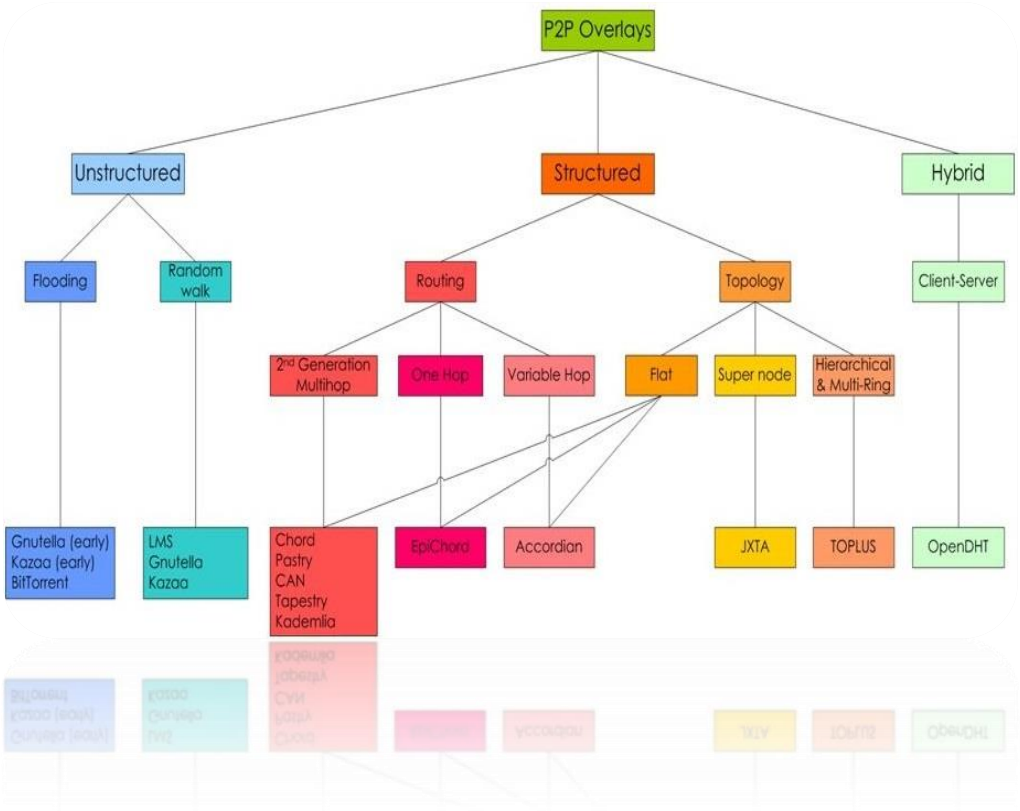
Πανεπιστήμιο Μακεδονίας
ΔΠΜΣ στα Πληροφοριακά Συστήματα
Μάθημα: Δίκτυα Υπολογιστών
Καθηγητής: Α. Α. Οικονομίδης

Peer-to-Peer Networks:

An overview of P2P Systems

Δίκτυα Peer-to-Peer:

Μια επισκόπηση των P2P συστημάτων



Karakerezis Ioannis **Sotiropoulos Thomas**

Thessaloniki May 2018

Περιεχόμενα

ABSTRACT:

When comparing Peer-to-Peer networks with the conventional, well-established Client-Server systems we can spot easily many differences mainly in the architecture and functionality, between them. Peer-to-Peer systems and applications are based on platforms that share resources among web internet users. In other words, Peer-to-Peer systems and underlying overlay networks are form α resource distribution mechanism between peers. These resources may be content, for example videos or files, CPU processing power (CPU cycles aggregation) or storage room. All peers or nodes (we are using the terms interchangeably) are in general, entities with equal privileges and similar capabilities. Every node can either act as a client or a server, often referred as "servent". So, it is about a network formed by multiple communicating clients and servers with interchangeable roles. An extraordinary gain in popularity of Peer-to-Peer networks has been witnessed from millions of internet users the last twenty years. A significant number of Peer-to-Peer networks for content sharing have been presented, developed and deployed. Popular representatives among others are Napster, Gnutella, Kazaa and BitTorrent continuing the legacy of the older Peer-to-Peer platforms, Arpanet and Usenet. Also, the broader idea of Peer-to-Peer computing inspired new structures and philosophies in many areas of human interaction. In this overview our goal is to make a short historical flashback in various Peer-to-Peer platforms, explain the underlying structure of overlays and classify Peer-to-Peer systems by their topology. The benefits and the drawbacks of each p2p architecture will be explained and selected Peer-to-Peer platforms will be presented. Furthermore, certain aspects regarding security in Peer-to-Peer systems will be reviewed and the most important security topics will be elaborated. As a conclusion an overview in contemporary Peer-to-Peer systems will be made and the idea of specialized (or diverse peers) and their contribution in modern p2p networks will be highlighted.

ΠΕΡΙΛΗΨΗ:

Κατά την σύγκριση των δικτύων Peer-to-Peer (δίκτυα ομότιμων) με τα συμβατικά καθιερωμένα δίκτυα Client-Server (πελάτη-διακομιστή ή εξυπηρετητή) διαπιστώνουμε εύκολα διαφορές, κυρίως στην αρχιτεκτονική αλλά και στον τρόπο λειτουργίας. Σε γενικές γραμμές τα συστήματα Peer-to-Peer και οι εφαρμογές αυτών στηρίζονται σε πλατφόρμες όπου διαμοιράζουν πόρους μεταξύ των χρηστών του internet. Με άλλα λόγια, τα συστήματα Peer-to-Peer και τα υποκείμενα δίκτυα επικάλυψης (overlays) συγκροτούν ένα μηχανισμό κατανομής πόρων μεταξύ ομότιμων (peers). Όλοι οι peers ή αλλιώς οι κόμβοι (nodes) του δικτύου (χρησιμοποιούμε ενναλακτικά τους δύο όρους) αποτελούν γενικά οντότητες με ίσα προνόμια και με πανομοιότυπες ικανότητες. Κάθε κόμβος μπορεί να δρα ως πελάτης (client) ή ως εξυπηρετητής(server), όπου αναφέρεται και ως «servent». Πρόκειται λοιπόν για δίκτυο δομημένο από πολλαπλές οντότητες server και client σε εναλλασσόμενους ρόλους. Τα δίκτυα Peer-to-Peer άρχισαν να γίνονται αξιοσημείωτα δημοφιλή τα τελευταία 20 χρόνια μεταξύ εκατομμυρίων χρηστών του internet. Σημαντικός αριθμός από δίκτυα Peer-to-Peer για διαμοιρασμό περιεχομένου παρουσιάστηκαν, αναπτύχθηκαν και βελτιώθηκαν. Δημοφιλείς εκπρόσωποι είναι μεταξύ άλλων το Napster, η Gnutella, το Kazaa και το BitTorrent συνεχίζοντας την κληρονομιά των αρχαιότερων Arpanet και Usenet. Επίσης η ευρύτερη έννοια πίσω από τα Peer-to-Peer συστήματα αποτέλεσε έμπνευση για νέες δομές και φιλοσοφίες σε πολλά πεδία της ανθρώπινης αλληλεπίδρασης. Σε αυτή την επισκόπηση στόχος μας είναι να κάνουμε μία σύντομη ιστορική αναδρομή σε διάφορες Peer-to-Peer πλατφόρμες, να αναφερθούμε στο υποκείμενο δίκτυο επικάλυψης (overlay network) και να

κατηγοριοποιήσουμε τα Peer-to-Peer συστήματα αναλύοντας την τοπολογία τους. Τα πλεονεκτήματα και τα μειονεκτήματα κάθε αρχιτεκτονικής θα επεξηγηθούν και επιλεγμένες πλατφόρμες Peer-to-Peer θα παρουσιαστούν. Ολοκληρώνοντας, θα γίνει μια εισαγωγή στο ζήτημα της ασφάλειας στα Peer-to-Peer συστήματα και θα αναπτυχθούν τα σημαντικότερα θέματα που αφορούν την ασφάλεια και τα μέτρα προστασίας των Peet-to-Peer συστημάτων. Ως επίλογος θα γίνει μια αναφορά στο μέλλον του Peer-to-Peer και στην ιδέα των εξειδικευμένων-διαφοροποιημένων peers και τον τρόπο συνεισφοράς τους στα σύγχρονα Peer-to-Peer συστήματα με κατανεμημένους διακριτούς ρόλους (peer diversity).

KEYWORDS*:*

Client – Server Model, Churn rate, Graceful leaving, Ungraceful Leaving, Flat Peer-to-Peer network, Content Sharing, Fault-Tolerance, Resource Discovery, Network Security, Overlay Networks, Decentralized Architecture, Collaborative Peer-to-Peer, Pure Peer-to-Peer, Hybrid Peer-to-Peer, Distributed Hash Tables (DHT indexing), Query Flooding with TTL, Random Walking, Heterogeneity, Scalability, Deterministic Peer-to-Peer, Super-Nodes, Super-Peers, Usenet, BitTorrent, Kazaa, Napster, Gnutella

I.INTRODUCTION:

Peer-to-Peer networks became very popular with Napster file sharing application in 1999. Napster was an internet platform implemented by Shawn Fanning and Sean Parker for music track sharing that changed immensely the landscape of distributed computing, introducing the Peer-to-Peer networks in millions of internet users (Choon Hoong Ding, Sarana Nutanong, Rajkumar Buyya, 2004). The idea of a more active network model, that brings more power to the end user than just web browsing and email exchange, became the trend. Millions of internet users could form large groups and go even beyond file sharing. Using the already powerful personal computers "formed groups and collaborating to became user-created search engines, virtual supercomputers and filesystems" (Nelson Minar and Marc Hedlund, edited by Andy Oram, 2001). Peer-to-Peer systems are not always use the approach of full decentralization. In fact, they are more efficient when a central control is applied for storing peer IP addresses (creating a dictionary that assigns IP addresses to peers) and buffering content (instant messages storing for example) when users are offline. Peer-to-Peer networks can take advantage of a central control mechanism, by assigning more responsibilities to some peers, thus form a hierarchical structure. These "ultra-peers" or "super-nodes" are assigned with administration roles and are less prone to stability issues (Nelson Minar and Marc Hedlund, edited by Andy Oram, 2001). Peer-to-Peer systems have significant advantages over conventional well-established client-server model networks. They are:

- Scalable: Adding new nodes to the network, will not affect the complexity ratio of the system and ideally the new nodes will be attached and integrated well in the system.
- Reliable: When a node malfunctions, disconnects or attacked by malicious software, the overall stability of the system is not being affected.
- Adaptable: When massive node populations join the network while, at the same time other nodes leave the network (also referred as high "churn rate"), the Peer-to-Peer system is designed to maintain a minimum functional threshold and adapt to the new environment (new peers).

- Resilient: Peer-to-Peer networks and especially later hybrid p2p systems are capable to adapt in difficult circumstances regarding “high churn rate” and security attacks.

II.PEER-TO-PEER HISTORICAL DEVELOPMENT

Back in 1969 in Arpanet, father of the internet, network nodes were like peers, requesting and serving content like in a Peer-to-Peer network by using packet switching. By breaking that data files into smaller parts, the packets, and send them from one node to another, this functionality resembled the later Peer-to-Peer networks. A decade later, in 1979, Usenet was the first original attempt to implement a Peer-to-Peer network. Usenet was a kind of an electronic big magazine with newsgroups and various topics and subjects written in many languages under each newgroup. Users were communicating with each other by posting articles. They could also respond to other users, by posting articles or by sending messages and mails. Usenet was originally developed as a technological forum for Unix community. Users reported and discussed with other users, problems concerning Unix operating system, by sending messages to each other. Founders of Usenet however underestimated the need of people to communicate. Soon Usenet grew big by attracting a lot of people. Making possible the communication between them and even form relationships online. “Without the time and effort put in by its users, Usenet would not be what it is today.” (Hauben, 1995). In the middle of 1990s computer files increased in size because of multimedia technologies and simultaneously increased the need of sharing this multimedia content. Consequently, especially after 1999 the Peer-to-Peer revolution started and a lot of Peer-to-Peer platforms for sharing content developed. Napster, Gnutella, Direct Connect and BitTorrent were some of the most popular Peer-to-Peer protocols (Gera Jaideep, Dr Bhanu Prakash Batula, 2016) (**Table 1: Popular P2P protocols**). Napster was the first large Peer-to-Peer platform introduced to the public and undoubtedly its success was legendary. A lot of Napster clones developed and tried to gain a part of its glory, offering even better efficiency and network security. It was an application that allowed users to share multimedia (music) files and it was officially operational from 1999 to 2001, although it was a Peer-to-Peer system, it used a centralized approach for indexing and resource discovery and querying. Later in 2000, Gnutella protocol introduced and addressed many issues emerged with its pure decentralized structure, using an “ad-hoc” structure to handle querying and let every node (peer) act as “servent”. Gnutella suffered from peers with small bandwidth, lower transfer speeds or peers who only used the platform to download content and not uploading. A year later BitTorrent developed by Brad Cohen at university of Buffalo and used also a decentralized topology, a “tracker” file feature to pack all the information about the IP addresses of the peers sharing the desired content and algorithms to favor uploaders. Thus, addressed inefficiency problems that Gnutella suffered (Washbourn, 2015).

P2P Protocol	First Released
Freenet	July 1999
Napster	September 1999
Direct Connect	November 1999
Gnutella	March 2000
BitTorrent	April 2001

Table 1: Popular P2P protocols

III.EVOLUTION OF PEER-TO-PEER FILE SHARING SYSTEMS

The centralized topology of Client-Server systems (**Figure 2:Client Server vs P2P**) undoubtedly lead to restrictions regarding the scalability and the adaptability of the platform. Users (clients) request content from the server and the server responds. As the number of requests in a timeframe increases, network traffic increases consequently. More network bandwidth, more storage and processing resources needed. The performance of the server will eventually reduce and when a critical threshold is reached severe performance slowdown may occur. The high cost of the server components and the redundancy which demanded may render the upgrade impossible or unprofitable. Also, if the server is removed or if it is unavailable there will be no alternative in the topology and no requests can be served or handled anymore (Rajesh Kumar Maurya, Prof Suman Pandey, Vinod Kumar, 2016). When large number of peers join a Peer-to-Peer network, they form greater groups or clusters that share resources like bandwidth. Thus, there will be observed an overall increase in performance of the network, especially if a hierarchical topology is used with some peers charged with administrative roles and enhanced responsibilities (ultra-peers or super-nodes) in specific regions of the network. The peers simultaneously can download and upload content and while a number of peers preparing to leave the network, new peers are ready to join. This process occurs repeatedly and dynamically, and the end user (peer) does not realize the complexity of the system (Karthikeyan .R, Dr. T. Geetha, Santhini .T, Santhiya .R, 2017). Peer-to-Peer systems are not completely different from conventional Client-Server systems, in some of them, there is present a central control that is responsible of storing the meta-information of the content (**Figure 1:Napster's centralized structure**). This meta-information is usually indexing tables with IP addresses of the peers with the requested content (Rajesh Kumar Maurya, Prof Suman Pandey, Vinod Kumar, 2016). Peer-to-Peer systems with a centralized control are less fault-tolerant because of this. Furthermore, the scalability is limited compared to pure or fully decentralized Peer-to-Peer platforms, which the are also more resilient and robust, because of this lack of centralization (Philip Kisembe, Wilson Jeberson, 2017). Peer-to-Peer networks which eliminate central control are also referred as "flat" Peer-to-Peer networks because all the roles are in the same level. For example, the first version of Gnutella. (Ou, 2010). With the evolution of Peer-to-Peer systems it is interesting to highlight the notions of "graceful" and "ungraceful" leaving of a peer from the network, in the first case the peer informs its neighbors for his imminent leaving and transfers the sharing content to other peers, obviously this is not happening in an "ungraceful" leaving (Ou, 2010).

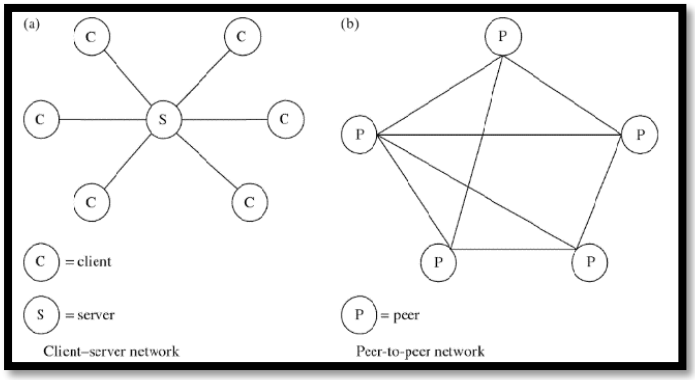


Figure 2:Client Server vs P2P *(Guiran Chang, Chuan Zhu, Wei*

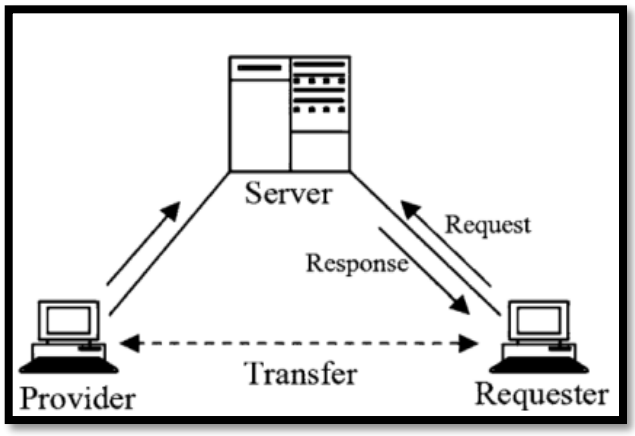


Figure 1:Napster's centralized structure

IV.ARCHITECTURE: OVERLAY NETWORK TOPOLOGY

The topology of a network is of very high importance and in Peer-to-Peer networks the need for providing stable performance and acceptable quality of service is undoubtedly of very high priority. The use of an overlay network above the physical network layer may address the performance issues and provide a solution without the need of changing the underlying architecture of the physical network. Measurements on the physical network performance when overlay Peer-to-Peer network topology design is applied, have been shown that the parameters of the network were improved, and traffic demands addressed efficiently (**Figure 3: P2P overlay network**). To reduce the complexity of the network, designing an optimal overlay network topology requires not only statistical analysis of network key values, but also its necessary to use simulation tools and techniques to study every possible scenario regarding network traffic conditions or node behavior and thus modelling the problem efficiently. Generally, overlay links form paths (may be subsets of physical network nodes) in top of the underlying physical network that allow overlay nodes to establish direct communication. It is desired to find a balance between the cost of making new overlay links, the traffic handling and routing needed (Mina Kamel, Caterina Scoglio, Todd Easton Optimal Topology for Overlay Networks, 2007). Peer-to-Peer networks can be classified by their overlay network topology and by the presence or not of a centralized control entity that handles the resource indexing and monitors peer state. Structured, unstructured regarding the topology. Centralized, decentralized or pure and hybrid if a central control is being applied.

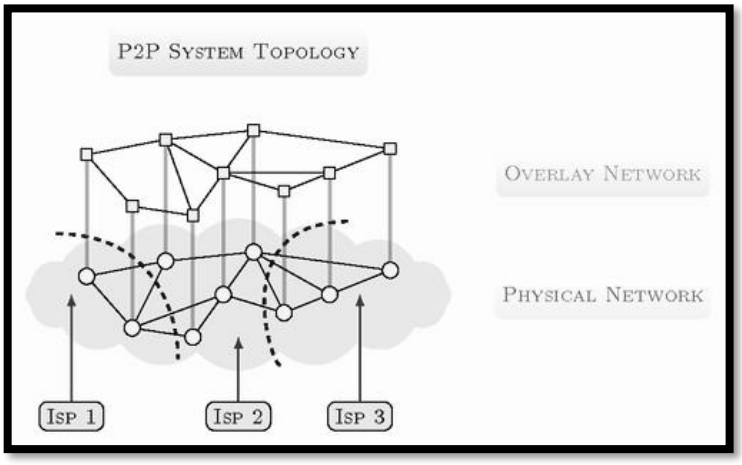


Figure 3: P2P overlay network

i.Structured Peer-to-Peer networks

In structured Peer-to-Peer networks, the topology of the overlay network can be either hierarchical or flat, and the locations of the content is indexed and mapped in dictionaries. It is very easy to locate a file within the Peer-to-Peer network because the information needed is provided immediately by the dictionaries. Peer connection in structured Peer-to-Peer networks considered to be deterministic because it is very important to ensure that the structure of the network will not be altered by changing the place of the nodes. The search algorithm is highly efficient in exact match searches and it is of high importance that when a peer asks for a resource, even if it is of extreme rarity, the resource will be found and

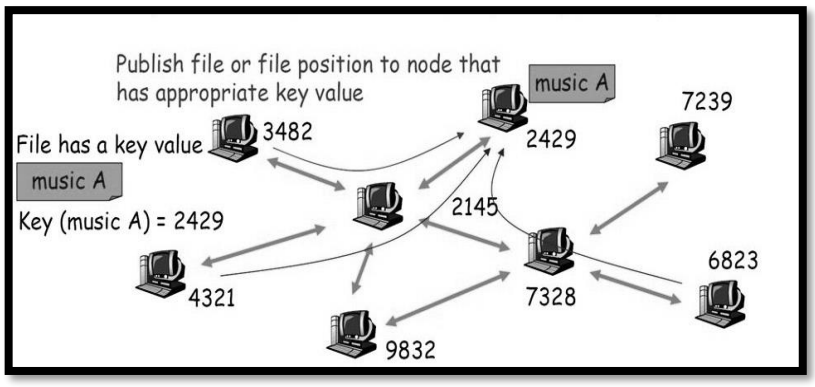


Figure 4: Mapping in Structured P2P

delivered. Constraints in structured Peer-to-Peer networks should be applied to make certain that the rules of hierarchy and positioning of the peers will not be changed. Specifically, in these deterministic systems the indexing (mapping), between a file location (IP address) and the actual content must be always accessible so that the system will respond immediately when a peer asks for content. The most common indexing that is used to structure Peer-to-Peer systems is the Distributed Hash Tables (DHTs). The DHT provides a lookup service with key and value pairs that are stored in the dictionary. When a peer asks for specific content, the system checks in the DHT if this content is available matching the search criteria with the values already stored in the dictionary. If the value found the DHT returns the key which is the IP address or the peer that owns the content and the exchange-transfer will happen after that. (**Figure 4: Mapping in Structured P2P**, **Figure 5:DHT indexing**). System has great response in exact match queries, but as more and more peers are joining the structure more overhead will be created and scalability issues may be occurred because of the number of the requests. Furthermore, a deterministic connection between peers would hinder the performance of the system when populations of peers joining and leaving (high churn rate), because it is hard to maintain the structure (such as neighbor lists, etc.) required for routing to function efficiently. The higher the churn is, the more difficult it becomes for the network to maintain its consistency and because of the limitations of the peers to have a whole view of the overlay network it is necessary to address this side-effect with efficient churn estimation algorithms (Andreas Binzenhofer, Kenji Leibnitz, 2007). Although these systems with highly structured topology are very good in locate the resources when a successful "exact match search" happen, on the other hand their performance is questionable when using simple searching with a keyword that is not a perfect match (Qin Lv, Sylvia Ratnasamy, Scott Shenker, Can Heterogeneity Make Gnutella Scalable, 2002). Most important highly structured DHT-based systems are Chord, Pastry, Tapestry and CAN and although they were all based on common principles, they implemented differently their routing strategies and within them, structure of the nodes varies (Ce Zhu, Yuenan Li, Xiamu Niu, 2010).

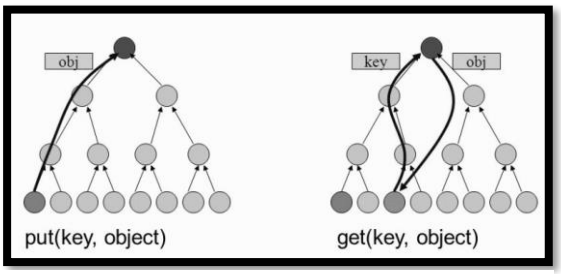


Figure 5:DHT indexing

ii.Unstructured Peer-to-Peer networks

Unstructured Peer-to-Peer systems may be flat or hierarchical and their overlay topology is considered to be non-deterministic. These ad-hoc networks, compared to deterministic Peer-to-Peer systems are resilient in high churn rate, when the transient populations of peers joining and leaving the overlay. These systems can efficiently satisfy complex query searches with various criteria. To spot peers with the desired content, algorithms are used like flooding (e.g. among the super-peers in Kazaa), random walking and expanding-ring (e.g. TTL counter in Gnutella). According to the level of central control and administration applied, we classify unstructured

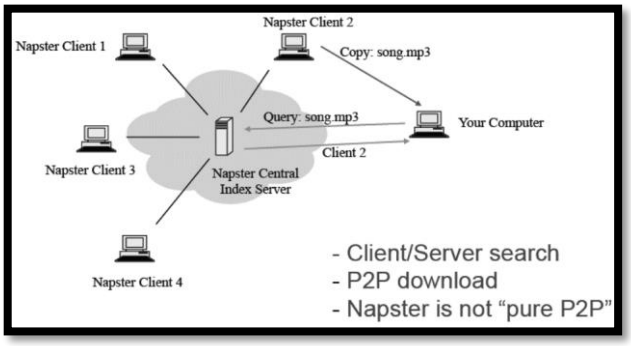


Figure 6:Napster's Central Index Server

Peer-to-Peer networks into three major categories, Centralized, decentralized or pure, and hybrid (Xing Jin, S.-H. Gary Chan Unstructured Peer-to-Peer Network Architectures, 2009).

a. Centralized

In Peer-to-Peer systems where a central control is applied, the indexing of the contents and the administrative functions of the systems take place in this central entity. This centralization of some features of the system does not include any algorithms that define the node position in the overlay because the ad-hoc nature of this unstructured systems could not be deterministic. Most popular Peer-to-Peer platform, Napster's centralized model relies on a central administration server or a group of servers. The main role of these servers is to store meta-data in a central directory, containing info about the place of the content exchanged between peers in the overlay network. (**Figure 6:Napster's Central Index Server**). When a peer enters Napster's platform the main server updates the central directory with the IP address of the peer and the content that this peer makes available for sharing. The database is dynamically updated every time a peer joins the overlay, and it's IP address mapped with the sharing content in its local computer. Napster was not perfect however, it's flaws have to do with this centralized administration. The size of the databases with the meta-information about the content and peers may increase rapidly and as a result, responses to queries of the peers may be slowed down over time and service request bottlenecks may be observed. Adding more servers is an expensive solution and probably temporary. In the worst cases the system may collapse because of the server incapability to handle the requests (Ce Zhu, Yuenan Li, Xiamu Niu, 2010). The BitTorrent protocol is another example of a centralized unstructured Peer-to-Peer overlay network. The difference of BitTorrent platform is the use of .torrent tracker files. Internet users can easily spot and download these tracker files using web search engines or find them published in various internet web sites. When a tracker file is loaded in the BitTorrent client application, a connection is established between the user and the tracker's computer. A list of peers that own parts of file or the whole file is received. The file is split into small data packets with typical size of 256 kb. Then the BitTorrent client software contacts these peers from the list and start downloading different file sections from multiple peers at the same time (**Figure 7:Joining a .torrent**). After a packet of the file is downloaded in the user's computer it is immediately available for downloading by other users(peers) in the Peer-to-Peer network, provided that the user that owns parts of this file will choose to share them (Rajesh Kumar Maurya, Prof Suman Pandey, Vinod Kumar, 2016).

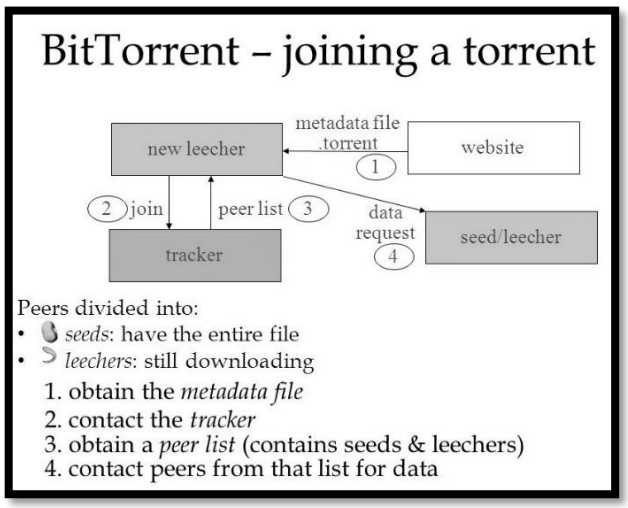
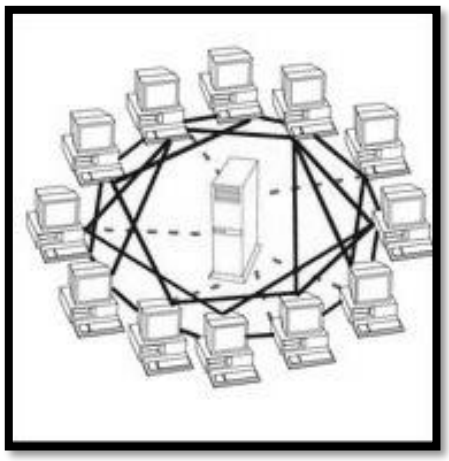


Figure 7:Joining a .torrent



b. Decentralized or Pure

All peers are equally privileged and their roles are similar when the topology of the overlay network is decentralized unstructured. They called servents (blend from server and client, introduced by Gnutella network technology). There are no peers with special administrative roles and the topology referred also as flat. Gnutella is an unstructured decentralized Peer-to-Peer network where the meta-information of the shared content is stored locally in the peers (**Figure 8:Gnutella topology: flat and unstructured**).

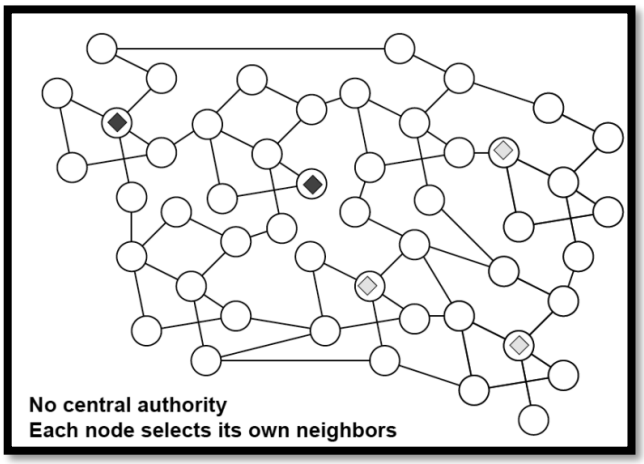


Figure 8:Gnutella topology: flat and unstructured

A peer may join Gnutella network after establishing communication with one peer already joined the overlay, the "bootstrapping" peer. Then, these bootstrap peers send information about their neighboring peers in the "joining" peer including the IP address of these neighbors. Because of this neighborhood formation of peers, searching mechanism in unstructured flat overlays is often a flooding algorithm forwarding queries from neighbor to neighbor (**Figure 9:Gnutella flooding mechanism with TTL**). When a peer is searching for specific files and the query is flooding the overlay network a counter is activated (Time-to-Live Counter). As the flooding continues, the depth of the searching procedure is increased. If there are no results matching the query criteria within the TTL counter time limits, the flood stops. When a query flooding is happening, and the content is found within the time limit, the peer that owns the content send a response indicating that the content is found to the peer that started the query. The response from the peer with the content forwarded to the original peer that started the query using the same path in the opposite direction. Is it possible that many responses will return back to the peer that initiated the query.When multiple responses occurred simultaneously by many peers on the overlay, the query initiator selects one of the peers responded and downloads the desired content through a direct connection on the TCP layer. Gnutella's flat topology without any kind of central administration at the overlay network, is not always efficient and in many cases as the querying rate increases by the peers, the system is incapable to respond with success and performance suffers a critical hit. Query response rate reduced dramatically, and system failure may occur (**Figure 10: Gnutella's Scalability Issues**) This happens mainly because network traffic will grow linearly as the rate of submitted query by peers is increased. Also, in these flat and decentralized topologies rare content actually is very difficult to locate and download. (Ce Zhu, Yuenan Li, Xiamu Niu, 2010). Gnutella's performance suffers from peers with slow transfer speeds and

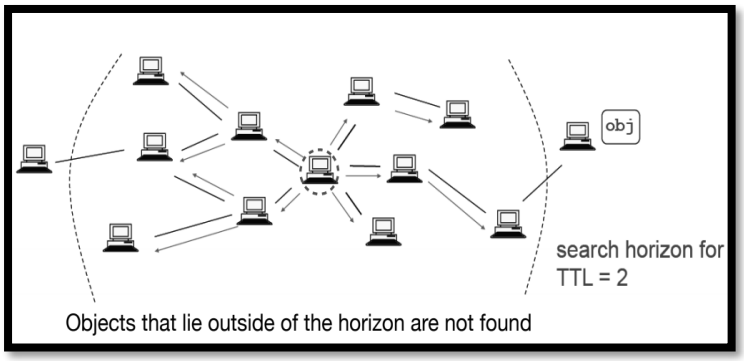


Figure 9:Gnutella flooding mechanism with TTL

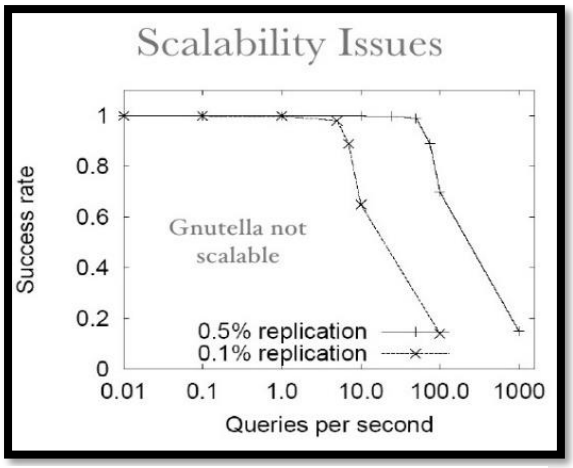


Figure 10: Gnutella's Scalability Issues

also from users who do not like to share content although they are using the platform to download content from other peers. Network's overall responsiveness takes a critical hit as the users with slow transfer speeds increased (Choon Hoong Ding, Sarana Nutanong, Rajkumar Buyya, 2004). In order to stay in Gnutella's network, nodes have to periodically "ping" their neighbors updating the peer list and stay connected only with the "alive" peers. These checks happen within the overlay between peers and create overhead in the network, thus reduce efficiency and performance. Crippled, slowed down nodes may also reduce network performance (Washbourn, 2015).

c. Hybrid

Unstructured overlay network topologies may improve their efficiency and performance by adding the element of central control. Peers with administrative privileges and upgraded roles are supporting the other peers providing resources and services. Gnutella's improved successor (aka Gnutella 0.4) is a hierarchical unstructured overlay with enhanced features. In hybrid Peer-to-Peer overlays a peer in may change roles and from regular client-peer can become an ultra-peer or super-node charged with administrative responsibilities in the network (Ce Zhu, Yuenan Li, Xiamu Niu, 2010). Searching in hybrid Peer-to-Peer systems is much more efficient because of the central control (Beverly Yang, Hector Garcia Molina , 2001). Kazaa protocol, is a hybrid unstructured Peer-to-Peer overlay and the central control is assigned to Super-Nodes(SNs) which have greater responsibilities from the Ordinary-Nodes(ONs). (Oxford Dictionary of English) When an Ordinary-Node joins the overlay is assigned to a Super-Node and a TCP connection is established. It is very important to understand that Kazaa's exploits the heterogeneity of the nodes using a two-level hierarchy with the nodes in higher levels being more powerful in terms of CPU power, connectivity and bandwidth from the ordinary nodes in lower levels. (**Figure 11: Example of Kazaa Network with Super-**).

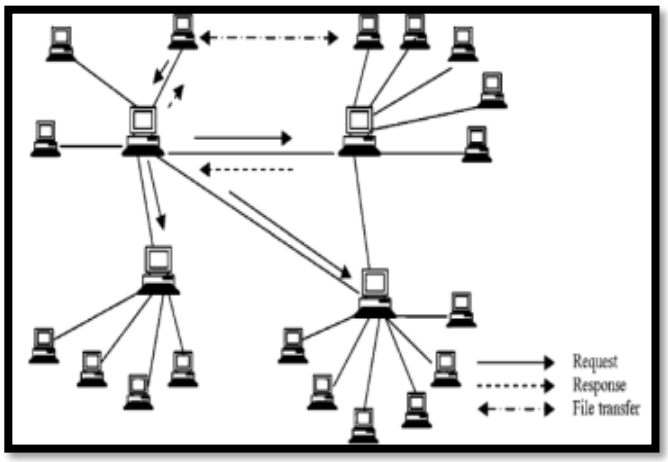


Figure 11: Example of Kazaa Network with Super-Nodes

Super-nodes in KaZaa maintain a database with all information about the content of its children (ordinary-nodes) assigned to them, like meta-data, IP addresses and file identifiers (Rakesh Kumar, 2004). Queries in these networks assigned to Super-Nodes and then it is flooded in the overlay network of the Super-Nodes (Ce Zhu, Yuenan Li, Xiamu Niu, 2010). Hybrid Peer-to-Peer systems designers studied Peer-to-Peer systems in great depth to find a perfect balanced topology. Hybrid means that a system or technology has its origins on two or more inconsistent elements. According to Oxford dictionary something is characterized

"hybrid" when "derived from heterogeneous sources or composed of incongruous elements" (Oxford Dictionary of English). Hybrid Peer-to-Peer systems, because of their mixed character, considered to be inherently better than pure solutions. The weaknesses of both pure and centralized approaches are being mitigated and the heterogeneity of peer population is exploited to form hierarchical overlays and improve the characteristics of Peer-to-Peer network (Darlagiannis, 2005).

V.SECURITY AND TRUST IN PEER-TO-PEER NETWORKS

Security in Peer-to-Peer networks, although thoroughly researched remains always a challenge. In this section we focus on identifying some key security issues regarding Peer-to-Peer networks. The problem's source is that inherently a Peer-to-Peers system, is designed to provide anonymity and the central control in many popular protocols is reduced significantly. Nobody can assure that the identities of the peers in a Peer-to-Peer network are real and without a verification of the identities is difficult to provide security services. Every peer is identified with an alias that is selected by the peer for itself and at the same time is ready to join the overlay. No standard authentication procedure for these pseudonyms is followed and is possible that a malicious user can use more than one aliases at the same time when joining the Peer-to-Peer network. Pseudospoofing, refers to a peer creating and handling multiple pseudonyms. It is possible an attacker to make use of hundreds of pseudonyms (S. Balfe, A.D. Lakhani, K.G. Paterson, 2005).

i.Attacks on Peer-to-Peer Networks

In client-server systems all services provided by the server and when an attack happens usually a single entity is targeted, the server. Server may be secured from attacks or viruses by apply user authentication protocols, registration procedures, cryptography, antivirus software and sophisticated firewalls.

This is not the case in Peer-to-Peer networks. Individual peers are susceptible to attacks but there is no general effect on the whole overlay peer population. An attack may be successful by shutting down peers with specific content that is not available by other peers. To make a Peer-to-Peer network more secure and resilient in attacks or threats every peer should be responsible to trace if content with a virus or malware is forwarded in the network and send warning messages to its neighbor peers when a possible threat is detected (Vasileios Vlachos, Stephanos Anroutselis-Theotokis, Diomidis Spinellis, 2004). Classification on attacks connected to Peer-to-Peer networks can be found in (**Figure 12: Classification of P2P Network Attacks**) (J. Schafer, K. Malinka, P. Hanacek, 2009). "Since each overlay node plays a role in routing traffic through the network, malicious users can perform a variety of *routing attacks*, or *denial of service attacks* (DDoS). Example of common routing attacks include: *incorrect lookup routing* when malicious nodes deliberately forward requests incorrectly or return false results, *incorrect routing updates* when malicious nodes corrupt the routing tables of neighboring nodes by sending them false information and *incorrect routing network*

Type of attack	Attack Example
Attacks on Peer-to-peer network	• Listening queries • Filtering queries • P2P network disintegration
Attacks realized through peer-to-peer networks	• Malware spreading • DDoS attack • Setting up Botnets
Attacks on users of Peer-to-peer network	• Content Verification • Anonymity weakening • Stealing Identity

Figure 12: Classification of P2P Network Attacks

partition, when new nodes are joining they bootstrap via a malicious node, which places the new node in a partition of the network that is populated by other malicious nodes" (Quang Hieu Vu, Mihai Lupu, Beng Chin Ooi, 2010).

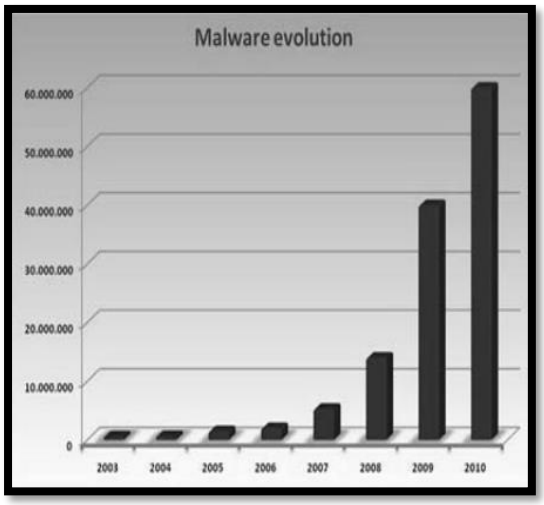


Figure 13: Malware Evolution

ii.Malware in Peer-to-Peer Networks

Not only viruses and worms are a major threat for internet users today. Malware is software intentionally designed to harm computer and slow it down hindering its network performance (**Figure 13: Malware Evolution**). It can take control of the internet browser and install malicious scripts or suspicious search engines and track the history of the web sites visited by the user. Computers can easily be infected by malware. Often, these annoying malicious scripts comes bundled with other programs (Kazaa and other file sharing programs seem to be the biggest bundlers), and they are penetrating in Peer-to-Peer file-sharing systems (Prof.Puram.Pradeep Kumar, Naini Shekar Reddy, Saleha Saudagar, T. Puneeth Chandra, Ch. Kishor Kumar , 2012). A study analyzing traffic on the Kazaa network found that 15% of the 500.000 file sample taken were infected by one or more of the 365 different computer viruses that were tested for (Jan Goebel, Thorsten Holz, Carsten Willems, 2007). Corrupted data can also be distributed on Peer-to-Peer networks by modification of shared content. On the FastTrack network, the RIAA, managed to hack downloaded music files and infect them with malware. Files infected with the RIAA virus were destroyed because of the contained malicious code. The RIAA is also known to have uploaded fake music and movies to Peer-to-Peer networks in order to track and prevent illegal music file sharing (Sorkin, 2003). Consequently, the P2P networks of today have seen significant increase in their security measures against attacks and malware threats and provide sophisticated file validation algorithms. Modern hashing, packet verification and cryptography have made most networks resistant to almost any threat.

VI.CONCLUSION

Peer-to-Peer is an important technology that has been developed and evolved significantly. Resource discovery and content sharing in Peer-to-Peer systems improved and new generations of hybrid Peer-to-Peer systems introduced. A more centralized approach with hierarchical levels in these Hybrid Peer-to-Peer Networks, with peer upgraded roles, improved stability and performance. Emerging collaborative Peer-to-Peer systems are going beyond the era of peers doing the same things while sharing resources. Peer diversity or clusters of peers that can bring unique resources and capabilities to the virtual community and accomplish greater tasks will be beneficial to every individual peer.

VII. References

Andreas Binzenhofer, Kenji Leibnitz. (2007). Estimating Churn in Structured P2P Networks. *Managing Traffic Performance in Converged Networks. Lecture Notes in Computer Science, vol 4516. Springer, Berlin, Heidelberg*.

Beverly Yang, Hector Garcia Molina . (2001). *Compare Hybrid Peer-To-Peer Systems.* Stanford University, Computer Science Department.

Ce Zhu, Yuenan Li, Xiamu Niu. (2010). *Streaming Media Architectures, Techniques and Applications: Recent Advances.* New York: Information Science Reference.

Choon Hoong Ding, Sarana Nutanong, Rajkumar Buyya. (2004, 2 10). Peer to Peer networks for Content Sharing. *Technical Report, GRIDS-TR-2003-7, Grid Computing and Distributed Systems Laboratory, University of Melbourne, Australia, December 2003*.

Darlagiannis, V. (2005). Hybrid Peer-to_Peer Systems. Στο R. W. Steinmetz, *Peer-to-Peer Systems and Applications.* Springer.

Gene Kan, edited by Andy Oram. (2001). *Peer-to-Peer: Harnessing the Power of Disruptive Technologies, chapter 8 : Gnutella.*

Gera Jaideep, Dr Bhanu Prakash Batula. (2016). Survey on the present State-of-the-Art of P2P Networks, Their Security Issues and Counter Measures. *International Journal of Applied Engineering Research*.

Guiran Chang, Chuan Zhu, Wei Ning. (2008). P2P SIP : Network Architecture and Resource Location Strategy. Στο M. I. Syed A. Ahson (Editor), *SIP Handbook: Services, Technologies, and Security of Session Initiation Protocol* (σ. 614). CRC Press.

Hauben, M. (1995). Chapter 3 - The Social Forces Behind the Development of Usenet. Στο M. H. Ronda Hauben, *Netizens Netbook.*

J. Schafer, K. Malinka, P. Hanacek. (2009). Peer-to-Peer Networks: Security Analysis. *International Journal On Advances In Security, vol 2, no 1*.

Jan Goebel, Thorsten Holz, Carsten Willems. (2007, 7 12). Measurement and Analysis of Autonomous Spreading Malware in a University Environment. *DIMVA '07 Proceedings of the 4th international conference on Detection of Intrusions and Malware, and Vulnerability Assessment*, σσ. 109-128.

Karthikeyan .R, Dr. T. Geetha, Santhini .T, Santhiya .R. (2017, 8). Classification of Peer-to-Peer Architectures and Applications. *IJESC Volume 7 Issue No.8*.

Mina Kamel, Caterina Scoglio, Todd Easton Optimal Topology for Overlay Networks. (2007). *NETWORKING 2007. Ad Hoc and Sensor Networks, Wireless Networks, Next Generation Internet: 6th International IFIP-TC6 Networking Conference, Atlanta. (Lecture Notes in Computer Science).*

Nelson Minar and Marc Hedlund, edited by Andy Oram. (2001). *Peer-to-Peer: Harnessing the Power of Disruptive Technologies, chapter 1 : A network of peers.*

Ou, Z. (2010). *Structured peer-to-peer networks : hierarchical architecture and performance evaluation.*

Oxford Dictionary of English. (χ.χ.). Oxford dictionaries.

Philip Kisembe, Wilson Jeberson. (2017, 8). Future of Peer-To-Peer Technology with the rise of Cloud Computing. *International Journal of Peer to Peer Networks(IJP2P) Vol.8*.

Prof.Puram.Pradeep Kumar, Naini Shekar Reddy, Saleha Saudagar, T. Puneeth Chandra, Ch. Kishor Kumar . (2012). Analysis and Prevention of Malware in P2P. *(IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 3 (1)*, σσ. 3162 - 3169.

Qin Lv, Sylvia Ratnasamy, Scott Shenker, Can Heterogeneity Make Gnutella Scalable. (2002). *Peer-to-Peer Systems: First International Workshop, IPTPS Peter Druschel, Frank Kaashoek, Antony Rowstron (Eds).* Cambridge, MA, USA: Springer.

Quang Hieu Vu, Mihai Lupu, Beng Chin Ooi. (2010). *Peer-to-Peer Computing: Principles and Applications.* Springer.

Rajesh Kumar Maurya, Prof Suman Pandey, Vinod Kumar. (2016, 4). A survey of peer to peer networks. *International Journal of Advanced Research in Computer and Communication Engineering*.

Rakesh Kumar, J. L. (2004, June). Understanding KaZaA.

S. Balfe, A.D. Lakhani, K.G. Paterson. (2005, 12 12). Trusted Computing: Providing Security for Peer-to-Peer Networks. *Peer-to-Peer Computing, 2005. P2P 2005. Fifth IEEE International Conference on*.

Sorkin, A. R. (2003). Software bullet is sought to kill music piracy. *New York Times*.

Vasileios Vlachos, Stephanos Anroutselis-Theotokis, Diomidis Spinellis. (2004, 6). Security Applications of Peer-to-Peer Networks. *Computer Networks, 45(2):195–205*.

Washbourn, L. (2015). *A survey of Peer-To-Peer Network Security.* Arxiv.

Xing Jin, S.-H. Gary Chan Unstructured Peer-to-Peer Network Architectures. (2009). Unstructured Peer-to-Peer Network Architectures. Στο *Handbook of Peer-to-Peer Networking pp 117-142.*