Department of Information Science and Engineering

Global Campus, Jakkasandra Post, Kanakapura Taluk, Ramanagara District, Pin Code: 562 112

2020-2021

A Project Phase 2 Report on

"MULTIMEDIA STREAMING AND DATA TRANSFERRING"

Submitted in partial fulfilment for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

INFORMATION SCIENCE AND ENGINEERING

Submitted by

V K SAKTHI SWARUP
17BTRIS044

JONNAVARAM MAHESWAR REDDY
17BTRIS019

GANGIREDDY HARI SAINATH REDDY
17BTIRS018

PARA ASHOK
17BTRIS024

Under the guidance of

Prof. Mathiyalagan R
Assistant Professor
Department of Information Science & Engineering

Faculty of Engineering & Technology
JAIN Deemed to be University

Department of Information Science and Engineering

Global Campus, Jakkasandra Post, Kanakapura Taluk, Ramanagara District, Pin Code: 562 112

CERTIFICATE

This is to certify that the project work titled **"MULTIMEDIA STREAMING AND DATA SHARING"** is carried out by **V K Sakthi Swarup (17BTRIS044), Jonnavaram Maheswar Reddy(17BTRIS019), Gangireddy Hari Sainath Reddy (17BTRIS018), Para Ashok (17BTRIS024),** a bonafide students of Bachelor of Technology at the Faculty of Engineering & Technology, Jain Deemed-to-be University, Bangalore in partial fulfillment for the award of degree in Bachelor of Technology in Information Science & Engineering, during the year **2020-2021**.

Prof. Mathiyalagan R
Assistant Professor
Dept. of ISE,
Faculty of Engineering & Technology,
Jain Deemed to be University
Date:

Prof. Arun N
Head of the Department,
Dept. of ISE,
Faculty of Engineering & Technology,
Jain Deemed to be University
Date:

Dr. Hariprasad S A
Director,
Faculty of Engineering & Technology,
Jain Deemed to be University
Date:

Name of the Examiner	Signature of Examiner
1.	
2.	

DECLARATION

We, **V K Sakthi Swarup (17BTRIS044), Jonnavaram Maheswar Reddy (17BTRIS019), Gangireddy Hari Sainath Reddy (17BTRIS018), Para Ashok (17BTRIS024),** are student's of eighth semester B.Tech in **Information Science & Engineering**, at Faculty of Engineering & Technology, **Jain Deemed to-be University**, hereby declare that the project titled **"Multimedia streaming and data sharing"** has been carried out by us and submitted in partial fulfilment for the award of degree in **Bachelor of Technology in Information Science & Engineering** during the academic year **2020-2021**. Further, the matter presented in the project has not been submitted previously by anybody for the award of any degree or any diploma to any other University, to the best of our knowledge and faith.

Signature

Name1: V K Sakthi Swarup
USN: 17BTRIS044
Name2: Jonnavaram Maheswar Reddy
USN: 17BTRIS019
Name3: Gangireddy Hari Sainath Reddy
USN: 17BTRIS019
Name4: Para Ashok
USN: 17BTRIS024

Place : Bangalore
Date :

ACKNOWLEDGEMENT

It is a great pleasure for us to acknowledge the assistance and support of a large number of individuals who have been responsible for the successful completion of this project work.

First, we take this opportunity to express our sincere gratitude to Faculty of Engineering & Technology, Jain Deemed to be University for providing us with a great opportunity to pursue our Bachelor's Degree in this institution.

In particular we would like to thank ***Dr. Hariprasad S A****, Director, Faculty of Engineering & Technology and* ***Dr. Kuldeep Sharma,*** *Dean Department of Computer Science and Engineering, Jain Deemed to be University for their constant encouragement and expert advice.*

It is a matter of immense pleasure to express our sincere thanks to ***Prof. Arun N*** *,* ***Head of the department****,* ***Information Science & Engineering****, Jain Deemed to be University****,*** *for providing right academic guidance that made our task possible.*

We would like to thank our guide ***Mathiyalgan R****, Professor,* ***Dept. of Information Science & Engineering****, Jain Deemed to be University, for sparing his/her valuable time to extend help in every step of our project work, which paved the way for smooth progress and fruitful culmination of the project.*

We would like to thank our Project Coordinator ***Prof. Soumya K N/Dr. Jai Ganesh*** *and all the staff members of Information Science & Engineering for their support.*

We are also grateful to our family and friends who provided us with every requirement throughout the course.

We would like to thank one and all who directly or indirectly helped us in completing the Project work successfully.

Signature of Students

ABSTRACT

- As we know sharing and streaming of data has been a major concern these days and internet is required to share the data between parties. Streaming and sharing larger data over the internet could be a costly process. Instead, through our application we can minimize this issue to some extent by transferring the data/files between two parties which are connected locally (through the application) and without help of internet.

- The goal of our project is to represent the optimized Mobile Web Services (MobWS) on the android platform providing the web services. Now a days the use of smart phone devices having high hardware configuration are rapidly increasing enabling the sharing of video, images, live event. People can share the live events by streaming mobile to mobile within the same network. Our project proposes the applications which enable the hosting of websites on android smart phones which can be accessible by any http web browser.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter – 1

1. Introduction

- **Streaming media** is multimedia that is constantly received by and presented to an end-user while being delivered by a provider. The Stream refers to the process of delivering or obtaining media in this manner. Streaming refers to the delivery method of the medium, rather than the medium itself. Distinguishing delivery method from the media distributed applies specifically to telecommunications networks, as most of the delivery systems are either inherently streaming (e.g. radio, television, streaming apps) or inherently non-streaming (e.g. books, video cassettes, audio CDs). Livestreaming is the real-time delivery of content during production, much as live television broadcasts content via television channels. Livestreaming requires a form of source media (e.g. a video camera, an audio interface, screen capture software), an encoder to digitize the content, a media publisher, and a content delivery network to distribute and deliver the content. Streaming is an alternative to file downloading, a process in which the end-user obtains the entire file for the content before watching or listening. The term "streaming" was first used for tape drives manufactured by Data Electronics Inc. that were meant to slowly ramp up and run for the entire track; slower ramp times lowered drive costs. "Streaming" was applied in the early 1990s as a better description for video on demand and later live video on IP networks. It was first done by Starlight Networks for video streaming and Real Networks for audio streaming. Such video had previously been referred to by the misnomer "store and forward video

- **File sharing** is the practice of distributing or providing access to digital media, such as computer programs, multimedia (audio, images and video), documents or electronic books. File sharing may be achieved in a number of ways. Common methods of storage, transmission and dispersion include manual sharing utilizing removable media, centralized servers on computer networks, World Wide Web-based hyperlinked documents, and the use of distributed peer-to peer networking. A peer-to-peer exchange is similar to a centrally indexed exchange because you must use a software client to log into the network. The main difference is there is not an indexed server. Each client also acts as a server in this type of exchange. When a search is done each client is asked individually. Some examples of a peer to peer exchange would be KaZaA, Morphius, Overnet, and eMule.Peer-to-peer file sharing is the distribution and sharing of digital media using peer-to-peer (P2P) networking technology. P2P file sharing allows users to access media files such as books, music, movies, and games using a P2P software program that searches for other connected computers on a P2P network to locate the desired content.[1] The nodes (peers) of such networks are end-user computers and distribution servers (not required).Peer-to-peer file sharing technology has evolved through several design stages from the early networks like Napster, which popularized the technology, to later models like the BitTorrent protocol. Microsoft uses it for Update distribution (Windows 10) and online playing games (e.g. the mmorpg Skyforge[2]) use it as their content distribution network for downloading large amounts of data without incurring the dramatic costs for bandwidth inherent when providing just a single source.

Chapter – 2

2. Literature survey

2.1 Research papers:

- **Robust Super-Peer-Based P2P File-Sharing Systems**

This paper has presented an efficient fault-tolerant approach for super-peer-based P2P file-sharing systems. File-query tracking and logical connection techniques are used to tolerate the departure (failure) of the regular peer and super-peer, respectively. Compared with the previous approaches, a shared file in a regular peer is not required to be replicated in other regular peers in advance. In addition, the selection of the backup super-peer can be done easily and efficiently. The simulation results show that the proposed approach has better average query hit rate and average file retrieval latency than the replication-based approaches. With regard to the fault-tolerant overhead, the proposed approach incurs a trivial overhead in the file-query tracking technique. As for the logical connection technique, the static and on demand methods have different overhead trade-off between the storage space required for the additional registrations and the response delay for handling the unsuccessful file query. Our future work will extend the proposed approach to improve the lookup performance in the mobile P2P environment.

- **Enhanced UPnP QoS Architecture for Network-adaptive Streaming Service in Home Networks**

The existing versions of UPnP QoS architecture have several limitations associated with how they provide QoS-guaranteed streaming service over dynamically time-varying networks. In this paper, an enhanced UPnP QoS architecture is proposed to support network-adaptive media streaming in home networks. The proposed streaming scheme is verified by realizing a prototype implementation over an UPnP streaming testbed

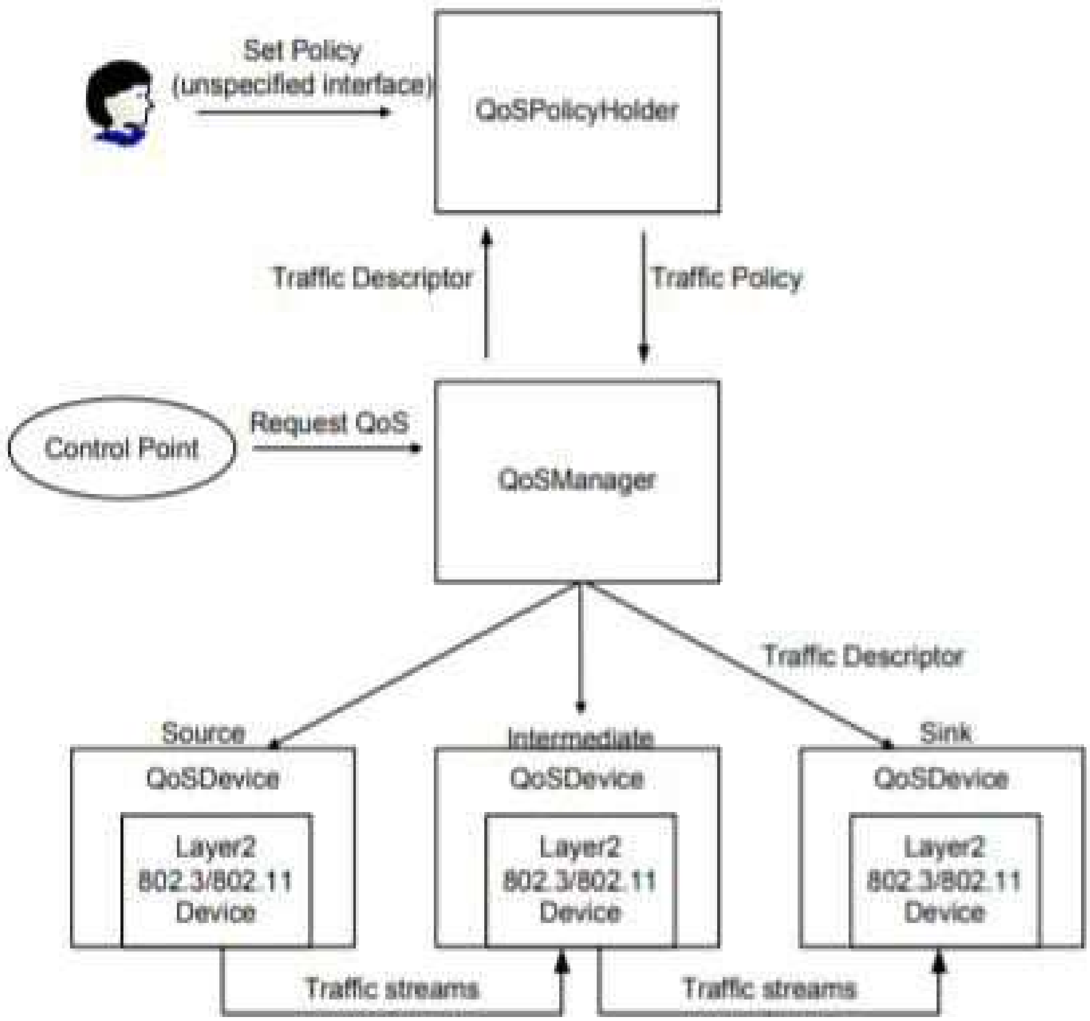

Fig 2.1.1. UPnP QoS architecture overview.

- **Design of a Media Stream Relay Engine on the Android OS**

When large scale multimedia delivery on a single wireless LAN environment, it is difficult to providing smooth multimedia streaming because a large amount of traffic is concentrated on the wireless AP and each device are shared same channel, it is bringing about bandwidth decrease. These problems can be solved using hierarchical network structure, it constructs using Wi-Fi direct network and IEEE 802.11 infrastructure network. However, it networks structure is used for media streaming service, the role of the source device is important. Because of delay of group owner may be affected to performance of streaming data transmission of whole Wi-Fi direct network group. And the group owner device can be bottleneck. Thus we propose the media stream relay engine architecture and sequence diagram for development relay engine on Android OS.

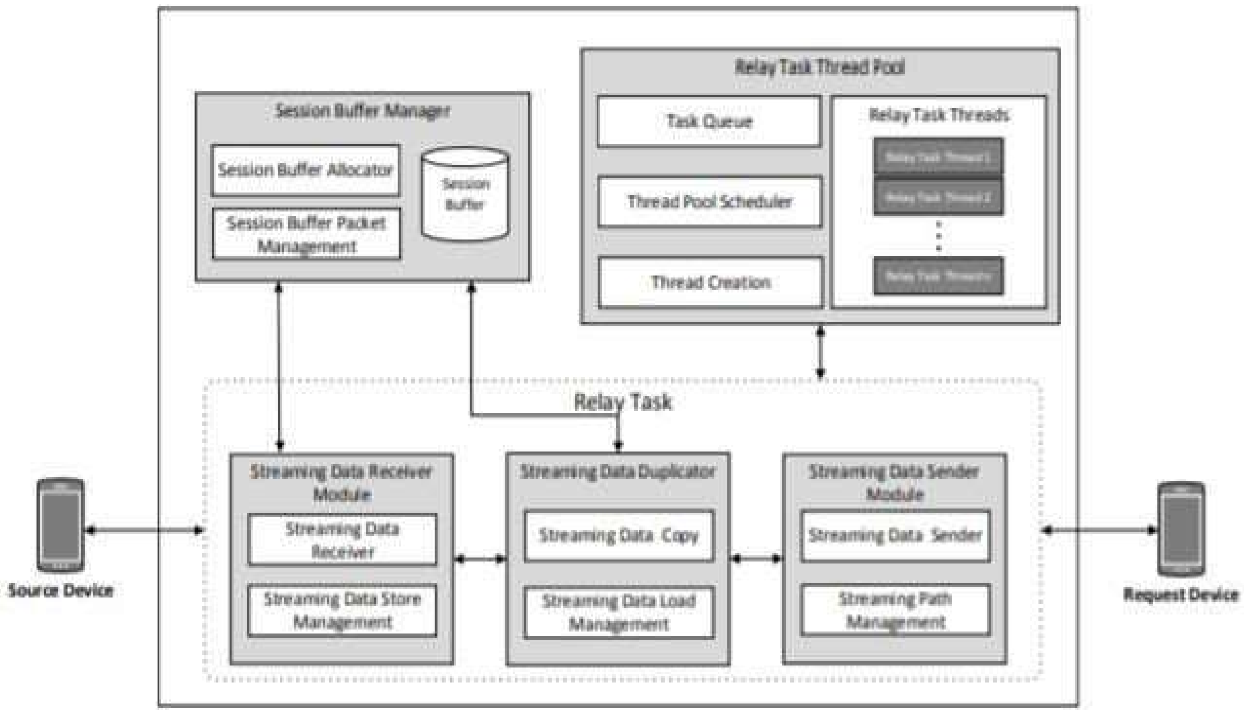

Fig 2.1.2. Media Stream Relay Engine Architecture

- **Mobile Network Configuration for Large-scale Multimedia Delivery on a Single WLAN**

As smart devices become disseminated and more popular, N screen technology that provides content to multiple screens is spreading. However, the issues of traffic limits that arise when using multiple devices simultaneously has not yet been resolved. In particular, techniques to increase traffic are insufficient when multimedia streaming is used on multiple devices. Therefore, we need to measure the performance of the wireless network in configurations that can overcome the traffic limitations and can allow for multimedia streaming.

Wireless networks were configured using an infrastructure network, an ad hoc network, and a Wi-Fi Direct network. The performance of these configurations was measured for RTT and bandwidth. The performance of the ad hoc and Wi-Fi Direct networks were similar; both had higher performance than an infrastructure network. However, ad hoc networks are not easy to configure. It is easy, however, to conduct multimedia streaming by configuring the network using Wi-Fi Direct. We looked for a change in performance when using WiFi Direct to connect to multiple devices. We found similar RTT values, but bandwidth was inversely proportional to the number of devices. This result indicates that configuring a network for multimedia streaming to multiple devices is not a smooth process. Furthermore, Wi-Fi Direct restricts the number of connected devices depending on the manufacturer and performance. Thus, the configuration of the network is difficult using multiple devices.

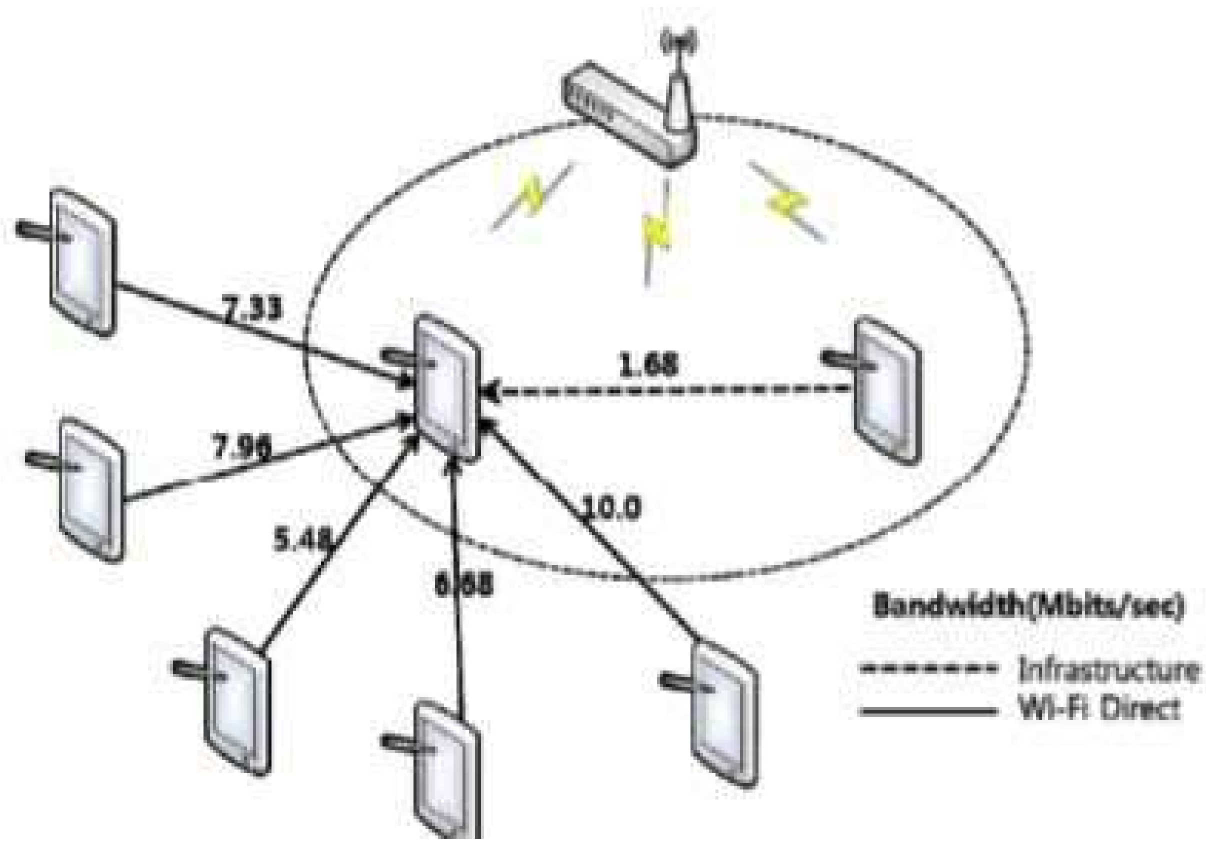

Fig 2.1.3. The performance measurement of a network structure for multimedia streaming: Bandwidth

- **P2P Networking: An Information-Sharing Alternative**

P2P networks differ markedly from the client-server interactions that typify applications in the TCP/IP world. A client-server scenario like the Web depends on a single server storing information and distributing it to clients in response to their equests. The information repository remains essentially static, centralized at the server, and subject only to updates by the provider. Users assume a passive role in that they receive, but do not contribute, information. A P2P network, on the other hand, considers all nodes equal in their capacity for sharing information with other network members. Each user makes an information repository available for distribution, which, combined

with anyone's ability to join the network, leads to the fast growth of a network omposed of distributed information repositories.

In a P2P model, each member node can make information available for distribution and can establish direct connections with any other member node to download information. Instead of looking at what is available in a centralized repository, such as MP3.com,a client seeking information from a P2P network searches across scattered collections stored at numerous member nodes, all of which appear to be a single repository with a single index, such as Napster. Member nodes primarily use existing P2P networks,

such as Gnutella or Napster, to share information of some kind. The TCP/IP Internet typically admits these nodes without restraint, and any node that can access the network can join. Networks can refine admission policies, however, to restrict membership to interest groups. Likewise, the operation domain can be a controlled environment such as an intranet rather than the public Internet.

- **In-Kernel Relay for Scalable One-to-Many Streaming**

The bulk fan-out test shows that OMSS eliminates the CPU bottleneck and reaches link capacity. For TCP and UDP streams, the bulk fan-out test showed 56 and 59 percent improvement, respectively, over Daemon. On the embedded platform, OMSS increased the number of concurrent streams from 45 to 85, compared with Daemon. Moreover, the payload-size test showed the benefit of the payload-sharing mechanism when the payload size is greater than 200 bytes. In the future, we plan to investigate the buffer-management drop policy and compare the performance of relay mechanisms, such as IP-layer hooks and I/O subsystems. It is also necessary to look at QoS when playing multimedia under various buffer-management mechanisms, such as slowest-sink-drop, taildrop, and RED. The payload-sharing mechanism is currently realized by the scatter-gather I/O, but low-end platforms in future implementations might not be equipped with an NIC with the scattergather I/O. To make OMSS portable, we intend to design a payload-sharing mechanism without the scatter-gather I/O.

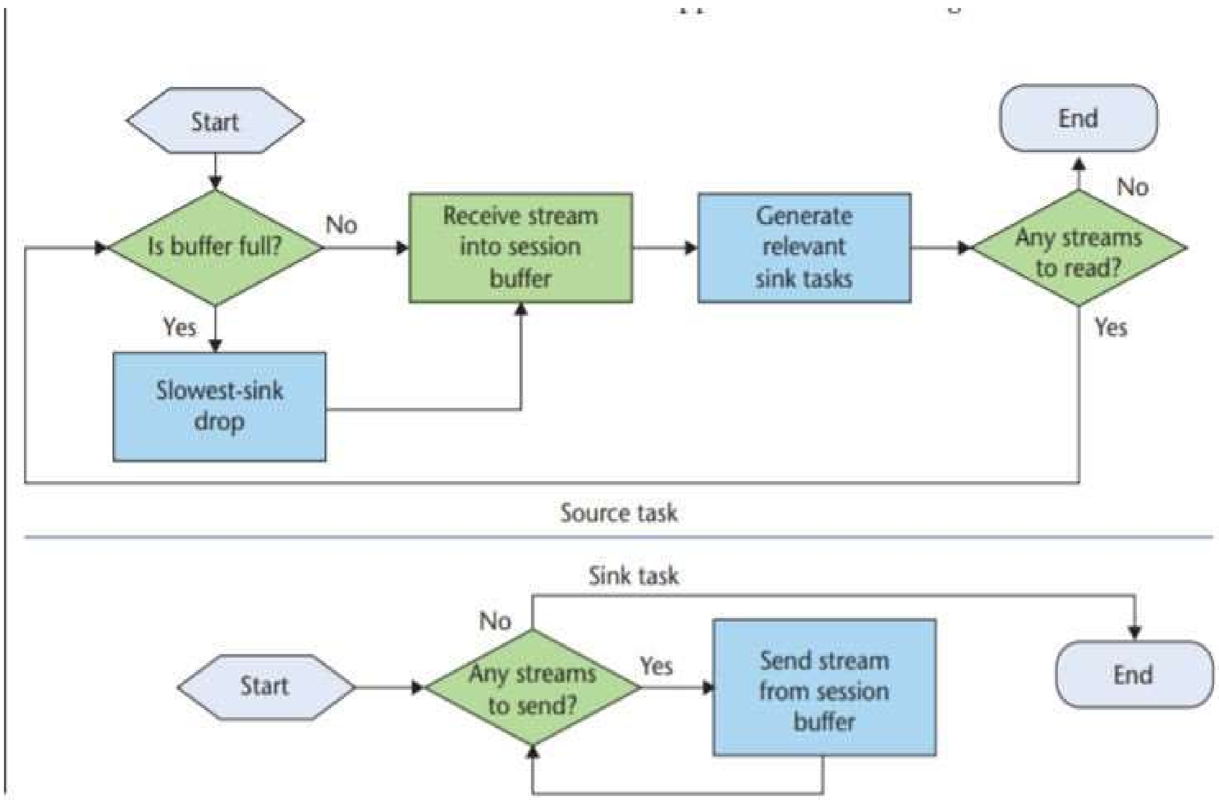

Fig.2.1.4. OMSS flow chart. OMSS provides buffer management for one-to-many streaming over the socket layer and uses the slowest sink-drop policy to reserve buffer space for newly arriving content.

- **A Transport Protocol for Supporting Multimedia Streaming in Mobile Ad Hoc Networks**

The fundamental problem of transport protocol design for multimedia applications in mobile ad hoc networks is that such networks exhibit a rich set of network behaviors, including congestion, channel error, route change and disconnection that must be reliably detected and reacted. Detection of such behaviors is challenging because measurement data is noisy. Existing approaches typically rely on the network-layer notification support at each router. This paper explores an alternative approach that relies solely on end-to-end mechanisms. To robustly detect network states in the presence of measurement noise, we propose a multiple-metric-based joint detection technique. In this technique, a network event is signaled only if all the relevant metrics detect it. The simulations show that ADTFRC is able to significantly reduce the false detection probability while keeping the incompatible detection errors low, thus greatly improve the transport performance in a TCP friendly way. This demonstrates that the end-to-end approach for multimedia transport is also viable for ad hoc networks. In a broader context, the robustness of multimedia transport, as well as TCP, has not been equally well explored in the current research. Ad hoc networks offer a good example to demonstrate its importance. The issue of robust detection of network states in ad hoc networks in the presence of transient dynamics and measurement noise deserves further attention. In fact, this issue is not only valid in the context of our end-to-end approach, it also holds for network-oriented design. Each router still suffers from imperfect monitoring and observation noise and, thus, may wrongly send out notification signals. Our multimeric technique is in principle also applicable in this context. This will be our future work.

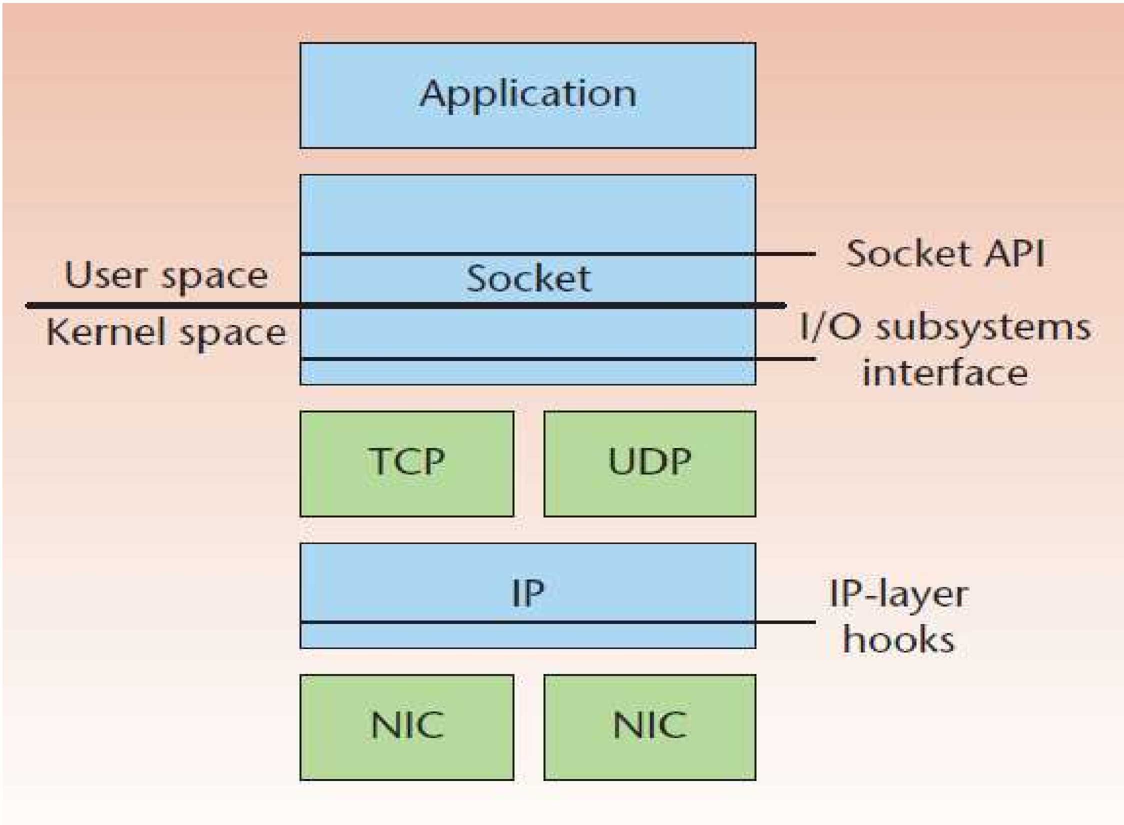

Fig2.1.5. Three categories of relay data paths.

- **Collaborative Streaming-based Media Content Sharing in WiFi-enabled Home Networks**

In this paper, we introduce DOMS that coordinates the data transfers between WiFi-equipped devices for media content sharing within home networks. Specifically, DOMS realizes flexible configurations that assume each WiFi-equipped home networking device as decentralized data storage. For efficient data transfers, DOMS exploits the temporal direct link setup between distributed WiFi devices via iDLS. The centralized scheduling of the collaborative data sharing significantly improves the achievable overall system throughput performance. We summarize this paper by reviewing the prototype-based performance evaluation results.

Review Experimental Evaluation Results All results from experiments are obtained from the prototype-based evaluations at practical configurations to tailor the Wi-Fi-based home networks. We randomly select the location of Wi-Fi devices spread over 20-by-20 meters to cater the randomnature of consumer electronics placement in home network configurations. The hardware configurations of the prototype implementation, embedded computing machine with 1GHz single-core CPU speed and 512 Mbyte RAM, is intended to take the capability of modern digital appliance and handheld devices into account. Under the on-demand video streaming scenario, the performance and scalability of prototype implementation are evaluated. The configured scenario reflects such environments where a partially non-collaborative Wi-Fi devices are existing. The results indicate that the scalability in terms of the number of concurrent sessions achieving QoS requirements is improved by 200% when compared with conventional noncollaborative (i.e., server-client) streaming. The computational load, which is one of the key concerns in the system scalability, is also evaluated. Our measurement shows that centralized scheduling only produces less than 10% of CPU usage under the scenario includes 8 WiFi devices (4 collaborative and 4 non-collaborative) as clients.

Fig 2.1.6. Various types of media content sharing amongst WIFI-Direct enabled devices in a home network.

- **A Case for End System Multicast**

The conventional wisdom has been that IP is the natural protocol layer for implementing ulticast related functionality. However, more than a decade after its initial proposal, IP Multicast is still plagued with concerns pertaining to scalability, network management, deployment and support for higher layer functionality such as error, flow and congestion control. In this paper, we Xplore an alternative architecture that we term End System Multicast, where end systems implement all multicast related functionality including membership management and packet replication. This shifting of multicast support from routers to end systems has the potential to address most problems associated with IP Multicast. However, the key concern is the performance penalty associated with such a model. In particular, End System Multicast introduces duplicate packets on physical links and incurs larger end-to-end delays than IP Multicast. In this paper, we study these performance concerns in the context of the Narada protocol. In Narada, end systems selforganize into an overlay structure using a fully distributed protocol. Further, end systems attempt to optimize the efficiency of the overlay by adapting to network dynamics and by considering application level performance. We present details of Narada and evaluate it using both simulation and Internet experiments. Our results indicate that the performance penalties are low both from the application and the network perspectives. We believe the potential benefits of transferring multicast functionality from end systems to routers significantly outweigh the performance penalty incurred.

- **A Distributed Search Service for Peer-to-Peer File Sharing in Mobile Applications**

The concept of Passive Distributed Indexing, a general-purpose distributed document search service for mobile file sharing applications. PDI is based on peer-to-peer technology, i.e., PDI does not require any centralized infrastructure for providing searching capabilities. Building blocks of PDI constitute local broadcast transmissions of query- and response-messages together with caching results of popular queries at every device participating in PDI. Using these building blocks, PDI eliminates the need of flooding the whole network with query messages. Furthermore, it can be configured to support different system environments and application requirements by adjusting three parameters, namely the size of the local index caches, the maximum number of hops that a message will be forwarded, and the document timeout used for expiry of cache entries. In extensive simulation studies, we demonstrated the usefulness of PDI for different systems and applications. We conclude from this performance studies, that for systems with a high density of mobile devices and applications bearing low locality in the query stream, sufficient index cache sizes should be provided. In systems with a medium density of mobile devices and medium wireless transmission ranges, 2-hop packet forwarding should be enabled, whereas packet forwarding should be disabled if either the number of mobile devices or their transmission range is high. A large number of documents contributed to PDI by each mobile device can be handled by providing sufficient index size. Finally, PDI provides an initial filling of index caches in very short time, eliminating the need of sophisticated warm-up mechanisms.

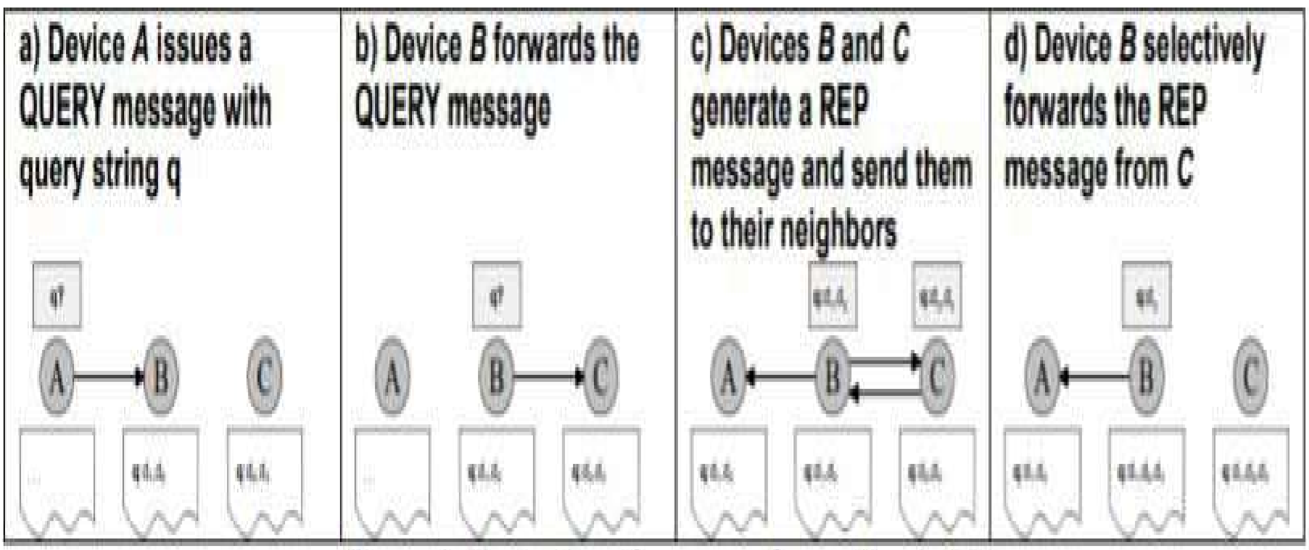

Fig2.1.7. Illustration of message forwarding in PDI

- **A Special-Purpose Peer-to-Peer File Sharing System for Mobile Ad Hoc Networks**

In this paper, we presented the Optimized Routing Independent Overlay Network (ORION), a special-purpose approach for P2P file sharing tailored to MANET. ORION is completely implemented on the application layer and does not depend on support of a MANET routing protocol.

As building blocks, ORION comprises of an algorithm for construction and maintenance of an application-layer overlay network that enables routing of all types of messages required to operate a P2P file sharing system, i.e., queries, responses, and file transmissions. It combines application-layer query processing and overlay network construction with the network layer process of route discovery and transparently aggregates redundant transfer routes on a per-file basis. The ORION transfer protocol enables efficient file transfers on top of the overlay connections established by the search algorithm. In a detailed ns-2 simulation study, we illustrated the performance gains of ORION compared to an off-the-shelf approach based on a P2P file sharing system for the wireline Internet, TCP and a MANET routing protocol. We found that ORION significantly increases search accuracy and reduces overhead for searching. Furthermore, ORION enables more reliable file transfers with lower overhead compared to the offthe-shelf approach.

- **On-demand Video Streaming in Mobile Opportunistic Networks**

Providing on-demand video streaming services to large numbers of nodes in mobile wireless networks is a challenging task. We propose MOVi (Mobile Opportunistic Video-on-demand), a mobile peer-to-peer video-on-demand application based on ubiquitous WiFi enabled devices such as smartphones and Ultra Mobile PCs. MOVi poses new challenges in on-demand video streaming caused by limited wireless communication range, user mobility and variable user population density. MOVi addresses these challenges by exploiting the opportunistic mix use of downlink and direct peer-to-peer communication to significantly improve achievable overall system throughput. Through simulation and actual implementation, we observed that MOVi is able to increase the number of supported concurrent users two fold compared with unicast based on-demand video streaming, as well as reducing video start-up delay by half. Significantly, these performance improvements can be achieved without increasing power consumption at the end device. Using MOVi, users not only share and distribute on-demand video such as YouTube clips for their own performance gain but also for the benefit of the collaborative group.

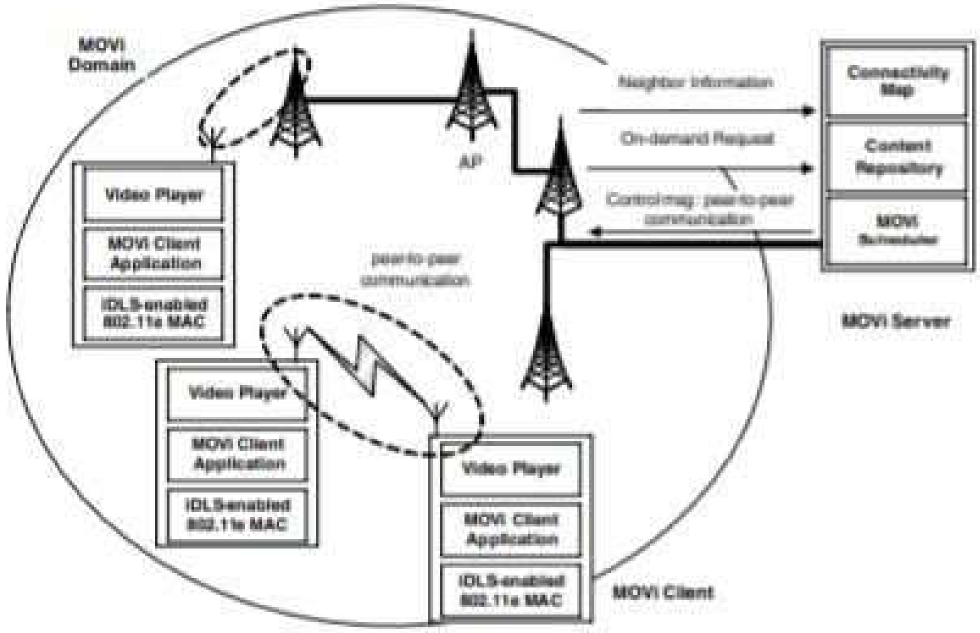

Fig 2.1.8. MOVi Architectural Components.

- **An Efficient Implementation of File Sharing∗ Systems on the Basis of WiMAX and Wi-Fi**

In this paper they have proposed a layered P2P file sharing system on the important WiMAX and Wi-Fi wireless techniques. Requesting end host obtains the requested file from three types of positions: it first searches neighbor end hosts under the same SS; if this step does not find the requested file, then the requesting end host sends lookup messages within the nearest SSs mesh network using CDSR protocol; if all the above steps fail, the requesting end host launches a lookup process on Chord based BSs ring overlay network. By statistical analysis and practical simulations, we have shown that our layered P2P file sharing system can greatly reduces the number of lookup messages in physical networks and prevents over expenses of precious bandwidth in wireless metropolitan area networks. Our future work aims at implementing the system with consideration of the mobility of wireless end hosts and also combining wireless P2P file sharing architecture along with more powerful and stable cable network P2P applications.

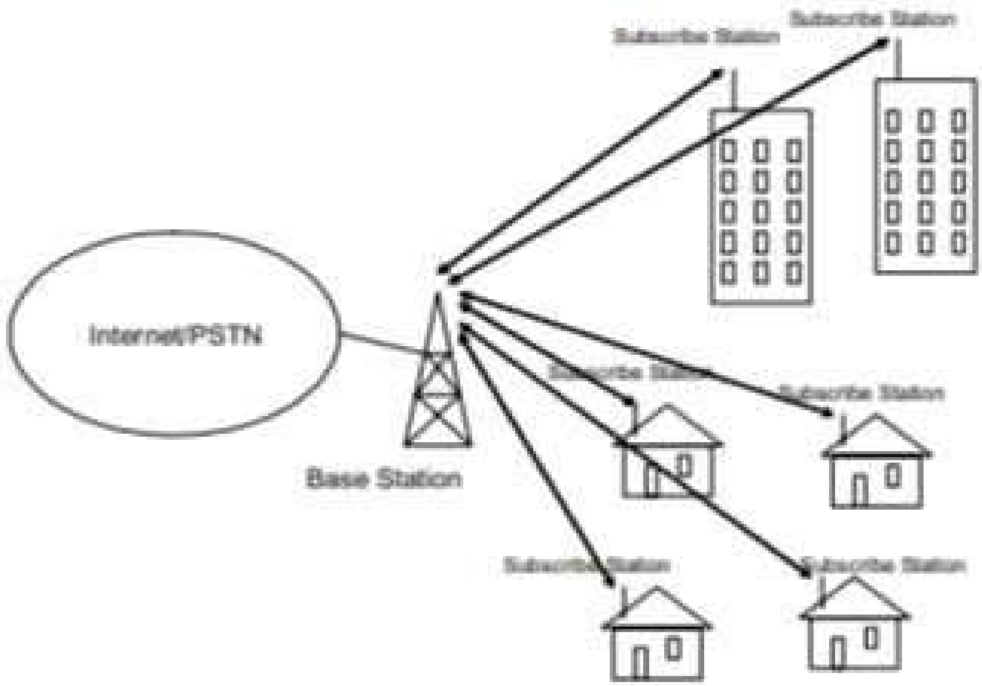

Fig 2.1.9. WIMAX structure

- **A Data-driven Overlay Network for Peer-to-Peer Live Media Streaming**

We report performance measurements and analyses of a variety of wireless networks to arrive at an appropriate network structure configuration for smooth multimedia streaming service with a variety of smart devices in a single wireless network environment. Unlike the usual infrastructure network, configurations such as the IEEE 802.11 ad hoc network and the Wi-Fi direct network use direct connections between devices without going through a wireless AP. Therefore, these configurations prevent concentrating traffic at the wireless AP. We generated three types of wireless network performance measurements and arrived at a suitable network for multimedia streaming service using a variety of smart devices. The wireless network was configured in a

structure for efficient multimedia streaming service in a single wireless LAN environment, and subsequently, the network performance was measured.

As smart devices become disseminated and more popular, N screen technology that provides content to multiple screens is spreading. However, the issues of traffic limits that arise when using multiple devices simultaneously has not yet been resolved. In particular, techniques to increase traffic are insufficient when multimedia streaming is used on multiple devices. Therefore, we need to measure the performance of the wireless network in configurations that can overcome the traffic limitations and can allow for multimedia streaming.

Wireless networks were configured using an infrastructure network, an ad hoc network, and a Wi-Fi Direct network. The performance of these configurations was measured for RTT and bandwidth. The performance of the ad hoc and Wi-Fi Direct networks were similar; both had higher performance than an infrastructure network. However, ad hoc networks are not easy to configure. It is easy, however, to conduct multimedia streaming by configuring the network using Wi-Fi Direct.

They looked for a change in performance when using Wi- Fi Direct to connect to multiple devices. We found similar RTT values, but bandwidth was inversely proportional to the number of devices. This result indicates that configuring a network for multimedia streaming to multiple devices is not a smooth process. Furthermore, Wi-Fi Direct restricts the number of connected devices depending on the manufacturer and performance. Thus, the configuration of the network is difficult using multiple devices.

- **An Efficient Peer-to-Peer File Sharing Exploiting Hierarchy and Asymmetry**

This paper presented the design and preliminary evaluation of PASS, a novel peer-to-peer distributed file sharing system. Our system provides a self-organized, adaptive, and scalable distributed file searching and routing. PASS can publish and search a file with the constant routing hops (at most four in the current implementation). PASS divides the network into autonomous areas so that the file directory can be distributed and replicated over the areas. The routing information is also replicated for a fast local area search. Our preliminary evaluation with simulations demonstrates that PASS can provide the efficient search performance after a system warming period. PASS has been implemented in Java with SDK1.4.0. It uses RMI and the asynchronous message passing technology. We are currently improving PASS to provide fast failure detection and recovery mechanisms for supernodes. We are investigating to find the proper size for the condensed key. Efficient disk swapping is necessary in case of large table management. Currently, PASS can search only the files, which contain the exactly matching file key. We are investigating on extending PASS to provide the keyword-based searching.

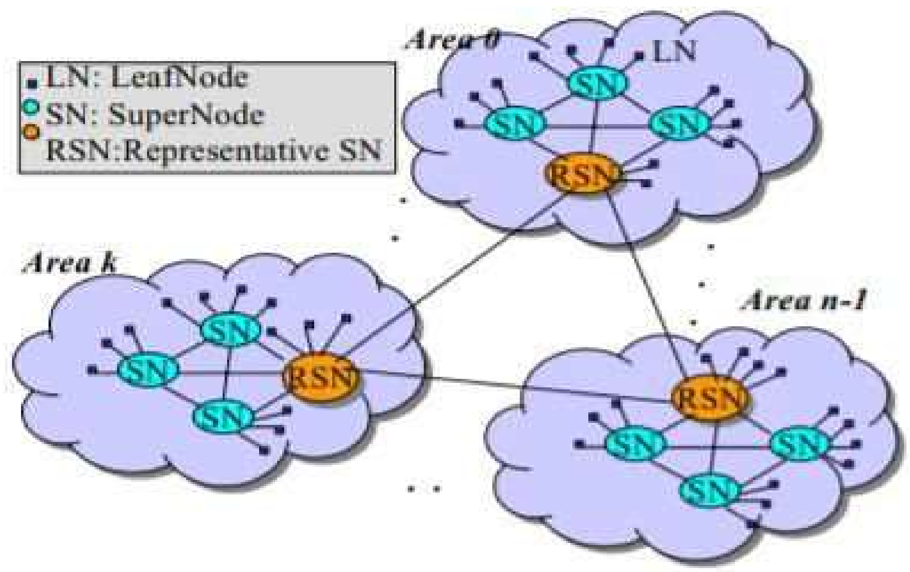

Fig2.1.10. PASS Architecture

- **File Sharing System in Wireless Home Environment**

In this paper they have a solution that makes family members to share files whenever and wherever they want. The whole system is divided into two parts. The first part is the family member with mobile phone outside the home environment sharing files with the family members with PC inside the home environment. The second part is the family member with mobile phone back at home sharing files with other family members with PC in the home environment. We assume two situations in first part, PC online and PC offline. We use Sharing Server in IMS network to control and manage the system over SIP. When the PC is offline, we use Storage Server to temporarily store and forward the files later on. In the second part of the system, we use WiFi and FTP to share files effectively and for free. This paper just proposes the architecture and the SIP flow. There are still many details for further discussion. Further study can be done in real-time video stream sharing and the design of the client software.

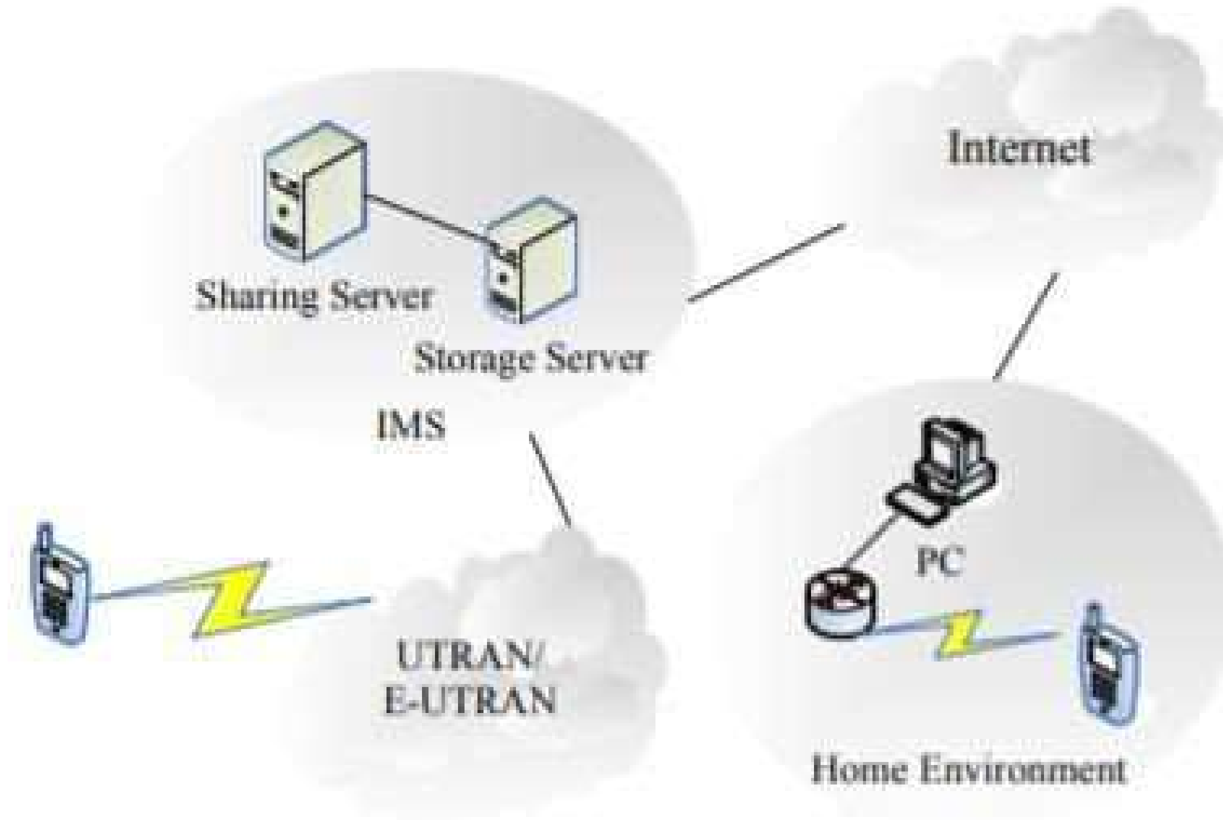

Fig 2.1.11. Architecture of the system

- **A Social-Network-Aided Efficient Peer-to-Peer Live Streaming System**

In this paper, they have proposed SAVE, a social-network-aided efficient P2P live streaming system. SAVE supports successive and multiple-channel viewing with low switch delay and low server overhead by enhancing the operations of joining and switching channels. SAVE considers the historical channel switching activities as the social relationships among channels and clusters the frequently interacted channels together by merging overlays or building bridges between the overlays. It maximizes the probability that existing users can locate their desired channels within its channel cluster and can take the bridges for channel switches. In addition, each node has a friendlist that records nodes with similar watching patterns, which is used to join a new channel overlay. SAVE also has the channel-closeness-based chunk pushing strategy and capacity-based chunk provider selection strategy to enhance its system performance. Our survey on user video streaming watching activities confirms the necessity and feasibility of SAVE. Through the experiments on the PeerSim simulator and PlanetLab testbeds, we prove that SAVE outperforms other representative systems in terms of overhead, video streaming efficiency and server load reduction, and the effectiveness of SAVE's two strategies. Our future work lies in further reducing the cost of SAVE in structure maintenance and node communication. Also, we will design algorithms for cluster separation and decentralized cluster head election.

- **Video and Audio Streaming Issues in Multimedia Application**

In this paper, different video streaming conception and examinations over the video bit rate has been analyzed. The main focus of this paper is to enhance the standard of video and transmission ability and improve the quality of video before refetching for mobile users. A more improved version of video streaming is required to upgrade the quality of video. Thus lot of future work can still be done in this field. Different applications are being used to compare the different methods of streaming and with help of the obtained result new algorithm can be designed to improve the quality and reduce the bandwidth and storage space.

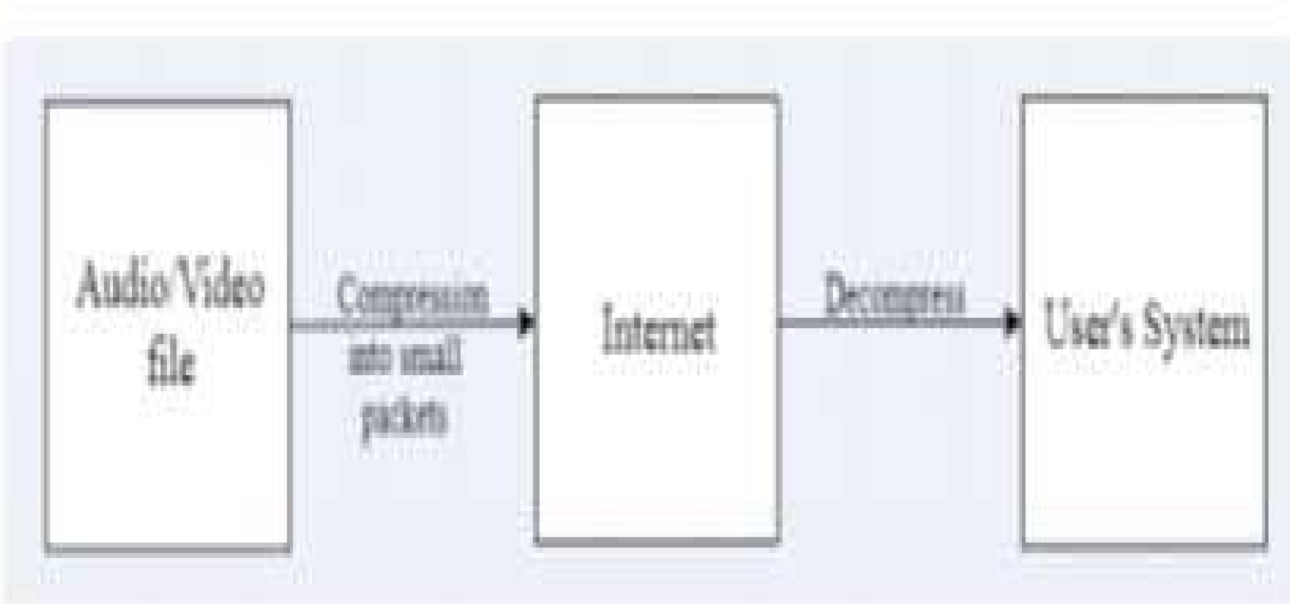

Fig2.1.12. File over the network

- **A Survey on Various File Sharing Methods in P2P Networks**

Recently, the P2P file sharing systems are more popular in the different applications. In Peer- toPeer file sharing system, Applications are organized via different peer at dynamically. Here the communication between the peers will be taking some risk because of no previous knowledge and the experiences. The various Trust based model is also discussed in the development of Peer-to-Peer system and calculating the risks that are used for efficient P2P file sharing systems. This survey paper discovers the different file sharing methods and important concerns that communicate to peers systems and discussing about the various study actions of Peer-to-Peer file systems.

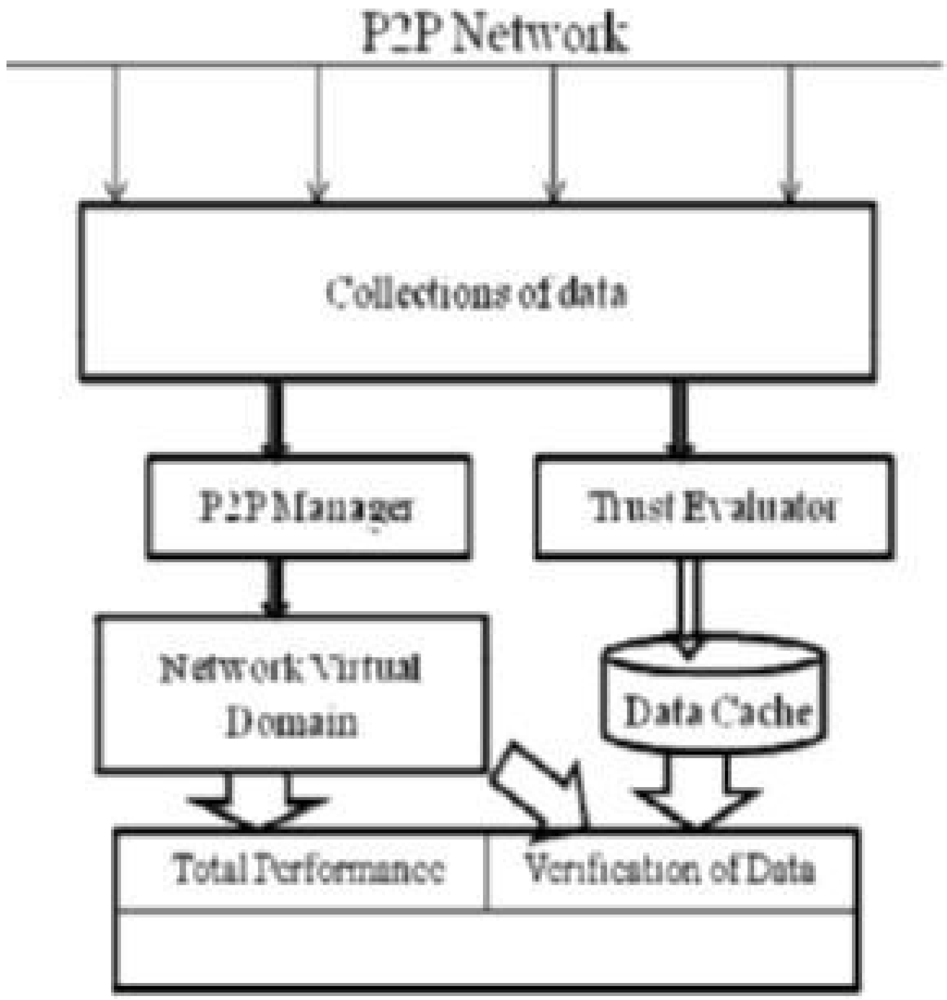

Fig 2.1.13. Structure of Trust and Reputation Based Model

- **A Study of Live Video Streaming System for Mobile Devices**

Multimedia application on mobile phone platform is a hot topic in Information Technology industry accompanying with the advances of technologies in Internet, smart operating system and wireless network. The development of streaming media technology boosts the audio and video application over the Internet. This paper describes a mobile video live streaming system with Wi-Fi/4G mobile phone to capture video and disseminate. Availability of mobile video live streaming system is given and then a mobile video live streaming system based on streaming media technology is designed and realized. The results indicates that this system implements the function of video capturing in terminal, decoding and encoding raw data, media playing, real-time streaming media transmission and so on. It is very practical and portable in the actual operating environment

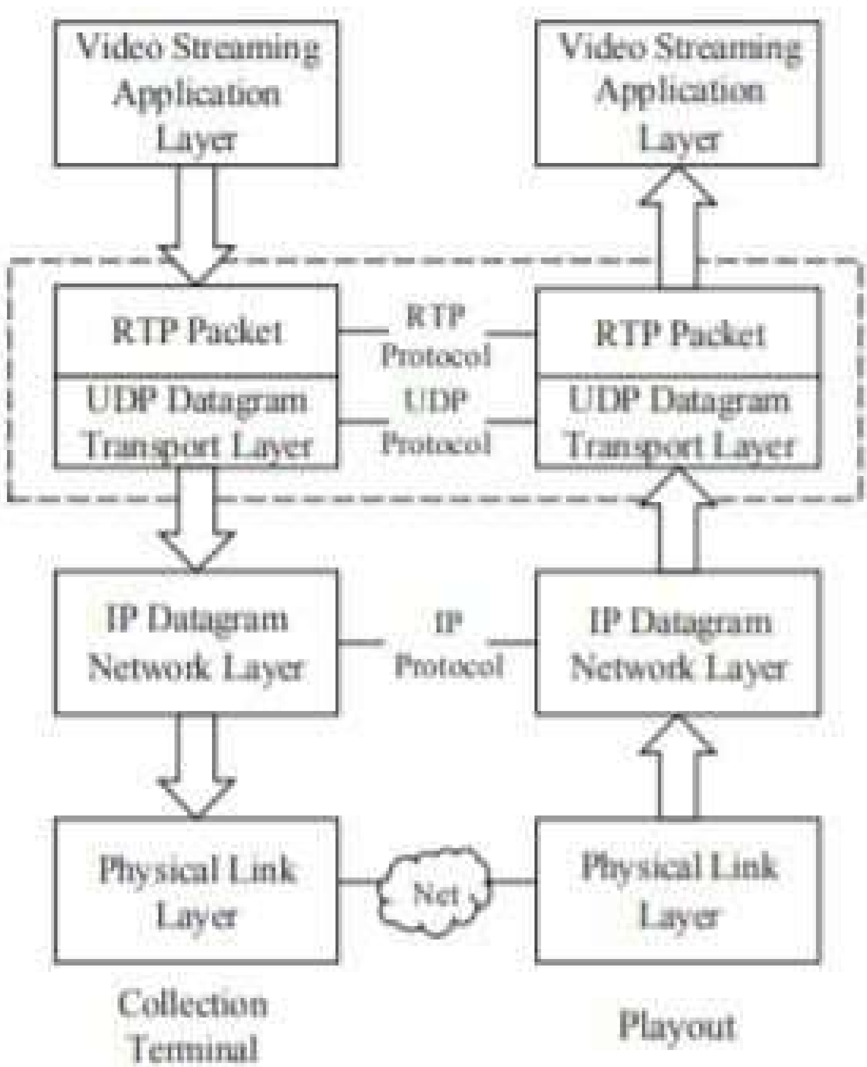

Fig 2.1.14. Structure of file streaming

- **Cool-SHARE: Offload Smartphone Data by Sharing**

With the volume of mobile data traffic almost doubling each year, the mobile data offloading challenge has become ever relevant. This paper takes on a practical approach to offload the cellular network. A solution called Cool-SHARE is proposed and implemented on Android. It is an app for seamlessly sharing apps over short-range links with limited cellular control and can be extended for multimedia data sharing. Two types of sharing scenarios are defined, i.e., social sharing and opportunistic sharing. The challenges of sharing are identified and analysed. The discovery, connection, and data transfer delays are measured and analysed on a Samsung Galaxy S III using Bluetooth and Wi-Fi Direct. The challenges including seamless discovery and connection, authenticity and payment are solved. Furthermore, suggestions for reducing delays of the low-level Wi-Fi Direct implementation are proposed

In this paper we propose a practical solution to tackle the challenge in mobile data offloading by exploiting short-range communication technologies to offload smartphone apps downloading which can be extended to multimedia data sharing. We have successfully implemented a prototype app, Cool-SHARE, capable of doing social sharing and opportunistic sharing. This solution can also be usable in areas with expensive 3G data plans or poor cellular network coverage. Measurements were carried out on Samsung Galaxy S III Android devices to analyse discovery delays, connection delays, and throughput of Bluetooth and Wi-Fi Direct. The discovery and connection delays of Wi-Fi Direct were measured for both autonomous mode and standard mode. The results were analysed and suggestions were made on how to reduce these delays in the observed implementation.

Fig2.1.15. Illustration of cool share

- **Streaming Video over the Internet: Approaches and Directions**

Due to the explosive growth of the Internet and increasing demand for multimedia information on the web, streaming video over the Internet has received tremendous attention from academia and industry. Transmission of real-time video typically has bandwidth, delay, and loss requirements. However, the current best-effort Internet does not offer any quality of service (QoS) guarantees to streaming video. Furthermore, for video multicast, it is difficult to achieve both efficiency and flexibility. Thus, Internet streaming video poses many challenges. To address these challenges, extensive research has been conducted. This special issue is aimed at dissemination of the contributions in the field of streaming video over the Internet. To introduce this special issue with the necessary background and provide an integral view on this field, we cover six key areas of streaming video. Specifically, we cover video compression, application-layer QoS control, continuous media distribution services, streaming servers, media synchronization mechanisms, and protocols for streaming media. For each area, we address the particular issues and review major approaches and mechanisms. We also discuss the tradeoffs of the approaches and point out future research directions.

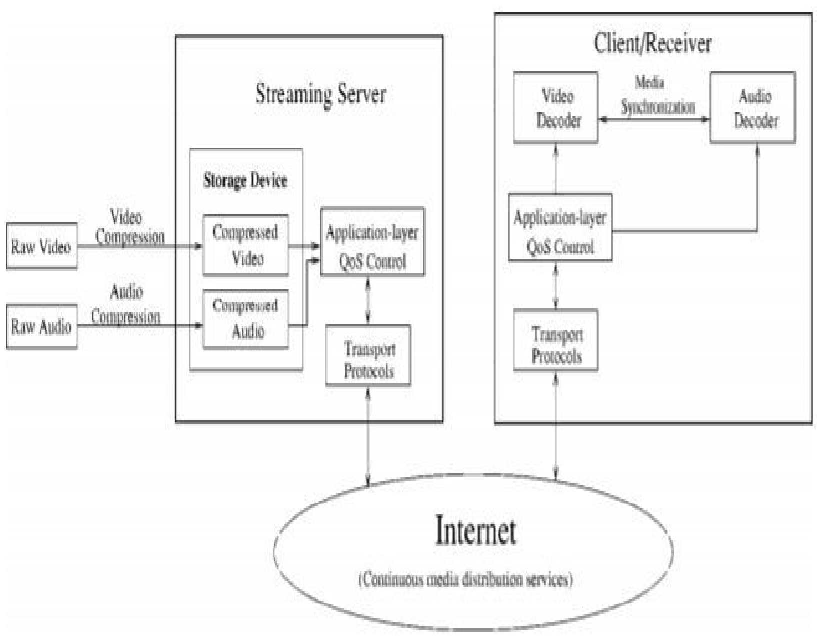

Fig2.1.16. Architecture for video streaming

2.2 Comparison of existing papers

Sl. No	Title	Authors	Year of publication	Type of data	Methodology	Limitations
1	Robust Super-Peer-Based P2P File-Sharing Systems	Jenn-Wei Lin and Ming-Feng Yang	2010	Research Paper	An efficient approach for improving file availability in super-peer-based peer-to-peer (P2P) file-sharing systems. In the super-peer-based P2P file-sharing system, peers are organized into multiple groups	Can be used sadvanced protocols for effective transmission
2	Enhanced UPnP QoS Architecture for Network-adaptive Streaming Service in Home Networks	HyunYong Lee, SungTae Moon and JongWon Kim	2007	Research Paper	an enhanced UPnP QoS architecture is proposed to support network-adaptive media streaming in home networks	Interruption in network provides loss of data
3	Design of a Media Stream Relay Engine on the Android OS	Huigwang Je, Dongwoo Kwon and Hongtaek Ju	2015	Research Paper	Proposed as a solution to the problem of decreased network performance for large-scale multimedia delivery in a single wireless LAN environment.	It requires the administrator access which causes a vulnerability in the security
4	Mobile Network Configuration for Large-scale Multimedia Delivery on a Single WLAN	Huigwang Je, Dongwoo Kwon, Hyeonwoo Kim, and Hongtaek Ju	20114	Research Paper	The wireless network was configured in a structure for efficient multimedia streaming service in a single wireless LAN environment, and subsequently, the network performance was measured.	If the number of connected devices increases then data transfer rate decreases.
5	P2P Networking: An Information-Sharing	Manoj Parameswaran,	2001	Research Paper	Uses client-server technologies that incorporate	If all the clients simultaneously

	Alternative	Anjana Susarla Andrew, B. Whinston			networking as an ancillary, value-added feature. Peer-to-peer computing that offers a radically new way of isolating and focusing on the networking aspect as the business model's mainstay.	request data from the server, it may get overloaded. This may lead to congestion in the network
6	In-Kernel Relay for Scalable One-to-Many Streaming	Ying-Dar Lin, Chia-Yu Ku, Yuan-Cheng Lai, Chia-Fon Hung	2013	Research Paper	The in-kernel One-to-Many Streaming Splicing (OMSS) relay method can help improve the relay data paths of critical nodes to reduce computing power for UDP and TCP streams and enhance the subscriber capacity.	Uses OMSS method in which manipulation of data can be done easily
7	A Transport Protocol for Supporting Multimedia Streaming in Mobile Ad Hoc Networks	Zhenghua Fu, Xiaoqiao Meng, and Songwu Lu	2013	Research Paper	The design and implementation of a transmission control protocol (TCP)-friendly transport protocol for ad hoc networks. Our key design novelty is to perform multimetric joint identification for packet and connection behaviors based on end-to-end measurements	TCP protocol provides reliable data delivery support but high latency problems arises
8	Collaborative Streaming-based Media Content Sharing in WiFi-enabled Home Networks	Hayoung Yoon, JongWon Kim	2010	Research Paper	Proposed DOMS (Decentralized collaborative Media content Streaming) that realizes flexible media content sharing by exploiting	DOMS requires a stable connection and more processing

					collaborative segment-based streaming amongst WiFi devices via the temporarily-established direct links	
9	A Case for End System Multicast	Yang-hua Chu, Sanjay G. Rao, Srinivasan Seshan and Hui Zhang	2002	Research Paper	Explored an alternative architecture that we term End System Multicast, where end systems implement all multicast related functionality including membership management and packet replication. This shifting of multicast support from routers to end systems has the potential to address most problems associated with IP Multicast.	IP Multicast is the first significant feature that has been added to the IP layer since its original design and most routers today implement IP Multicast. Despite this, IP Multicast has several drawbacks that have so far prevented the service from being widely deployed.
10	A Distributed Search Service for Peer-to-Peer File Sharing in Mobile Applications	Christoph Lindemann and Oliver P. Waldhorst	2002	Research Paper	Enabling resource-effective searching for files distributed across mobile devices based on simple queries	It searches all the data in the device which makes the privacy effected
11	A Special-Purpose Peer-to-Peer File Sharing System for Mobile Ad Hoc Networks	Alexander Klemm, Christoph Lindemann, and Oliver P. Wald horst	2003	Research Paper	a special purpose system for searching and file transfer tailored to both the characteristics of MANET and the requirements of peer-to-peer file sharing	The researchers proposed the use of this structure for audio; it is not suitable for high-capacity media such as streaming video

12	On-demand Video Streaming in Mobile Opportunistic Networks	Hayoung Yoon, Jong Won Kim, Feiselia Tan, and Robert Hsieh	2008	Research Paper	Proposed MOVi (Mobile Opportunistic Video-on-demand), a mobile peer-to-peer video-on-demand application based on ubiquitous WIFI enabled devices such as smartphones and Ultra Mobile PCs	It exploits sparsely distributed access points, user mobility , fluctuating to provide a high bitrate on-demand video streaming service.
13	An Efficient Implementation of File Sharing∗ Systems on the Basis of WiMAX and Wi-Fi	Jingyuan Li, Liusheng Huang, Weijia Jia, Mingjun Xiao and Peng Du	2006	Research Paper	Proposed an efficient algorithm for P2P file sharing systems based on WiMAX mesh mode and Wi-Fi technologies	WiMAX is very power intensive technology and requires strong electrical support
14	A Data-driven Overlay Network for Peer-to-Peer Live Media Streaming	Xinyan Zhang, Jiangchuan Liut, Bo Lis, and Tak-Shng Peter Yum	2005	Research Paper	a Data-driven Overlay Network for live media streaming. The core operations in DOTNet are very simple: every node periodically exchanges data availability information with a set of partners, and retrieves unavailable data from one or more partners, or supplies available data to partners.	Can be used low-latency protocols for data transfer instead of high latency protocols
15	An Efficient Peer-to-Peer File Sharing Exploiting Hierarchy and Asymmetry	Gisik Kwon Kyung D. Ryu	2015	Research Paper	Peer-to-peer Asymmetric file Sharing System (PASS), a novel approach to P2P file sharing, which accounts for the different capabilities and network locations of the	Some peer-to-peer (P2P) file sharing operation models over asymmetric networks have several

					participating machines.	shortcomings that may affect system and network performance
16	File Sharing System in Wireless Home Environment	Yongjia Liu, Yong Sun, Qing Liao, Xiangming Wen	2010	Research Paper	a system aiming to enable family member with mobile phone to share files with other family members with a PC at home no matter whether the mobile phone user is outside the home environment or inside the home environment	Uses a protocol similar to DLNA, which trades- of low latency with quality
17	A Social-Network-Aided Efficient Peer-to-Peer Live Streaming System	Haiying Shen, Yuhua Lin, and Jin Li	2014	Research Paper	Proposed a Social-network-Aided efficient live streaming system (SAVE). SAVE regards users' channel switching or multichannel watching as interactions between channels	Cannot be uses where internet services are poor
18	Video and Audio Streaming Issues in Multimedia Application	Puja Smiti, Swapnita Srivastava	2018	Research Paper	Different streaming technique has been analyzed to procure the applicability and optimization of audio/video has been done through FFmpeg software.	FFmpeg does not ensure bitrate consistency across a set of files
19	A Survey on Various File Sharing Methods in P2P Networks	Vimal S, Srivatsa S K	2017	Research Paper	discovers the different file sharing methods and important concerns that communicate to peers systems and discussing about the various study actions of Peer-to-Peer file systems	DLNA protocol is used, which trades- of low latency with quality
20	A Study of Live Video	Jiushuang Wang,	2016	Research	describes a mobile video live	Requires high

	Streaming System for Mobile Devices	Weizhang Xu, Jian Wang		Paper	streaming system with Wi-Fi/4G mobile phone to capture video and disseminate.	speed internet services

2.3 Summary of Literature Survey

- Most of the reference papers speaks about the design, configuration and development of the streaming application. Commonly used medium is WI-LAN, commonly used method is DOMS (Decentralized collaborative media content streaming) and MOVi (Mobile opportunistic video-on-demand).
- Few papers speak about the effective file management and storage.
- The efficient algorithm for P2P file sharing system is based on WiMAX mesh mode.
- Discovery of different methods of file sharing like cool-share.

2.4 Limitations of current work

- Most of the applications available today are delivering the content through a medium called internet which may not be available for every device.
- Most of the devices may not be having enough storage to store the multimedia in their devices.
- For streaming, most of the users uses screen mirroring which makes both the devices involved.
- For connection between devices there is no efficient method such as QR code.
- Services are often expensive.

Chapter – 3

3. Objective and Methodology

3.1 Objective

- To implement an android application which helps in steaming and sharing of data between the devices connected through a local network.
- To implement seamless data streaming and sharing between the android devices connected through the local network.
- To implement the application in devices having Android versions 8 and above.
- To implement the connection between the host and the client using the feature WIFI direct.
- To implement HTTP Server that is established in the host device for steaming the multimedia files.

3.2 Algorithm used

- **HTTP Server**

The term *web server* can refer to hardware or software, or both of them working together.

1. On the hardware side, a web server is a computer that stores web server software and a website's component files. (for example, HTML documents, images, CSS stylesheets, and JavaScript files) A web server connects to the Internet and supports physical data interchange with other devices connected to the web.
2. On the software side, a web server includes several parts that control how web users access hosted files. At a minimum, this is an *HTTP server*. An HTTP server is software that understands URLs (web addresses) and HTTP (the protocol your browser uses to view webpages). An HTTP server can be accessed through the domain names of the websites it stores, and it delivers the content of these hosted websites to the end user's device.

At the most basic level, whenever a browser needs a file that is hosted on a web server, the browser requests the file via HTTP. When the request reaches the correct (hardware) web server, the (software) *HTTP server* accepts the request, finds the requested document, and sends it back to the browser, also through HTTP. (If the server doesn't find the requested document, it returns a 404 response instead.)

To publish a website, you need either a static or a dynamic web server.

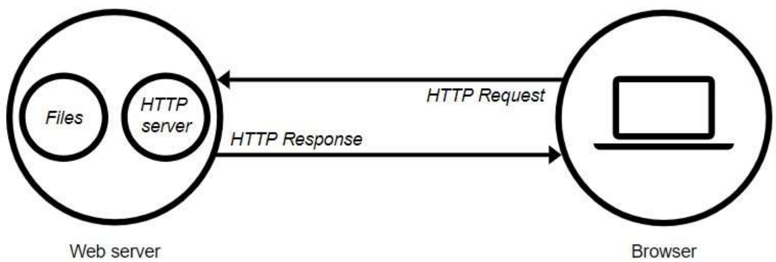

Fig.3.2.1 HTTP Server architecture

A **static web server**, or stack, consists of a computer (hardware) with an HTTP server (software). We call it "static" because the server sends its hosted files as-is to your browser.

A **dynamic web server** consists of a static web server plus extra software, most commonly an *application server* and a *database*. We call it "dynamic" because the application server updates the hosted files before sending content to your browser via the HTTP server.

For example, to produce the final webpages you see in the browser, the application server might fill an HTML template with content from a database. Sites like MDN or Wikipedia have thousands of webpages. Typically, these kinds of sites are composed of only a few HTML templates and a giant database, rather than thousands of static HTML documents. This setup makes it easier to maintain and deliver the content.

- **Streaming Protocol**

Each time you watch a live stream or video on demand, streaming protocols are used to deliver data over the internet. These can sit in the application, presentation, and session layers.

Online video delivery uses both streaming protocols and HTTP-based protocols. Streaming protocols like Real-Time Messaging Protocol (RTMP) enable speedy video delivery using dedicated streaming servers, whereas HTTP-based protocols rely on regular web servers to optimize the viewing experience and quickly scale. Finally, a handful of emerging HTTP-based technologies like the Common Media Application Format (CMAF) and Apple's Low-Latency HLS seek to deliver the best of both options to support low-latency streaming at scale.

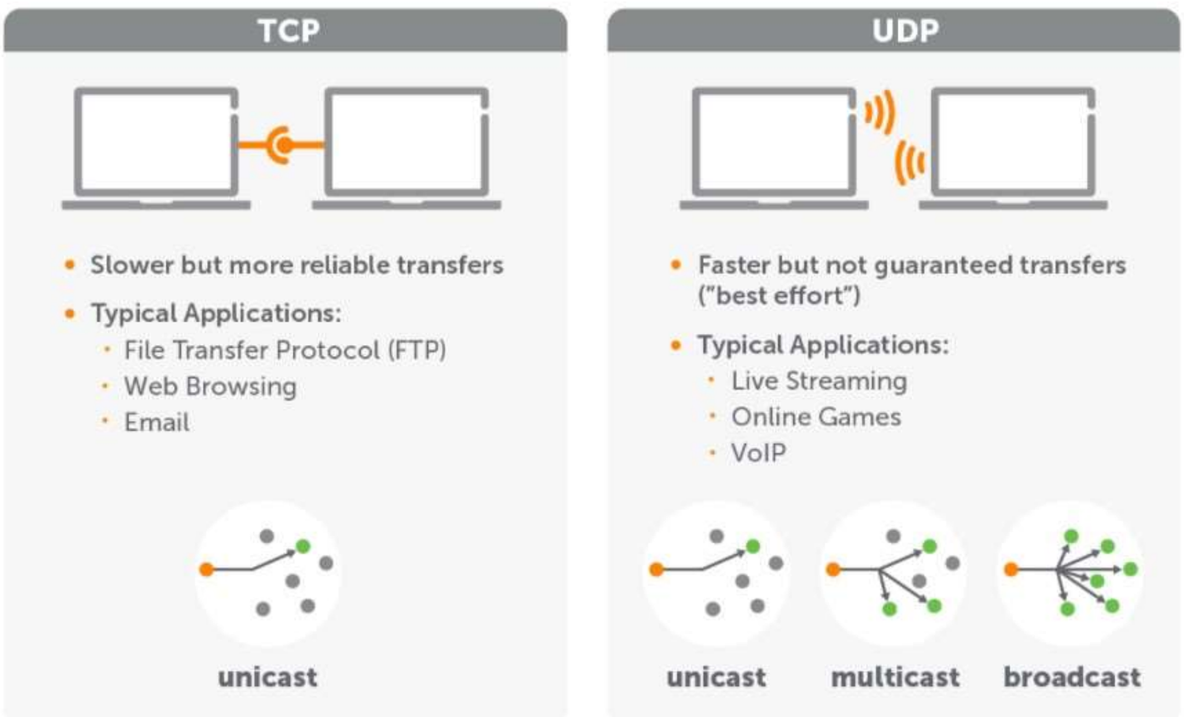

Fig3.2.2. TCP and UDP

- **HLS Low latency protocol**

HTTP Live Streaming (HLS) was initially created and released by Apple in 2009 to solve the problems of scaling. The HLS protocol has become one of the most popular protocols used today, and is widely supported. It's biggest disadvantage, as with other similar HTTP based streaming protocols, is latency.

To combat latency, the industry and community sought to find solutions. Twitter's Periscope with LHLS in 2016 (with the help of experts at THEO!), as well as community L-HLS in 2018. In 2019, Apple released their extension of the protocol: LL-HLS. If you want to read more about the evolution of the specification, check out our blog here. Low Latency HLS, aims to provide the same scalability as HLS, but achieve it with a latency of 2-8s (compared to 24-30 second latencies with traditional solutions).

The HTTP Live Streaming (HLS) protocol delivers live and on-demand content streams to global-scale audiences. Historically, HLS has favored stream reliability over latency. Low-Latency HLS extends the protocol to enable low-latency video streaming while maintaining scalability. The new low-latency mode lowers video latencies over public networks into the range of standard television broadcasts.

Backend production tools and content delivery systems must implement new rules to enable low-latency stream playback. Low-Latency HLS offers new functionality in these areas:

- Generation of Partial Segments
- Playlist Delta Updates
- Blocking of Playlist reload
- Preload hints and blocking of Media downloads
- Rendition Reports

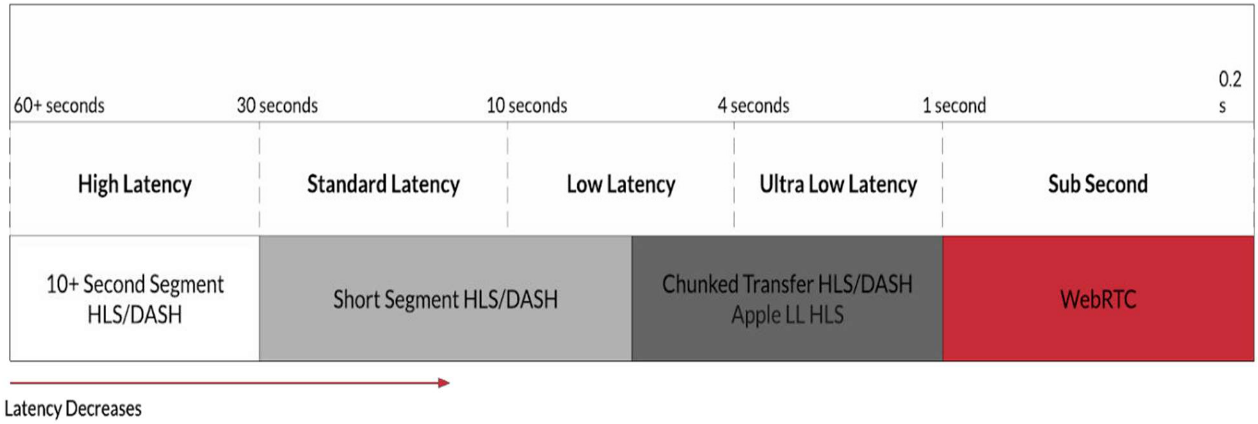

Fig3.2.3. HTTP Low latency rate

- **Mini HTTP Server**

HTTP request splitting is a new concept where the TCP connection and data transfer phases are dynamically split between servers without using a central dispatcher or load balancer. Splitting is completely transparent to the client and provides security due to the inaccessibility and invisibility of the data servers. We study the performance of mini Web server clusters with request splitting. With partial delegation in which some requests are split, throughput is better, and response times are only marginally less than for an equivalent non-split system. For example with partial delegation, for a four-node cluster with a single connection server and three data servers serving 64 KB files, and for a three-node cluster with two connection servers and a single data server serving 4 KB files, the respective throughput improvements over non-split systems are 10% and 22%, with only a marginal increase in response time. In practice, the throughput improvement percentages will be higher and response time gaps will be lower since we ignore the overhead of a dispatcher or load balancer in non-split systems. Although these experiments used bare PC Web servers without an operating system/kernel for ease of implementation, splitting and clustering may also be implemented on conventional systems.

MiniWeb is a high-efficiency, cross-platform, small-footprint HTTP server implementation in C language. It implements GET and POST methods and works on multiple platforms ranging from server, desktop, game console and IoT hardware.

Features

- small footprint HTTP server written in pure C language
- GET & POST actions with basic HTTP authentication
- easy-to-extend with URL handler callbacks
- cross-platform and scalable
- serial port access via HTTP
- integrated UDP server interacting with HTTP server

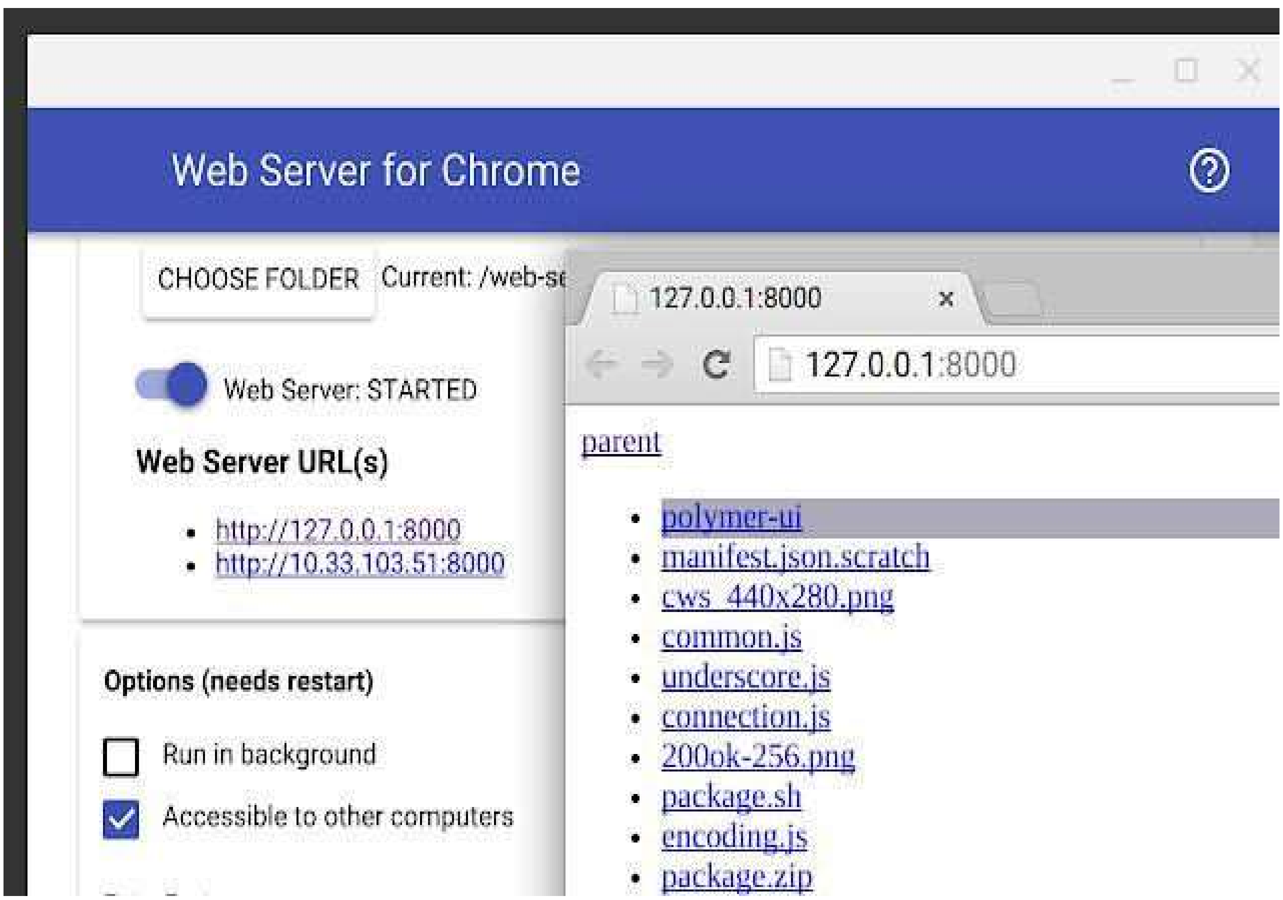

Fig3.2.4. Mini HTTP Server

- **SFT Protocol**

SFTP (SSH File Transfer Protocol, also known as Secure FTP) is a popular method for securely transferring files over remote systems. SFTP was designed as an extension of the Secure Shell protocol (SSH) version 2.0 to enhance secure file transfer capabilities. SFTP supports file access, file transfers, and file management functionalities without command or data channels. Instead, both data and commands are encrypted and transferred in specially formatted binary packets via a single, secured connection using SSH.

Ordinary FTP clients cannot be used with SFTP servers. They require dedicated SFTP clients, which are programs that use SSH to access, manage, and transfer files. The Command-Line Interface (CLI) in UNIX and Mac OS X hosts can be used as SFTP clients. There are also many graphical FTP clients, such as the free FTP client for Windows, FTP Voyager®, which supports file transfers via SFTP.

An SFTP connection can be authenticated in two ways:

- Basic authentication requires a user ID and password from the SFTP client user to connect to the SFTP server.
- SSH authentication uses SSH keys to authenticate SFTP connections instead of, or in combination with, a user ID and password. An SSH public key and private key pair are required in this case.
- Generate a key pair on your computer (SFTP client), and copy the public key to the SFTP server.
- When the server authenticates your connection to it, PuTTY generates a signature using your private key.
- The server, which has the matching public key, can verify this signature and authenticate your connection.

Even if the SFTP server is hacked or spoofed, the attacker gains only one signature, not your private key or password. Because signatures cannot be re-used, they actually gain nothing.

SFTP operates over SSH, making it inherently secure. Unlike in FTPS and FTP, the encryption cannot be triggered or turned off using AUTH commands. Port 22 is generally configured for SFTP connections.

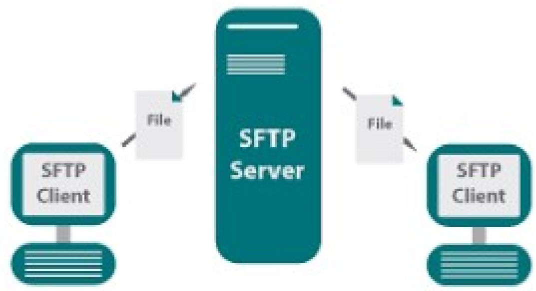

Fig3.2.5. SFTP Architecture

- **SCP**

Not to be confused with the SCP Foundation (an online creative writing exercise), the SCP Protocol stands for Secure Copy Protocol. This is an SSH-based method of transferring files and data securely. The SCP network protocol relies upon SSH certificate authentication. Those who want to ensure that their files are secure while in transit may want to learn more about SCP protocol, how it is supported, how it works, and whether it's safe.

Like SFTP, SCP protocol runs on SSH. It uses both encryption and public key authentication to encrypt the files while they are being transferred and to unencrypt them once they have reached the target destination. Two-factor authentication can be used through SCP protocol and SCP protocol is often used for security standards such as HIPAA and PCI-DSS. As a network protocol, everything related to SCP is done under the hood. The programmer or developer themselves uses the shell program to transfer files, and the SCP protocol takes over when actually transferring and encrypting. Consequently, the person uploading files doesn't need to know in-depth about what the SCP protocol does, though they likely should know about it in terms of security.

Fig3.2.6. STP Architecture

- **Peer – to – peer**

Stands for "Peer to Peer." In a P2P network, the "peers" are computer systems which are connected to each other via the Internet. Files can be shared directly between systems on the network without the need of a central server. In other words, each computer on a P2P network becomes a file server as well as a client.

The only requirements for a computer to join a peer-to-peer network are an Internet connection and P2P software. Common P2P software programs include Kazaa, Limewire, BearShare, Morpheus, and Acquisition. These programs connect to a P2P network, such as "Gnutella," which allows the computer to access thousands of other systems on the network.

Once connected to the network, P2P software allows you to search for files on other people's computers. Meanwhile, other users on the network can search for files on your computer, but typically only within a single folder that you have designated to share. While P2P networking makes file sharing easy and convenient, is also has led to a lot of software piracy and illegal music downloads. Therefore, it is best to be on the safe side and only download software and music from legitimate websites.

A peer-to-peer network allows computer hardware and software to communicate without the need for a server. Unlike client-server architecture, there is no central server for processing requests in a P2P architecture. The peers directly interact with one another without the requirement of a central server.

Now, when one peer makes a request, it is possible that multiple peers have the copy of that requested object. Now the problem is how to get the IP addresses of all those peers. This is decided by the underlying architecture supported by the P2P systems. By means of one of these methods, the client peer can get to know about all the peers which have the requested object/file and the file transfer takes place directly between these two peers.

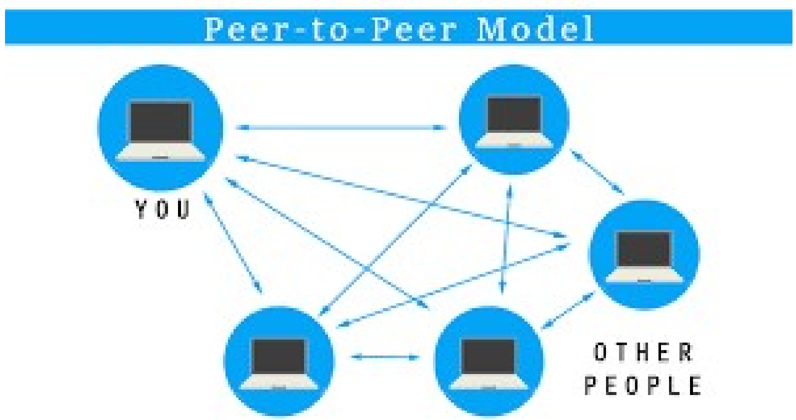

Fig3.6.7. Peer-to-peer model

- **HLS Protocol**

HLS stands for HTTP live streaming. In short, HLS is a media streaming protocol for delivering visual and audio media to viewers over the internet.

Apple first launched the HTTP live streaming protocol in the summer of 2009. Apple timed this release to coincide with the debut of the iPhone 3. As you may recall, previous iPhone models had experienced many problems with streaming media online. These issues arose, at least in part, because those older devices often switched between Wi-Fi and mobile networks mid-stream.

Prior to the release of HLS, Apple used the QuickTime streaming server as its media streaming standard. Though a robust service, QuickTime used non-standard ports for data transfer and so firewalls often blocked its RTSP protocol.

Combined with slow average internet speeds, these limitations doomed QuickTime Streaming Server. As a result, this early experiment in live streaming technology never reached a wide audience.

That said, HTTP live streaming ultimately drew from the lessons learned from creating and rolling out the QuickTime service.

How HLS Works

HLS Container Format

Unlike most HTTP-based protocols, which use the MPEG-4 Part 14 (MP4) container format, HLS initially specified the use of MPEG-2 Transport Stream (TS) containers. This changed in 2016, when Apple announced support for the fragmented MP4 (fMP4) format. Today, fMP4 is the preferred format for all HTTP-based streaming (including MPEG-DASH and Microsoft Smooth). These video files typically contain AVC/H.264 encoded video and AAC encoded audio.

HLS Segment Delivery

As described above, video streams delivered via HLS are broken up into segments of data (also called chunks or packets) at the media server rather than being delivered as a continuous flow of information. Segmented delivery allows the player to shift between different renditions depending on available resources, while also driving down latency.

- **File Transfer Protocol (FTP)**
- FTP stands for File transfer protocol.
- FTP is a standard internet protocol provided by TCP/IP used for transmitting the files from one host to another.
- It is mainly used for transferring the web page files from their creator to the computer that acts as a server for other computers on the internet.
- It is also used for downloading the files to computer from other servers.

- It provides the sharing of files.
- It is used to encourage the use of remote computers.
- It transfers the data more reliably and efficiently.

Although transferring files from one system to another is very simple and straightforward, but sometimes it can cause problems. For example, two systems may have different file conventions. Two systems may have different ways to represent text and data. Two systems may have different directory structures. FTP protocol overcomes these problems by establishing two connections between hosts. One connection is used for data transfer, and another connection is used for the control connection.

There are two types of connections in FTP:

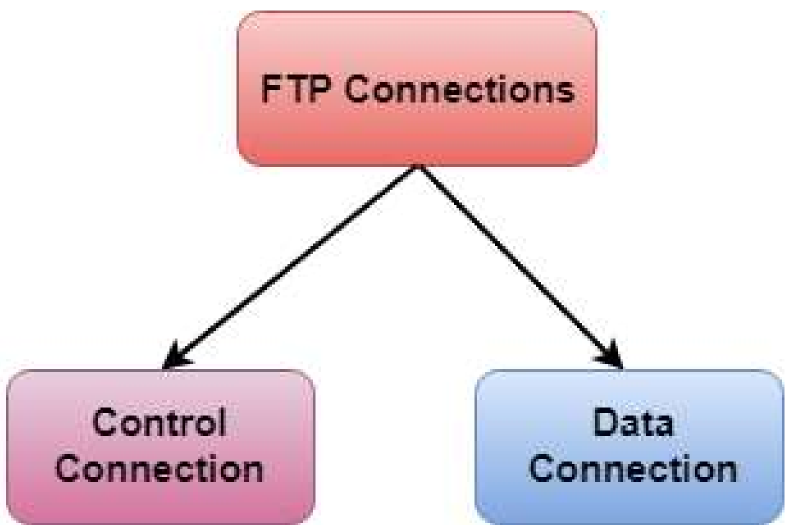

Fig3.6.8. FTP connections

- **Control Connection:** The control connection uses very simple rules for communication. Through control connection, we can transfer a line of command or line of response at a time. The control connection is made between the control processes. The control connection remains connected during the entire interactive FTP session.

- **Data Connection:** The Data Connection uses very complex rules as data types may vary. The data connection is made between data transfer processes. The data connection opens when a command comes for transferring the files and closes when the file is transferred.
- FTP client is a program that implements a file transfer protocol which allows you to transfer files between two hosts on the internet.
- It allows a user to connect to a remote host and upload or download the files.
- It has a set of commands that we can use to connect to a host, transfer the files between you and your host and close the connection.
- The FTP program is also available as a built-in component in a Web browser. This GUI based FTP client makes the file transfer very easy and also does not require to remember the FTP commands.

- **WIFI-Direct**

Wi-Fi Direct is a connection that allows for device-to-device communication, linking devices together without a nearby centralized network. One device acts as an access point, and the other device connects to it using Wi-Fi Protected Setup (WPS) and Wi-Fi Protected Access (WPA/WPA2) security protocols. The standard was developed and incorporated in devices in the early 2000s.

"Wait, that sounds like Bluetooth" might be your response, and while the technologies may look similar at a glance, there are some crucial differences. One of the most important is that Wi-Fi Direct can handle more information at higher speeds than Bluetooth — around 10 times the rate in optimal conditions. This increase makes Wi-Fi Direct a great choice when a peer-to-peer connection needs to transmit data-rich content, like a high-resolution image or a video — or when the WIFI is down.

One of the most significant advantages of Wi-Fi Direct is how versatile it can be where there's no Wi-Fi network to act as a go-between for devices. Multiple devices can link to each other and share important files in casual settings or desperate circumstances alike, without the security worries (and time-consuming process) that come with connecting to a hub or central network first.

You can often tell when a device offers Wi-Fi Direct because when you are searching, it will pop up with a wireless network of its own, usually one that starts with "DIRECT" followed by a product name or number.

Fig3.2.9. WIFI direct

3.3 Methodology

Streaming

- The streaming part of the application uses the low-latency streaming protocol over traditional streaming protocols which are emphasizes on quality of the content
- HTTP based streaming protocol called low-latency HLS is used in streaming process of the application
- The mini-HTTP server created in the host device and low-latency HLS protocol facilitates the delivery of the data between the devices.
- HLS stands for HTTP Live Streaming. In short, HLS is a media streaming protocol for delivering visual and audio media to viewers over the internet.

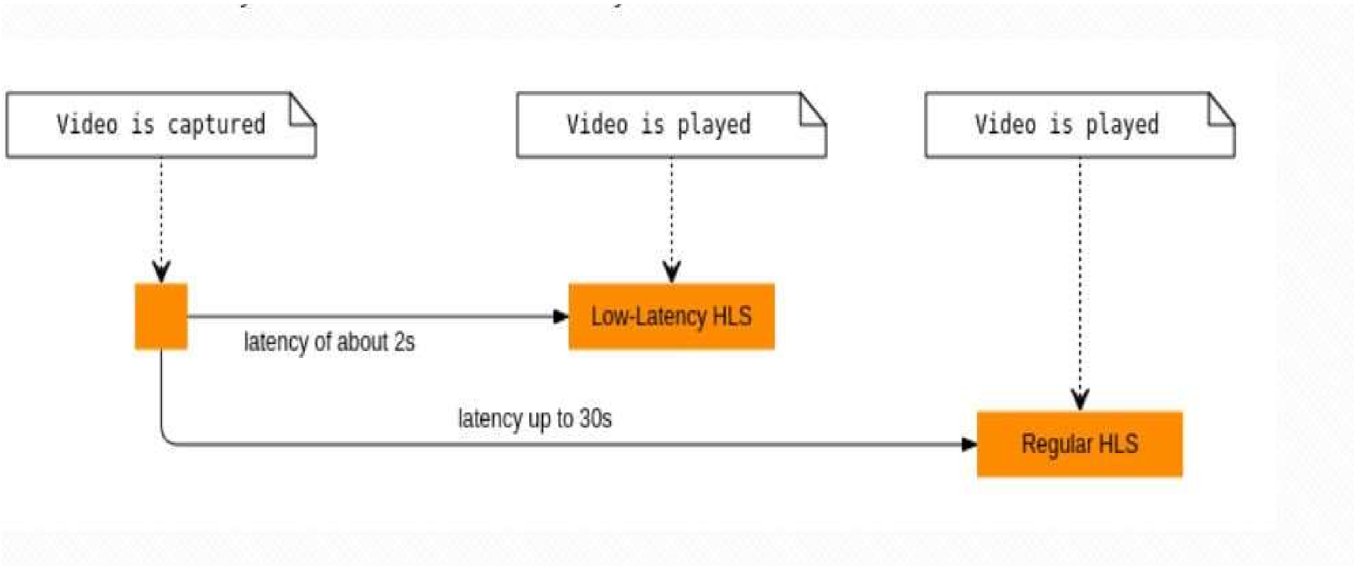

Fig 3.3.1. HLS protocol architecture

- First, the HLS protocol chops up MP4 video content into short (10-second) chunks with the .ts file extension (MPEG2 Transport Stream). Next, an HTTP server stores those streams, and HTTP delivers these short clips to viewers on their devices.
- HLS will play video encoded with the H.264 or HEVC/H.265 codecs.
- The HTTP server also creates an M3U8 playlist file (e.g. manifest file) that serves as an index for the video chunks. That way, even if you choose to broadcast live using only a single quality option, the file will still exist.
- Now, let's consider how playback quality works with HLS video streaming. With this protocol, a given user's video player software (like an HTML5 video player) detects deteriorating or improving network conditions.
- If either occurs, the player software first reads the main index playlist and determines which quality video is ideal. Then the software reads the quality-specific index file to determine which chunk of video corresponds to the point at which the viewer is watching. If you're streaming with Dacast, you can use

your M3U8 online player to test your HLS stream. Though this may sound technically complex, the entire process is seamless for the user.

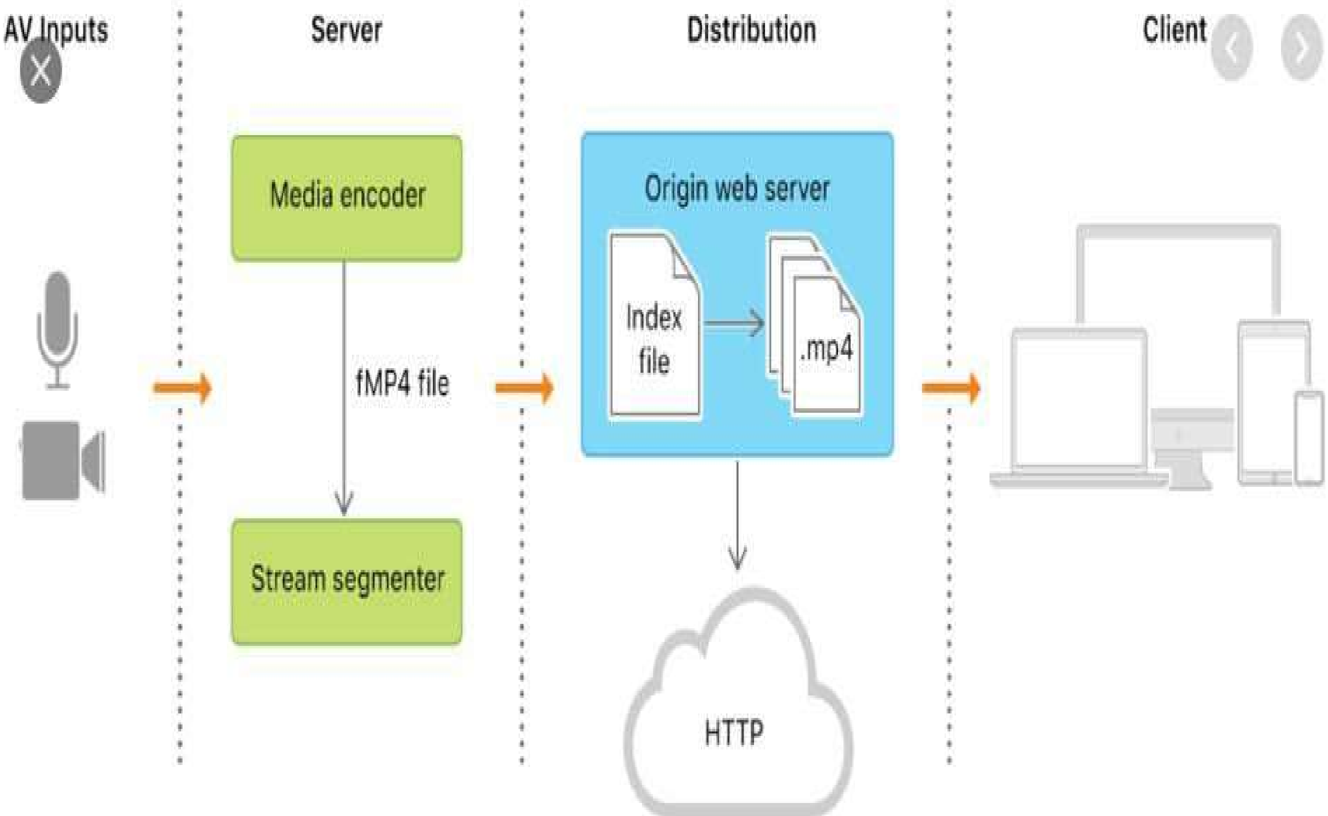

Fig 3.3.2. Working of mini http server

file transferring

- The sharing feature of the application uses peer-to-peer(p2p) file transfer protocol i . e . , FTP protocol, and transfers the data between devices.
- Application uses the WIFI-direct feature to facilitate data transfer between the devices.
- FTP (File Transfer Protocol) is a network protocol for transmitting files between computers over Transmission Control Protocol/Internet Protocol (TCP/IP) connections. Within the TCP/IP suite, FTP is considered an application layer protocol.
- In an FTP transaction, the end user's computer is typically called the local host. The second computer involved in FTP is a remote host, which is usually a server. Both computers need to be connected via a network and configured properly to transfer files via FTP. Servers

must be set up to run FTP services, and the client must have FTP software installed to access these services.

- Although many file transfers can be conducted using Hypertext Transfer Protocol (HTTP) -- another protocol in the TCP/IP suite -- FTP is still commonly used to transfer files behind the scenes for other applications, such as banking services. It is also sometimes used to download new applications via web browsers.

FTP is a client-server protocol that relies on two communications channels between the client and server: a command channel for controlling the conversation and a data channel for transmitting file content.

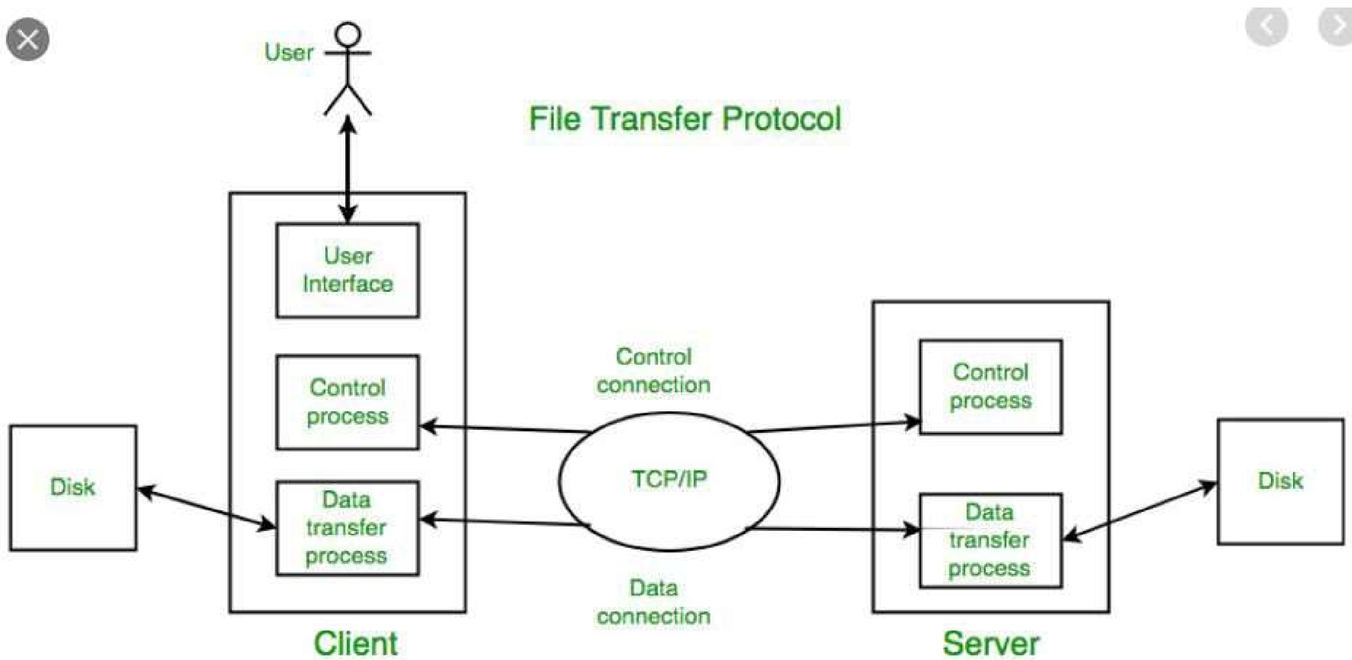

Fig3.3.3. File Transfer protocol architecture

Here is how a typical FTP transfer works:

A user typically needs to log on to the FTP server, although some servers make some or all of their content available without a login, a model known as anonymous FTP. The client initiates a conversation with the server when the user requests to download a file. Using FTP, a client can upload, download, delete, rename, move and copy files on a server.

Chapter – 4

4. System Design

4.1 Use Case Diagram

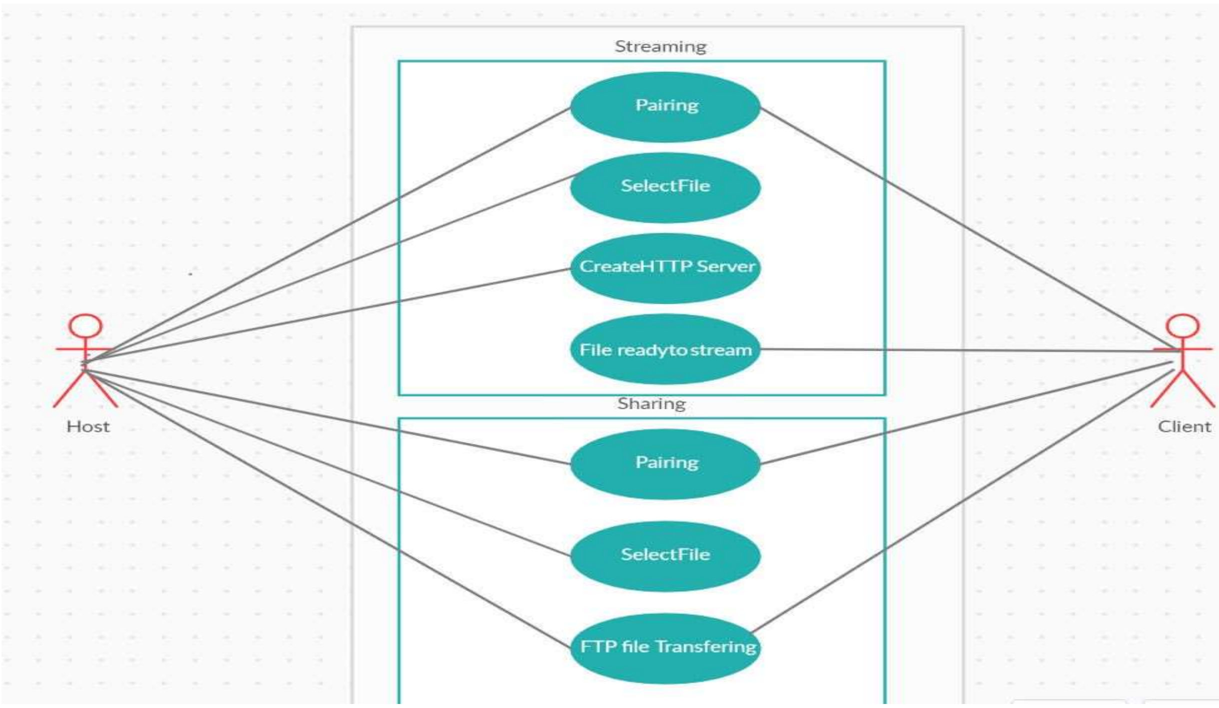

Fig 4.1.1 Use case Diagram

Explanation of use case diagram:

Host side: When the application starts from the host side which means when the person who wants to send the data starts the application the first part is to click on send button in the home activity of the application and then the host has to select the files or contents which can be one or more and then click on send there after the host has to be connected with the client so that he can send the one or more files or contents to the client (recipient). For that host need to join the same network which the client has joined which basically means the both host and client should be connected in same network , it can be any local network which means it can be a open network or it can be a hotspot connection or it can be a WIFI direct feature or can be anything but should be connected in the same network . If it is connected, the host can able to see the device in the pairing sequence activity. If it is not connected to same network or local network it will allow you to setup a local network such as WIFI hotspot ,host can enable hotspot directly from the default settings of the android device or the application allows you to setup a local network directly in the pairing sequence. When you setup the local network you will be shown the name of the network so that the client can able to connect to the network setup by the host and the pairing sequence activity also allows both the devices to use the QR code option which is more reliable and efficient to connect to the network. When the host enables the hotspot from the application in the pairing sequence the QR code will be visible in the activity, and the client can open the default scanner from the setting in the device or can use the scan code feature from the application itself by which client can scan the QR code of the host to connect to the network setup by the host . By this pairing sequence now both the devices are connected in the same network, which means the files or contents can be shared to the client. After that the transferring of files will happen there after when the transferring completes both the devices can exit from the application.

Client side:

When the application starts from the client side which means when the person who wants to receive the data starts the application the first part is to click on receive button in the home activity of the application so that client can directly enter into pairing sequence where connection between the devices happens (Between the host and client). For that client need to join the same network which

the host has joined which basically means the both host and client should be connected in same network, it can be any local network which means it can be a open network or it can be a hotspot connection or it can be a wifi direct feature or can be anything but should be connected in the same network. If it is connected, the host can able to see the device in the pairing sequence activity. If it is not connected to same network or local network it will allow you to setup a local network such as Wi-Fi hotspot, client can enable hotspot directly from the default settings of the android device or the application allows you to setup a local network directly in the pairing sequence. When you setup the local network you will be shown the name of the network so that the host can able to connect to the network setup by the host and the pairing sequence activity also allows both the devices to use the QR code option which is more reliable and efficient to connect to the network. When the host enables the hotspot from the application in the pairing sequence the QR code will be visible in the activity, and the host can open the default scanner from the setting in the device or can use the scan code feature from the application itself by which client can scan the QR code of the client to connect to the network setup by the client. By this pairing sequence now both the devices are connected in the same network, which means the files or contents can be received from the host After that the transferring of files will happen there after when the transferring completes both the devices can exit from the application.

4.2 Data Flow Diagram

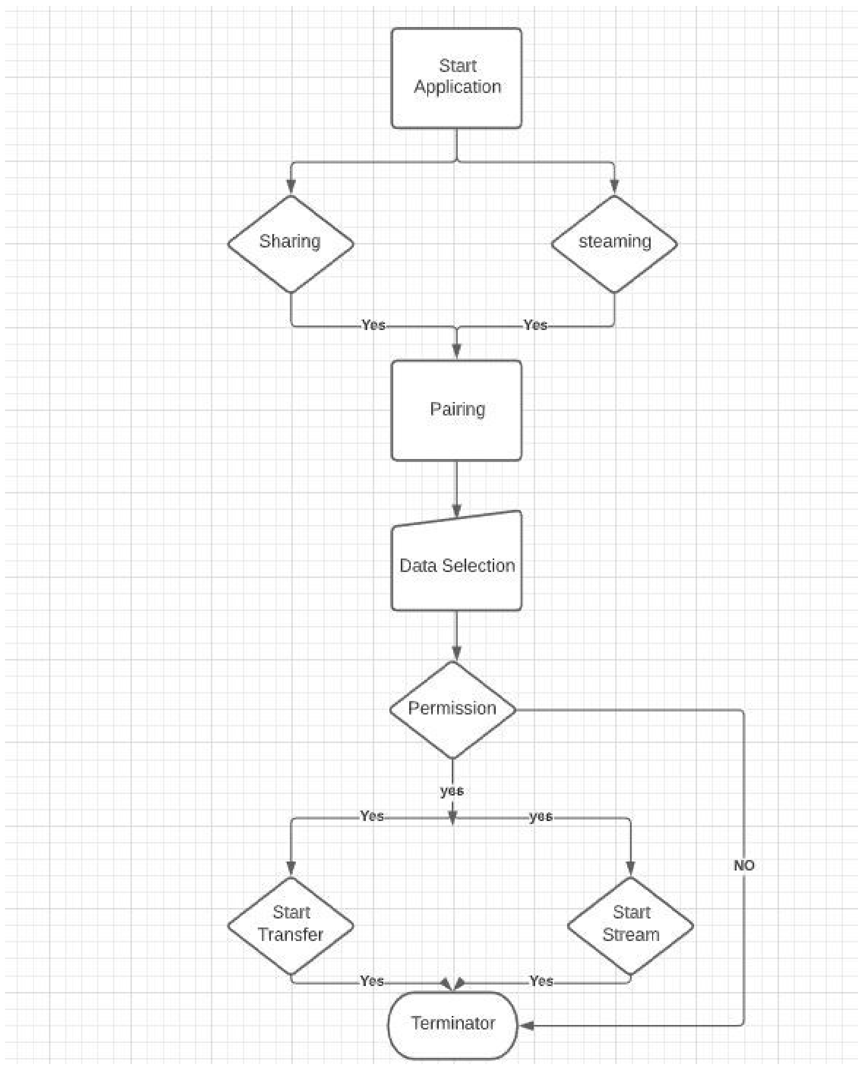

Fig 4.2.1 Data flow diagram

Explanation of data diagram:

When the application is started from either side of host or client there will be options to send or receive files, the users have to select the respective action according to their choice of action to perform. There after sender will be having certain actions to perform and receiver will be having certain action to perform. In the sender side firstly after clicking on send button there will be an activity to select the files or the contents which can be one or more. Thereafter on selecting the files there will be an icon which shows the send action then the paring sequence will happen, in which the

connection between the sender and receiver will happen. The connection between the devices should be the local connection which means both the devices should be connected in the Sam network in order to perform the sharing. The local network can be a open network connection or a WIFI hotspot connection or Wi-Fi direct connection or can be any other type but should be connected to the same network. If both devices are not connected to any type of local network the application will provide the option to setup the hotspot which is again a local network. By that the client can able to join the network created by the host. The application also provides the QR code feature by which the client can scan the code in order to join the network created by the host. After successful connection between the devices Transferring of files or contents will happen, after the successful sharing user can exit from the application. If the user selects the receive option more over the process is same but after clicking on receive button the receiver will directly go to pairing and the pairing process is exactly same as the sender. But the connection should be necessarily be a local network which means both the devices should be connected to same network. There after successful transferring user can exit from the application

Chapter – 5

5. Hardware and Software Requirements

5.1 Hardware Requirement

- Laptop/Desktop
- RAM (Min. 4GB)
- ROM (Min. 8GB)
- 1280 x 800 minimum screen resolution
- Android device

5.2 Software Requirement

- 64-bit Microsoft Windows 8/10Android studio
- Android OS (version 8 & above)
- Java and XML

Conclusion

We have studied different methodologies for the multimedia streaming and sharing which helps the users to share data over and over again. Despite of having many applications to do so it is very costly and time taking. Our application will let the user to transfer the multimedia data over a local connection which can be a open network or a WIFI hotspot connection or a Wi-Fi direct connection and our application allows users to connect in a more reliable and efficient way by providing the QR code feature to setup the local network and can connect to it . The application does not require an internet connection. For streaming we are using low latency HLS protocol which allows users to stream the multimedia from the host device in which a mini http server is created when the application is started, And when user enters the ip address provided by the host device in any of the clients such as chrome, Safari or any kind of browser which enables the user to choose from the contents from the host which are shown in the browser .For all this process the devices should be in a local network same as done for sharing of files. There after the streaming can be done without any internet.

Future scope

- Further, we are trying to use different protocols other than HLS protocol for better efficiency.
- Improvement of user interface is on the way.
- We will try to implement new features such as chatting while the files are being transferred.
- We will try to implement to stream larger files by using different protocols such as SFTP,STP.

REFERENCES

[1] Author1, Author2, .. AuthorN , "Paper/ Publication or Book Title," in *Journal or conference name;* <Journal issue details – volume no, page no., or Place of conference if applicable>, Year of publication.

<Please see examples below>

[2] G. Eason, B. Noble, and I.N. Sneddon, “On certain integrals of Lipschitz-Hankel type involving products of Bessel functions,” Phil. Trans. Roy. Soc. London, vol. A247, pp. 529-551, April 1955.

[3] J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68-73.

[4] I.S. Jacobs and C.P. Bean, “Fine particles, thin films and exchange anisotropy,” in Magnetism, vol. III, G.T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271-350.

[5] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, “Electron spectroscopy studies on magneto-optical media and plastic substrate interface,” IEEE Transl. J. Magn. Japan, vol. 2, pp. 740-741, August 1987

[6] T. Shiao-Li and E. C. Cheng, "PIANO: a power saving strategy for cellular/VoWLAN dual-mode," *Wireless Network*, vol. 14, pp. 683-698, 2008.

[7] A. Yuvraj, C. Ranveer, W. Alec, B. Paramvir, C. Kevin, and G. Rajesh, "Wireless wakeups revisited: energy management for voip over wi-fi smartphones," in *Proceedings of the 5th international conference on Mobile systems, applications and services;* San Juan, Puerto Rico: ACM, 2007.

[8] T. Shiao-Li and E. C. Cheng, "PIANO: a power saving strategy for cellular/VoWLAN dual-mode," *Wireless Network*, vol. 14, pp. 683-698, 2008.

[9] J. Eun-Sun and H. V. Nitin, “Improving IEEE 802.11 power saving mechanism,” Wireless *Network,* vol. 14, pp. 375-391, 2008.

[10] T. Shiao-Li and E. C. Cheng, "PIANO: a power saving strategy for cellular/VoWLAN dual-mode," *Wireless Network*, vol. 14, pp. 683-698, 2008.

[11] G. K. W. Wong, Z. Qian, and D. H. K. Tsang, "Joint Optimization of Power Saving Mechanism in the IEEE 802.16e Mobile WiMAX," in *Global Telecommunications Conference, 2009. GLOBECOM 2009. IEEE*, 2009, pp. 1-6.

[12] E. Shih, P. Bahl, and M. J. Sinclair, "Wake on wireless: an event driven energy saving strategy for battery operated devices," in *Proceedings of the 8th annual international conference on Mobile computing and networking* Atlanta, Georgia, USA: ACM, 2002.

[13] G. K. W. Wong, Z. Qian, and D. H. K. Tsang, "Joint Optimization of Power Saving Mechanism in the IEEE 802.16e Mobile WiMAX," in *Global Telecommunications Conference, 2009. GLOBECOM 2009. IEEE*, 2009, pp. 1-6.