Improving Video-on-Demand Performance by Multi-channel/Multicast Approach in Wireless Networks

Reza Shokri Kalan

Submitted to the
Institute of Graduate Studies and Research
in partial fulfillment of the requirements for the Degree of

Master of Science
in
Computer Engineering

Eastern Mediterranean University
January 2013
Gazimağusa, North Cyprus

Approval of the Institute of Graduate Studies and Research

Prof. Dr. Elvan Yılmaz
Director

I certify that this thesis satisfies the requirements as a thesis for the degree of Master of Science in Computer Engineering.

Assoc. Prof. Dr. Muhammed Salamah
Chair, Department of Computer Engineering

We certify that we have read this thesis and that in our opinion it is fully adequate in scope and quality as a thesis for the degree of Master of Science in Computer Engineering.

Assoc. Prof. Dr. Işık Aybay
Supervisor

Examining Committee

1. Assoc. Prof. Dr. Işık Aybay
2. Assoc. Prof. Dr. Muhammed Salamah
3. Asst. Prof. Dr. Gürcü Öz

ABSTRACT

Video on-demand (VoD) services and applications have become popular in wireless networks. VoD refers to requesting video which is stored on remote media servers. Many commercial products (e.g. PPLive and PPStream) are developed to support VoD applications. Two popular approaches in VoD area are peer-to-peer (P2P) and client-server models. In a network (WAN, LAN, WWAN, or WLAN) when a client initiates a video request, it can stream the video from the servers on the network or from peer clients. Many networks use P2P technology to improve the VoD performance. In P2P models, users store watched videos in their own local storages and can distribute them through the network to other peers when other users want to watch the same video. In this manner clients can download video from suitable neighbors.

There are two ways for accessing a video stream: (a) over wired networks, (b) over wireless networks. Video on-demand streaming has been applied in wired networks with success. However, it remains a challenging task in wireless networks due to bandwidth problems and the effect of wireless interference and client mobility. This study aims at improving VoD performance in wireless networks based on multicasting and patching in clients. The simulation results show that the suggested methods improve VoD performance.

Keywords: Video on demand, Wireless networks, Cellular networks, Frequency band, and Buffer management

ÖZ

Video on-demand (VoD) hizmetleri ve uygulamaları kablosuz ağlarda popüler olarak kullanılmaktadır. VoD uzaktan medya sunucuları üzerinde depolanan bir videoyu talep etmek anlamına gelir. VoD alanında iki popüler yaklaşım peer-to-peer (P2P) ve istemci-sunucu modelleridir. WAN,LAN,WWAN,veya WLAN türü bir ağda bir istemci bir video talep ettiğinde, ilgili videonun bölümlerini uzak sunuculardan veya "komşulardan" indirebilir. P2P modelinde kullanıcılar izledikleri videoları kendi yerel bellekerinde tutarak, aynı ağ içinde başka istemcilerden gelecek istekleri cevaplamak üzere devreye girebilirler. Böylece bir istemci , izlemek istediği videonun bölümlerini " komşusundan" indirebilir.

Bir videoyu kablolu veya kablosuz ağlardan indirmek mümkündür. Kablolu ağlar için bu konuda yeterli çalışma yapılmıştır. Ancak, kablosuz ağlarda bağlantı hızı, kullanıcının hareketliliği ve girişini sorunları nedeniyle etkin video transferi halen önemli bir araştırma konusudur. Bu çalışmada, kablosuz ağlarda video transfer performansını arttırmak içın çoklu iletim (multicasting) ve yamalama (patching) yöntemleri önerilmiş ve modellenerek simülasyonu gerçekleştirilmiştir. Bulunan sonuçlar, önerilen yöntemlerin performansı arttırabildiğini göstermektedir.

Anahtar Kelimeler: Talep edilen video, Kablosuz ağlar, Hücresel ağlar, Frekans bandı ve Tampon bellek yönetimi.

Dedicated to my parents with love

ACKNOWLEDGMENT

I would like to express my deepest gratitude and appropriation to my supervisor, Assoc. Prof. Dr. Işık Aybay, for his patient guidance and encouragement throughout my education. His knowledge, experience, and discipline have been an important lead for my study.

I would like to thank all faculty members of Computer Engineering Department, Eastern Mediterranean University, who helped me during my education.

PREFACE

In the middle of the journey of our life I came to myself within a dark wood where the straight way was lost.

You shall find out how salt is the taste of another man's bread, and how hard is the way up and down another man's stairs.

“Dante Alighieri”

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS OR LIST OF ABBREVIATIONS

ARFCN ....Absolute Radio Frequency Channel Numbers

AMPS ....Advanced Mobile Phone System

CCK ....Complementary Code Keying

CDMA ....Code Division Multiplexing Access

DPCCH ....Dedicated Physical Common Channels

DPDCH ....Dedicated Physical Data Channels

DSSS ....Direct-Sequence Spread Spectrum

EDGE ....Enhanced Data Rates for GSM Evolution

EVDO ....Evolution Data Optimized

FHSS ....Frequency-Hopping Spread Spectrum

GPRS....................................General Packet Radio Service

GSM ....Global Communication for Mobile system

HSDPA ....High-Speed Downlink Packet Access

HSPA ....High-Speed Packet Access

LTE ....Long Term Evolution

MUD ....Multi User Detection

OFDM ....Orthogonal Frequency Division Multiplexing

OFDMA ....Orthogonal Frequency Division Multiple Access

PCS ....Personal Communication Service

PDCP.................................... Packet Data Convergence Protocol

PDU......................................Program Data Unit

RLC.......................................Radio Link Control

RNC ....Radio Network Control

RRC.......................................Radio Resource control

TACS ....Total Access Communication System

TD-CDMA............................Time Division –CDMA

TFCI......................................Transport Format Combination Indicator

TPC ....Transmit Power Control

UMB ....Ultra Mobile Broadband

UMTS ....Universal Mobile Telecommunication System

WCDMA ....Wideband Cod Division Multiplexing Access

Chapter 1

INTRODUCTION

Video on demand (VoD) is a multimedia system which enables clients to access and play video from remote media server through wired or wireless networks. VoD systems can be categorized into two main approaches: (a) client/server and (b) peer-to-peer. In the client/server technique, the video files are stored at multiple servers. Each client independently connects to the server and plays the requested video from the server. Although the client/server model is the simplest, with increasing number of clients, bottleneck (or I/O) problems reduce server performance. Peer-to-peer (P2P) as a decentralized and multi point communication technique reduces server loads and tries to avoid bottleneck problems on the server side by saving severs bandwidth.

Internet bandwidth is mainly allocated to three services: data, voice, and video. As the numbers of multimedia applications grow, we need more bandwidth and storage area to support different users' services. Mobile users like to access multimedia anywhere, anytime. Due to user mobility, there may be problem in channel allocation in wireless cellular networks. A wireless cellular communication system should be able to provide continuous service when a user is moving from one cell to another (hand off/ hand over) [1]. Under this condition, we need channel reservation that includes process of changing frequency, time slot, and etc.

The simplest approach to stream media data to a client is to create a unicast connection from the server to each requesting user, where the media server is situated behind the Base Station (BS). The BS has limited memory and it cannot work as a proxy server. In this case, we need individual free channels for each user's communication. Communication between Mobile Station (MS) and BS is started if there is a free channel, otherwise MS must be waiting until channel allocation. When a new user arrives in cell coverage area, a request for media file is sent to BS. For avoiding the bottleneck in the media server, first of all, BS will be looking for requested media in other MS's in its coverage area. A peer-to-peer (P2P) connection can be established between two MSs through the BS if there is at least one MS which has the media file. Otherwise, BS sends new user's request to the media server and gets the media file. Unfortunately, this approach will quickly consume all available bandwidth of the downlink communication when multiple users concurrently request the media services. Also there are some other problems in this approach including:

- ✓ Bottleneck problem in media server- too many request from clients lead media server to bottleneck or I/O problems.
- ✓ Caching problem- Mobile devices have limited memory size for caching
- ✓ Distance problem- It is impossible P2P communication without BS, because two peers may be far away each other
- ✓ Free channel problem- MS-BS and P2P communication is not possible in lack of BS free channel
- ✓ The MS with the resource may leave the coverage area

Although recently mobile devices come with more memory size which can be used for caching media streams, cannel allocation is still a big problem. Multicast as a stream sharing mechanism can improve the frequency spectrum utilization of wireless communication in multipoint services.

Resource consumption management is the main aim of a VoD system. In particular, our proposed architecture tries to improve the following:

- ✓ Start-up latency- The time between issuing the request for playback, and the start of the playback. If the waiting time is almost zero, the system delivers true VoD. If the waiting time is significant but smaller than the length of the video, the system delivers near true VoD [2, 3].
- ✓ Server and core network traffic- Improving server load and network traffic by limiting the number of unicast communication.
- ✓ Buffer content management- Providing new algorithms in order to improving video locality.

In our approach (multi-channel/multicast technique), the downlink bandwidth (server-to-client) is bigger than uplink bandwidth (client-to-server) side. Therefore uplink bandwidth is more expensive. A wireless cellular network covers a wide area, but the limited bandwidth causes unicast communication to be costly. On the other hand, WLAN supports a small coverage area, but it has more bandwidth and is also cheaper than cellular networks. The aim of our approach is minimizing individual unicast connections to the media server. Caching system at the client side improves data access latency. Due to limited bandwidth, each client can forward cached segment to one client at the same time. The sending client switch to active mode during cache forwarding and stays in this mode until cache forwarding is finished.

All former Periodic Broadcasting techniques try to reduce service latency and minimize resource requirements by reducing the number of unicast requests and all of the techniques focus on the server side. Although installing Proxy Server (PS) on BS reduces server load and service latency, it will have high cost. We set up a few proxy servers in Base Station Control (BSC) area. In cellular networks all BSC's know the network topology. This does not increase network complexity and is more cost effective. BSC's are located at a fixed position and have high percentage of resources such as memory, computing power, and battery life.

To observe the efficiency improvement, we simulated our approach using OpNet simulator. According the simulation results, multi-channel/ multicast technique improve start up latency and increase speed up. Switching the client between multicast channels has an overhead but it decreases throughput. Furthermore, switching among too many channels is not possible in practice. Therefore, multi-channel/ multicast approach has better efficiency when number of channels is limited to two or three.

Chapter 2

RELATED WORK AND WIRELESS NETWORKS ARCHITECTURE

2.1 Related Works

Providing on-demand services to a large number of clients in *‘‘real time’’* imposes a high resource requirement on the underlying network and server [4]. Therefore, traditional client/server architecture cannot provide a scalable solution, as it requires a dedicated communication channel per user. VoD servers can be arranged as a centralized, distributed, and proxy or content delivery network (CDN) architecture [4, 5]. Although there is a good background of multimedia services in wired networks [6, 7], there is less research reported on multimedia services in wireless networks.

In Cache-*and-Relay* approach, peers cache downloaded video in their memory, ready to relay the stored video to other peers in future, leading to asynchronous P2P video sharing [8]. Proxy caching improves startup latency and network traffic, but it is costly. VoD eliminates the need for proxy server or a centralized coordinator, and it improves performance by both multithreading and peer-to-peer cooperation based on the locality principle [6]. Therefore, service providers believe that VoD service will become a factor in finding establishing new subscribers, and they are developing cost effective solutions to provide VoD service [9]. Periodic broadcast technique avoids the client/server bottleneck problem (e.g. I/O, storage, and network bottleneck) and it is suitable for transmitting popular videos to a large number of clients in a short period of time, while client/server and P2P techniques are better suitable for non-popular videos or for videos requested by a small client population [10].

Previous research [6, 7, 9 - 16] in VoD in wired and wireless networks improve system performance using different methods such as:

- ✓ using proxy server,
- ✓ using multi-path and multi-source
- ✓ broadcasting protocol for video on demand
- ✓ batching client request in the server through multicasting
- ✓ VoD streaming system in hybrid wireless mobile peer-to-peer networks

2.1.1 Proxy Server

A VoD system includes several servers and distributed clients over the entire wired/wireless network. It is possible to reduce network traffic by installing a proxy server between the media servers and the client. Proxy servers cache popular video streams and attempt to improve the performance of network by:

- ✓ reducing the server load,
- ✓ reducing server latency,
- ✓ reducing network traffic,
- ✓ increasing QoS and
- ✓ Increasing quality of experience (QoE)- quality which sense by end user.

However, proxy server increases network cost and complexity. The problem of the need for frequently updating the Proxy server content is another disadvantage. Algorithms like the least frequently used (LFU) not work well for continuous video streams. Finally, according to Internet/ network nature, proxy server has to have enough flexibility to support clients with a different quality. Elimination of the proxy server or centralized coordinator is one of the main reasons for using peer-to-peer systems. Also, using a proxy server may need extra hardware and software and hence may increase the total system cost.

2.1.2 Periodic Broadcasting Techniques

In the periodic broadcast (PB) VoD technique, a video is first partitioned into a number of segments [12]. Then, each segment is transmitted over a separate broadcast channel periodically. Clients receive a video by switching between channels at a time to download the data. Broadcast approach can use system resources more efficiently than P2P, while multiple copies of a program (NVoD) are broadcast at short time intervals (e.g. 10–20 minutes) [17]. In order to support VoD services, we consider a hybrid approach (by merging, multicasting, proxy server, and buffer management).

2.1.2.1 Staggered Broadcasting

Staggered broadcasting (SB) is the simplest broadcasting protocol [10, 12, and 18]. In this approach, a video is partitioned into k equal-size segments $\{s_1, s_2 \ldots , s_k\}$ which are repeatedly broadcasted over k channels with a transmission rate equal to the communication rate. If the total video time is V, the duration of each segment is s =V/ k. Consider a video, which is divided into 5 segments (broadcasting over 5 channels). If the total video time is equal 30 seconds, the maximum start up delay in worst case is V/k=30/5=6 seconds.

Unfortunately, SB technique has high start-up latency. If during the broadcasting of segment 1 (s_1), a new client arrives in the wireless coverage area and requests this video, s/he has missed broadcasted packets that belong to s_1. So s/he must wait until the next broadcast of this segment [10]. Figure 1 illustrates this approach. If a client requests segment $s_1$at time $t_0 + s + \lambda$ ($0 < \lambda < s$), it must wait until time $t_0 + 2s$ to start downloading segment s1. Hence, the service delay is $s – \lambda$. In the worst case, service delay is s.

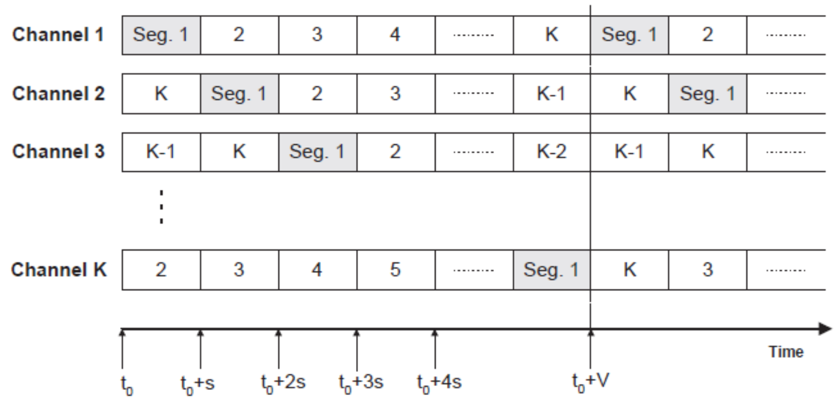

Figure 1: Broadcasting video segments in staggered broadcast manner [16]

The playback algorithm mentioned in [10] is as follows:

1) *Tune in a random channel i. Suppose that channel i is currently broadcasting segment s_h*

   (a) If h = k - 1, let j = i and GOTO Step (2)

   (b) Else, compute

$$j = \begin{cases} i + h - k, & if\ i + h - k > 0 \\ i + h, & otherwise \end{cases} \quad (1)$$

   Channel j must be currently broadcasting segment s_K and about to broadcast segment s_1.

2) *Wait until channel j starts broadcasting segment s_1 (first one) and join this channel when that time comes*

3) *Play the video data received from this channel and quit when the video is finished playing*

2.1.2.2 Skyscraper Broadcasting

To reduce the start-up delay of SB technique, the skyscraper broadcasting (SkB) was developed [12]. Figure 2 / formula (2) show that in this technique, a video is divided into different segment sizes [10, 12]. The system then broadcasts video segments over k channels. At the client side, users receive consecutive segments having the same sizes. Users need to download data from at most two channels at any time and the receiver buffer requirement is constrained by the size of the last segment [12]. The SkB technique supports clients with low bandwidth [19].

$$S_i = \begin{cases} s_i, & i = 1 \\ 2s_i, & i = 2,3 \\ 2s_{i-1} + 1, & i \bmod 4 = 0 \\ s_{i-1}\ , & i \bmod 4 = 1 \\ 2s_{i-1}\ + 2, & i \bmod 4 = 2 \\ s_{i-1}\ , & i \bmod 4 = 3 \end{cases} \qquad (2)$$

E.g. =1, 2, 2, 5, 5, 12, 12, 25, 25, 52, 52

If segment size grows more, a large caching space would be required at the client side. The client cache space and bandwidth are related as follows:

$$\textit{Cache space} = (W-1)\ s_1\ B/k \qquad (3)$$

If *v* is a video file, there is a relationship between the serviced delay and the control factor *W* is [13]:

$$\textit{Delay} = s_1 = \frac{V}{\sum_{i=1}^{k} \min\,(s_{i\,,}\ w)} \qquad (4)$$

Where, W restricts segments size to be large.

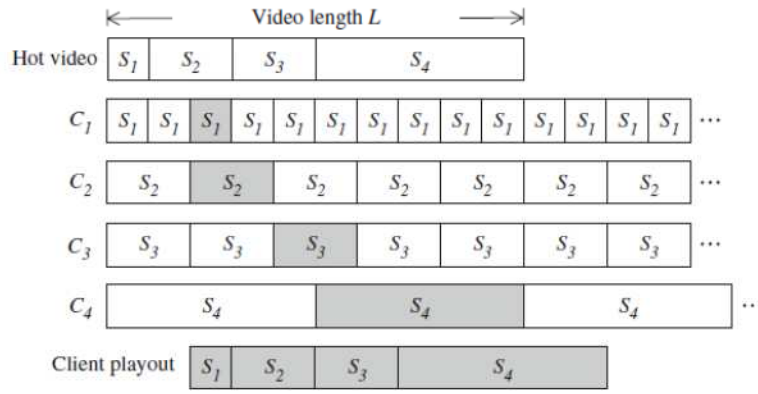

Figure 2: Illustration of segment downloading for the SkB technique, where number of channels is k=4, [19]

2.1.3 Patching Technique

The patching scheme was developed to prepare a true (zero-delay) video stream in broadcast manner. "*The idea of patching is that clients are able to download data on two channels simultaneously when they request for the videos [12]*". As illustrated in figure 3, server keeps a list of the users that have buffered the video.

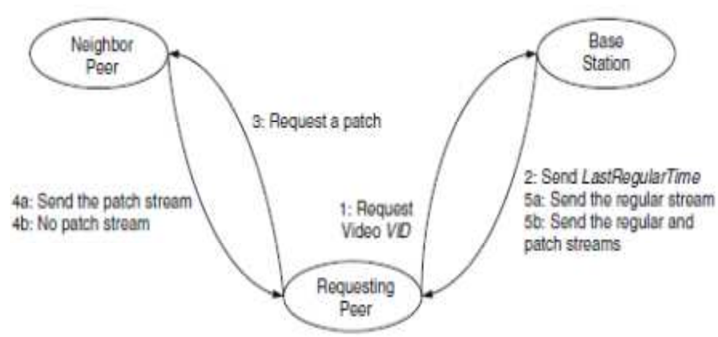

Figure 3: Collaboration diagram of peer patch [20]

When a new client queries the server for the media file, server returns a list of peers cached patches of the video streams. Each node buffers the video stream in FIFO order, therefore, the new client can receive the stream from peers if it still has the first stream in cache. If there are such peers, the new client can stream the patch from peers and concurrently receive regular multicast stream from media server. Otherwise the media server (or BS) must be sent the regular multicast and patching streams.

2.1.4 Multi Path/Multi Source

Multiple independent paths technique is a main technique for the multi-source VoD application in wireless mesh networks. Distribution of video segments over multiple independent paths improves the routing performance, stability, and robustness [11]. Video requests are registered in the server when a new client joins into network and makes a video request. The server keeps a list of all clients that have buffered the video. If there are such users, the new user can stream the video from peers.

New client downloads video from local peers through one or multi-hop. When the peers are located in Internet, client needs to rout path from the gateway to peer in the Internet (Figure 4).In general, because of dynamic channel allocation overhead static channel allocation strategies are used in wireless mesh routers. Therefore, usually the downloaded channel does not change.

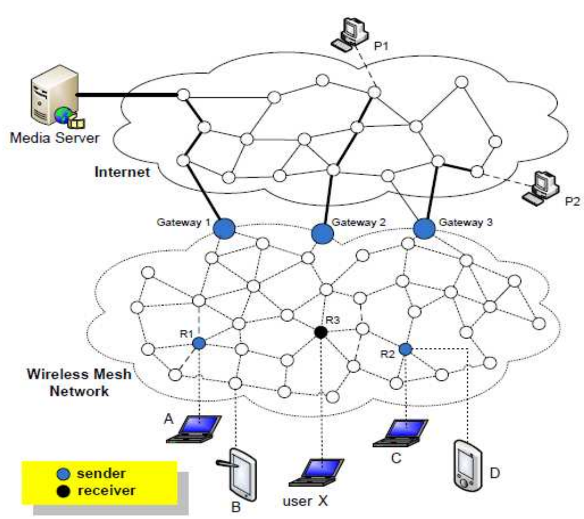

Figure 4: VoD in Wireless mesh networks [11]

2.2 Wireless Network Environment

In general, wireless networks are divided as infrastructure and ad hoc models. There is a central management system such as Base Station (BS) or access point (AP) which is responsible for functionality in infrastructure design. In ad hoc networks, a client is not familiar with the network topology. The new client just advertises its presence and listens to other neighbors' acknowledgement to fine network topology and to learn how to reach them. The communication between clients in ad hoc networks is like the P2P model. So, in lack of central management, all clients are responsible for their own functionality such as managing and routing messages.

Wireless technologies such as WiMAX (IEEE 802.16a-d, e) for long distance and Wi-Fi (IEEE 802.11a, b, and g) and Bluetooth for short distances communication enable mobile users to access resources on wireless networks anywhere and anytime. IEEE 802.11 standards are the most popular form of wireless communication that can support 54Mbps in a maximum of 100 meter coverage area. Coverage area is limited to 10 meter for infrared and Bluetooth networks. Worldwide Interoperability for Microwave Access (WiMAX) appears in two versions [21]: (a) IEEE 802.16a, b, c, and d for fixed user and (b) IEEE 802.16e for mobile users up to a speed of 120km/h. WiMAX support 70 Mbps data rate in 50 km coverage area. As shown in figure 5, it can be used in point-to-point and point-to-multipoint forms.

In recent years, new standards are defined for wireless regional area network (WRAN) known as super Wi-Fi (IEEE 802.22) which is using *white spaces. ("in telecommunication literature, white space point to frequencies allocated to a broadcasting service but, not used locally ")* in the TV frequency spectrum. WRAN networks are based on BS and customer-premises equipment (CPE) that work in point to multipoint (P2MP) manner. The BSs will be managing the medium access for all the CPEs that connect to BS through a wireless link. Each WRAN channels will support up to 22 Mbps without interfering with existing TV broadcast stations [22].

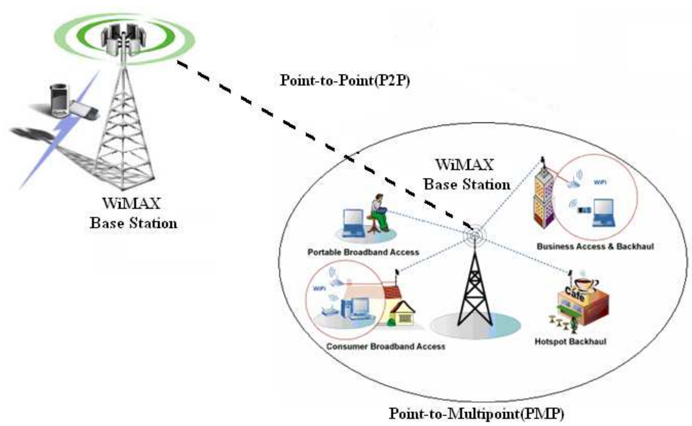

Figure 5: WiMAX communications [21]

2.3 Evolution of Cellular Wireless Networks

First generation cellular and cordless telephone networks such as advanced mobile phone service (AMPS) are based on analog technology and use frequency modulation (FM). The second generation (2G) digital networks (e.g. GSM, PCS) replaced the first generation analog systems and provide speeds of 10 to 20 Kbps. Nowadays 2.5G networks are widely available and cover a wide range of new applications such as multimedia. Furthermore, 2.75G technology supports EDGE networks.

The aim of the third generation (3G) wireless network is to provide a single set of standard that can support a wide range of wireless applications [23]. Third generation (3G) supports demanding multimedia, video conferencing, Voice-over-IP (VoIP) by providing up to 2 Mbps speed. It is compatible with existing GSM technologies like 2G and EDGE. Unlike GSM system, 3G provide multiple interface and platform to a wide range of devices and applications. Unfortunately 3G networks are not ubiquitous, therefore, when 3G subscribers roam out of 3G network coverage, service returns to the 2G/2.5G networks in lack of 3G technologies in new cells.

A fourth generation (4G) is a future technology that includes LTE and UMB. The 4G system provides mobile ultra-broadband Internet access, for varying mobile devices (e.g. Laptops with USB wireless modems). Table 1 summarizes different cellular network technologies.

Table 1: Evolution of Cellular networks [24]

Specifications	Protocol	Maximum speed	Features
1G Analog Network	AMPS	N/A	-Voice service only
	FDMA		
	TACS		
2 G	CDMA	Up to 20 Kbps	-Digital voce service -Short message service(SMS) -Voice mail -Email and web browsing - Conference calling
	GSM		
	PCS		
	TDMA		
2.5 G	CDMA 2000 1*RTT	Up to 144Kbps (typical 60-80Kbps)	All 2G features plus: - MMS (Multimedia Message service) - Web browsing - Real-time location-based services - Basic multimedia, including support for short audio and video clips, games and images
	GPRS	Up to 114Kbps (30-40Kbps)	
	HSCSD	Up to 64Kbps	
	EDGE	Up to 384Kbps	
2.75 G	GPRS 2	473Kbps (uplink) to 1.2Mbps (downlink)	Better performance for all 2/2.5G services
3 G	UTMS	Up to 2.4Mbps	Support for all 2G and 2.5G features plus: • Full motion video • Streaming music • 3D gaming • Faster Web browsing
	WCDMA	Up to 2.4Mbps	
	CDMA 2000 EVDO-Rev A	Up to 3.1Mbps	
3.5 G	HSPDA	Up to 14.4Mbps	Support for all 2/2.5/3G features plus: • On-demand video • Video conferencing • Faster Web browsing
	CDMA 2000 EVDO Rev B	Up to 46Mbps	
4 G	WiMAX	100+Mbps	Support for all prior 2G/3G features plus: • High quality streaming video • High quality video conferencing • High quality Voice-over-IP (VoIP)
	UMB	35Mbps	
	LTE	100Mbps	

2.4 Cellular Network Architecture

Mobile station[1] (MS), base station[2] (BS), base station controller (BSC), and mobile switching center (MSC) are the four main components of cellular wireless networks (Figure 6). The MS may be any portable device, such as a mobile phone, personal digital assistant (PDA), and laptop. The BS coverage area is known as its "cell". The shape and size of cells depend on natural and terrain coverage area [1]. Depending on the type of cellular networks (e.g. Pico cell, micro cell, cell, and macro cell) one cell might cover a radius of from 10m up to 9 miles. Each BS is connected with a neighbor BS and through BSC to MSC, which acts as a gateway from the cellular network to the existing wired networks: the Internet, and the public switched telephone network (PSTN). Each MS communicates via the specific BS of the cell it is currently residing in. In this study, we assign the media server is located behind the BS. In satellite cellular communication, distance between BS (in this case satellite) and the MS is much longer [25]. Due to the large distance between satellite and earth (36000 km in LEO), it is impossible to have cells with diameters less than 100km even with low earth orbit (LEO) [25]. Therefore, transmit power needs to be larger.

In a cellular wireless network, each cell is illustrated in hexagonal shape, therefore the cells fit together tightly. Each cell is assigned a frequency range, so a small cell size enables reusing frequency in other cells without interfering, but this increases network cost. Static channel allocation strategies are widely used between BSs or BSs and BSC. It is because a BS always stays in a fixed position. Each BS periodically broadcasts own cell information such as available bandwidth, number of nodes and etc, to other BS and BSC. Therefore, each BSC knows the global topology

[1] Client, peer, node, and user have a same mean in this paper.
[2] Server

of the wireless cellular network and available bandwidth on all the links in the network and patching video streams are buffered by each node.

Typically, the BS receives information from the MS and sends it towards the BSC. The BS connects to MSC through BSC. The BSC handles channels allocation and handoff operation inside BSC from one BS to others, but handoff between two BSC covered by MSC. In addition, all sites information is stored in the BSC database.

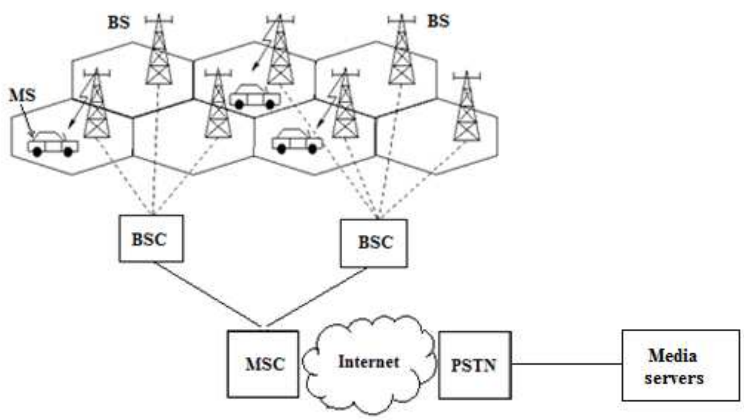

Figure 6: Architecture of a cellular wireless network

2.4.1 Traffic Load and Cell Size

Each cell has different channel frequencies and as the power of a cell transmitter is limited, the same channels can be reused in different cells. The frequency band allocated to a cellular system can be reused in other cells. In the cellular wireless networks, same frequency may be used in other cells. Every cluster includes a set of frequencies which are assigned to adjacent cells. In general practical cluster sizes are 3, 4, 7, and 12. Small cluster size (e.g. N= 1) leads to high interference, while big cluster sizes (e.g. N=27) wastes system resources by insuring a very low interference level that is much lower than the maximum acceptable value. For avoiding interference between cells or clusters we need to calculate reusable distance as follows [25]:

$$N= i^2+j^2+ij \quad (5)$$

$$N=\sqrt{3}\ R \quad (6)$$

$$D/R= \sqrt{3N} \quad (7)$$

$$D/d=\sqrt{N} \quad (8)$$

Where $i>= 0$ and $j>= i$

R = radius of cell

N = number of cells in a cluster

d = distance between centers of adjacent cells

D = minimum distance between centers of cells that use the same frequency band (channels)

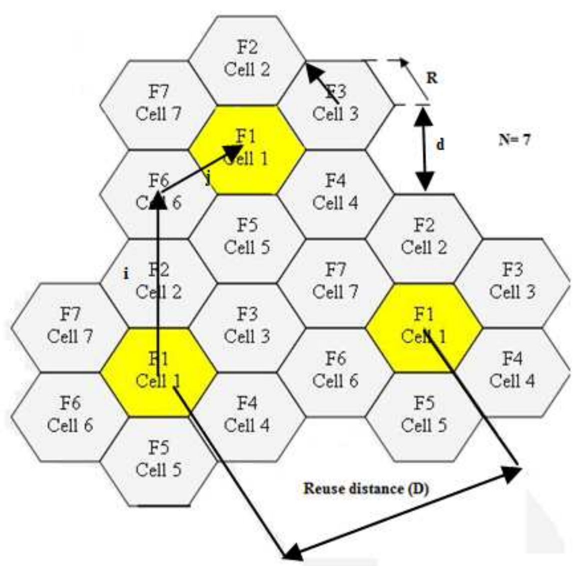

Figure 7: Cluster of N= 7 cells

In general, one BS supports all subscribers in one cell. Increasing subscriber density in cell area leads to more traffic. Therefore, it is necessary to install more BS to achieve QoS. However, installing more BS is costly and the cost of equipment may limit the number of BSs. Sectorizing (figure 8) is the simple approach to overcome this limitation. A Sector antenna divides cell area into three cells, serving 120°

sectors with different channel groups. Power radiated from sectored directional antenna is minimal and has low interference with neighbor cells.

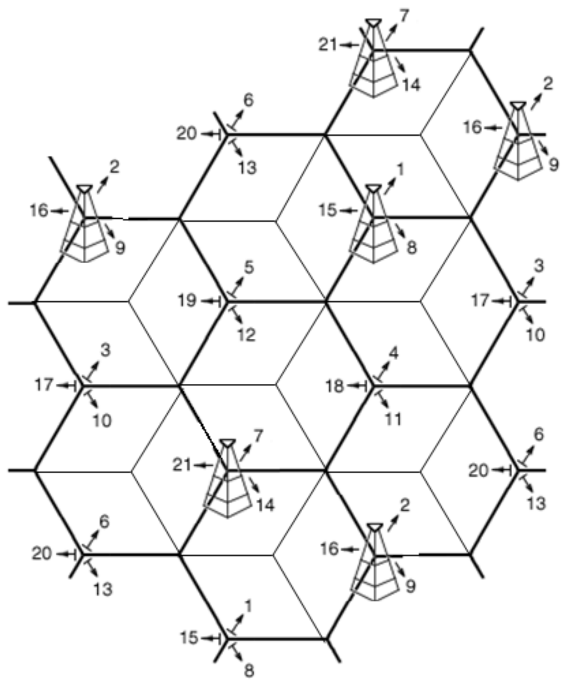

Figure 8: Multi frequency and frequency reuse based on sector antenna, where N=7

2.4.2 Second Generation Technology (2G)

Global system for mobile communication (GSM) uses both FDMA/TDMA techniques, further combined with FDD (frequency division duplex). Uplink and downlink uses separated frequency bands for communication. The lower band is used for communication from MS to BS (uplink) and the upper band is used for communication from BS to MS (downlink). In FDMA, allocated spectrum (uplink and downlink) is divided into 124 individual carrier frequencies. Carrier separation is 200 kHz. The frequency space between uplink and downlink (offset) is 45 MHz. The outer 100 kHz of each 25 MHz band is used as a guard band for avoiding of interference with adjoined spectrum. By dividing total bandwidth (25 MHz) into 200 kHz we have 125 sub frequency bands. The remaining 124(125-1) sub frequency bands are known as Absolute Radio Frequency Channel Numbers (ARFCNs) and deal with channel number to be assigned to one uplink and one downlink.

As illustrated in figure 9, in TDMA each frequency is divided into 8 time slots. Each time slot lasts 576.92 µs (0.577 ms). MS communicate with BS during one time slot. Each set of 8 time slots is known as a TDMA frame. The duration of a TDMA frame is 4.615 ms (576.92 µs × 8). The handset device uses different frequency band in the uplink and downlink. Furthermore, send and receive never happens at the same time. The combination of frequency band and time slot number is known as physical channel. The data type that can be transferred over one physical channel depends on the logical channel.

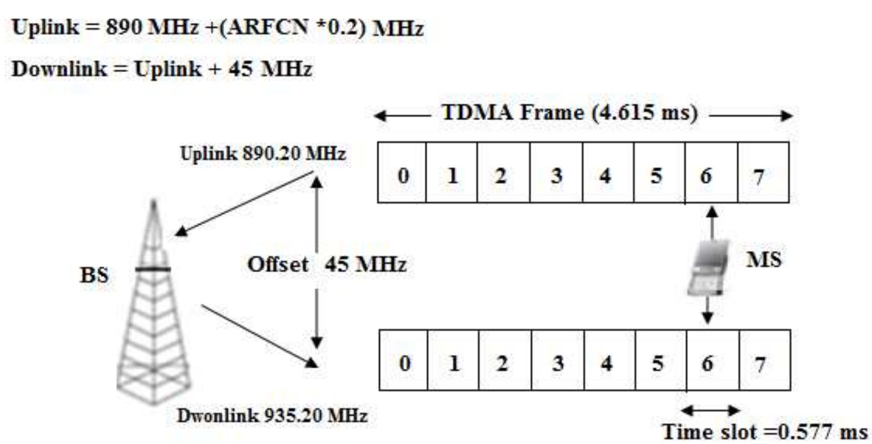

Figure 9: GSM 900 ARFCN

Due to Gaussian Minimum Shift Keying (GMSK) modulation technique, each 200 kHz sub band (one carrier) supports a data rate of 271 kbps (270.833 kbps) [25]. Therefore, maximum transmission rate in one time slot (576.92 µs) will be 156.25 bits. The data carried during in one time slot is known as a *burst*. Figure 10 illustrates the structure of a normal burst. Each burst permits 8.25 bits for *guard time* within a time slot to avoid bursts from overlapping with transmissions in other time slots. Therefore, 148 bits are for each burst (156.25-8.25= 148 bit). Maximum bit rate for single time slot under GMSK modulation is 24.7 kb/s.

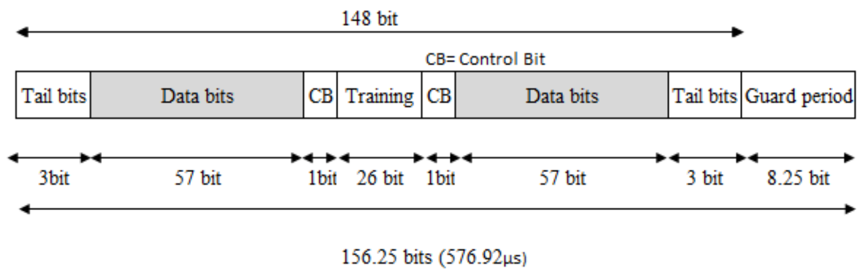

Figure 10: Functions of the bits in normal transmission burst.

The physical channel includes information which is carried by 200 kHz during 0.577ms.The logical channel carrier information about the physical channels. Logical channels can be different types of data that is transmitted only on certain frames in a certain time slot. Traffic channel and control channel are two main categories of logical channels in GSM systems. Traffic channels transport speech and data information. Control channels deal with network management messages and channel maintenance tasks [25].

2.4.3 Beyond 3G Technology

From the technical point of view, GSM and GPRS are not suitable for mobile video communication, because of low data rate. Providing VoD in a cellular network needs high data rate technology such as the 3G system. Therefore, it is necessary to use 3G and beyond 3G technology to achieve VoD with acceptable QoS.

Code division multiple accesses (CDMA) is a spread spectrum technology, it enables many users to concurrently access media at the same time/ frequency by assigning unique codes to each client in the same spectrum. CDMA is transmits simultaneous signals over a shared portion of the spectrum. There is no frequency reuse problem in CDMA technology. Therefore, interference is not a problem unless the number of clients reaches a saturation point which disrupts the voice coding system [23].

Wideband CDMA (W-CDMA) is an evolution of GSM technologies and is the main technology for 3G cellular networks. W-CDMA is part of tree generation partnership project (3GPP), international mobile telephony (IMT-2000), and universal mobile telecommunications system (UMTS) groups. UMTS uses W-CDMA technology to perform VoD and web browsing by handsets.

Although, theoretically UTMS can support 42 Mbps with high speed packet access (HSPA) which combination of two protocols [26]: (a) high speed downlink packet access (HSDPA) for downlink, and (b) high speed uplink packet access (HSUPA) for uplink, in reality it decrease to 7.2 Mbps for downlink. This speed is fast enough to cover video streaming. Enhanced HSPA allows 21Mbps download speeds. The W-CDMA characteristics are [25]:

- ✓ One frame = 15 time slot
- ✓ Channel Bandwidth: 5 MHz
- ✓ Frame length= 10 ms or 20ms
- ✓ One time slot = 0.666ms =2560 chips
- ✓ Support both FDD & TDD modes provide
- ✓ High chip rate (3.84 Mchip/s) and data rates (up to 2Mbps)
- ✓ Support multi user detection (MUD) and smart adaptive antennas
- ✓ Provide multi-rate services using variable spreading and multi-code

Long term evolution (LTE) is based on GSM/UMTS standards. LTE offers maximum download speeds of 299.6 Mbps and 74.5 Mbps for uplink by using new digital signal processing (DSP). Different countries use different frequency bands for LTE wireless communication. Therefore, single frequency band device may not work in another country. Furthermore, unlike UTMS, GSM, and CDMA200 which use circuit switching, LTE only supports the packet switching technique. Finally, LTE can support at least 200 active data subscribers in 5 MHz cells.

Chapter 3

SYSTEM STRUCTURE AND ALGHORITHMS

3.1 Basic Idea of VoD in WLAN

In this study, we developed a typical VoD system based on WLAN and extended it over cellular wireless networks. First of all, we describe VoD in WLAN (in this case MANET), and then we focuses on WWAN. In general VoD systems are based on the client/server architecture and the video download / display rates are main factors that effect video streaming performance [1]. Downloading a video segment with minimum delay/stop is also important to achieve acceptable level of QoS. To avoid single point of failure, there are more than one media servers located behind the BS or AP.

A client connects to video server through access point (AP) to access media files. Communication between server-client is in multicast mode, whereas the communication between clients in the same WLAN is based on peer-to-peer mode (Figure 11). Depending on the number of portable devices in the wireless network, video server may need to store different copies/types of same video file. In multicasting, the client can join to a suitable multicast group. Typically, clients have small buffer size, not enough to store the whole video file. We consider a video file as a collection of equal size segments. Each segment includes some fixed-length portion of video file. So, one advantage of multicasting is the capability of client to receive video file, segment by segment in lack of huge buffer sizes.

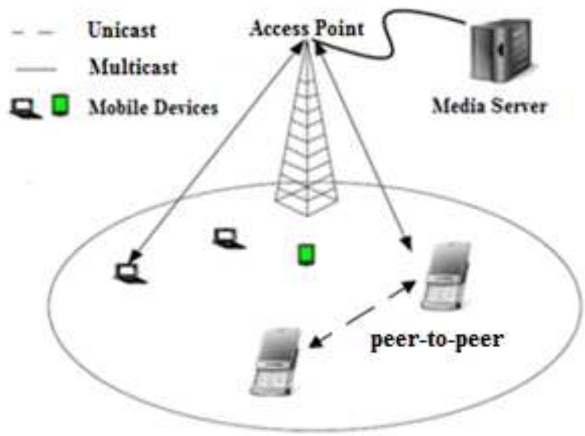

Figure 11: VoD in WLAN

Broadcast services can save free channels but different devices use different media types, so a variety of media files may lead to bottleneck in media servers. For overcoming channel allocation and server bottleneck problems, we consider scheduled multicast and patching mechanisms. According to patching mechanism, when a new client arrives to a cell, it joins to an existing multicast group for the reminder of media file, and downloads missing part of media from peers as a patching stream. The new client waits for a new multicast group if the existing multicast group does not have free channels for allocation, but it can concurrently receive patching stream from peers during the waiting time. The BS/ AP forwards client request to media server when there are no a peers which have the requested media file. If there is no free channel, the client must wait until a free channel is available. Figure 12 illustrates the multicast process in wireless networks. Also, the scenario is discussed in algorithm .1.

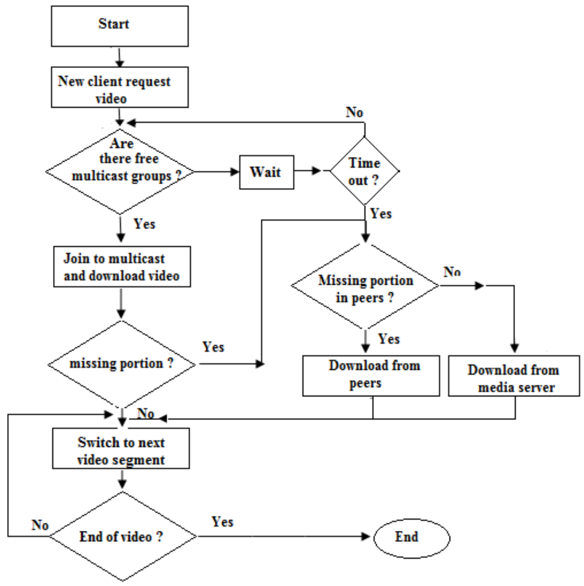

Figure 12: Media services for new client

Algorithm.1 media file services for client request

M_v : multicast group for video v, U : unicast for video v

t_s : start time of multicasting segment s

t_c : time of accessing client to segment s

Begin

Client C arrives in wireless network and requests video

If M_v exist **then** **Begin**

join to M and receive current segment

if $t_c > t_s$ ***then*** run unicast () // missing portion exit

else **while** $(s_i < s_{end})$ **then** i=i+1 // switch to next segment

end while

end

else wait for next M_v **while** time out, **then** run unicast() from peers

end

```
unicast()
Begin
        If  peers has segments of  video (v)
                then
                        download video from peers which is not busy
        else if peers busy, but responsible faster  than next broadcast time,
                then
                        wait for peer and try again
        else            download from media server
end
End.
```

In a WLAN, clients periodically send cache content information to the access point. AP stores multicast video and client's information in own database (multicast group and client information table). When a client need a video, first checks its local buffer and play the video if it is stored in local buffers, otherwise, it makes a request to AP (figures 13-14). AP cheeks its database to see if video exist. There are three cases:

Case1: If a multicast group exists for requested video, then join client to that multicast group.

Case2: If there is no multicast group, check if other peers have segments of video.

Case3: Otherwise get video from remote media server.

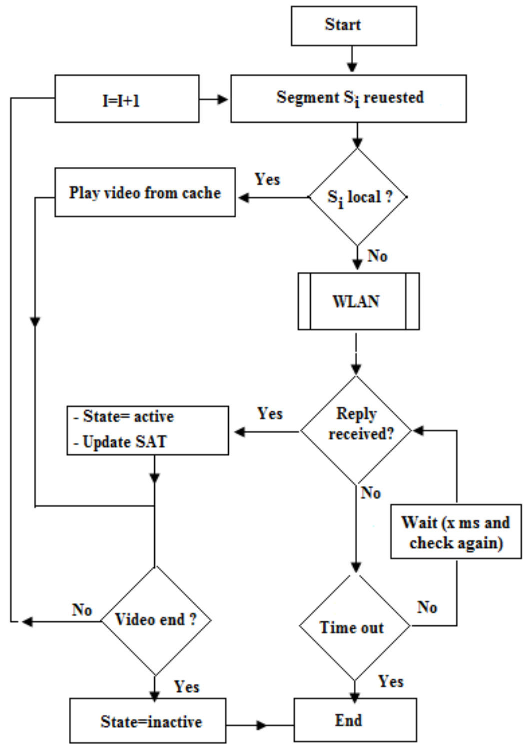

Figure 13: Client side function for requested video by WLAN clients

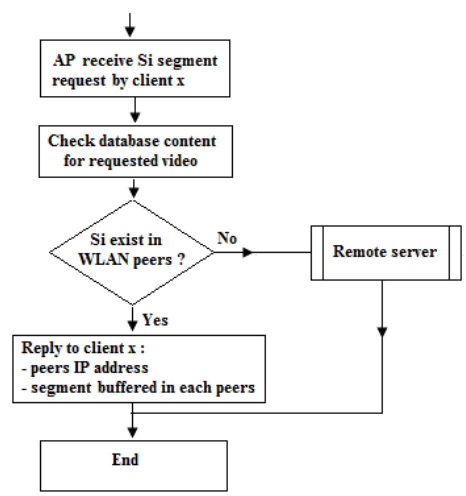

Figure 14: AP side diagram for WLAN function

3.2 Scheduled Multicast and Patching

Although broadcast mechanisms are widely used on the Internet and wired networks, it is not suitable for wireless networks [14, 18]. Routing decisions based on imprecise information due to dynamic change of network topology, traffic load and channel characteristic are the main problems [27]. On the other hand, because of their high capacity, broad coverage and broadcast nature, wireless networks are ideal platforms for multicast [15]. In general all broadcasting techniques have a start-up delay. This delay depends on the efficiency of the broadcasting scheme, channel bandwidth and receiver buffer sizes [12].

For overcoming this problem, we consider Schedule Multicast (SM) and patching over a multi-channel scheme. In the SM solution, each media file is divided into several chunks and every chunk is multicast over different communication channels. We consider scheduling multicast communication for popular video streams and P2P (unicast) connection for unpopular video streams (missing portion of video). Each mobile device uses different types of video streams, so the media server prepares a variety of media stream types. In this approach, each client joins to the most suitable multicast group and plays the media file.

Patching technique is able clients to download data from two channels concurrently. Consider multicasting start at time t_1 and assume a new client arrives in a cell at time $t_1+\theta$, and makes a video request. It can join the multicasting channels while watching the videos. Simultaneously, it receives patching video stream from peers which already have part of this video. If there is no regular multicast channel currently serving video V, a free channel is used to start a new regular multicast for video V, and peers in the selected batch receive the entire video from this new regular multicast at time t_2. In this approach, it is possible that clients belong to different cells. Unfortunately mobile devices have limited buffer size. Therefore, they can just cache a small part of video.

A video file is divided into k equal size chunks (e.g. chunk1, chunk2 … chunk k). Depending on QoS, video quality, and coding technique, transfer rate (t_r) maybe less, equal, or bigger than display rate (t_d). Assuming transfer rate and display rate are equal, the display time of each chunk is $t_{chunk} = t_d = V/k$, where V is the total video displaying time and k is the number of channels. Total bandwidth is divided in k channels and each channel multicasts repeatedly video file as a sequence of segments (s_1, s_2, s_3… s_n). For example, if video size is V=60 minutes and number of channels is limited to k=5, then t_d=V/k=60/5=12 minute.

As illustrated in figure 15, when a new client arrives at time t_0, it can join channel 1at the beginning of first chunk, start download and display segment one (s_1). The client switches to next chunk at the same channel after receiving all segments which belong to the first chunk and this scenario continues until end of video. If client arrives at time $t_0 + V/k + \theta$ where $t_0+V/k<t_0+V/k+\theta <t_0+2V/k$, rather than waiting for next multicast at time t_0+2V/k, it will receive the missing portion of the first video chunk (e.g. s_1, s_2, s_3) from peers or media server in unicast mode and simultaneously buffer the video segments from the multicast channel 2 [12]. The unicast channel is disconnected after receiving missing segments, and the client can be served by the multicasting channel 2 until the end of the video. Therefore, the patching technique can significantly reduce the resource requirements.

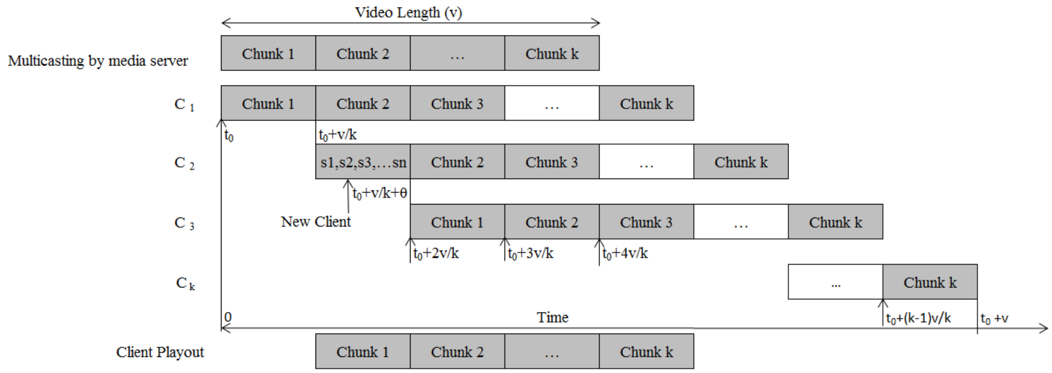

Figure 15: Scheduled multicast mechanisms

For overcoming the startup latency, we consider a scheduled/batching multicast technique instead of using periodic multicasting. In scheduled/batching multicast technique, we consider groups of same video request that arrive close in time (with a threshold) to start multicast for servicing them. Patching technique is proposed on top of batching to allow late coming clients to join a service [28]. Therefore, if the number of requests for getting video increase quickly and become more than the threshold, multicasting video in new channel will start faster than scheduled multicast time, otherwise multicasting will start at regular time. This definition decreases startup latency. In the worst case, the scheduled multicast latency is the same as periodic multicast. In popular video streaming (e.g. goals of a football match), there is more request for video at the beginning, therefore, using scheduled multicasting decreases startup latency. For example, if video length is 45 minutes and it is divided into five chunks (five channels), startup latency for the worst case is 45/5=9 minutes, but in scheduled multicasting, depending on request arrival time, it can be less than 9 minutes. This approach is greedy and consumes all channels if the number of clients grows fast. Scheduling multicast detail is described in algorithm 2.

Algorithm .2 Scheduling multicast

```
timer
t_m: multicast time
n_c: number of clients which request a video
n_s : minimum number of clients as threshold to start next multicast

Begin
t_m=n, timer=0
    while (n_c<n_s)
            timer+=1
            If   timer=tm  then
                  begin
                           start multicast
                           timer=0
                           n_c=0
                   exit
            else if    new client arrives  then
                           n_c=n_c+1
       end while
 start multicast
End.
```

Since, each portable device has a different downlink bit rate, it is hard for a multicast sender to send at a rate that is suitable to all nodes. Video streaming application must be compatible with different network conditions by increasing/decreasing the bit rate [29]. We believe, batching the same device in one multicast group improves network performance and it is easier to manage. For example handset devices have limited resources, but laptop or PC which connect to a cellular network through EDGE modem (3G) have acceptable resources such as memory and computing power. Therefore, in same condition when handset and PC buffered a requested segment, PC node has priority to join P2P connection and sends requested segments.

In our approach, every client has three main components:

- ✓ Playback buffer (playing) - video must be played in correct sequence order of segment with few interrupts or stops. So video buffering is necessary to achieve quality of display.
- ✓ Forward buffer (cache) - client caches watched segments in the cache buffer in FIFO order. The cache buffer has two main advantages: video locality and send video file to neighbor clients in P2P mode.
- ✓ Segment allocation table (SAT)- each client stores video segment information such as video name, and stored segment numbers in its SAT and announces server or other clients

Finally by using proxy server, it is possible to overcome the low downlink channel rate and we may achieve high quality of service (QoS). This approach has several advantages including:

- ✓ Channel saving- compared to the traditional VoD system in which the individual client is served by unicast channel, the patching scheme reduce the resource requirements [10].
- ✓ Handoff- it may improve hand off during move of one client form one cell to other. By limiting the number of unicast communication, there are more free channels for new clients arriving to cell.
- ✓ Video locality- it possible use proxy server behind BSC to decrease the media server load. It is not more costly because we assume one proxy server for multiple BS is instead of using an individual proxy server for each BS.

3.3 Resource Management

When a new client joins to a wireless network and detects that it lost a portion of first multicast chunk, it must wait for next multicast. For example if first multicast starts at time t_1, and client X requests video in time t_2 where $t_1 < t_2 < t_1+s$ and s is the time of beginning first segment of next multicast, instead of waiting for next multicast, it may look for existing neighbor in transmission range area which buffered the missing portion of first chunk. If such a client exists, download and play missing portion from neighbor in unicast mode and concurrently receive the remaining part of chunk from forwarder (e.g. BS or AP) in multicast mode. Otherwise, the missing portion of chunk must be downloaded from the media server in unicast mode.

The aim of our approach is minimizing individual unicast connections to the media server. Caching system at the client side improves data access latency, bandwidth consumption, reduces network traffic, and saves battery life [30]. Due to limited bandwidth, each client can forward cached segment to one client at a same time. The sending client switch to active mode during cache forwarding and stays in this mode until cache forwarding is finished. There are three cache mechanisms [16]: (a) all-cache: all clients cache the first segment,(b) random-cache: only a number of selected clients need to cache, and (c) dominating-set-cache (DSC): only the clients that belong to a dominating set of the clients cache the first segment. We consider all-cache case according to the flowing parameters:

- ✓ Hop cont: if client finds requested video in its own buffer, hop count is zero, otherwise cache distance can be one or two hops. There is at least 2-hop between the source and the destination in wireless cellular networks and infrastructure wireless networks.

- ✓ Startup overhead: client mobility speed, direction, and video size effect link failure. The client needs to re-establish connection to receive the remaining part of chunk from other clients again. Link failure increase startup overhead. In this case, all-cache is a better option than others, because of minimum hop count.
- ✓ Service delay: all-cache has better service delay than others, because of minimum hop count and maximum number of clients.

The BS receives video streams from media server and distributes it toward the mobile client. In general, P2P traffic is transmitted through the BS. Therefore, there is 2-hop count between source and destination. The limited hop count requirements minimize number of peers which have patching streams. Although increasing hop count increases peer domain which has patching streams, due to node mobility, network topology immediately changes in wireless networks and may lead to a break of connection path. So, path re-establishing delay may cause loss of some packets, which is not acceptable for a multimedia application [27, 29]. Also, any peer failure immediately affects the quality of service (QoS) [4].

It is possible that a mobile client downloads cached file from more than 2-hop neighbors, but wireless resource specification such as small bandwidth and low power life time, complicate transmitting big video files over multiple mobile devices. Although using compressing technique decreases the video size, in some cases (e.g., in medicines) we need high quality video. Therefore, we don’t recommended using P2P communication in cellular networks.

In multicasting technique, the downlink bandwidth (server-to-client) is bigger than uplink bandwidth (client-to-server) side. Therefore uplink bandwidth is more expensive. A wireless cellular network covers a wide area, but the limited bandwidth causes unicast communication to be costly. On the other hand, WLAN supports a small coverage area, but it has more bandwidth and is also cheaper than cellular networks.

In real time application clients may need to pause, fast forward/backward, and jump forward/backward. Playing video segments are suspended during pause operation, but client can keep on buffering video segment from current multicast channel. In terms of Video Cassette Recording (VCR) interaction, the client has to switch to other multicast channels if the required contents cannot be found in the buffer. The resource with best quality and low cost will be select if there is more than one resource (e.g. channel or client) available. If requested segments of video are not available in current multicast channel or neighbor peer, service is required from the media server in unicast mode. More cache memory in client side improves VCR interaction. Although VCR functions are well supported in WLAN because of more cache memory, under limited buffer size, in cellular wireless network it is difficult to guaranty continuous VCR functions. Therefore, buffer management is also a big challenge in this area.

3.4 Architecture of our Proposed System

All former Periodic Broadcasting techniques try to reduce service latency and minimize resource requirements. Although broadcasting technique improves server bottleneck by reducing the number of I/O requests and decreases usage of backbone bandwidth, all of the techniques focused on the server side. A wireless cellular network is a heterogeneous network in which, each device may have deferent capability. In scheduled multicasting technique a video file is multicasting over multiple channels, the client must be turned on each channel to receive each video segment. In practice, it is difficult to concurrently fetch video from multiple channels. Assigning high reception bandwidth for the client may solve this problem, but this has high cost. Therefore, a practically multicasting video file into many channels is impossible.

As shown in figure 16, the architecture of our proposed system infrastructure includes:

1) Media servers- which store media files and multicast them periodically
2) Public switching telephone network (PSTN)- which connects the cellular network to the media server
3) Proxy servers (PS)- located in BSC between media servers and BSs
4) Base station controller (BSC)- which stores all client information
5) Base station (BS)- which supports clients in the coverage area
6) Mobile station (MS)- which connect to BS and request and receive media files

The media server periodically multicast video files through multiple channels and mobile clients receive media files through cellular networks by switching between channels. With video segmentation and multicasting video segments over multiple channels we try to reduce the startup latency. Bandwidth problems, channel allocation and handset capability increase complexity of network design.

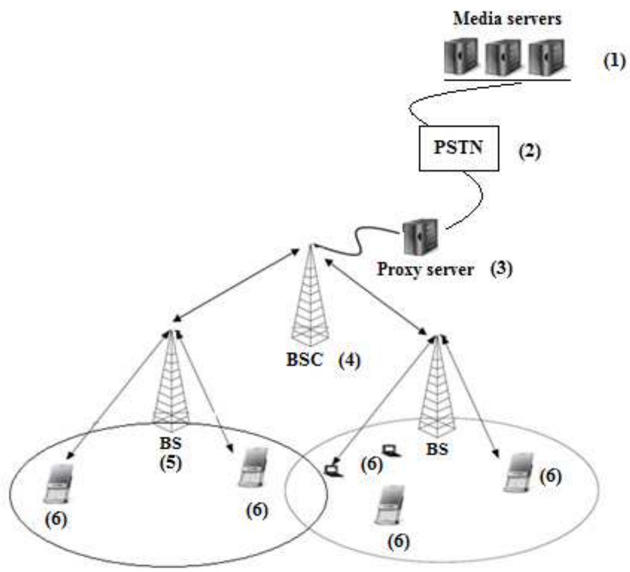

Figure 16: Architecture of Proposed System

The Proxy Server (PS) content depends on the buffering size (memory size), popularity of requested video files and media server multicasting. The PS receives a broadcasted video file, buffers it and may multicast it according to its own policy. Although installing PS on BS is possible, it will have high cost. We set up a proxy server in BSC area. In cellular networks all BSC's know the network topology. Also they are located at a fixed position and have high percentage of resources such as memory, computing power, and battery life. All P2P communication in the cellular network is routed through BS, P2P communication is costly because it causes loss of energy and more delays. Algorithm 3 illustrates the BS mechanism for our architecture.

Algorithm .3 Communications between client and BS

Threshold member: number or user required for starting multicast

Threshold time: waiting time for starting multicast

Begin

All clients periodically send own information such as energy, buffer size, distance, and access rate to BS.

BS creates / updates network topology information.

When client request video v ***begin***

If multicast v available, ***then*** Join client to most suitable multicast M_v group,

Else BS checks it database

If there is candidate peers (MS) with cached segment exsit, ***then***

P2P communication initializing in unicast mode through BS

// initial startup latency is very low, cost is low

Else, (video is not in wireless cell) ***then***

Send client request to proxy server (PS) located in BSC.

If the requested video is present in PS or neighbor PS in same BSC

begin

Insert new client to multicast group

If *number of client in multicast group* **>=** *threshold members* **then**

start multicast

Else wait until *threshold time* and start multicast

end // initial startup latency is medium, cost is low

Else if the requested video is not present in the PS, ***then***

Requests send to media server

```
                    //     initial startup latency is high,   cost is high

                end

   End.
```

By installing a proxy server in BSC, channel management capability increases widely. Due to proxy server attributes, it is possible to divide each chunk of video into sub chunks. On the other hand, we can divide each channel into some logical channels. Start up latency decreases by multicasting each sub chunk through logical channels.

Video file reaches the proxy server when it is multicast by media servers. Proxy server checks own content and stores the recent video file if it is not buffered before. Simultaneously, it forwards video file to client through dedicated channels. Proxy server manages video chunk based on bandwidth and number of videos multicast. By increasing the number of multicast videos, it is necessary to assign new channels for each video file, unless, current channels can be share among all videos. If number of videos is increased to N_v =5, then startup delay will be:

$$\text{Delay} = V * N_v / k \tag{9}$$

(Where all videos have an equal size)

Resource limitation is one of the big challenges in cellular wireless networks. Patching mechanism in multicasting technique decreases the battery power to help this. Clearly increasing buffer size in client side is not only costly, but also leads to loss of battery power because of the need for transferring more segments to other peers. Energy is also lost related to the number of user arrivals into cells or departures from cells. Providing a sub-channel/segment mechanism over SM by using a proxy server leads to improving waiting time and has a good effect in energy consumption in mobile device. Mobile devices loss battery power by: (a) downloading video from BS, and (b) P2P communication through BS. The number of missing segments which must be received from peers directly relates to the multicast period. Increasing the number of channels or sub-channeling improves startup latency. Therefore new client need to receive a few segments from other peers.

One of the main advantages of caching media files in client side or proxy server is reduced access time through locality. Ability to receive media files from local resources instead of receiving from remote servers decreases access time. Also caching decreases network traffic. Looking for media file in neighbor clients is an effective way to improve network performance for minimizing access time. Although looking for missing portion of requested video in other peers is easer in LAN and WLAN networks, it is not suitable for cellular clients because of the following reasons:

- ✓ In cellular wireless networks all P2P communication is supported by BS. All send or received pass through the BS. This increases the number of hops, so round trip time (RTT), and delay increases too.
- ✓ Typically downlink speed of mobile client is faster than uplink speed. Therefore video uploading is time costly.
- ✓ In general all mobile clients have low resources such as memory and power.
- ✓ Mobile client losses more energy during upload time.

Using video segments stored in BS buffers, and installing proxy server in BSC instead of looking for missing segments in peer's buffers, improve access cost, because of video locality. There are four cases for VoD in cellular networks with ignoring P2P connection. It indicated clearly in figure 17 and algorithm 4.

Case 1: the requested video is in local buffer.

Case 2: the requested video is in BS caches.

Case 3: the requested video is located in proxy server.

Case 4: the requested video is located just on media servers.

Algorithm .4 Request for video or missing segments

Begin

Case 1: If requested video stored in local buffers **then** display it immediately

    else *go to case 2*

Case 2:

        If requested video is currently multicasting now ***then***

            Join client to multicast groups and

            looking for mission segments // video locality

        else go to case 3

Case 3 *:*

        If requested video exists on proxy server ***then***

            return video form proxy server

        else go to case 4

Case 4:

        If requested video exist on media servers ***then***

return video form media servers

else

return "requested video is not find "

End.

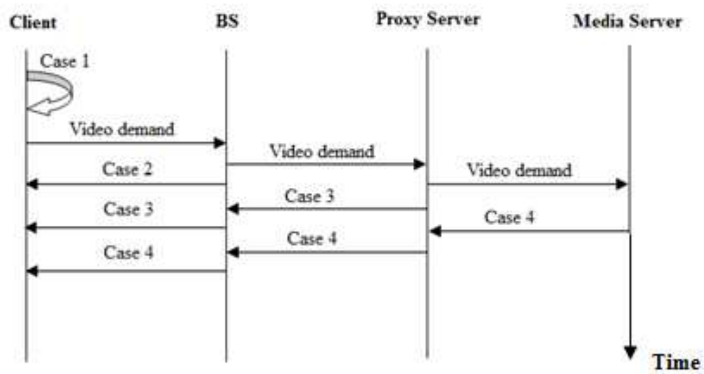

Figure 17: Video on demand in proposed system

When a client creates a video request, in case 1, the time for displaying video is nearly zero. It is because requested video is stored in local buffers. According to Table 2, total time for sending request and returning response in other cases are described (without more details) as follows:

Table 2: Characteristic of proposed system

T_r	Channel (link) transmission rates
T_s	Time for transferring requested video
T_{C2B}	Time to travel request from client to BS
T_{B2C}	Time to travel request from BS to client
T_{B2B}	Time to travel request between two BS
T_{B2BSC}	Time to travel request from BS to BSC
$T_{C\ process}$	Processing time in client
$T_{BS\ process}$	Processing time in BS
$T_{BSC\ process}$	Processing time in BSC
$T_{BSC\ 2\ M\ server}$	Time to travel request from BSC to media server
N_b	Number of BS between source and destination

T_{P2P}=T_{C2B} + $T_{BS\ process}$ +T_{B2C} + $T_{C\ process}$ +$T_s/T_{r\ uplink}$ + (N_b -1)*(T_{B2B} + T_s/T_r) +T_s/T_r downlink

Where $T_{C2B\ =}$ T_{B2C} then

$$T_{P2P}=2T_{C2B} + T_{BS\ process} + T_{C\ process} + T_s/T_{r\ uplink} + T_s/T_{r\ downlink} \quad (10)$$

$$T_{case\ 2}=T_{C2B} + T_{BS\ process} + T_s/T_{r\ downlink} \quad (11)$$

$$T_{case\ 3}=T_{C2B} + T_{BS\ process} + N_b * T_{B2B} + T_{proxy\ process} + T_s/T_{r\ downlink} + N_b * T_s/T_r$$

$$T_{case\ 3}=T_{C2B} + T_{BS\ process} + N_b (T_{B2B} + T_s/T_r) + T_{proxy\ process} + N_b * T_s/T_{r\ downlink} \quad (12)$$

$$T_{case\ 4}= Tcase\ 3 + T_{BSC\ 2\ M\ server} + Ts/Tr_{\ M\ server}$$

$$T_{case\ 4}=T_{C2B} + T_{BS\ process} + N_b (T_{B2B} + T_s/T_r) + T_{proxy\ process} + N_b * T_s/T_{r\ downlink} +$$

$$T_{BSC\ 2\ M\ server} + Ts/Tr_{\ M\ server} \quad (13)$$

When a client requests a video, some parameters including network load, content availability and distance are used to decide and find suitable media servers. Increasing the number of requests leads to bottleneck and storage (I/O) problems in the media server. In our proposed system, proxy servers are between the client and the media server and act as content delivery networks (CDN). CDN technology is especially well suited to stream audio, video, and Internet television (IPTV) programs [31]. Distributed proxy servers in multi BSC is used to improve start up latency by forwarding clients requests to closer proxy servers. Proxy server replies with content in multicast mode when requested videos arrives at time closer, otherwise, the video file is sent in unicast mode. This approach has several advantages including:

- ✓ Improve core network traffic by limiting video request from media servers
- ✓ Reduces startup latency by caching content which is closer to clients
- ✓ Reduces bottleneck and I/O problem in media server
- ✓ Content redundancy (backup) by caching content
- ✓ Reduces media server overload

3.5 Buffer Content Management

In general, more servers and storage are available in media provider centers, but a few proxy servers with limited resources are available in BSC. Due to the static content of video files using proxy server in BSC is an effective way to reduce request response time and network traffic. Therefore, buffer management in proxy server is an important issue. Two cache content replacement policies can be considered:

a) single-factor cache content replacement policy
   - ✓ Based on size or cost
   - ✓ First-in-first-out (FIFO)
   - ✓ The least recently used (LRU)
   - ✓ The least frequently used (LFU)

b) multi-factor cache content replacement policy
   - ✓ Greedy dual size (GDS) based on cost/size

Two types of video files (large and small) are stored in media servers. Large video files are used less than small video files. For cache content replacement, we consider a combination of factors such as cost, life time, popularity, and frequency of use as discussed below.

$$\text{Cache replacement} = \text{Cost}.\, \alpha + L + P \qquad (14)$$

$$\text{Cost} = \frac{Size}{V} \qquad (15)$$

$$L = \text{least recently used} \qquad (16)$$

$$\text{Age} = t_{cur} - t_{sit} \qquad (17)$$

$$P = \frac{R}{Age} \qquad (18)$$

Table 3: The characteristic of cache replacement

Age	The staying time of the video in cache (life time)
t_{cur}	The current time
t_{sit}	The first time vide sit in buffer
R	Number of requesting video during life time
Size	Proxy server memory size
P	Video popularity
V	Video size
α	Average size of video files

Figures 18-19 illustrate buffer discipline for an example with seven different video files. In this example, total buffer capacity is 100 units. A new video is stored in buffer if there is enough free buffer space, otherwise, the buffer management policy removes one or more videos from buffer to create free space for the new video file. Three methods that can be considered are:

- ✓ Based on size- larger file remove from buffer
- ✓ Based on frequency- least used file remove from buffer
- ✓ Based on FIFO order- first arriving file remove from buffer

Our proposed approach combines all above parameters and calculated a replacement factor. The video file with the smallest value is removed from buffers. The small size videos or videos which are requested more have more chance to stay in cache. In the example, when new video (e.g. video 8= 13 unit) arrives, video 5 is removed from the buffer and the new video is stored in buffer. We simulated our algorithms and compared with FIFO and LRU technique. Efficiency of our algorithms is better than both of them. Programming code in C++ is available in appendix.

Buffer Replacment Discipline

	Size	Popularity	Least requency used	Life time (FIFO)	My approach
Video 1	12	5	12	20	7.85
Video 2	14	6	2	2	11.00
Video 3	7	3	5	3	9.67
Video 4	17	4	3	4	8.75
Video 5	19	1	3	5	7.80
Video 6	3	20	6	6	11.33
Video 7	28	25	3	7	11.00

legend	
video size	
popularity	
frequency	
life time	
My approach	
buffer size	100
α	14.29

Assigned buffer	100%
Free buffer space	0%

Figure 18: Buffer contents before the arrival of new video (video 8)

Buffer Replacment Discipline

	Size	Popularity	Least requency used	Life time (FIFO)	My approach
Video 1	12	5	12	21	8.34
Video 2	14	6	2	3	10.19
Video 3	7	3	5	4	9.53
Video 4	17	4	3	5	8.93
Video 8	12	1	1	1	9.53
Video 6	3	20	6	7	11.24
Video 7	28	25	3	8	11.03

legend	
video size	
popularity	
frequency	
life time	
My approach	
buffer size	100
α	13.29

Assigned buffer	93%
Free buffer space	7%

Figure 19: Buffer contents one second after the arrival of new video (video 8)

Chapter 4

SIMULATION RESULTS

In wide area communication, data segments pass through intermediate gateways or routers. In general segments are routed by routers without fragmentation. As illustrated in figure 20, Oversize segments are fragmented by router and router drops a segment if it cannot be fragmented (figure 20, case 2). Segment resizing increases network delay. The maximum segment size (MSS) is the largest amount of data that every communication network device can support.

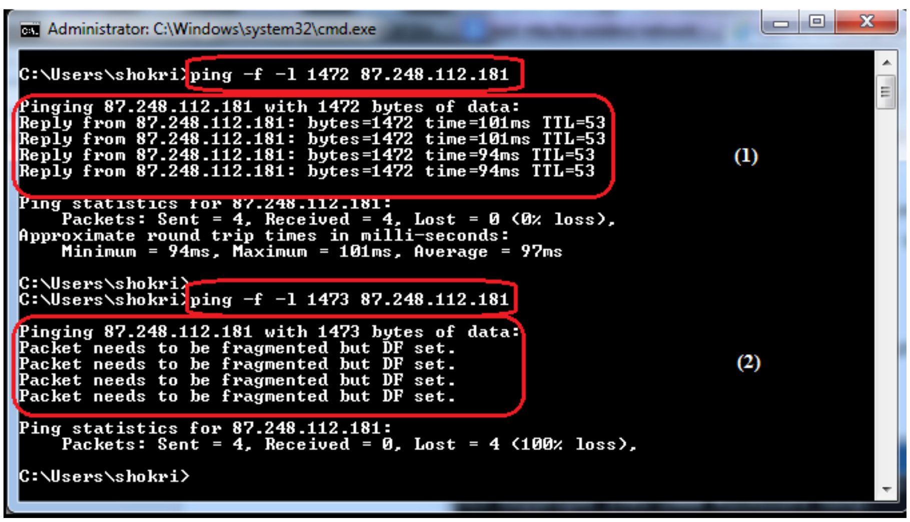

Figure 20: Maximum Segment Size (MSS)

In general MSS is between 536-1472 byte. By increasing segment size from 536 bytes to 1472 bytes, communication efficiency increases, but a segment size bigger than 1500 byte decreases efficiency. The MSS increases to 1500 bytes when we adding 28 byte IP/ICMP overhead. Therefore, by assigning more than 1500 byte to MTU decreases connection efficiency. In this study we consider MSS as 1500 bytes.

Table 4: Simulation parameters for comparing speed up in multi-channeling

Parameters	Value (s)
Video size	15 s
Video packet size	1500 bytes
Make a video request	5 ms
Buffering video in client side	1s
Prepare video by proxy server	10ms
Prepare video by media server	15ms
Packet transfer time from BS	5 ms
Packet transfer time from BSC	10ms
Packet transfer time from the core network	50 ms

4 .1 When Transfer Rate is Less than Display Rate

According to the information shown above, transmitting a one second video file (one second video size=912 kb) in a GSM cellular network which supports 22.8 kb/s transfer rate, takes 40 second (912 kb /22.8 kb = 40 s). The multi-channel approach overcomes this shortcoming and improves the bit rate. In the multi-channel technique, video is multicast over more than one channel concurrently and the client switches between channels and receives the video segment (figures 21-25). Increasing number of channels lead to higher efficiency. The client must have the capability to switch between different frequency bands. Also, the number of channels must be limited. In this example we consider five channels as one multiplexing group. Therefore, time for buffering video in playback buffer is reduced to 8 seconds (912 kb /22.8 kb *5 = 40/5= 8 s).

S =total video segment

T_s= time for receiving one segment

MSS= 1500 byte (1472 +28)

V=912 kb

K=5

$$S = V/MSS = (912 * 1024) / (1472 * 8)$$

$$= 79.3 = 80 \text{ segment}$$

$$T_s = 1500 * 8 / 22.8 \text{ kb/s} = 0.5 \text{ s}$$

Figures 19-23 illustrate multicasting one second video file on five channels concurrently. Client switches between channels and receive video segments. Each figure shows channels status in different time scale (e.g. in 0, 0.5, 1, 2.5, and 8 second).

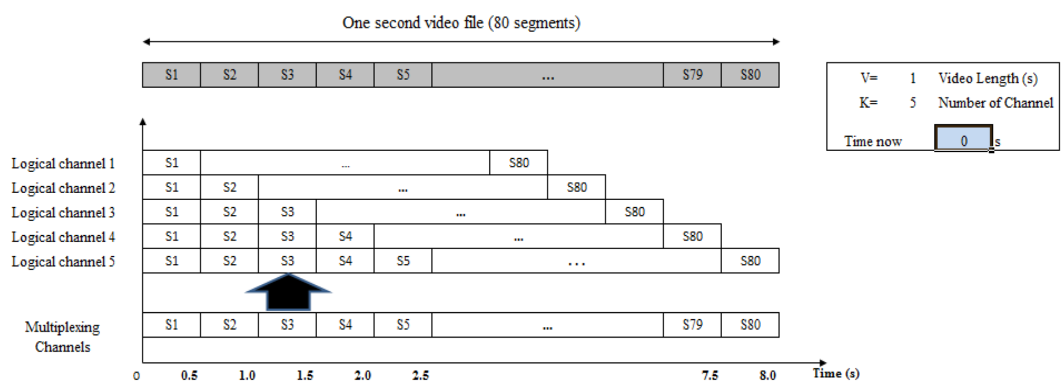

Figure 21: Multiplexing channel/multicast, time= 0.0 s

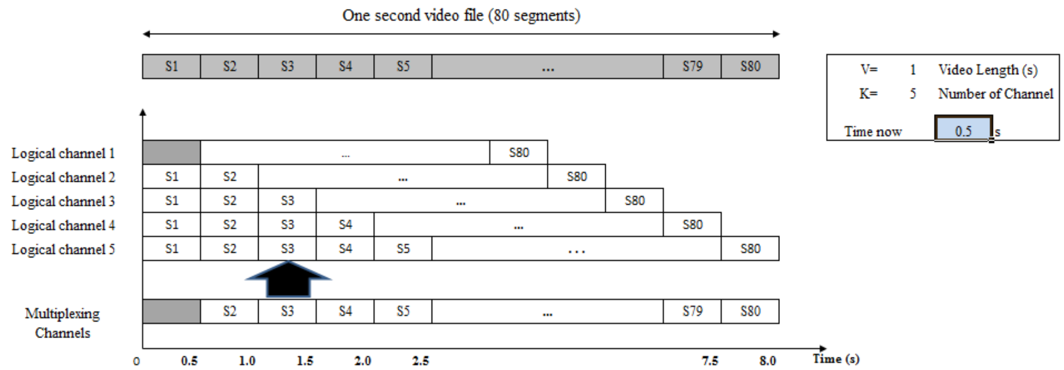

Figure 22: Multiplexing channel/multicast, time= 0.5 s

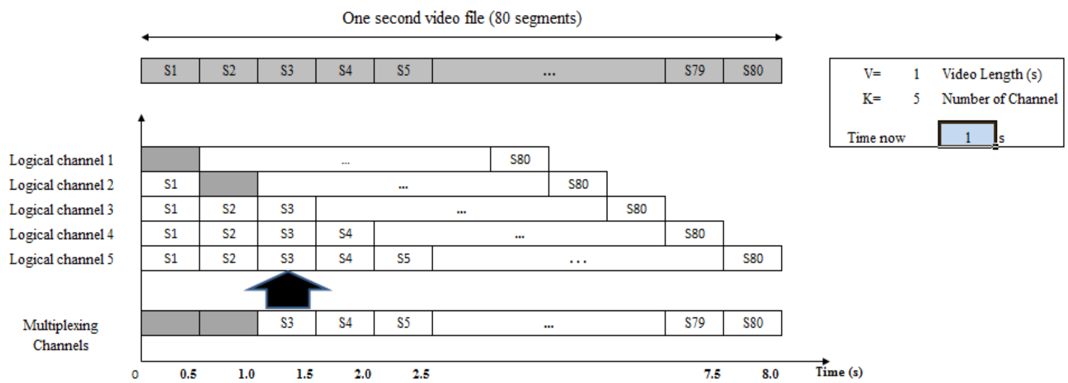

Figure 23: Multiplexing channel/multicast, time= 1 s

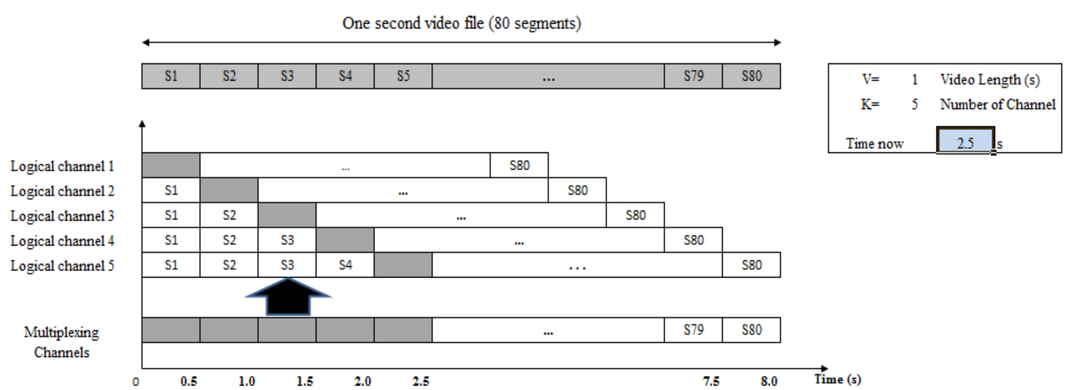

Figure 24: Multiplexing channel/multicast, time= 2.5 s

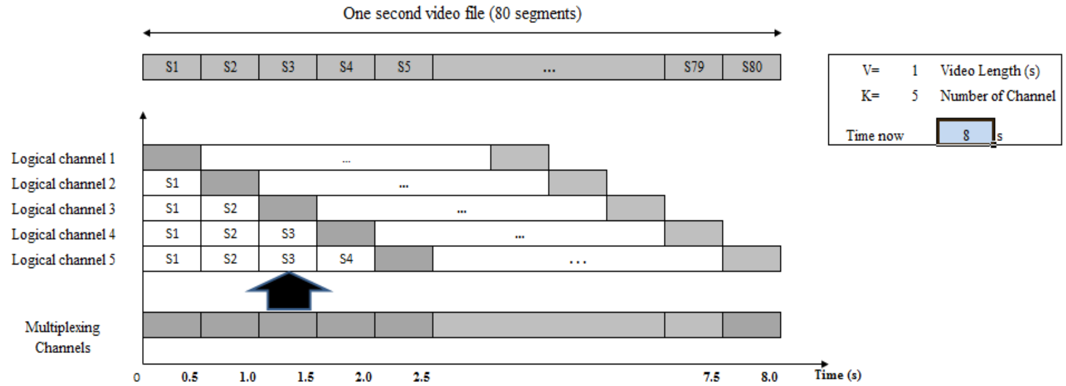

Figure 25: Multiplexing channel/multicast, time= 8 s

This technique works well, but waiting time for next multicast over multi-channel increases by increasing video size. In this example, next multicast must be started after 8 second.

According to the SM definition, startup delay is D=V/k (waiting time for next multicast). It is because all of previous work consider same speed for transferring and displaying (t_r=t_d). Unfortunately, in GSM cellular networks transfer rate is less than display rate. Therefore, GSM does not support QoS for real time traffic because of more delay (waiting more time to buffering video and then start playing them). Theoretically, in this example, increasing number of channels to 40 is means receiving all segments of video during one second (912 kb/ (22.8 kb *40) = 1). In this case, we need 8 multiplexed channels, with each of them containing 5 channels, known as sub-channel. This approach is more costly. Furthermore, switching a client between 40 channels is practical impossible. Finally, it is greedy and consumes more resources such as the frequency band.

4 .2 When Transfer Rate and Display Rate are Equal

In 3G networks, download speed at least is equal to1.5 Mbps (t_r=1.5 Mbps). Assume a one second video file in normal MPEG-1 format is 1.5 Mb (1.2 - 1.5 Mb). If video size is V=15 seconds and transfer rate (t_r) is equal to display rate (t_d), it takes 15 second to receiving whole video file through one channel. By multicasting the video file over three channels, the startup delay in worst case will be 5 seconds.

Figures 26-30 show multicasting one video file (v=15 s) over three channels (k=3) at different time. As shown in following figures, second and third multicast starts periodically after 5 (delay=v/k=15/3=5 s) and 10 seconds. Each figure shows channel status in different time scales (e.g. in 1, 6, 11, 15, and 20 s).

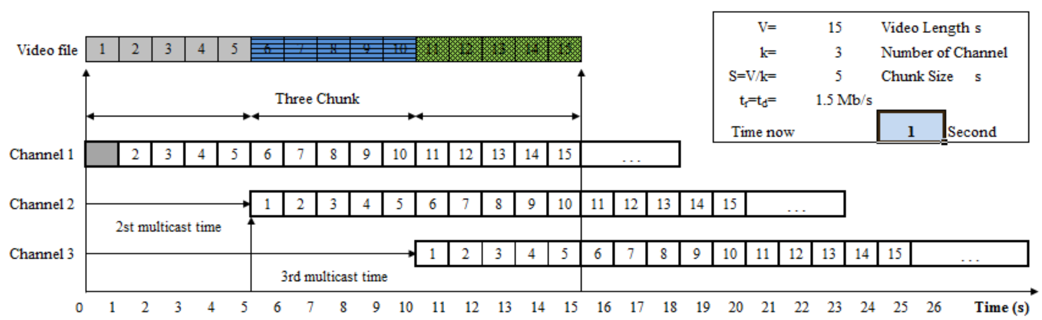

Figure 26: Multi-channel/multicast implementation (1st multicast, time =1 s)

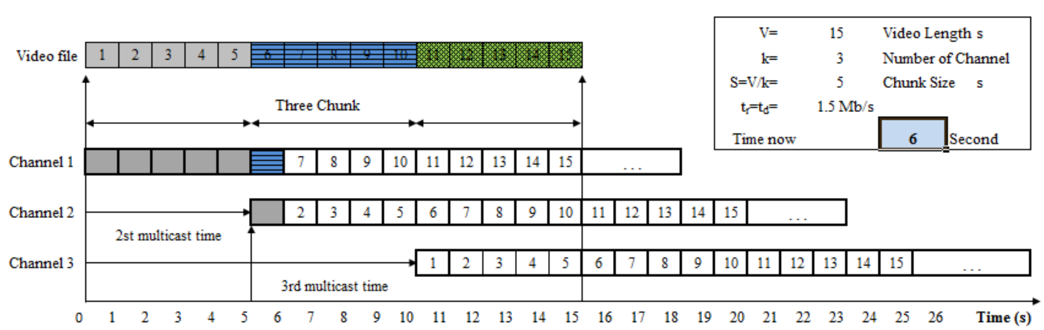

Figure 27: Multi-channel/multicast implementation (2nd multicast started, time=6 s)

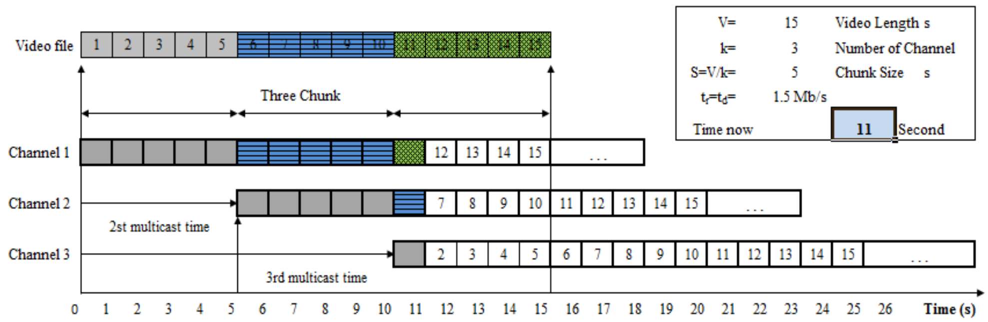

Figure 28: Multi-channel/multicast implementation (3rd multicast started, time=11 s)

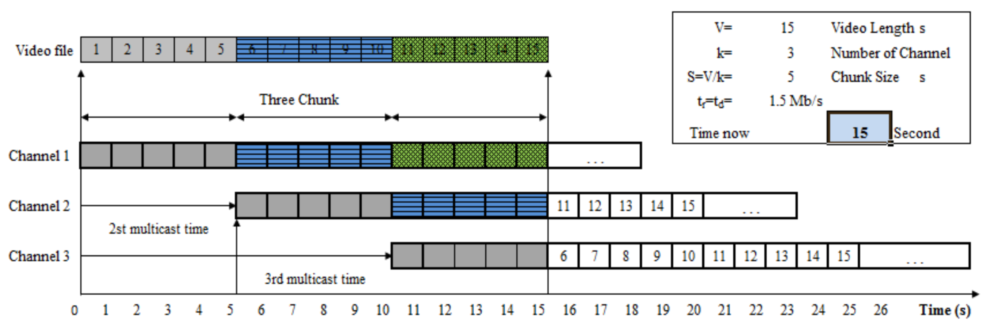

Figure 29: Multi-channel/multicast implementation (1st multicast finished, time=15s)

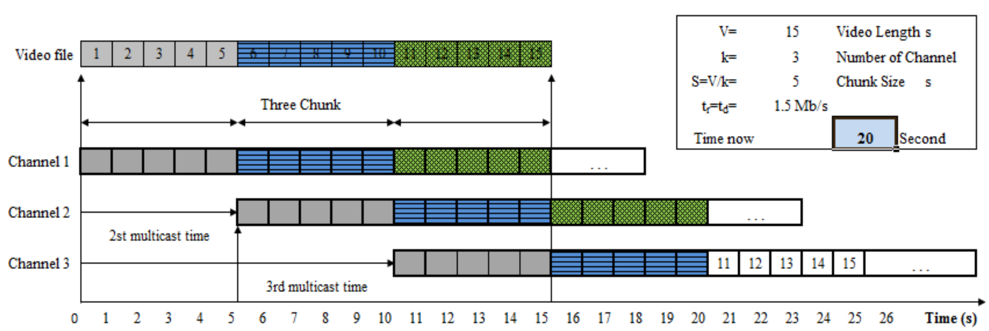

Figure 30: Multi-channel/multicast implementation (2nd multicast finished, 3rd multicast continuing, time=20s)

When client $C_{1,}$ joins to network at time =0 and requests the first segment of the video, it immediately receives and displays segments of chunk 1. Then, it switches to the next chunk until end of video. Under this condition, there are two options for client C_2 when it joins to network at time=$t_0+\theta$, where $0<\theta<1$, and requests the first segment. Firstly, it can wait for next multicast, and secondly it can join to current multicasting for remain part of video and simultaneously receive missing portion from: (a) other peers, (b) base station, and (c) proxy server.

4.3 When Transfer Rate is More than Display Rate

Beyond 3G technology supports high data rate communication. This type of network provides high speed downlink for mobile client. Therefore, mobile client can enjoy high QoS multimedia services. Consider a 7.5 Mbps downlink speed (t_r) for transferring a five second video file with MPEG-1 (t_d=1.5 Mbps) format. In this case, the whole video file can be downloaded in one second.

$$5*1.5 \text{ Mbps}= 7.5 \text{ Mbps}$$

Therefore, we can download a 75 second video file during 25 (75/5=25 s) second. So, next multicast will start 25 seconds after first multicasting. In this example startup latency in worst case is 25 second. Increasing number of channels (multi channeling) improves startup latency, but decreases channel efficiency. Also, channel assignment is more costly. Therefore, we propose a multi channels/multi video approach beside video locality technique to reduce waiting time by storing video in proxy server and BS side.

Figures 31-34 illustrate multicasting of five videos over three channels. By dividing video file over three channels start up latency or delay for next multicast is D=V/k=75/3=25. As described before, a five second video file can be transferred in one second. Therefore, 25 second video can be downloaded in 5 seconds. During next 20 seconds the channel is idle. If there is no other video file in media server for multicasting, next multicasting will start immediately. Otherwise, the channel is assumed to other files. For best results, it is recommended to divide each chunk of video file in sub chunks. For example we can multicast five sub chunks in five different time slots, instead of multicasting the whole 25 second video stream at once.

For example, consider client X joins to network at time =6 and requests the first segment of video one. Rather than waiting for next video multicasting in channel 2 at time=25, the client joins to channel 1 and downloads the remaining part of video, and concurrently requests the missing portion from proxy server or BS.

Figures 31-35 show multicasting five video files (V=75 sec for all) over three channels. Each video accesses the channel at round robin manner and sends data. Scheduling is done depending on video length, video quality and transfer rate. Assume each video has same quality. First video accesses the channel during five seconds (t_r/t_d = 7.5/1.5 = 5 s). The client joins to channel one and receives data during five seconds and plays video for 25 seconds. In this case, a five second time slot is assigned to each video. A small time slot decreases delay, but it has more overhead.

The second and third multicast start after t=25 and 50 seconds. Depending on arrival times a client joins to suitable channels and receives data. Following figures show channel status in different time scales (e.g. in t=5, 15, 25, 50, 100 s).

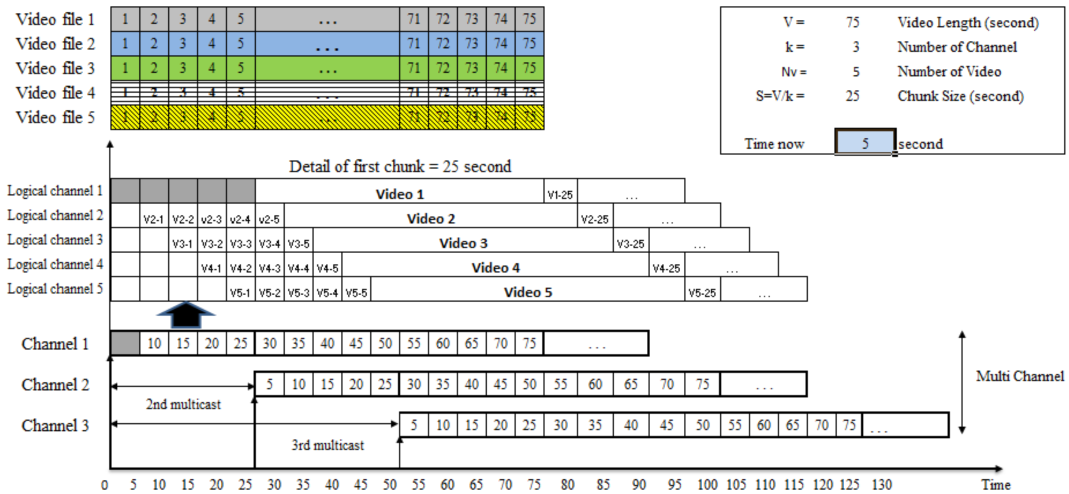

Figure 31: Multi-channel /multi-video implementation at time=5 s

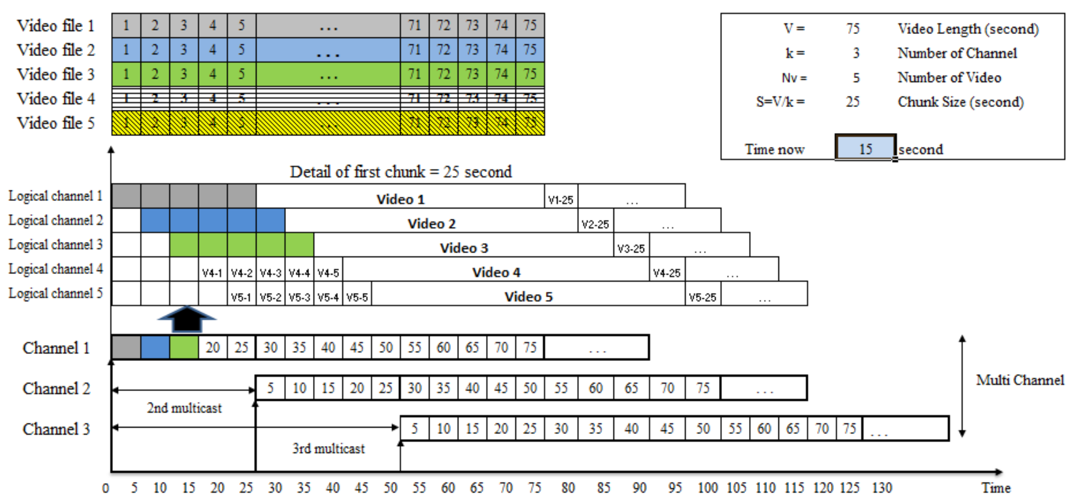

Figure 32: Multi-channel /multi-video implementation at time=15 s

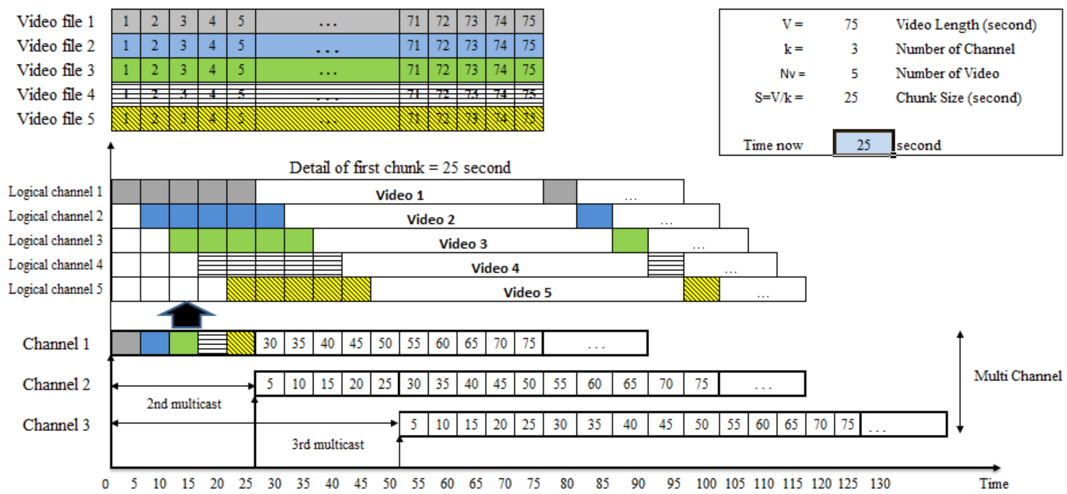

Figure 33: Multi-channel /multi-video implementation at time=25 s

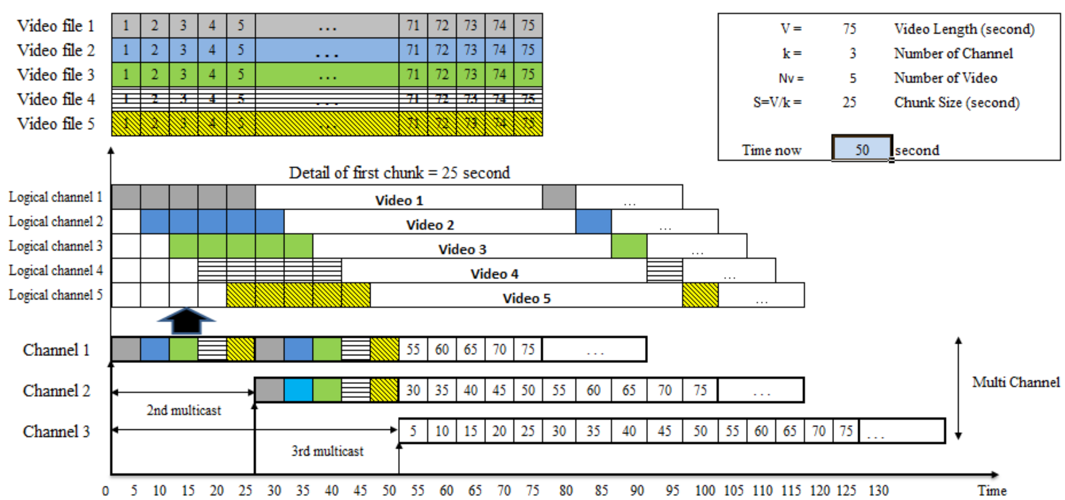

Figure 34: Multi-channel /multi-video implementation at time=50 s

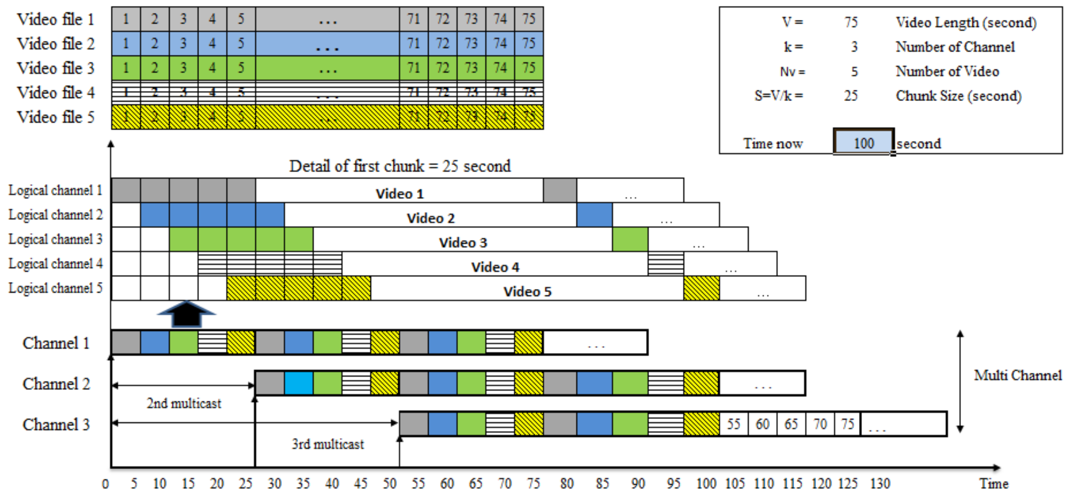

Figure 35: Multi-channel /multi-video implementation at time=100 s

In wireless networks channel quality and QoS changes time to time. This is because of interference, fading, jitter, client movement and cell density. Signal inference between two or more frequencies decreases channel qualities. Therefore, interferences cause a decrease in data rates. Figure 36 illustrates differences of two cases. Two P2P communications (communication 1 and communication 2) are started at the same time. Both P2P connections has same configuration (1Mb/s, packet size 512 kb, number of packets 250, inter arrival time 0.004s), but each of them work in different frequency band. As shown in figure, before t =5 minutes both communications have slower data rate (send/receive). After finishing communication 1, at t=5, the throughput in communication 2 is increased.

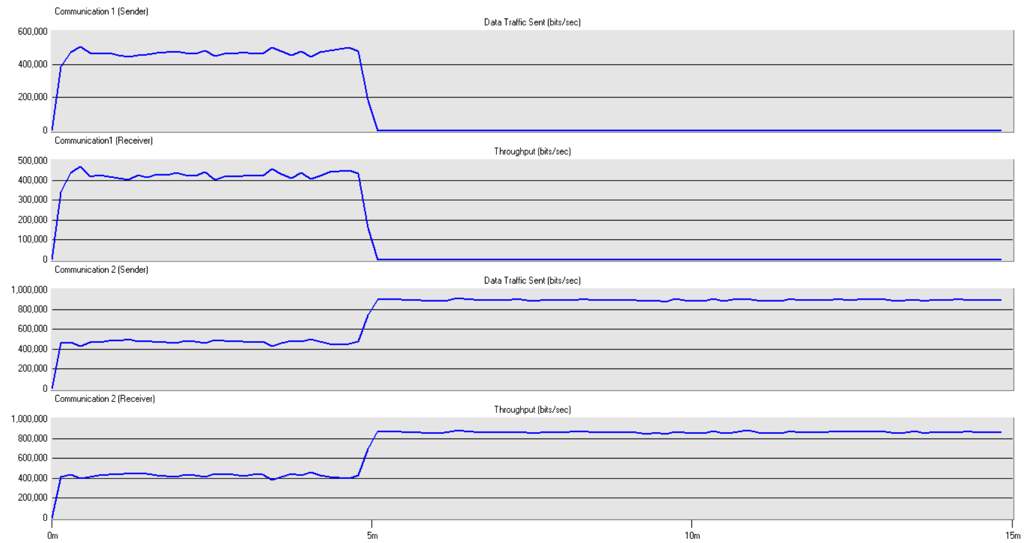

Figure 36: Effect of interferences in wireless communication

In cellular networks, channels have minimum interference with each other. Therefore, video can be multicast over multi channels and a client can receive a video file by switching between channels.

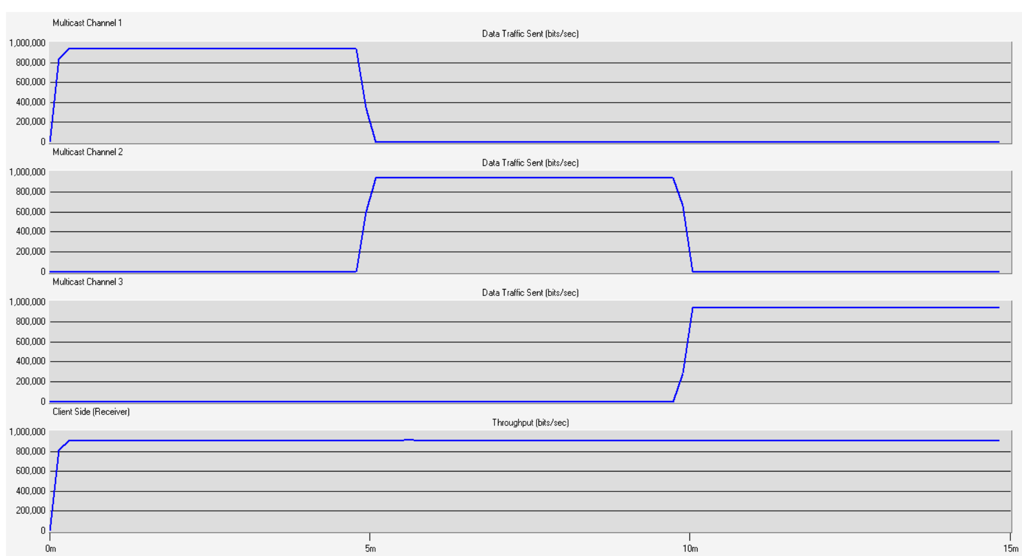

Figure 37: Result of multicasting video over three channels

Figure 37 shows a case for video multicasting over three channels. There are two ways for multicasting video over multi channels. (a) Whole video is multicast on one channel. Second multicast starts after d=V/k. Client joins to channel and receives the whole video segment by segment. (b) Depending on the number of channels, the video file is divided into multi chunks and each chunk is multicast over one channel periodically (SB). In this case, clients need to switch between channels to receive video segments.

In multi-channel/multicast technique a client receives data by switching between channels. As illustrated in figure 38, by increasing the number of channels, throughput decreases. This is because of switching overhead and channel interference. Therefore, multicasting over many channels is not recommended. Practically, a multicast over two or three channel leads to better results.

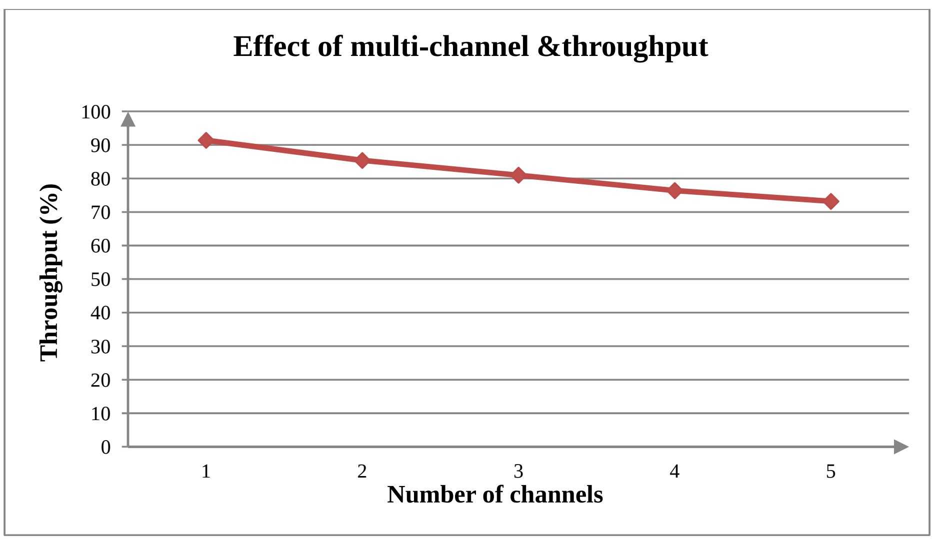

Figure 38: Effect of multi-channel and throughput

QoS at the end user site depends on channel capacity, bandwidth, and client buffer size. If a client does not have enough buffer space, incoming video packets will be lost. Furthermore, lower waiting time for buffering incoming packets in client buffers increases the number of stops during video playing, so QoS decreases.

Packet size is the other attribute that effects data rates. Increasing packet size up to 1472 bytes increases the data rate. Figure 39 shows the results of changing packet size and the number of clients that use share channels.

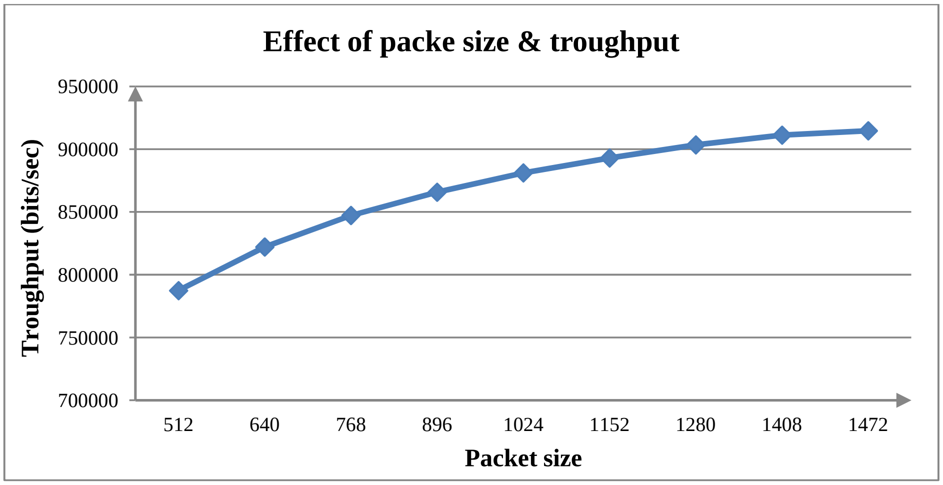

Figure 39: Effect of packet size and throughput

Time between the video request and the display of video is defined as the response time. Increasing number of interrupts during display time decrease seeped up. Therefore, speed up is effected by channel quality and data rates. Three cases may be considered here:

- ✓ Worst case- transfer data rate is less than display rate ($t_r < t_d$)
- ✓ Average case- transfer rate is equal with display rate ($t_r = t_d$)
- ✓ Best case- transfer rate is more than display rate ($t_r > t_d$)

Second generation cellular networks such as GSM is an example of worst case. These type of networks do not support high data rates. Providing multiplexing channels for worst case improves speedup, but it is more costly and consumes more resources.

Main advantages of multi-channeling technique are the decrease of startup latency and reducing the size of missing portion of videos. Assume that a one second video file coded with MPEG-1 format includes 128 segments. If length of video is V=15 seconds, it is multicast over one channel and some client X arrives after six seconds (t=6) and makes a request for the video. It joins the multicast channel and downloads the remaining part of video. Also, it needs to receive the missing portion of video (6*128=768 segments) from other peers or BS. Broadcasting same video over three channels improves startup latency and decreases the number of missing segments. It is because video length is divided into three equal size chunks. Therefore, when client X makes a video request at time t=6, it joins the second multicast channel which started one second before (chunk size 15/3=5 s) and receives the remaining part of video, simultaneously looking for 128 (1*128=128) missing segments from other peers or BS. Table 5 shows the effects of multi-channeling on startup latency.

Table 5: Average of startup latency in different channels

Video size (Second)	Total Number of segments	Average of startup latency (s)		
		One channel	Three channels	Five channels
15	1920	7.5	2.5	1.5
45	5760	22.5	7.5	3.5
75	9600	37.5	12.5	7.5

As shown in figure 40, increasing the number of frequency bands decreases delay. In this way, a multi channel approach will be capable of supporting more clients that make the same request. As mentioned in algorithm 2 in section 3.2, new multicast will be started if all existing channels are saturated and requests for the same video (in small period of time) reach the threshold request number.

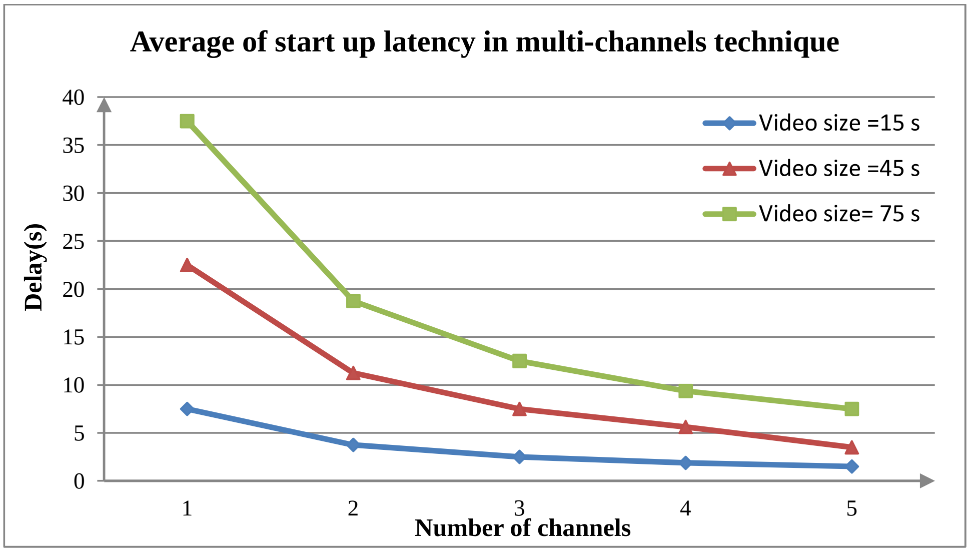

Figure 40: Effects of multi-channeling/multicast on start up latency

One important issue in multi-channeling is the ability of sharing channels with other videos (only in best case) in order to use channel capacity more efficiency. In this approach, one channel is assigned to a specific video in a certain time slot. If video size is equal to V=15 s and transfer rate is t_r= 7.5 Mbps, and video is multicasted on three channels, start up latency and chunk size will be 5 second. In general, MPGE display rate is between 1.2 and 1.5 Mbps, therefore, it is possible to download 5 seconds of video files during one second, then the channel is free during the next four seconds and it can be shared with other four videos. In fact, chunk size is divided into equal time slots and is assigned to different videos. In such a case we need to calculate videos and channels overhead. This way, each video can take more time slots for transferring while there are not any other videos.

Video speed up is affected by transfer rate, display rate, startup latency, and buffer size. A high quality video needs a high data rate. Capability of switching between channels increases download rate in low data rate technology. Patching and video locality improve start up latency. Our proposed approach for managing buffer contents attains better results compared to LRU and FIFO algorithms. This technique increases the probability of video locality which leads to speed up. Figure 41 shows the average result of our proposed algorithm in 100 different simulations. Fewer numbers of faults mean better efficiency.

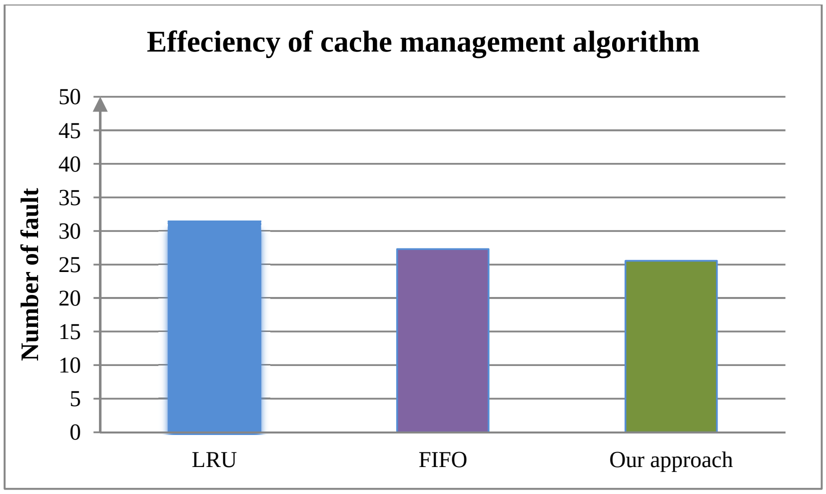

Figure 41: Result of our algorithm compared with FIFO and LRU in 50 requests

Figure 41, indicates that LRU techniques has more fault than FIFO and our proposed approach. Although in some cases FIFO has a better results, in long term simulation, our proposed approach is better than both LRU and FIFO orders.

Video locality decreases the response time and improves speed up. It is because downloading video from local server is faster than downloading video from the remote media server. Furthermore, an individual access to remote media server increases core network traffic and causes a bottleneck problem in storage and I/O devices that causes to low efficiency. Figure 42 shows effect of multicasting over multichannel with focusing of video locality when t_r=t_d (average case).

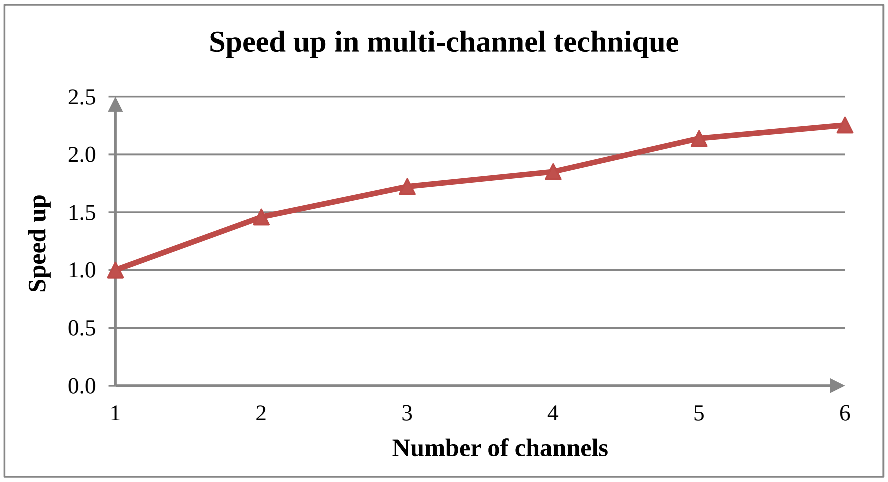

Figure 42: Speeds up in multi-channeling

Figure 42 indicates that multi-channel technique increases speed up by reducing startup latency and join clients to suitable channels.

Proxy server in BSC limits the number of requests to remote media server and improves start up latency by forwarding requests from local server which is closer to the client. Figure 43 shows the effect of using proxy server in BSC and indicates that video locality improves speed up.

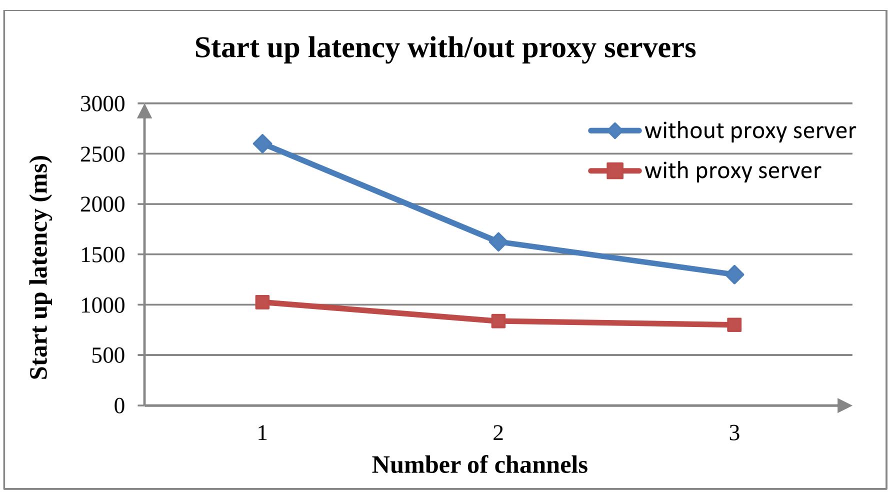

Figure 43: Effects off using proxy server in cellular networks

Chapter 6

CONCLUSION

In recent years broadcasting video over wireless networks has become more popular. Various types of wireless networks provide real time and non real time communication such as voice, video, and data for mobile users. Content delivery latency as a part of QoS is an important issue in real time communication. VoD in wireless networks (especially in cellular wireless networks) enable users to request and view the video anytime anywhere. Multicasting technique is an efficient way that supports VoD for clients when more clients request the same video at the same or closer time.

Speed up in cellular networks depends on both cellular network and core networks (Internet) specifications. Also, mobile clients need to have enough memory space to save more segments. Providing a proxy server and patching mechanism improves speed up by limiting the number of video request from the remote server. Furthermore, multi channeling decrease start up latency and increase speed up.

In general, bandwidth is limited in cellular networks, but video communication needs more bandwidth. Latest versions of cellular networks such as LTE may improve this shortcoming but it is not popular yet. Furthermore, new technologies need new infrastructure and resources such as BS, antenna, application software, and also new version of mobile devices.

The main aim of this study is improving video on demand efficiency in wireless networks by focusing on multicasting/ patching technique, multi channeling, and video locality. We try to improve VoD efficiency in currently popular wireless networks such as WLAN and 3G and beyond 3G cellular networks. We improve shortcomings of related work such as response time, startup latency and delay limitations by focusing on three main points:

- ✓ Multi channeling and periodic broadcasting technique
- ✓ Scheduled multicasting by increasing number of request
- ✓ Proxy server and video locality

Also we provide a new algorithm for cache management. According to the simulation results, our approach is better than both LRU and FIFO algorithms.

The main results of this study are:

- Multicasting many videos over multi channels improve start up latency and overcomes overflowing problem in high data rate networks.
- Increasing packet size up to 1500 bytes improves network efficiency. However packet size bigger than 1500 bytes leads to lower efficiency because of packet fragmentation.
- Using proxy server in each BS is costly and increases network complexity, using proxy server in BSC (which is located in MSC) improves speed up and start up latency. Furthermore, this approach decrease core network (Internet) traffic and overcomes remote media server bottleneck problems.

Future Work

In this study we setup one proxy server for each BSC. The number of BSC's, in a mobile network is related with the total number of mobile subscribers. Contents of each proxy server may be different from others. When proxy server does not find the requested video in its own memory, it sends the request to other peers before forwarding it to remote media servers. Another idea might be to form a P2P network between BSC's. The architecture of such a P2P based proxy servers is illustrated in figure 44.

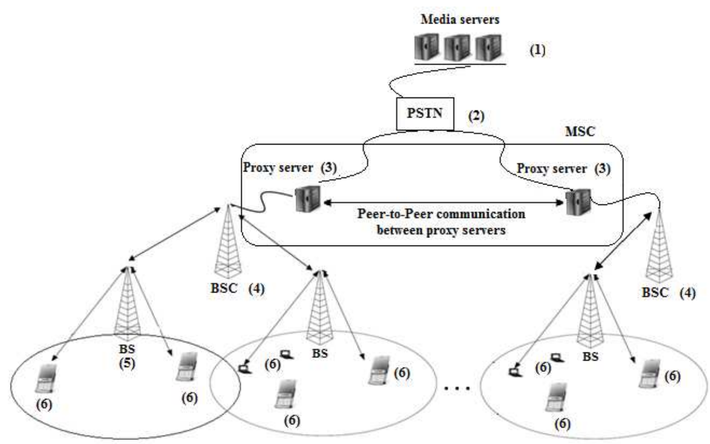

Figure 44: Peer-to-Peer communication based proxy server

Handoff is not considered in this study. When a client is moving from current cell coverage area to enter another cell's coverage area, handoff occurs. In non-CDMA networks neighbor cells use different frequency bands. So the frequency band changes during handoff. In CDMA networks, neighbor cells may use the same frequency band, so source and destination cells may have the same frequency. In this case there is no need for handoff.

REFERENCES:

[1] Abdulova, V., Aybay, I. (2011). Predictive Mobile-Oriented Channel Reservation Schemes in Wireless Cellular Networks. *Wireless Netw* 17:149–166

[2] Bar-Noy, A., Goshi, J., & Tam, K. (2004) Comparison of stream merging algorithms for media-on-demand. *Multimedia Syst* 9(5):411–423

[3] Buchinger, S., & lavacs, H. (2010). Optimal Server Bandwidth for Mobile Video On Demand. *Ann. Telecommun*. 21, 65:31–46.

[4] Yang, X.Y., Cores, F., Hernandez, PA., & Ripoll, E. (2010). Designing an Effective P2P System for a VoD System to Exploit the Multicast Communication. *J. Parallel Distrib. Comput.* 70, 1175–1192

[5] Chen, S., Wang, H., Zhang, X., Shen, B., & Wee, S. (2005). Segment-Based Proxy Caching for Internet Streaming Media Delivery. *IEEE Multimedia.* 12 (3), 59-67.

[6] Sarper, H., Aybay, I. (2007). Improving VoD Performance with LAN Client Back-End Buffering. *IEEE Multimedia*, Volume 14, Issue 1, 48-60

[7] Sarper, H., Aybay, I. (2007). A Video Locality Based Buffering Mechanism for VoD Systems. *IEEE Transactions on Consumer Electronics*, Vol. 53, No. 4, 1521-1528

[8] Liang, C., Fu, Z., Liu, Y., & Wu, C.W. (2010). Incentivized Peer-Assisted Streaming for On-Demand Services, Parallel and Distributed Systems, IEEE Transactions.Volume:21,Issue:9 Page(s): 1354 – 1367

[9] Choi, J., Reaz, A. & Mukherjee, B. (2012). A Survey of User Behavior in VoD Service and Bandwidth-Saving Multicast Streaming Schemes. *IEEE Communications Surveys & Tutorials*, VOL. 14, NO. 1, FIRST QUARTER 2012

[10] Tran, A., Nguyen, T. (2010). Broadcasting Techniques for Video-on-Demand in Wireless Networks. Department of Computer Science, University of Dayton, Dayton, OH 45469

[11] Ding, Y., Yang, Y. & Xiao, L. (2011). Multi-Path Routing and Rate Allocation for Multi-Source Video On-Demand Streaming in Wireless Mesh Networks. *IEEE INFOCOM* 2011, 2051-2059

[12] http://encyclopedia.jrank.org/articles/pages/6785/Large-Scale-Video-on-Demand-Systems.html. 4/5/2012

[13] Hua, K.A., & Sheu, S. (1997). Skyscraper Broadcasting: A New Broadcasting Scheme for Metropolitan Video-On-Demand Systems. *In Proc. of the ACM SIGCOMM'97*, pages 89–100.

[14] Febiansyah, H., Kwon, J.B. (2012). Dynamic proxy-assisted scalable broadcasting of videos for heterogeneous environments, *Multimedia Tools and Application, Published Online.*

[15] Kuo, W.H., & Lin, T.Y. (2012). Multi-Hop Multicast Path Construction in Modern Wireless Relay Networks. *Computer Networks.*

[16] Tran, D. A., Hua, K. A., & Le, M. (2004). Mobivod: A Video-On-Demand System Design for Mobile Ad Hoc Networks. *In Proceedings of IEEE International Conference on Mobile Data Management (MDM 2004*). Berkeley, CA, USA.

[17] K.M. Ho, K.T. Lo. (2012). Provision of VCR Functionality for Video-on-Demand System with Cooperative Clients in Broadcast Environment. *Proceedings Of International Multi Conference Of Engineer And Computer Scientists*, Vol I, IMECS, March 14-16, Hong Kong

[18] Kwon, J. B., & Heom, H.Y. (2002). Providing VCR Functionality in Staggered Video Broadcasting. *IEEE Transactions on Consumer Electronics*, 48(1):41-48.

[19] Yu, H.F., Yang, H.C., Wang, Y.T., Fan, P.L., & Chien, C.Yi. (2009). Broadcasting Scheme with Low Client Buffers and Bandwidths for Video-On-Demand Applications. *Multimedia Tools.* 42:295–316.

[20] Do, T.T., Hua, K.A., Jiang, N., & Liu, F. (2009). A Scalable Video-On-Demand Streaming System in Hybrid Wireless Mobile Peer-to-Peer Networks. *Peer-to-Peer Netw Appl*, 2:182–201

[21] Shokri kalan, R. (2010). Wireless Networks (CWNA). Page 180. Eshraghi. ISBN 978-964-388-273-0

[22] http://standards.ieee.org/develop/wg/WG802.22.html. 9/24/2012

[23] Rappaport, T.S. (2002). Wireless Communications Principle and Practice.

[24] http://www.motorola.com/web/Business/Products/Mobile%20Computers/Handheld%20Computers/_Documents/staticfile/3G_Whitepaper_0608.pdf" 23/11/2012

[25] Molisch, A.F. (2005). Wireless Communications. John Wiley & Sone, Ltd. ISBN-13 978-0-4470-84887-6 (HB)

[26] Kaaranen, H., Ahtiainen, A., laitinen, L., Naghian, S., & Niemi, V. (2005). UTMS Networks. John Wily & Sone, Ltd. ISBN-10-0-470-01103-3 (H/B)

[27] Tammaa, B.R., Badam, A., Murthy, Rao, R.R., & Tree, K. (2010). A Multiple Tree Video Multicast Protocol for Ad Hoc Wireless Networks. *Computer Networks.* 54, 1864–1884.

[28] Sheu, I-P., Wang, H-Lih., Chang, C-H., & Chee, Y. (2004). Tseng, A Fast Video-on-Demand Broadcasting Scheme for Popular Videos. *IEEE Transactions on Broadcasting*. Vol. 50, No. 2.

[29] Sayit, M., & Seckin, G. (2012). Scalable Video with Raptor for Wireless Multicast Networks. *Vidiator technolog., USA, p 336-341.*

[30] Tabassum, K., Sultana, A., & Damodaram, A. (2012). Flexible Data Dissemination Strategy for Effective Cache Consistency in Mobile Wireless

Communication Networks. *International Journal of Distributed and Parallel* (IJDPS). Vol.3, No.3.

[31] http://whatis.techtarget.com/definition/content-delivery-network-CDN. 21/11/2012

APPENDICES

Appendix A: Programming code of my proposed algorithm for cache management.

```
#include <iostream.h>

#include <conio.h>

struct video{

                int size;

               }v[100];

int memory=100; //MB

int req_order[50]={8,0,9,7,1,4,3,8,3,5,1,2,1,0,9,9,2,3,4,6,5,8,7,0,5,

                          3,3,9,3,0,4,7,7,1,2,4,0,9,7,6,4,5,8,7,0,1,6,2,7,4};

/*int req_order[50]={8,0,9,7,1,4,3,8,3,5,1,2,1,0,9,9,2,3,4,6,5,8,7,0,5,

                          3,3,9,3,0,4,7,7,1,2,4,0,9,7,6,4,5,8,7,0,1,6,2,7,4};

int req_order[50]={8,0,9,7,1,4,3,8,3,5,1,2,1,0,9,9,2,3,4,6,5,8,7,0,5,

                          3,3,9,3,0,4,7,7,1,2,4,0,9,7,6,4,5,8,7,0,1,6,2,7,4};

int req_order[50]={8,0,9,7,1,4,3,8,3,5,1,2,1,0,9,9,2,3,4,6,5,8,7,0,5,

                          3,3,9,3,0,4,7,7,1,2,4,0,9,7,6,4,5,8,7,0,1,6,2,7,4};

int req_order[50]={8,0,9,7,1,4,3,8,3,5,1,2,1,0,9,9,2,3,4,6,5,8,7,0,5,

                          3,3,9,3,0,4,7,7,1,2,4,0,9,7,6,4,5,8,7,0,1,6,2,7,4};*/

//-----------------------------------------------

int queue[100],front=-1,rear=-1;

void add(int x)

{

 queue[++rear]=x;

}
```

```
int delet()
{
 return queue[++front];
}
int exist(int x)
{
 int f=0;
 for(int i=front+1;i<=rear;i++)
  if(queue[i]==x)f=1;
 return f;
}
//-------------------------------------------------
void FIFO()
{
v[0].size=20;v[1].size=25;v[2].size=37;v[3].size=22;v[4].size=23;
v[5].size=12;v[6].size=26;v[7].size=10;v[8].size=15;v[9].size=18;
 int i,j,t,count=0;
 int empty=memory;
 cout<<"\n\n--------------------------------------------------------------------------------\n";
 cout<<"\n\n                    Faulted Videos using FIFO Strategy :\n\n\n";
 for(i=0;i<50;i++)
 {
  if(!exist(req_order[i]))
  {
   if(v[req_order[i]].size<=empty)
```

```
    {
     add(req_order[i]);
     empty=empty-v[req_order[i]].size;
    }
    else
    {
     while(v[req_order[i]].size>empty)
     {
      t=delet();
      cout<<t<<"  ";
      empty=empty+v[t].size;
      count++;
     }
     add(req_order[i]);
     empty=empty-v[req_order[i]].size;
    }
  }
 }
 cout<<"\n\n\nVideo fault Count= "<<count;
}
//-------------------------------------------------
void LRU()
{
 int mem[50],m=0,used[50],count=0;
 v[0].size=20;v[1].size=25;v[2].size=27;v[3].size=32;v[4].size=43;
```

```
v[5].size=30;v[6].size=36;v[7].size=22;v[8].size=5;v[9].size=18;
int i,j,t,k;
int empty=memory;
cout<<"\n\n\n-------------------------------------------------------------------------------\n";
cout<<"\n\n                    Faulted Videos using FIFO Strategy :\n\n\n";
for(i=0;i<50;i++)
{
 int f=0;
 for(k=0;k<m;k++)
  if(mem[k]==req_order[i])f=1;
 if(!f)
 {
  if(v[req_order[i]].size<=empty)
   {
    mem[m++]=req_order[i];
    empty=empty-v[req_order[i]].size;
   }
   else
   {
    for(k=0;k<50;k++)
    used[k]=0;
    while(v[req_order[i]].size>empty)
    {
     if(i>10)k=i-10;
            else k=0;
```

```
    for(j=0;j<m;j++)
    {
     for(t=k;t<i;t++)
          if(mem[j]==req_order[t])
             used[j]++;
    }
    int min=0,h=1;
    while(h<m)
    {
     if(used[h]<used[min]) min=h;
     h++;
    }
    t=mem[min];
    for(h=min;h<m-1;h++)mem[h]=mem[h+1];
    m--;
    cout<<t<<"  ";
    count++;
    empty=empty+v[t].size;
   }
   mem[m++]=req_order[i];
   empty=empty-v[req_order[i]].size;
  }
 }
}
cout<<"\n\n\nVideo fault Count= "<<count;
```

```
}
//-----------------------------------------------
void kalan()
{
 int mem[50],m=0,used[50],count=0;
 v[0].size=20;v[1].size=25;v[2].size=27;v[3].size=32;v[4].size=43;
 v[5].size=30;v[6].size=36;v[7].size=22;v[8].size=5;v[9].size=18;
 int i,j,t,k;
 int empty=memory;
 cout<<"\n\n\n--------------------------------------------------------------------------------\n";
 cout<<"\n\n                    Faulted Videos using FIFO Strategy :\n\n\n";
 for(i=0;i<50;i++)
 {
  int f=0;
  for(k=0;k<m;k++)
   if(mem[k]==req_order[i])f=1;
  if(!f)
  {
   if(v[req_order[i]].size<=empty)
    {
     mem[m++]=req_order[i];
     empty=empty-v[req_order[i]].size;
    }
    else
    {
```

```
for(k=0;k<50;k++)
used[k]=0;
while(v[req_order[i]].size>empty)
{
 float Alfa=0,cost[20];
 for(k=0;k<m;k++)
         Alfa=Alfa+(float)v[mem[k]].size;
 Alfa=(float)Alfa/m;
 for(k=0;k<m;k++)
         cost[k]=(100*Alfa)/v[mem[k]].size;
 if(i>8)k=i-8;
        else k=0;
 for(j=0;j<m;j++)
 {
  for(t=k;t<i;t++)
       if(mem[j]==req_order[t])
          used[j]++;
 }
 int prt[10];
 for(j=0;j<m;j++)
        prt[j]=cost[j]+used[j];
 int min=0,h=1;
 while(h<m)
 {
  if(prt[h]<prt[min]) min=h;
```

```
      h++;
      }
      t=mem[min];
      for(h=min;h<m-1;h++)mem[h]=mem[h+1];
      m--;
      cout<<t<<"  ";
      count++;
      empty=empty+v[t].size;
     }
     mem[m++]=req_order[i];
     empty=empty-v[req_order[i]].size;
    }
  }
 }
 cout<<"\n\n\nVideo fault Count= "<<count;
 cout<<"\n\n\n--------------------------------------------------------------------------------\n";
}
//------------------------------------------------
main()
{ clrscr();
 FIFO();
 LRU();
 kalan();
 getch();
}
```

```
//---------------------------------------------------
```