IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661, p- ISSN: 2278-8727Volume 10, Issue 2 (Mar. - Apr. 2013), PP 07-12
www.iosrjournals.org

Service Based Content Sharing in the Environment of Mobile Ad-hoc Networks

Hashmi Vallipalli, A.V.Praveen Krishna
M. Tech student, Dept of C.S.E, K L University, Vaddeswaram, Vijayawada, India
Assistant Professor, Dept of C.S.E, K L University, Vaddeswaram, Vijayawada, India,

***Abstract*:** *The peer-to-peer network is the one of the traditional client server networking model. The file sharing on mobile devices is not easily achieved to the user for limited bandwidth and high cost. The irregular disconnection and IP address changes occur due to network migration. We holds the short range networking technologies as Bluetooth with no cost to the user and it is sufficiently fast to make file transfer very practical. A peer-to-peer model that permits efficient file sharing between mobile smartphones over a low-cost transport. Our paper results that peer-to-peer file transfer between todays mobile devices are practical. But the server limits must be applied on the transfers. These are unique to the mobile device environment. The upload-over-download ratio should be relatively low, due to higher current drain on transmit. Where the target file system is very slow then the larger file segments "Direct Memory Access" (DMA) mode can be utilized as different to "Program Input Output" (PIO) mode. The use of UDP for content sharing is more ideal than the use of OBEX. We will overcome some of the barriers to acceptance through our design approach. Peer discovery and content distribution occurs automatically without connection from the user. At last the transport is also implemented in the application layer and uses existing standard protocols without modification.*

Keywords: *Peer-to-peer network, Direct Memory Access, Program Input Output, mobile device environment, mobile smartphones, Bluetooth, network migration, mobile ad-hoc networks*

I. Introduction

1.1. Peer-to-peer network:

Peer-to-peer network is the one of the traditional client-server networking model. Where the existing P2P file applications in wired networks are not suitable for mobile hosts in wireless mobile networks because of the movement of peers in wireless mobile networks. There are two concerns that affect resource discovery and retrieval for P2P file sharing applications in wireless mobile networks.

- Peers movements in wireless mobile networks
- Peers join and leave in a P2P file sharing network.

The early peer-to-peer networks like ARPANET are storage model of the Internet is changing from a "content located in the center" model to a "content located on the edge" model. The evolution of P2P architecture comprises development from the first generation then to the second generation then to the third generation. The new wave of P2P architectures took place in the late 90s.The first new-wave P2P systems are more like a mixture of client-server and peer-to-peer models. This denomination comes from the fact that in these systems many of the functions related to peer and resource discovery were made in centralized servers or server pools.

The next step in the evolution of P2P was the development of pure P2P systems, where every peer had equal functionality without any centralized servers. These types of P2P systems are called the second generation of P2P. We use Gnutella Network as an example of this generation. Second-generation P2P systems strived to solve many problems of first-generation server-centric systems. Many improvements were successful, but the new model suffered from major overhead generated by the binding messages and queries propagating around the Internet. The current generation, the third generation of P2P, is a mixture of the first and the second generation. In third-generation systems, some of the peers are so-called super-peers. These super-peers are organized dynamically. Unlike in the earlier generations, only super-peers are used in peer and resource discovery, which significantly decreases the stress caused to the network. Also several binding and routing optimization methods are used to decrease the overhead. The use JXTA is an example of 3rd generation P2P systems.

The growing usage of mobile devices "smart phones and PDAs" and thus a need for mobile P2P applications bring up new challenges to peer-to-peer networking. Compared to desktop computers, mobile devices have many restrictions: network bandwidth, memory, processing capacity and battery life are limited. Also the network barriers, NATs (Network Address Translator) and firewalls bring challenges to mobile P2P networking. The evolution of mobile network technologies and devices decreases hardware limitations and thus makes it possible to use P2P applications also in mobile devices. Mobile environment also opens new creative ways to use P2P technology. It introduces ways to develop P2P towards a new mobile generation.

1.2. Gnutella Peer To Peer File Sharing Networks:

The Gnutella is a decentralized peer-to-peer file sharing protocol established in the year 2000 using an installed Gnutella client users can search, download and upload files across the internet. The popular Gnutella clients are Bear Search, LimeWire shareaza. Where the earlier version of the Gnutella protocol did not scale well enough to match the network’s popularity. Technical improvements solved these scalability issues at least partially. Gnutella remains fairly popular but less so than some other peer to peer networks. That are Bit Torrent, eDonkey 2000.The Gnutella is technically a distinct peer to peer n/w then the newer “Gnutella-2”.

1.3. Bluetooth:

Bluetooth is a short range radio communication system designed for communication of devices like mobile phones, PDAs, notebooks, PCs, printers, headsets etc. It should be cheap and low power consumption for easy implantation into mobile devices. First it should mainly replace cables, now this is forming complex system for communication which is able to create Pico networks (’radio LAN’) not only based on packet data transfer (ACL) but also for voice services (SCO). Bluetooth 2.0 is present using model for simple pairing procedure. Where the Bluetooth is to be accepted as the standard feature of the current smartphones. The Bluetooth 2.0 is operates on the 2.4GHz radio frequency. The Bluetooth is using EDR (Enhanced Data Rate) for having 3Mb/s data rate. Where it is using 2.1 Mb/s in the usage.it ranges up to 10meters.

1.4. Smartphone’s:

A Smartphone is a mobile phone built on a mobile computing platform. It has more advanced computing ability and connectivity than a feature phone. The first smartphones were devices that mainly combined the functions of a personal digital assistant (PDA) and a mobile phone or camera phone. Today's models also serve to combine the functions of portable media players, low-end compact digital cameras, pocket video cameras, and GPS navigation units. Modern smartphones typically also include high-resolution touchscreens, web browsers that can access and properly display standard web pages rather than just mobile-optimized sites, and high-speed data access via Wi-Fi and mobile broadband.

The most common mobile operating systems (OS) used by modern smartphones include Apple's iOS, Google's Android, Microsoft's Windows Phone, Nokia's Symbian, RIM's BlackBerry OS, and embedded Linux distributions such as Maemo and MeeGo. Such operating systems can be installed on many different phone models, and typically each device can receive multiple OS software updates over its lifetime.

The distinction between Smartphone’s and feature phones can be vague and there is no official definition for what constitutes the difference between them. One of the most significant differences is that the advanced application programming interfaces (APIs) on smartphones for running third-party applications can allow those applications to have better integration with the phone's OS and hardware than is typical with feature phones. In comparison, feature phones more commonly run on proprietary firmware, with third-party software support through platforms such as Java ME or BREW.

II. Existing System

The Peer-to-Peer model is the most popular model in the wired communication and ad-hoc networks. In this model the Peers use to communicate with each other via ad-hoc connections in systems, desktops. Where these uses the Bluetooth technology for simple point-to-point pairing approach file transfer between two mobile phones (nodes). There is no content subscription, no automatic peer discovery, and no simultaneous or time-shared file transfer involving multiple phones in vicinity. The user micro-manages all aspects.

The Peer-to-Peer model is facing some limitations in the mobile context:

- On processing power
- On-board device memory
- Wireless data bandwidth
- Upload-to-Download ratio to conserve battery energy
- Partially centralized network

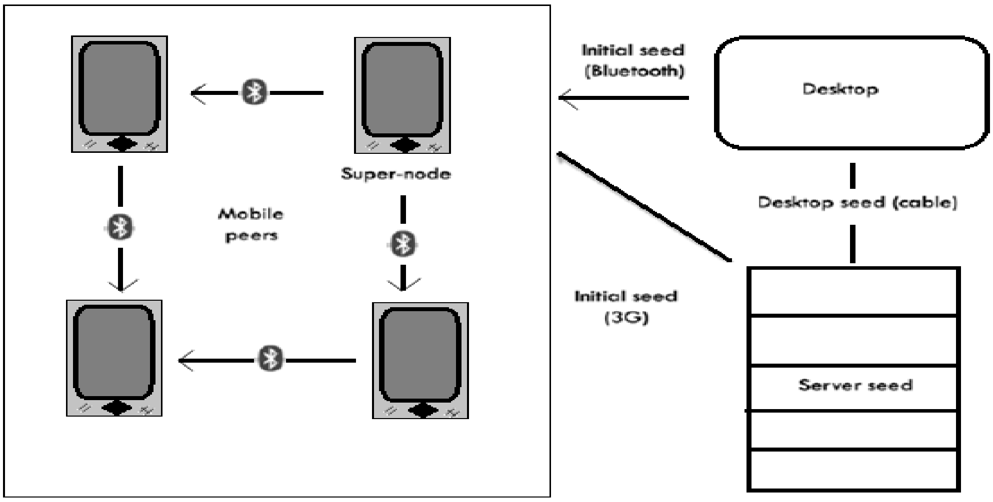

Fig 1. System model for mobile service-based peer-to-peer sharing

III. Proposed System

The Gnutella File Sharing Network Protocol is used for file sharing in mobile environment. This network is different from a simple point-to-point pairing approach commonly found in a Bluetooth file transfer between two mobile phones. The service based content sharing system utilizing the Bluetooth transport. Where the Peer-to-Peer sharing allows for efficient content distribution using low-cost links that do not impose a load on the mobile carrier infrastructure. It is mainly occurs in the transport layer. The service based content sharing system mainly uses Bluetooth because, Bluetooth is low-cost, high performance and very economical by following terms

- Service discovery
- Connectivity
- Throughput
- Power conception
- Connection pairing
- File segmentation and sharing strategy

A node that has more connectivity options such as 3G may download content directly from a server to obtain an initial seed or from a desktop computer with peer-to-peer Connectivity itself. This so-called super-node will then share its file content with multiple other peers using a peer-to-peer transport. Note that a single peer may draw segments from multiple sources and assemble them locally. A peer may create multiple simultaneous connections or transfer its content using one connection at a time.

The user initially subscribes to a service of interest, such as an OS update or video sharing service. Peers discover each other through the process of scanning. Compatible services are discovered, and content that is assigned to a service is indexed and may be requested to be placed on a transfer queue. Individual segments are transferred and concatenated at the receiver end. Once all segments arrive, they are assembled, and the content is registered with applicable application handlers so that it becomes visible to the user.

IV. Methodology

Peers regularly broadcast their available services. In which peers form “alliances” with nearby nodes. The cache information based on the local advertisements of services that they broadcast. An alliance will essentially consist of nodes within communication range. Request for services may be made in a pull manner if a cache miss occurs. The new content may be announced through a push-based technique to other nodes.

Algorithms used in the system are:
1. Receiver-driven Discovery Control (RDC)
2. File Provider Selection (FPS) algorithm

A.*Re*ceiver-driven Discovery Control (RDC)

While Discovery Time is reached do
//Step1: send discovery messages to obtain status of peers that share files;
Peer Discovery ();

//Step2: re-calculate the next discovery period;
If (CurrentTransRate > RequiredTransRate) then
Discovery Time = Discovery Time ×β;
Else If (CurrentTransRate < RequiredTransRate) then
Discovery Time = Discovery Time ÷β;
End If
//Step3: schedule the next discovery period;
Schedule (Discovery Time);
//Step4: avoid too small Discovery Time;
If Discovery Time < $Threshold_{boundary}$ then
Stop file transferring and wait a random time to discovery again;
End If
End While

Fig 2: The Receiver-based Discovery Control (RDC) algorithm.

Symbols definition:
Discovery Time: A time unit that defines the period of a peer sending discovery messages.
Required Trans Rate: A threshold determining a connection quality.
Current Trans Rate: the current transmission rate of an active connection that receives data packets.
β: A constant that defines the extend and shrink factor of discovery time
(β > 1)
$\text{Threshold}_{\text{boundary}}$: A constant is defined to avoid too small Discovery Time
End definition

B. File Provider Selection (FPS) algorithm

While an active connection $ActiveConn_i$ is disconnected do
//Step1: FPS checks the history table that records discovered peers in response cache;
Provider List=FindDiscoveredPeers ($ActiveConn_i$);
//Step2: FPS estimates bandwidth of each resource providing peer and finds the best one for resuming the interrupted connection;
if Number(*Provider List*) 6≠ 0 then
Bandwidth List=EstimateBandwith (*Provider List*);
Best Provider=FindBestProvider (*Bandwidth List*);
Connect Provider (*Best Provider*);
end if
end while

Fig 3: The File Provider Selection (FPS) algorithm

Symbols definition:

ActiveConni: An active connection i;
Best Provider: A resource providing peer that has the best connection quality;
Provider List: A list that is used to store candidate resource providing peers;
Band Width List: A list that is used to store estimated bandwidth information of a resource providing peer list;
End definition

V. Implementation

The core features of a file-sharing protocol on two real smartphone platforms so that we could assess its practical feasibility with confidence.

A. Blackberry

In this case study, they have developed functioning server and client applications from scratch for a RIM BlackBerry device that permits peer-to-peer file transfer over Bluetooth 2.0. They were used the BlackBerry JDE (Java Development Environment) and SDK (Software Development Kit) version 4.7 from RIM. It application supported J2ME (Java 2 Mobile Edition) and the Connected Limited Device Configuration (CLDC), which is a strict subset of the class libraries present in J2SE (Java 2 Standard Edition). They utilized the JSR-82 API from Sun Microsystems to allow our Java applet to utilize Bluetooth connectivity. In which they installed the applets on two BlackBerry 8300-series ("Curve") test devices, with one acting as a client and the other the server. The Curve is a smartphone with an Intel PXA90164 312 MHz processor, 64 MB of flash memory, 16 MB of SDRAM, and connectivity on the GSM, GPRS, and EDGE networks.

These devices ran version 4.5 of the BlackBerry applications software, with support for Bluetooth 2.0, including OBEX (Object Exchange). OBEX is a communications protocol that facilitates the exchange of binary data objects between devices. It is a reliable transport layer that is implemented above the RFCOMM layer, which is a virtual serial port protocol that allows a Bluetooth device to simulate the functions of a serial port. In turn, RFCOMM is implemented on top of L2CAP (Logical Link Controller Adaptation Protocol) that acts as a data multiplexer for all other Layers.

They configured the test devices to support data transfer via the serial port profile and identified the application service using a 128-bit UUID (Universally Unique Identifier).

They used OBEX for the transfer, which allowed the application to specify metadata in addition to the payload, such as the file name, file type, and file size, which is indispensable for the purpose of file sharing. They have measured the transaction times for sending file segments of various sizes between two BlackBerry Curve devices, from 0.5 MB to 3 MB in size, and writing them from RAM to flash memory. When normalized the times by dividing by the amount transferred, they were found that the segment size did not have a significant effect on performance. Here they observed that after continuously running the tests, the Java Virtual Machine would undertake garbage collection at periodic intervals to free data that had been written to flash and was no longer referenced in RAM, which would pause the user interface. This unavoidable activity was not included in the measurements.

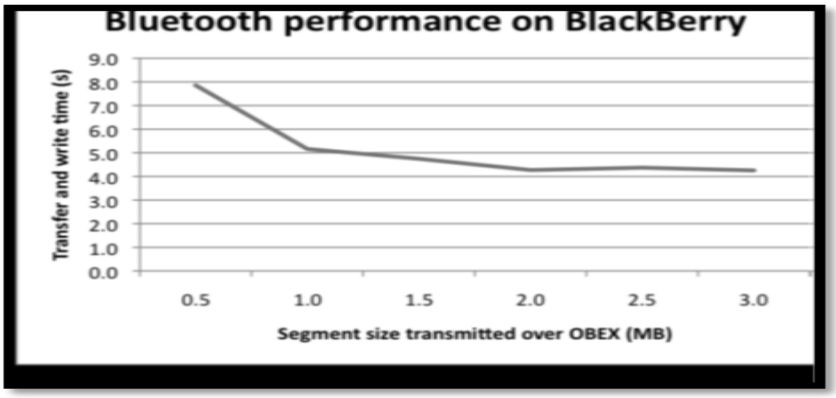

Fig. 4. Normalized transaction times in the BlackBerry device test

B. Windows Mobile

They also implemented a peer-to-peer file sharing protocol as a client application that runs on the Windows Mobile smartphone platform. In contrast to the serial port implementation on BlackBerry. They utilized the User Datagram Protocol (UDP) transport using socket connections. They have connected multiple Windows Mobile devices together to measure the throughput of file transfers with multiple sources seeding a single file. It varied two parameters:

1. The number of devices i.e., 2 peers communicating with each other, versus 3 and 4 peers in total,
2. The segment size i.e., from 100 to 1400 bytes, in 100- byte increments.

The mobiles devices utilized were Compaq iPAQ 3870 units with an Intel Strong ARM-based 206 MHz processor, 64 MB RAM, and running the Pocket PC 2002 operating system. Each packet was transmitted 10,000 times, at duration of approximately 300 seconds overall, to obtain average measurements. The main difference in the Windows CE Bluetooth stack compared to that of the BlackBerry is the TDI (Transport Driver Interface), an interface that serves as an adaptation layer to the socket interface API and is implemented above RFCOMM.

The throughput remained largely the same, with varying packet sizes, evidently due to segmentation into smaller packets occurring in the L2CAP layer. The L2CAP resource manager can actually perform traffic shaping to ensure that the protocol data units (PDU's) sent to the baseband conform to a specified quality of service. L2CAP is similar to UDP, but it enforces delivery order. The default maximum packet size is 672 bytes, but this can be negotiated up to 65,535 bytes.

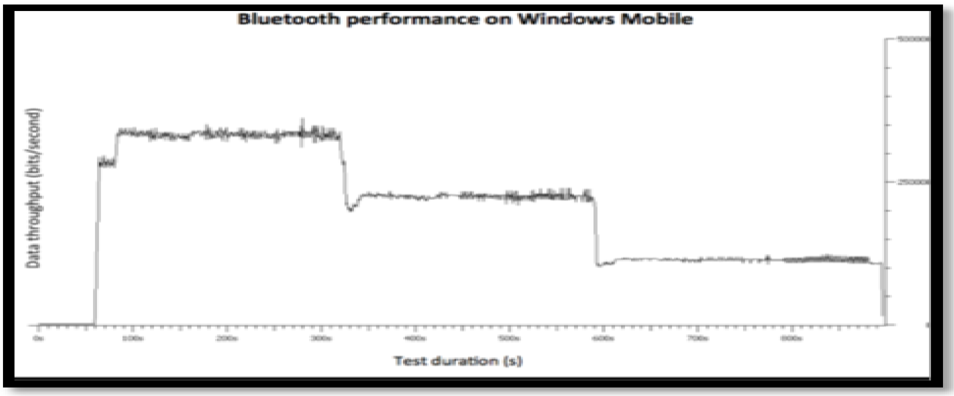

Fig. 5. Transaction times in the Windows Mobile device test, with different peer numbers and segment sizes utilized. The throughput rate levels shown (in bits per second) are for 2, 3, and 4 peers communicating with each other, respectively, in three successive blocks of time, each approximately 300 seconds in duration. Within each block, for the same number of peers, the segment size was increased from 100 to 1400 bytes, but had little effect on the rate.

VI. Conclusion

The Service Based model that permits efficient file sharing between mobile smartphones over a low-cost transport using Bluetooth. This results lead us to conclude that Service Based file transfers between today's mobile devices are practical using Bluetooth. But severe restrictions must be enforced on the transfer of data. Where the Gnutella networks are used in the mobile device environment. Gnutella networks are uses in order to reduce the amount of traffic for the mobile device and providing support for device mobility. Where the Service Based Sharing allows for efficient content distribution using low-cost links that do not impose a load on the mobile carrier infrastructure. Through experimentation on leading smartphones, we have found various optimal strategies, including minimizing the upload-to-download ratio to conserve battery life, using larger file segments to increase throughput, and using sockets to decrease memory overhead.

References

[1] A. Oram, "Peer-to-Peer: Harnessing the Benefits of a Disruptive Technology". O'Reilly, 2001.

[2] K. Kant, "An analytic model for peer to peer file sharing networks," in IEEE International Conference on Communications, May 2003, pp. 1801 – 1805 vol. 3.

[3] A. Legout, G. Urvoy-Keller, and P. Michiardi, "Rarest First and Choke Algorithms Are Enough," in IMC'06, Rio de Janeiro, Brazil, 2006.

[4] Yang, X. et al, "Service capacity of peer to peer networks," in 23rd Joint Conference of IEEE Computer and Communications, 2004.

[5] Hu, T. H. et al, "Supporting mobile devices in Gnutella file sharing network with mobile agents," in 8th IEEE Int. Symposium on Computers and Communication, Sep. 2003, pp. 1035 – 1040 vol. 2.

[6] Dhurandher, S. K. et al, "A swarm intelligence-based p2p file sharing protocol using bee algorithm," in IEEE/ACS Int. Conf. on Computer Systems and Applications, 2009.

[7] Asadi, M. et al, "A scalable lookup service for p2p file sharing in manet," in Proc. of the 2007 Int. Conference on Wireless Comm. and Mobile Computing. New York, NY, USA: ACM, 2007.

[8] Huang, C.-M. et al, "A file discovery control scheme for P2P file sharing applications in wireless mobile environments," in Proc. of the 28th Australasian Conference on C.S., 2005.

[9] O. Ratsimor, D. Chakraborty, A. Joshi, T. Finin, and Y. Yesha, "Service discovery in agent-based pervasive computing environments," Mob. Netw. Appl., vol. 9, no. 6, pp. 679–692, 2004.

[10] G. P. Perrucci and F. Fitzek, "Measurements campaign for energy consumption on mobile phones," Aalborg University, Tech. Rep., 2009.

[11] A. Kammer et al, "Bluetooth Application Developer's Guide: The Short Range Interconnect Solution". Rockland, MA: Syngress, 2002.

[12] D. Barkai, "Peer-to-Peer Computing". Intel Press, 2001.

[13] S. Androutsellis-Theotokis et al, "A survey of peer-to-peer content distribution technologies," in ACM Computing Surveys, Dec. 2004.

[14] M. Connolly et al, "Analysis of UDP Performance over Bluetooth," in IT&T2003: Proc. of the IT & Telecommunications Conference, Oct. 2003.

[16] A. Misic and V. Misic,"Performance Modeling and Analysis of Bluetooth Networks". Boca Raton, FL: Auer Bach Publications, 2006.

[17] A. Huang et al, "Bluetooth Essentials for Programmers". Cambridge University Press, 2007