INTERNATIONAL JOURNAL OF COMMUNICATION SYSTEMS
Int. J. Commun. Syst. 2010; **00**:1–27
Published online in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/dac

A Survey and Comparison on Overlay-Underlay Mapping Techniques in Peer-to-Peer Overlay Networks

Humaira Ijaz[1]* Michael Welzl[2] and Bushra Jameel[1]

[1]*Department of CS&IT,University of Sargodha, Sargodha, Pakistan*
[2]*Department of Informatics, University of Oslo, Oslo, Norway*

SUMMARY

Peer to Peer (P2P) overlay networks were developed initially for file sharing such as Napster, Gnutella but later they have become popular for content sharing, media streaming, telephony applications etc. Underlay-unawareness in P2P systems can result in sub-optimal peer selection for overlay routing and hence poor performance. In this paper, we present a comprehensive survey of the research work carried out to solve the overlay-underlay mapping problems up till now. The majority of underlay-aware proposals for peer selection focus on finding the shortest overlay routes by selecting nearest nodes according to proximity information. Another class of approaches is based on passive or active probing for provision of underlay information to P2P applications. Some other optimizations propose use of P2P middleware to extract, process and refine underlay information and provide it to P2P overlay applications. Another class of approaches strive to use ISPs or third parties to provide underlay information to P2P overlay applications according to their requirements. We have made a state-of-the-art review and comparison for addressing the overlay-underlay mismatch in terms of their operation, merits, limitations and future directions.

Received . . .

KEY WORDS: Peer-to-Peer; Overlay; Underlay-Unawareness; proximity; use of ISPs; probing; grouping of nodes; Middleware

1. INTRODUCTION

A Peer-to-peer overlay network is a virtual or logical network of overlay nodes connected by virtual or logical links, formed on the top of another physical network that is called underlay. Whereas every virtual or logical link is like a path consisting of one or many physical links of underlay. There is a vast variety of networks that conform with this definition, ranging from the World Wide Web (WWW) and overlays of historical relevance, such as the Multi-cast Backbone (MBone) and the Active Networks testbed "ABone", to Peer-to-Peer (P2P) applications.

What all these systems have in common is presence of a certain (deliberate!) ignorance about the underlay that enable(d) the quick deployment of new technology without the need to change the underlay. This unawareness also allows formation of a completely new network at application layer that is completely under the control of the application it is designed for. Hence, many overlay networks perform their own routing functions, at the application layer called overlay routing, thereby allowing end nodes to choose paths themselves. On the negative side, this underlay unawareness often causes a topology mismatch [1, 2, 3] as shown in Figure 1. There is an overlay network of five nodes {A, B, C, D, F} connected with a network of backbone routers

*Correspondence to: Department of CS&IT, University of Sargodha, Sargodha, Pakistan. E-mail: humaira.bilalrasul@uos.edu.pk

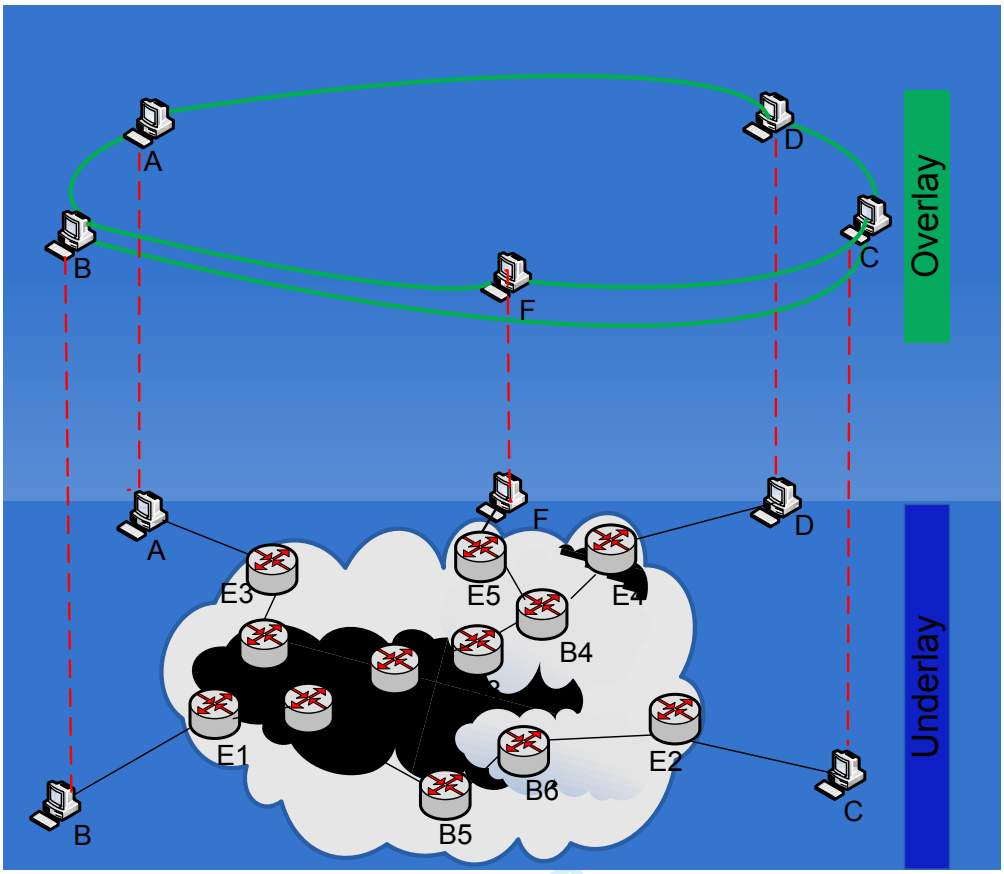

Figure 1. Overlay-Underlay Mapping

$\{B_1, B_2, B_3, B_4, B_5, B_6, B_7,\}$ through edge routers $\{E_1, E_2, E_3, E_4, E_5\}$. Suppose node D wants to download a file. After lookup, node D finds nodes B and F as sources of that file. There are two overlay paths of same length from node D to B i.e., O_1, O_2 and three underlay paths i.e., U_1, U_2, U_3 whereas $O_1 = U_1$ and $O_2 = U_2$.

$O_1 = \{D \rightarrow A \rightarrow B\}$
$O_2 = \{D \rightarrow C \rightarrow B\}$
$U_1 = \{D \rightarrow E_4 \rightarrow B_4 \rightarrow B_3 \rightarrow B_2 \rightarrow B_1 \rightarrow E_1 \rightarrow B\}$
$U_2 = \{D \rightarrow E_4 \rightarrow B_4 \rightarrow B_3 \rightarrow B_2 \rightarrow B_1 \rightarrow B_7 \rightarrow E_3 \rightarrow A \rightarrow E_3 \rightarrow B_7 \rightarrow B_1 \rightarrow E_1 \rightarrow B\}$
$U_3 = \{D \rightarrow E_4 \rightarrow B_4 \rightarrow B_3 \rightarrow B_2 \rightarrow B_1 \rightarrow B_5 \rightarrow B_6 \rightarrow E_2 \rightarrow C \rightarrow E_2 \rightarrow B_6 \rightarrow B_5 \rightarrow B_1 \rightarrow E_1 \rightarrow B\}$

Similarly there are two overlay paths from node D to F i.e., O_3, O_4 and three underlay paths i.e., U_4, U_5, U_6 whereas $O_3 = U_4, O_4 = U_5$

$O_3 = \{D \rightarrow A \rightarrow B \rightarrow F\}$
$O_4 = \{D \rightarrow C \rightarrow F\}$
$U_4 = \{D \rightarrow E_4 \rightarrow B_4 \rightarrow B_3 \rightarrow B_2 \rightarrow B_1 \rightarrow B_7 \rightarrow E_3 \rightarrow A \rightarrow E_3 \rightarrow B_7 \rightarrow B_1 \rightarrow E_1 \rightarrow B \rightarrow E_1 \rightarrow B_1 \rightarrow B_2 \rightarrow B_3 \rightarrow B_4 \rightarrow E_5 \rightarrow F\}$
$U_5 = \{D \rightarrow E_4 \rightarrow B_4 \rightarrow B_3 \rightarrow B_2 \rightarrow B_1 \rightarrow B_5 \rightarrow B_6 \rightarrow E_2 \rightarrow C \rightarrow E_2 \rightarrow B_6 \rightarrow B_5 \rightarrow B_1 \rightarrow B_2 \rightarrow B_3 \rightarrow B_4 \rightarrow E_5 \rightarrow F\}$
$U_6 = \{D \rightarrow E_4 \rightarrow B_4 \rightarrow E_4 \rightarrow F\}$

For lookup, overlay can use only O_1, O_2, O_3, O_4 paths. Due to underlay unawareness overlay cannot see U_3 and U_6 which are direct underlay paths from $D \rightarrow F$ and $D \rightarrow B$ respectively. This

content finding results in sending the same content multiple times on same links, being unable to find physically closest nodes, and last hop problem in which one hop may span several underlay hops. Furthermore, after lookup, overlay selects node B according to its metrics for downloading. Again due to underlay unawareness overlay cannot see that node F is closer than node B and selection of node B will result in more delay and more Internet traffic. So this topology mismatch results in sub-optimal overlay routing and peer selection which can result in a large amount of unnecessary P2P traffic [4, 5, 6]. Internet service providers have started enforcing a quota for P2P traffic and applying smart policies like throttling and deprioritizing, to reduce share of P2P traffic.

Other factors that contribute towards P2P traffic include increase in the Internet usage by home users and advancement in Internet technology. At the same time, the popularity of P2P applications translates into huge volumes of data that these applications generate. Underlay unawareness affects predominantly the said factors because it multiples their effect. All these problems stress the use of underlay awareness for peer selection.

Furthermore, an overlay application is normally only interested in finding optimal routes for its own users without considering other applications, i.e. it performs selfish routing — whereas in the underlay, service providers often use Traffic Engineering (TE) to efficiently utilize the available physical resources for all their users. Both overlay and other ("underlay-based") applications could theoretically be supported with TE by calculating a set of physical level routes on the basis of an input matrix that consists of traffic demands from both the overlay and the underlay. If TE does not have this complete knowledge, the different objectives of TE and overlay routing can however affect the utilization of resources, causing a performance degradation [7, 8, 9, 10, 11, 12]. Similarly, without TE, optimal performance can only be achieved if an overlay has full knowledge of, and control over, the underlay [13].

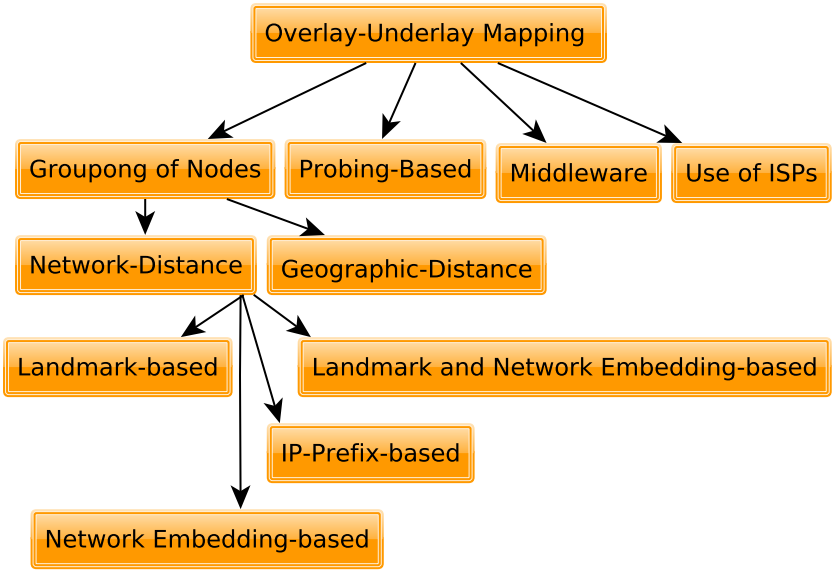

Figure 2. Overlay-Underlay Mapping

It is clear that the overlay-underlay mapping problem consists of various facets, making it difficult to tackle. What is not so clear, it seems, is the amount and the quality of solutions that exist for this problem. In this paper, we address this issue with a comprehensive survey of the research work carried out to solve the overlay-underlay mapping problem up till now. There exists a survey [14] about alleviation of topology mismatch problem in P2P systems, classifying their solutions for structured and unstructured P2P systems but a further classification is missing. However in our paper, we have considered various overlay optimization techniques used by these approaches to get

underlay information with a focus on their tools, metrics and lookup cost etc. For this reason, we have categorized these optimization approaches into grouping of nodes, probing based techniques, middleware and use of ISPs as shown in Figure 2

Here, first category in Section 2 describes the approaches that use proximity information to arrange nodes into groups. Grouping of Nodes is further classified into Network-distance based grouping by determining network position of nodes and geographic-distance based grouping via geographical positions of nodes (e.g., [15]). P2P applications can get network distance information by estimating the network positions of peers with reference to a) landmarks (e.g., GNP [16]) b) IP-Prefixes [17] and c) super peers [18]), via network measurements like Round-Trip Time (RTT) between peers. Therefore we have classified Network-distance based grouping into into four sub groups i.e., Landmark-based groups, Network Embedding-based groups, Landmark and Network Embedding-based groups and IP-Prefix-based groups. In second category presented in Section 3, we have placed those overlay optimization mechanisms that are based on passive or active probing. The third category in Section 4 estimates practical ways of achieving what these applications need, proposing middleware to enable the necessary information flow for better overlay-underlay mapping. Finally, the fourth category in Section 5, we present more "ideal" solutions, presenting overlay optimizations that are specific to use ISPs or third parties to provide underlay information to P2P overlay applications according to their requirements.

2. GROUPING OF NODES

The approaches discussed below use proximity information for arranging nodes into groups to optimize peer selection and overlay routing. Proximity information can be estimated by determining network distance between nodes or geographical distance between nodes.

Based on this, we have classified groups into two major categories:

1. Network-distance based groups
2. Geographical-distance based groups

2.1. Network-distance based groups

Clearly, the "network distance" is an important metric that significantly affects the performance of applications. It can be estimated by determining network positions of nodes via network measurements, for example, Time-to-Live(TTL), Round-Trip-Time (RTT), the number of hops between peers with a variety of tools such as ping, traceroute, pathchar [19] and others. There are many overlay optimization approaches that do grouping of nodes by determining their network positions.

We have classified these approaches into four groups as follows:

1. Landmark-based groups
2. Network Embedding-based groups
3. Landmark and Network Embedding-based groups
4. IP-Prefix-based groups

Following is the detailed description of these methods.

2.1.1. Landmark-based Groups

Landmark is an object that helps other objects to determine their relative positions. The landmarks measure the distance among themselves and the distance between ordinary nodes and landmarks. This is illustrated in Figure 3. Many overlay optimization approaches have a framework of dedicated special nodes called landmarks that is used as a frame of reference for estimating network positions of nodes. These approaches are discussed hereafter.

Id Maps Id Maps is an infrastructure that measures distance between any pair of IP addresses or every globally reachable Address Prefix (AP) on the Internet to answer distance queries [20].

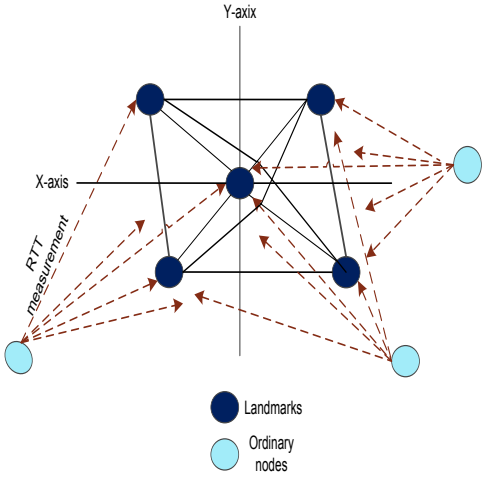

Figure 3. Landmark-based groups

IDMaps serves as an underlying service to provide the distance information to Higher-level services by using protocol such as SONAR/HOPS. Overlays can use this service to get distance estimation between any nodes in terms of latency (e.g., round-trip delay). It has two main components: Tracers and Servers. Tracers serve as landmarks, distributed around the Internet and are chosen from existing systems. Tracers measure and advertise raw Internet distances "Virtual Links (VLs)"to Id Maps clients(APs) by using network probe daemon (NPD) with traceroute. The VLs are raw distances between Tracers (Tracer-Tracer VLs) and between Tracers and APs (Tracer-AP VLs). The distance between any two APs is the sum of distance from these APs to their nearest tracer(Tracer-AP VLs) plus the distance between these tracers(Tracer-Tracer VLs). Servers collect this distance information from different Tracers and provide it to different Id maps clients in response to their queries. These IdMaps clients can build a virtual distance map of the Internet. IdMaps can be considered as pioneering work to examine the network distance estimation problem. However it needs global network topology information for estimating distance. It does not provide precise estimate of distance rather relative distance.

Dynamic Landmarks The authors of [21] describe a locality-aware network for arranging nodes into groups. This grouping criterion uses neighboring groups of a group as landmarks called dynamic landmarks. Nodes measure distances to dynamic landmarks of every group and join the group that has the same distance to dynamic landmarks as the node itself has with these dynamic landmarks. So a group consists of nodes that are close to one another. There are two types of links in this locality-aware overlay : intra-group links and inter-group links. The distance metric can be network latency, RTT, bandwidth or any user defined cost function. It uses dynamic landmarks instead of static landmarks to get the locality information for reduction in average intra-group distance and to direct more traffic inside than that outside the group. There is no upper bound for a group so the number of nodes in each group is not equal. Another drawback is that in mOverlay each node belongs to the group in which it is first introduced. The new node might not be in optimal group as the group choosing process ends after a certain number of rounds [22]. To overcome this problem an adaptive landmark selection approach was proposed in [23] that enahanced mOverlay with learning automata. Learning automata is used as an adaptive decision-making device for landmark selection that is robust during the operation of network. Peers are divided into clusters by locating algorithm of mOverlay and learning automata is used for making robust landmark selection. Afterwards, the authors proposed another adaptive algorithm to reduce toplogy mismatch problem that is based on learning automata and Segregation Model (SSM). Every peer after joining the network uses a neighborhood selection algorithm for finding neighbors and then use the Learning automata to tune it's neighborhood radius. The environment of every peer consists of its immediate

neighbors and neighborhood of its immediate neighbors. Every peer uses Learning automata to find and update an appropriate candidate peer as immediate neighbor from its environment having minimum delay between itself and all peers in environment. Afterwards every peer uses SSM to exchange its connections with that candidate peer in order to reduce the topology mismatch by keeping the degree of network same.

Hierarchical Clustering After mOverlay, the authors of mOverlay proposed hierarchical clustering of peers [24] that uses delay metric as a clustering parameter for joining, splitting, merging and leader election of peers. Peers measure distence from leaders of different clusters and join that cluster having round trip time (RTT) value below predefined threshold. Two clusters are merged if it contains few members and distance between leaders of them is below the predefined threshold for that level. In hierarchy for every layer a different round trip time (RTT) value is used. At leave level there are real clusters that consist of real nodes and at above level there are virtual clusters consisting of virtual nodes. Virtual clusters are used for performance monitoring of join, split and merge functions by intra-cluster and inter-cluster measurements.

Proximity-Aware and Interest-clustered P2P file sharing System (PAIS) [25] is another hierarchical clustering method for peers that is based on both peer interest and physical proximity. PAIS arranges physically closer nodes into a cluster and every cluster is further divided into sub-clusters on basis of proximity and common interests. It further uses proactive file information, bloom filters to further refine the search. Node proximity information is generated by using landmark clustering method. Nodes measure their physical distance to landmarks and nodes having similar distance to landmarks are closer to each other. These nodes are placed in a cluster and than sub clusters are formed on the basis of common interests.

2.1.2. Network Embedding-based groups

In network embedding, network distance measurements or inter-node latency is embedded into a low dimensional metric space for assigning coordinates to every node. Network distance measurements can be obtained by using RTT between nodes. Network embedding can be used to estimate network positions because it estimates proximity of nodes. The distances between coordinates represent real-world latencies between nodes. Many researchers have proposed Network embedding as a way to improve performance of overlay networks. In this section we will review Network Embedding-based optimization methods one by one.

Vivaldi Vivaldi is a decentralized, adaptive Euclidean coordinate system. Vivaldi computes coordinates by using inter-host Internet RTTs. It assigns coordinates to nodes in such a way that the distance between coordinates predicts the latency between nodes [26]. To minimize prediction errors in latency, these coordinates are augmented by a direction-less height vector. The distance between two nodes is the sum of the Euclidean distance and height vectors of nodes. The height vector shows transmission delays on the access links from the node to the core and the Euclidean distance shows delays of geographical distance of core network. The communication overhead is low because it assigns coordinates to nodes by getting information from only a few of them. It does not need any fixed infrastructure of landmarks. An evaluation shows that errors in latency prediction are as low as with the landmark-based coordinate system GNP. Vivaldi operates on latency piggyback so no information is given about measurement overhead [27]. Vivaldi is not suitable for selecting nearby peers so it does not reduce cross-ISPs traffic [28].

Meridian Meridian [29] is a framework built by measuring the network distance between nodes to help in location-based node and arranges these nodes in concentric, non-overlapping rings of increasing selection. Every node measures the distance to a small fixed number of nodes directly without any landmark or distributed coordination radii. Latency is used as a distance metric in this framework. Whenever a node sends a query to find the closest node, it performs a multi-hop search and the query is forwarded along the structure of rings so that each hop exponentially reduces the distance to the destination. Every node has to keep a record of nodes and arrange them into

concentric non-overlapping rings. Meridian shows lesser error to discover the closest node. The main focus of Meridian is on individual node requests, instead of building a global coordinate service, which could help in multiple distance estimations [30].

2.1.3. Land-marks and Network Embedding-based Groups

Network position can be determined with reference to either landmarks, coordinate system or both. There are various approaches to determine the network position, like GNP [31], ICS [32], MITHOS [33] etc. This commonalities are:

1. There is a framework of dedicated special nodes — the "beacon nodes" of ICS, the "landmark" of GNP . In what follows, we will generally use the term "landmark".
2. The landmarks measure the distance among themselves and the distance between ordinary nodes and landmarks. This is illustrated in Figure 3.

After measuring the distance to their landmarks, every method uses a different way of transforming this distance into a network position. A brief overview of these methods is elaborated next:

Global Network Positioning (GNP) GNP models the Internet by mapping nodes to points in a geometric space and carrying out a distributed computation to obtain synthetic/Euclidean coordinates that characterize the positions of nodes [31, 16]. The Distance between Coordinates of nodes represents the Round Trip Time (RTT).

As a first step, inter-landmark distances are measured through ICMP ping messages and transmitted to a central node. The central node then computes a set of landmark coordinates by applying a geometric function and returns the coordinates to the landmarks. These coordinates serve as frame of reference for ordinary nodes. These nodes measure their distance to landmarks with ICMP ping messages and calculate their coordinates in the same way as that of landmarks by applying a geometric function. GNP host coordinates predict the Internet network distances which can be used for a topology inference. Furthermore, the communication overhead is reduced to $O(K.D)$ from $O(K^2)$, here K represents number of hosts and D shows dimensionality of geometric space. In GNP, every node probes all landmarks to compute its coordinates, which can cause performance bottlenecks. The coordinates are not unique in their definition [32]. GNP uses Simplex Downhill to minimize the difference between he measured network distance in the distance data space and the Euclidean distance in a Cartesian coordinate system. Due to initial value used in Simplex Downhill method a single host may have different coordinates. These are absolute coordinates not relative coordinates. Moreover, according to [34] network coordinates shows noticeably worse performance as compared to results shown in [31, 26].

Network Positioning System (NPS) To resolve the above said issue, the authors of GNP designed and implemented the Network Positioning System (NPS) [35] NPS is a system to represent Internet network distance between end hosts. It can provide network position service to different applications like overlays, web applications, bandwidth demanding applications. Overlays can use NPS to determine network position of nodes. In NPS every node that has determined its position can serve as a reference point for other nodes, and the system does not halt on temporary landmark failures. It is a hierarchical architecture that computes the network positions of nodes in Euclidean space in a distributed fashion, while maintaining consistency, adaptivity and stability of host network positions over time. The authors used NPS for network distance estimation of nodes in peer to peer applications mentioned in [36, 37]. This technique is static and also needs external information for calculation of coordinates of reference points. Moreover, there is no guarantee of assigning unique coordinate to every node.

Internet Coordinate System (ICS) ICS is a Principal Component Analysis (PCA) based coordinate system having an infrastructure of beacon nodes [32]. These beacon nodes periodically measure Round Trip Times (RTT) between themselves. The distance between nodes is represented by a distance vector whose dimensions are equal in number to beacon nodes. By using PCA

DOI: 10.1002/dac

based transformation method, this distance data space is projected into a new coordinate system and a transformation matrix is calculated to obtain coordinates of ordinary hosts. After the above procedure, an ordinary host measures its delays to all or a small set of beacon nodes and gets a distance vector. This node gets its coordinates by multiplying this distance vector with transformation matrix. This coordinate system has much smaller dimensions and retains topological information.

ICS also enables an ordinary node to estimate network distance to other nodes without direct delay measurement. An ordinary node can report its coordinates to a DNS-like server that keeps coordinates of ordinary nodes. Any node can get estimated distance of other nodes from this DNS-like server as long as it has these coordinates. ICS shows a better performance in a hierarchical topology with low computation, low measurement overhead and low estimation error for large number of hosts. For better performance, beacon nodes should be well distributed. If the beacon node is the median node of the cluster, estimation errors are smaller than with a the randomly selected node.

Binning Binning [38] is another network-embedding scheme that divides the whole space into bins on the basis of network latency by using an infrastructure of landmarks. Nodes first measure their distance to these landmarks through RTT and arrange the landmarks in an order of increasing RTT. This ordering of landmarks represents the bin of node. Topologically closer nodes will have the same ordering of landmarks, and therefore these nodes belong to the same bin. The nodes in the same bin are closer to one another than nodes of the other bins. This binning scheme requires periodic refreshes for distance measurement and ultimately ordering of landmarks on these measurements for every node.

It needs a fixed infrastructure of landmarks but it is less vulnerable to the landmark's availability than GNP because binned nodes remove the failed landmark from their bin identifier. However every node needs to update the changing status of the landmark [39].

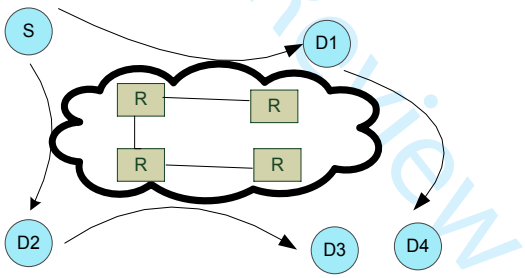

Figure 4. Application layer multicast

2.1.4. Topology Aware Grouping through Application Layer Multicast Topology Aware Grouping (TAG) of multicast nodes at the Application layer [40, 41] is another way to exploit underlying network topology data as shown in Figure 4. The normalized overlay tree cost $L_{TAG}(n)$/û is defined as $L_{TAG}(n)$, the total number of hops consumed by all members n, divided by average number of hops for unicast, û. It uses redundancy of paths for multicast tree construction to reduce duplicate messages and delay penalty and efficient utilization of bandwidth.

Afterwards, based on the concept of TAG, a Multi-domain Topology-Aware Grouping (MTAG) [42] is introduced. The Internet consists of many domains so, MTAG has a concept of a special node in each domain for managing other nodes of the same domain to increase efficiency and reduce time for discovery of topology. This special node is called domain manager. Performance of TAG is good in packet duplication and delay reduction but for larger overlays, it slows down as compared to MTAG.

The same working group that gave the concept of Multi-domain Topology-Aware Grouping (MTAG) introduced subnet topology-aware grouping (STAG) [43]. STAG is also based on concept

DOI: 10.1002/dac

of TAG but with the idea to broadcast the JOIN message before or at the same time in the subnet and to source member to reduce the time required for topology discovery and execution of path matching algorithm.

The Distributed Domain Name Order(DDNO) technique groups the unstructured nodes belonging to same domain [44]. In DDNO, a node keeps half of its connections as sibling connections for nodes belonging to same domain and other half connections as random connections for random nodes. The sibling connections are made by multicast lookup messages, using zone caches. This 1st 1/2 degree connections technique will keep most of the traffic in the same domain to improve performance. The other 1/2 connections will keep the overlay connected, maintain the structure of the overlay (keeping it unstructured) thus reducing the end-to-end delay diameter.

2.1.5. IP-Prefix-based Clustering

IP-Prefix-based clustering means grouping of nodes on the basis of similar IP-Prefixes (network distance in some cases). It gets topological information to solve the overlay-underlay mismatch. Following is a brief overview of these clustering methods one by one.

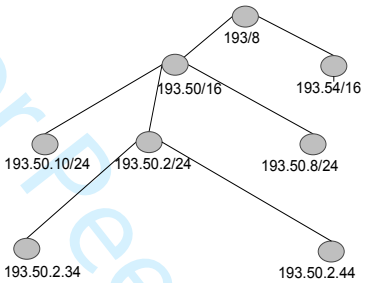

Figure 5. IP-Prefix-based Clustering

TOPLUS Topology-Centric Look-Up Service TOPLUS [17] is a lookup service that is based on hierarchical clustering of peers according to network IP prefixes. This lookup service exploit the topological structure of underlying Internet. TOPLUS extracts information from BGP routing tables, prefixes of ISPs and corporate LANs etc. By using this information, the nodes having common IP-Prefixes are arranged in groups. Afterwards, the groups that are topologically close form super groups and super groups in the same way form hyper groups and so on as shown in Figure 5. It is a good idea to group topologically close nodes but TOPLUS requires external information input from BGP tables or any other source to extract IP-Prefixes. TOPLUS uses an X-OR metric of closeness for routing. For this, TOPLUS has to maintain very large routing tables that is not feasible because of dynamic behavior of P2P applications [39]. Furthermore, there can be correlated node failures because TOPLUS can replicate data on successor nodes in the same group. Failure of that group may cause failure of availability of data so the data should be replicated in different groups.

Cluster graph/Internet Topology graphs Internet Topology graphs are used to model the Internet Topology. These Internet Topology graphs can be generated at two levels of granularity: AS level and Router level. AS graph is very simpler to represent the near to exact picture of Internet topology. A Router graph, on the other hand, is very expensive to generate and too fine grained. However, there is another better model of Internet topology called cluster graph. A cluster graph is at an intermediate level of granularity between AS graph and Router graph. To generate a cluster graph, network aware clustering [45] is used which is an IP-address-based grouping method. It groups topologically close nodes by using the information from BGP routing tables. These cluster graphs can be generated in three ways: *Hierarchical cluster graph* consisting of two levels: i) higher level (AS level graph) and ii) lower level (a mesh of the clusters present in the AS graph). It uses BGP routing table information or updates of BGP routing tables. *Traceroute-based cluster graph*

DOI: 10.1002/dac

models the real topology by sending traceroute to nodes of clusters for constructing cluster graphs. *Synthetic cluster graph* is made by observing some characteristics of the cluster topology such as power law etc. with some interesting metrics such as degree of node or weight of node ect. This clustering is used to model the Internet topology for the application layer [46] and P2P content location and sharing system [47]. Cluster graph is as easy to obtain as an AS graph but represents a more fine grained topology than that of an AS graph.

Topology Inference from BGP Routing Dynamics Topology Inference from BGP Routing Dynamics is another BGP-based passive topology discovery approach that clusters IP address prefixes, using information from BGP routing tables [48]. It groups the IP address prefixes that are updated within same time window according to BGP updates. Afterwards, a standard clustering algorithm is applied to join these groups into larger groups. This temporal clustering produces a more reliable topology there, it can be used by the topology created from pure BGP table by other Internet mapping techniques for path selection, node selection etc.

IP-based Clustering for Peer-to-Peer Overlays IP-based clustering (IPBC) is proximity-based neighbor selection technique that does not use probing to get proximity information between neighbors, rather it uses information available from IP addresses of the nodes for proximity estimation [49]. For this, IPBC investigates the correlation between LCPL of communicating nodes and latency. As a result, IPBC uses the longest common IP prefix length as a measure of proximity among neighbors. This proximity information is stored and used for clustering in the overlay network by making it available to all nodes. Although IPBC is a simple clustering method used for neighbor selection, yet use of static information without any active measurement does not support dynamic network changes [50].

GEO-LPM Geo-LPM is another clustering approach that arranges nodes into clusters on the basis of Longest Common Prefix (LCP) of IP Prefixes and network distance (latency) [39]. In every cluster, there is a managing node called o-router. Every node measures its latency with the o-router of cluster having LCP. If the value is less than a certain threshold value, the node joins the cluster, otherwise it acts as o-router of new cluster itself. In this way, the nodes that are physically close are arranged in a cluster. These clusters are further arranged in a hierarchical manner by aggregating IP prefixes. These clusters form an IP prefix tree that arranges common prefixes of clusters in a CIDR hierarchy (clusters at a higher level aggregate the addresses of their child clusters). The routing is based on Longest Prefix Matching (LPM) which enables a quick and easy way of locating nodes in the Internet. Geo-LPM is more scalable and self-organizing than other IP prefix methods such as IDMaps [20] and TOPLUS [17]).

2.2. *Geographical distance-based Grouping*

Some other overlay optimization approaches strive to use Geographical distance-based group to solve the overlay-underlay mismatch. Geographical distance between nodes is estimated by determining geographical positions of nodes. Afterwards, this geographical distance is used to group nearby nodes. These methods are discussed one in the following.

Globase.Kom Globase.kom [51] is a hierarchical, super peer-based overlay that takes into account geographical distance to perform its operations such as lookup, neighbor selection etc. It divides the whole world into rectangular, non-overlapping zones that are arranged in a tree. Every zone is managed by a super peer that indexes locations of all peers in a zone, super peers of inner zones, its parent super peer, root super peer and interconnected super peers. It is assumed that every peer knows its geographical position by using appropriate devices or databases. Every peer knows about the location of interconnected peers of a zone, the parent super peer and root super peer. All these attributes help a super peer to manage its zone and perform overlay operations. The main drawback of the hierarchical approach is that super peers at higher level can become a bottleneck, resulting in

failure of the whole system. In addition, this solution has not been demonstrated to work for mobile peers [52].

Geo Sensitive Gnutella GnuViz is another tool to draw the network on any geographical map [53]. It first uses a network crawler to explore the network structure by using ping and pong messages. These explored nodes are assigned geographical coordinates (the longitude and the latitude) by a location retriever. Afterwards, a java based script is used to show this geographical map.

Finally, in order to automatically get the geographical location of peers, the authors propose an extension of 8 bytes for geographical coordinates in ping and pong descriptors. It facilitates automatic geographical clustering. Gnuviz tries to map the overlay to structure of underlay. However, it needs to know the geographical position of nodes in an easily comparable unit that causes overhead. It is not necessary that the topological distance is the same as geographical distance [54]. Furthermore, finding low cost nearby nodes has high overhead [55]. The algorithm should not cause more overhead than normal overlay routing.

Geo-Partitioning The clusters that share an LCP might not be close to one another as a network IP address is not correlated to the underlying physical network topology. To overcome this problem, the working group which developed GEO-LPM [39] designed and implemented an overlay that closely matches the underlying network topology [15]. First, [15] uses GEO-LPM for clustering and geographical partitioning (Geo-Partitioning) is used to group geographically-close clusters that have low latency in a tree like structure. Geo-Partitioning divides the whole geographical space into partitions in a tree like structure. These partitions are connected to one another with a very small latency value and every partition is managed by a head node called an o-router. Every cluster measures its distance with the o-router to find its closest partition and later on, this cluster measures its distance to the o-router of partitions of the next level until it finds the closest partition. Every cluster finds its neighbors in its own partition by measuring the latency value in other partitions. The search algorithm of Geo-Partitioning scheme has complexity of O(logM) where M is the number of clusters. The resulting overlay consists of clusters that are close in terms of latency.

The first category of underlay-aware proposals, analyzed in section 2 is grouping of nodes according to different parameters i.e., network distance and geographical distance. The majority of these appoaches focus on proximity based grouping of nodes for finding the shortest overlay routes by selecting nearest nodes according to proximity information. Apparently, this grouping could yield better performance for overlay construction and peer selection [56, 57] – but proximity based grouping alone is not sufficient.

We have summarized the comparison of tools, metrics, overhead of these methods used to get proximity information for this proximity-based grouping in Table I:

Table I. Comparison of Proximity based Grouping.

Begin of Table

Type	Name	Tool	Metric	Overhead	Moni-tors/Landmarks Deployment	References
Landmarks-based Groups	IdMaps	Traceroute	Latency, bandwidth	Proximity-based clustering of APs, O(C2 +AP) measurements	Transit AS's	[20]
	PAIS	Traceroute	Latency, Peer Interest	Proximity-based clustering of peers, ? measurements	peers	[25]

Continuation of Table I

Type	Name	Tool	Metric	Overhead	Moni-tors/Landmarks Deployment	References
	Dynamic Landmarks	active probing	Network latency, RTT, bandwidth or any user defined Cost function	$O(logN)$	Groups of moverlay serve as landmarks	[21]
Network Embedding-based Groups	Vivaldi	Ping	RTT	No information provided	No landmarks	[26]
	Meridian	Ping	Latency RTT	$O(logN)$	No landmarks	[29]
Landmarks and network Embedding-based Groups	GNP	Ping	RTT	$O(K.D)$	Static landmarks deployment on fixed nodes of Infrastructure	[31, 16]
	NPS	Ping	RTT	same as GNP	Any set of existing nodes can serve as landmarks	[35]
	ICS	Ping	RTT	ICS $<$ GNP	Beacon nodes Infrastructure	[32]
	Binning	Ping	RTT	$NH2$	Fixed infrastructure of nodes	[38]
	TAG	Tracer-oute,pathchar	Delay, Bandwidth	$L_{TAG}(n)$/û	No land-marks/member of a multicast session	[40, 41]
IP-Prefix-based Clustering	TOPLUS	Traceroute, King	Latency	Time comparable to IP routing	BGP Routing Tables, the Prefixes of ISPs and the corporate LANS	[17]
	Cluster Graph	Traceroute, Power law	RTT, Update Time of BGP Routing Table, Characteristics such as Power law	?	BGP Routing Table	[45]
	Topology Inference from BGP Routing Dynamics	IP-Prefixe + Updates from BGP	Weighted Sum of number of Updtaes of IP-Prefixes within same Time Window	$O(n^2\ log\ n)$	BGP Routing Table	[48]
	IPBC	Correlation between Longest Common IP Prefix Length (LCPL) and Latency	Latency	Establishment of Correlation between (LCPL) and Latency + DHT based lookup	DHT, Overlay	[49]
	GEO-LPM	IP Prefixes + Proximity	Longest Matching Prefix + RTT	GEO-LPM $<$(Binning,GNP)	o-routers	[39]

Continuation of Table I

Type	Name	Tool	Metric	Overhead	Moni-tors/Landmarks Deployment	References
Geographical Distance based Grouping	Globase.Kom	Plate Caree projection, Geographical coordinate	latitude-longitude points of geographical location	Protocol overhead + Load Balance among the Peers	Super Peer	[51]
	Geo sensitive gnutella	Ping, PONG, Geographical coordinates	RTT, the longitude and the latitude of geographical location	Network structure exploration + geographical position evaluation of each explored node + drawing and display of geographical network map	Beacon Server, Network Crawler, NetGeo	[53]
	GEO-Partitioning	IP Prefixes + Proximity + Geographical Partitioning	Longest Matching Preix + RTT	$O(logM)$	o-routers	[15]

End of Table

3. PROBING-BASED OVERLAYS

In this section, a few P2P applications are reviewed that use probing to get underlay information such as topology, bandwidth etc. and use it for overlay optimization regarding overlay formation, peer selection, fast failure detection and recovery etc. Probing can be active probing or passive probing or a combination of both approaches. Some of these well known approaches are examined in this section.

Resilient Overlay Network Resilient Overlay Network (RON) is an application-layer overlay architecture that enables end hosts and applications to quickly recover from path failure and gives them flexibility in choosing end-to-end paths [58]. It improves the performance of the underlying Internet routing protocol called BGP by using a combination of active and passive probing. BGP takes more time to recover from path failures because it compromises reliability for scalability. RON, on the other hand, quickly recovers from path failures. For this, it uses an application-level protocol to communicate with the other nodes in the RON and continuous probing to monitor links based on three metrics: latency, packet loss and throughput. RON stores this information in a performance database of every RON node and uses it for path selection, monitoring the functioning and quality of Internet paths, failure detection and recovery. Failure could be link failures or path failures, leading to application outage and performance failure. It also allows policy routing, path selection and routing decision based on application requirements. Though the results that are obtained by deploying RON provide an application with resilient network connections, there are some issues that need to be discussed. Its scalability is an issue because RON compromises scalability for the sake of reliability. RON can scale up to 50 AS because for reliability, a high network monitoring is required with probing cost reaching up to $E{=}O(n^2)$ for a network of n nodes. The overall performance of network is degraded if every application chooses its own RON to improve performance. RON is suboptimal if nodes are behind a NAT [59].

Resilient Overlay Routing Resilient Overlay Routing (ROR) [60] like RON is also a probing-based overlay used for quick route failure detection and recovery but it is designed for structured

P2P. For fast failure detection, ROR precomputed backup paths, sends periodic probes to these normal, backup paths and continuously estimates link quality with total bandwidth consumed (TBC). For route failure recovery, first reachable link selection (FRLS) is used that chooses a route from all available backup paths whose link quality is above a defined threshold. If FRLS does not work in the next step constrained multicast (CM) is used in which messages are sent to multiple outgoing links. After recovery from failure, there next step is maintenance of routing redundancy by replacing the failed route with a new route and restoring the pre-failure level of path redundancy. ROR also supports tunneling to allow traffic of legacy applications to be sent through this overlay. When compared with RON, the probing interval of RON is high as compared to ROR. The probe overhead is 56Kbps for 200 nodes whereas, in RON it is 33Kbps probe overhead for 50 nodes. RON is also used for failure detection and recovery but it is used for unstructured overlays with probing cost $E=O(n^2)$.

Alternate/Backup Path Path switching is used for dynamic selection of the best alternate path when multiple paths are available. There is a significant improvement in performance when path switching is used [61, 62]. In [63], a relay node is used for producing multiple alternate paths. The relay node is selected from candidate relay nodes that gives the cast in terms of information storage and processing. It finds alternate paths by using AS–level path disjointness information combined with an earliest-divergence rule that uses only local AS level information and selects paths that diverge from the default path at the earliest possible point; [64] places relay nodes for intra-domain path diversity at the IP layer by using an algorithm that gives minimum penalty.

Alternate/backup paths with a primary path is also used for path failure recovery. The primary and backup paths may be separated at the IP or overlay layer but they are not disjoint because they can share a physical link [65, 66] and failure of that link causes failure of both primary and backup paths. Since paths originating in different service providers often overlap, the primary and backup paths should be disjointed at the physical layer too. In [67], for backup path allocation, a Correlated Link Failure Probability Model is introduced. It calculates a route for backup paths by minimizing joint path failure probability between the primary and the backup paths.

Path Independence at IP layer Applications like RON and ROR exploit the routing redundancy to send packets on an alternate path when the primary path fails. Prior studies [68] have shown that these applications are unable to recover from 40-50 percent of path outages. The alternate path can fail due to sharing of physical links, failure of paths present in the same administrative domain due to failure of the domain, failure of network access point having both paths and geographical adjacency. These factors showed that overlay paths should be disjoint at IP layer. Topology-aware overlay networks [69] is a framework that tries to maximize path independence at the IP layer by exploiting path redundancy and using multi-homing at endpoints. There is an off-line topology analysis for placing overlay nodes. Afterwards, topology-aware node placement heuristics are made by measuring the diversity between different Internet Service Providers (ISPs) and also between different overlay nodes inside each ISP to ensure path diversity. This heuristic selects how many ISPs and how many nodes inside each ISP will be enough for path diversity to avoid path failures. On top of this topology-aware overlay framework, a routing mechanism is used that shows the same level of path diversity and performance for both single-hop overlay routing and multi-hop routing in more than 90 percent of cases. Next in [70], the group designed a topology-aware network by adding latency along with topology knowledge for node placement, use of source based single-hop overlay routing and increase of multi homing at endpoints.

Fewest Common Hops (FCH)[71] is a proximity-path-disjointedness based peer selection method that combines two dimensions (modes) of adaptations in peer selection:

1. **Proximity based peer selection** *The proximity based peer selection means client selects the candidate peer that is nearest to the client with shortest source-destination path in terms of number of hops or rtt etc.*
2. **Path-disjointedness based peer selection** *Path disjointedness means that selected paths should have no common intermediate routing hops or links.*

The disjointedness criteria is used in conjunction with the proximity to select the maximally-disjoint shortest paths. A maximally-disjoint path shares the fewest number of common routing hops with the source-destination paths of already selected peers. This means that a path established between a source node and a destination node is the shortest and shares the least number of routing hops with the paths of already selected peers. To do so, a client peer gets list of candidate peers, sends ping to these candidate peers gets and stores the resulting path topology from the client to these candidate peers. FCH used ping because it is the simplest tool that can be used to indicate path disjointedness. The number of pings is equal to the number of peers got from tracker; the storage and time effort is therefore $O(m)$, where m is the number of peers. Afterwards, for downloading a file or pieces of file, the path topologies are compared to select the peers that have the fewest routing hops and maximum path disjointedness with peers that are already selected. The routing metrics are the path hop-count to calculate proximity and common hop-count to calculate path disjointedness.

It is another path disjointedness based peer selection approach that uses a genetic algorithm to select a partner as a parent for streaming that has maximum paths between peers and set of candidate partners[72]. Traceroute is used to extract physical path information between itself and candidate partners This information is exchanged periodically through buffermap messages. So the overhaed is small.

Path Segment The overlay network monitoring systems usually require E=$O(n^2)$ measurements to recover from path failure and improve performance. This measurement overhead creates heavy traffic and becomes an issue in its scalability. To overcome this issue, in [73] a tomography-based network monitoring system is proposed that uses path segment to describe properties of all paths. Path segment means selection of a basis set of k paths that completely describes all E=$O(n^2)$ paths by using an algebraic approach. After selecting the basis set of paths, these path segments are monitored by tomography and loss rate of these paths is computed. From this loss, the loss rate of all paths is inferred. The measurement overhead is $O(n\ log\ n)$. The metric can also be extended to latency. A further enhancement is suggested in [74, 75] that handles the issues of dynamic topology changes for instance, nodes joining/leaving, path change, routing change, and load balancing, errors in measurement, checking of scope of scalability. Finally, in [76], there is the real implementation of the system as Adaptive Overlay Streaming Media with emphasis on enhancements made in [73].

LTM Location-Aware topology matching (LTM) in P2P systems [56, 77], is a source probing based overlay in which every peer cuts down low productivity connections and connects with physically closer ones. For this, every peer sends/floods a detector containing TTL that records delay information. This information is used to make connections to peers with closer nodes as direct neighbors and cut of slow, inefficient, redundant links. LTM proposes an amendment to the Gnutella protocol by adding a header containing TTL. LTM uses an application level measurement method that requires complicated control lacking sufficient accuracy [78]. The measurement overhead of LTM is $O(n)$ because it needs to synchronize all peering nodes.

To resolve the above said issue, the author of LTM designed and implemented "A Two-Hop Solution"to solve Topology Mismatch [79], in which one special query message type Piggy Message (PM) including two fields: Neighbor IP Address and Neighbor Distance is added in Gnutella 0.6 P2P protocol. By using PM a peer measures distance with direct neighbors and puts the longest distance peers in its will-cut list. A peer also maintains a distance cache that keeps list of already probed peers to avoid duplicate probing. Further to reduce the load on network a peer uses this PM by two selection policies: pure probability-based (PPB) policy and new neighbor triggered (NNT) policy. PPB has a predefined probability for every query to include PM message and each PM is piggybacked for only one hop. By using NNT a peer sends PM along query messages on detection of a newly arrived neighbor for only two hops.

SAT-Match [80] is another effort in which a P2P system adaptively changes its overlay structure to match the underlying physical topology. Peers by using lightweight probing to its neighbors adaptively change the overlay network connections to minimize the average logical link latency and ultimately the average response time of lookup routing.

DOI: 10.1002/dac

One Hop Lookups for Peer-to-Peer Overlays One Hop Lookups for Peer-to-Peer Overlays [81] is a robust and low latency based peer-to-peer lookup system that route lookup queries in just one hop. For this every node stores complete routing table that has complete membership information of all overlay nodes. Similarly in [82] authors proposed OnehopMANET that combines structured P2P with underlay routing protocol to get lookups in one hop with routing complexity of O (1) for the overlay routing. It uses cross-layering to establish a channel that passes routing information between the adopted underlay routing protocol and OneHopOverlay4MANET. This underlay routing information is used to build the overlay and populate its routing tables. A cross-layer channel is used to pass routing information between the adopted underlay routing protocol and OneHopOverlay4MANET.

An analytical summary of the probing-based optimization works discussed above is shown Table II.

Table II. Comparison of Probing-based Overlays

Type	Name	Tool	Metric	Overhead	Moni-tors/Landmarks Deployment	References
Probing based Overlays	RON	Active and Passive Probing	Latency, packet loss rate, throughput	$O(n^2)$	Nodes deployment in different Internet Routing Domains	[58]
	ROR	Ping	Network latency, RTT	$ROR < RON$	Overlay aware Client Daemon	[60]
	How to Select a Good Alternate Path in Large Peerto- Peer Systems?	Traceroute, Ping	IP Hops,AS Hops	?	Peer	[63]
	Path Independence at IP layer	Traceroute, Ping	Path diversity, Latency	Gathering of direct and indirect path information + Placement of Overlay nodes inside an ISP network + Choosing a set of ISP networks	ISPs,Peer	[69],[70]
	Path segment	Tomography	Loss rates, Latency	O(n log n)	Overlay Network Operation Center (ONOC)	[73],[74, 75],[76]
	LTM	TTL2-detector	delay	$O(n)$	Peer	[56, 77]
	Fewest Common Hops (FCH)	Traceroute, Ping	Common IP Hops	$O(C.Peers)$	Peer	[71]
	Genetic Algorithm for P2P video streaming systems	Traceroute	IP Hops		candidate partners	[72]

This Section 3 presents approaches that are based on passive or active probing. These probing based overlays can be realized for small or medium sized P2P systems because the discovery of the topology or alternate paths requires a lot of active and passive probing messages and their maintenance also requires continuous probing which generates a large amount of unnecessary traffic, limiting the application's scalability. Redundancy is another issue in these probing based P2P systems because the network itself has already a picture of the network from different network vantage points. To rediscover this information by probing is a waste of time and resources.

4. MIDDLEWARE

Another way to address the overlay-underlay mapping problem is to use a middleware between overlay and underlay as shown in Figure 6. Its purpose is to extract information from the underlay, refine it according to requirements of the overlay and provide it to the overlay. It is similar to the middle layer of a three-tier single system architecture, but it is stretched across multiple systems or applications. A few of the many proposed middlewares in overlay are discussed here.

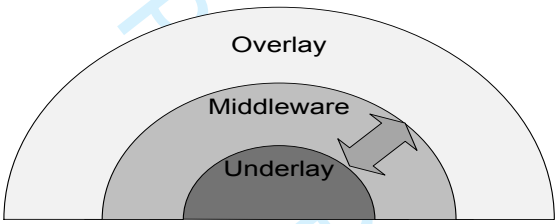

Figure 6. Middleware

A Routing Underlay for Overlay Networks [83] describes routing underlay that lies between overlay and underlay. It extracts and aggregates topology information from the underlay and provides it to the overlay for application specific routing. It is a layered structure comprising of:

1. *Topology probing kernel* the bottom most layer provides a set of basic operations to library of routing services such as graph of known topology, path taken by packets and distance between any two points. For this, it refines raw topology information provided by BGP routing tables by sending probes (pings, traceroute).
2. *Library of Routing services* the upper layer that gets information provided by the topology probing kernel. It uses this information, its own heuristics and dynamic probing to provide a set of services, for example disjoint paths between two nodes, nearest neighbors in terms of distance and building a representative mesh of the underlying Internet.
3. *overlay services* at the top-most layer provides these services to different applications according to their requirement of topology information.

The idea to build a shared underlay seems good as it aggregates probes and reduces costs. However, the actual effectiveness lies in how well this underlay meets requirements of different overlays.

Matrix: Adaptive middleware for Distributed Multiplayer Games Massive multi-players online games (MMOG) had difficulty in dynamic load management and provision of low latency to its clients. To solve these problems of MMOG, a layered and distributed middleware called –Matrix– [84] was proposed.

Matrix is another layered and distributed middleware that was especially proposed for massively multi-player on-line games to reduce the latency for their players and dynamic load management. It has four components.

1. *Game Clients*
2. *Game Servers*
3. *Matrix Server*
4. *Matix Coordinator*

MMOG handles Game Clients and Game Servers whereas Matrix itself handles the Matrix Server and the Matrix Coordinator.

It is based on the idea of a radius or zone of visibility defined for every player. The whole space is divided into different non-overlapping partitions. There are different clients or players in every partition and every partition is assigned to a distinct Game Server. A Game Server is updated by every player about its activity, status and in return the Game Server updates a player with only those events that occur in its zone of visibility. Players can dynamically change their Game servers. Every Game Server informs a Matrix Server about the radius of visibility of it's players and it's current load. Game Servers forward all packets to a Matrix Server for further processing. A Matrix Server is responsible for routing of packets and load balancing between Game Servers. If packets are spatially tagged, a Matrix Server finds a peer Matrix Server from overlap tables, provided by the Matrix Coordinator (MC). It knows the range of game servers connected to it. MC calculates overlap regions for all the Matrix Servers in the game with the help of geometric algorithms, map range and radius of visibility provided by the Matrix server. It sends the overlap regions to every Matrix Server along with the list of Matrix servers that should be updated with events. MC calculates overlap tables again on changes. The main objective of Matrix is to provide a middleware for game developers that not only facilitates dynamic load management of partitions, but also Game Clients by reducing latency.

Performance Enhancement via Two-Layer Support for Peer-to-Peer Systems using Active Networking This middleware is based on active networking technology to bridge the gap between overlay and underlay [85]. It is a two layered framework that controls and coordinates network resources to enhance user-perceived service and optimize network performance. This middleware consists of a framework of active nodes placed at network edges that serve as a coordination and control framework between overlay and underlay. The underlay could aggregate routing and topology information by capturing packets from edge active nodes using BGP, SNMP, netflow etc., it could also provide it to an overlay with the help of modules like measurement, traffic engineering and service control and differentiation.

A Modular middleware for High-level Dynamic Network Management SmartG [86] is a social insect paradigm-based modular and distributed middleware that is proposed for high level network management. It is modular and multi-layered, consisting of three layers as given below:

1. *Smart Signaling Layer (SSL)* is ant-based monitoring layer that collects information about network status by running ants or mobile swarm agents on every node. These ants move across the network and collects information about resources and store it in a data-warehouse layer.
2. *Data-warehouse Layer* is an intermediate layer used for information exchange between Smart Signaling Layer (SSL) and Smart Resource Management Layer (SRML).
3. *(Smart Resource Management Layer SRML)* performs high level management tasks such as load balancing. SRML is implemented by using application-specific intelligent agents. It takes decision of transferring load from overloaded nodes to free nodes on the basis of information gathered by these local agents according to application requirements and information stored in the data warehouse by the SSL.

Anthill: A Framework for the Development of Agent-based Peer-to-Peer Systems Anthill [87] is another social insect based framework that facilitates design and development of P2P systems. It consists of a system of interconnected nests, acting as middleware layer. Anthill acts as an interface in different P2P applications requesting different services like storage management, communication and topology management and ant scheduling. Every nest has three logical modules communication layer, resource manager and ant scheduler. For provision of services, these nests generate ants that move across the nests and share their computational and storage resources to fulfill request. Anthill facilitates P2P applications having dynamic requirements. Furthermore, it includes a simulation environment that helps developers to evaluate the performance of P2P applications before their deployment. We have made a comparison of existing works on middleware to solve overlay-underlay mismatch in Table III.

Table III. Comparison of middlewares

Type	Name	Tool	Metric	Moni-tors/Landmarks Deployment	References
middleware	A Routing Underlay for Overlay Networks	Traceroute, ping, BGP routing tables	AS hops, router hops, measured latency(RTT)	Topology probing kernel	[83]
	Matrix: Adaptive middleware for Distributed Multiplayer Games	spatial coordinates, radius of visibility	latency	Game Servers,Matrix Servers,Matrix Coordinator (MC)	[84]
	Performance Enhancement via Two-Layer Support for Peer-to-Peer Systems using Active Networking	Active Networking	routing and topology information	edge nodes	[85]
	A Modular middleware for High-level Dynamic Network Management	mobile swarm agents	network status, availability or location of nodes etc	data-warehouse layer	[86]
	Anthill: A Framework for the Development of Agent-based Peer-to-Peer Systems	swarm agents	application specific	The Nests	[87]

In this Section 4, we have critically analyzed different P2P middlewares that were proposed to extract, process and refine underlay information and provide it to overlays/applications according to their requirements. The idea of using middleware is good for reducing the burden of processing, probing and storage on P2P applications and provision of information according to requirement only. However, these approaches introduce an additional layer that has an issue of additional overhead. Moreover, middleware like probing also face the problem of redundancy by rediscovering information that network already has, from different vantage points. It will increase cross-ISPs traffic and the burden on networks.

5. USE OF ISPS

This section reviews some approaches that use ISPs/third parties for providing underlying network information (topology map, bandwidth, storage capacities I.e., ALTO etc) to these P2P applications.

It helps in better utilization of these resources i.e., overlay construction, routing and peer selection. Some approaches also propose interception of P2P traffic by ISPs/third parties and redirection of it to local peers for promoting traffic localization to reduce costly cross-ISPs traffic, whereas some other approaches propose modifications in the P2P protocol for traffic localization. In this section we have discussed these approaches one by one.

One of the first approaches that propose ISPs to intercept P2P-traffic at edge routers and redirect them to P2P-clients within the same ISP was presented in [88]. Internet service providers apply smart policies like throttling, and de-prioritizing, to reduce the traffic redundancy for decreasing the share of traffic and ultimately cross-ISPs traffic [89]. Bandwidth throttling by ISPs reduces the cross-ISPs traffic but causes a significant increase in download time. There are two other ways to reduce this cross-ISPs traffic by using caches [90, 91, 92] and gateway peers [88] but these solutions need higher bandwidth nodes acting as cache or gateway peer to keep download time optimal. It needs an infrastructure, limiting their scalability. To overcome these problems, there is a need of a collaboration between ISPs and peers. These approaches are discussed below:

Biased neighbor Selection Biased Neighbor Selection (BNS) can reduce cross-ISPs traffic by keeping the download rate optimal. In BNS, the process of neighbor selection is biased by choosing majority, not all of its neighbors within the same ISP and remaining outside ISPs instead of random neighbor selection [93]. Peers within the same ISP form a cluster, connected with other clusters. BNS can be implemented by either modifying trackers and clients or by using P2P traffic shaping devices. ISPs can easily use P2P traffic shaping devices for implementation of BNS instead of modification in the tracker and client because this modification requires extra effort. Biased neighbor selection requires no extra infrastructure, and can be combined with other ISP policies such as throttling, cache etc. to improve them further.

BNS has no scalability issue or need of an infrastructure. However, BNS increases the probability of unchoking of a nearby peer for download, but it cannot guarantee it [94].

ALTO The Application Layer Traffic Optimization (ALTO) IETG working group develop standards to provide underlying network information with the help of ISPs or third parties [95] to P2P content distribution applications for better peer selection. The ALTO architecture consists of

1. ALTO Server: It is operated by ISPs and responds to queries from ALTO clients. The ALTO Server can provide a number of services as follows:
   (a) The Map Service provides a network map and a cost map to the clients, and leaves the computation of the best path to the client.
   (b) The Map Filtering Service is a Filtered version of the Map Service to be used on resource constrained clients. The clients can specify the parameters to be used in the filtering.
   (c) The Endpoint Property Service can be used to query the service for the properties of individual hosts. Examples of such properties are the network location or the connection type.
   (d) The Endpoint Cost Service provides way to compute the costs between one or more source addresses and one or more destinations. The results can be numerical or ordinal.
2. The ALTO Client: is an application that can be run on P2P clients, P2P trackers or other users that need network related information. This network related information includes topology information, bandwidth availability, provider's policies and connection types of hosts.
3. ALTO Service Discovery entity: is used to discover the location of the server. For the communication between the server and client, the ALTO Protocol is used.

A Peer-to-Peer client can use the information provided by the service of the ALTO server to determine which of the other known peers are good candidates to be chosen as neighbors, based on path costs or other properties provided by the server.

P4P P4P [96] also uses ISPs to explicitly provide required information for better-than-random initial peer selection to different applications such as P2P content distribution. P4P uses a service called iTracker to provide the required information to peers. iTracker provides three interfaces for users as given below:

1. Policy: It shows preferences of ISPs regarding connection with other ISPs
2. P4P-distance: It provides the distance and cost of connection to different peers to requesting peer. For this, it arranges these peers into a simple ordered list according to distance or cost. the peer can use this information for peer selection.
3. Capability: It shows which services are provided by P4P to different P2P systems. These services will help in selecting nearby peers resulting in reducing cross-ISPs traffic and latency. The P4P server is present in the same as the client peer so peer selection requires a simple query.

Ono Ono [97] is another plug-in to improve download speed of BitTorrent by identifying nearby peers. It adds nearby peers in a neighbor set of peer. To identify nearby peers, Ono uses Content Delivery Networks (CDN). Users of CDN are directed to their nearest replica by DNS. Peers that are connected to the same replica are most likely to belong to the same AS and close to one another than other peers. This information is used in BitTorrent to add nearby peers to the neighbor set of peers. This will reduce latency and number of AS hops. Ono forms a cluster of users that are directed to the same CDN server by reusing network measurements from these content distribution networks. Measurements show that the latency and the number of IP and AS hops are significantly decreased by this system.

Biased Unchoking BNS [93] helps to make the neighbor set have a major portion of local or nearby peers. At unchoking decision is taken on the basis of metric MO(x) which is the download rate in the last 10 seconds in case of BitTorrent. The unchoking process in Bittorent is not locality aware because there is no hard binding for the unchoking process to choose among these nearby peers. Biased unchoking (BU) adds locality information (number of hops) of neighbors as a metric in the unchoking process [94]. A peer exchanges data about ”good” locality $L(x,y)$ with neighbors. Every peer defines a threshold value of this locality information and divides the candidate peers into two sets ($L(x,y)<=T$, $L(x,y)>=T$) based on this locality information. Afterwards, the peer can unchoke peers from the set of nearby candidate peers. BU is more effective in high load when there are more interested peers than available slots and BU gives preference to local peers. However, BNS when combined with BU, is more effective in terms of reducing cross-ISPs traffic and download time. We have shown an analysis of different tools, metrics and overheads of overlay optimizations proposed for multi-source downloading in Table IV.

Table IV. Comparison of overlay Optimization by using ISPs

Begin of Table						
Type	**Name**	**Tool**	**Metric**	**Overhead**	**Moni-tors/Landmarks Deployment**	**References**
Use of ISPs	Biased neighbor Selection	Traffic shaping devices, Modification in Tracker and client	latency	Calculation of Download time + ISP traffic redundancy	ISPs, Tracker	[93]
	ALTO	map service	topology,cost etc	provision of all informa-tion(topology, bandwidth, cost etc) to clients	ALTO Server at ISPs	[95]

Continuation of Table IV

Type	Name	Tool	Metric	Overhead	Moni-tors/Landmarks Deployment	References
	P4P	p4p-distance	Completion time, P2P bandwidth-distance product, etc	Implementation of iTracker with p4p-distance interface and appTracker	P4P servers within same AS	[96]
	Ono	DNS	Hops,	Periodic DNS lookups on popular CDN names for maintaining ratio maps + Comparison of ratio maps with those of other peers to determine cosine similarity + Bias traffic towards peer having similar redirection behavior	Replica of CDN	[97]
	Biased Unchoking	Locality	IP Hops AS Hops	Exchanges of data with neighbors with a "good" locality value $L(x,y)$ + Selection of candidate peers from two sets($L(x,y)<=T$, $L(x,y)>=T$)	Information Server	[94]

End of Table

Another class of approaches in this Section 5 strive to use ISPs or third parties to provide underlay information (proximity) for node selection. However, ISPs, third parties, other P2P applications or end users might not cooperate because they do not have a natural incentive to do so. Moreover, there are issues of security, privacy, storage, bandwidth consumption, and continuous maintenance of information. Efforts are therefore being made to solve these problems.

6. ANALYSIS SUMMARY

We have discussed the merits and limitations of overlay optimization mechanism that we have classified into four major classes. Table I summarizes the comparison of tools, metrics and overhead of methods used to get proximity information for this proximity-based grouping. Table II presents an overview of the operation, metrics and overhead of existing probing-based optimization works described in previous section. Similarly Table III shows a comparison of existing works on middleware to solve overlay-underlay mismatch. The table IV compares different overlay optimizations proposed for multi-source downloading in terms of tools, metric, overhead and monitors deployment. While by no means comprehensive, we believe that these tables capture the essence of the discussion and analysis done in the previous sections.

7. CONCLUDING REMARKS

This article has presented a brief overview of various schemes in P2P overlays that are proposed to solve the overlay-underlay mapping problem. We have made a state-of-the-art comparison for addressing the overlay-underlay mismatch in terms of their operation, merits, limitations and future directions.

The majority of underlay-aware proposals for peer selection focus on finding the shortest overlay routes by selecting nearest nodes according to proximity information. Locality-awareness is considered as a promising approach to increase the efficiency of content distribution in P2P networks for instance, in BitTorrent. It is intended to reduce the inter-domain traffic which is costly for Internet service providers (ISPs) and simultaneously increase the performance from the P2P users viewpoint in terms of reducing download times. This win-win situation should be achieved by a preferred exchange of information between peers that are located close to one another in the underlying network topology. Traffic localization can reduce inter-ISP traffic but increase traffic on intra-ISP links that may potentially downgrade the download speed at the peers [98].These locality policies require different system configuration parameters to work like number of unchoked peers [99] and they require a modification of the existing protocol, or interventions by Internet service providers (ISPs).

8. WHAT IS STILL MISSING?

Conclusively, we can say that existing solutions largely focus either on proximity based, probing based or third parties cooperation based peer selection. Most of these solutions do not consider path-disjointness between selected peers along other selection parameters. Path disjointness means that selected overlay paths should have a minimum number of common intermediate hops. If the selected shortest overlay routes are shared between peers, this may cause shared bottleneck both at the access and core networks. It will result in congestion on these shared paths and ultimately in sub optimal performance. So these proximity based nearest peer selection will not always improve the download speed but can slow down. This gap was identified through a critical reading and analysis of the literature. To improve this situation, further research is needed on the combination of these features: a Peer-to-Peer system that uses proximity along path disjointness both at the access and core networks for peer selection.

REFERENCES

1. Ripeanu M. [15] peer-to-peer architecture case study: Gnutella network. *Proceedings of the First International Conference on Peer-to-Peer Computing*, P2P '01, IEEE Computer Society: Washington, DC, USA, 2001; 99–. URL http://dl.acm.org/citation.cfm?id=882470.883281.
2. Ripeanu M, Iamnitchi A, Foster I. Mapping the gnutella network. *IEEE Internet Computing* Jan 2002; **6**(1):50–57, doi:10.1109/4236.978369. URL http://dx.doi.org/10.1109/4236.978369.
3. Aggarwal V, Bender S, Feldmann A, Wichmann A. Methodology for estimating network distances of gnutella neighbors. *GI Jahrestagung (2)*, *LNI*, vol. 51, Dadam P, Reichert M (eds.), GI, 2004; 219–223. URL http://dblp.uni-trier.de/db/conf/gi/gi2004-2.html#AggarwalBFW04.
4. Sen S, Wang J. Analyzing peer-to-peer traffic across large networks. *IEEE/ACM Trans. Netw.* Apr 2004; **12**(2):219–232, doi:10.1109/TNET.2004.826277. URL http://dx.doi.org/10.1109/TNET.2004.826277.
5. Zhang H, Kurose JF, Towsley DF. Can an overlay compensate for a careless underlay? *INFOCOM*, IEEE, 2006. URL http://dblp.uni-trier.de/db/conf/infocom/infocom2006.html#ZhangKT06.
6. Stutzbach D, Rejaie R, Sen S. Characterizing unstructured overlay topologies in modern p2p file-sharing systems. *Internet Measurment Conference*, USENIX Association, 2005; 49–62. URL http://dblp.uni-trier.de/db/conf/imc/imc2005.html#StutzbachRS05.
7. Liu Y, Zhang H, Gong W, Towsley D. On the interaction between overlay routing and traffic engineering. *in Proceedings of IEEE INFOCOM*, 2005.
8. Liu Y, Zhang H, Gong W, Towsley DF. On the interaction between overlay routing and underlay routing. *INFOCOM*, IEEE, 2005; 2543–2553. URL http://dblp.uni-trier.de/db/conf/infocom/infocom2005.html#LiuZGT05.
9. Qiu L, Yang YR, Zhang Y, Shenker S. On selfish routing in internet-like environments. *Proceedings of the 2003 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, SIGCOMM '03, ACM: New York, NY, USA, 2003; 151–162, doi:10.1145/863955.863974. URL http://doi.acm.org/10.1145/863955.863974.
10. Jiang W, Chiu DM, Lui JC. On the interaction of multiple overlay routing. *Performance Evaluation* 2005; **62**(14):229 – 246, doi:http://dx.doi.org/10.1016/j.peva.2005.07.005. URL http://www.sciencedirect.com/science/article/pii/S0166531605000842, performance 2005 24th International Symposium on Computer Performance, Modeling, Measurements and Evaluation.
11. Keralapura R, Taft N, Chuah CN, Iannaconne G. Can ISPs Take the Heat from Overlay Networks? 2004.
12. Laoutaris N, Smaragdakis G, Bestavros A, Byers J. Implications of Selfish Neighbor Selection in Overlay Networks. *Proceedings of IEEE Infocom*, 2007.

13. Seetharaman S, Ammar MH. On the interaction between dynamic routing in native and overlay layers. *INFOCOM*, IEEE, 2006. URL http://dblp.uni-trier.de/db/conf/infocom/infocom2006.html#SeetharamanA06.
14. Moustakas V, Akcan H, Roussopoulos M, Delis A. Alleviating the topology mismatch problem in distributed overlay networks: A survey. *Journal of Systems and Software* 2016; **113**:216–245.
15. Le H, Hong D, Simmonds A. A self-organising model for topology-aware overlay formation. *ICC*, 2005; 1566–1571.
16. Ng TSE, 0001 HZ. Global network positioning: a new approach to network distance prediction. *Computer Communication Review* ; **32**(1):61. URL http://dblp.uni-trier.de/db/journals/ccr/ccr32.html#NgZ02.
17. Garces Erice L, Ross KW, Biersack EW, Felber PA, Urvoy Keller G. Topology-centric look-up service. *NGC 2003, 5th International Workshop on Networked Group Communications, September 16-19, 2003, Munich, Germany*, Munich, GERMANY, 2003. URL http://www.eurecom.fr/publication/1205.
18. Yu J, Li M. CBT: A proximity-aware peer clustering system in large-scale BitTorrent-like peer-to-peer networks. *Comput. Commun.* Feb 2008; **31**(3):591–602, doi:10.1016/j.comcom.2007.08.020. URL http://dx.doi.org/10.1016/j.comcom.2007.08.020.
19. Jacobson V. pathchar. http://www.caida.org/tools/utilities/others/pathchar/. Accessed August 7, 2014.
20. Francis P, Jamin S, Jin C, Jin Y, Raz D, Shavitt Y, Zhang L. Idmaps: A global internet host distance estimation service. IEEE Press: Piscataway, NJ, USA, 2001; 525–540, doi:10.1109/90.958323. URL http://dx.doi.org/10.1109/90.958323.
21. Zhang XY, Zhang Q, Zhang Z, Song G, Zhu W. A construction of locality-aware overlay network: moverlay and its performance. *IEEE J.Sel. A. Commun.* Sep 2006; **22**(1):18–28, doi:10.1109/JSAC.2003.818780. URL http://dx.doi.org/10.1109/JSAC.2003.818780.
22. Rahimian F, Le Nguyen Huu T, Girdzijauskas S. Locality-awareness in a peer-to-peer publish/subscribe network. *Proceedings of the 12th IFIP WG 6.1 International Conference on Distributed Applications and Interoperable Systems*, DAIS'12, Springer-Verlag: Berlin, Heidelberg, 2012; 45–58, doi:10.1007/978-3-642-30823-9_4. URL http://dx.doi.org/10.1007/978-3-642-30823-9_4.
23. Saghiri AM, Meybodi MR. A distributed adaptive landmark clustering algorithm based on moverlay and learning automata for topology mismatch problem in unstructured peer-to-peer networks. *International Journal of Communication Systems* 2017; **30**(3).
24. Demirci S, Yardimci A, Sayit M, Tunali ET, Bulut H. A hierarchical p2p clustering framework for video streaming systems. *Computer Standards & Interfaces* 2017; **49**:44–58.
25. Shen H, Liu G, Ward L. A proximity-aware interest-clustered p2p file sharing system. *IEEE transactions on parallel and distributed systems* 2015; **26**(6):1509–1523.
26. Dabek F, Cox R, Kaashoek F, Morris R. Vivaldi: a decentralized network coordinate system. *SIGCOMM Comput. Commun. Rev.* Aug 2004; **34**(4):15–26, doi:10.1145/1030194.1015471. URL http://doi.acm.org/10.1145/1030194.1015471.
27. Elmokashfi A, Kleis M, Popescu A. Netforecast: A delay prediction scheme for provider controlled networks. *GLOBECOM*, IEEE, 2007; 502–507. URL http://dblp.uni-trier.de/db/conf/globecom/globecom2007.html#ElmokashfiKP07.
28. Wang G, Zhang C, Qiu X, Zeng Z. Replacing network coordinate system with internet delay matrix service (idms): A case study in chinese internet. *CoRR* 2013; **abs/1307.0349**.
29. Wong B, Slivkins A, Sirer EG. Meridian: A lightweight network location service without virtual coordinates. ACM: New York, NY, USA, 2005; 85–96, doi:10.1145/1090191.1080103. URL http://doi.acm.org/10.1145/1090191.1080103.
30. Donnet B, Gueye B, Kafar MA. A survey on network coordinates systems, design, and security. *IEEE Communications Surveys and Tutorials* 2010; **12**(4):488–503. URL http://dblp.uni-trier.de/db/journals/comsur/comsur12.html#DonnetGK10.
31. Ng TSE, Zhang H. Towards global network positioning. *Proceedings of the 1st ACM SIGCOMM Workshop on Internet Measurement*, IMW '01, ACM: New York, NY, USA, 2001; 25–29, doi:10.1145/505202.505206. URL http://doi.acm.org/10.1145/505202.505206.
32. Lim H, Hou JC, Choi CH. Constructing internet coordinate system based on delay measurement. IEEE Press: Piscataway, NJ, USA, 2005; 513–525, doi:10.1109/TNET.2005.850197. URL http://dx.doi.org/10.1109/TNET.2005.850197.
33. Waldvogel M, Rinaldi R. Efficient topology-aware overlay network. *In Hotnets-I*, 2002.
34. Choffnes DR, Sanchez M, Bustamante FE. Network positioning from the edge - an empirical study of the effectiveness of network positioning in p2p systems. *INFOCOM*, IEEE, 2010; 291–295. URL http://dblp.uni-trier.de/db/conf/infocom/infocom2010.html#ChoffnesSB10.
35. Ng TSE, 0001 HZ. A network positioning system for the internet. *USENIX Annual Technical Conference, General Track*, USENIX, 2004; 141–154. URL http://dblp.uni-trier.de/db/conf/usenix/usenix2004g.html#NgZ04.
36. Ng TSE, Chu YH, Rao SG, Sripanidkulchai K, 0001 HZ. Measurement-based optimization techniques for bandwidth-demanding peer-to-peer systems. *INFOCOM*, 2003. URL http://dblp.uni-trier.de/db/conf/infocom/infocom2003.html#NgCRSZ03.
37. Ng TSE, 0001 HZ. Predicting internet network distance with coordinates-based approaches. *INFOCOM*, 2002. URL http://dblp.uni-trier.de/db/conf/infocom/infocom2002.html#NgZ02.
38. Ratnasamy S, Handley M, Karp R, Shenker S. Topologically-aware overlay construction and server selection. *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 3, 2002; 1190– 1199 vol., doi:10.1109/INFCOM.2002.1019369. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1019369.

39. Le H, Hong D, Simmonds A. Geo-lpm- an efficient scheme for locating nodes in the internet 2005.
40. Kwon M, Fahmy S. Topology-aware overlay networks for group communication. *Proceedings of the 12th international workshop on Network and operating systems support for digital audio and video*, NOSSDAV '02, ACM: New York, NY, USA, 2002; 127–136, doi:10.1145/507670.507688. URL http://doi.acm.org/10.1145/507670.507688.
41. Kwon M, Fahmy S. Path-aware overlay multicast. *Comput. Netw.* Jan 2005; **47**(1):23–45, doi:10.1016/j.comnet.2004.06.025. URL http://dx.doi.org/10.1016/j.comnet.2004.06.025.
42. Cui J, He Y, Wu L, Xiong N, Jin H, Yang LT. Multi-domain topology-aware grouping for application-layer multicast. *Proceedings of the Third international conference on High Performance Computing and Communications*, HPCC'07, Springer-Verlag: Berlin, Heidelberg, 2007; 623–633. URL http://dl.acm.org/citation.cfm?id=2401945.2402015.
43. Cui J, He Y, Wu L. More efficient mechanism of topology-aware overlay construction in application-layer multicast. *IEEE NAS*, IEEE Computer Society, 2007; 31–36. URL http://dblp.uni-trier.de/db/conf/nas/nas2007.html#CuiHW07.
44. Zeinalipour-Yazti D, Kalogeraki V. Structuring topologically aware overlay networks using domain names. *Comput. Netw.* Nov 2006; **50**(16):3064–3082, doi:10.1016/j.comnet.2005.12.003. URL http://dx.doi.org/10.1016/j.comnet.2005.12.003.
45. Krishnamurthy B, Wang J. Topology modeling via cluster graphs. *Proceedings of the 1st ACM SIGCOMM Workshop on Internet Measurement*, IMW '01, ACM: New York, NY, USA, 2001; 19–23, doi:10.1145/505202.505205. URL http://doi.acm.org/10.1145/505202.505205.
46. Krishnamurthy B, Wang J. On network-aware clustering of web clients. *Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, SIGCOMM '00, ACM: New York, NY, USA, 2000; 97–110, doi:10.1145/347059.347412. URL http://doi.acm.org/10.1145/347059.347412.
47. Krishnamurthy B, Wang J, Xie Y. Early measurements of a cluster-based architecture for p2p systems. *Internet Measurement Workshop*, Paxson V (ed.), ACM, 2001; 105–109. URL http://dblp.uni-trier.de/db/conf/imw/imw2001.html#KrishnamurthyWX01.
48. Andersen DG, Feamster N, Bauer S, Balakrishnan H. Topology inference from bgp routing dynamics. *Proceedings of the 2nd ACM SIGCOMM Workshop on Internet measurment*, IMW '02, ACM: New York, NY, USA, 2002; 243–248, doi:10.1145/637201.637239. URL http://doi.acm.org/10.1145/637201.637239.
49. Karwaczynski P, Mocnik J. Ip-based clustering technique for structured p2p overlays 2007.
50. Eom H, Wolinsky DI, Figueiredo RJO. Solare: Self-organizing latency-aware resource ensemble. *HPCC*, Thulasiraman P, Yang LT, Pan Q, Liu X, Chen YC, Huang YP, Chang LH, Hung CL, Lee CR, Shi JY, *et al.* (eds.), IEEE, 2011; 229–236. URL http://dblp.uni-trier.de/db/conf/hpcc/hpcc2011.html#EomWF11.
51. Kova!cevi A, Liebau N, Steinmetz R. Globase.kom - a p2p overlay for fully retrievable location-based search. *Proceedings of the Seventh IEEE International Conference on Peer-to-Peer Computing*, P2P '07, IEEE Computer Society: Washington, DC, USA, 2007; 87–96. URL http://dl.acm.org/citation.cfm?id=1306874.1307182.
52. Picone M, Amoretti M, Zanichelli F. Proactive neighbor localization based on distributed geographic table. *MoMM*, Kotsis G, Taniar D, Pardede E, Awan I, Saleh I, Ibrahim IK (eds.), ACM, 2010; 305–312. URL http://dblp.uni-trier.de/db/conf/momm/momm2010.html#PiconeAZ10.
53. Schollmeier R, Kunzmann G. Gnuviz - mapping the gnutella network to its geographical locations. *Praxis der Informationsverarbeitung und Kommunikation* 2003; **26**(2):74–79. URL http://dblp.uni-trier.de/db/journals/pik/pik26.html#SchollmeierK03.
54. Padmanabhan VN, Subramanian L. An investigation of geographic mapping techniques for internet hosts. *SIGCOMM*, 2001; 173–185. URL http://dblp.uni-trier.de/db/conf/sigcomm/sigcomm2001.html#PadmanabhanS01.
55. Cikryt C. Beyond music filesharing: A technical introduction to p2p networks. Seminararbeit, Freie Universitt Berlin 2 2010.
56. Liu Y, Liu X, Xiao L, Ni LM, 0001 XZ. Location-aware topology matching in p2p systems. *INFOCOM*, 2004. URL http://dblp.uni-trier.de/db/conf/infocom/infocom2004.html#LiuLXNZ04.
57. Castro M, Druschel P, Hu YC, Rowstron A. Future directions in distributed computing. chap. Topology-aware Routing in Structured Peer-to-peer Overlay Networks, Springer-Verlag: Berlin, Heidelberg, 2003; 103–107. URL http://dl.acm.org/citation.cfm?id=1809315.1809337.
58. Andersen D, Balakrishnan H, Kaashoek F, Morris R. Resilient overlay networks. *Proceedings of the eighteenth ACM symposium on Operating systems principles*, SOSP '01, ACM: New York, NY, USA, 2001; 131–145, doi:10.1145/502034.502048. URL http://doi.acm.org/10.1145/502034.502048.
59. https://courses.engr.illinois.edu/cs525/sp2011/review.0210.2011.nospam.txt.
60. Zhao BY, Huang L, Stribling J, Joseph AD, Kubiatowicz JD. Exploiting routing redundancy via structured peer-to-peer overlays. *Proceedings of the 11th IEEE International Conference on Network Protocols*, ICNP '03, IEEE Computer Society: Washington, DC, USA, 2003; 246–. URL http://dl.acm.org/citation.cfm?id=951950.952229.
61. Tao S, Xu K, Xu Y, Fei T, Gao L, Guerin R, Kurose J, Towsley D, Zhang ZL. Exploring the performance benefits of end-to-end path switching. *SIGMETRICS Perform. Eval. Rev.* Jun 2004; **32**(1):418–419, doi:10.1145/1012888.1005746. URL http://doi.acm.org/10.1145/1012888.1005746.
62. Tao S, Xu K, Estepa A, Fei T, Gao L, Guérin R, Kurose JF, Towsley DF, Zhang ZL. Improving voip quality through path switching. *INFOCOM*, 2005; 2268–2278.
63. Fei T, Tao S, Gao L, Guérin R. How to select a good alternate path in large peer-to-peer systems? *INFOCOM*, 2006.
64. Cha M, Moon SB, Park CD, Shaikh A. Placing Relay Nodes for Intra-Domain Path Diversity. *Technical Report* 2006.

65. Akella A, Pang J, Maggs B, Seshan S, Shaikh A. A comparison of overlay routing and multihoming route control. *SIGCOMM Comput. Commun. Rev.* Aug 2004; **34**(4):93–106, doi:10.1145/1030194.1015479. URL http://doi.acm.org/10.1145/1030194.1015479.
66. Han J, Jahanian F. Impact of path diversity on multi-homed and overlay networks. *DSN*, IEEE Computer Society, 2004; 29–. URL http://dblp.uni-trier.de/db/conf/dsn/dsn2004.html#HanJ04.
67. Cui W, Stoica I, Katz RH. Backup path allocation based on a correlated link failure probability model in overlay networks. *Proceedings of the 10th IEEE International Conference on Network Protocols*, ICNP '02, IEEE Computer Society: Washington, DC, USA, 2002; 236–. URL http://dl.acm.org/citation.cfm?id=645532.656185.
68. Andersen DG, Snoeren AC, Balakrishnan H. Best-path vs. multi-path overlay routing. *Proceedings of the 3rd ACM SIGCOMM Conference on Internet Measurement*, IMC '03, ACM: New York, NY, USA, 2003; 91–100, doi:10.1145/948205.948218. URL http://doi.acm.org/10.1145/948205.948218.
69. Han J, Watson D, Jahanian F. Topology aware overlay networks. *INFOCOM*, 2005; 2554–2565.
70. Han J, Watson D, Jahanian F. Enhancing end-to-end availability and performance via topology-aware overlay networks. *Comput. Netw.* Nov 2008; **52**(16):3029–3046, doi:10.1016/j.comnet.2008.06.019. URL http://dx.doi.org/10.1016/j.comnet.2008.06.019.
71. Sadia Saleem HI, Welz M. Fewest common hops (FCH): An improved peer selection approach for p2p applications February 2013; :449 – 453URL http://heim.ifi.uio.no/michawe/research/publications/.
72. Karayer E, Sayit M. A path selection approach with genetic algorithm for p2p video streaming systems. *Multimedia Tools and Applications* Dec 2016; **75**(23):16 039–16 057, doi:10.1007/s11042-015-2912-y. URL https://doi.org/10.1007/s11042-015-2912-y.
73. Chen Y, Bindel D, Katz RH. Tomography-based overlay network monitoring. *Proc. ICM'03*, 2003, doi:10.1145/948205.948233.
74. Chen Y, Bindel D, Song H, Katz RH. An algebraic approach to practical and scalable overlay network monitoring. *SIGCOMM*, Yavatkar R, Zegura EW, Rexford J (eds.), ACM, 2004; 55–66. URL http://dblp.uni-trier.de/db/conf/sigcomm/sigcomm2004.html#ChenBSK04.
75. Zhao Y, Chen Y, Bindel D. Towards deterministic network diagnosis. *Proceedings of the Joint International Conference on Measurement and Modeling of Computer Systems*, SIGMETRICS '06/Performance '06, ACM: New York, NY, USA, 2006; 387–388, doi:10.1145/1140277.1140333. URL http://doi.acm.org/10.1145/1140277.1140333.
76. Chen Y, Bindel D, Song HH, Katz RH. Algebra-based scalable overlay network monitoring: algorithms, evaluation, and applications. *IEEE/ACM Trans. Netw.* 2007; **15**(5):1084–1097. URL http://dblp.uni-trier.de/db/journals/ton/ton15.html#ChenBSK07.
77. Liu Y, Xiao L, Liu X, Ni LM, Zhang X. Location awareness in unstructured peer-to-peer systems. *IEEE Trans. Parallel Distrib. Syst.* Feb 2005; **16**(2):163–174, doi:10.1109/TPDS.2005.21. URL http://dx.doi.org/10.1109/TPDS.2005.21.
78. Akiyama T, Kawai Y, Iida K, Zhang J, Shiraishi Y. Proposal for a new generation sdn-aware pub/sub environment. *ICN 2014, The Thirteenth International Conference on Networks*, 2014; 210–214.
79. Liu Y. A two-hop solution to solving topology mismatch. *IEEE Transactions on Parallel and Distributed Systems* 2008; **19**(11):1591–1600.
80. Ren S, Guo L, Jiang S, Zhang X. Sat-match: a self-adaptive topology matching method to achieve low lookup latency in structured p2p overlay networks. *Parallel and Distributed Processing Symposium, 2004. Proceedings. 18th International*, IEEE, 2004; 83.
81. Gupta A, Liskov B, Rodrigues R, *et al.*. One hop lookups for peer-to-peer overlays. *HotOS*, 2003; 7–12.
82. Al Mojamed M, Kolberg M. Onehopmanet: One-hop structured p2p over mobile ad hoc networks. *Next Generation Mobile Apps, Services and Technologies (NGMAST), 2014 Eighth International Conference on*, IEEE, 2014; 159–163.
83. Nakao A, Peterson L, Bavier A. A routing underlay for overlay networks. *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, SIGCOMM '03, ACM: New York, NY, USA, 2003; 11–18, doi:10.1145/863955.863958. URL http://doi.acm.org/10.1145/863955.863958.
84. Krishna Balan R, Ebling M, Castro P, Misra A. Matrix: adaptive middleware for distributed multiplayer games. *Proceedings of the ACM/IFIP/USENIX 2005 International Conference on Middleware*, Middleware '05, Springer-Verlag New York, Inc.: New York, NY, USA, 2005; 390–400. URL http://dl.acm.org/citation.cfm?id=1515890.1515910.
85. Xie L, Hutchison D, Smith P. Performance enhancement via two-layer support for peer-to-peer systems using active networking. *ISADS*, 2005; 695–700.
86. Brocco A, Hirsbrunner B, Courant M. A modular middleware for high-level dynamic network management. *Proceedings of the 1st workshop on Middleware-application interaction: in conjunction with Euro-Sys 2007*, MAI '07, ACM: New York, NY, USA, 2007; 21–24, doi:10.1145/1238828.1238834. URL http://doi.acm.org/10.1145/1238828.1238834.
87. Babaoglu O, Meling H, Montresor A. Anthill: A framework for the development of agent-based peer-to-peer systems. *Proceedings of the 22 nd International Conference on Distributed Computing Systems (ICDCS'02)*, ICDCS '02, IEEE Computer Society: Washington, DC, USA, 2002; 15–. URL http://dl.acm.org/citation.cfm?id=850928.851860.
88. Karagiannis T, Rodriguez P, Papagiannaki K. Should internet service providers fear peer-assisted content distribution? *Proceedings of the 5th ACM SIGCOMM Conference on Internet Measurement*, IMC '05, USENIX Association: Berkeley, CA, USA, 2005; 6–6. URL http://dl.acm.org/citation.cfm?id=1251086.1251092.

89. Dischinger M, Mislove A, Haeberlen A, Gummadi KP. Detecting bittorrent blocking. *Proceedings of the 8th ACM SIGCOMM Conference on Internet Measurement*, IMC '08, ACM: New York, NY, USA, 2008; 3–8, doi: 10.1145/1452520.1452523. URL http://doi.acm.org/10.1145/1452520.1452523.
90. Lehrieder F, Dán G, Hoßfeld T, Oechsner S, Singeorzan V. The impact of caching on bittorrent-like peer-to-peer systems. *10th IEEE International Conference on Peer-to-Peer Computing 2010 - IEEE P2P 2010*, Best Paper Award, Delft, the Netherlands, 2010. URL http://dblp.uni-trier.de/db/conf/p2p/p2p2010.html#LehriederDHOS10.
91. CacheLogic: advanced solutions for P2P networks. http://www.cachelogic.com/index.php Jan 2007.
92. Saleh O, Hefeeda M. Modeling and Caching of Peer-to-Peer Traffic. *ICNP '06: Proceedings of the Proceedings of the 2006 IEEE International Conference on Network Protocols*, IEEE Computer Society: Washington, DC, USA, 2006; 249–258, doi:10.1109/icnp.2006.320218. URL http://dx.doi.org/10.1109/icnp.2006.320218.
93. Bindal R, Cao P, Chan W, Medved J, Suwala G, Bates T, Zhang A. Improving traffic locality in bittorrent via biased neighbor selection. *Proceedings of the 26th IEEE International Conference on Distributed Computing Systems*, ICDCS '06, IEEE Computer Society: Washington, DC, USA, 2006; 66–, doi:10.1109/ICDCS.2006.48. URL http://dx.doi.org/10.1109/ICDCS.2006.48.
94. Oechsner S, Lehrieder F, Hoßfeld T, Metzger F, Staehle D, Pussep K. Performance study of locality-aware peer selection algorithms ; .
95. Seedorf J, Kiesel S, Stiemerling M. Traffic localization for p2p-applications: The alto approach. *Peer-to-Peer Computing*, Schulzrinne H, Aberer K, Datta A (eds.), IEEE, 2009; 171–177. URL http://dblp.uni-trier.de/db/conf/p2p/p2p2009.html#SeedorfKS09.
96. Xie H, Yang YR, Krishnamurthy A, Liu YG, Silberschatz A. P4P: provider portal for applications. *SIGCOMM 2008*, 2008.
97. Choffnes DR, Bustamante FE. Taming the torrent: A practical approach to reducing cross-isp traffic in peer-to-peer systems. *Proceedings of the ACM SIGCOMM 2008 Conference on Data Communication*, SIGCOMM '08, ACM: New York, NY, USA, 2008; 363–374, doi:10.1145/1402958.1403000. URL http://doi.acm.org/10.1145/1402958.1403000.
98. Torres R, Mellia M, Munaf MM, Rao SG. Characterization of community based-p2p systems and implications for traffic localization. *Peer-to-Peer Networking and Applications* 2013; **6**(2):118–133. URL http://dblp.uni-trier.de/db/journals/ppna/ppna6.html#TorresMMR13.
99. Liu B, Cao Y, Cui Y, Lu Y, Xue Y. Locality Analysis of BitTorrent-Like Peer-to-Peer Systems. *Consumer Communications and Networking Conference (CCNC), 2010 7th IEEE*, 2010.