INDIAN INSTITUTE OF INFORMATION TECHNOLOGY,
ALLAHABAD

Load balanced Multimedia Streaming in a Peer-to-Peer Network

Tushar Murarka, IIT2015091
Deepak Yadav, IIT2015124

Supervisor:
Dr. Bibhas Ghoshal

A project report submitted in fulfillment of the requirements for the VII Semester Project

November 18, 2018

Abstract

In the era of the internet, the accessibility of smart devices like mobile phones, tablets has increased by a large amount. Almost everyone in todays era has an access to one of such smart devices. These devices are equipped with advance features and are capable of doing fast processing and are also capable of storing multimedia files and almost all types of data.

In this project, we propose a peer-to-peer (p2p) network of such smart devices which is built over WiFi through an android application. This p2p network of smart devices with different processing, storing capabilities can be used in several practical applications such as crowdsensing or it can be used as a distributed multimedia storage like BitTorrent. The application that we are targeting in this project is distribution of multimedia content in a load-balanced, distributed fashion.

The key idea is to facilitate the user in the network to seamlessly stream any media file present in the network. The major challenges in implementing such scheme is creating the network of devices, maintenance of information present on all peers, developing an algorithm that ensures the multimedia streaming requests are balanced according to the load and ensures that the streaming is seamless even if multiple nodes (peers) are involved.

The nodes can participate in the network by sending candidacy request to the network. Once the nodes are joined, a leader node referred as Smart Head is elected among all the nodes by running a leader election algorithm. The peers in the network can request the smarthead to stream any media file present in the network. The smart head ensures that the media file is seamlessly streamed keeping factors like load balancing into consideration.

Keywords :- Peer-to-peer network, load balancing.

DECLARATION

We hereby declare that the work presented in this project report entitled **Load Balanced Multimedia streaming in a Peer-to-Peer Network**, submitted as mid-semester report of 7th Semester report of B.Tech. (IT) at Indian Institute of Information Technology, Allahabad, is an authenticated record of our original work carried out from July 2018 to December 2018 under the guidance of Dr. Bibhas Ghoshal. Due acknowledgements have been made in the text to all other materials used. The project was done in full compliance with the requirements and constraints of the prescribed curriculum.

Tushar Murarka, IIT2015091
Deepak Yadav, IIT2015124

Place: Allahabad
Date: November 18, 2018

Contents

Chapter 1

Introduction

The growth in the use of smart devices like smartphones, ipads, laptops is continuously increasing. These devices have excellent sensing capabilities, computing capabilities and have power to process and store data. These capabilities can be exploited to build smart and innovative solutions around this distributed peer to peer network of devices.

Individual devices might not be capable of solving complex tasks but when a network of devices, connected in a peer to peer fashion, come together, complex tasks or computations can be done over this network by efficiently distributing the tasks among these devices and balancing the load of the network. Some of the problems that can be solved using the proposed network of devices are crowdsensing, data-preprocessing and filtering for cloud platforms, big-data mining also known as edge mining [1], etc. Not only IoT related solutions but the proposed scheme can be used as a data store which can facilitate the peers in the network to access files which have size beyond the capacity of the individual device.

The aim of this project is to exploit the power of one such p2p network of smart devices. In this project we create a p2p network where the peers in the network can stream media files which are present on any of the nodes in the network. Though its a p2p network but theres a need of a centralised controller who controls the overall operations, serves and distributes the requests to desired nodes, maintains the status and information of each of the peer. We determine this node by conducting a leader election algorithm. All the requests in the network then goes through this leader. Load balancing is one important factor for the project as if its not considered, some peers in the network can become bottleneck. Hence, proper load balancing algorithm is designed in order to ensure that the file if present on multiple nodes, is strategically divided into parts according to the individual node capacity and power. One major challenges which aroused from the streaming the file

into parts is to ensure that the file is streamed progressively without breaks. To prove this and to evaluate the proposed model, an android application is built.

Chapter 2

Motivation

Nowadays, with the advancement of the technology, smart devices such as smartphones, tablets, laptops, desktop systems have become a basic need of the society. Almost every individual have an access to at least one of these devices. Majority of these devices have the capabilities such as processing power, ability to store data and multimedia files, ability to connect to wireless network, data transmission capabilities and so on.

The clusters or networks of smart devices can come out as platforms with great computing capabilities and have the capability to solve complex tasks with ease. The best part is the low computation cost when compared to conventional cloud based platforms. These individual devices have good computing capability, good storage and are equipped with WiFi which can itself serve as great medium to share data and information at good bandwidth.

Multimedia streaming has seen a tremendous growth as its one of the ways to see user-created content over the internet. Platforms like Youtube, TikTok, Instagram are the major stakeholders in this industry. But when it comes down to streaming videos in a local area, in a small vicinity where the bandwidth is not that high, there are no such applications built which provide such functionality. The motivation for the project comes from the unavailability of such platform. Distributed p2p networks has been in the research for quite a long time. We, by doing this project, want to explore and proof that these networks have a very important use case in this domain as well.

Chapter 3

Analysis of Previous Work

A lot of work has already been done in the domain of load balanced p2p network. Most common p2p network that we find almost everywhere is local area network but it does interests many researchers. Bittorrent is one of the most commonly used protocols for transferring electronic files over the internet. There are many other peer-to-peer file sharing architecture like Napster [1], Gnutella, FreeNet etc. These architectures over the time proved the potential of distributed data sharing and computation.

A.D Masi [2] explored p2p network especially for IOT devices and they proposed state of the art algorithm for dynamic load balancing. He also collected data from experimental outcomes that was used to determine the feasibility and cost-effectiveness of a load balanced P2P smart phone-based applications. He in his research showed that the use of a p2p network of devices was able to reduce the traffic sent to the cloud by a significant amount.

Graffi et.al [3] explored the load balancing for multimedia streaming in heterogeneous cluster. They proposed a DTH based load balancing algorithm and they achieved 53% better load balancing for peer-to-peer network. This network has various types of systems raging from old commodity personal computers to latest modern systems.

Chapter 4

Background

4.1 Peer-to-Peer Network

A peer-to-peer network means a network of peers (computing devices) that are interconnected and can share information, resources among themselves without the requirement of a centralized web server. Each of the peer in the network can act as a client as well as a server. Examples of p2p networks are BitTorrents, Napster, etc.

Figure 4.1 shows a p2p network of 6 nodes where each of the nodes are interconnected via some physical links (ex. LAN cables).

4.2 Distributed Computing

Distributed computing refers to use of distributed systems for computation [4]. When systems (nodes) of a network coordinate and communicate among themselves to achieve a common goal, the overall process of computation is called distributed computing. Distributed computation involves resources from different nodes of the network that may be geographically well apart from each other. The main goal in a distributed computing environment is to make use of the available resources to the fullest however taking care of problems like fault tolerance, node crashing, etc.

4.3 Load Balancing

When theres a cluster of nodes or a network of nodes in the backend, it becomes important for the system to distribute the requests among this network of nodes in such fashion that a single or a couple of systems do not become

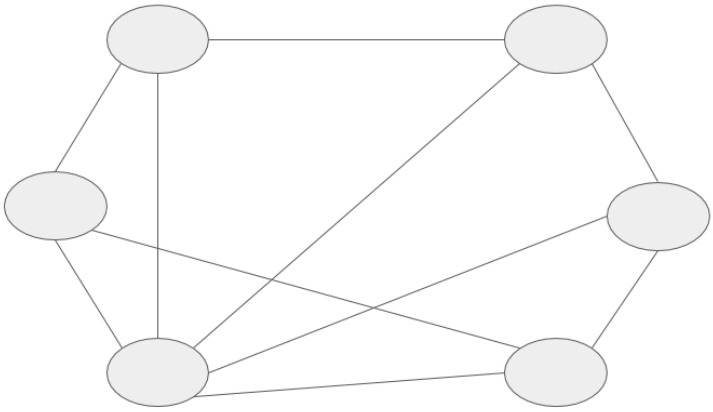

Figure 4.1: A P2P network of computers

the bottleneck, i.e. majority requests are being routed to them and the rest of the nodes remain idle. The load can be anything, client requests, amount of data to store, etc. The algorithm used to distribute this load among the nodes where it takes care to balance the load among the nodes so as the increase the throughput of the system and to enhance the performance, such algorithmic technique is called load balancing.

Load balancing is very crucial when it comes down to performance of a distributed system. The better the load balancing technique used, the better the performance. Some of the famous load balancing techniques include Consistent Hashing [5], Rendezvous Hashing [6].

Chapter 5

Problem Definition

5.1 Problem Statement

1. To design a model in which smart devices are connected over Wifi in p2p fashion for streaming multimedia content.

2. To propose a load-balancing algorithm to ensure that the streaming (when multiple devices are involved as seedrs) is done considering the load on an individual device.

3. To develop an android application in order to implement and evaluate the proposed model.

5.2 Constraints

1. No node can leave the network when a file is being streamed.

2. Fault Tolerance issues like crash/failure of nodes are not being considered.

Chapter 6

Proposed Methodology and Implementation

As a proof of concept, we have developed an android application which provides an interface to the end user to join the P2P network and stream videos stored on peers' devices. All the complex tasks like smart head election, request allocation and load balancing, are abstracted from the end user. All user sees is an uninterrupted streaming of video. Following subsections discuss the implementation in detail.

6.1 Peer's Arrival Announcement

A network is formed from the devices connected to the a WiFi network. When a device joins the network, it announces its arrival to the network and election is conducted to elect the new smart head. Peer with the maximum battery percentage is elected as the smart head. This is achieved using Java multicast sockets. Each peer listens to a multicast IP address chosen at the time of implementation and when a new device joins the network, it sends its resources information to all other peers in the network.

In Figure 6.1 battery percentage is given for all the peer in the network and as a result of smart head election, Peer with maximum battery percentage is elected as the smart head.

6.2 File Indexing and Hash Table

To compare the video files efficiently, we index them according to their checksum. Specifically, we have used md5Sum in our implementation. It is safe to assume that different video files will have unique md5Sum but according to

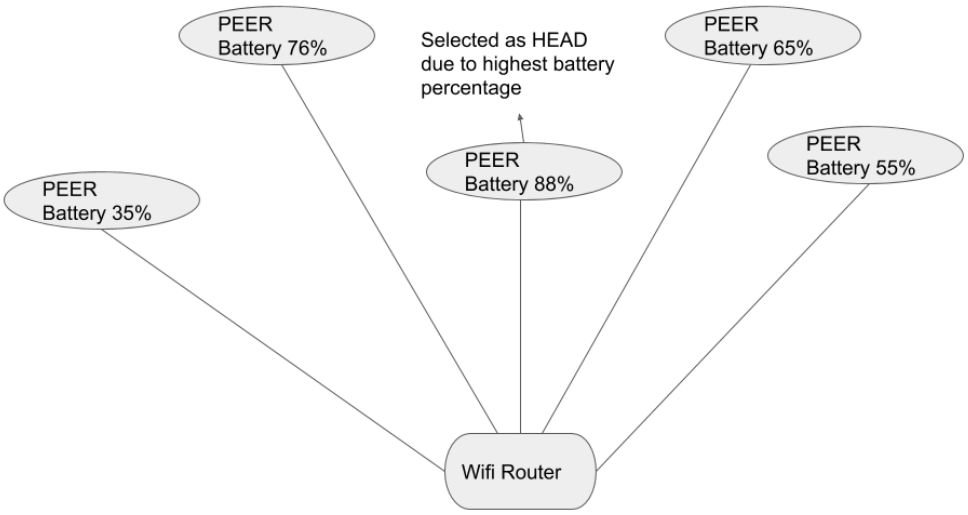

Figure 6.1: Peer with the maximum battery percentage is elected as the smart head

pigeon hole principle, it can never be achieved in theory. After a new smart head is elected, every peer in the network sends following hash table to the smart head.

Key	Value
md5Sum of file	{"size of file in seconds", "name of file", "location of file"}

Table 6.1: Hash table sent to the leader by each peer

Smart head collects the above table from each peer and itself maintains the following table.

Key	Value
md5Sum of file	{"file name", "file size in seconds", "list of peers having this complete file"}

Table 6.2: The centralised hash table maintained by the leader node

Since the same file may have different file name on different devices, therefore we select any one file name from the list. List of peers contains a Peer object corresponding to each peer having the complete file, this object has all

the basic information of the peer device like battery percentage, IP address and unused RAM.

6.3 Request for video files

When the user taps on Request button, a network call is made to the smart head to get the list of all the files in the network. Smart head returns the following table to the requesting peer and peer uses this table to present the this list to the end user.

Key	Value
md5Sum of file	Name of the file

Table 6.3: Information sent by the smart head to the peer requesting to view the files

When the user taps on a particular file name, a network call is made to the smart head with md5Sum of the file as payload. As mentioned earlier that smart head maintains the hash table of all the files in the network, therefore smart head figures out which all peers have the requested file and it sends the following table to requesting peer.

Key	Value
IP address of peer having the file	Duration in seconds to be streamed

Table 6.4: Information containing the duration of the video file to be streamed from the different peers

Using the above table requesting peer starts streaming sequentially from other peers.

In Figure 6.2, one peer sends md5sum of a video file it wants to stream and smart head in return sends details of the peer having this file.

6.4 Load Balancing

To balance the load of the network, smart head takes battery percentage of the devices into account. Complete file is divided into several chunks and this division is done as per battery percentage ratio.

$$\frac{battery\ percentage}{Sum\ of\ battery\ of\ all\ peers\ in\ the\ network} \times Total\ length\ of\ video \tag{6.1}$$

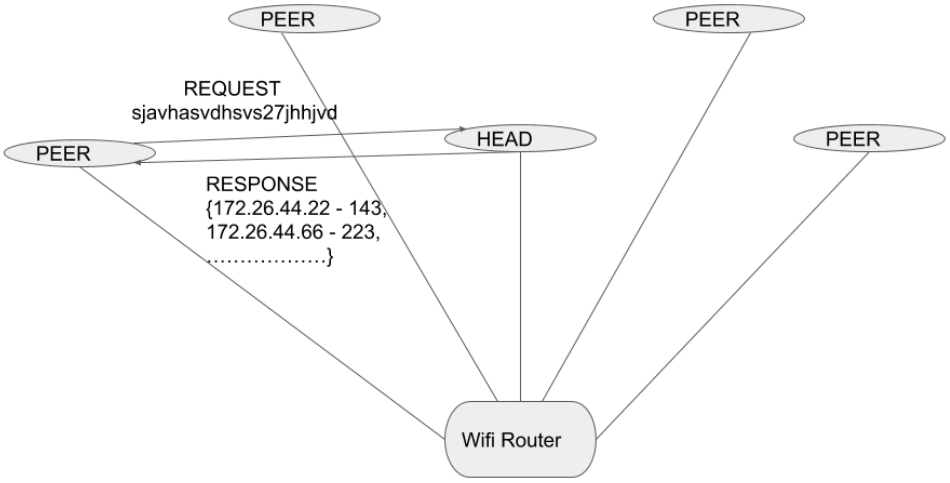

Figure 6.2: Information transfer between peer and the smart head

In this way, device with maximum battery percentage gets to deliver maximum length of video. We have not considered cpu and primary storage as a deciding factor of for dividing the file because nowadays very device has enough resources to perform complex tasks.

Chapter 7

Results and Conclusion

We developed an android app which helped us to validate that load balancing concepts can be applied to smart devices connected to the same wireless network. This model can help to reduce load on cloud based servers by reusing data present in the local network. We considered battery as the only factor to distribute load to the entire network because nowadays cpu and memory is not an issue for smart devices, every device has at least 2 GB RAM and 1.8 Ghz processor. Electing smart head every 30 minutes helps to avoid smart head to become the bottleneck of the entire system. Since this model is only applicable for the devices in the vicinity, therefore streaming latency is negligible. Few screenshots of the android application are given below.

In Figure 7.1, Screen A is the handling screen of the application. It shows the IP address, RAM Unused and Battery percentage of the device. When user taps in JOIN IN button, a broadcast request is sent to all the peers and election is held to elect smart head.

In Figure 7.2, Screen B shows smart heads IP address and battery percentage. Smart head is responsible for the load balancing part. When user taps REQUEST VIDEO, a request is sent to the smart head, to get the list all video files present in the network.

In Figure 7.3, Screen C shows the list of all the video file present in P2P network. User can start streaming video file by just tapping on the file name.

In Figure 7.4, Screen D shows the playing video, user can pause and resume the video at any point.

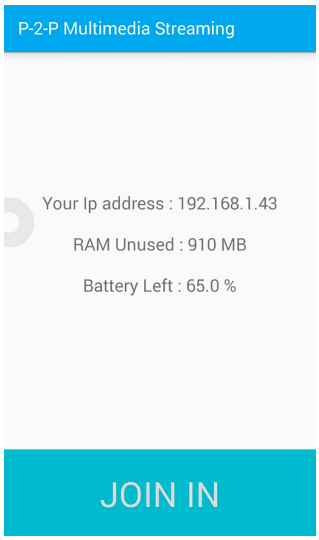

Figure 7.1: Screen A

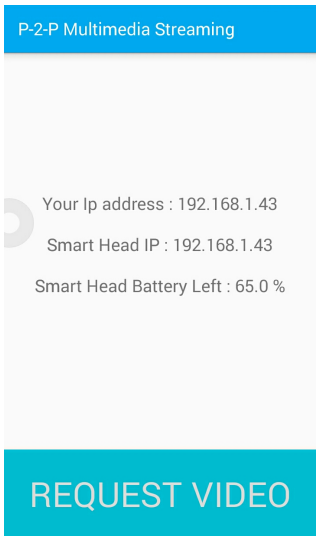

Figure 7.2: Screen B

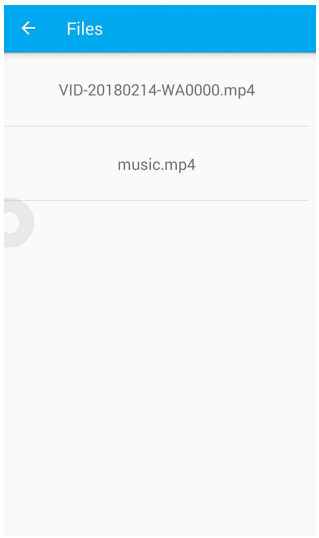

Figure 7.3: Screen C

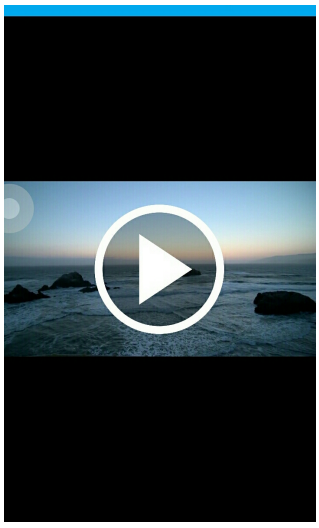

Figure 7.4: Screen D

Chapter 8

Future Work

The proposed model acts as a basic prototype for a load-balanced content distribution network. It has a huge scope for improvement in some critical areas namely Fault Tolerance, Node departure, etc. Some of the points where we would like to work upon in the future are -

1. Node Departure - Proposing new techniques/methods to handle departure of nodes from the network.

2. Fault Tolerance - Considering and handling cases of Fault tolerance such as node crash, failure, etc.

3. Permission based sharing - To add permissions to the end user as what all videos he/she wants the other users in the network to see.

Bibliography

[1] E. I. Gaura, J. Brusey, M. Allen, R. Wilkins, D. Goldsmith, and R. Rednic, "Edge mining the internet of things," *IEEE Sensors Journal*, vol. 13, no. 10, pp. 3816–3825, 2013.

[2] A. De Masi *et al.*, "Load balancing in p2p smartphone based distributed iot systems," 2015.

[3] K. Graffi, S. Kaune, K. Pussep, A. Kovacevic, and R. Steinmetz, "Load balancing for multimedia streaming in heterogeneous peer-to-peer systems," in *Proceedings of the 18th International Workshop on Network and Operating Systems Support for Digital Audio and Video*, pp. 99–104, ACM, 2008.

[4] Wikipedia contributors, "Distributed computing — Wikipedia, the free encyclopedia." `https://en.wikipedia.org/w/index.php?title=Distributed_computing&oldid=867913365`, 2018. [Online; accessed 18-November-2018].

[5] Wikipedia contributors, "Consistent hashing — Wikipedia, the free encyclopedia." `https://en.wikipedia.org/w/index.php?title=Consistent_hashing&oldid=865828440`, 2018. [Online; accessed 18-November-2018].

[6] Wikipedia contributors, "Rendezvous hashing — Wikipedia, the free encyclopedia." `https://en.wikipedia.org/w/index.php?title=Rendezvous_hashing&oldid=867073959`, 2018. [Online; accessed 18-November-2018].

[7] B. Carlsson and R. Gustavsson, "The rise and fall of napster-an evolutionary approach," in *International Computer Science Conference on Active Media Technology*, pp. 347–354, Springer, 2001.