Indian Institute of Information Technology, Allahabad

Project Report

Load Balancing in Mobile Peer to Peer Network

Arpita Saha	IIT2014134
Bhairavee Bawane	IIT2014070
Harsh Shah	IIT2014071

22nd November, 2017

CANDIDATES DECLARATION

We hereby declare that the work presented in this project entitled **Load Balancing in Mobile P2P"**, submitted as a VII Sem B.Tech(IT) mini project, is an authentic record of our original work carried out under the guidance of Dr. Bibhas Ghoshal. Due acknowledgements have been made in the text to all other material used.

Place: IIIT Allahabad
Date: 22nd November, 2017

Arpita Saha (IIT2014134)
Bhairavee Bawane (IIT2014070)
Harsh Shah (IIT2014071)

CERTIFICATE FROM SUPERVISOR

This is to certify that the project work Load Balancing in Mobile P2P is a bonafide work of Arpita Saha (IIT2014134), Bhairavee Bawane (IIT2014070), Harsh Shah (IIT2014071) who carried out the project work under my supervision.

Place: IIIT Allahabad
Date: 22nd November, 2017

Dr. Bibhas Ghoshal
IIIT Allahabad

ACKNOWLEDGEMENT

We acknowledge with much appreciation the crucial role of Dr. Bibhas Ghoshal for his contribution in this endeavor of ours. His efforts, constant support and perseverance has guided us through this project. His engagement through the process of this project has been precious and irreplaceable.

Abstract

Smartphones present unique challenges to the technologies designed for traditional devices due to their limited battery, memory and bandwidth. Currently, file sharing in smart phones is done by familiar server-client model. This method takes up a lot of bandwidth and battery power which is unacceptable in mobile devices. This project present a load balanced scheme of file sharing in the mobile networks. We present the method of electing a smart head among the group of peer devices. Smart head's responsibility include maintaining the list of available files on the network, distributing the file for optimal load balancing, and brokering the communication between peers. An android application was created to demonstrate the feasibility of this scheme and to test its performance against the traditional schemes.

Contents

1 Introduction

This is the era of smartphones. Smartphones are present everywhere with everyone having access to at least one device. Studies suggest that in 2017, there are estimated 2.32 billion smartphones in the world.[1] Many such devices can connect to the Internet and interconnect with other devices in the network via WiFi, Bluetooth, Cellular networks and Near Field Communication (NFC). Future is composed of billions of mobile devices connected to the Internet, continuously transmitting data.

Mobile devices often contains sensors such as accelerometers, magnetometers and gyroscopes allowing detection of orientation and motion. Mobile devices may also provide biometric user authentication such as face recognition or fingerprint recognition. This presents an opportunity to build innovative applications and services by exploiting these devices untapped rechargeable energy, sensing, networking and processing capabilities. These devices can be used to create smart homes, smart commercial buildings and smart cities.

Internet of Things(IoT) is the new trend in the era smartphones. The Internet of things (IoT) refers to the inter-networking of smart devices, buildings, and other items embedded with electronics, software, sensors, actuators, and network connectivity which enable these devices to collect and exchange data. New devices with low energy communication abilities are continuously being integrated into our daily lives.

2 Literature Survey and Motivation

Although smartphones have variety of features, they are not without limitations. Since devices are mobile, they must be operated by battery. Further, smartphones tend to have low processing power and memory as both require battery power. Smartphones extensively use wireless connections to communicate. And since wireless networks typically have low bandwidth compared to their wired counterparts, network bandwidth is quite limited on smarphones. Hence, smartphones present unique challenges with its limited battery power, CPU, bandwidth and memory.

Technology has not been updated to address these challenges. Today, smart-

phones still use many technologies made for desktops and personal computers. Such technologies may not be designed to efficiently utilize smartphone's limited resources. Also, they may not use the features provided by smartphones to efficiently perform the tasks.

One of the example of such technology is file sharing. Currently, file sharing is done either by client - server model or peer-to-peer model. In client-server model, one of the devices called as server takes the complete responsibility of transferring file to the client. Both the client and the server must be alive during the complete file transfer. Similar is the case for peer-to-peer model without load balancing, in which both peers must be alive the full duration. This means resources such as battery, memory and bandwidth is consumed on both devices the entire time.

Karger et al. [2] discusses about new algorithms for load balancing in peer to peer networks. They give new protocols for many scenarios, whose performances are within a constant factor of optimal. Their protocol eliminates the requirement of virtual nodes while maintaining a balanced load.

Load balancing concepts can be applied to IoT scenario linked to smart devices. De Masi [3] presents a peer-to-peer load balancing scheme that can distribute tasks among plethora of mobile smart devices in the IoT world. This scheme is able to reduce the traffic send to the Cloud and the energy consumption of the devices.

3 Requirement Analysis

We present the solution of above problem by using load balancing in peer to peer network. The aim of our project is addressing the development of load balancing technique for P2P network of mobile smart devices, allowing load-balanced distribution of tasks among peers.

Peer-to-peer (P2P) computing or networking is a distributed application architecture that partitions tasks or workloads between peers. Peers are equally privileged, equipotent participants in the application. Load balancing is the distribution of workloads across multiple computing resources, such as computers, a computer cluster, network links, central processing units, or disk

drives. Load balancing aims to optimize resource use, maximize throughput, minimize response time, and avoid overload of any single resource.

Based on the above aim, we gather the following as project requirements:

1. Upload a new file.
2. Show the list of files available on the network.
3. Download a file from the list of available files.
4. Elect a smart head to track the status of the files and manage communications between peers.

4 System Design

We divide the task of file sharing into many modules described as follows:

4.1 Smart Head

Smart head is the device which will be maintain the list of available files on the network, perform the load balancing and manage the communication between peers.

4.2 Device Discovery

Devices are discovered in the local network using UDP broadcast. Any new device connecting to the network will announce its presence to the network. All the connected devices will update their list of present devices and will respond back to the announcement.

4.3 Election

Election process will happen in the new device whenever it connects to the network. This new device will ask for resources present in all other devices

and run the election. After election, it will announce the presence of new smart head to all the devices in the network.

4.4 File Uploading

All new devices in the network will announce their list of files on the network as soon as they join. Present devices will update their own list of files in the network. This list is then shown to the users, after which they can select which files they wish to download.

4.5 File Downloading

To download a file, a device will send download request to smart head. Smart head will divide the file into many parts, based on resources present with each device. Smart head will forward transfer requests to the different devices, after which file transfer takes place between devices.

5 Implementation

We created an Android application to test our presented idea for file sharing. Android Studio was used to create the application. Following subsections discusses important implementation details.

5.1 Communication

Communication between devices is done by sending UDP broadcast packets across the network. Commands are sent across the network, each device will respond as per the command. All the devices will listen to commands on UDP port 5678.

5.2 Election Algorithm

We decide the smart head based on election algorithm as follows:

1. Find devices with charging status.
2. If there are more than one devices with charging status and battery greater than 30%, select the one with highest combination of battery level and available memory from the charging ones.
3. Combination of battery level and memory is calculated as 0.75 * battery percentage + 0.25 * normalized available memory.
4. In case of no devices with charging status, select the device with highest combination of battery level and memory.
5. If new smart head is different than the old one, inform all devices in the network about the new smart head.

5.3 Sending File and Receiving File

Files are sent using TCP. The receiving device will listen for connections at TCP port 6579. Sending devices will open an unused port and connect to receiving device at port 6579. File transfer takes place once the connection is established. Files are sent as raw bytes across the network. File is divided into different parts before sending for load balancing purposes.

5.4 File Division

We have performed weighted division of file as follows:

1. Calculate combination of battery level and memory as ((0.75 * battery percentage) + (0.25 * normalized available memory)) for each device.
2. Calculate percentage for each device by dividing the current device's combination by sum of all device's combination.
3. Multiply file size by the above percentage to obtain the part of file for each device.

5.5 Handling device dropping

Whenever a device leaves the network, it will send QUIT command to the network. Other devices will remove all the associated entries with that device and will show the updated list to the user. If the dropped device was a smart head, election will take place again and new smart head will be elected.

6 Software Testing

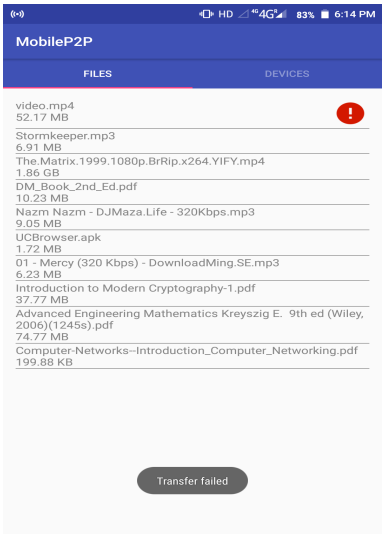
We tested our android application to verify the claim presented in the report. For testing purposes, we selected a video file having size of 52.17 MB. First, we transferred it from one device, 25.41 seconds were taken to complete the transfer. Then we transferred it from two devices and it took 18.3 seconds. This result is tabulated as follows:

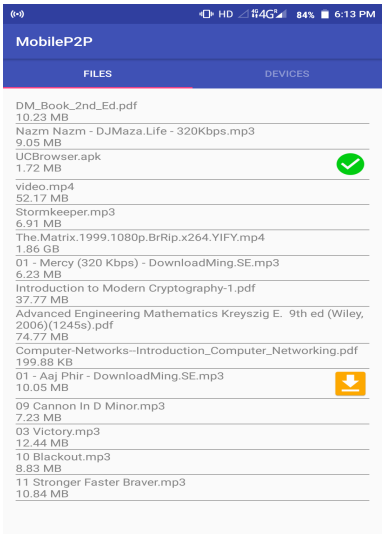
No. of devices	Time taken	Transfer Speed
1	25.41	2.05 MB/s
2	18.3	2.85 MB/s

As we can see, there's an increase in transfer speed when file transfer is taking place by two devices. Hence, we can clearly see decrease in download time and increase in transfer speed.

7 Project Screenshots

First screenshot displays an example of failure in file transfer due to disconnection of one of the devices. Second screenshot shows an example of on-going file transfer. Third one shows the list of available devices in the network.





8 Conclusion and Future Scope

As seen from the testing section, our proposed scheme achieved significant reduction in download time even for a file size as small as 50 MB. Therefore, this scheme can replace traditional schemes in many areas. This will help reduce the amount of battery consumption and network bandwidth used in mobile devices.

A References

[1] Statistica. (2017). Number of smartphones users worldwide, [Online]. Available: https://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide/ (visited on 2017).

[2] D. Karger and M. Ruhl, "New algorithms for load balancing in peer-to-peer systems", Jan. 2003.

[3] A. De Masi, "Load balancing in p2p smartphone based distributed iot systems", 2015.

B Suggestion by Board Members