

On Topology Control of Wireless Peer-to-Peer File Sharing Networks: Energy Efficiency, Fairness and Incentive

Andrew Ka-Ho Leung and Yu-Kwong Kwok

Department of Electrical and Electronic Engineering
The University of Hong Kong, Pokfulam Road, Hong Kong

Abstract—Given the recent rapidly developing high speed wireless communication technologies and high traffic demand for P2P file sharing applications, wireless P2P file sharing is widely reckoned as a key component of the next generation communication network. However, running P2P applications in a wireless medium entails different constraints compared with those in the traditional wired Internet. One of the challenges is that portable wireless devices are energy-limited since they are battery-operated and the battery has inevitably limited life. Fairness and incentive are also important issues. Unfortunately, designing a protocol taking all these factors into account is still a relatively unexplored problem. In this paper, we propose a topology control protocol called TCP2P for wireless P2P file sharing network. TCP2P increases the fairness and provides incentive in wireless P2P file sharing applications and is energy-conserving.

Index Terms—topology control, wireless networking, P2P systems, file sharing, energy efficiency, fairness, incentive, network protocols.

I. INTRODUCTION

It is reported in a recent survey [23] that Peer-to-Peer (P2P) applications generate one-fifth of the total Internet traffic, and it is believed that it will continue to grow. Indeed, given the recent advances of high speed wireless communication technologies, including 3G, post-3G and WLAN, it is widely envisioned that file sharing over a wireless P2P network will naturally be the next step. However, running P2P applications on top of such a wireless network requires network developers to meet several research challenges. Firstly, mobile devices are battery-operated and energy-limited. If a peer is frequently asked to provide or relay files, its battery would be quickly exhausted. Short battery life has been found to be a crucial problem in 3G mobile phones network [15], [17], energy efficiency is thus uncompromising factors in the design of next generation wireless P2P network. Secondly, fairness and incentive issues also need to be addressed. “Why should I provide files to you?” is a generally asked question. In wired Internet, people ask this kind of question because uploading files to other peers means that part of their bandwidth is consumed, sacrificing the file transfer in their own applications. In an energy-limited wireless P2P network, it is not only a bandwidth-related problem but also an energy-related problem.

Corresponding Author: Yu-Kwong Kwok (Email: ykwok@hku.hk). This research was supported by a grant from the Research Grants Council of the HK-SAR under project number HKU 7157/04E.

In this paper, we attempt to *unify* the above factors and propose a decentralized and dynamic topology control protocol called TCP2P. To the best of our knowledge, this is the first design that has the capability of linking up “P2P” and “wireless”, and can integrate important performance concerns including energy efficiency, fairness and incentive. We investigate the performance of this protocol using a new metric called *Energy Based Data Availability* (EBDA). Based on this important performance measure, we demonstrate that our proposed protocol can remarkably increase the energy efficiency and fairness of the participants in a wireless P2P system.

The remainder of this paper is organized as follows. In the next section we discuss the background and related work on P2P networking. In Sections III and IV, we describe our proposed TCP2P protocol. In Sections V and VI, we present our simulation results. We provide some concluding remarks in the last section.

II. RELATED WORK

There is a considerable amount of work that has been done in P2P networking. Most of the the work is designed for fixed, wired network. Now we describe some proposed protocols that are related to energy efficiency, fairness and incentive issues.

Muqattash and Krunz propose a distributed power control protocol in [14]. Their design conserves energy by using appropriate levels of transmit power in mobile nodes. Another way to conserve energy is *sleeping*, in which some nodes are selected to maintain connectivity of the network while other nodes are turned into sleep mode to conserve energy. One example is SPAN, proposed by Chen *et al.* [3]. A theoretical discussion on a distributed algorithm optimized for energy consumption in a mobile wireless network can be found in [20].

Incentive and fairness issues are other important topics in P2P networking. BitTorrent (BT) [2], as one of the latest common decentralized P2P protocols deployed in the Internet, addresses the issue by dividing each file into pieces and asking each downloader to upload some pieces of the file that they have already downloaded. BT provides incentive for users to upload files and help the file sharing process.

Some schemes are proposed in the literature to study the behavior of nodes in ad hoc networks. “Selfish Networks” [22] and “Reputation System” [5] are interesting topics that deserve

to be mentioned. In particular, Wang *et al.* [22] propose a “payment scheme” to increase nodes’ willingness to forward other nodes’ packets and report the true cost involved in the routing process. In [5], the authors propose protocols for evaluating the trust-worthiness and behaviour of nodes in ad hoc networks.

III. OUR PROPOSED APPROACH

A. “WhoIs My Neighbor?”

The main function of routing protocols is to find out the appropriate path from the sender to the receiver. No matter which routing protocol is used, the question is “Why does someone need to help the source node to route the packets?” Our view is that by taking some factors into consideration, one can decide the worthiness of helping others to route their packets. Now we first define the system model of a P2P file sharing network.

B. System Model

Consider a P2P file sharing application running on top of a mobile wireless ad hoc network with N users. We denote each user by n_i :

$$U = \{n_i | i = 1, 2, 3, \dots, N\} \quad (1)$$

We use E_i to denote the remaining energy level (battery level) of a mobile user n_i . We assume that there are totally M different file objects in the network:

$$F = \{f_j | j = 1, 2, 3, \dots, M\} \quad (2)$$

In this P2P file sharing platform, we quantify the popularity of each file object and represent it using a weight, with respect to a particular user. Different files are of different popularity in the network, some file objects (e.g., latest pop music) are of higher ranking in the mind of some users. Thus, we represent the weight (rank) of an object f_k in user n_i ’s mind by a weight w_{ik} . Different file objects could have different weight values for different people, thus we need a N by M weight matrix W to denote weights of all file objects for all users:

$$W = \begin{bmatrix} w_{11} & \dots & w_{1M} \\ \dots & \dots & \dots \\ w_{N1} & \dots & w_{NM} \end{bmatrix} \quad (3)$$

Traditionally, “neighboring nodes” refer to those nodes which are in close proximity of the source node. However, our point is that even a node, say n_i , is within the transmission range of the source node n_s , it is not a “must” that n_i needs to be the next-hop of n_s . In TCP2P, we define *adjacency set*, $\text{Adj}(s)$, which represents next-hop nodes of source node n_s :

$$\text{Adj}(s) = \{n_i | d_{n_s \rightarrow n_i} \leq R \wedge n_i \text{ is willing to be the next-hop of } n_s\} \quad (4)$$

where $d_{n_s \rightarrow n_i}$ denotes the distance between n_i and n_s , R is the transmission range of n_s . Now the question is, “Is n_i willing to be the next-hop of n_s ?” To answer this question, n_i needs to consider several factors.

The first factor that n_i has to consider is *fairness*. If n_i has already helped a lot in routing packets for other nodes and contributed in the maintenance of the P2P network for long, it is

desirable to exclude n_i in the adjacency set of n_s , so that the relaying and routing task can be more evenly distributed among all nodes.

The second factor is the *aggressiveness* of the source node n_s . If n_s always asks for files but does not contribute anything, n_i is likely to be a selfish user and this would reduce the willingness of other nodes to relay n_s ’s packets.

The third factor needs to be considered is the *popularity* of file objects held by n_i . Apart from the aggressiveness of n_s , n_i also needs to consider the number of files that will be downloaded from n_i . If n_i holds a large number of files or a smaller number of popular files, then there can be many other peers asking for files from n_i . This would not only occupy n_s ’s bandwidth but it would also be a burden in both energy and bandwidth sense for the neighbor(s) of n_s because the neighboring node(s) has to frequently route its packets outwards.

The final factor considered is the *remaining energy levels* of nodes. Being a relaying node (receives then transmits files) requires more energy consumption than simply being a client-peer (receives files) or server-peer (transmits files). It is obviously undesirable to ask a peer with a very low remaining energy level to be the relaying node. Also, it is not enough to define a single threshold so that all peers with energy levels above the threshold act as relaying nodes and those with low energy levels only enjoy service from the high energy peers, which is simply unfair. We thus propose the use of a relative metric, i.e., if a node n_i has a higher energy level than n_s , we say that n_i is more eligible to be a next-hop for n_s and it is also more desirable for n_i to relay packets from n_s .

C. Preference Function

Now we quantify the above four factors and then combine them into a “Preference Function” to evaluate the worthiness of being an intermediate node to relay packets for the source node.

1) *Contribution Metric*: In any P2P network, a network node is both a user and a component of the network infrastructure for routing packets for other nodes. They are also supposed to share files with each other. Now one may ask “How about a selfish user who only downloads files but does not upload files?” In fact, this phenomenon is shown to be common in file sharing applications. Adar and Huberman [1] show that nearly 70% of Gnutella users do not share any files. These users are known as “free-riders”. Because of this, our idea is to evaluate the level of contribution of each user and link it with the preference function. If a node n_i contributes more, then other nodes are more willing to be the relay of n_i .

We classify network activities into two types. The first type is “public service”, where a node exchanges control packets with other nodes for maintaining network infrastructure (for route discovery, adding or leaving of network nodes), handling file searching query, helping in transferring files or being the original source of files (server). All these actions are considered as making contribution to the whole P2P community. The second type is “private service” where a node downloads files from other nodes. The contribution metric is then defined as:

$$\frac{T_{public}^i}{T^i} \quad (5)$$

where T_{public}^n denotes the time duration in which node n , is engaging in public service and T^n is time duration for which n , has joined this P2P network as one of the peers.

Given this metric, we can allocate the relaying task more evenly among all nodes, nodes who have already provided a long period of public service can be excluded from the adjacency set.

2) *Aggressiveness Metric*: As we have just mentioned, we use “public service” and “private service” to classify network activities. To assess the aggressiveness of n , we define the following metric:

$$\frac{T_{private}^s}{T_{public}^s} \quad (6)$$

where $T_{private}^s$ and T_{public}^s denote the time duration in which n , is engaging in private service and public service respectively.

These two metrics use the behavior of n , and n , known so far to predict the upcoming traffic demand of n_i and n . The service time can be easily recorded by a local timer. As time goes by, these metrics would be changed. Therefore, our protocol would update these values in a distributed manner to reconstruct the adjacency set of n_s dynamically.

3) *File Popularity Metric*: In a file sharing network, one of the most valuable resources is the file objects owned by network users. Some files are more popular (e.g., latest song albums by popular singers, latest movie trailers, etc.). More popular files are usually requested more by users. Indeed, work has been done on studying the relationship between the number of requests or number of replica of a particular file in a P2P network and the popularity of this file [6]. Assume that we can give a rank r to a file f to represent its popularity, then, according to Zipf's distribution [113],

$$prob(f) = \frac{K}{r^\kappa} \quad (7)$$

for some constant K and κ is close to unity, which means “the probability that a file searching query is associated with the r^{th} most requested file is inversely proportional to its rank value.” For example, the most popular file (rank = 1) would be the file most probably requested since the fraction in Equation (7) would be of the smallest value. This is known as the *Zipf's law*.

Zipf's distribution is believed to be a promising distribution to model the behavior in Internet [6]. In our design, we make a simple use of Zipf's law. Our view is that if a user holds a larger number of popular files, this peer is more likely to be a server-peer and more energy would be required for being the relaying node of this node.

Let r_i be the average rank of the file objects held by user (node) n_i ,

$$r_i = \frac{\sum_{j=1}^J r'(f_j)}{J} \quad (8)$$

where $r'(\cdot)$ denotes the rank of file f_j , J is the total number of files that user n_i holds.

We then define the *File Popularity Metric* with respect to user n_i as:

$$\frac{K}{r_i^\kappa} \quad (9)$$

Set $\kappa = 1$, the value of K can be found as $\frac{1}{\sum_{m=1}^M \frac{1}{m}}$. Thus, the File Popularity Metric is defined as:

$$\frac{1/\sum_{m=1}^M \frac{1}{m}}{r_i^\kappa} \quad (10)$$

This probability is calculated based on the average rank of files held by n_i and it serves as a rough measure on the chance that a file searching query is associated with n_i .

4) *Energy Metric*: Finally, we define the energy metric as:

$$\frac{E_i}{E_s} \quad (11)$$

where E_i and E_s denote the remaining energy levels of n_i and n , respectively. Our idea is that it is more desirable to have a node with higher remaining energy level than the source node's to be the relaying node (i.e., the fraction is larger than one). Also, a bigger difference between E_i and E_s makes this expression larger and represents a higher preference for n_i to act as the next-hop of n . If $E_s \geq E_i$, this expression would be less than one, meaning that it is undesirable to have n_i acting as a relaying node for n .

5) *Defining the Preference Function*: Now we combine all the above metrics into one *Preference Function*. We define the preference function as a weighted arithmetic sum of the above four metrics:

$$Pref(s, i) = \phi_1 \frac{T_{public}^i}{r_i} + \phi_2 \frac{T_{private}^s}{T_{public}^s} + \phi_3 \frac{1/\sum_{m=1}^M \frac{1}{m}}{r_i^\kappa} + \phi_4 \frac{E_i}{E_s} \quad (12)$$

where $\phi_1, \phi_2, \phi_3, \phi_4$ represent the relative importance of these four metrics. ϕ_1, ϕ_2 and ϕ_3 should be negative because: (1) a larger value of contribution metric means that n_i has already helped a lot beforehand, (2) a larger value of aggressiveness metric means the source node is likely to be a free-rider and does not deserve other nodes to relay too much traffic for it and (3) a larger value of file popularity metric means that the source node has a higher level of outgoing traffic. All these three cases reduce the intention of other nodes to be the relaying node and therefore they are multiplied by negative weights to reduce the value of preference function.

Now we can refine the definition of adjacency set as:

$$Adj(s) = \{n_i \mid d_{n_s \rightarrow n_i} \leq R \text{ \& } Pref(s, i) > \Lambda\} \quad (13)$$

Simply put, in TCP2P, a node n_i is willing to be the next-hop of n , and is responsible for relaying packets from n_s if the value of its preference function with respect to n , is higher than certain threshold value Λ . Given a fixed set of weight values $\{\phi_1, \phi_2, \phi_3, \phi_4\}$, the topology of a wireless P2P file sharing network is governed by the choice of the value of Λ because it affects “who should be who's next-hop”. Now we see that TCP2P is an interesting protocol in the sense that Λ *unifies* the issues of energy efficiency, fairness and incentives. By finding an appropriate value of Λ and using TCP2P, we can control the topology of a wireless P2P network with respect to the distribution of energy levels and file objects among the nodes.

One may wonder why the description above only focuses on what n_i does for n , when n_i agrees to be the next-hop for n ,

because if n_i and n , keep in contact then it seems that both of them can have equal responsibility for each other. But this is *not* true in TCP2P. The role of nodes in adjacency set is asymmetric. That is, given $n_i \in \text{Adj}(s)$, it is not necessarily true that $n, \in \text{Adj}(i)$. The construction process of adjacency set for all nodes is independent in TCP2P.

IV. PROTOCOL DETAILS

Firstly, TCP2P is a protocol running in-between network layer and MAC layer (see Figure 1). From this layering perspective, TCP2P is similar to SPAN [3], which is also a topology control protocol but it uses sleeping to conserve network nodes' energy.

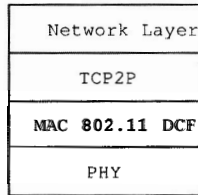


Fig. 1. TCP2P protocol structure.

Our protocol intelligently constructs the next-hop set (i.e., *adjacency set*) of each node. As described above, this requires nodes to exchange a small amount of information to calculate the value of the preference function. This small amount of information can be incorporated into RTS/CTS packets, so that extra control packet is not needed. This method is used in Muqattash and Krunz's distributed power control protocol [14]. "Connectivity Set" in their protocol is similar to the adjacency set in TCP2P, but TCP2P unifies incentive, fairness and energy-aware factors and is tailor-made for file sharing applications. Their pioneering exploration reveals that distributed algorithm with information exchange among peers is practical and does not have significant negative impact on energy efficiency due to the energy spent on transmission of small amount of control overhead (relative to the file size, e.g., in TCP2P for P2P file sharing applications). As suggested by Muqattash and Krunz, when RTS/CTS exchange is not frequent enough to update the connectivity, HELLO packets can be used to exchange information for construction of $\text{Adj}(i)$.

After getting the required information ($T_{public}^s, T_{private}^s, E_i, \mathbf{E}, \text{etc.}$), each node determines a preference function value for the nodes that it can connect to and let these nodes know if it is willing to be their next-hop.

One may argue, "how about those who are cheating and giving incorrect values of $T_{public}^s, T_{private}^s$, etc.?" Firstly, it is the protocol but not the user to report these values, if a user wants to cheat he or she has to hack the source code of the protocol stack in his or her mobile devices, which may not be easy to accomplish. Secondly, there are schemes such as *Reputation System* [5] which can be explored to measure the truthfulness or behavior of peers.

V. SIMULATIONS

A. Simulation Platform

1) Energy Consumption Model: We study the performance of TCP2P through simulations. The wireless devices in the simulations are assumed to have three possible modes of operation: Transmit, Receive and Idle. The energy consumption ratio of the three modes is set as 1 : 0.6 : 0.5, as indicated by the experimental measurements done by Feeney and Nilsson [4].

So the energy consumption on a node is set as:

$$P_{Tx}T_{Tx} + P_{Rx}T_{Rx} + P_{IDLE}T_{IDLE}$$

where the P terms represent power consumption in Transmit, Receive and Idle modes, respectively, and the T terms represent corresponding time durations that the mobile devices are in different modes.

2) Physical Layer Characteristics: The mobile devices in the simulations have similar wireless transmission parameters as those commonly found in IEEE 802.11b WLAN adapters available in the market which operate at the 2.4 GHz ISM band, where the transmit power generally ranges from 13 dBm to 30 dBm, depending on the brand of the WLAN adapter. We set the transmit power as 20 dBm in our simulations. For simplicity we assume that the transmission rate of all users is fixed at 2 Mbps. At this bit rate the receiver sensitivity is set as -91 dBm, the same as Orinoco 802.11b Gold PC card [16]. Other parameters, including processing gain, interference margin, fading margin, cable loss, etc., can be found from Chapter 8 in [9]. From these values we can determine the maximum allowed path loss to be 139.9 dB. Using the *Okumura-Hata Model* [9] we can estimate the transmission range of mobile devices.

3) Traffic Model and Mobility Model: We only consider file exchange traffic (since TCP2P is solely designed for P2P file sharing). We assume that the inter-request arrival time τ is exponentially distributed (similar to inter-call arrival time in traditional cellular phone traffic engineering [19]). In each file request event generated, we randomly select a peer and this peer requests a file object that it does not possess. The probability of requesting a file f_j is directly proportional to the weight of this file with respect to the user.

Our mobility model assumes mobile devices' velocities follow a Gaussian distribution [13] with mean equals 3 km/hr, (i.e., 0.83 m/s) and variance equals 0.54.

4) File Searching and Fetching Model: The sizes of file objects are quantized as 1 MB, 2 MB and 5 MB. File objects can be MP3 songs, ring tones for mobile phones, short movie trailers in relatively low resolution, etc. Each peer is assumed to be a PDA running WLAN. Typical storage capacity of a PDA is around 32–128 MB with CF card extension [21]. We set it as 128 MB. We assume a underlying perfect file searching engine and a maximum hop of five for searching a file (i.e., if there exists a target file owned by a user within five hops of neighbors we assume that the searching engine must be able to find this user). A maximum hop number of five is used by referring to Gnutella. In Gnutella [8], a "Time To Live (TTL)" value of seven is used (i.e., it only searches a file for a radius of seven hops). But Gnutella is for wired Internet. For mobile devices, we expect that the mobility of nodes would make the

route between the server-peer and client-peer more vulnerable to breakage and thus, a more conservative value of five is used. All routing and file searching are running on the available nodes contained in the adjacency set of TCP2P.

VI. PERFORMANCE RESULTS

A. Benchmarks for Comparison

In fact, the simplest way for a node to construct its next-hop set is to include every node within its transmission range (see Figure 2, left, source nodes are in black color, relaying nodes are in grey color, other nodes are in white color). This scenario is the *benchmark* we use to compare with a network running TCP2P, which only includes those nodes with preference function values higher than a certain threshold A (see Figure 2, right).

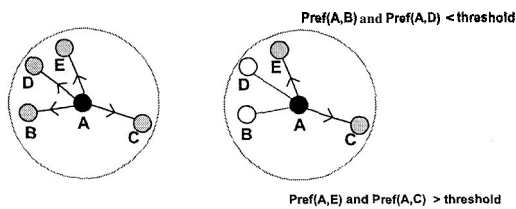


Fig. 2. Construction of neighbor set without (left) and with (right) TCP2P.

B. Performance Metrics

1) **File Object Request Failure:** We consider the total number of cumulative file object request failure and calculate the percentage of file object request failure as:

$$\frac{\text{total no. of file object request failure}}{\text{total no. of file object request}} \times 100 \quad (14)$$

2) **Average Remaining Energy:** We consider the average remaining energy of nodes at time t as:

$$E_{\text{avg}}(t) = \frac{\sum_{i=1}^N E_i(t)}{N} \quad (15)$$

which gives an approximation on nodes' energy consumption status at a particular time instant.

3) **Energy Based Data Availability:** The above two metrics are common. However, we believe that solely considering them is *not* enough for an energy-limited P2P file sharing network. We propose a more meaningful brand new performance metric called **Energy Based Data Availability (EBDA)**.

Using the notations defined in Section III-B, we consider a P2P network with N users n_1, \dots, n_N and a set of M file objects f_1, \dots, f_M , the energy-based data availability of the user n_i is defined as:

$$D_i = E_i \sum_{k=1}^M w_{ik} \delta_{ik} \quad (16)$$

The EBDA of the whole network is:

$$D = \sum_{i=1}^N (E_i \sum_{k=1}^M w_{ik} \delta_{ik}) \quad (17)$$

We interpret this performance metric in the following way. Firstly, "availability" of file objects in a P2P network depends on how many copies of each file are there in the network, this is represented by δ terms in our metric. Secondly, given the same number of copies of the same set of file objects distributed over two groups of users, each having N users, we see that a higher value of $\sum_{i=1}^N D_i$ for a group has the implication that (1) the file object resources possessed by the whole group is more "durable", more future sharing is possible before the energy of the holders is exhausted and (2) on average, the holder of a file object can keep his or her favorite files longer.

C. Results

First of all, we find that the successful rate of file searching is not affected by the use of TCP2P. Assuming a perfect file searching mechanism, both the simulations in benchmark scenario and TCP2P record a **zero value** for file searching failure. We believe that this failure rate actually depends on the value of A and ϕ , and is related to the size of network and density of peers. Now we set all ϕ_1, ϕ_2, ϕ_3 as minus one and ϕ_4 as one, which means that all factors are equally important. As mentioned before, ϕ_1, ϕ_2, ϕ_3 are negative because they are factors which reduce others' intention for being the relaying node. We find that the value of A has significant effect on the individual contribution ratio of nodes, an inappropriate value of A would not give a fair "relaying job allocation". By trial and error we find that a value of minus four for A gives better performance. We will explore it in our future work.

We find that the average remaining energy of nodes is nearly the same no matter TCP2P is used or not (see Figure 3). This means TCP2P does not conserve energy significantly in a macroscopic point of view *but* from the EBDA measurements we find that the EBDA of the whole network is higher in TCP2P than the case without TCP2P.

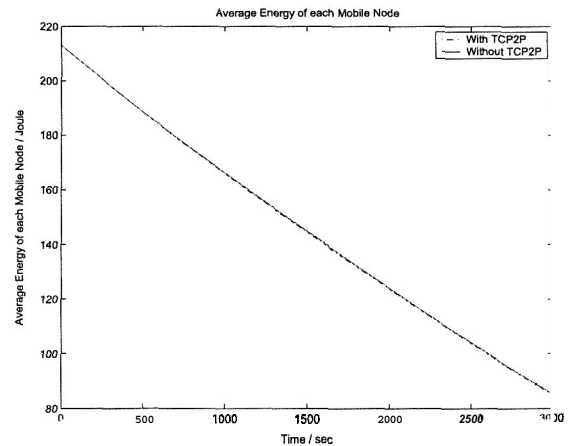


Fig. 3. Change of average energy of nodes over time.

Considering Figure 4, we see that the general trend of the change of value of EBDA is decreasing, which means that although more and more files are exchanged and the number of copies of files in the network is increased, the drop in energy levels of nodes would inevitably make EBDA decrease, no

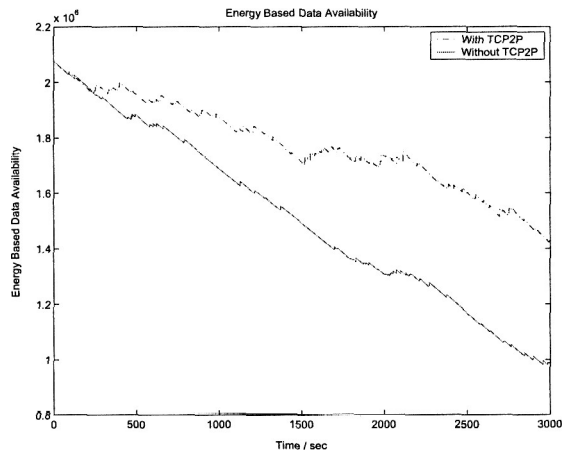


Fig. 4. Change of EBDA over time.

matter TCP2P is used or not. But TCP2P is capable of slowing down this decrease, we interpret this as TCP2P's ability to control the topology in such a way that requests and relaying actions are allocated implicitly and intelligently so that the amount of relaying job and server role is allocated proportionally while requests from all users are still satisfied to give a more "durable" P2P file sharing network.

This implicit but intelligent "job allocation" is done by controlling the next-hop of every single node. Basically, "stronger nodes" (with higher energy levels) would be the next-hop of more nodes and thus having more connections, we call these nodes "higher-degree nodes". This degree is further adjusted by the level of contribution that this node has already made to the P2P community, aggressiveness of the source node and the popularity of file objects owned by this node. In Figure 5, the individual contribution ratio (defined as $\frac{T_{public}}{T_i}$) of each node is plotted. We see that, with TCP2P, there are less "sharp peaks", which means that the public service is distributed more evenly. TCP2P avoids allocating too much amount of public service to a single node but instead tries to distribute them evenly among all nodes. This increases the fairness of the P2P network.

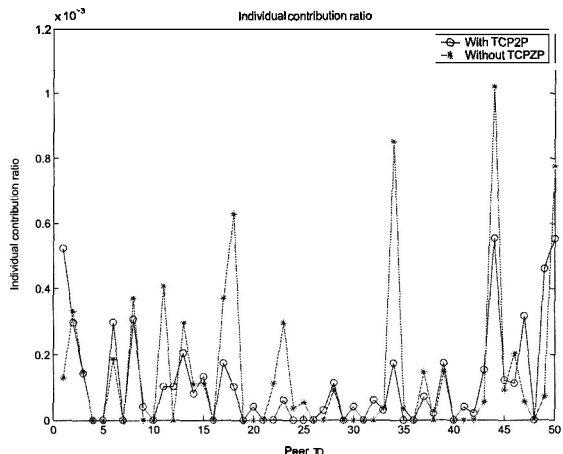


Fig. 5. Individual contribution by each node.

VII. CONCLUSIONS

We have presented a novel topology control protocol, TCP2P, and studied its performance through simulations. Based on the knowledge of distribution of energy levels and file objects among the nodes, TCP2P virtually controls the macroscopic usage of energy but does not compromise the satisfaction of peers so that the whole P2P file sharing network becomes more "strong" and "durable".

REFERENCES

- [1] E. Adar and B. A. Huberman, "Free riding on gnutella," *First Monday*, http://www.firstmonday.dk/issues/issue5_10/adar/, 2004.
- [2] BitTorrent, <http://bitconjurer.org/BitTorrent/>, 2004.
- [3] B. Chen, K. Jamieson, H. Balakrishnan and R. Moms, "Span: an energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks," *ACM Wireless Networks*, vol. 8, pp. 481-494, Sept. 2002.
- [4] L. M. Feeney and M. Nilsson, "Investigating the energy consumption of a wireless network interface in an ad hoc networking environment," *Proc. IEEE INFOCOM 2001*, vol. 3, pp. 1548-1557, Apr. 2001.
- [5] S. Ganeriwal and M. B. Srivastava, "Reputation-based framework for high integrity sensor networks," *Proc. ACM SASN 2004*, pp. 66-77, Oct. 2004.
- [6] Z. Ge, D. R. Figueiredo, J. Sharad, J. Kurose and D. Towsley, "Modeling peer-peer file sharing systems," *Proc. IEEE INFOCOM 2003*, vol. 3, pp. 2188-2198, Mar.-Apr. 2003.
- [7] C. Gkantsidis, M. Mihail and A. Saberi, "Random walks in peer-to-peer networks," *Proc. IEEE INFOCOM 2004*, vol. 1, pp. 120-130, Mar. 2004.
- [8] Gnutella, <http://www.gnutella.com/>, 2004.
- [9] H. Holma and A. Toskala, *WCDMA for UMTS*, Wiley, 2002.
- [10] S. Kishore, J. C. Chen, K. M. Sivalingam and P. Agrawal, "A comparison of MAC protocols for wireless local networks based on battery power consumption," *Proc. IEEE INFOCOM 1998*, vol. 1, pp. 150-157, Mar.-Apr. 1998.
- [11] W. Li, "Zipf's Law", <http://linkage.rockefeller.edu/wli/zipf/>, 2004.
- [12] Z. Li and M. H. Ammar, "A file-centric model for peer-to-peer file sharing systems," *Proc. IEEE ICNP 2003*, pp. 28-37, Nov. 2003.
- [13] J. G. Markoulidakis, G. L. Lyberopoulos, D. F. Tsirkas and E. D. Sykas, "Mobility modeling in third-generation mobile telecommunications systems," *IEEE Personal Communications*, vol. 4, no. 4, pp. 41-56, Aug. 1997.
- [14] A. Muqattash and M. M. Krutz, "A distributed transmission power control protocol for mobile ad hoc networks," *IEEE Trans. on Mobile Computing*, vol. 3, pp. 113-128, Apr. 2004.
- [15] NEASIA, "Short battery life is top complaint among business mobile device users," <http://www.nikkeibp.asiabiztech.com/wcs/frm/leaf?CID=onair/asabt/news/146420>, 2004.
- [16] Orinoco, "Orinoco 802.11b gold PC card data sheet," <http://www.proxirn.com/products/wifi/11b/>, 2004.
- [17] C. Panasik, "Getting the most from 3G: battery life is key," http://www.findarticles.com/p/articles/mi_m0MLY/is_9_3/ai_108331198, 2004.
- [18] M. Papadopouli and H. Schulzrinne, "Effects of power conservation, wireless coverage and cooperation on data dissemination among mobile devices," *Proc. ACM/SIGMOBILE MOBIHOC 2001*, pp. 117-127, Oct. 2001.
- [19] T. S. Rappaport, *Wireless communication, principle and practice*, Prentice Hall, 1996.
- [20] V. Rodoplu and T. H. Meng, "Minimum energy mobile wireless networks," *IEEE Journal on Selected Areas in Comm.*, vol. 17, pp. 1333-1344, Aug. 1999.
- [21] Sony CLIE handheld, <http://sonyelectronics.sonymstyle.com/micros/clie/>, 2004.
- [22] W. Wang, X. Y. Li and Y. Wang, "Truthful multicast routing in selfish wireless networks," *Proc. ACM MOBICOM 2004*, pp. 245-259, Sept. 2004.
- [23] The Washington Times Online, <http://www.washtimes.com/technology/20040303-094741-3574r.htm>, 2004.