Peer-to-Peer Networking and Applications (2020) 13:905–931
https://doi.org/10.1007/s12083-019-00839-8

Next level peer-to-peer overlay networks under high churns: a survey

Ashika R. Naik[1] · Bettahally N. Keshavamurthy[1]

Received: 25 February 2019 / Accepted: 16 October 2019 / Published online: 21 December 2019

Abstract
Peer-to-Peer (P2P) technology has seen a remarkable progress due to its decentralized and distributed approach. A wide range of applications such as social networking, file sharing, long range interpersonal communication etc. are carried out with ease by employing P2P protocol candidates. There exists a huge span of such P2P protocols. In this paper, we review advanced protocols like ZeroNet, Dat, Ares Galaxy, Accordion etc. evolved from classic peer-to-peer (P2P) overlay networks. We utilize term classic to allude protocols like Chord, Pastry, Tapestry, Kademlia, BitTorrent, Gnutella, Gia, NICE etc. While coming to their design, several challenges existed with classic approach under high churn environment with growing network communication rate. To address these multifaceted network issues with classic P2P systems, novel approaches evolved which helped researchers to built new application layer networks on existing P2P networks. We contribute in this paper by systematically characterizing next-level P2P (NL P2P) and examining their key concepts. Arrangement of distributed networks is completed by numerous analysts, which incorporates classic P2P systems. In this work, we therefore aim to make a further stride by deliberately talking about protocols created from classic P2P systems, and their performance comparison in dynamically changing environment. Different aspects of P2P overlay frameworks like routing, security, query, adaptation to non-critical failure and so forth dependent on developed conventions are additionally examined. Further, based on our review and study we put forward some of the exploring challenges with NL P2P frameworks.

Keywords Overlay networks · Next-level P2P networks · Classic P2P networks · Re-decentralization · Network partitions

1 Introduction

As the technology is enhancing with rapid growth, with growing user demands and network complexity, overlay networking based approach was improved further with addition of new technologies. To facilitate effective file sharing and its resources, application-level overlays were employed using peer-to-peer (P2P) networks in a loose manner. Decentralization was comprehended for exchange of information from local source to a more confined liberal framework. In today's world, the current scenario of network virtualization is taken to an extent, where users can retrieve information with high speeds in different applications. Especially, with P2P class of overlay networks, wide variety of applications exists. Researchers have contributed significantly to efficient routing, communication, security, and various other functioning of such networks. Using P2P as base, designs of social networks are also simplified. Novel techniques in the field of P2P domain helps in retrieving information efficiently. Diverse applications needs are also fulfilled, with effectiveness in cost [1].

Routing in P2P networks gets more attractions in research communities, since better the routing protocol, better is the overall performance of the network. Also, resource discovery, load balancing and privacy maintenance are other technical issues focused upon. To ensure efficient traffic forwarding with best QoS (Quality of Service), networking path selection criteria need to be idealistic. Adrian has discussed overlay routing in his work [2]. The two types of routing protocols: reactive and proactive are employed in most of the networks. However, these work well under low churn, as in classic networks. But with increasing network complexities, churn rate (a rate at which peers join and leave the network) also increases rapidly, which calls for improving routing techniques in classic P2P protocols to be deployed in current generation networks.

✉ Ashika R. Naik
ashika3000@nitgoa.ac.in

Bettahally N. Keshavamurthy
bnkeshav.fcse@nitgoa.ac.in

[1] Department of Computer Science and Engineering, National Institute of Technology Goa, Farmagudi, India

Topology of network is also equally important; it should reduce the cost function of creating links as well as routing cost. Kamel discussed one such near-optimal technique for a given traffic matrix; formulating problem as ILP (Integer Linear Programming) [3]. For developing near optimal topology, greedy approach along with clustering, maximum hop number and traffic volume were used. Besides topology; research enhancement in the security of classic P2P network is also highly motivated in present world. To bridge the connectivity with enhanced security mechanism, urgent call for technological development in classic networks was raised. Additionally, classic P2P approach does not give dependably widespread network. An extensive part of the system lies behind firewalls. A significant and developing architecture of hosts are behind Network Address Translators (NATs), and intermediaries. Managing these functional issues is tedious, but essential to appropriation.

P2P networks implementation on realistic networks requires good network scalability, so that it can be employed on long term basis. Therefore, network design criteria invites for novel approaches which can target wide range of applications. Besides scalability, indexing multi-dimensional data is also of prime importance. Considering real world scenario, design complexity with classic P2P networks cannot be efficiently handled, since their performance degrades in such environment. The popular BitTorrent also fails in current web due to increasing demand in the security of files shared. The cutting edge web incorporates open and private spots for networks, without pitching information to publicists. Structured at first for research information, P2P expands on the current web while giving more control to users. Re-decentralizing enables clients to share specifically and build up new models for advanced cooperation.

In this paper, we center on next-level P2P (NL P2P) and current research in the field of P2P systems, examining how innovative developments rose advanced P2P systems from classic P2P systems. Different researchers have talked about P2P systems. The regular pattern of scientific categorization and characterization runs with ordering P2P into structured and unstructured systems, with further exchange of different systems in each, structured and unstructured classifications of P2P systems. We allude such systems as exemplary or classic, which incorporates commonplace conventions like Chord, Pastry, Tapestry, Kademlia, Gnutella, BitTorrent, Gia and so forth. In view of these exemplary systems, different headways has prompted further advancement of P2P systems, which we allude to them as next-level peer-to-peer (NL P2P) systems, since they are developed and built on existing P2P systems.

Lua and Crowcroft [4] has discussed various classic P2P networks, with respect to their architecture, lookup protocols, system parameters, routing performance, routing state, security, reliability etc. Structured P2P networks like CAN, Chord, Pastry, Tapestry, Kademlia, Viceroy and unstructured networks like Freenet, Gnutella, FastTrack/Kazaa, BitTorrent were considered for P2P network comparison. A similar P2P based approach was also taken by Malatras [5] for pervasive computing environments, in which additional classic P2P networks like P-Grid, Skip-Net, UMM, Gia, Phenix were discussed. Classic P2P networks serves as base for peer to peer overlay networks. But with network intricacy, advanced protocols have been introduced and are built on classic networks to meet the competing demands of current generation networks. These are lacking in the previous survey of P2P networks, so we examine and discuss these NL networks in this paper, to help to get good insight into such networks and protocols.

Considering standard trend of classic P2P network characterization, we audit NL P2P also into two classifications: structured and unstructured P2P.

- Firstly we examine issues with classic P2P networks, which motivated the evolution of NL P2P systems.
- We examine, why in-spite of having powerful support and base, they are not completely solid in specific applications.
- We also discuss why Distributed Hash Table (DHT) based approach in structured P2P is not fully suitable in the present overlay systems.
- We bring together cutting edge P2P networking protocols, examining their key concepts
- Additionally, elimination of network partitions like problems by NL P2P is also presented.

The paper then examines NL P2P in subsequent sections. In structured, we talk about NL P2P dependent on four well known classic P2P viz. Chord, Pastry, Kademlia and CAN. In light of Chord ring, next P2P level conventions are EpiChord, Accordion and Koorde. In light of Pastry, are Bamboo, SCRIBE and SimMud. Further, we likewise considered Broose and Overnet dependent on Kademlia and also Meghdoot, a CAN based protocol. In unstructured systems, classic BitTorrent based ZeroNet, Dat, Tribler; Gnutella based Ares Galaxy; IPFS based Filecoin are presented in detail. Figure 1 demonstrates rundown of some mainstream classic and their NL P2P systems. NL P2P systems like EpiChord, Accordion, SCRIBE, Overnet, ZeroNet, Filecoin and so on are talked about in this paper regarding their query, routing, adaptation to non-critical failure, security and so forth. Some NL P2P protocols employ dual or multi classic protocols. Example is Shareaza which uses Gnutella, Gnutella2 in addition to classic BitTorrent.

It is to be noted here that there exist vast number of classic P2P protocols other than those considered in this paper. For instance Viceroy, Tapestry, Skip-Net, Gia etc. [4, 5]. However, in the survey we considered those classic P2P which serves as base to NL P2P and are also most commonly used. In addition

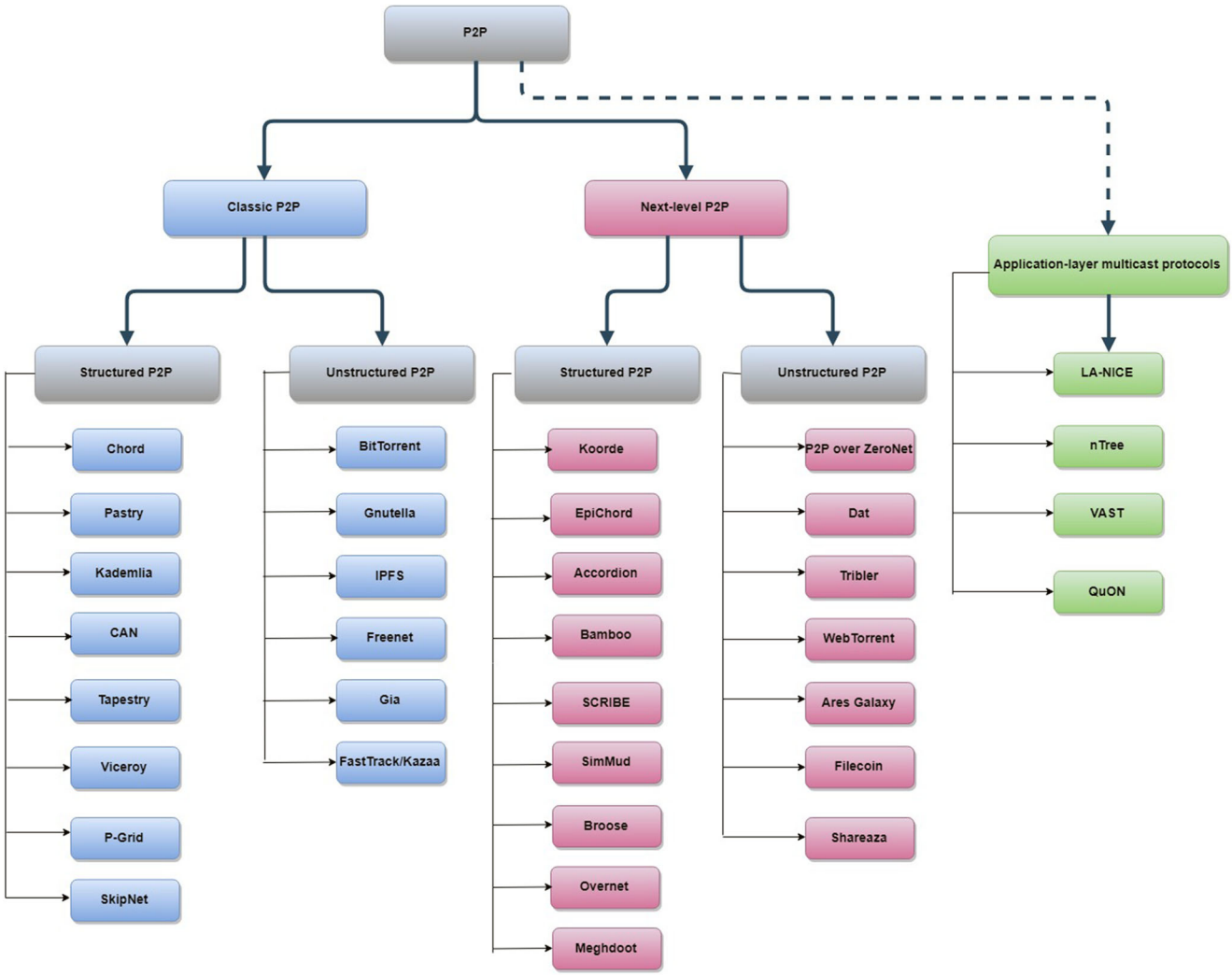

Fig. 1 Classic and NL P2P networks

to this, we carry our discussion further with application-layer multi-cast protocols. The various existing application-layer multi-cast and their successors are discussed with respect to their bandwidth of routing path, end-to-end delays, neighbor selection mechanism etc.

2 Classic peer-to-peer networks basics

Classic P2P networks of overlays are important and have seen a remarkable progress as it eliminates centralized approach based on traditional client-server model. Further, reliability issues can also be tackled using such networks. The present generation networks are built using network virtualization concept considering P2P as the base. Resource location of both peer nodes and data needs efficient protocols which can handle resource discovery in P2P networks. In this section, we discuss various technical research issues with existing classic P2P networks.

2.1 Technical characteristics and associated deficiency

Foundation DHT is the base of classic structured P2P networks. Besides offering good scalability, it also supports majority of structured P2P for their routing between nodes with bound on number of hops [6]. It helps in mapping identifiers set to node set, which helps in node location in well organized manner. DHT based approach offers major merits such as good scalability, very high availability, low-latency, and high threshold. However, it fails when performing under high churn. This is the major drawback of DHT based approaches for being not suitable in networks with their growing complexities. Besides this, it also suffers from various common security attacks like Sybil attacks, DDoS attacks etc. The weak resistance against security attacks further affects its peer discovery process as intruder interrupts the communication.

Lookup In most structured P2P (for e.g. Chord, Kademlia, P-Grid [5], Skip-Net), O (log N) numbers of peers are required

to be contacted for storing, retrieving or searching data. In classic unstructured P2P, the search algorithms are based on flooding and random walk [5]. Dorrigiv, Pralat [7] illustrated that flooding works out at its good, when handling lesser number of messages, but with large number of messages present, its outcome significantly degrades, as in classic P2P. The performance had been tested by them for different graphs and topologies. It was found that for clustered topologies, flooding is less effective.

Performance under churn Under high churn, where both network dynamicity and complexity increases twofold, classic P2P networks are not able to perform well with their existing routing mechanism and lookup. Some of the reasons are:

- Requirements of extensive updates and refreshes in the routing tables.
- Poor mechanism for peer discovery.

For instance, the pastry protocol is found to be less reliable under medium and high churn. Network recovery ability also reduces with its reactive approach and short session time. This may also affect peer connectivity resulting into network splitting. Similarly, the chord too fails under high churn with large lookup latencies [6]. The various file-sharing applications are mostly obtained from unstructured P2P networks. Search techniques in such networks are random with flooding mechanism. This is highly unsuitable for very large network size, as peer needs to query all the nodes in the network and also result in traffic overhead. Further, to decrease traffic overhead due to flooding, additional techniques needs to be incorporated in classic networks, which not only degrades performance but also increases further complexities.

2.2 Routing mechanism

Iterative This type of routing follows the communication of nodes with their source only. When searching for a specific key, requesting peer P1 sends a demand to its finger which lies close to the predecessor of the key ID. This peer restores its nearest finger before continuing with further iterations. Subsequent to getting this data, P1 sends the demand to the next new peer which is closer to the ideal key and at that point, sits tight for the appropriate response. This process proceeds iteratively with the peers lying on its path until the peer that lies before the key ID is located. When successor of key is reached, it realizes that key is found, and the immediate predecessor holds the required key. The initiator node P1 likewise screens the entire routing path by utilizing a timer for each hop to decide whether the queried hub is missing or the bundle has been lost and another packet must be sent. Iterative routing offers primary points of interest: First, initiator node P1 can monitor the query course without much of a stretch and can respond to issues such as wrong routing information quickly. Also, when the query flops because of a missing peer, P1 is soon mindful of the disappointment and presumably proceed with the query alongside the missing peer [8]

Recursive Unlike iterative, in recursive routing peer P1 requesting for a key simply forwards the query to the next peer P2. P2 then forwards query to next peer say P3 and so on until the point at which the demand achieves the key's predecessor. At that point, peer at last returns the key's successor to P1. This seems to be good concerning hop delay since P1 don't have to wait for acknowledgement from each subsequent peer. However, varying finger table sections (under high churn) are the principle issue with recursive routing. Expecting a shared system where peers join and leave frequently, fingers likewise gets updated all the time. As finger sections are not refreshed promptly, the likelihood of utilizing a finger that is never again taking an interest in the system is noteworthy. These outcomes in the loss of query packets and they cannot be conveyed to the missing peer. Once again, the initiator peer P1 screens the query request with a timer. While each hop can be checked independently when utilizing iterative routing, a peer gets no data about the directing advancement when recursive routing is utilized. This is the reason the timer must be picked perceptibly more. Likewise, it becomes more difficult to discover missing peers. Queries consequently flop all the time, and are not seen until the global time expired [8]

Such routing is good but offers latency along with irregular query delivery. For a network to be robust, it should therefore eliminate the majority of its technical deficiencies. Figure 2 lists significant features of a P2P robust network. Besides lookup/ peer discovery, foundation, fault-tolerance; handling churns along with good ability of handling network partitions like parameters are also important. These will be discussed in subsequent sections with respect to NL P2P. We'll see these features are more accountable in NL P2P. Next, before discussing NL P2P, we briefly go through technical concept of their existing classic P2P base.

2.3 Some classic P2P networks that serves as base for NL peer-to-peer networks

BitTorrent is unstructured P2P network. It is popular as a file-sharing tool. This distributed P2P network aims for easy file distribution with less bandwidth consumption. Large number of files thus can be downloaded using BitTorrent. The file splitting mechanism of BitTorrent allows users for simultaneous access of files, with high speed. Its open-source nature has further attracted large users and research communities. As far as resource allocation is concerned, no centralized method exists with it [9]. However, it did not possess strong base as far security is concerned. To tackle this issue, P2P were

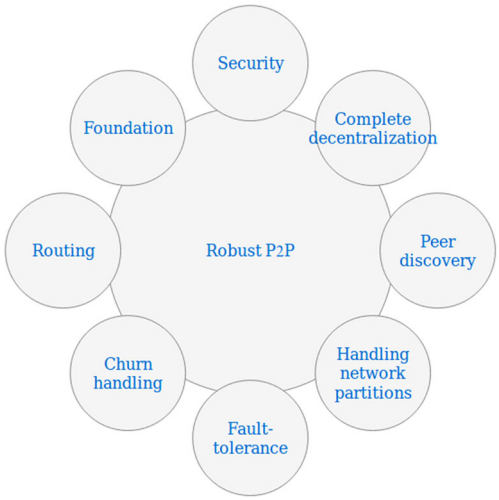

Fig. 2 Robust P2P and its features

developed further considering BitTorrent as base. One such research led to evolution of P2P-over-ZeroNet networks which incorporates security features (https://zeronet.readthedocs.io/en/latest/).

Gnutella This open source P2P was among the first decentralized approach in classic P2P. It is popular as file-sharing tool with P2P search protocol. Parallel fashion working Gnutella clients maintains its development. Peers form an application-level Gnutella network, by running software compatible with Gnutella protocol. It follows a multi-modal distribution, incorporating both quasi-constant and power law distribution [10]. This allows network reliability of Gnutella network. Although this feature also make it robust to malicious attacks, but when considering large network, its security was not powerful enough.

InterPlanetary file system (IPFS) IPFS is a content-addressable P2P network, designed for hypermedia storage and sharing purpose in distributed file system. It is an open-source project with contributions from over different communities. This P2P approach connects all computing devices having same system files. In 2014, IPFS added features of Bitcoin to improve its performance. Due to this, storing unalterable data, removing redundant contents and address information access was possible [11]. Although IPFS uses DHTs, we discussed it in unstructured P2P networks since it employs unstructured BitTorrent protocol.

Chord It is fully distributed protocol, in which all nodes have equal importance. They are robust and are employed in loosely organized P2P networks. It achieves load balancing through consistent hashing and uses periodic stabilization when new node adds to the network to restore its balance. Without any parameter tuning, it is found to scale with logarithmic lookup, O (Log N), where N being the network size. It works in both iterative and recursive manner. Its protocol consists of query, finger table, peer joining mechanism and stabilization procedure. O (Log N) nodes are reached with huge probability using Chord [12]. However, it is seen to have less resistant to high churns [13].

Pastry It is also based on DHT approach like Chord. For successful joining of nodes, IP addresses of nodes in the network are used. Dynamic routing tables are then used for further communication between nodes. Through network locality, it minimizes the distance for quick message delivery. Like most P2P, it is self-organizing with mechanism for handling failure of nodes [5, 14]. It serves as good base for NL P2P like Bamboo, SCRIBE, and SimMud. Although Pastry offers good scalability, but due to its dependency on DHT, its flexibility in maintaining routing overheads reduces when operating under high churns. It tends to further increase network complexity in non-static environments.

Kademlia DHT based Kademlia locates nodes by employing XOR dependent metric. It utilizes asynchronous parallel queries to solve timeout delay issues. Peers in Kademlia use keys lying in 160-bit key space [15]. Due to its XOR based approach, lookup queries are received from uniform distribution of nodes. It provides good consistency and reliable performance compared to other classic P2P networks. However, in complex nature with growing users, existing networks do not meet required availability. Nonetheless, Kademlia serves as good base for Overnet like proprietary protocol.

CAN Is a distributed hash table based structure P2P network with good scalability features, along with fault-tolerance and self-organizing ability. It uses d-torus topology with virtual Cartesian space of d -dimension. The space is partitioned with nodes positioned in the partitions. The space is also used for storing a pair (Key, Value). The CAN has average routing path of $((d/4)*(n^{1/d}))$ with node degree of 2. For node failure handling, takeover algorithm is used by CAN in which neighboring node takes in place of a failed node. This is although good, but it requires refreshes as the (Key, Value) pair is lost along with node failure [16]. Also, security and load balancing features are not strong in CAN [5].

2.4 Distributed hash tables (classic P2P) and De Bruijin graphs (NL P2P)

Exemplary systems use DHTs as the base for their structure. Using DHT, overlay topology for all framework sizes can be organized. DHT is observed to be less viable under the unbound condition. It is inclined to Sybil assaults, Eclipse assaults, directing and capacity attacks. This also gives call for further future attacks. The lookup request sent also gets interrupted, which results in unsuccessful delivery of queries

[17]. But in real time environment, to employ any protocol, its practical implementation demands for security uprightness. The overhead in the routing data maintenance has also been reported in the DHT based techniques. Network complexities tend to increase under highly non-static environments. Consequently, either DHT based ought to be included with incredible security highlights or altogether new methodology is to be utilized. De Bruijin was one such approach over DHT.

De Bruijn graphs have numerous qualities that settle on them an appropriate decision for the topology of an overlay organize. These incorporate consistent degree at every hub, logarithmic measurement and an exceedingly normal topology that licenses peers to make solid presumptions about the worldwide structure of the system [18]. In chart hypothesis, an n-dimensional De Bruijn diagram of m symbols is a coordinated diagram indicating overlays between successions of symbols. Linear De Bruijn Graph is a d-dimensional with unidirected nature. For a node set V, graph is represented in the form G = (V, E) where E is the edge set [19]. NL P2P like Koorde, Broose employs De Bruijin graphs as it improves self-stabilizing property of the network. De-Bruijin offers merits over DHTs, in that;

- Self-balancing feature provides adaptation to internal failure while keeping up consistent node degree.
- It improves self-stabilizing property of the network.
- Less overhead as compared to DHT, since it includes constant degree at every node [19].

3 NL P2P networks (NL P2P)

The advanced peer-to-peer networking is a powerful tool itself in overlay networks. The wide range of file-sharing and other applications has striving need for robust application layer architecture for their optimum operation under high churn. NL P2P is one such solution to fulfill these needs. In this section, we first considered NL P2P in unstructured topology followed by structured topology.

3.1 Networks advanced from basic unstructured P2P networks

With unstructured P2P approach, wide assortment of conventions is created. It has no structure to sort out its peers, which disposes the overhead and the need to keep the system structure. Distinctive kinds of conventions are created and kept up with further upgrades in their execution. While BitTorrent, Freenet, Gnutella are among the prominent hopefuls, IPFS additionally pulls in different research and modern networks. In this area, we discuss NL P2P and their associated features. Figure 3 shows unstructured NL P2P protocols and their classic P2P substrate. Table 1 shows the key characteristics of various NL P2Ps networks along with their classic counterparts. All networking protocols offer s pure P2P nature when peers communicate with each other. However, the peer discovery may not be direct peer to peer i.e. to locate peers a virtual third party involves (e.g. BitTorrent). Also, the decentralization behavior is not completely P2P. Further portion of this section discusses NL P2Ps based on some common classic P2P networks.

3.1.1 Based on BitTorrent

P2P over ZeroNet ZeroNet is a most current shared system which utilizes BitTorrent and Bitcoin innovation (https://medium.com/@zeronet/zeronet-bitcoin-crypto-based-p2p-web-393b5bc967e5). The open, pure P2P, censorship resistant network was released in 2015. Utilizing BitTorrent, ZeroNet system is utilized for sites distributing and editing purpose. The sites are in the form of *public Bitcoin addresses*. Peers publish onto sites by signing into them using their *private keys*. *Private keys* are generated securely using SHA512 hash techniques. After signing, users publish and modify their websites content. Visiting peers requires the public keys. Joining of a peer to a group pursues similar procedure from BitTorrent. The peers seeding the sites are identified and file downloads then follows. If peer updates it file content over site, then this is notified to all other peers interested in those sites. Real time updates are possible in ZeroNet through Websocket API. Special API called *ZeroFrame* is also used by ZeroNet for updates (https://zeronet.readthedocs.io/en/latest/site_development/zeroframe_api_reference/). This enables peers to get aware with newly modified file contents. ZeroNet also supports multiple users for sites. Visiting peers can also publish their contents over sites that they are visiting. For this, the site owner grant permissions to peers interested, after securely authenticating them. *BIP32* innovation is used for different sites to generate unique addresses and private keys.

P2P over ZeroNet uses the ZeroNet and is created with primary intension of tackling security issues related with Tor and I2P. The novel secure technique utilizes two layer P2P protocol. The peers are refreshed consequently and steadily. The real favorable position of such system lies in its anonymity attributes. Such systems additionally counteract tracking and forensics since it uses two layers P2P systems. Two layers of P2P utilizes onion router [20] of Tor, for choosing superhubs and ZeroNet's site structure. There are three primary modules in P2P over ZeroNet systems viz. local editor module, secure transfer module and remote receive module. In local module, user configurations, along with packet and other processing is carried out. Encryption of packets is likewise conveyed in local editor module. Secure transfer module is concerned to peers locating sites, which were initially visited

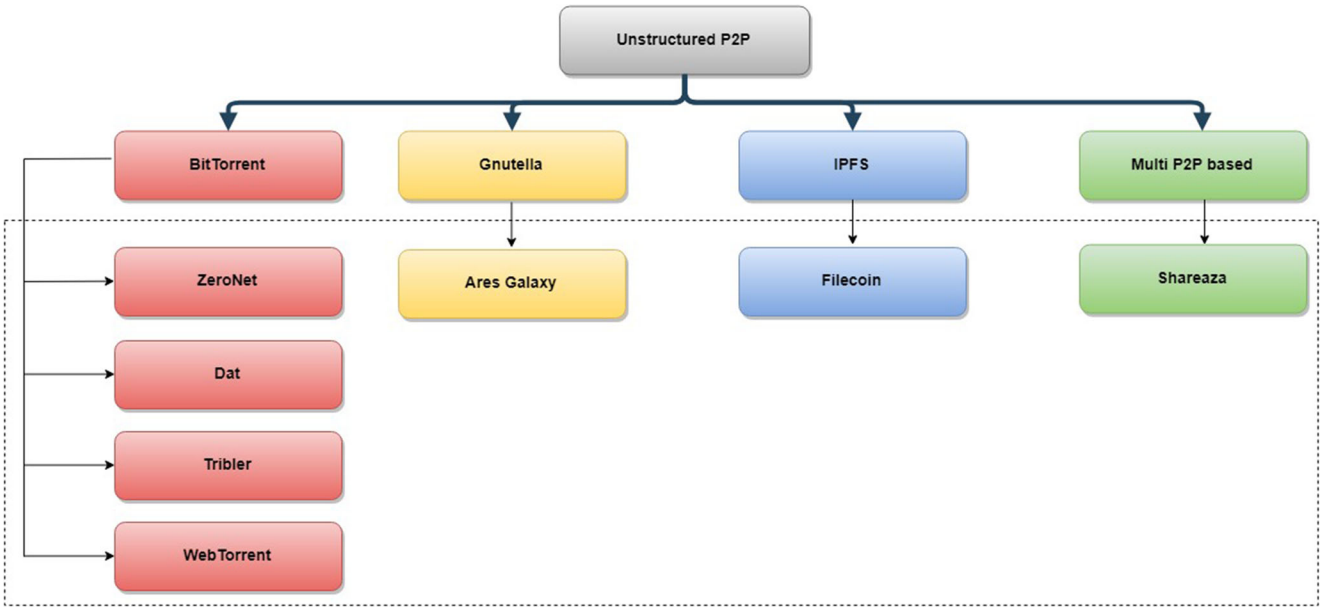

Fig. 3 Unstructured NL P2P networks and their classic substrates

by different users. This is accomplished through ZeroNet's decentralized P2P component. In the remote receive module, incoming messages are tuned in and processed from ZeroNet's peers. Message decryption also occurs in this module.

Encryption and authentication is achieved through security techniques i.e. hashing site contents and their locations (as used in Bitcoin). Also, with BitTorrent technology, once files are kept for users to download, update of files is not possible. This is solved in ZeroNet, which allows for data update option

Table 1 Decentralized nature of Unstructured classic and NL P2P networks

Unstructured P2P networks	Decentralization nature	Peers discovery	Peers communication
BitTorrent	Partial	censorship dependent; Uses *tor*	Direct
ZeroNet	Pure	Censorship resistant	Direct
Dat	Pure	Censorship resistant	Direct
Tribler	Pure	Censorship resistant	Direct
WebTorrent/WebRTC	Pure	Censorship resistant	Direct
Gnutella	Pure	Censorship resistant	Direct
Ares Galaxy	Pure	Censorship resistant	Direct
IPFS	Partial	Censorship resistant	Direct
Filecoin	Pure	Censorship resistant	Direct
Multi P2P based Shareaza	Pure	Censorship resistant	Direct

in real time environment. Rather than using IP addresses, BitTorrent based protocol uses public keys for identifying its peers (but using Bitcoin technology). However, private keys are used for encrypting the contents of BitTorrent distributed files https://bravenewcoin.com/insights/ZeroNet-expands-key-distributed-and-anonymous-features). It is ideal later on use since the ZeroNet site in probably not going to be closed down because of large number of users visiting it and anybody can act as host.

Dat-data distribution tool In view of exemplary Bittorrent, Dat protocol is developed with the intention to enhance file sharing uses of BitTorrent. It considers files whose information continually changes. Dat can hence be utilized as a powerful asset in applications like static websites hosting and also in developing host-less applications. This enables clients to download distributed updates from any companion in the system, as though they are straightforwardly downloading from source distributors. It uses *hypercore* feeds which are cryptographically hashed and signed binary appended streams (https://www.datprotocol.com/deps/0002-hypercore/). Merkel trees are utilized for *hypercore* feeds. Such trees are represented in the network and public keys are used for their identification. The datasets are also hashed for its identification. The *content* and *metadata* feeds are used which contains information about files such as size, name etc. The most common problem such as link rot is eliminated in DAT through eliminating HTTP shared datasets. When peer fetches contents over DAT network, DAT url in the form *dat://publickey/optionalsuffix* is required to be known. DAT peer uses *discovery peers* to locate other dat peers using Dat's *public key* (https://datprotocol.github.io/how-dat-works/). In addition, DAT protocol has (*post /login, logout, Get /account*) resources. These are used for creating fresh session, terminating ongoing session and to obtain information about current session respectively. When file data changes, Dat synchronizes different peers and replicates *content* and *metadata* feeds. While choosing between security in addition to speed and straightforwardness, Dat is the most reliable tool.

DAT decentralized 2018 report concentrated on Dat, for P2P decentralizing document sharing convention [21]. Dat influences intelligent data management and sharing under complex clustered networks. The developers trust that presenting decentralization at an level will permit existing storehouses (institutional information stores and others) to share data, making information less demanding to get, enhancing repetition, and shaping the premise of a helpfully run information conservation network. Dat uses *hyperdiscovery* which helps in replicating contents for a given Dat ARCHIEVE_KEY. For sharing files, Dat nodes first locate the file contents and then import them. This is achieved through *dat.importFiles()* and *dat.joinNetwork()* respectively (https://docs.datproject.org/dat-node).

Considering file updates, circulated data sets needs to have synchronization, since classic file sharing tools does not allow file updates without new data set redistribution. The key highlights and properties of Dat such as content integrity, decentralized mirroring, security of networks and efficient synchronization are significant with respect to Dat design [22]. Dat uses source discovery mechanisms for decentralized mirroring. The advantage of this is that network can be created where data can be discovered regardless of whether original information source vanishes.

Tribler - social file sharing P2P A social based P2P network, Tribler was developed as an extension to existing BitTorrent. The limitations such as partial decentralization, availability, security and network transparency issues in BitTorrent are addressed by Tribler. For file-sharing application, Tribler consists of modules like social networking module. The module carries out functions of storage and making available information about social groups. Tribler uses *Buddycast algorithm* for peer and content discovery [23] and uses permId (permanent identities) for user action identification. Such Ids are stored in the form of public or private key-pair, for signing every message. Peer communication is set with a BitTorrent swarm. Peers then communicate using BuddyCast protocol. For this, peer discovery is done by connecting them to super-peers. The search mechanism of Tribler is followed by its streaming, channels and reputation. TTL is set to one, allowing only neighbors to remote search. This reduces flooding process. Two streaming types viz. video-on-demand and live-streaming are supported. VOD differs from that in case of BitTorrent. In Tribler, first few pieces are downloaded first while playback commences.

Also, priorities are being allocated as high, low and mid. This helps pieces to be downloaded and play-backed based on their importance. High priorities ones are downloaded first followed by mid and low priority pieces. This ensures overall health of swarms. Moreover, Tribler supersede and replaces the default BitTorrent motivating force component (Tit-for-tat) with Give-to-Get. This motivating force component will rank peers as per their sending rank. A measurement portraying how well a peer is sending pieces to different peers. For live streaming, Tribler alters the real downpour document; since in live streaming pieces are not known in advance. For that, check plans are supplanted by open key determination of the first source in the deluge record. In this manner legitimacy of pieces are checked using only open keys. Channels in Tribler are implemented using *Dispersy*, which is a BuddyCast successor. One of the BitTorent missing features was cross-swarm peer identification. This is solved in Tribler, by using permIds for Tribler peer identification lying in different swarms [24].

3.1.2 Other BitTorrent based P2P

WebTorrent/WebRTC With its first initial release in 2013, WebTorrent is actively serving as a P2P streaming client. It

is developed to work in browser applications supporting connection of wide range of decentralized and distributed browser-browser networks, efficiently. Its resiliency and effectiveness increases with growing user number over websites boosted by WebTorrent. This is a strong motivating point for WebTorrent, since unlike other networks whose reliability reduces with growing network users. In re-decentralizing web, WebTorrent significantly contributes by being the first such candidate. This powerful tool is also employed in Wikipedia and Internet Archive like projects. Fast and low-cost access to content further increases its reliability while maintaining compatibility with BitTorrent.

The streaming torrent client is employed in web browsers and uses the normal seeding procedure as in BitTorrent i.e. peers downloads pieces of contents from the other peers who has already finished downloading. However, for peer-to-peer transport facility, it utilizes WebRTC (Web Real-Time Communication) (https://webtorrent.io/); unlike BitTorrent which uses TCP/UDP as transport layer protocol.

3.1.3 Based on Gnutella

Ares galaxy Advanced from classic Gnutella, Ares Galaxy (https://en.wikipedia.org/wiki/Ares_Galaxy) is an open-source file sharing system. It utilizes its own decentralization component with super node/leaf architecture. It considers simple and quick access interfacing with the assistance of in-built sound video viewing choices. Other than using Gnutella's highlights, it has likewise been expanded (version 1.9.4) to exploit and support BitTorrent systems. Because of complexities in understanding its convention engineering, it isn't prevalent. However, it is so ground-breaking system that it takes into account information sharing intersection firewall limits. It is bit by bit in more use with increment in the document sharing applications. It requires a few enhancements in its recently joined peers locating component. To permit super-hubs into its systems, it depends on embedding hash links in the location bar. Ares incorporates "hash links" functionality, it can look for companions with records relating to a hash and download from them. Ares likewise utilizes hash links for its chat-rooms and its immediate chat tool. Its open source nature, accessibility of documents, and lacking corporate greed has brought about a populace of no less than a few hundred thousand people.

Ares galaxy utilizes a framework in which a peer from the ordinary client phase could be elevated to a peer that additionally plays role similar to that of a server i.e. a super-node. Similar strategy of peer hierarchy also exists in the Fast-Track [5], a non open-source P2P system. The super-node server should not be confused with traditional servers of client-server model since the former is not owned by any centralized company. With the intention to make it more troublesome for clients to be followed by; for instance the music and motion picture industry, countermeasures are utilized in Ares network. Truth be told, the system does not give measurements regarding the number of users and the quantities of their mutual documents. Looking is naturally restricted by the framework by countermeasures in both the ordinary user mode and the super-node mode. Kolenbrander and others presented a measurable investigation of utilizing Ares network worldwide in connection with the circulation of CAM [25]. The traces on a computer, running forensic analysis of Ares Galaxy P2P, depicts that Ares galaxy is powerful P2P for securing file-sharing applications.

3.1.4 Based on IPFS

Filecoin Like Bitcoin, Filecoin is a circulated electronic file stockpiling system (https://filecoin.io/#research). It was produced over the IPFS P2P networks by protocols Labs and Juan benet. Its working is pretty much like Bitcoin, in that Filecoin utilizes blockchain like Bitcoin. It is essential to note here that, Filecoin is worked over IPFS and not on Bitcoin. We referenced Bitcoin in the context of Filecoin, to explain it, since Filecoin utilizes a portion of the highlights of Bitcoin to enhance its framework execution. This data storage network is a decentralized stockpiling system which is auditable, freely verifiable and structured on incentives. Authors of Filecoin referred it as *"file storage network that turns cloud storage into an algorithmic market"* [26]. It employs dual nodes viz. storage and retrieval. The users can match their stockpiling as per their needs while simultaneously maintaining retrieval speed, redundancy and price balance. Filecoin uses its base (IPFS) for addressing and data moving.

Filecoin protocol is robust, and it achieves its robustness through content replication and dispersion. Users can chose the proper replication perimeters for security threat resistance. Cloud storage nature of protocol also offers security features with file content encryption at both ends of users. Its developers coined protocol into 4 elementary components which are DSN (Decentralized Storage Network), novel proofs-of-storage, verifiable markets and useful proof-of-work. The DSN uses three sub-protocol like *Put*, *Get* and *Manage* [27]. The *Put* is used for content storage under unique identifier key and *Get* is used for retrieving content stored using key. Whereas the *Manage* sub-protocol is used for network coordination, managing storage and auditing services. Besides offering data integrity and retrievability, the DSN also offers good management and storage fault-tolerance. The Earlier protocol uses proof-of-retrievability for verification, the next version of protocol came with storage proof of mining using sequential and frequent proof-of-replication; along with proof-of-storage protocols.

3.1.5 Others

Shareaza To support multiple P2P networks, a single file-sharing tool was developed and is popular as Shareaza. It has support for classic Gnutella (both Gnutella and Gnutella2) along with BitTorrent, eDonkey like peer-to-peer networks (https://en.wikipedia.org/wiki/Shareaza). This makes it suitable for present generation applications which can cover huge range of users. Further, it allows its users to download any file-type.

Its multi-network capability along with its security features, different modes of operation, IRC (Internet Relay Chat) makes it suitable in various applications (https://wikivisually.com/wiki/Shareaza). Speed of downloading files is also quite good since a single file can be downloaded from different networks at once. This is done by hashing files. Users can search for respective content through hash values. It uses filters which are an extendable XML schema for the security, and can be edited inside Shareaza.

3.1.6 Discussion

In the present P2P world, there is a propelling innovation to enhance diverse file sharing applications using advanced unstructured networks. Different P2P talked about in the paper bolsters this statement. BitTorrent utilizes TCP connections with UDP packets which are not working in today's web since security mechanism is not fully developed in BitTorrent**.** Although some of the classic networks are popular, and are welcome for technology advancements; but they do lack in their performance in the fast changing corporative world. Therefore, some advanced new technology for P2P networks, especially in unstructured category serve the purpose, under high churn environment. Using classic networks as the base, advance networks achieve boost in their performance. Such advanced P2P networks are the technical further steps for integrating security as well as freedom. Employing them in target applications, bandwidth costs also sees reduction on popular files downloads, distributed file transfer boost with support for advanced security features. Using advanced non-structured P2P like Dat, decentralized record sharing, programmed document forming, and secure information reinforcement has been achieved. Also, planning and synchronizing complex process conditions crosswise over various elite figuring bunches. Through Dat protocol's performance, it has been figured out that decentralized registering networks can enhance logical information executives [22]. In classic P2P approach, specialized issues were profoundly engaged upon, without giving much attention to social communities. But, with the high connectivity environment evolution; social network likewise needs measure up to significance. Tribler of NL P2P took into account this issue. It avoided partial centralization existing with BitTorrent, without disturbing its compatibility with its substrate BitTorrent. Also, it eliminated flooding based search methods in classic unstructured networks through (TTL =1); employing only its neighbors for a remote search [24]. This has reduced overhead in routing since flooding is eliminated. The default approach of BitTorrent which is to download the rarest piece in the first place, for guaranteeing the soundness of all pieces in the swarm, is also improved. In VOD of Tribler, peers need to download only the initial few pieces at the earliest opportunity to begin playback as before as would be prudent, thereby fastening communication process. Also, query mechanism in classic and NL P2P differs. While peers in BitTorrent like networks queries using IP addresses, most NL P2P protocols employs content based query. This provides advantage of getting direct access to file. Figure 4 depicts this scenario. New technology of Ares might encounter some developing torments; it in any case is an essential player in the P2P world. In addition, new client is being tested by the development team of Ares which should re-eliminate various issues surrounding this community. Besides being Gnutella based, it supports BitTorrent, thus chances of integrating Gnutella-BitTorrent technology exists with this protocol (https://sourceforge.net/projects/aresgalaxy/editorial/). As of present research, examiners are revealing that more genuine cases on Ares Galaxy (Ares) are made by them in contrast to other open P2P networks.

Table 2 lists major merits and demerits of NL P2P networks. Most of them are less prevalent since they are still under the ongoing innovation. Nonetheless, due to their secure platform, the protocols offer good reliability and anonymity.

Very little work has been carried out on programmed prioritization with the data derived from information that is accessible on P2P systems. Multi-cast is an effective component to help these classes of utilizations as it decouples the extent of the beneficiary set from the measure of state kept at any single hub and possibly keeps away from excess correspondence in the system. The previous decade has brought various application-level ways to deal with broad communications appropriation. With an application-level methodology, end frameworks design themselves in an overlay topology for information conveyance utilizing regular unicast ways. All NL P2P usefulness is executed toward the end frameworks, rather than at the limited users, giving the majority of the advantage of the system layer approach while maintaining a distance from the organization and versatility issues with such strategy. While noteworthy advancement toward this vision has been made over these most recent couple of years, supporting astounding, data transmission, serious applications in agreeable situations remains a test. Unlike classic P2P which suffers from security attacks, most advanced P2P due to its security feature integrated with it are best suitable in present world. Dat, Tribler, Filecoin, ZeroNet, Tribler and Shareaza are secure networks. ZeroNet security is good over BitTorrent, however possibility of ransomware attacks has been raised it [28].

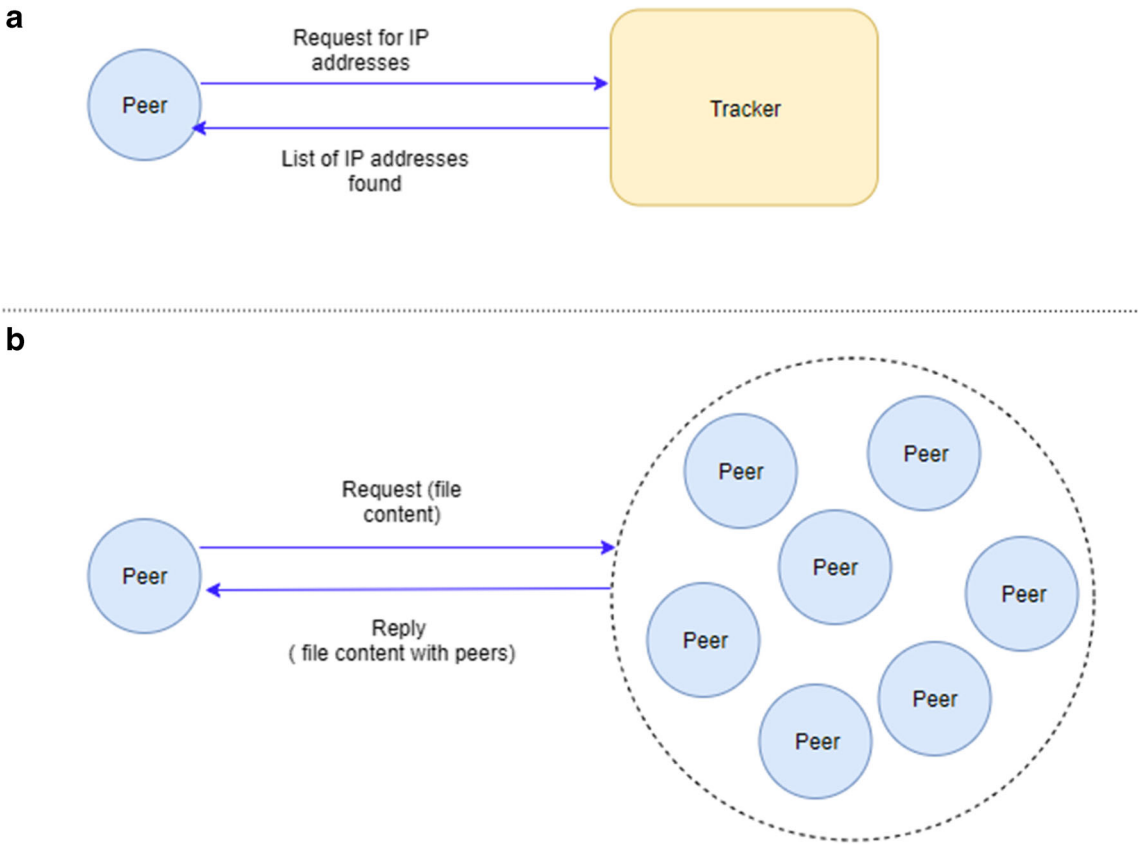

Fig. 4 Peer request mechanism in (a) classic P2P and (b) NL P2P networks

The peers in classic networks employ randomized approach for querying its neighbors. This flooding approach is eliminated by NL P2P through super peer node architecture. This is shown in Fig. 5. This reduces the overhead due to flooding, and helps in faster query delivery, since nodes do not need to contact all peers. Also, the decentralization nature

Table 2 NL P2P networks advantages over classic substrate

	Merits	Demerits
ZeroNet	Web decentralization with no traffic littering; Security and privacy ; anonymous	Less prevalent
Dat	Security and privacy; anonymous	
Tribler	Security and privacy; anonymous	
WebTorrent	Reliable	
Ares Galaxy	Security	
Filecoin	Efficient decentralized storage	
Shareaza	Multi P2P based	

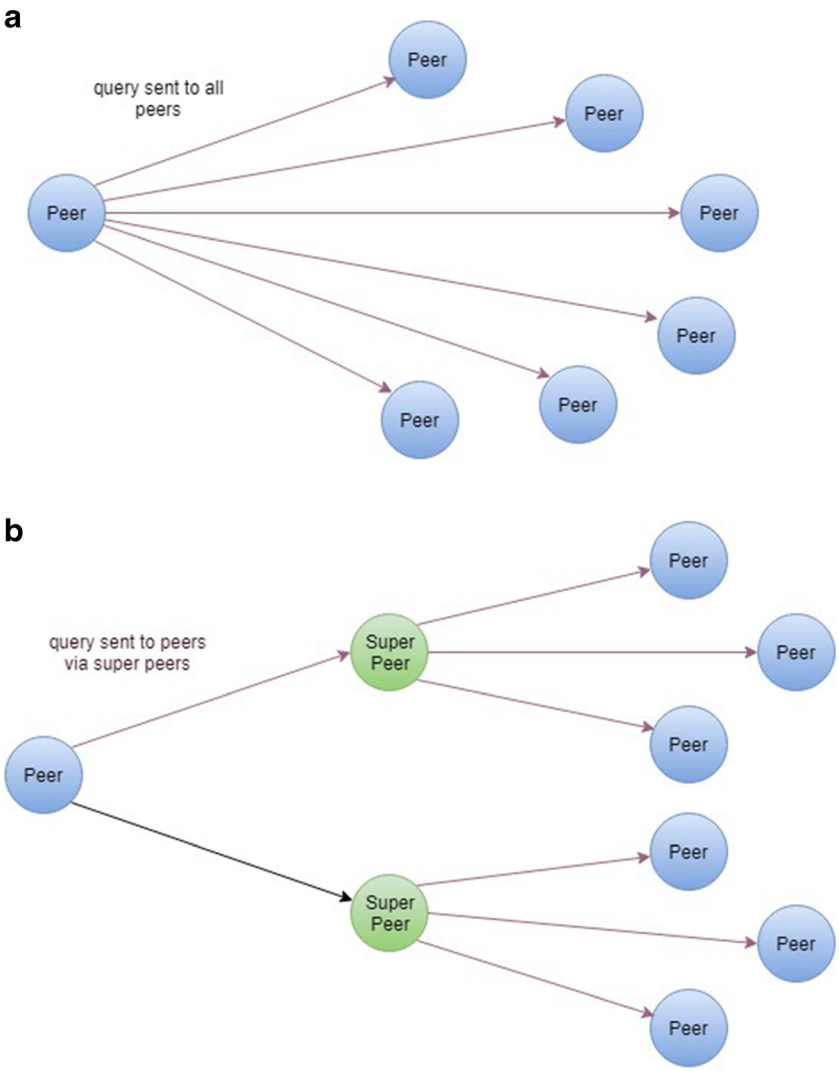

Fig. 5 Peer discovery in **a** classic P2P and **b** NL P2P networks

makes significant effect over overall overlay network. Figure 6 depicts the decentralization which could be partial or pure. In Fig. 6.a, a virtual server for e.g. tracker in case of BitTorrent, locates the contents over other peers (communication between peers follows sequence indicated against numbers 1-2-3-4 in Fig. 6.a). The pure decentralization as in NL P2P eliminates any virtual control, illustrated in Fig. 6.b and thus peer discovery is also direct between peers. Recently, integrating Block-Chain based crypto-currencies to overlay networks has also seen a remarkable progress. Wide networks of machines are well organized for high computational performance, using NL P2P (https://cryptorum.com/resources/filecoin-whitepaper-cryptocurrency-operated-file-storage-network.29/). SDN (Software Defined Network) is another technology which is looking a good integration with overlays for traffic engineering problems. Belzarena, Sena and Vaton [29] have accomplished such integration for better QoS routing. This can also be taken for NL P2P to add further suitability in wide overlay networks (including social networking) through efficient traffic handling ability. There is by all accounts a lot of space to create variations or augmentations of P2P for different applications. For instance, we have seen that the progressed P2P takes into consideration social applications, block-chain innovation. Progressed P2P are currently beginning to get huge consideration from the algorithmic network, and there have been various ongoing outcomes in such domain. In light of their propelled security, the NL P2P will keep on being utilized in current system frameworks in new and intriguing ways. As more data gets created with demands and emergence of new target applications, storage data networks like Filecoin greatly helps for efficient and reliable storage.

Further, integrating cryptographic concepts it boost its security power [27]. The robust nature of this protocol is better achieved with content replication and dispersion. While most of the advanced P2P are based on single or dual protocols, some advanced P2P supports multi-P2P. Shareaza is one of them, and is also a base for futuristic projects. While it was initially designed with support for windows, it was later improved for other operating systems. It is a very powerful tool

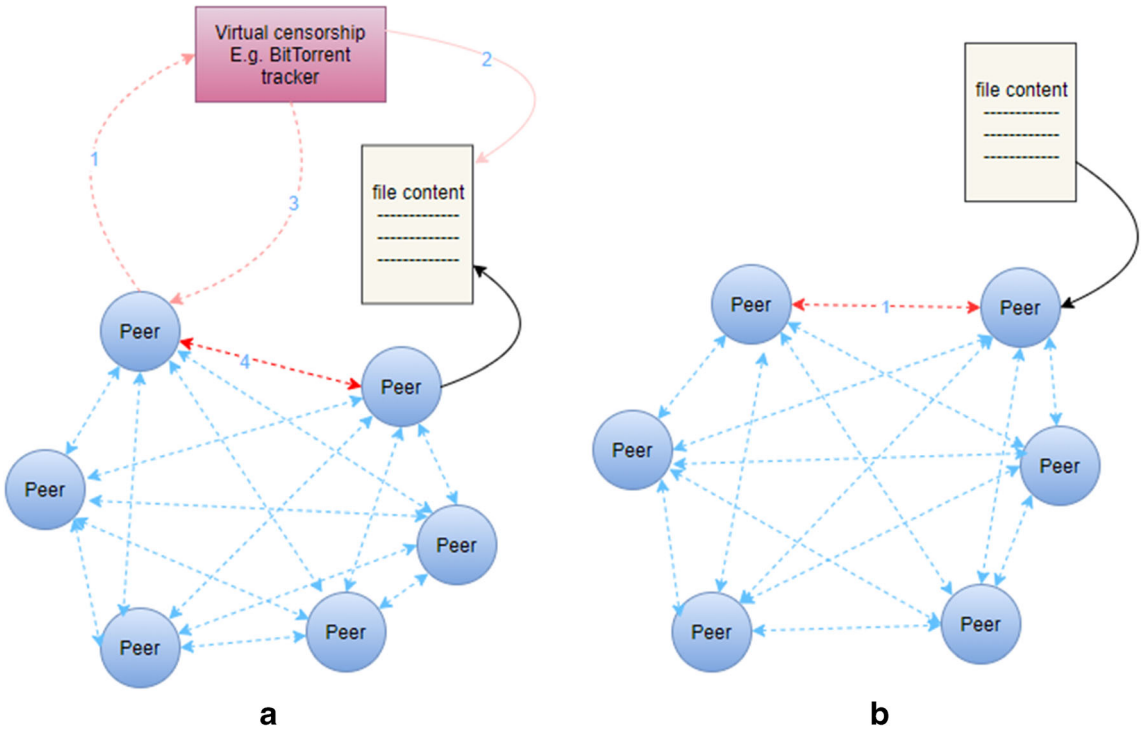

Fig. 6 Decentralization in **a** classic and **b** NL P2P networks

including intelligence to identify among corrupted file contents. The WebTorrent, which is also a new P2P, aims to take BitTorrent protocol further in the direction of web decentralization. While its robust nature is attractive, it is however subjected to security issues. Its open issue of security and privacy has been raised recently in 2018. The discussion (https://github.com/brave/browser-laptop/issues/12631) raised risks such as traffic tracking, unlocking public IP addresses associated with files download. Also, WebRTC which is used as transport protocol in WebTorrent is not secure enough concerning IP addresses. The network and its protocol need to be freely available to ensure its long term sustainability. Majority of advanced P2P are developed with their open source nature in order to be available to large number of users.

The Table 3 depicts comparison between various NL unstructured P2P networks with respect to their architecture/topology, peer discovery mechanism, routing, fault-tolerance, security etc. Almost all NL P2P are completely decentralized. This is a meritorious point in the design of advanced P2P networks. Not only because it eliminates centralized control, but also it has potentials to eliminate network partitioning like problems. With no central control, for example central authority will have no control to disconnect its peers from other peer networks. Further, working with the underlying protocols of NL P2P does not pose more difficulty since it employs widely accepted classic P2P networks. Based on above discussion, we list some separate peer and network level characteristics under high environment and is depicted in Fig. 7.

3.2 Networks evolved from basic structured peer-to-peer networks

The P2P structured networks are mostly based on DHT approach. In this section we discuss NL P2P networks spunned from classic Chord, Pastry, Kademlia and CAN. While O (Log N) number of hops was putting bound in Chord, we see how Koorde, EpiChord and Accordion solve this lookup and other related problems in Chord. Also, security, routing and load balancing features are considered. Figure 8 briefs out structured NL P2P based on their classic counterpart. In Table 4, performance comparison of various NL P2Ps and their classic substrate under high and low churn is illustrated.

3.2.1 Based on classic chord

Koorde It is a distributed hash table (DHT) based protocol evolved from Chord, which uses De Bruijin graphs and hypercube topology. It combines the advantages of simple nature of Chord as well as eliminates limitations with Chord by meeting lower bound of O (log N) imposed over Chord. Also, Koorde [30] allows for minimum overhead for maintenance. In Koorde, O (log N) numbers of nodes are contacted with state per node equal to O (1), for looking up a key. As mentioned earlier, De Bruijin graphs are employed in Koorde. It is for lookup requests forwarding process. Similar to Chord, De Bruijin pointer in Koorde, is also given importance. By following

Table 3 Comparison of various unstructured NL P2P overlay networks

	Classic P2P network used	Architecture/ Topology	Lookup/Peer Discovery	Routing	Fault-tolerance	Security	Specific Application	Protocol and its platform supported
P2P over ZeroNet	BitTorrent, Bitcoin cryptography	Hierarchical	Local peer discovery using DHT	P2P onion router (first layer)	N/A	Address and private key generation using Bitcoin cryptographic hash; Uses public keys rather than IP addresses for site identification	Websites publish and editing	Open source; Windows, Linux, OS X, FreeBSD, Android
Dat	BitTorent	Hypercore protocol using Merkel tree	Decentralized mirroring	No specific mechanism	Yes	Public-key addressed, secure sync; signed and integrity checked updates	Decentralized web data sharing	Open source; MacOS, Linux
Tribler	BitTorrent	Hierarchical	Eliminates flooding,uses TTL =1; Buddycast algorithm	P2P onion routing	Yes	Message signing using public/private keys called PermIDs	Social networking	Open source; Linux, Windows, OS X
WebTorrent	BitTorrent	No hierarchy; WebRTC	No specific mechanism	No specific mechanism	Yes	N/A	Decentralizing Web	Open Source; Linux, Windows, macOS, Free BSD
AresGalaxy	Gnutella (version 1.9.4 supports BitTorrent)	Hierarchical	Broad-cast kind of searches; Hashing functionality	Gnutella query routing	Yes	Security decreases with increase in file-sharing	File-sharing tool	Open Source;Microsoft Windows, macOS
Filecoin	IPFS	Block-Chain based with no hierarchy	No specific mechanism	No specific mechanism	Yes	LibP2P for security, Cryptocurrency	Decentralized storage network	Open source
Shareaza	Multi-P2P protocols; Supports Gnutella, Gnutella2, eDonkey, BitTorrent and many others like HTTP.	No hierarchy	Similar to BitTorrent	Similar to BitTorrent	Yes	Security intelligence for identifying corrupted contents	File-sharing tool	Open source; windows

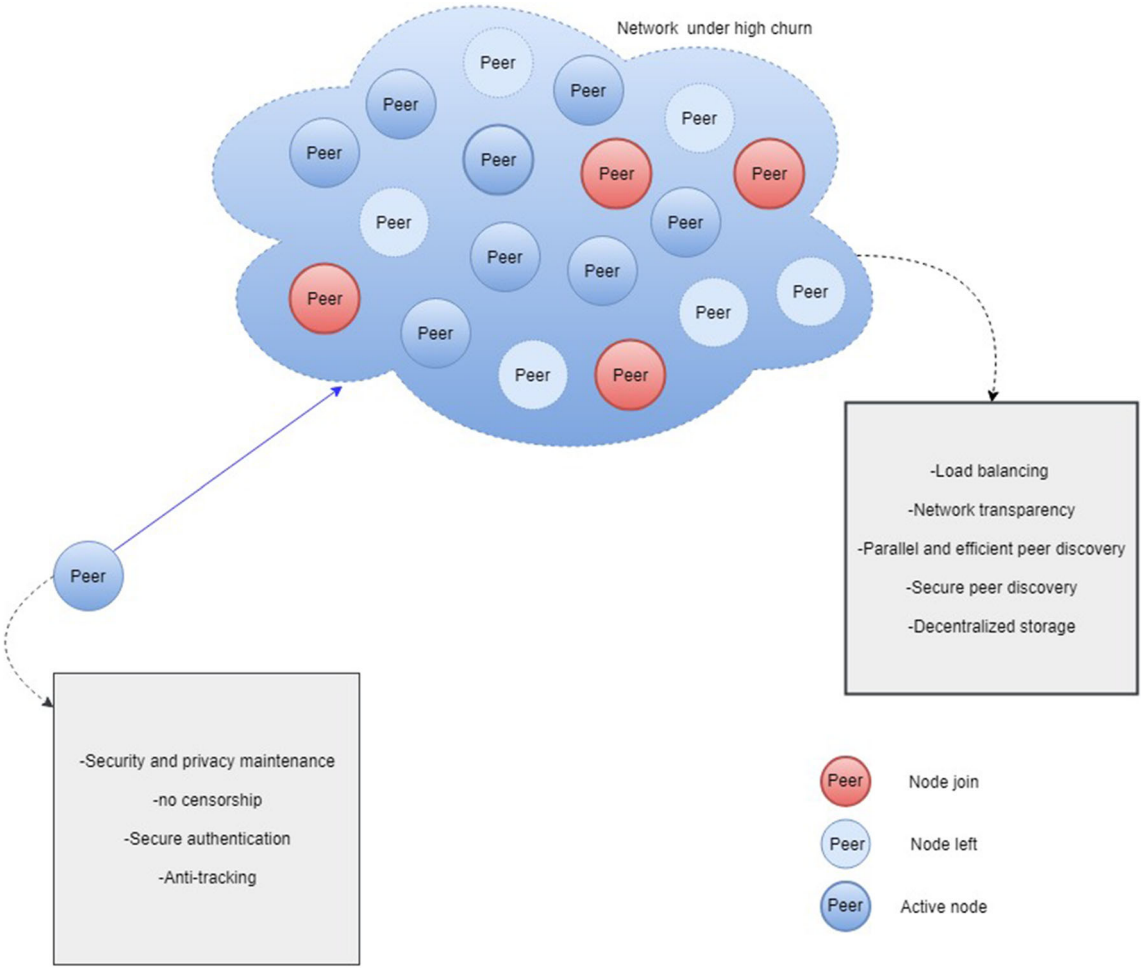

Fig. 7 Network and peer level characteristics under high churn

successors, a query is delivered to its destination. Due to these characteristics, Koorde can utilize join algorithm of Chord. Chord's stabilization algorithms and successor list are used. However, self-stabilizing property which exists in Chord, is doubtful in Koorde. The degree and hop count trade-off is achieved in Koorde by extending it to degree-K of De Bruijin graphs. Fault-tolerance in Koorde is achieved through choosing K = log N.

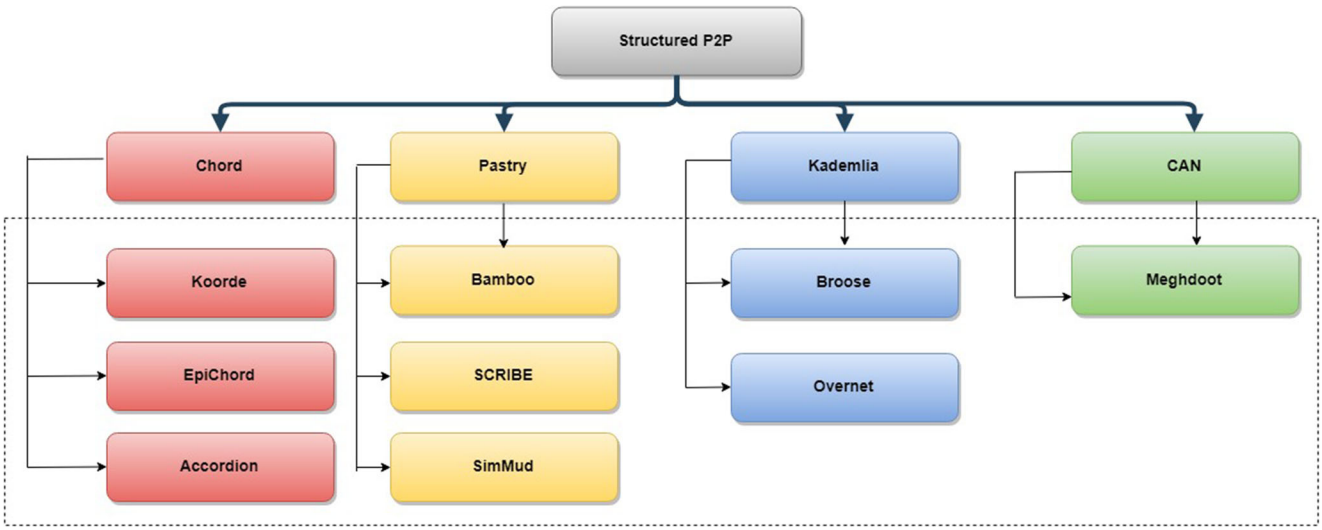

Fig. 8 Structured NL P2P networks

Table 4 Classic and NL P2P under high churn

	Advantages	Performance under	
		High churn	Low churn
Chord	–	×	✓
Koorde	Improves lookup	✓	✓
EpiChord	Improves lookup	✓	✓
Accordion	Efficient bandwidth usage	✓	✓
Pastry	–	×	✓
Bamboo	Improves routing	✓	✓
SCRIBE	Scalability, fault-tolerant	✓	✓
SimMud	Scalability, fault-tolerant	✓	✓
Kademlia	–	×	✓
Broose	Smaller routing table	✓	✓
Overnet	Availability	✓	✓
CAN	–	×	✓
Meghdoot	Good load balance	✓	✓

Routing Koorde effectively uses sparsely populated identifier ring using De Bruijin graph, which in other protocols are ignored since they considers at any given time, only few nodes are joined, and other nodes are imaginary to solve problem of collisions in network. To achieve this, every joined node say 'm', keeps tracks of successor addresses, about two other nodes. The first is its successor and second is the m's first De Bruijin node. The lookup algorithm finds successor (k), for looking up a key k and employs extension of De-Bruijin routing. Koorde simulates path, by traversing through next real predecessor say i, to solve problems with incomplete nature of De Bruijin graph. Also, Koorde, reduces routing cost to 2b, with reduction of successor hops per shift to one.

Lookup With 'b' number of identifier bits, Koorde lookup algorithm is able to contact O (b) number of nodes. Also, O (log N) hops can be reduced by choosing an appropriate imaginary beginning node. For lookup, with node 'm', where query originates, if node m is found to be the node responsible for imaginary nodes between itself and successive nodes, than any De Bruijin node 'i', is chosen which is between m and its successor. The distance between node 'm' and its successor, crosses (2^b/n^2) with high probability. This implies that imaginary nodes are present in m region space. In Koorde, peer points at predecessor (km) and the k nodes to its immediate neighbor, instead of predecessor (2 m). This allows for using constant hops via real nodes for simulation of individual imaginary-node hop. This completes routing in ($\log_k N$) number of hops, thereby solving the lower bound for a network of degree k.

Fault-tolerance To achieve fault-tolerance, minimum degree of log N is increased. For fault-tolerance of immediate successors, Koorde uses same successor list maintenance protocol which is used in Chord. With this, even if node fails with half of the probability, then one node in the successor list stays alive with huge probability value, almost at all times. This ensures existence of routing paths, even in worst case, by following the live successor pointers.

EpiChord EpiChord [31] is also a DHT based network, which aids for storing data over a large scale dispensed systems. It eliminates the restrictions of O (log N) logarithmic hops which have been put by the most of already existing DHT topologies. In order to maintain routing states, nodes are piggybacked with additional network information on query lookup.

Similar to Chord, EpiChord is organized into 1D circular address space in which a unique node identifier is allocated to each node. Key is handled by a node whose identifier mostly nearly relates to the key. It maintains k succeeding nodes list, with additionally maintaining a list of k preceding nodes. Unlike Chord, which keeps track of a finger table, EpiChord maintains a cache of nodes. Nodes observe look-up traffic thereby updating their cache and insert a new entry at point when they get aware of a node existing in the cache. Stale nodes are being removed, since nodes within the cache have a timeout. So EpiChord is like a Chord with a cache of additional node addresses.

Routing EpiChord incorporates a reactive routing strategy amortizing network maintenance costs into peer discovery queries. Lookup performance of O(1) can even be achieved under ideal condition, with low maintenance cost. Routing in EpiChord is opportunistic updating where maintenance of routing table relies over lookup load and bandwidth available. When using parallel requests, it uses an iterative lookup technique since it avoids forwarding same additional queries. Also, it permits its node (asking for a query) to fetch all details relative to the query path, which will modify its cache with new entries. To lookup a key id owning information item, a peer node will provoke 'p' queries to the node, in parallel fashion on the spot succeeding id to the p-l nodes preceding id. When query is received, Epichord nodes gives response based on if it its self id, predecessor id or the successor id of the node. If it is its self id, incentives in association with the Id are sent back giving its predecessor and successor data. And if its predecessor id, successor peers details are forwarded along with the 'l' best subsequent hops in target direction and vice-versa.

Lookup To deal with worst scenario of lookup state of O ($\log_2$ ((N)), every nodes splits the address space into segments of exponentially smaller slices. Every node keeps their cache in a way that each slice has minimum of j/ (1-γ) cache entries all the times, where j being a network parameter and γ denotes neighborhood probability that a cache access is expired. To make certain that there are enough unexpired cache entries, nodes check for their cache slices periodically.

Accordion Accordion [32] is additionally one of the advanced protocols having a place with organized group of distributed systems. It adjusts to give best execution for various system sizes and churn rates without crossing bandwidth limit. Utilizing hash function, it can create 128 or 160 bits of one of a kind identifier. It can naturally modify network parameters like routing table size to accomplish good performance. System bandwidth budget parameter of Accordion controls resource consumption which is the most typical issue looked by users. It additionally enhances low-inertness queries (which are having low latency). It utilizes reliable hashing to designate keys to respective nodes. It makes utilization of existing Chord convention, by using its successor rundown and Chord join protocol for maintaining list of successors.

Routing Accordion uses iterative routing. Nodes are learned, lookups are forwarded directly to the next hop. Based on past lookup rate, the levels of parallelism is set using adaptive algorithm. On query origination, node marks some of the parallel copy with a flag so as to give them higher priorities. If some nodes do not have sufficient enough bandwidth, they drop non-primary copies of queries. Non-parallel lookup paths are traced by primary lookup packets. Also, they are optional which reduces delay and increases information learned.

Lookups When searching for a key, Accordion discovers its successor, which is a node whose Id firmly follows key in the ID space. At the point when a node starts query for a key k, node first look its routing table to discover node whose ID nearly precedes k, and sends a query to next node. The procedure then repeats of forwarding query to next successive neighbors which nearly precedes k. On reaching last node (i^{th} node), where k lies between n_i, a reply is sent with the identity of its successor directly by a node to other node which has begun query. At the point when node in Accordion network advances a query, an affirmation is acquired containing a set of its neighbors. This enables nodes to learn from queries. The information about availability of next hop is acquired through acknowledgment. Utilizing parallel way to deal with query, data transfer capacity is effectively used. Accordion utilizes (1/x) appropriation for probabilistic neighbor determination methodology.

3.2.2 Based on classic pastry

Bamboo Bamboo [33] is a structured P2P overlay network whose structure is dependent upon Pastry. It differs from Pastry in the sense that it maintains the same geometry in spite of churns. It is preferred for its efficient routing performance. In addition, it helps in locating rare objects better as compared to unstructured networks. However, it suffers from certain problems of maintaining routing information overhead, increasing network traffic considerably thereby increasing complexity in P2P network. Also, in Bamboo [34], every node possesses same abilities and tasks with symmetric communication. Its flat architecture cannot perform well when churn rates are too high due to calculation of message timeout, proximity neighbor selection. Short session time also degrades the performance, similar to Pastry. Increase in latency can result in the partition of the network. These problems however, can be handled within the Bamboo protocol itself using the hierarchical architecture. The development of this protocol is focusing on a system that can handle high levels of churn. This is its advantage over the classic P2P protocols, which breaks down under high churn rates.

Lookup In flat architecture, peers are organized in a network while offering distributed hash table (DHT) capabilities. It assigns unique node IDs which are generated and distributed using a secure hash algorithm having either a public key or the port number and IP address combination. Every node in this network makes use of two sets of neighbor information viz. leaf set and routing table information. The immediate successors and the predecessors which are numerically closest in circular key space are included in the leaf set whereas the nodes sharing a common prefix that are used for improving the lookup performance are included in the routing table.

Routing It follows recursive and iterative routing. In its hierarchical architecture, routing tables are arranged into $\log_2{}^b N$, where N being the number of super-nodes and with every row having 2b-1 entries. Bamboo allocates a unique 160-bit ID to the node. A set of existing node IDs is distributed uniformly, with the help of a secure hash (for e.g. SHA-1). This is followed by reliable routing of messages subscribed to a specific key to the node. A message can be forwarded to any node within $\text{Log}_2{}^b$ N, where N being node number within that network. Every node in the Bamboo network maintains a leaf set of 2*k nodes.

SCRIBE- large-scale decentralized multi-cast network SCRIBE [35] is an application level protocol with multi-cast infrastructure. It is also large-scale decentralized event notification structure employed in publish-subscribe applications. It is evolved and built on classic Pastry network, and can scale a considerable number of publishers, subscribers and topics. It provide simple API like *create* for creating topic with topicId, *subscribe* allows local node to subscribe topic created, *unsubscribe* allows local node to unsubscribe from created topic, *publish* allows event to be published in the topic created. To create and manage topics, SCRIBE makes use of Pastry. Since, it is decentralized approach; every node has equal capabilities i.e. any node can behave as publisher or subscriber.

Fault-tolerance As far as fault tolerance is concerned, SCRIBE relies over Pastry's self-organizing ability. The default unwavering quality guarantees programmed tuning of multi-cast tree to failure of node and network as a whole. On the best effort basis, event dissemination is carried out; however, consistency of delivery events is not guaranteed. SCRIBE provides good scalability, with support for huge number of nodes. This extends its capability to support applications with different characteristics. Besides, it can balance load with reduced delay and minimum stress. The protocol is also reliable in handling node damages, and it achieves its reliability through replication process. It supports message forwarding redundancy, in which messaged are re-forwarded which have been previously stored in the buffer of its root node. This is useful when its multi-cast tree breaks down.

Routing While Pastry uses prefix routing and proximity neighbor choosing criteria, SCRIBE utilizes reverse path forwarding [36] scheme. SCRIBE has multi-cast union tree which is formed as a result of path union from receivers to the root. Node joining is same like Pastry i.e. node sends join request with destination key. Request is routed to node whose Id is closer to group Id of a group, to which node wants to join. SCRIBE offers advantage over Pastry by using reverse forwarding mechanism as multi-cast tends to have shorter edges as it moves from root to the tree leaves. SCRIBE uses mesh overlay network priority type [37].

SimMud To solve issues like applications requiring frequent updates, with further forwarding those updates considering time constraints, and bandwidth restrictions, SimMud was developed. There are many improvements done in SimMud to improve its applications. However, in the paper we consider SimMud version, which focuses on performance and availability related issues, particularly in online games applications. In that, SimMud is developed over SCRIBE and classic Pastry protocol.

Lookup, routing and fault-tolerance For its applications, SimMud is developed in which various peers are mapped to the Pastry key space. Each area is relegated with ID, utilizing SHA-1 calculation. And they are mapped to a node manager, using DHT. For N number of hubs, it takes O (log N) jumps with average rate of messages scaling with O(log N) [38]. SimMud is an application layer multi-cast protocol [39]. Pastry, SCRIBE does not offer good fault-tolerance as their routing is sensitive to network failure. So to solve fault-tolerance like issue, SimMud was developed in its target applications. Three assumptions were made viz. independent node failures, low failure frequency and message routing to the correct node. The protocol was enhanced considering fact that peers display locality of interest, and thus are viable to achieve self-organizing features.

The fault-tolerant protocol also achieves consistency as far as network scalability is concerned. For maintaining object sharing consistency, coordinator strategy is utilized, which assigns coordinators to every object. Although, protocol is developed considering Pastry and SCRIBE, it can be extended and simplified in other classic hashing based protocol like ring-based Chord. The grouping of peers and objects is carried out with respect to their regions. The Pastry key space is then used for mapping peer nodes so that the various regions can be distributed over different peers. Hash functions like SHA-1 algorithm, are used for hashing region which is also used for calculating IDs to be assigned to the regions.

3.2.3 Based on classic Kademlia

Broose It is also DHT based protocol but additionally uses De Bruijin topology. Broose [40] was developed to improve upon practical protocol like Kademlia based on same De-Bruijin topology. It solves loose framework for DHTs in Kademlia. Broose stores an association on k nodes rather than one, to obtain high reliability in the context of node failures, in a manner similar to Kademlia. There are various version of Broose, one such is a optimized version of Broose with bucket redesign which handles tight bounds over routing table. In Broose, key collision hot-spots balance is achieved, which makes it a decent base when managing file sharing peer-to-peer applications.

The main problem with Kademlia was to select nodes for association storage for a given key in a free fashion. Kademlia was based on hypercube topology resulting in O (log N) routing table. This is eliminated by Broose, through use of De-Bruijin topology. All node hash table keys and node identifiers are n bits positive integer, with n large enough value to avoid collisions. Similar to Kademlia, Broose uses XOR metric it measures if identifiers has long common prefix.

Routing Besides, constant routing table size of O(k), steps lookup with routing table size of O(k log N) for obtaining O(log n/log log N) steps is also achieved in Broose network. Broose uses refreshing buckets policy similar to Kademlia, for closest k number of nodes. Broose allows storage of only close contacts unlike Kademliia, which results in routing table size to be reduced.

Lookup With constant routing table size of O(k), it can contact peers with lookup in O(log N) hops. And with O(k log N) routing table size, it can likewise be parameterized for query of O(log N/log N) hops.

a. *Right-Shifting:* There is right shifting lookup in Broose, wherein each hub keeps two containers/buckets R0, R1, for contact storage with identifier near to right shifted peer identifier. For a query of key 'w', a hub first calculates

distance 'd', in number of bounces to a node storing w. Alpha as a convention parameter is utilized for accelerating queries with respect to node failure. To discover some k's nearest node to key w, right shifting approach ought to be efficient enough. With this, a current relationship with key w can be found.

b. *Brother lookup:* This type of lookup approach maintains B as brother bucket, with which identifiers close to that of node can be stored.
c. *Left-shifting:* This type of lookup is used for reinforcement of buckets through requests. It is more or less similar to right-shifting lookup approach. Broose can also employ accelerated lookups when shifting more than one bit at a time is required. This is particularly used for minimizing traffic and speeding up lookups.

Overnet Overnet is DHT-based file sharing system and is among the less widely employed protocol. Details about this protocol are scarce since it is a closed-source protocol and third party over -net clients exist for it. Some of the clients such as MLDonkey and KadC libraries exist as toll for learning about this protocol. It depends on Kademlia for its underlying DHT protocol. A file-sharing P2P network is built by overlay networks along with an overlay organization and message routing protocol. In [41], Overnet designers developed a model concerning the availability of host, considering following points about Overnet viz. Overnet uses randomly generated 16-bytes ID for identifying its users rather than IP address, which also solves the host aliasing problem via DHCP. Also, that all peers are equal in Overnet structure has taken into consideration which makes system measurement simpler and also helps to understand protocol easily.

Lookup and routing For lookup and routing mechanism, every host maintains its neighbors list along with their corresponding IP addresses, in order to derive hosts set, Overnet developers have crawled the Overnet by continuously requesting for 25 generated IDs in a random fashion, until nearly 30,000 host IDs are obtained. Of which, a subset of nearly 1000 host IDs are selected and were probed to each other in the subset every 1 h to determine if it is available at that time. Only a subset of few hosts is chosen as the overhead of probed hosts puts restrictions over the frequency of cycling through the hosts.

Next, probe for a host with some ID I is performed using a lookup for I. If host having ID I sends responds, than lookup is successful which implies an availability of host. The probes look similar to normal traffic of protocol which is in contradiction with the previous measurements of P2P networks that use TCP Syn packets. The above mentioned strategy offers certain merits. Besides, that it removes the IP address aliasing issue due to the use of DHCP, it also allows the passing of probes through the firewalls along with all other Overnet messages. Also, due to lookup procedure usage, probes sending need not be repeated to hosts which are not available over long period of time.

3.2.4 Based on CAN

Meghdoot- publish/subscribe over P2P It is a DHT based CAN extension with particular use for content based publish/subscribe networks. It offers advantages such as good scalability and load balance between its peer nodes. The network employing Meghdoot structure is scalable for around 10 K nodes. Also it holds marginal load balance. Subscription storage and its event routing are done by this protocol [42]. It is suitable to work under churn as it permits flexible joining of peer nodes in the network. The model of this protocol uses system represented in the form of *attribute* sets, with every attribute having three parameters; *name*, *type* and *domain*. The subscription process is carried out with one or more attributes. Subscription uses predicates over its attributes. If the predicate of subscription set satisfies event specified attribute value, than a match between that event and subscription set occurs. Following this, the events are then forwarded to the subscribers.

To carry out load uniform distribution, Meghdoot employs and uses two techniques: *zone replication* and *splitting.* The Meghdoot developers utilize DHT maintenance via logical space, with lower and upper bound on them. Logical space is partitioned (zone) and peers are allocated to every partitioned space. Meghboot takes $O(d*N^{1/d})$ for routing the subscription to its peer, where N is number of nodes and d being the Cartesian space dimensionality. This publish-subscribe can adapt under high churn environment, with zone replication strategy to reduce overhead. However, Meghdoot may face performance challenges with using 2n-dimensional Cartesian space for handling attribute set of too large size. Although, it is a DHT based approach, it differs from CAN as data delivery to peers is direct i.e. content based.

3.2.5 Discussion

The various issues in structured P2P are mostly related with their fixed lookup, poor security under high churn and low fault-tolerance, especially under dynamic environment. The NL P2P improves them by employing efficient approach. This led the foundation in some classic P2P networks like Chord, Kademlia etc., to get improved from DHT to De-Bruijin graphs. The classic P2P networks are limited in their lower bound for O (Log N) hops, which are handled by Koorde and EpiChord. EpiChord has O(1) under ideal condition. However O (1) further led to increase in the traffic as the size of the network increases [32]. The churn sensitive poor performance has also been reported in the classic P2P. Classic

P2P handle churn rate, but with large network size, their performance was found to degrade [32]. The logarithmic lookup in most of the NL P2P thus differs. CAN based Meghdoot has lookup of $O(d*N^{1/d})$.

Accordion protocol achieves good bandwidth usage and can handle churn even with large network size, along with minimum delay. The performance boost is thus achieved with Accordion as it better tunes network parameters over different network sizes. Also the parallel lookup mechanism further improves it. However, redundancy may increase since multiple lookup copies are sent via different paths. The most of the structure P2P are flat, in which peers are having equal priorities. But Bamboo is among one which has super-peer architecture (hierarchical), in addition to its flat topology. Under low churn environment, it uses reactive routing like EpiChord. However, under high churn, it switches to periodic routing. Other Pastry based P2P, like SCRIBE and SimMud are achieves good resistance against network or its nodes failure. While SCRIBE uses self-organizing property of Pastry to provide fault-tolerance, SimMud assumes failure of individual peers like situation by considering locality interest. Besides, fault-tolerance bandwidth usage also observes significant and efficient improvement in the SimMud. However, its peer subscription takes comparatively longer time, since it uses DHT based routing [43]. Employing SCRIBE like P2P technology attains comparable boost in its performance contrast to IP, multi-cast.

The De-Bruijin based Koorde although eliminates problems with Chord, but, the stabilizing properties of Chord is not integrated with Koorde. However, getting aware to newly nodes joined can be used to eliminate stabilization algorithm. Broose (based on Kademlia) also integrate De-Bruijin design in its protocol. Similar to Kademlia, Broose also supports refreshing policy; however, the policy differs in both. To keep size of routing table, only closest peer contacts are saved. By comparing both Chord-based Koorde and Pastry-based Broose, it can be noted that the common De-Bruijin topology in two achieves a lookup which is variable. Besides O (log N) lookup hops, Koorde has O (log N/ log (log N)) hops, while Broose has O(log N/ log log N) number of hops. The De Bruijin graphs also offers alternate route in an independent manner. However, the complexity with the routing using De Bruijin graphs also increases since it needs to be aware of the graph size for accurate simulations identifying edges.

To account for peer-to-peer network availability, Overnet serves as a good candidate. The problem of overlapping IP addresses is also solved by Overnet. However, its non open-source nature makes it less available to the users. It still finds its applications but with limited functionality. We discussed Overnet in structure P2P since its base Kademlia is structured P2P network. The Overnet protocol however got merged with eDonkey2000 with more than 645 K users. The bandwidth demands associated with Overnet nodes are also reduced. Compared to classic unstructured like Gnutella, Overnet is found to have almost double success rate for a same set of shared files [44]. Overnet is also a most preferred for large network size. However, Overnet network can be exploited using DDoS attacks like P2P distributed index and routing table index [45].

In classic P2P, CAN was subjected to failure when network partitions occurred. Also, there was no good load balancing mechanism. This is solved by Meghdoot [42] which supports load balancing along with scalability. A good scalability is also far determined by the load-balancing ability of the network itself [44]. SCRIBE, Meghdoot are typical publish/subscribe P2P applicants. Also, this advanced P2P work well under the dynamic environments. Its content distributing feature for subscription and other events makes it meritorious over other networks which uses uniform hash function. As far as bandwidth consumption is concerned, Bamboo is the preferable structured P2P network. Its performance stability even under high churn environment; make it suitable over Pastry and Kademlia. Sharing resources to best utilize them is one.

of the important point in the design of any network. The application layer multi-cast networks allows for such resource sharing between its peers. Likewise, the fault-tolerance is also an important parameter of a good network design. If network has good resistance against failure of its peer nodes, it can efficiently work in large size peers networks. The advanced P2P offers good fault-tolerance. Koorde uses its successor list maintenance protocol to handle minimum log N degree to achieve fault-tolerance. Accordion network likewise adapt by matching its networks parameters under dynamic scenarios. Although NL P2P offers performance improvements in different ways, it is important to note here that they still exhibit little similarity with their substrate protocol. For instance, Koorde employs consistent hashing for its node mapping (nodes and keys are mapped in 2^b identifier space). The network diameter of Koorde is also log N as in Chord. The routing tables in classic P2P had logarithmic nature while the NL P2P like Koorde has a constant (or near constant routing table size). Below we list advantages and summarize key similarities and differences of structured NL P2P over classic P2P networks.(Table 5).

a. Advantages

- *Improved peer discovery under high churn environment.*
- *Sustain scalability and load-balancing features by tuning network parameters.*
- *Good resilience to network failures.*
- *Variable lookup depending on routing table sizes.*

b. Performance similarities

- *Geometry and routing similar under low churn.*
- *Similar load balancing mechanism* e.g. *koorde and Chord uses consistent hashing for node mapping (nodes and keys follows uniform distribution in 2^b).*

c. Performance differences

- *EpiChord maintains cache of nodes unlike finger tables in case of Chord.*
- *Bamboo maintains same geometry in spite of churn unlike Pastry.*
- *Supports both flat and hierarchical architecture* e.g. *Bamboo*

The security highlight of convention is likewise a critical component. While there is no particular security algorithm in classic structured P2P, the NL Pastry based Bamboo uses secure hashing algorithm for distribution of its peer node IDs. The algorithm is not complex as it uses same port number or public key for security purpose. The parallel node discovery (for example, in Accordion), can help reduce network delays. But it can result in overhead and redundant data, if same copies of different lookups are delivered. Nonetheless, it is preferred over classic P2P. In Table 5, we present a comparison between various NL structured P2P in regard to their architecture/topology, peer discovery process, routing, fault-tolerance, security etc. Algorithmic events listed in table below are both; that is which are common in classic and NL as well as those which are solely found in NL networks.

4 Application layer P2P and multi-cast protocols

Application Layer Multicast (ALM) [46] approaches (in the application layer rather than data link layer) are emerging over IP multicast due to its more time efficient nature concerning message delivery. In this section, we list and discuss ALM P2P protocols.

4.1 Nice

NICE (http://www.cs.umd.edu/projects/nice/) is an acronym for Internet Cooperative Environment. This application layer multicast protocol was developed with aim to carry out efficient sharing of its resources with the peers of same cooperative group. It finds it extensive use in networks, where packets use same links in the network often. To carry out efficient routing by using bandwidth efficiently, NICE serve the purpose. The different layers follow sequential numbering and consist of member nodes. It uses hierarchical architecture for its member node arrangement. The cluster head of hierarchical cluster has O (k*log N) neighboring peers, and it is a good candidate for multi-cast applications with less end-to-end delay.

4.1.1 La-Nice

LA-NICE [47] is also a hierarchical application layer protocol and is an advanced version of NICE protocol which enhances message delivery mechanism and further reduction in end-to-end delay, considering the fact that different links are having different bandwidth capacity. To achieve this, Link Aware NICE (LA-NICE) keeps the hierarchical cluster structure as it, while improving upon the member joining and tree maintenance part of NICE. While NICE member joining process was through contacting every member to locate the closest member, and waiting for acknowledgement from other member nodes, (which increases delay), LA-NICE tries to further reduce delay by contacting only selected potential clusters. The strategy is to find cluster leader having maximum value of bandwidth per number of cluster members. For member leave procedure, the cluster leader itself may depart from the cluster, in which case, LA-NICE selects upon the closest member in the center. For tree maintenance, when the size of cluster is observed to be exceeding or becoming lesser than the limits set, LA-NICE considers link load as parameter. The link bandwidth criteria considers 3 nearby nodes and further elects one node as the cluster leader depending upon their bandwidth per number of cluster members ratio. Thus, there is a good bandwidth value in case of LA-NICE.

The high bandwidth leader chosen also reduces delay in message delivery. LA-NICE is also better over SCRIBE (which is also an application layer multi-cast protocol), in terms of its message delivery mechanism, since the later assigns random IDs to its nodes. Additionally, LA-NICE also takes into account proximity feature of NICE, while executing its node joining and tree maintenance algorithms. In terms of routing hops, although LA-NICE is almost same is performance compared to its counterpart (NICE), but with increase in number of peer nodes, it is found to outperform NICE, with less delay. The bandwidth utilization criteria of LA-NICE make it a good preferable candidate in the highly congested network, and also under high churn environment.

4.2 N-tree

The N-tree [48] based application layer multicast focuses on peer interest rather than their distance. It takes into account the fact that event ordering should be done in less time. This helps in eliminating intermediate peers for its event ordering process. The protocol uses the tree topology. The scalable, load-balancing nature of the protocol makes it suitable in application like multi-player games. It is preferred over other multi-cast approaches such as distributed quad-tree [49] which uses

Table 5 Comparison of various structured NL P2P overlay networks

	Classic P2P network substrate	Lookup/Peer Discovery	Routing	Architecture/ Topology	Fault-tolerance	Security	Algorithmic events	Application description
Koorde	Chord	O(Log N) hops for degree 2; O(log N/log(log N)) for O(log N) connected neighbors	Employs extended version of De-Bruijin routing	De Bruijin, ring, hypercube topology (e.g. with degree 2)	Yes; Using successor list maintenance protocol	N/A	Handling concurrent joins	General P2P applicants
EpiChord	Chord	Iterative lookup with O(1) hops under ideal condition	Reactive routing	Circular address space	N/A	N/A	Node caching (instead of finger tables in Chord)	Data storage in large distributed systems
Accordion	Chord	Recursive parallel lookup with O(log N); variable hop	Iterative, Greedy routing	Ring topology (Similar to Chord)	Yes; Tuning network parameters under different network sizes	N/A	Uses Consistent hashing, successor list, join protocol;	Bandwidth sensitive user-specified applications
Bamboo	Pastry	O(log N) hops even under large broken link network	Periodic and reactive routing under high and low churn respectively	Ring, PRR tree	N/A (flat architecture); Yes (Hierarchical architecture)	Yes (Secure hash algorithm)	Periodic recovery, timeout cancellation algorithm, proximity neighbor selection (PNS) algorithm	General P2P applicants
SCRIBE	Pastry	Same as Pastry	Reverse- Path forwarding	Multi-cast tree	Yes; Uses self-organizing feature of Pastry	N/A	Topic maintenance, subscription based on Pastry	Application-layer Mutli-Cast protocol in publish-subscribe applications
SimMud	Pastry, SCRIBE	O(log N)	DHT routing	Muti-cast tree	Yes; (Better than SCRIBE and Pastry)	N/A	Locality of Interest; partial replication	P2P technology in massively multi-player games
Broose	Kademlia	O(log N) and O(log N/ (log log N)) for O(k) and O(k log N) size of routing table	Routing table: Constant size of O(k)	De-Bruijin topology	Yes; using same De Bruijin topology for hotspot balance of key collisions	N/A	left, right-shifting and brother lookup algorithms; Constant size routing tables	File-sharing P2P networks
Overnet	Kademlia	Iterative Logarithmic lookup	Uses Cached peer list	Flat DHT topology	Yes;	Limited security; Proprietary protocol	Good search load balance	File-sharing P2P networks
Meghdoot	CAN	$O(d*N^{1/d})$	DHT and hyperplane zones to route events	DHT topology	Yes; using hyperplane zones	N/A	Zone replication and splitting mechanisms	Content based publish-subscribe

three-layer OpeN architecture consisting of application, core services and connectivity layers. Additional structured Chord like protocol used for key-based routing mechanism is eliminated in N-tree. It is well suited to handle complex queries and hence can be employed in various complex applications, where classic structured P2P cannot handle complex queries [5]. N-tree considers message delivery as a prime factor for contrasting its internal peer processing costs. Minimum requirement of messages along with its peer interest priority also makes it more attractive over other approaches.

4.3 Vast

Voronoi-based Adaptive Scalable Transfer (VAST) is fully distributed and finds applications in *Massively Multi-player online games* [50]. To discover peers in the network, it makes use of voronoi and Area of Interest (AOI). Although the protocol implementation works with assumptions of circular AOIs of same sizes, it can be extended to varying size AOIs [51].

The neighboring peers search occurs mostly through boundary peers of users. Neighbors get updates from mobile peers. The queries are then sent by boundary neighbors and AOI overlapping if any is checked upon. The newly added peers are notified to the moving peer, which stores them in a neighboring list locally managed. The voronoi diagrams are refreshed upon new peer nodes updates. For peer joining, the current neighbor information is used by which every node built their self voronoi diagram. The direct connection is then established between peers and their neighbors. The protocol has been re-discussed and revisited by Backhaus and Krause. They included extended version of Fortune's sweepline algorithm in order to re-build vonoroi diagram on newly added peer updates. In the revisited VAST protocol, application was developed on top of existing VAST. The movement models (Random Waypoint and Group-based Random Waypoint) stressed on experience and team-play engage like factors of players.

4.4 QuON

Quon [52] is a Quad-Tree based network which is in particular use in massively multi-player Internet games. This decentralized protocol helps in better connecting number of player over the network. Integrating P2P technology in such applications further helps in handling churns rate at ease. In the existing technology, there were overhead maintenance problems. To handle growing users (players) rate, the protocol assigns a circular region as Area of Interest (AOI). It uses quad tree topology. It allows its peers to deliver messages in single hop virtually over overlay networks. This considerably reduces routing overhead in the top layers at overlays. The AOI mentioned above offers merits of connecting users which are having similar interest. Also, the users are connected to other AIO which is helpful to prevent partitioning inside the network.

Peer discovery in equivalent to players discovery. This happens in two ways in Quon. When new entry is noted, peer checks for its AOI, following which it establishes direct communication with its newly added friend. Other way is through binding neighbors list. On new peer addition, the neighboring list is updated and sent to other peers. The peers then checks to see if the newly added node lies in its proximity or not. QuON uses point Quad-Tree in order to distinguish among its neighbors and also for managing them. It also consists of positions of the peer nodes and their corresponding neighbors.

The point quad-tree topology of Quon makes it more reliable, since it has simple nature of construction with support for range queries along with k number of neighboring lookup/searches. Also, when users are newly added with its neighboring nodes, quad-tree creates non-overlapping partitions for ensuring maximum one point in the partitioned region. To make sure connectivity exists and is not interrupted, QUON supports backup mechanism for failure detection and repairing same. One uses AOI buffers and other method uses binding neighbors as backup neighbors.

5 Current research issues and challenges

The most issues related with current age systems are addressed by NL P2P networks. While different P2P are progressed, there is as yet flourishing need in different applications. For instance in social overlay systems, overlay partitions is one of the issue that still needs more consideration. Such network partitions should not be confused with virtual world partitioning [53] used in the Networked Virtual Environments (NVEs). The former results in disconnecting peer connections and in the later, partitions are done intentionally to efficiently handle various networks subsets. Building up effective correspondence inside such partitioned systems is a noteworthy test yet. In this section, we discuss some recent approaches and attempts that have been made to tackle network partitions.

5.1 Partitioned social overlay networks

The network partitions results from various factors. One such source of network partitions is government temporarily disconnecting its country's Internet with remaining part of the world. Also, failure of router or other networking devices at the underlying physical layer could also result in network partitions. This is very important issue that needs to be solved. With the ascent in the network partitions inside the P2P systems, new directing

algorithms are therefore required for smooth communications inside such systems. One methodology depended on eMDR (extended multi-dimensional routing) is proposed by Ahssain Hussain [54]. Network called Social Interest Overlay (SIO) was constructed considering classic Chord as the base. It takes into account various measurements like geographical locations, social interest alongside time zones. The methodology works in three phases. Right off the bat, starting stage deals with counting associated peers. This is then trailed by novel SIO configuration utilizing Data Mining Association Rules (DMAR). At last, the last stage manages applying created eMDR to SION and figuring their directing probabilities. The network partitions sometimes also result from various security attacks [55].Such network partitions can be solved using NL P2P. The reason for this can be understood from following characteristics of NL P2P:

a. *Pure decentralization*

As we have discussed in previous sections, classic P2P networks are not fully decentralized. The partial central control (for example in BitTorrent) does allow its users for free communication. However, they need to rely on them for further communication. Also, the content location over the peer is known to the tracker. On the other hand, almost all the NL P2Ps are completely decentralized with more power delivered at the hands of peer nodes. This allows its users to fully exploit this property thereby establishing direct communication with other peers with no third party intervention. Hence, network partitions by central controller can be easily eliminated.

b. *Security*

The NL P2Ps are most secure networks than classic P2P networks. They eliminate the common DDoS and Sybil like attacks. This prevents any attacker from distributing unwanted messages over the routing path. The complete decentralization and security properties of P2P are actually related. With no authorization by third party, peers dependency on them can be eliminated. Hence, peers do not need to trust and rely on them. With this, network partitioning resulting from security attacks can thus be easily minimized to a huge extent depending upon how powerful is the security of the NL P2P.

5.2 Challenges with NL P2P networks

In the paper, NL P2Ps has been discussed concerning network complexity like high churn environment. No doubt, NL P2Ps outperforms over classic P2P networks. However, various challenges exist with NL P2Ps which are especially the main reasons for their less deployment as compared to classic networks. Below we list some challenges:

- *Not fully developed*

Majority of the NL P2Ps discussed especially under unstructured types, are not fully developed. They require a good real time test environment in order to be widely used. Security is better compared to classic counterparts, but with fast growing hardware and software complexity, enhancement in the secure framework is also highly encouraged for their protocol utility.

- *Continuous availability of online peers*

By online peers, we mean active peers in the file-sharing networks. Similar to classic BitTorrent, ZeroNet which is a NL P2P also needs peers to be online in order to serve other demanding/requesting peers in the network. Further, if the peers storing contents are slow, this may frustrate the requesting peer. Although, these issues are not critical, but it requires other peers to store and share similar content that it is searching for.

- *Security limitation due to DHTs*

Many of the structured NL P2P approaches show enhanced performance such as considerable bandwidth usage (e.g. Accordion), availability (e.g. Overnet) etc. However, they still rely over DHT approach which may trade off with their security. And to our knowledge, not much work is done with respect to their security test.

- *Restrictions in multiuser sites*

NL P2Ps has pure P2P networking behavior. However, multi-user sites in ZeroNet protocol maintain a site owner. Although site owner is not participating in peer communication but peer processes such as user authentication and removal of unwanted users is handled by it.

6 Conclusion

P2P networks have attracted significant research and industrial communities over a recent couple of years. The various P2P protocols like Chord, Pastry, Kademlia, CAN, BitTorrent, Gnutella etc. offers advantages such as scalability, load-balancing, availability etc. These P2P protocols are prevalent and widely employed. However, they lack in their performance under high churn environment. Further, partial decentralization, poor peer discovery, overhead in flooding queries, poor security mechanism makes them unsuitable in present

generation networks. In this paper, we brought together a series of newly emerged P2P protocols built on classic P2P, referring to them as next level P2P. Majority of them are completely decentralized and eliminates censorship of any third parties used for peer authentication. Also, problems like network partitions can be tackled using such P2Ps. The applications in which NL P2P systems are utilized, display remarkable execution with such enhanced technology. We have carried out their performance correlation separately in structured and unstructured topology. NL P2P concept is additionally carried out with respect to application layer multi-cast protocols. Many of them need to be tested in different real time environments such as under different churn types and churn rates. Especially, the current unstructured P2P such as ZeroNet, Dat which are newly emerged in last few years requires more practical implementations to be widely employed. Nonetheless, the NL P2P protocol candidates serve as promising protocols in the future of overlay networking environment.

Acknowledgements This research work is funded by Science and Engineering Research Board (SERB), DST, under Grant (EEQ/2016/000413) for Secure and Efficient Communication Inside Partitioned Social Overlay Networks project, currently going on at National Institute of Technology Goa, Ponda, India.

References

1. Ktari S, Hecker A (2011) A peer-to-peer social network overlay for efficient information retrieval and diffusion. In: Park JJ, Yang LT, Lee C (eds) Future information technology. Communications in Computer and Information Science, vol 185. Springer, Berlin, Heidelberg
2. Popescu A (2005) routing in overlay networks: developments and challenges. IEEE Global Communication Letter, IEEE Communications Magazine 43(8)
3. Kamel M, Scoglio C, Easton T (2007) Optimal Topology Design for Overlay Networks. In: Akyildiz IF, Sivakumar R, Ekici E, Oliveira JC, McNair J (eds) Networking. Ad Hoc and Sensor Networks, Wireless Networks, Next Generation Internet. Lecture Notes in Computer Science, vol 4479. Springer, Berlin, Heidelberg
4. Lua EK, Crowcroft J, Pias M, Sharma R, Lim S (2005) A survey and comparison of peer-to-peer overlay network schemes. IEEE Commun Surveys Tutor 7(2):72-93. https://doi.org/10.1109/COMST.2005.1610546
5. Malatras A (2015) State-of-the-art survey on P2P overlay networks in pervasive computing environments. J Netw Comput Appl 55:1-23
6. Korzun D, and Gurtov A (2013) Flat DHT Routing Topologies. In: Korzun, Dmitry and Gurtov, Andrei (eds) Structured Peer-to-peer Systems: Fundamentals of Hierarchical Organization, Routing, Scaling and Security, pp.25-42. https://doi.org/10.1007/978-1-4614-5483-0_2
7. Dorrigiv R, Lopez-Ortiz A, Pralat P (2007) "Search Algorithms for Unstructured Peer-to-Peer Networks," 32nd IEEE Conference on Local Computer Networks (LCN), Dublin, pp. 343-352. https://doi.org/10.1109/LCN.2007.65
8. Kunzmann G (2005) "Iterative or Recursive Routing? Hybrid!," In KIVS, Kurzbeiträge und Workshop der 14. GI/ITG- Fachtagung in Kaiserslautern, vol. P-61 of Lecture Notes in Informatics (LNI)
9. Cohen B (2003) Incentives build robustness in BitTorrent. Workshop on economics of PeertoPeer systems.6
10. Ripeanu M (2001) "Peer-to-peer architecture case study: Gnutella network," Proceedings First International Conference on Peer-to-Peer Computing, Linkoping, Sweden, pp. 99-100. https://doi.org/10.1109/P2P.2001.990433
11. Benet J (2014) "IPFS - content addressed, versioned, P2P file system," CoRR, vol. abs/1407.3561, 2014. [Online]. Available: http://qrxiv.org/abs1407.3561, arXiv preprint arXiv:1407.3561
12. Stoica I et al (2003) Chord: a scalable peer-to-peer lookup protocol for Internet applications. IEEE ACM Trans Netw 11(1):17-32. https://doi.org/10.1109/TNET.2002.808407
13. Medrano-Chávez AG, Pérez-Cortés E, Lopez-Guerrero M (2015) A performance comparison of Chord and Kademlia DHTS under high churn scenarios. Peer-to-Peer Netw Appl 8:807. https://doi.org/10.1007/s12083-014-0294-y
14. Rowstron A, Druschel P (2001) Pastry: scalable, decentralized object location, and routing for large-scale peer-to-peer systems. In: Guerraoui R (ed) Middleware. Lecture notes in computer science, vol 2218. Springer, Berlin, Heidelberg
15. Maymounkov P, Mazières D (2002) Kademlia: a peer-to-peer information system based on the XOR metric. In: Druschel P, Kaashoek F, Rowstron A (eds) Peer-to-peer systems. Lecture notes in computer science, vol 2429. Springer, Berlin, Heidelberg
16. Ratnasamy S, Francis P, Handley M, Karp R, Shenker S (2001) A Scalable Content-Addressable Network. ACM SIGCOMM Comput Commun Rev 31. https://doi.org/10.1145/383059.383072
17. Urdaneta G et al (2011) A survey of DHT security techniques. ACM Comput Survey (CSUR) 43:8:1-8:49
18. Swamy N, Frangiadakis N, Bitsakos K (2019) A Distributed Algorithm for Constructing a Generalization of de Bruijn Graphs
19. Richa A, Scheideler C, Stevens P (2011) Self-stabilizing De Bruijn networks. In: Défago X, Petit F, Villain V (eds) Stabilization, safety, and security of distributed systems. SSS. Lecture notes in computer science, vol 6976. Springer, Berlin, Heidelberg
20. Zhang J, Su J, Wu D (2017), Poster: A Novel P2P-over-Zeronet Anonymous Communication Platform, 38th IEEE Symposium on Security and Privacy
21. Robinson DC, Hand JA, Madsen MB, McKelvey K (2018) The Dat project, a new approach to support data preservation through decentralization. Sci Data 5:180221. https://doi.org/10.1038/sdata.2018.221
22. Ogden M (2017) "Dat - Distributed Dataset Synchronization and Versioning." OSF Preprints. January 31. https://doi.org/10.31219/osf.io/nsv2c
23. Pouwelse J, Garbacki P, Wang J, Bakker A, Yang J, Iosup A, Sips HJ (2007) Tribler: a social-based peer-to-peer system. Concurr Comput Pract Exp 20(2):127-138. https://doi.org/10.1002/cpe.1189
24. Zeilemaker N, Pouwelse JA (2012) Open source column: Tribler: P2P search, share and stream. ACM SIG Multimed Rec 4:20-24. https://doi.org/10.1145/2206765.2206767
25. Kolenbrander F, Le-Khac N-A, Kechadi T (2016) *Forensic analysis of Ares galaxy peer-to-peer network, 11th annual adfsl conference on digital forensics*, Security and Law At: Florida, USA
26. Labs P (2018) Filecoin: a decentralized storage network. [online]. Available: https://filecoin.io/filecoin.pdf
27. Techical Report (2014) Filecoin: A Cryptocurrency Operated File Network. http://filecoin.io/filecoin.pdf
28. Dori A (2016) "The future of Ransomware-ZeroNet protocol" threat intelligence and research
29. Belzarena P, Sena GG, Amigo I, Vaton S: SDN-based overlay networks for qos-aware routing. In: proceedings of the 2016 workshop on fostering Latin-American research in data communication

networks, LANCOMM '16, pp. 19-21. ACM, New York, NY, USA. https://doi.org/10.1145/2940116.2940121
30. Kaashoek MF, Karger DR (2003) Koorde: a simple degree-optimal distributed hash table. In: Kaashoek MF, Stoica I (eds) Peer-to-peer systems II. IPTPS. Lecture notes in computer science, vol 2735. Springer, Berlin, Heidelberg
31. Furness J, Chowdhury F, Kolberg M (2013) An Evaluation of EpiChord in OverSim. 5th International Conference on Network Communication (NetCom). https://doi.org/10.1007/978-3-319-03692-2_1
32. Li J, Stribling J, Morris R, Frans Kaashoek M (2005). Bandwidth-efficient Management of DHT Routing Tables. Proceeding NSDI'05 Proceedings of the 2nd conference on Symposuim on Networked systems Design & Implementation -Volume 2 pp 99-114
33. Tian Z, Wen X, Sun Y, Zheng W, Cheng Y "Improved bamboo algorithm based on hierarchical network model," 2009 ISECS International Colloquium on Computing, Communication, Control, and Management, Sanya, pp. 297-300. https://doi.org/10.1109/CCCM.2009.5270447
34. Dischinger M (2004) A flexible and scalable peer-to-peer multicast application using bamboo. University of Karlsruhe, University of Cambridge. https://www.cl.cam.ac.uk/research/srg/netos/projects/archive/futuregrid/
35. Rowstron AIT, Kermarrec A-M, Castro M, Druschel P (2001) SCRIBE: The Design of a Large-Scale Event Notification Infrastructure. 2233. 30-43. Conference: Networked Group Communication Proceedings. https://doi.org/10.1007/3-540-45546-9_3
36. Zhang R, Hu Y (2003). Borg: A hybrid protocol for scalable application-level multicast in peer-to-peer networks. Proc Int Workshop Netw Oper Syst Support Digital Audio Video 172-179. https://doi.org/10.1145/776322.776349
37. Kaiqi Z, Xiumin Z. "A new approach to simulate ALMI using NS2, " 2010 3rd International Conference on Advanced Computer Theory and Engineering(ICACTE), Chengdu, pp. V5-614-V5-617. https://doi.org/10.1109/ICACTE.2010.5579343
38. J. Chen, S. Grottke, J. Sablatnig, R. Seiler and A. Wolisz, "Scalability of a distributed virtual environment based on a structured peer-to-peer architecture," 2011 Third International Conference on Communication Systems and Networks (COMSNETS), Bangalore, 2011, pp.1-8. https://doi.org/10.1109/COMSNETS.2011.5716424
39. Knutsson B, Lu H, Xu W, Hopkins B (2004) "Peer-to-peer support for massively multiplayer games," IEEE INFOCOM, Hong Kong pp107 https://doi.org/10.1109/INFCOM.2004.1354485
40. Gai A-T, Viennot L. Broose: A Practical Distributed Hash table Based on the De-Brujin Topology. [Research Report] RR- 5238, INRIA. 2004, pp.16. <inria-00070760>
41. Bhagwan R, Savage S, Voelker GM (2003) Understanding Availability. In: Kaashoek MF, Stoica I (eds) Peer-to-Peer Systems II. IPTPS 2003. Lecture notes in computer science, vol 2735. Springer, Berlin, Heidelberg
42. Gupta A, Sahin OD, Agrawal D, ElAbbadi A (2004) Meghdoot: Content-Based Publish/Subscribe over P2P Networks. Proceedings of the 5th ACM/IFIP/USENIX International Conference on Middleware. 3231. 254-273. https://doi.org/10.1007/978-3-540-30229-2_14
43. Hu Y, Bhuyan L, Feng M (2012) P2P consistency support for large-scale interactive applications. Comput Netw 56:1731-1744. https://doi.org/10.1016/j.comnet.2012.01.012
44. Qiao Y, Bustamante FE (2006) "Structured and unstructured overlays under the microscope: a measurement-based view of two P2P systems that people use." USENIX Annual Technical Conference, General Track
45. Naoumov N, Ross KW (2006) Exploiting P2P systems for DDoS attacks. International Conference on Scalable information systems (InfoScale) 47. https://doi.org/10.1145/1146847.1146894
46. Sampaio A, Sousa P (2018) An adaptable and ISP-friendly multicast overlay network. Peer-to-Peer Netw Appl 12:809-829. https://doi.org/10.1007/s12083-018-0680-y
47. Helal D, Naser A, Rehan M, El Naggar A (2014) Link-Aware Nice Application Level Multicast Protocol. Int J Comput Netw Commun Secur 6:13-27. https://doi.org/10.5121/ijcnc.2014.6302
48. GauthierDickey C, Lo V, Zappala D (2005) "Using n-trees for scalable event ordering in peer-to-peer games", Proceedings of the international workshop on Network and operating systems support for digital audio and video, Stevenson, Washington, USA. https://doi.org/10.1145/1065983.1066005
49. Tanin E, Harwood A, Samet H (2005) "A distributed quadtree index for peer-to-peer settings," 21st international conference on data engineering (ICDE'05), Tokoyo, Japan, pp. 254-255. https://doi.org/10.1109/ICDE.2005.7
50. Prodan R, Iosup A (2016) Operation analysis of massively multiplayer online games on unreliable resources. Peer-to-Peer Netw Appl 9:1145. https://doi.org/10.1007/s12083-015-0383-6
51. Backhaus H, Krause S (2007) Voronoi-based adaptive scalable transfer revisited: gain and loss of a Voronoi-based peer-to- peer approach for MMOG. 49-54. 6th ACM SIGCOMM workshop on Network and system support for games (NetGames0, Melbourne, Australia. https://doi.org/10.1145/1326257.1326266
52. Backhaus H, Krause S (2010) QuON: A quad-tree-based overlay protocol for distributed Virtual Worlds. Int J Adv Media Commun 4(2):126-139. https://doi.org/10.1504/IJAMC.2010.032139
53. Buyukkaya E, Abdallah M, Simon G (2015) A survey of peer-to-peer overlay approaches for networked virtual environments. Peer-to-Peer Netw Appl 8:276-300. https://doi.org/10.1007/s12083-013-0231-5
54. Hussain A, Keshavamurthy BN (2018) A multi-dimensional routing based approach for efficient communication inside partitioned social networks. Peer-to-Peer Netw Appl 12:830-849. https://doi.org/10.1007/s12083-018-0683-8
55. Srivatsa M, Liu L (2004) Vulnerabilities and security threats in structured overlay networks: A quantitative analysis. In ACSAC '04: Proc. 20th Annual Computer Security Applications Conf., pages 252-261. IEEE Computer Society

Ashika R. Naik has completed Bachelor of Engineering Degree in Electronics & Tele-Communication Engineering, and Master of Engineering Degree in Micro-Electronics from Goa College of Engineering, India. She is currently working on SERB sponsored project titled "Secure & Efficient Communications Inside Partitioned Social Overlay Networks" at NIT, Goa, India.

Dr. Keshavamurthy B. N. obtained his Ph.D from Indian Institute of Technology Roorkey and presently working as Assoc. Prof., Dept. of Computer Science & Engineering at NIT Goa. His main research interests include & not limited to the following: Data Mining, Privacy Preserving Social Mining.