SCIENTIFIC and TECHNOLOGICAL COOPERATION

between

RTD ORGANISATIONS in GREECE

and

RTD ORGANISATIONS in U.S.A, CANADA, AUSTRALIA, NEW ZEALAND, JAPAN, SOUTH KOREA, TAIWAN, MALAISIA and SINGAPORE

SecSPeer: Secure and Scalable peer-to-peer computing and communication systems

Contract No. ΗΠΑ-021

Deliverable 1.1: Requirement Analysis

Abstract: *Peer-to-peer systems allow users to connect to each other, collaborate and share their resources in a scalable and flexible way. Their popularity and their architectural and deployment challenges have raised the attention of both the scientific and business community. The aim of SecSPeer, Secure and Scalable Peer-to-Peer Systems, is to study the existing systems, identify their strengths and weaknesses and provide solutions to open issues, like security and scalability.*

Contractual Date of Delivery	*29/12/2004*
Actual Date of Delivery	
Deliverable Security Class	*Public*
Editor	*E.P Markatos*
Contributors	*Antonatos Spiros, Tsigkos Dimitris, Gasparis Stelios*

The SecSPeer Consortium consists of:

FORTH-ICS	Coordinator	Greece
University of Pittsburgh	Partner	United States of America
Virtual Trip Ltd.	Partner	Greece

Table of Contents

1 Introduction

Peer-to-peer systems are gaining increasing attention because of their unstructured nature and the popular applications built upon them. Leaving the rigid client-server model [6], peer-to-peer systems allow users to form an overlay network, providing ways for direct communication and resource sharing [21, 22]. While in client-server model, a client put its request to a predefined server and expects a single answer, in a typical peer-to-peer system a client's request is broadcasted to several clients and multiple responses are given back to the client, permitting him to connect to other clients and use their resources. Building a peer-to-peer system is a challenging process that is affected by numerous parameters. SecSPeer comes to provide both architectural designs and practical solutions to these challenges.

One of the most popular applications of peer-to-peer systems is file sharing such as in the Kazaa, Gnutella, DirectConnect and Morpheus systems which serve hundreds of thousands of users and several terabytes of shared files each day. Recent measurements have shown that most of the Internet traffic is caused by peer-to-peer systems [19], raising the interest of both administrators and researchers. However, Peer-to-peer systems have applications reaching far beyond file-sharing. For example, file storage systems, like Freenet, use peer-to-peer technologies to provide remote storage services. Furthermore, the distributed nature and the large number of peer-to-peer clients are suitable for distributed computation, in a sense like peer-to-peer grids like Seti@Home project [10], and collaboration, like Groove Networks [11]. Although such systems are not widely deployed, their challenges and scalability issues are of high interest.

The aim of SecSPeer is described by its name: ***Sec***ure and ***S***calable ***peer***-to-peer systems. The scope of SecSPeer is, however, not limited to the security and scalability aspect but extends to expressiveness and quality of service issues. While there are numerous proposed ways that cover each aspect separately, finding a solution that will glue up all pieces together is a process that comes in face with various trade-offs. In the following subsections, we will (i) try to provide a detailed overview of these issues, (ii) spot the trade-offs and (iii) describe the characteristics of a secure and scalable peer-to-peer system.

2 Terminology

When referring to peer-to-peer systems there is a number of terms that need to be defined. The terms **node**, **client**, **peer** and **user** all refer to an entity connected to a peer-to-peer systems. **Topology** means the way clients are connected to each other, and **overlay network** is a network built upon Internet and includes a set of clients connected to each other independently of the routing at the IP level.

3 Background

The history of peer-to-peer systems is relevantly recent. The explosion of these systems began with file-sharing systems such as Napster [3], Gnutella [5] and Audiogalaxy [14]. The first of these clients was Napster, which appeared in early 1999 and reached mainstream popularity within a few months. Its popularity made it appear in major headlines of technology and financial magazines (Figure 1). Kazaa [15], Morpheus [4], BitTorrent [16] and DirectConnect [17] are more recent systems that have also met great popularity. Gnutella, Kazaa, Morpheus and BitTorrent are all unstructured networks. Unstructured networks have no specific topology which makes them resilient to attacks. Napster and Audiogalaxy were using centralized indexing services, facing the problem of single point of failure. In the last years, structured systems have been introduced, like Chord [1] and CAN. Structured networks have a predetermined topology but are more vulnerable to attacks. Most widely deployed and used peer-to-peer networks, nowadays, are unstructured with no specific topology.

Figure 1 : Napster was the first peer-to-peer system that enabled users to share their, acquiring thousands of users in a short time.

4 Desirable characteristics and challenges

4.1 Scalability

The notion of peer-to-peer systems is closely related to scalability. The scalability issue has nowadays great impact, especially if we consider that thousands of users use peer-to-peer systems and a considerable large amount of traffic on the Internet is generated by these systems. While in the previous decade, most of the traffic was web pages and e-mail, they now are a small portion of the traffic. As shown in Figure 2, file-sharing clients like Kazaa and Gnutella dominate on the Internet traffic. Unstructured peer-to-peer systems that do not impose any rigid topology among connected clients permit thousands of users to connect to such networks. The main advantage of such systems is that connecting and disconnecting to them is an easy process that requires no extra communication with other clients.

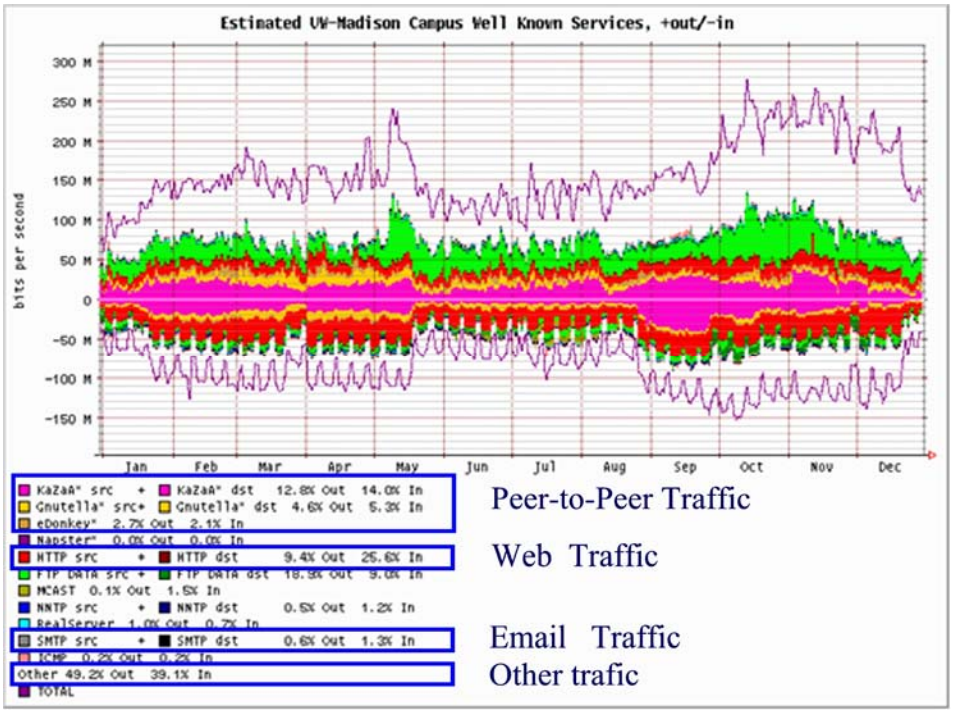

Figure 2 Internet traffic distribution for the University of Wisconsin for the year 2002. Incoming HTTP traffic (World Wide Web) comes up to 25,6% of total traffic, while peer-to-peer systems (KaZaA, Gnutella, eDonkey) take over the 21,4% of traffic. Furthermore, a large 39,1% part is classified as "Other" but it is considered as peer-to-peer traffic sent to dynamic ports.

A severe drawback is that their performance in terms of bandwidth consumed and user-perceived delay is still poor, limiting their scalability. Gnutella [5], for example, permits every client to have a set of other clients, called neighbors, that he can communicate with directly. Every request expires after it reaches a number of clients, a value defined by a time-to-live field. A client broadcasts a request - in a tree-like way - its neighbors and the neighbors to their neighbors and so on, until a time-to-live is reached. In Figure 3 and 4, a simple example of how queries are transferred into a Gnutella network is displayed. As a result, the network is flooded with requests and response may take a long time to go back to the client that made the request. Even worse, the original client may take no response back as the response was located to clients that could only be reached with higher time-to-live values. Optimizing the search mechanism in order to avoid flooding has been studied but such mechanisms have not been deployed on real systems [13]. On the other hand, systems like Chord [1] define a rigid topology that allows clients to take an answer in logarithmic time against to the number of clients connected to the system, achieving high scalability. An example of Chord's topology can be viewed at Figure 5, where all clients are given a unique identifier and put into a logical circle. Files are also hashed to identifiers and mapped to the closest client to the circle. Their main drawback is that connecting and disconnecting to these systems imposes communication overhead and engagement of propagation mechanisms. Additionally, systems like Chord do not permit searching for wildcards but only for specific keywords, raising a trade-off between performance and user-perceived quality of service. More recent peer-to-peer systems, organize their clients into hubs according to interests but this organization limits the number of results to responses. Hybrid architectures also exist, where clients can connect to each other and cooperate with a set of servers to locate data and peers. Skype, a voice over IP application, follows this model. SecSPeer should provide a flexible design in order to maintain the good characteristics of unstructured systems but also providing guarantees for fast

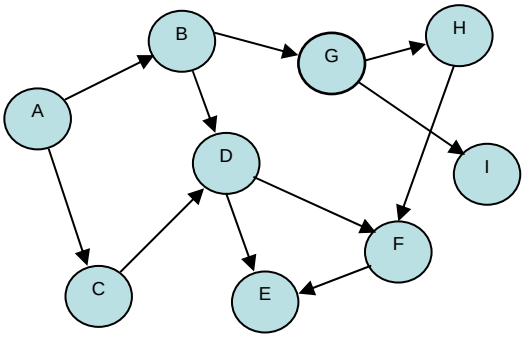

Figure 3. An example of Gnutella overlay network. Arrows indicate neighbor relationship. For example, A has B and C as its neighbors

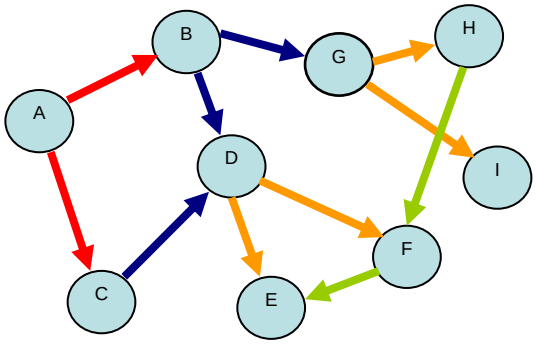

Figure 4. Client A makes a query to his neighbors B and C (red line). B and C forward the query to their neighbors (blue line), them to their neighbors (orange line) and so on

responses and respect to network performance. Caching is an open-ended issue that can reduce the cost of searching in an unstructured network and should be examined carefully throughout the SecSPeer project. Careful design and placement of peer-to-peer caches can dramatically reduce the network cost of file-sharing, in terms of messages and bandwidth allocation. Experiences from web caches might help but caching in a peer-to-peer system shall be viewed from a different perspective, as the number and size of files exchanged in a peer-to-peer system differ dramatically from web pages. Hardware solutions that provide caching for various peer-to-peer protocols exist but their high cost and limited extensibility constitutes them prohibitive. Peer-to-peer caches using commodity hardware and appropriate software, similar to traditional web caches, were also proposed, taking advantage of the data locality and forcing clients to keep copies of frequently demanded files [12] but their performance is poor and are based on unrealistic assumptions, like that all clients are willing to contribute disk space.

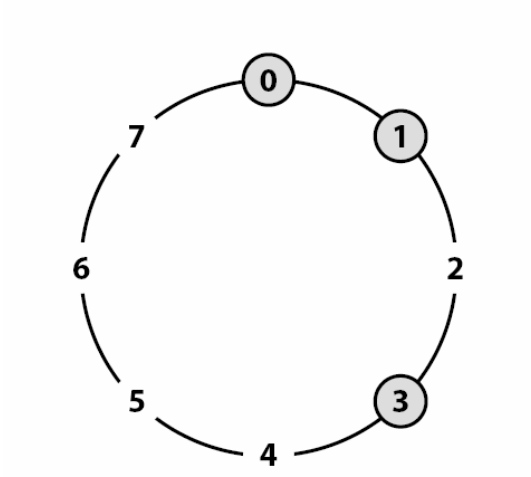

Figure 5 An example of Chord topology. Nodes (0, 1 and 3) are placed in a circle. Files are hashed into keys and keys are mapped to closest node of the circle. For example, if file is hashed into key 2 then node 3 is the closest and has the file.

4.2 Security

Security in peer-to-peer systems interacts with four major dimensions: (i) availability, (ii) anonymity, (iii) authenticity and (iv) access control. All of these dimensions are desirable characteristics of SecSPeer but there is need for careful design in order to combine them. A peer-to-peer system should provide all guarantees that it will be always available even if some clients are under attack and that all quality-of-service requirements are satisfied so as its clients do not get disappointed. These guarantees must be provided in respect to performance issues and be based on real-case conditions. Although a respectful amount of work has been done on each dimension separately, a complete solution that covers all issues together has not yet been proposed. SecSPeer aims at such a solution and additionally tries to prevent a peer-to-peer system from becoming an underlying platform for attacks outside the overlay network.

4.2.1 Availability

Availability is related to the degree of tolerance of a peer-to-peer system against faulty or unreachable nodes. A peer-to-peer system must be resilient to clients that do not respond to requests or misbehave. Although one might expect that clients will always be up and running, in real-case scenarios, clients may be unavailable, for

example due to attacks. Indeed, in the case of web servers, a client may become victim of denial-of-service attack, where he is receiving a flood of messages losing all his available bandwidth. For example, in the Gnutella p2p system, malicious clients could become super-nodes (clients with additional roles) that redirect all requests to a victim, in order to flood it with information and to cripple its normal operation. Additionally, complex queries that require large amounts of processing time may be used to attack on the CPU availability of a client, which in turn may spend a lot of processing time to examine the query. Quality-of-service can be also attacked, for example when a client serves a file slowly and the receiver gets disappointed and disconnects or when it serves a wrong file. Techniques for detecting availability failures have been developed but require high communication overhead and assume pairwise connectivity that is not realistic.

4.2.2 Anonymity

Anonymity aims to protect the privacy of clients. Recently, legal issues have aroused concerning illegal trade of files and their financial impact on media producers. As peer-to-peer systems are not targeted specifically for file-sharing only, censorship resistance is required. There are several types of anonymity, like hiding the owner of a file, hiding the clients that access a specific file, and/or hiding the clients that have a copy of a file. Several systems like Free Haven [18], Freenet [2] and Crowds [9] have achieved anonymity through proxies but there is a tradeoff between anonymity and performance. Adding additional layers to achieve anonymity increases latency to the client's side. Also most of the anonymity protocols do not behave well under attack. Inspired from ants, which transfer their food hand by hand, recent advancements propose a similar way, where multiple clients cooperate to transfer a file from its source to the one that requested the file. The idea, however, is novel and needs further study, in terms of performance and how well the privacy is preserved.

4.2.3 Authenticity

In order to attack the normal operation of peer-to-peer systems, malicious peers may provide false (non-authentic) responses to other client's requests. A peer-to-peer system should be able to distinguish in real-time which responses are authentic and which are fake. There are four main approaches to the definition of authenticity [20].

The first one considers the oldest copy of a file as authentic. Although there are timestamping systems based on that definition, their deployment is limited. The second one is expert-based, where authoritative nodes take the decision of which files are authentic based on offline digital signature schemes. Their main drawback is that there are single points of failures. An extension to expert-based systems is voting-based where the votes of experts are gathered to form a decision. Such systems need additional protection from vote spoofing. The last approach to authenticity is reputation-based systems where votes are weighted but they introduce administrative cost for maintenance and propagation of weights.

4.2.4 Access Control

A peer-to-peer system should be able to restrict accessibility to files and resources. Up to now, peer-to-peer systems cannot enforce copyright laws and as a result these laws are violated. The access control comes as a solution to free and uncontrolled distribution but limits the utilization and scope of the system. Fine-grained access control mechanisms are needed to customize the utilization of a peer-to-peer system.

4.2.5 Shielding the peer-to-peer infrastructure

A peer-to-peer system cannot be subjected to an attack itself but it can become the underlying infrastructure for launching an attack to the Internet. It can be used as a pool of hosts that can be compromised to spread a virus or even participate to a Denial of Service attack. A real example can be taken from Kazaa, where vulnerability was found in the client software that could lead to host compromise. Hopefully, the vulnerability was fixed before it could be misused but imagine what would happen if two million users of Kazaa were all compromised. Furthermore, an attacker can gather very quickly a large set of IP addresses in order to build a hitlist and spread a worm. Our recent measurements, using the Gnutella network, showed that a 24-hour period suffices for gathering two hundred thousand unique IP addresses, expecting that this number can be much higher using other more popular clients. Shielding the peer-to-peer infrastructure is a necessity that guarantees that the peer-to-peer system will be used for the purpose it will be used. In SecSPeer, shield mechanisms will be proposed and examined thoroughly to protect the peer-to-peer systems from uwilling intentions.

4.3 Expressiveness and quality of service

A desirable characteristic of SecSPeer is the expressiveness of its search mechanism. A simple key lookup is not expressive enough for most of the users, which want to put their queries in form of keyword queries including wildcards or partial searches. Systems like Chord support only key lookup in order to achieve logarithmic time in locating the data by using the key as a guide to the overlay route. SecSPeer should not be limited to simple key lookups but should address systems that allow substring queries. On the other hand, such queries may adversely affect the performance of the system, like in case of PIER system where the support of SQL as a query language caused performance hotspots [7]. Support of aggregates like summation or count is also desirable.

Expressiveness is part of a more general aspect, this of quality-of-service. Besides expressiveness, quality-of-service includes the number of results to a request, the delay of responses and the satisfaction of the client. Fast but limited answers are usually not desirable while on the other hand a user cannot wait long for a complete set of answers. Partial search systems try to locate a subset of responses and not the complete answer and in parallel gather as many responses aiming at exceeding a threshold that satisfies the user. SecSPeer should take under consideration the user-perceived qualities and adapt its mechanisms to satisfy them in respect to system's performance and extensibility.

4.4 Adaptation to new applications

Peer-to-peer systems have traditionally been used for file sharing among peers. Their flexibility and scalability, however, can lead to support of new applications that cannot be built with existing technology. Such an application is the distributed access and processing of data in real-time. Another application is discovery of malicious attacks, like worms and viruses. Nowadays, on the Internet, there are thousands of systems that collect information about the security of the systems they guard. Such systems can be firewalls, anti-virus systems and Intrusion detection systems. Composing data from all these systems, one could information for the global security status of the Internet, like the geographical distribution of a virus spread and the

creation and spread of novel worms. Because of the huge amount of data produced by these systems, centralized solutions for data gathering and processing is prohibitive. On the contrary, a peer-to-peer system can locate and filter the information faster as the needed operations are performed near the information source. Peer-to-peer business value and exploitation

In addition to providing ways for flexible file-sharing and better collaboration, peer-to-peer technology is finding new roles in the enterprise world. The decentralized nature of peer-to-peer systems offers to companies a variety of systems to choose in order to achieve desktop-to-desktop collaboration, remote storage, business process management and composite applications.

Desktop-to-desktop collaboration, in particular, has great potential, allowing professionals in different vertical markets around the globe to work on the same project, thereby increasing productivity and decreasing travel costs. Despite security concerns and a misunderstanding of peer-to-peer multiple forms and levels of control, the benefits for connecting people, resources and processing power are slowly emerging from the hype.

4.5 Examples from the real world

An example of financial exploitation comes from the Kazaa client. Kazaa comes embedded with software that allows companies to deliver advertisements. Having in mind that such applications are used by thousands of users, commercial exploitation is very effective. Another example comes from Groove Networks, which provides Groove. Groove is a commercial software product that lets users collaborate in real time from anywhere. As long as users download the software and are online simultaneously, Groove lets them collaborate remotely on anything from document authoring to complex code writing. Peer-to-peer Voice-over-IP is another example that may have business benefits, as indicated by the Skype application. Companies that own copyrighted files may join a peer to peer network if they can sell their copyrights. Apple has introduced itunes, where you can buy a song for $ 0.99. A peer-to-peer system where you can buy music, videos, books or other resources could have commercial value. A membership peer-to-peer where users can share computation resources (like Seti@Home).

4.6 Business opportunities and requirements

Peer-to-Peer systems are usually perceived as “on-line community utilities” that are mostly used for very useful but non-commercial activities. More over, one could say that peer-to-peer systems are recognized as “anti-business” systems, mainly because of cases like Napster. The popularity of P2P networking among home consumers is well known, but that popularity does not easily translate to enterprise users. Enterprise resists to adopting P2P networking on corporate LANs and therefore slow the adoption rate of the technology among business users. Frost & Sullivan forecasted that by the end of 2001 the U.S. market would have approximately 61,410 enterprise users of some form of P2P networking technology. This number is expected grow substantially to 6.2 million enterprise users by 2007. As these numbers grow, so will the revenues within the markets they serve. In the content delivery market, Frost & Sullivan predicted that revenues would total $840,185 in 2001. By the end of 2007, P2P networking technology is expected to help generate over $422.9 million with a compound annual growth rate (CAGR) of 182 percent. Similar positive growth patterns are expected in the following markets:

- **P2P Supply Chain Management**: $446,020 in 2001 and $365.6 million by 2007 for a CAGR of 205.9 percent
- **P2P based Business Exchanges**: $372,960 in 2001 and $2.59 billion by 2007 for a CAGR of 336.8 percent
- **P2P-based Collaboration Solutions**: $39.4 million in 2001 and $976.7 million by 2007 for a CAGR of 70.8 percent
- **P2P-based Knowledge Management**: $2.6 million in 2001 and $604.4 million by 2007 for a CAGR of 147.8 percent
- **P2P-based Search Engines**: $35,000 in 2001 and $1.4 million by 2007 for a CAGR of 84.2 percent

The only market with significantly more revenue potential over the forecast period is P2P networking-based securities trading. Revenues in that market were expected to exceed $53 million in 2001, and by 2007 are expected to surpass $4.5 billion, for a CAGR of 110 percent. Enterprise P2P networking will become more prevalent in the work place over the next seven years; however the distrust of P2P networking technology will restrain its growth potential. P2P proponents forecast that businesses can save billions by using distributed computing setups that take advantage of unused

bandwidth and resources. Messaging tools and affinity communities can open up intellectual property and data that are otherwise hidden in departmental offices and servers. Many also see P2P computing as a solution that will relieve network bottlenecks, unleash vast amounts of computing power from underutilized processors throughout an enterprise, and enhance collaboration within workgroups, both inside and outside the organization.

The SecSPeer project need to take business / commercial exploitation into account, since it is the critical factor for the getting peer-to-peer systems in the value chain of ICT and ICT-related industries.

4.6.1 Business models

It is required that the following business models should be supported:

- **Pure-license-based**

In this case, one can just purchase an instance of a client and through this will connect to a number of peer-to-peer networks that are “associated” to this package.

- **Pure-subscription-based**

In this case, one gets for free the client(s) s/w and pays subscription fees proportionally to which peer-to-peer network(s) is connected and other qualitative and/or quantitative criteria.

- **Hybrid (license & subscription)**

The Hybrid model is just the combination of the above two models.

- **Consulting**

The Consulting business model is probably the more interesting among the four proposed models. It is considered that a *“Free Peer-to-Peer Service Infrastructure”* exists, where a number of services are provided through various peer-to-peer networks. However, consulting services are required in order to make the offered services useful for each participating organization. For example, Skype-like P2P

telephony could be incorporated in a company having offices at several sites. This would require major integration and consulting services.

4.6.2 Technical requirements for support business goals

SecSPeer project needs to address the issues of ***service monitoring***. Since appropriate tools for monitoring, as well as for authentication and authorization are available, the ***billing*** issue, mast be addressed, especially for subscription-based business model.

Distributed and trustworthy management of billing and authentication / authorization services present a technical challenge to **SecSPeer** project, with major business implications.

Finally, interoperation with trust-management systems, including distributed reputation and recommendation systems, would be highly desirable from a business perspective, since content- and service-trust are required for business exploitations.

4.6.3 The GRID / Peer-to-Peer business opportunity

Peer-to-peer systems suffer from low quality-of-service, which is an obstacle for a number of commercial applications. This happens mainly because of the internet infrastructure, which is, in a large extent, not reliable.

The GRID paradigm, which is widely accepted by ICT industries, will provide a solid infrastructure for the deployment of services. However, a well-know disadvantage of GRID-based systems is low flexibility.

A peer-to-peer approach for the deployment of e-Services on GRIDs would probably overcome this serious problem. Moreover, this will lead to the creation of business synergies, since major ICT vendors, including IBM, Oracle, SUN, etc, have invested in GRID Computing and would be very interested in getting a channel for deploying services targeted to the market of peer-to-peer systems.

5 Bibliography

[1] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H.Balakrishnan. “Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications” In ACM SIGCOMM, pages 149-160,August 2001.

[2] Ian Clarke, Oskar Sandberg, Brandon Wiley and Theodore W.Hong
”Freenet : A Distributed Anonymous Information storage and Retrieval System”
Designing Privacy Enhancing Technologies: International Workshop on Design Issues in Anonymity and Unobservability , LNCS 2009

[3] Napster website. *http://www.napster.com*

[4] Morpheus website. *http://www.musiccity.com*

[5] Gnutella website. *http://www.gnutella.com*

[6] Edelstein, Herb. “Unravelling Client/Server Architecture” *DBMS 7*, 5 (May 1994): 34(7).

[7] R. Huebsch, J. M. Hellerstein, N. Lanham, B. T. Loo, S. Shenker, and I. Stoica. “Querying the Internet with PIER” In *Proc. VLDB*, May 2003.

[8] Daswani, N., Garcia-Molina, H. “Query-flood DoS Attacks in Gnutella” Proceeding of Ninth ACM Conference on Computer and Communications Security, Washington, DC (2002)

[9] Reiter, M.K., Rubin, A.D. “Crowds: anonymity for Web transactions” ACM Transactions on Information and System Security, 1(1):66{92 (1998)

[10] Seti@Home website. *http://setiathome.ssl.berkely.edu*

[11] Groove Virtual Office website. *http://www.groove.net*

[12] Li Xiao, Xiaodong Zhang, Artur Andrzejak, Songqing Chen “Building a Large and Efficient Hybrid Peer-to-Peer Internet Caching System” IEEE Transactions on Knowledge and Data Engineering, Vol. 16, Issue 6, p. 754-769, June 2004

[13] D. Zeinalipour-Yazti, Vana Kalogeraki, Dimitrios Gunopulos “Exploiting locality for scalable information retrieval in peer-to-peer networks” IEEE CiSE Magazine , Special Issue on Web Engineering, IEEE Publications, pp.12-20., July/August 2004

[14] Audiogalaxy website *http://www.audiogalaxy.com*

[15] Kazaa website *http://www.kazaa.com*

[16] BitTorrent website *http://bitconjurer.org/BitTorrent/*

[17] DirectConnect website *http://dcplusplus.sourceforge.net/*

[18] Roger Dingledine, Michael J. Freedman, David Molnar. “The Free Haven Project: Distributed Anonymous Storage Service” In Proceedings of the Workshop on Design Issues in Anonymity and Unobservability, July 2000 (LNCS 2009)

[19] Thomas Karagiannis, Andre Broido, Michalis Faloutsos, kc claffy “Transport layer identification of p2p traffic” ACM SIGCOMM/USENIX Internet Measurement Conference (IMC 2004), Taormina, Italy, October, 2004.

[20] Neil Daswani, Hector Garcia-Molina, and Beverly Yang “Open Problems in Data-Sharing Peer-to-Peer Systems” ICDT 2003

[21] Matei Ripeanu and Ian Foster “Mapping Gnutella Network” 1st International Workshop on Peer-to-Peer Systems (IPTPS’02), Cambridge, MA, March 2002

[22] Evangelos P. Markatos “Tracing a large-scale Peer to Peer System : an hour in the life of Gnutella” 2nd IEEE/ACM International Symposium on Cluster Computing and the Grid, 2002