralized unstructured and decentralized structured networks. Unfortunately no approach has equally addressed these problems in order to form a robust solution. So the field seems to be, still, fertile for any, new, clever idea.

REFERENCES

Andersen, D., Balakrishnan, H., Kaashoek, F., and Morris, R. 2001. Resilient overlay networks. In *SOSP '01: Proceedings of the eighteenth ACM symposium on Operating systems principles*. ACM, New York, NY, USA, 131–145.

Androutsellis-Theotokis, S. and Spinellis, D. 2004. A survey of peer-to-peer content distribution technologies. *ACM Comput. Surv. 36,* 4, 335–371.

Castro, M., Druschel, P., Charlie, Y., and Rowstron, A. 2002a. Topology-aware routing in structured peer-to-peer overlay networks. Tech. rep., Microsoft Research, Redmond, WA 98052.

Castro, M., Druschel, P., Charlie, Y., and Rowstron, H. A. 2002b. Exploiting network proximity in peer-to-peer overlay networks. Tech. rep., Microsoft Research.

Castro, M., Druschel, P., Hu, Y. C., and Rowstron, A. 2002. Exploiting network proximity in distributed hash tables. In *in International Workshop on Future Directions in Distributed Computing (FuDiCo*. 52–55.

Chawathe, Y. 2000. An architecture for internet content distribution as an infrastructure service. Ph.D. thesis, University of California at Berkeley.

Chawathe, Y., Ratnasamy, S., Breslau, L., Lanham, N., and Shenker, S. 2003. Making gnutella-like p2p systems scalable. In *SIGCOMM '03: Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*. ACM, New York, NY, USA, 407–418.

Chu, Y., Rao, S., Seshan, S., and Zhang, H. 2001. Enabling conferencing applications on the internet using an overlay muilticast architecture. *SIGCOMM Comput. Commun. Rev. 31,* 4, 55–67.

Chu, Y., Rao, S. G., Seshan, S., and Zhang, H. 2002. A case for end system multicast. *IEEE Journal on Selected Areas in Communications 20,* 8 (Oct.), 1456–1471.

Chu, Y., Rao, S. G., and Zhang, H. 2000. A case for end system multicast (keynote address). In *SIGMETRICS '00: Proceedings of the 2000 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*. ACM, New York, NY, USA, 1–12.

Clarke, I., Sandberg, O., Wiley, B., and Hong, T. W. 2001. Freenet: a distributed anonymous information storage and retrieval system. In *International workshop on Designing privacy enhancing technologies*. Springer-Verlag New York, Inc., New York, NY, USA, 46–66.

Cohen, E. and Shenker, S. 2002. Replication strategies in unstructured peer-to-peer networks. In *Proceedings of the ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*. Pittsburgh, PA, USA, 177–190.

Cox, R., Dabek, F., Kaashoek, F., Li, J., and Morris, R. 2004. Practical, distributed network coordinates. *SIGCOMM Comput. Commun. Rev. 34,* 1, 113–118.

Dabek, F., Kaashoek, M. F., Karger, D., Morris, R., and Stoica, I. 2001. Wide-area cooperative storage with cfs. In *SOSP '01: Proceedings of the eighteenth ACM symposium on Operating systems principles*. ACM, New York, NY, USA, 202–215.

Duan, H., Lu, X., Tang, H., Zhou, X., and Zhao, Z. 2006. Proximity neighbor selection in structured p2p network. In *CIT '06: Proceedings of the Sixth IEEE International Conference on Computer and Information Technology*. IEEE Computer Society, Washington, DC, USA, 52.

Eisner, J. 2005. Diffserv - the scalable end-to-end quality of service model. White paper, Cisco Systems. Available online (19 pages).

Freedman, M. J., Vutukuru, M., Feamster, N., and Balakrishnan, H. 2005. Geographic locality of ip prefixes. In *IMC '05: Proceedings of the 5th ACM SIGCOMM conference on Internet Measurement*. USENIX Association, Berkeley, CA, USA, 13–13.

Garey, M. R. and Johnson, D. S. 1979. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman.

Gnutella. http://rfc-gnutella.sourceforge.net/.

Hong, F., Li, M., Yu, J., and Wang, Y. 2005. Pchord: Improvement on chord to achieve better routing efficiency by exploiting proximity. In *ICDCSW '05: Proceedings of the First International Workshop on Mobility in Peer-to-Peer Systems*. IEEE Computer Society, Washington, DC, USA, 806–811.

Hsiao, H.-C., Liao, H., and Huang, C.-C. 2009. Resolving the topology mismatch problem in unstructured peer-to-peer networks. *IEEE Transactions on Parallel and Distributed Systems 99,* 1.

Huijin, W. and Yongting, L. 2007. Cone: A topology-aware structured p2p system with proximity neighbor selection. In *FGCN '07: Proceedings of the Future Generation Communication and Networking*. IEEE Computer Society, Washington, DC, USA, 43–49.

Jannotti, J., Gifford, D. K., Johnson, K. L., Kaashoek, M. F., and James W. O'Toole, J. 2000. Overcast: Reliable multicasting with on overlay network. In *OSDI'00: Proceedings of the 4th Conference on Symposium on Operating System Design & Implementation*. USENIX Association, Berkeley, CA, USA, 14–14.

Jiang, S., Guo, L., Zhang, X., and Wang, H. 2008. Lightflood: Minimizing redundant messages and maximizing scope of peer-to-peer search. *IEEE Trans. Parallel Distrib. Syst. 19,* 5, 601–614.

Karwaczynski, P. and Mocnik, J. 2007. Ip-based clustering for peer-to-peer overlays. *JSW 2,* 2, 30–37.

Kaune, S., Lauinger, T., Kovacevic, A., and Pussep, K. 2008. Embracing the peer next door: Proximity in kademlia. In *Peer-to-Peer Computing*. 343–350.

Kazaa. http://www.kazaa.com/.

Kwon, M. and Fahmy, S. 2002. Topology-aware overlay networks for group communication. In *NOSSDAV '02: Proceedings of the 12th international workshop on Network and operating systems support for digital audio and video*. ACM, New York, NY, USA, 127–136.

Liu, Y. 2008. A two-hop solution to solving topology mismatch. *Parallel and Distributed Systems, IEEE Transactions on 19,* 11 (Nov.), 1591–1600.

Liu, Y., Liu, X., Xiao, L., Ni, L., and Zhang, X. 2004. Location-aware topology matching in p2p systems. *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies 4*, 2220–2230 vol.4.

Liu, Y., Xiao, L., and Ni, L. 2004. Building a scalable bipartite p2p overlay network. In *Proceedings of the 18th International Parallel and Distributed Processing Symposium*. Santa Fe, New Mexico, USA, 46.

Liu, Y., Xiao, L., and Ni, L. 2007. Building a scalable bipartite p2p overlay network. *IEEE Trans. Parallel Distrib. Syst. 18,* 9, 1296–1306.

Liu, Y., Zhuang, Z., Xiao, L., and Ni, L. 2003. Aoto: adaptive overlay topology optimization in unstructured p2p systems. *Global Telecommunications Conference, 2003. GLOBECOM '03. IEEE 7*, 4186–4190 vol.7.

Liu, Y., Zhuang, Z., Xiao, L., and Ni, L. M. 2004. A distributed approach to solving overlay mismatching problem. In *ICDCS '04: Proceedings of the 24th International Conference on Distributed Computing Systems (ICDCS'04)*. IEEE Computer Society, Washington, DC, USA, 132–139.

Lua, E. K., Crowcroft, J., Pias, M., Sharma, R., and Lim, S. 2005. A survey and comparison of peer-to-peer overlay network schemes. *IEEE Communications Surveys and Tutorials 7,* 1-4, 72–93.

Lv, Q., Cao, P., Cohen, E., Li, K., and Shenker, S. 2002. Search and replication in unstructured peer-to-peer networks. In *ICS '02: Proceedings of the 16th international conference on Supercomputing*. ACM, New York, NY, USA, 84–95.

Markatos, E. P. 2002. Tracing a large-scale peer to peer system: An hour in the life of gnutella. In *2nd IEEE International Symposium on Cluster Computing and the Grid (CCGrid 2002)*. Berlin, Germany, 65–74.

Maymounkov, P. and Mazières, D. 2002. Kademlia: A peer-to-peer information system based on the xor metric. In *IPTPS '01: Revised Papers from the First International Workshop on Peer-to-Peer Systems*. Springer-Verlag, London, UK, 53–65.

Morimoto, S. and Teraoka, F. 2007. Chop6: A dht routing mechanism considering proximity. In *SAINT-W '07: Proceedings of the 2007 International Symposium on Applications and the Internet Workshops*. IEEE Computer Society, Washington, DC, USA, 59.

Ng, T. S. E. and Zhang, H. 2001. Towards global network positioning. In *IMW '01: Proceedings of the 1st ACM SIGCOMM Workshop on Internet Measurement*. ACM, New York, NY, USA, 25–29.

Qiu, T., Chen, G., Ye, M., Chan, E., and Zhao, B. Y. 2007. Towards location-aware topology in both unstructured and structured p2p systems. In *ICPP '07: Proceedings of the 2007 International Conference on Parallel Processing*. IEEE Computer Society, Washington, DC, USA, 30.

Ratnasamy, S., Francis, P., Handley, M., Karp, R., and Schenker, S. 2001. A scalable content-addressable network. In *SIGCOMM '01: Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications*. ACM, New York, NY, USA, 161–172.

Ratnasamy, S., Handley, M., Mark, H., Karp, R., and Shenker, S. 2002. Topologically-aware overlay construction and server selection. In *INFOCOM'02 Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies*. Vol. 3. New York, NY, USA, 1190 – 1199.

Ratnasamy, S., Stoica, I., and Shenker, S. 2002. Routing algorithms for dhts: Some open questions. In *IPTPS '01: Revised Papers from the First International Workshop on Peer-to-Peer Systems*. Springer-Verlag, London, UK, 45–52.

Ren, S., Guo, L., Jiang, S., and Zhang, X. 2004. Sat-match: A self-adaptive topology matching method to achieve low lookup latency in structured p2p overlay networks. *Parallel and Distributed Processing Symposium, International 1*, 83a.

Ripeanu, M., Iamnitchi, A., and Foster, I. 2002. Mapping the gnutella network. *Internet Computing, IEEE 6,* 1 (Jan/Feb), 50–57.

Ritter, J. 2001. Why Gnutella can't scale. No, really.

Rowstron, A. I. T. and Druschel, P. 2001. Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems. In *Middleware '01: Proceedings of the IFIP/ACM International Conference on Distributed Systems Platforms Heidelberg*. Springer-Verlag, London, UK, 329–350.

Saroiu, S., Gummadi, K. P., Dunn, R. J., Gribble, S. D., and Levy, H. M. 2002. An analysis of internet content delivery systems. In *OSDI '02: Proceedings of the 5th symposium on Operating systems design and implementation*. ACM, New York, NY, USA, 315–327.

Sen, S. and Wang, J. 2004. Analyzing peer-to-peer traffic across large networks. *Networking, IEEE/ACM Transactions on 12,* 2 (April), 219–232.

Shi, G., Long, Y., Chen, J., Gong, H., and Zhang, H. 2008. T2MC: A Peer-to-Peer Mismatch Reduction Technique by Traceroute and 2-Means Classification Algorithm. In *NETWORKING 2008 Ad Hoc and Sensor Networks, Wireless Networks, Next Generation Internet*. Lecture Notes in Computer Science, vol. 4982. Springer Berlin / Heidelberg, 366–374.

Stoica, I., Morris, R., Karger, D., Kaashoek, M. F., and Balakrishnan, H. 2001. Chord: A scalable peer-to-peer lookup service for internet applications. In *SIGCOMM '01: Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications*. ACM, New York, NY, USA, 149–160.

Stutzbach, D. and Rejaie, R. 2006. Understanding churn in peer-to-peer networks. In *IMC '06: Proceedings of the 6th ACM SIGCOMM conference on Internet measurement*. ACM, New York, NY, USA, 189–202.

Subramanian, L., Stoica, I., Balakrishnan, H., and Katz, R. H. 2004. Overqos: an overlay based architecture for enhancing internet qos. In *NSDI'04: Proceedings of the 1st conference on Symposium on Networked Systems Design and Implementation*. USENIX Association, Berkeley, CA, USA, 6–6.

Sun, M. and Zhang, Z. 2008. Quasi-chord: physical topology aware structured p2p network. In *Proceedings of the 11th Joint Conference on Information Sciences (JCIS-2008)*. Advances in Intelligent Systems Research. Atlantis Press.

Waldvogel, M. and Rinaldi, R. 2002. Efficient topology-aware overlay network. In *Hot Topics in Networks (HotNets-I)*.

Wong, B., Slivkins, A., and Sirer, E. G. 2005. Meridian: a lightweight network location service without virtual coordinates. In *SIGCOMM '05: Proceedings of the 2005 conference on Applications, technologies, architectures, and protocols for computer communications*. ACM, New York, NY, USA, 85–96.

Wu, C.-J., Liu, D.-K., and Hwang, R.-H. 2007. A location-aware peer-to-peer overlay network: Research articles. *Int. J. Commun. Syst. 20,* 1, 83–102.

Xing-feng, L., Bao-ping, Y., and Wan-ming, L. 2008. Overlay multicast network optimization and simulation based on narada protocol. *Advanced Communication Technology, 2008. ICACT 2008. 10th International Conference on 3*, 2215–2220.

Xiong, J., Zhang, Y., Hong, P., and Li, J. 2005. Chord6: Ipv6 based topology-aware chord. In *ICAS-ICNS '05: Proceedings of the Joint International Conference on Autonomic and Autonomous Systems and International Conference on Networking and Services*. IEEE Computer Society, Washington, DC, USA, 4.

Xu, Z., Tang, C., and Zhang, Z. 2003. Building topology-aware overlays using global soft-state. In *ICDCS '03: Proceedings of the 23rd International Conference on Distributed Computing Systems*. IEEE Computer Society, Washington, DC, USA, 500.

Xu, Z. and Zhang, Z. 2002. Building low-maintenance expressways for p2p systems. Tech. Rep. HP Laboratories Palo Alto, HP Laboratories, Internet Systems and Storage Laboratory. Mar.

Yang, B. and Garcia-Molina, H. 2002. Improving search in peer-to-peer networks. In *ICDCS '02: Proceedings of the 22 nd International Conference on Distributed Computing Systems (ICDCS'02)*. Vienna, Austria, 5 – 14.

Zeinalipour-Yazti, D. and Kalogeraki, V. 2006. Structuring topologically aware overlay networks using domain names. *Comput. Netw. 50,* 16, 3064–3082.

Zhang, X. Y., Zhang, Q., Zhang, Z., Song, G., and Zhu, W. 2004. A construction of locality-aware overlay network: moverlay and its performance. *IEEE Journal on Selected Areas in Communications 22,* 1 (Jan.), 18–28.

Zhao, B. Y., Kubiatowicz, J. D., and Joseph, A. D. 2001. Tapestry: An infrastructure for fault-tolerant wide-area location and routing. Tech. rep., Berkeley, CA, USA.

Zheng, H., Lua, E. K., Pias, M., and Griffin, T. G. 2005. *Internet Routing Policies and Round-Trip-Times*. Lecture Notes in Computer Science, vol. Volume 3431/2005. Springer Berlin / Heidelberg, 236–250.

Zhu, Z., Kalnis, P., and Bakiras, S. 2008. Dcmp: A distributed cycle minimization protocol for peer-to-peer networks. *Parallel and Distributed Systems, IEEE Transactions on 19,* 3 (March), 363–377.