Convergence among Peer-to-Peer and Programmable Networks

André Ribeiro Cardoso, Ahmed Serhrouchni
Télécom-Paris, LTCI-UMR 5141 CNRS,
Paris, France +33(0)1.45.81.80.90, +33(0)1.45.81.79.94
{acardoso, ahmed}@enst.fr

Joaquim Celestino Junior
State University of Ceara
1700 Paranjana av., Fortaleza-CE, Brazil
celestino@larces.uece.br

Mikaël Salaün
FranceTelecom R&D, MAPS/MMC, 2
22300 Pierre Marzin Av.
Lannion, France
mikael.salaum@francetelecom.com

Abstract

Nowadays it is unquestionable that Peer-to-Peer (P2P) networks have created a considerable volume of internet traffic generating new service provider challenges. Some application aware mechanisms provide alternative solutions including application-level routing, as well as application-specific resource discovery, and replication. Programmable networks have been researched for years and many studies have been done on network programmability. Although both P2P and programmable networks have their particularities, they deal straightforwardly with distributed systems. In this paper a new concept searching the complete integration among P2P and programmable networks is presented. The originality of this work lies in the complete convergence of P2P and programmable networks with the goal of rendering service providers the ability to quickly create and provide new and customized services in response to market demand. This should be the key factor in determining the performance of these service providers. Programmable networks as infrastructure for P2P traffic identification and optimization are considered and examples like the file push and the management of joining/leaving peers highlight the usefulness of this proposal in the solution of traffic challenges.

1. Introduction

Deregulation and the increasingly number of service providers are stimulating and accelerating the introduction of new services. Competition among service providers has simplified data transmission turning it into an added-value service. The differentiation of service provision will depend on the level of sophistication, degree of flexibility, and speed of deployment of services that a future provider can offer. Therefore it is necessary "to open" the equipment to achieve this goal.

Two main approaches can be considered in the domain of network programmability: active networks proposed by DARPA [1] and open signaling networks proposed by Opensig community [2]. Whereas active networks are networks that execute services programmed in the packets (capsules), Opensig networks are essentially networks that can be programmed to execute services in any packet. The network programmability concept enables the deployment of new added-value network services and the enhancement of the quality of end-to-end application services. However this increases the complexity of current networks, since it features the “opening of equipment” and distributing computation throughout the network.

A Peer-to-Peer (P2P) network may be characterized by the capacity of participants to share their hardware resources (CPU, memory, bandwidth, printers, etc). These shared resources provide the service and content

Proceedings of the Fourth European Conference on Universal Multiservice Networks (ECUMN'07)
0-7695-2768-X/07 $20.00

offered by the network. P2P systems can be viewed as systems that apply P2P networks and may be defined as distributed systems consisting of nodes that can join and leave the P2P network at any time and that have equal capabilities and roles, without any centralized control [3]. They provide interesting features such as wide-area routing architecture, efficient search of data items, redundant storage, permanence, fault-tolerance, self-organization and massive scalability.

P2P and programmable networks are technologies well suited to distributed systems. While the former can be viewed as an overlay network [4] and as such are especially propitious to level 7, the later is more concerned with the level 3 of the OSI stack.

Contribution: By combining these two technologies, a synergic and innovative approach to the field of distributed systems is presented: (i) the use of a P2P system to enhance the service deployment in active networks and (ii), in the other sense, a programmable network to identify and optimize packet flows in P2P networks. The main idea is to attain a complete integration among P2P and programmable networks in order to offer new network architecture to service providers. P2P systems aimed the dynamic deployment of services in active networks were already presented in [5]. This paper's main purpose is to continue this convergence. It presents a programmable network as an infrastructure based on PromethOS [6] for P2P traffic identification and optimization. Some examples like the file push and the management of joining/leaving peers are presented to highlight the usefulness of this approach.

This paper is structured as follows: Section 2 presents P2P and programmable networks interacting in a synergic way. Section 3 shows the architectural aspects of programmable networks as infrastructure for P2P systems. Some applications to demonstrate the usefulness of the proposals in this paper are presented in section 4. Finally, the conclusions of the paper and possible future works are presented in section 5.

2. Synergy among peer-to-peer and programmable networks

The main goal of programmable networks is to develop communication networks that exhibit a high degree of flexibility, extensibility and customizability. Programmable networks are considered in this paper to encompass both Active Networks and Open Signaling approaches.

The convergence between programmable and P2P networks can then be achieved.

2.1. Peer-to-peer systems for active networks

Although there have been several studies of this subject [7] [8] [9], deploying new services in active networks is not as simple as expected. Most active network deployment processes require an explicit specification of which nodes will be used by the application. Complete understanding of the underlying network infrastructure is required by the entity in charge of deploying the code (the administrators of the active networks, the service providers or the end-user), and this inhibits the widespread use of active networks.

[5] emerged as an efficient way to discover active nodes capable of providing the desired services, overcoming the constraint of deploying service codes. Combining the technologies of P2P and active networks, the authors presented an innovative concept involving the use of a P2P system to enhance the service deployment in an active network. The concept of administrative domains has been taken into account in the solution presented, electing one active node as a "control node", which is the contact point for other administrative domains. The P2P system is used by nodes and by "control nodes" to advertise their capabilities inside one administrative domain and among different administrative domains. [5] deals with the control plane of active networks, and based on P2P systems could be classified as a new approach for service deployment.

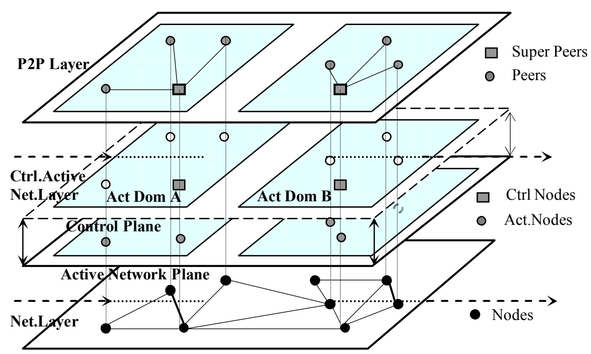

Figure 1. The layered P2P and active network

Active Network Transport System 2 (ANTS 2) with some extensions based on [10] has been widely used. The extended ANTS 2 is an Active Node approach where packets carry references to the code. In classical ANTS 2 architecture, the active nodes introduce new services in the system. However, in the proposed architecture [5] this is a task of control nodes, which must possess the service code. A client must first obtain the code that implements the desired service

Proceedings of the Fourth European Conference on Universal Multiservice Networks (ECUMN'07)
0-7695-2768-X/07 $20.00 © 2007 IEEE

before he is able to use that service and thus anyone should be able to deploy his own service anywhere. An active node which requires a service must belong to an Active Domain, and establish a relation with a control node responsible for that AD.

2.2. Open signaling networks for peer-to-peer systems

Currently, there are three main approaches to characterizing P2P traffic [11]: (i) the application signatures search for protocol specific patterns inside the packet payload, (ii) the transport layer port identification measures and characterizes the P2P traffic by analyzing flow-level at the level of IP address, network prefix, etc. and (iii) network/transport layer P2P pattern recognition based on heuristics uses simple network/transport layer patterns like the packet size distribution of a P2P flow between two peers. Although they have various advantages, all of these approaches have a common limitation; they rely on static techniques that lead to the inability to adapt to new protocols.

Active P2P traffic identification has been used to traverse and gather topological information about different types of P2P networks.

Application Level approaches are strongly dependent on application specific semantics [11]. A programmable networking infrastructure that enables the deployment of specific application aware identification and optimizer components is required. Network-layer controlling techniques do not depend on application protocol internals. The combined use of both network-level and application-level optimization techniques to enforce control mechanisms to optimize the overall network is a promising area of research and yields many integration results.

Programmable networks could play some important roles as infrastructure for P2P systems. The file push and the management of joining/leaving peers are some examples that can well illustrate this proposal and will be addressed in next sections.

3. Architectural design of programmable networks for peer-to-peer systems

[12] aspect considerations suggest the convergence proposal in this paper:

- Preliminary findings show that both locality awareness in P2P networks and interest-based clustering are present and could be leveraged in order to yield significant performance improvements;

- Previous studies evaluate the popularity as the number of requests per file, but in [12] file popularity is measured by its degree of replication.

- This attests of the presence of geographical clustering, i.e., peers requesting a given video file may in a large proportion of cases download it from peers in their own country, thus achieving low latency and network usage compared to downloading it from a randomly chosen peer.

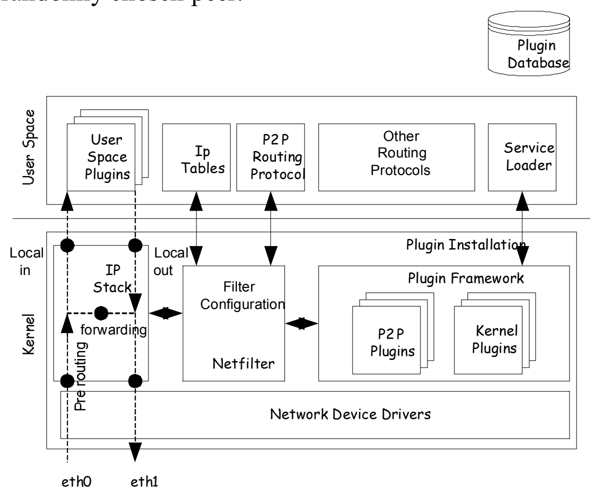

Figure 2. Extended PromethOS node for P2P applications

Therefore, as in [11], programmable networks can be used as infrastructure of level 3 to identify and optimize P2P systems. Since a P2P system was employed to allow the dynamic service deployment on specific active nodes [5], to complete the synergy among P2P and programmable networks, PromethOS [6] was chosen in the proposal of this paper: programmable networks to allow the P2P flow identification and optimization offering "file push" alternatives to increase the download task among participant peers and the management of joining/leaving peers.

The main objective of PromethOS node architecture is to build a modular and extendable networking subsystem that enables the deployment and configuration of packet processing components for specific flows. Based on Opensig approach [2], PromethOS nodes can then be programmed to execute services for any packet passing through the network nodes.

The proposal of this paper is to extend the use of PromethOS node enabling it to play a role as infrastructure for P2P systems (Figure 2).

3.1. Flow identification with programmable networks

Because P2P traffic differs from other internet traffic, it needs monitoring and caching techniques. Characteristics such as the large sizes of P2P files, long duration of sessions, the multiple TCP sessions required for a single P2P download and the fact of the majority of P2P sessions abort prematurely, demand the formulation of a meaningful approach to monitor P2P traffic and devising a unique caching algorithm [13].

From [11], one deduces that the best method to correctly identify P2P flows is to use the "active probing" approach since it reduces the scope of the traffic to be analyzed. Thus for the characterization of the P2P flow, probing peers, called crawlers, are well positioned in the network and issue search requests to collect the IP addresses from the answering peers. Then, active nodes are informed by the crawlers of the machines to be supervised (they maintain an IP list of peers) and they are directly able to look at the sizes of packages and the protocols employed by analyzing the headings. The program in the active node will be able to identify these flows and limit the workload by restricting the number of packages to be analyzed. The node credit could be configured to report near a machine of the operator for example. However, one is satisfied here simply to identify flows and nothing is done to improve the performances of the network or to optimize resources.

4. The "File Push" and the management of joining/leaving peers

P2P filesharing is one of the areas where P2P technology is known as most successful. If one ignores the problem of author royalties and contents, it is a powerful solution because everyone can share all their files with everyone else without having to use an infrastructure which is likely to create problems of scalability and overload.

Nowadays, three main classes of P2P systems may be identified: centralized, decentralized, and hybrid [14]. The main difference between them is the mechanism employed to look up resources in the P2P network. To overcome the single point of failure and the case of strongly required popular files, P2P systems proposed decentralized solutions like Gnutella. However new problems may appear due to the flooding of search requests in such networks. Many studies propose "to structure" these decentralized P2P systems in order to improve the look up service, but this incur significant overheads to maintain the routing table at the application layer.

In this section, an innovative solution is proposed to make use of open signaling networks to manage the "push" of popular files towards other peers and avoid bottlenecks in information flow from a single a source peer. As in [12] this solution considers the geographical locality of file requests, since the push is done toward the nearby peers. We will also look at how one can solve some problems involved in the inopportune join/leave of peers in the network.

4.1. The "File Push" technique

<u>Problem description:</u>

Filesharing applications involving popular files of big size (i.e.: musical, films or albums) shared by an Original Peer (OP) lead quickly to significant overload of requests from remote peers and this has harmful effects on the entire community (Figure 1). OP performance decreases (i.e. flow of transfer, lost of packages, etc.) and the physical links become bottleneck points. As a consequence, P2P application community must wait in long queues for file availability (for at least a part of the file). Moreover, the OP sharing this file will have problems using his connection for other services (i.e. web, email, etc.).

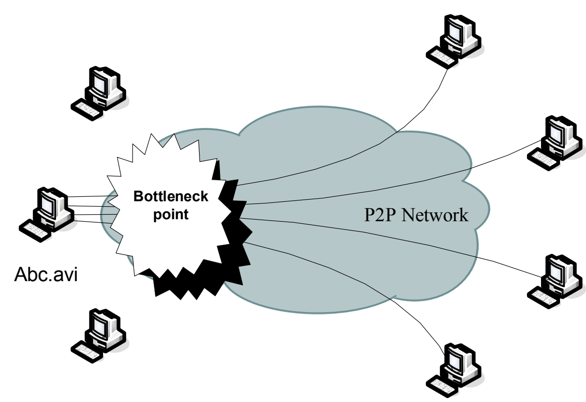

Figure 3. Bottleneck point: e.g. big files shared by a single peer

<u>Solution:</u>

A possible solution to this problem would be to send the file (or at least part of it) towards some peers judiciously selected in order to multiply the number of file sources (Figure 4). One calls this technique "file push".

In this sense, the OP must determine several elements such as the popularity of the file and the geographical clustering of peers requesting the file. Therefore, according to the number of requests from other peers, OP collects statistical data and, based on this information, OP determines: (i) how many peers it needs "to push" the file and (ii) if it pushes the entire file or only parts of it.

Nowadays an effort like this is carried out on the application layer by eDonkey [15] and eMule [16]. However, peers participating in the downloading process are arbitrarily chosen by the server. According to the proposal of this paper, peers (called Alternative Peers (AP)) which will receive the "pushed file" can be selected cleverly on the programmable network layer. To get this information, one can imagine OPs requesting Nearby Active Nodes (NAN) to acknowledge the performance and the state of the network. NANs enable OPs to make a pertinent choice of APs. Unlike classical conditions where the number of requests by OPs would be huge, the proposed file push technique allows OPs to select APs in its neighborhood with the best conditions (i.e. best geographical location, bandwidth, permanent connection, etc.) to upload the requested file.

To summarize, the main goal of this proposal is to have NANs, close to the OP informed that the "requested file" is also available in several other APs. These NANs will "re-route" requests to APs in a transparent way so that OP resources (e.g. bandwidth) will be free for other tasks in the network. One can imagine that the NAN program uses network management information to send the download requests to the most available AP.

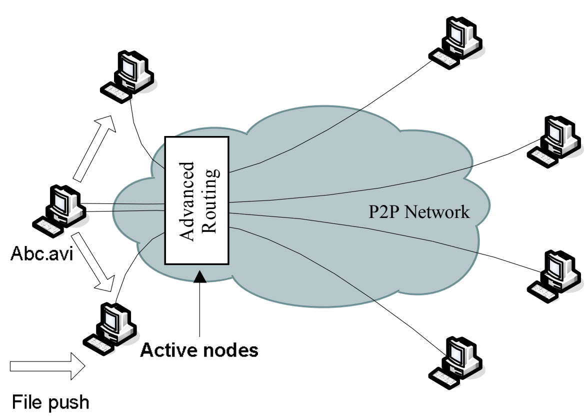

Figure 4. File pushed towards Alternative Peers

Let us now see more precisely how that could happen on the active network level.

- This solution proposes to use the Opensig approach and it is assumed that an active system of network management is running on active nodes/peers.

- Active nodes (i.e. NANs) store a list of the popular files shared on the P2P network and the address of OP and APs for each one of these files (Figure 5).

File	OP	perf	AP1	perf	AP2	perf
Abc.avi	153.176.72.3	3	153.176.72.12	3	153.176.72.4	N/A
Xyz.mpeg	178.233.87.15	3	178.233.87.3	3	N/A	N/A

Figure 5. Fields of each file stored in FileTable

For best performance and to avoid the overload of active nodes which are more critical than OPs, the list of popular files is limited from 50 to 100 files and only one OP and two APs are memorized for each file in the NAN FileTable. For each peer, a performance field of the peer is maintained up to date by the process of management of the network and information collected by active packages as shown bellow. The FileTable thus collected will have a maximum of 100 lines. This value is reasonable to maintain a powerful processing time.

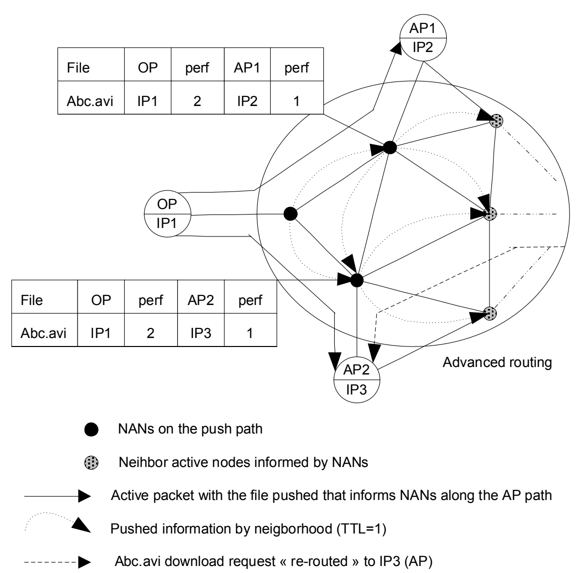

Figure 6. The "File Push" technique and FileTables in Nearby Active Nodes

For each packet, the NAN will access the payload and:

```
if (it is a P2P download request) then
    if (the file is a popular file in the NAN FileTable) then
        the NAN selects the better AP  // i.e. AP with the
                                          better performance
        the NAN re-routes the packet to the selected AP
        the NAN informs its neighbors (TTL = 1) the AP
    else
        classical routing
    endif
```

```
else
    classical routing
endif
```

In order to fill its FileTable, a NAN must know that the file was "pushed" towards other APs by the OP. The most effective way for OPs to do this is to do send an active packet to APs "pushing" the file (e.g. abc.avi) indicating to all NANs along its way that the AP will store the "pushed" file. This will allow the NANs to fill their FileTables. In addition, this active packet will be able to collect performance information providing additional parameters for use in the calculation of the columns "Perf" (Figure 6).

When the file or part of the file is successfully pushed and all FileTables of NANs are filled, the OP and APs begin the file upload and NANs provide a function of distribution of the load charge. It should be noted that the "push" is made near the "powerful" peers which are in the OP neighborhood and not at an unspecified place of the network. Neighborhood is used in this paper to mean two "hop" maximum.

After this initial phase, several techniques can be considered using the same principle. An AP can determine the popularity of a file that it is uploading and when "push" this file towards other APs which will then become OPs. At the routing level, this represents an improvement of the network. When NANs receive the OP active packet, either they already have information on the file stored in their FileTable or they don't. In the former case they will update their performance information fields (perf) if necessary and in the later case they will add a new line in their FileTable with the new information (File, OP, APs and perf).

4.2. The management of joining/leaving peers

Another problem in P2P systems is the management of joining/leaving of peers in the network. Indeed, when connecting to the network, the joining peer needs to know the address of at least one peer in the network to which it wishes to be connected. In many cases, the P2P software has a list of "Super-Peers" at point of entrance to the network. The length of this list ensures the connectivity but, in the case of a decentralized P2P, the concept of "Super-Peer" does not exist. Thus it seems more difficult to set up this method. Then, when a peer disconnects from the network, it must inform the network that it is disconnecting and that the files held by it are no longer accessible. This information is important in order to increase the effectiveness of the network by reducing the number of requests addressed to disconnected peers. However, a peer can lose the connection involuntarily and thus it can not inform the network of its disconnection. Today this is increasingly common due to the arrival of portable computers and other devices with wireless connections.

Active networks can meet these needs favorably:

Concerning the problems of connection, the active nodes can give joining peers the address IP of a device belonging to a P2P system. The new peer will make a "blind" request for an entrance point to the network. For active nodes, it is enough to have a small table keeping for each network P2P, the address IP of the last device which sent a packet for this network. To increase the chances of success of this method, two addresses instead of one can be stored in active nodes.

To avoid problems of inopportune disconnections, a suggested solution is to discharge peers from this task. For example, at the time of its connection, the peer sends an active program which will be executed in the active node to manage its future disconnection. The active node "will ping" repeatedly the peer and when it loses the contact, the active node will be able to take measures necessary to inform the other peers of the network of the peer disconnection. This small active program will be erased just afterwards.

5. Conclusion and Future Work

P2P and programmable networks are technologies well suited to distributed systems. By combining these two technologies, a synergic and innovative concept in the field of distributed systems is proposed. First of all, the use of a P2P system can enhance the service deployment in programmable networks. Secondly, a programmable network infrastructure can be used to identify and optimize packet flows in P2P systems.

In [5] the layered architecture to improve the dynamic deployment of service for active networks was presented. Defining the active/control planes, the active network offers more reliable and efficient service deployment. By placing a P2P layer over the active network, active node attributes can be disseminated throughout the network without scalability problems. Thus, some active nodes (control nodes) are enabled to select through which active nodes packets should be forwarded in order to execute the desired service.

This paper proposes several applications using programmable networks as infrastructure to support P2P systems. First of all, the identification and the optimization of P2P flows are possible using an open signaling network based on PromethOS [6].Secondly, it proposes a solution of "pushing" files that allows load balance for upload peers. Finally an approach to the management of joining/leaving peers is presented.

It is interesting to note that these solutions naturally lead us to discharge the P2P application from the network management by transferring this task to the programmable network. Thus, P2P applications can concentrate on end-user services while the programmable network infrastructure takes the responsibility for the network management.

To conclude, we think that the complete convergence among P2P and programmable networks can render service providers the ability to quickly create and provide new and customized services in response to market demand. This should be the key factor in determining the performance of these service providers.

Future works can deal with the concept of active P2P multicasting in order to reduce the bandwidth charge on the edges of the network.

6. References

[1] DARPA Active Network Research Program. Active Networks CBD Reference (BAA #98-03). [Online]. Available at http://www.sds.lcs.mit.edu/darpa-activenet/ 1998.

[2] Open Signaling Working Group. Available at http://comet.columbia.edu/opensig/ 1997.

[3] J.Mischke, "Scalability of Lookup and Search Overlays in Peer-to-Peer Networks," Thesis of Swiss Federal Institute of Technology, Zurich, 2004.

[4] Eng Keong Lua, Jon Crowcroft, Marcelo Pias, Ravi Sharma and Steven Lim. A Survey and Comparison of Peer-to-Peer Overlay Network Schemes. *Journal of Communications Surveys & Tutorials, IEEE (2005)*, pp. 72-93, 2005.

[5] André Ribeiro Cardoso, Ahmed Serhrouchni, Bertrand Mathieu and Mikaël Salaün. Service Deployment in Active Networks Based on a P2P System. In *Proceedings of 2nd International Workshop on Mobility Aware Technologies and Applications*, LNCS – Springer Verlag, 2005.

[6] R Keller, L Ruf, A Guindehi, andB Plattner. PromethOS: A dynamically extensible router architecture for active networks. In *Proceedings of of 4th Annual Int. Working Conference on Active Networking (IWAN 2002),* LNCS 2546, pp 204-215. December 2002.

[7] M. Bossardt, T. Egawa, H. Otsuki, and B. Plattner, "Integrated Service Deployment for Active Networks," IWAN'02, Dec 2002.

[8] R.Haas, "Service Deployment in Programmable Networks," Thesis of Swiss Federal Institute of Technology, Zurich, 2003.

[9] R. Keller, B. Plattner, "Self-Configuring Active Services for Programmable Networks," in *Proceedings of International Working Conference on Active Networks (IWAN 2003)*, Kyoto, Japan, December, 2003.

[10] R. Kilany, A. Serhrouchni, M. Riguidel, D. Zebiane, "A Control Architecture for Active Networks," *IEEE, International Conference on Software, Telecommunications and Computer Networks, SoftCom2001*, Oct. 2001 Croatia.

[11] I. Dedinski, H. De Meer, L. Han, L. Mathy, D. P. Pezaros, J. S. Sventek and Z. Xiaoying. Cross-Layer Peer-to-Peer Traffic Identification and Optimization Based on Active Networking. In *Proceedings of Seventh Annual International Working Conference on Active and Programmable Networks (IWAN 2005)*, November 2005.

[12] Le Fessant, F., Handurukande, S., Kermarrec, A.-M., Massoulie, L., Clustering in Peer-to-Peer File Sharing Workloads, in Proceedings of The 3rd International Workshop on Peer-to-Peer Systems (IPTPS'04), San Diego, February 26-27, 2004.

[13] N. Leibowitz, A. Bergman, R. Ben-Shaul, A. Shavit, Are File Swapping Networks Cacheable? Characterizing P2P Traffic. In *Proceedings of 7th International Workshop on Web Content Caching and Distribution (WCW'03)*, Boulder, CO, 2002.

[14] Eng Keong Lua, Jon Crowcroft, Marcelo Pias, Ravi Sharma and Steven Lim. A Survey and Comparison of Peer-to-Peer Overlay Network Schemes. *Journal of Communications Surveys & Tutorials, IEEE (2005)*, pp. 72-93, 2005.

[15] eDonkey2000. [Online]. Available at http://www.edonkey2000.com.

[16] eMule Project. [Online]. Available at http://www.emule-project.net.