Combining Load Balancing with Energy Saving in a Cluster – Based P2P System

Minas Tasiou, Konstantinos Antonis
Dept. of Computer Engineering
Technological Institute of Central Greece
Lamia, Greece
mtasios@teilam.gr, k_antonis@teiste.gr

Theofanis – Aristofanis Michail
Computer Technology Institute
Rion, Patras, Greece
michail@westgate.gr

Abstract **– Load balancing is an interesting domain for research in Peer-to-Peer (P2P) systems, because of the large scale, heterogeneity and dynamic nature of the peers. In this work, we properly change the load balancing algorithm presented in [2] for an unstructured P2P system, that is based on a partially centralized architecture, to lower the power consumption of the whole system, without performance unacceptable loss in load balancing measures. In this algorithm, we consider some peers of the system as "green" nodes, we locate the peer to serve a request with the use of heuristic methods combined with a simple mathematical model, and we use simulation to compare our results with the results of [2]. As the reader will see, we can save much energy with a very low percentage loss in load balancing measurements.**

Keywords – Energy saving, load balancing, cluster- based P2P systems, simulation

I. INTRODUCTION

A lot of research has been done during the last years on the problem of energy saving in large distributed systems like clouds, computational grids, peer-to-peer (P2P) systems, etc. In this work, we study the problem of energy saving in P2P systems, where users contribute their resources (storage space, computation time) and content (files, etc.) to the community. The users act both as a client and server. Although there are various potential domains of P2P systems, the file sharing systems (Napster [5], Kazaa, Gnutella [6], BitTorrent [7], etc.), received the greatest popularity among internet users [4].

The operation of any peer-to-peer content distribution system relies on a network of peer computers (nodes), and connections (edges) between them. This network is formed on top of—and independently from—the underlying physical computer (typically IP) network, and is thus referred to as an "overlay" network. Overlay networks can be distinguished in terms of their centralization and structure [4].

Considering network centralization the following three categories are identified:

- Purely Decentralized Architectures. All nodes in the network perform exactly the same tasks, acting both as servers and clients, and there is no central coordination of their activities.
- Partially Centralized Architectures. The basis is the same as with purely decentralized systems. Some of the nodes, however, assume a more important role, acting as local central indexes for files shared by local peers. The way in which these superNodes are assigned their role by the network varies between different systems. It is important, however, to note that these superNodes do not constitute single points of failure for a peer-to-peer network, since they are dynamically assigned and, if they fail, the network will automatically take action to replace them with others.
- Hybrid Decentralized Architectures. In these systems, there is a central server facilitating the interaction between peers by maintaining directories of metadata, describing the shared files stored by the peer nodes.

By structure, overlay networks are classified as following:

- Unstructured. The placement of content (files) is completely unrelated to the overlay topology. Unstructured systems are generally more appropriate for accommodating highly-transient node populations. Some representative examples of unstructured systems are Napster, Gnutella, Kazaa.
- Structured. In structured networks, the overlay topology is tightly controlled and files (or pointers to them) are placed at precisely specified locations. These systems essentially provide a mapping between content (e.g.file identifier) and location (e.g. node address), in the form of a distributed routing table, so that queries can be efficiently routed to the node with the desired content. Typical examples of structured systems include Chord [11], Content Addressable Network (CAN) [12], Tapestry [13] among others.

P2P systems offer a lot of interesting research domains, like searching, infrastructure, security, load balancing, etc. Load balancing is an interesting domain for research in P2P systems, because of the large scale, heterogeneity and dynamic nature of the peers. The research in this domain falls in two categories. In the first, techniques for better item distribution in the name space so as improvements in routing and searching can be accomplished (e.g.[11, 12, 13]). In the second, techniques for items' replicas placement to the network nodes, for improving the throughput and Quality of Service (QoS) provided to the users, can be included (e.g.[14]).

Another research domain that we consider here, is energy saving. Recently the energy consumption of Information Technology (IT) and networking infrastructure has attracted more and more attention. Energy consumption is now considered as an important factor of IT and communication system design [8, 9]. In a 2007 report ([10]) towards the Congress, the Environmental Protection Agency (EPA) stated that: "During the past five years, increasing demand for computer resources has led to significant growth in the

 ISBN: 978-1-61208-346-9

number of data center servers, along with an estimated doubling in the energy used by these servers and the power and cooling infrastructure that supports them".

In this work, we properly change the load balancing algorithm presented in [2] for an unstructured P2P system, that is based on a partially centralized architecture, to lower the power consumption of the whole system, without performance unacceptable loss in load balancing measures. The original algorithm uses replication to achieve better performance in load balance and to prevent the system from users acting as free-riders. We maintain the replication feature in our algorithm, too. But, the original algorithm, as well as the other algorithms for unstructured systems, concerns more to optimize the fairness index in order to achieve better balancing of the existing load, without taking into account the energy effiency, which is very crucial nowadays. To solve the problem, we consider some peers of the system as "green" nodes and we use some heuristic methods combined with a simple mathematical model to locate the peer able to serve a suitable request. To assess the performance of our algorithm we use simulation and we compare our results with the results of [2]. As the reader will see, we can save much energy with a very low percentage loss in load balancing measurements. Our algorithm can solve the problem of energy saving, if the nodes' energy efficiency is known. The reader can find how a node's energy efficiency can be computed in section IV.

The article is organized as following: section 2 presents related works in load balancing of various P2P systems under the energy saving assumption. Section 3 describes briefly the load balancing algorithm appeared in [2] for an unstructured P2P system. Section 4 presents our algorithm which combines the algorithm of [2] with the energy saving feature. Section 5 describes the experiment and the derived simulation results. Finally, section 6 presents our conclusions and presents our plans for future work.

II. RELATED WORK

In [15], a novel scheme called energy-efficient peer-to-peer caching with optimal radius for 4G hybrid networks is introduced, that reduces energy consumption and distributes load equitably among mobile peer nodes. In this scheme, a P2P overlay network is built among mobile nodes to facilitate cooperative data sharing in order to relieve the traffic bottleneck at the base station.

In [21], a P2P Minimum Boundary Rectangle (PMBR) is proposed, which is a spatial index specifically designed for mobile P2P environments. A node that contains desirable data item (s) can be easily identified by reading the PMBR index. Then, a selective tuning algorithm is proposed, called Distributed exponential Sequence Scheme (DSS, for short), that provides clients with the ability of selective tuning of data items, thus preserving the scarce power resource. The proposed algorithm is simple but efficient in supporting linear transmission of spatial data and processing of location-aware queries. The results from theoretical analysis and experiments show that the proposed algorithm with the PMBR index is scalable and energy efficient in both range queries and nearest neighbor queries.

In [16], the Round Robin method for reducing power consumption is discussed and then a load balancer method with queue system for reducing the total power consumption of a server peer in a peer-to-peer system is presented.

In [17], the legacy BitTorrent protocol is compared to EE-BitTorrent, a proxy based version recently proposed for energy efficiency, in a residential scenario. It is shown that the performance achieved by users is strongly influenced by the uplink throughput allowed by the access network. When the available uplink rate is low, the legacy BitTorrent protocol performs poorly and EE-BitTorrent outperforms it, in terms of average download time and energy consumption at the user's PC. The opposite occurs when the uplink rate is good. Motivated by these results, the researchers designed and implemented AdaBT, an adaptive algorithm that dynamically selects the most efficient BitTorrent option (i.e., legacy or proxy-based), depending on the operating conditions experienced by the user. The experimental results show that AdaBT is able to reduce significantly the download time provided by either the legacy BitTorrent or EE-BitTorrent, reducing this way the power consumption.

In [18], a new architecture for sharing resources amongst home environments is proposed. This approach relies on complete decentralization in a peer-to-peer like manner, and above all, aims at energy efficiency. Energy metrics are defined, which have to be optimized by the system. The system itself uses virtualization to transparently move tasks from one home to another in order to optimally utilize the existing computing power. An overview of the proposed architecture is presented as well as an analytical evaluation of the possible energy savings in a distributed example scenario where computers share downloads.

In [19], the power saving potential of P2P file sharing in two cases is revealed; popular and unpopular files. For popular files, it is derived, with regard to BitTorrent, an expression for the optimal time seeders should support leechers. For unpopular files, an existing model is extended by taking into account leechers' power consumption dependent on the load. Leechers are assumed to build a temporary cluster within the P2P-overlay. The required number of active leechers is determined to cope with a given load and results from an analytical model to simulation are compared. It is demonstrated that it is possible to reach almost optimal energy efficiency for the download scenario by comparing the local case without cooperation with the distributed case where leechers cooperate.

In [20], a mobile P2P video streaming and benchmarking platform is presented, which enables to assess and compare the energy consumption of different approaches in a precise manner through live assessments at runtime. The demonstrated platform includes a simple, yet high-performance tree-based mobile P2P streaming overlay which can be utilized to easily implement and assess further streaming overlay approaches.

In [23], a novel P2P overlay for Energy Level discovery in a sensornet is designed, the so-called Energy Level Distributed Tree (ELDT). Sensor nodes are mapped to peers based on their energy level. As the energy levels change, the sensor nodes would have to move from one peer to another and this operation is the most crucial for the efficient scalability of the proposed system. Similarly, as the energy level of a sensor node becomes extremely low, that node may want to forward it's task to another node with the desired energy level. It is experimentally verified that it achieves the best-known query performance of the above operation via an appropriate simulator designed for this purpose.

III. LOAD BALANCING IN A CLUSTER – BASED P2P SYSTEM

In this section, we shortly describe the model assumed and the algorithm we plan to reconsider as presented in [2], where a reader must refer for more details.

We consider a system with N Peers ($P_1 \dots P_n$), organized in M Clusters ($C_1 \dots C_m$), with each peer belonging to only one cluster [2] (see Figure 1). After joining the system a peer advertises its characteristics such as connectivity bandwidth, processing capabilities, mean online time of past connections, which are used for peer classification as leaf, SuperNode or Candidate SuperNode. Candidate SuperNodes are chosen to maintain copies of the above information, in order to replace the SuperNodes they are connected to, if they leave the system. Inside a cluster, the SuperNodes form a highly interconnected network, exchanging important information such as file index entries, peer's load and decision messages. Moreover, the SuperNodes maintain connection with SuperNodes of neighboring clusters, in order to exchange information and decide on peer migration or even clusters merging. Each SuperNode maintains a maximum number of connections with leaf nodes, while each cluster holds a maximum number of SuperNodes [2].

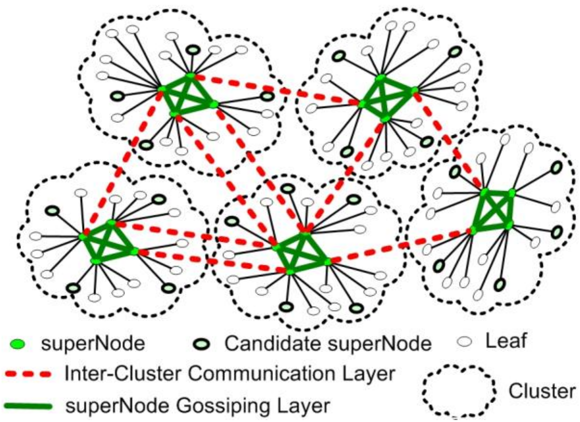

Figure 1. The architecture of the assumed P2P system.

When a peer joins the system, he publishes to the connected SuperNode information of documents he is willing to share and any other information needed to help a SuperNode to build a local metadata repository with information about all leaf nodes connected to him. A gossiping protocol like PlanetP [3] can be used to diffuse information to other SuperNode in the cluster [2] (see Figure 2). So, each SuperNode updates its repository with the received data.

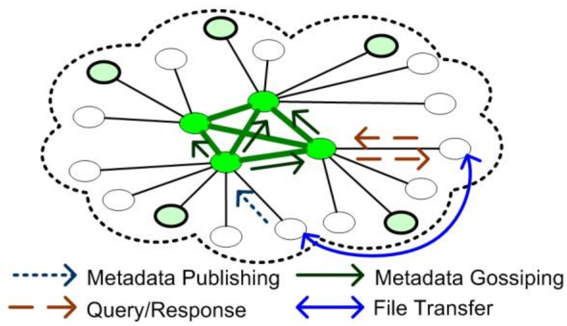

Figure 2. Communication in the P2P system.

To locate the node which will serve the response to a query (download) inside a cluster, a SuperNode makes a decision based on information about peers connection bandwidth or received load, leading to two different response strategies, as following:

MaxBw response strategy. When using this strategy, superNodes try to locate among all peers hosting the requested document, the one with the maximum connection bandwidth. This is not a load balancing response strategy but an approximation to the users' greedy behavior. We implement this strategy to use it as a basis to compare the performance of the rest strategies.

MinL response strategy. When using this strategy, superNodes try to locate among all peers hosting the requested document, the one with the minimum received load. This strategy may be unfair as far as the used bandwidth and the transfer rate is concerned but it distributes the load to more peers. The load is defined based on the consumed bandwidth for data transfers [2].

IV. COMBINING LOAD BALANCING WITH ENERGY SAVING

First of all, we need to introduce the concept of energy efficiency, which is based on [1]. The research presented in [1] is focused in energy efficiency in a data center environment. However, it considers every single computing system, which allows us to adjust it and take useful results out of it.

The final goal is to compute Peer Performance Per Energy (PPPE), which means work production per carbon energy of each individual peer. It is possible to calculate PPPE from 3 sub-metrics, Equipment Utilization (EU), Energy Efficiency (EE) and Green Energy Coefficient (GEC).

EU is a sub-metric to promote reduction in energy consumption by improving utilization rate of equipment and reduction of surplus equipment investment, computed as following:

$$EU = \frac{\text{Total measured energy (Wh)}}{\text{Total specification energy (Wh) (nameplate power rating)}} \quad (1)$$

 ISBN: 978-1-61208-346-9

EE represents an average of rated energy efficiency specific to the equipment. It is a metric showing efforts to procure energy saving equipment, given by the following equation:

$$EE = \frac{\text{Equipment rated Work capacity}}{\text{Total rated energy (Wh)}} \quad (2)$$

where

$$\begin{aligned} &\text{Equipment rated Work capacity} = \\ &\alpha * (\text{CPU Watts/CPU MTOPS}) + \\ &\beta * (\text{Watt/Gbyte of memory capacity}) + \\ &\gamma * (\text{Watt/Gbps of Network traffic}) \end{aligned} \quad (3)$$

where MTOPS in (3) means Million Theoretical Operations per second.

Factors α, β and γ are determined so that EE should become 1, if all the equipments have standard energy saving performance based on 2005 standard [1]. If all equipments have the performance that doubles the standard performance based on 2005 standard as of 2009 [22], the EE becomes 2 [1].

GEC provides ratio of renewable energy generated on-site to total energy consumed. Green energy purchased from external organization is not included in this metric. The maximum value of GEC should be limited to 0.8 [1]. GEC is computed as following:

$$GEC = \frac{\text{Total Measured Green Energy (Wh)}}{\text{Measured Energy Consumption (Wh)}} \quad (4)$$

PPPE, mentioned earlier can be calculated using (1), (2) and (4) sub-metrics, where higher value means better energy efficiency. Taking into consideration the standards of 2009 mentioned in [1], results that the range of PPPE is from 0 to 10.

$$\text{PPPE} = \text{EU} * \text{EE} * \frac{1}{1 - GEC} \quad (5)$$

We try to evaluate the energy consumption of each individual peer by using PPPE, as shown in (5) and try to form the connections between them based on that particular number. This will hopefully lead to a peer-to-peer network whose total energy consumption will be lower than a usual one.

In the original load balancing algorithm, two strategies were developed to response to search queries made by nodes [2]. A node that requests a specific document contacts the superNode which responds with the node which will serve the transfer. This decision was based on information about peers' connection bandwidth or received load.

We introduce three additional response strategies, except of those presented in [2], that involve the factor of energy efficiency.

- Energy response strategy. When using this strategy, superNodes try to locate among all peers hosting the requested document, the one with the maximum PPPE value. This response strategy performs poorly regarding load balancing metrics, as it will be shown later, because it is solely based on PPPE. In this case, peers with high energy efficiency degree collect the majority of requests. However, it offers the maximum possible energy efficiency and it is used as a basis to compare the performance of the rest strategies.
- MaxBw and Energy response strategy. When using this strategy, superNodes try to locate among all peers hosting the requested document, the one with which combines the maximum bandwidth and maximum PPPE, according to the following equation:

$$\text{MaxBw\_and\_Energy} = \alpha * \text{maxBw} + (1 - \alpha) * \text{PPPE} \quad (6)$$

Coefficient α in (6) defines the weight of each response strategy.

- MinL and Energy response strategy. When using this strategy, superNodes try to locate among all peers hosting the requested document, the one which combines the minimum load and maximum PPPE, according to the following equation:

$$\text{MinL\_and\_Energy} = \alpha * \text{minL} + (1 - \alpha) * \text{PPPE} \quad (7)$$

Coefficient α in (7) defines the weight of each response strategy. This method tries to balance the load between the peers, but also keeps an acceptable value in energy efficiency.

In order to tackle the free-riders or users that share documents with low popularity, the idea of replication has been adopted, where a document is copied to another peer in order to increase it's availability in the cluster [2]. The replication strategies used are the same with the response strategies mentioned above. That means for each simulation in our experiment, we have to decide which response and replication strategy will be used. Besides replication being an important factor of the simulation, it's search method has less effect than the response strategy on the overall results.

V. THE EXPERIMENT

A. Simulation Program Details

We adopted the same simulation algorithm that was used in [2]. It is a time-stepped-event flow-based simulation developed in MATLAB. The simulation process proceeds in rounds. In every round each node may query for documents. Then, nodes respond to these search queries triggering the execution of the replication algorithm. At the end of each round, there is a second phase during which the simulation of the flows among the network's nodes takes place.

The network parameters that are used for the experiments are based on the measurements on real systems. We simulate and evaluate the performance of a single cluster

 ISBN: 978-1-61208-346-9

consisting of 500 nodes in total, where 5 of them are considered as SuperNodes and 25 of them as energy efficient nodes whose PPPE value was randomly selected. The servicing node's energy value is set on querying node's queue, each time the last requests a document.

Factor α variable was set to 0.5. That provides equal participation of both strategies and avoids situations where the one technique conquers the other.

The rest parameters are exactly the same as the original algorithm described in [2].

Metrics

Fairness Index (FI): This is a system related metric which shows the distribution of load among peers. This index is measured based on fairness index as defined on by the formula [2]:

$$f(x) = \frac{\left[\sum_{i=1}^{n} x_i\right]^2}{n\sum_{i-1}^{n} x_i^{\ 2}} \tag{8}$$

In our system n stands for the number of peers in cluster and xi the received load on peer i. The value of the fairness index is always between 0 and 1. The closer the value is to 1 the fairer the load distribution becomes.

Quality of Availability: QoA is defined as the fraction of accepted requests for data transfer on the total requests for data transfer.

Throughput: This metric is defined as the total number of finished document transfers. The document transfers initiated by the replication process are not counted here.

Energy efficiency: This metric is defined as the average energy value, which is PPPE multiplied by 10, of the servicing nodes. The higher the value, the higher energy efficiency is observed on the system.

B. Comparative Results

Keeping in mind the original results, we run the three following scenarios. The replication method was set as minimum Load with a Pp = 0.2.

Sc-1 Energy response strategy: This simulation is based solely on Energy factor, it is trying to achieve the maximum Energy efficiency level on the cluster and thus FI is left unattained.

Sc-2 MaxBw and Energy response strategy: We simulate a greedy nodes' behavior combining also Energy factor.

Sc-3 MinL and Energy response strategy: There we merge minimum Load and Energy.

Sc-4 MinL (Original) response strategy: This comes from the original simulation based purely on minimum Load. We compare with the results of this strategy, because it was derived the best results among other heuristic strategies, as it was shown in [2].

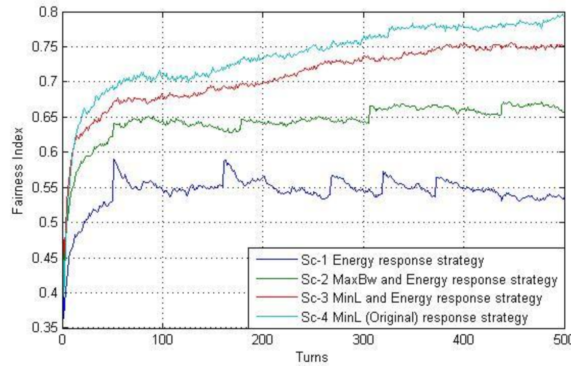

Figure 3. Fairness Index vs Simulation Turns

In Figure 3, we show the FI during the simulation. As we can see the original MinL approach performed better than the others. But as we can see, the combined MinL and Energy response strategy achieves competitive results, with a little loss in fairness.

TABLE I. SIMULATION METRICS

	#DocReq	#AccReq	QoA	En. Eff.	Thr/put
Sc-1	48977	17289	35.3%	4.4251	15025
Sc-2	38116	27684	72.6%	2.8237	22060
Sc-3	34504	23265	67.4%	3.5885	20106
Sc-4	33951	24713	72.8%	2.199	20804

As far the QoA and Throughput are concerned, **TABLE I** summarizes the results. TABLE I illustrates also the number of Documents Requested (#DocReq), the number of Accepted Requests (#AccReq) and the energy efficiency. As it was expected, pure Energy response strategy behaves poorly in comparison with the other 3. QoA of Sc-3 loses 5.4% from the original results (Sc-4) but there is a gain of 63% in energy efficiency. The reason that Sc-4 shows a high degree of energy efficiency is due to the fact that some nodes with high PPPE are chosen for their other attributes, such as low load in this case. Throughput's minor differences have not a significant impact on performance. So, it is clearly shown from the results of **Σφάλμα! Το αρχείο προέλευσης της αναφοράς δεν βρέθηκε.** that we can save much of energy with a little loss in load balancing metrics.

VI. CONCLUSIONS AND FUTURE WORK

In this work, we consider a load balancing algorithm [2] for an unstructured P2P system that is based on a partially centralized architecture. We change it properly to lower the power consumption of the whole system, without performance unacceptable loss in load balancing measures. To calculate the power consumption of a peer, we use the energy efficiency metric, appeared in [1] and introduce three new heuristics to balance the load between the peers of the system. Simulation results show that we can save much of energy with a little loss in load balancing metrics.

The plans for our future work involve the creation/alteration of load balancing algorithms, to address the power consumption problem for cloud systems. We also plan to measure the amount of saved energy in real life and not only via simulation experiments.

ACKNOWLEDGEMENT

This research is implemented through the Operational Program "Education and Lifelong Learning" and is co-financed by the European Union (European Social Fund) and Greek national funds.

REFERENCES

[1] Web Reference: Takao Shiino. “Green IT by all parties” Nomura Research Institute (NRI), March 4, 2010 http://www.oecd.org/sti/ind/45009540.pdf, [retrieved: 4,2014].

[2] J. Garofalakis and T. A. Michail, “Load Balancing in a Cluster-Based P2P System”, Fourth Balkan Conference in Informatics, IEEE, Thessaloniki, 17-19 Sept 2009, pp. 133-138.

[3] F. Cuenca-Acuna, F. Peery, R. P. Martin, and T. D. Nguyen, “PlanetP: Ysing Gossiping to Build Content Addressable Peer-to-Peer Information Sharing Communities”, Technical Report DCS-TR-487, Rutgers University, Sept. 2002.

[4] S. Androutsellis-Theotokis and D. Spinellis, “A Survey of Peer-to-Peer Content Distribution Technologies”, ACM Computing Surveys, Vol.36, No.4, December 2004, pp.335-371.

[5] S. Saroiu, P. Krishna Gummadi and S. Gribble, “Measuring and analyzing the characteristics of Napster and Gnutella hosts”, ACM Multimedia Systems Journal, Vol. 9, Issue 2, August 2003, pp. 170-184.

[6] Web Reference: Gnutella Protocol Development, http://rfc-gnutella.sourceforge.net/src/rfc-0_6-draft.html, [retrieved: 4/2014].

[7] Web Reference: BitTorrent Protocol Specification, http://www.bittorrent.org/beps/bep_0003.html, [retireved: 4/2014].

[8] V. Valancius, N. Laoutaris, L. Massoulie, C. Diot, and P. Rodriguez, “Greening the Internet with Nano Data Centers,” Proc. of ACM CoNEXT'09, 2009.

[9] A. Qureshi, R. Weber, H. Balakrishnan, J. Guttag, and B. Maggs, “Cutting the electric bill for Internet-Scale systems,” ACM SIGCOMM 2009, Barcelona (Spain), August 17-21 2009, pp. 123-134,.

[10] EPA Report to Congress on Server and Data Center Energy Efficiency, US Environmental Protection Agency, ENERGY STAR Program, 2007.

[11] I. Stoica, R. Morris, D. Karger, M. Kaahoek, and H. Balakrishnan, “Chord: A scalable peer-to-peer lookup service for internet applications”, In Proceedings of SIGCOMM 2001, San Diego (California), August 27-31, pp 149-160.

[12] S. Ratnasamy, P. Francis, M. Handley, and R. Karp, “A scalable content-addressable network”, In Proceedings of SIGCOMM 2001, San Diego (California), August 27-31, pp. 161-172.

[13] B. Zhao, J. Kubiatowicz, and A. Joseph, “Tapestry: An infrastructure for fault-tolerant wide-area location and routing”, Tech. Rep. UCB/CSD-01-1141, Computer Science Division, University of California, Berkeley, 94720. (April).

[14] G. On, J. Schmitt, and R. Steinmetz, “The effectiveness of realistic replication strategies on quality of availability for Peer-to-Peer systems”, In. Proc. Of the 3rd International Conference on Peer-To-Peer Computing, Linkoping (Sweden), 1-3 September 2003, pp. 57-64.

[15] M. Azami and B. Bhargava, “Energy-Efficient Peer-To-Peer Caching and Mobility Management in 4G Hybrid Networks”, TR08-030, Purdue University, 2008.

[16] A. Bhole and B. Nandwalkar, “Reducing Power Consumption in Peer-to-Peer System”, International Journal on Computer Science and Engineering (IJCSE), Vol.4, No. 06, June 2012, pp. 1199-1203.

[17] I. Giannetti, G. Anastasi, and M. Conti, “Energy-Efficient P2P File Sharing for Residential Bit Torrent Users”, IEEE Symposium on Computers and Communications (ISCC), Cappadocia (Turkey), 1-4 July 2012, pp. 524-529.

[18] H. Hlavacs, K. Hummel, R. Weidlich, A. Houyou, A. Berl, and H. de Meer, “Energy Efficiency in Future Home Environments: a Distributed Approach”, Home Networking Conference, Paris, 10-12 December 2007, pp. 69-84.

[19] H. Hlavacs, R. Weidlich, and T. Treutner, “Energy Efficient Peer-to-Peer File Sharing”, The Journal of Supercomputing, Vol. 62, issue 3, December 2012, pp. 1167-1188.

[20] M. Wichtlhuber, J. Ruckert, D. Stingl, M. Schulz, and D. Hausheer, “Energy- Efficent Moibile P2P Video Streaming”, IEEE 12th International Conference on Peer-to-Peer Computing, 3-5 Sept. 2012, Tarragona (Spain), pp. 63-64.

[21] K. Park and P. Valduriez, “Energy Efficient Data Access in mobile P2P Networks”, IEEE Transactions on Knowledge and Data Engineering, Vol. 23, No. 11, November 2011, pp. 1619-1634.

[22] ENRGY STAR, “Program requirements for Computers”, Version 5.0

[23] S. Sioutas, K. Oikonomou, G. Papaloukopoulos, M. Xenos, and Y. Manolopoulos, “Building an efficient P2P overlay for energy – level queries in sensor networks”, Proceedings of the ACM International Conference on Management of Emergent Digital EcoSystems (MEDES), 2009.

 ISBN: 978-1-61208-346-9