A Lightweight Model for Mobile Peer-to-Peer File Sharing Systems

G. Stephan, B. Traboulsi, R. Dbouk, and S. Sharafeddine
Department of Computer Science and Mathematics
Lebanese American University
Beirut, Lebanon
Email: sanaa.sharafeddine@lau.edul.lb

***Abstract*—Peer-to-peer file sharing applications such as BitTorrent, Gnutella, and eDonkey are becoming widely utilized using desktops and handheld mobile devices. In this paper, we propose a lightweight model for peer-to-peer file sharing systems using mobile devices. The proposed model splits the peer-to-peer mobile client into two main modules: a mobile module that runs on the mobile device and a gateway module that runs separately on a dedicated server. In order to reduce computational requirements and energy consumption, the gateway server handles the connections to distributed peers over the Internet and downloads the required chunks of a requested file. The mobile device then simply connects to the gateway server to check the status of the download process and to get the requested file as soon as the download is complete. Therefore, the gateway relieves the mobile device from searching and downloading the file chunks. We present the general design and prototype implementation of the proposed model.**

***Index Terms*—Energy reduction, mobile computing, peer-to-peer file sharing systems**

I. INTRODUCTION

Peer-to-peer (p2p) file sharing systems have become in recent years a widely adopted alternative to traditional centralized file transfer protocols. Traffic measurements show that traffic generated by p2p file sharing applications is starting to dominate major parts of the bandwidth in certain segments of the Internet, e.g., [1], [2], [3], [4]. In p2p file sharing systems, each peer or agent operates independently of other peers with no central authority. In addition, the total resources available are a collection of the resources shared by the participating peers. As a result, the performance and efficiency of p2p systems improve as the number of peers joining the system increases [5]. Currently, BitTorrent is one of the most successful p2p file sharing systems as it consumes a significant part of the Internet p2p backbone traffic [6], [7]. BitTorrent operation is based on various components and mechanisms to perform distributed file searching and retrieval [8].

With the advent of the pervasive era where the use of mobile devices is ubiquitous, there is a need to design customized p2p file sharing systems for wireless environments [9], [10]. To achieve this goal, there are various challenges that need to be addressed. These include data rate constraints of existing wireless technologies in addition to the limited computational capabilities, memory resources, and battery capacities of mobile devices, e.g., [11], [12].

In this work, we present the design and implementation of a lightweight model for mobile p2p file sharing systems. The proposed model is based on dividing the mobile client into a mobile module and a gateway module. The gateway module runs on a dedicated server while the lightweight mobile module runs on the mobile device. The gateway server can be administered by a mobile operator or enterprise network administrator depending on the mobile devices' usage scenario. The gateway module is responsible for the connections to the p2p system in order to search, download, and store all chunks of a requested file. The mobile device then utilizes the mobile module to connect to the gateway database in order to retrieve the downloaded file. Thus, the mobile device can initiate a file sharing request, go offline, and then go online later on to check if all file chunks have been collected and stored by the gateway. Therefore, connection setup and data communications take place only between the mobile device and the gateway server.

The main contributions of this work can be summarized as follows. First, we propose a lightweight model for mobile p2p file sharing systems. The mobile devices are relieved from many resource-demanding tasks needed to participate in a p2p file sharing session. This reduces their energy consumption due to reduced computational and communications activities. Second, the proposed model can be easily adapted to various existing p2p file sharing systems. Third, we present a prototype implementation of the gateway module on a dedicated server and the mobile module on a mobile device. The implemented gateway module is designed in a flexible way to allow for further enhancements while the mobile module is designed with a user friendly graphical user interface.

The paper is organized as follows. The general design of the proposed lightweight model for mobile p2p file sharing systems is presented in Section II. The prototype implementation of the mobile and gateway modules are discussed in Section III. Analysis of the general characteristics of the proposed model is presented in Section IV. Conclusions are drawn in Section V.

Manuscript received August 20, 2009; revised February 8, 2010; accepted February 8, 2010.

doi:10.4304/jait.1.2.67-72

II. LIGHTWEIGHT MOBILE P2P FILE SHARING

A. *Motivation*

P2P file sharing systems should be designed in a way that takes advantage of the growth in the number of peers to enhance scalability and system efficiency. In [5], several issues pertaining to p2p file sharing systems are addressed including peer evolution, scalability, file sharing efficiency, and incentives to prevent free riders. File sharing efficiency is an important challenge knowing that users enjoy different download and upload link bandwidths and different device capabilities. To enhance the system efficiency, the file is normally divided into small pieces called chunks and each peer can contribute to the system depending on its upload link bandwidth.

Any user joining a p2p file sharing system is expected to upload and download chunks of data. This fact poses much less challenges on desktop computers compared to handheld mobile devices. Desktop computers normally enjoy higher speed network links, powerful processors, more storage space, and unlimited power sources. On the other hand, running p2p applications on mobile devices in wireless environments has major restrictions. Studies presented in [13] show that screen brightness in addition to data reception and transmission result in high energy consumption for battery-operated mobile devices. Moreover, data transmission is shown to consume much more power than data reception. The authors show that a peer can send up to 500MB, whereas it can receive more than 2GB before running out of power. Studies presented in [14] show that transmitting one bit over the wireless card requires 1000 times more energy than a single 32-bit computation [14]. Therefore, reducing the amount of transmitted data can help dramatically towards extending the battery lifetime of mobile devices with wireless connectivity.

In [15], energy-aware approaches for mobile devices are grouped into various categories that include networking, operating systems, application design, and computer architecture. Many energy-aware protocols and techniques have been proposed at the different internet protocol stack layers, e.g., [16], [17], [18]. In [19], a task scheduling policy is presented in order to better balance power consumption in multiprocessor systems. Moreover, [20] and [21] handle design issues in a mobile operating system to handle the energy limitation problem. In [22], the authors propose a model to predict the energy consumed in mobile device during different activity states for networking applications. In [23], the authors introduce a new technique to increase the lifetime of a mobile device battery by shutting the device during inactivity periods in order to reduce the idle power consumption.

In this work, we consider the design and implementation of a lightweight model for mobile p2p file sharing systems that reduces the challenges imposed by mobile devices especially with regards to energy consumption. The proposed model is based on delegating a major part of the p2p protocol execution to a gateway server in order to reduce the processing and communications requirements at the device. A prototype of the proposed model is implemented based on the BitTorrent file sharing system [6], [24]. In BitTorrent, searching and distribution of files can be effectively done among participating peers. The collection of peers that actively participate in the distribution of a particular file is called a torrent. When a peer joins a torrent, it starts downloading chunks of the file and accumulating more and more chunks. Every torrent depends on an infrastructure node called a tracker that keeps track of all peers joining a torrent. It only contains meta-data of the peers and its role is to act as a meeting point for all torrent peers. Peers are either classified as seeders or leechers. Seeders have a complete copy of the file, while leechers have just part of it.

B. *Proposed Model*

In this section, we present the proposed model for mobile p2p file sharing systems. Figure 1 depicts the two main modules of the proposed model: the gateway module indicated as the server environment and the mobile module indicated as the mobile environment. The mobile module communicates with the server module through a wireless network connection such as GPRS, UMTS, WLAN, or WiMAX.

As shown in the figure, the server environment is composed of several components including a search engine, p2p client to other networks, download manager, and upload manager. Each component has its own p2p module that interacts with the mobile peer through the network. The p2p component modules in the server include a status module, a search module, an upload module, and a download module. On the other hand, the mobile environment contains several components including download status, search form, file sharing by uploading, and file sharing by downloading.

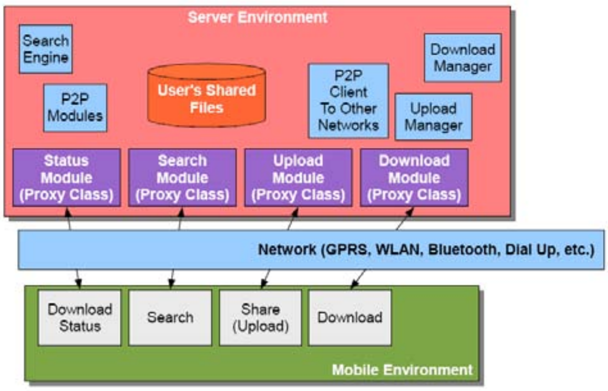

Figure 1: The Server and mobile environments.

The proposed model is based on the BitTorrent file sharing system. Therefore, the system model is composed of three main components which are the BitTorrent network, the gateway module, and the mobile module. The steps that the system takes to perform a file download operation are presented in Figure 2. First, the requesting mobile peer gets the .torrent file of the requested file from the Internet. After the .torrent file is downloaded, the mobile peer forwards it to the gateway server. The gateway server then expands the .torrent file and extracts from it the name of the tracker in addition to any additional needed information. It first checks if it already has the requested file in its cache or database of previously downloaded files. If it finds the file in the database, the gateway server transfers the requested file directly to the requesting mobile device. If not, the gateway server connects to the tracker in order to initiate the download process of the file chunks. It is important to note that the tracker keeps a list of all seeders and leechers, but it does not store the files. The tracker hooks the server with the necessary leechers or seeders to acquire the requested file. The server can then contact them directly. Once the file is fully acquired, it is ready to be transferred to the requesting mobile device (see Section IV for additional implementation details).

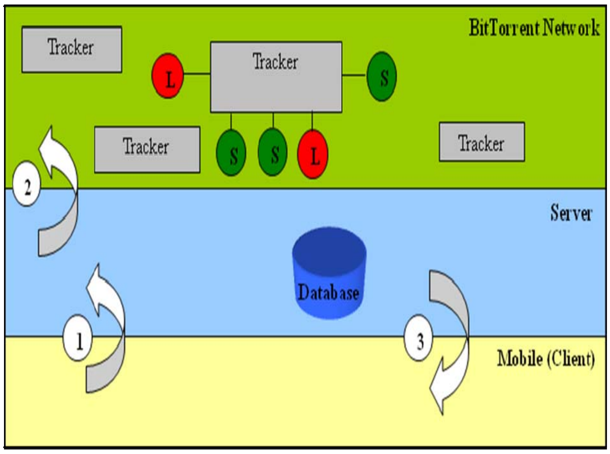

Figure 2: Steps to download a file by a mobile peer.

In regards to the upload functionality, the mobile peer will be prompted to enter some information that is needed to create a valid .torrent file if not available. The required information can include the following:

- The name of the file.
- The name to save the file as (optional).
- A description of the file.

The creation of the .torrent file requires additional information filled automatically by any mobile torrent server such as:

- URL of the tracker.
- A flag to determine if the torrent is decentralized. This only works with the Azureus BitTorrent client and, thus, it is set to false for compatibility with other clients.
- A flag to determine if the shared item is a file or a directory. This flag is set to a file by default.
- The file chunk size. This field is often set to 32KB. However, for larger files, the value can be as large as few megabytes. The problem with large chunks is that some BitTorrent clients might not be able to resume the download of a chunk. Therefore, if the transfer is interrupted, the whole chunk will have to be downloaded again.

When the above information is ready, a valid .torrent file can be built so it can be used by any BitTorrent client. After the .torrent file is ready, the mobile peer has to transfer the file to be shared to the server. For the transfer, the mobile client splits the file into smaller pieces, and these pieces are progressively sent one at a time. The gateway server now has both the file as well as its .torrent. We note that the server keeps copies of all downloaded files by a mobile peer in case it needs later on to share them with other peers. This reduces the overhead needed to transfer them again to the server.

III. Prototype Implementation

In this section, we present a prototype implementation of the proposed model in order to verify its operation and study some of its features. The proposed model consists basically of three main components: 1) the mobile client, 2) the gateway server, and 3) the BitTorrent network.

The mobile client is developed using J2ME. It runs on the mobile device and communicates with the gateway server which acts as a normal BitTorrent client. The gateway server is also implemented using Java by modifying an open source Java BitTorrent client. The modifications basically included adding 'hooks' to the API. These 'hooks' save the data flowing during the execution of the software to a Derby (also known as JavaDB) database. The Derby database was used for demonstration purposes. It should be straightforward to replace it with any other scalable database engine.

In Figure 3, we show the screen flow of the mobile client and how the user interacts with the p2p application. The user has first to register in order to be able to log in to the system using a username and a password. This authentication phase is important to have a more secure implementation. When logged in, a window is displayed with several options including download status, file search, file sharing, and file downloading from the gateway server.

The user can check the download status by clicking on the “My Downloads Status” link. This will direct the user to another window where the download status of each requested files is shown (see Figure 3 for examples). The second option is to “Search” for a specific file over the Internet. Once clicked, it will direct the user to a window with a search textbox to enter a keyword of the file to be found. We note that the mobile user has the option to go offline and disable the wireless interface rather than wait for the download to complete on the server. This is very important to reduce the energy consumption since the wireless interface consumes notable energy from the battery even in idle state. Another available option is “Share a File”, where the user can see all the previously posted files to be shared with other peers. The user can also upload new files to the server as explained in Section II. Finally, there is the “Copy to Mobile” option which can be used to direct the user to the “Completed Downloads” window. This window lists all the completely downloaded files at the server. The user can then select a file and request transferring it to the mobile device.

The mobile device is required to maintain only a single connection with the gateway server and is not responsible for any connections with leechers or seeders. The communication between the mobile device and the gateway server consists of control messages except when uploading or downloading files. The control messages are normally small in size and, thus, do not consume much energy from the battery. Moreover, the mobile device requests a file from the server only after the server has completely downloaded it.

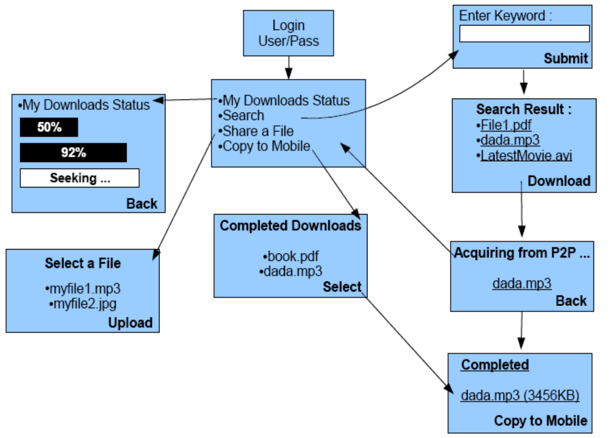

Figure 3: Mobile environment: screen flow example.

Figure 4 displays the server logs after receiving the .torrent file from the client. Here the server starts the download process by first contacting the tracker using the source given in the .torrent file URL source = “http://tracker.torrentbox.com:2710/announce?” and then getting the file. These example server logs demonstrate part of the operations that are taking place at the server side on behalf of the mobile device. Executing these operations at the device would require significant increase in energy consumption due to processing and communications activities. Figure 5 shows a screenshot of the mobile client application showing the status of the file downloads at the server side. This is mainly needed to know when the file is ready to be downloaded to the mobile client.

In order to optimize the usage of the network connection between the mobile device and the gateway server, the file is divided into blocks or pieces before transferring it from the gateway server to the mobile client. Once all the blocks are successfully received, they will be rearranged and grouped, thus regenerating the original file at the mobile device. This method of splitting a file and transferring its pieces is called file stripping and is demonstrated in Figure 6. Advantages of file stripping include more efficient data transfer via parallel streams and more efficient error recovery via retransmissions.

```
P2P Server Listenning on port 7788
Got A Connection
Got a Request !
Command is 'LGIN'
Before reading the body
Done reading the body
user : 1
password : 1
Got A Connection
Got a Request !
Command is 'ULTF'
0000001
0003462
[TBox] Shakira - Hips Dont Lie
[World Cup 2006 Live][SkidVid].torrent
0
58
Connected to DB.
DB Torrent ID : 2
Contact Tracker. URL source =
http://tracker.torrentbox.com:2710/announce?
info_hash=
T%87%9D%FE%12K%C3%3E$U%BC%C3%85%F9%20f%
FD%88%9F%03&peer_id=-BE0001-%F3%DF%AF%F7%08%
88%18%2F%8A%25%D5%F0&port=6881&downloaded=0&
uploaded=0&left=38203684&numwant=100&compact=1&event=
started
{interval=3600, complete=7, peers=[B@949f69,
min interval=3600, incomplete=14}
Peer List updated from tracker with 21 peers
41.232.121.3:11137 rate: 1.6ko/s
41.232.121.3:11137 rate: 1.6ko/s
Contact Tracker. URL source = http://tracker.torrentbox.com:2710/
announce?
info_hash=T%87%9D%FE%12K%C3%3E$U%BC%C3%85%
F9%20f%FD%88%9F%03&peer_id=-BE0001-
%F3%DF%AF%F7%08%88%18%2F%8A
%25%D5%F0&port=6881&downloaded=0&uploaded=0&left=
38203684&numwant=100&compact=1
{interval=3600, complete=9, peers=[B@1359c1b,
min interval=3600, incomplete=14}
Peer List updated from tracker with 23 peers
```

Figure 4: Server example logs.

Figure 5: Download status window.

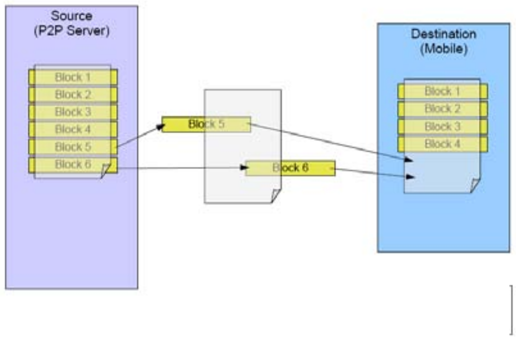

Figure 6: File stripping between the mobile client and the gateway server.

IV. Analysis

The main feature of the proposed model is its lightweight implementation at the mobile client side. For download functionality, the mobile client has to first to get the requested .torrent file and then upload it to the gateway server. The mobile client can then turn off the application, switch off the wireless interface, or even switch off the device itself. When the user wishes to check the status of the requested files to initiate some download requests, it connects to the server. The mobile client can use the server to fetch several files at the same time by sending multiple consecutive requests. For upload functionality, after uploading the files to be shared with their .torrent to the server, the load of responding to connections and requests from other peers takes place at the server side without invoking the mobile client and, thus, without energy consumption at the mobile device side.

There are four main features of the proposed model: availability, network efficiency, power efficiency, and user satisfaction. The proposed model offers availability for sharing files with other peers because files are stored on the server side. Therefore, no matter whether the status of a mobile peer is offline or online, the files that are shared will always be available for other peers. The proposed model increases network efficiency over the wireless interface used by the mobile device. This is achieved by reducing flooding especially when compared to normal p2p protocol operation since the gateway server will be taking care of various mechanisms including network discovery. Moreover, this reduces communication over the wireless network which normally has scarce bandwidth resources and is prone to errors.

The third feature is power consumption reduction knowing that mobile devices consume much more energy during data transmission and reception compared to idle mode operation. Therefore, delegating major processing and communications activities from the device to the gateway server results in significant energy reduction in the mobile device. For example, the file pieces are sent at once to the mobile device after the file is completely downloaded by the server. Therefore, the mobile device does not need to keep its wireless network interface active and listening for a long period of time to reliably receive incoming file pieces from distributed leechers and seekers. Overall, these features will lead to better user satisfaction in terms of an extended battery lifetime and faster file download speeds.

A major requirement of the proposed model is to have a highly capable gateway server in terms of processing power, memory, and bandwidth for external network connectivity. This is important for overall efficiency and system scalability. The gateway server can be administered by a mobile network operator or an intranet administrator depending on the system usage scenario. Finally, the prototype implementation of the server module is based on the BitTorrent protocol. However, the proposed model can be extended into other p2p protocols or architectures by modifying the mobile and server modules.

V. Conclusions

In this work, we presented an efficient and practical implementation of a lightweight mobile peer-to-peer file sharing system for mobile devices. The proposed model reduces the computational requirements and energy consumption at the mobile device. This is achieved by delegating major part of the p2p protocol execution from the mobile client to a gateway server in a coordinated way.

References

[1] J. Erman, A. Manhanti, M. Arlitt, I. Cohen, and C. Williamson, “Identifying and discrimination between web and peer-to-peer traffic in the network core,” in *World Wide Web (WWW) Conference 2007*, May 2007.

[2] W. John, S. Tafvelin, and T. Olovsson, “Trends and differences in connections behavior within classes of Internet backbone traffic," in *Passive and Active Measurement Conference (PAM)*, March 2008.

[3] S. Sen, O. Spatscheck, and D. Wang, “Accurate scalable in-network identification of p2p traffic," in *World Wide Web (WWW) Conference 2004*, May 2004.

[4] G. Bartlett, J. Heidemann, C. Papadopoulos, and J. Pepin, “Estimating p2p traffic volume at USC,” Tech. Report (IST-TR-645), USC/Information Sciences Institute, June 2007.

[5] D. Qiu and R. Srrikant, “Modeling and performance analysis of BitTorrent-like peer-to-peer networks,” in *ACM SIGCOMM’04*, August 2004.

[6] BitTorrent homepage, http://www.bittorrent.org, 2010.

[7] T. Karagiannis, A. Broido, N. Brownless, K. Claffy, and M. Faloutsos, “Is p2p dying or just hiding,” in *IEEE GLOBECOM’04*, November 2004.

[8] J.A. Pouwelse, P. Garbacki, D.H.J. Epema, and H.J. Sips, “The BitTorrent p2p file-sharing system: Measurements and analysis,” in *4th Int'l Workshop on Peer-to-Peer Systems (IPTPS)*, February 2005.

[9] Z. Despotovic and W. Kellerer, *Extension to Ubiquitous: Mobile Peer-to-Peer*, Book Chapter, Towards 4G Technologies: Services with Initiative, John Wiley & Sons Ltd., February 2008.

[10] S. Gurun, P. Nagapurkar, and B.Y. Zhao, “Energy consumption and conservation in mobile peer-to-peer systems,” in *MobiShare’06*, September 2006.

[11] R. Kravets and P. Krishnan, “Application-driven power management for mobile communication,” *Wireless Network*, vol. 6, issue 4. pp. 263-277, July 2000.

[12] L. Zhong and N.K. Jha, “Energy efficiency of handheld computer interfaces: Limits, characterization and practice,” in *3rd International Conference on Mobile Systems, Applications, and Services (MobiSys’05)*, June 2005.

[13] H. Lufei and W. Shi, “e-QoS: Energy-aware QoS for application sessions across multiple protocol domains in mobile computing,” in *QShine’06*, August 2006.

[14] Y. Zhang, W. Liu, W. Lou, and Y. Fang, “Location-based compromise tolerant security mechanisms in wireless sensor networks,” *IEEE J. Select Areas Commun., Special Issue Security Wireless Ad Hoc Networks*, vol. 24, no. 2, pp. 247-260, February 2006.

[15] K. Li, T. Nanya, and W. Qu, “Energy efficient methods and techniques for mobile computing,” in *Third International Conference on Semantics, Knowledge and Grid,* October 2007.

[16] K.M. Sivalingam, J.C. Chen, P. Agrawal, and M. Srivastava, “Design and analysis of low-power access protocols for wireless and mobile ATM networks,” *Wireless Networks*, vol. 6, issue 1, pp. 73-87, February 2000.

[17] S. Singh and C.S. Raghavendra, “PAMAS - Power aware multiaccess protocol with signalling for ad hoc networks,” *ACM SIGCOMM Computer Communication Review*, vol. 28, issue 3, pp. 5-26, July 1998.

[18] K. Brown K and S. Singh, “M-TCP: TCP for mobile cellular networks,” *ACM SIGCOMM Computer Communication Review*, vol. 27, issue 5, pp. 19-43, October 1997.

[19] A. Merkel and F. Bellosa, “Balancing power consumption in multiprocessor systems,” in *EuroSys Conference*, October 2006.

[20] J. Flinn and M. Satyanarayanan, “Energy-aware adaptation for mobile applications,” in *17th ACM Symposium on Operating Systems Principles*, December 1999.

[21] H. Zeng, C.S. Ellis, A.R. Lebeck, A. Vahdat, “ECOSystem: Managing energy as a first class operating system resource,” in *ACM SIGOPS Operating Systems Review*, October 2002.

[22] K. Mahmud, M. Inoue, H. Murakami, M. Hasegawa, and H. Morikawa, "Energy consumption measurement of wireless interfaces in multi-service user terminals for heterogeneous wireless networks," *IEICE Trans. on Communications*, vol. E88-B, no.3, pp.1097-1110, March 2005.

[23] E. Shih, P. Bahl, and M.J. Sinclair, “Wake on wireless: An event driven energy saving strategy for battery operated devices,” in *MOBICOM’02*, September 2002.

[24] M. Izal, G. Urvoy-Keller, E.W. Biersack, P.A. Felber, A. Al Hamra, and L. Garcés-Erice, “Dissecting BitTorrent: Five months in a torrent's lifetime,” in *5th Passive and Active Measurement Workshop,* April 2004.