Energy-aware Adaptive Solutions for Multimedia Delivery to Wireless Devices

Martin Kennedy, B.Eng.

A Dissertation submitted in fulfillment of the requirements for the award of Doctor of Philosophy (Ph.D.)

Dublin City University

School of Electronic Engineering

Supervisor: Associate Professor Gabriel-Miro Muntean

September 2017

Declaration

I hereby certify that this material, which I now submit for assessment on the programme of study leading to the award of Ph.D. is entirely my own work, and that I have exercised reasonable care to ensure that the work is original, and does not to the best of my knowledge breach any law of copyright, and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work.

Signed: _____

ID No.: 55502284

Date: _____

Acknowledgments

I have been absolutely blessed with my family and friends. They have all been so supportive during my Ph.D. journey and I cannot describe how much this has meant to me. There were many times over the years when this research was a struggle, but the amazing network of people in my life got me through it.

To Beccy, my better half. It has been a bit of a rollercoaster, eh? It may have taken slightly longer than the 3 years anticipated but thank you so much for always believing in me, for pushing me on, for the laughter, the music, the travel, the cat! You are the most wonderful person and I love you to bits!

To Mam and Dad, you guys are the best parents I could possibly have hoped for. The last 30 years have been filled to the brim with joy and music and family trips and lovely memories. You have given me every opportunity in life and I admire you both so much! This Ph.D. could never have happened only for your constant love and support along the way.

To my three sisters, Marianne, Caroline and Paula, what can I say? Yippee!!! Thank you all for always looking out for me and for being so deadly in general!

To Auntie, for all the candles and prayers over the years. You are such a kind and generous person and I love you very much. I hope you're ready for a trip down to Dublin for the graduation ceremony!

To Gabriel, my mentor, my supervisor, my friend. We have been doing research together since my John Holland summer internship in 2007. You are so knowledgeable and enthusiastic and you worked tirelessly to help me progress with my research and so that I could get experience on some Enterprise Ireland research projects too. I cannot express my gratitude for your support, patience and effort over the years!

To the rest of my colleagues from PEL and Xtremepush, thank you for always being there to bounce ideas off and figure things out together.

Finally, to my examination panel, Prof. Gregor Rozinaj, Dr. Conor Brennan and Dr. Derek Molloy, thank you for your time, expertise and feedback. It has been an invaluable experience.

List of Publications

Journal Papers

M. Kennedy, A. Ksentini, Y. Hadjadj-Aoul and G.-M. Muntean, "Adaptive Energy Optimization in Multimedia-Centric Wireless Devices: A Survey," in *IEEE Communications Surveys & Tutorials*, vol.15, no.2, pp.768,786, Q2 2013.

Conference Papers

M. Kennedy and G.-M. Muntean, "A study on the effect of transmission power adaptation and multi-hop path usage on power consumption and QoS in adaptive mobile video delivery," in *9th IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*, Beijing, China, June 2014.

M. Kennedy, H. Venkataraman and G.-M. Muntean, "Dynamic Stream Control for Energy Efficient Video Streaming," in *6th IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*, Nuremberg, Germany, June 2011.

M. Kennedy, H. Venkataraman and G.-M. Muntean, "Battery and Stream-Aware Adaptive Multimedia Delivery for Wireless Devices," in 7th *IEEE International Workshop on Performance and Management of Wireless and Mobile Networks (P2MNET)*, Denver, USA, October 2010.

D. Theurer, **M. Kennedy**, H. Venkataraman and G.-M. Muntean, "Analysis of Individual Energy Consuming Components in a Wireless Handheld Device," in *2010 China-Ireland International Conference on Information and Communications Technologies (CIICT)*, Wuhan, China, October 2010.

Book Chapters

M. Kennedy, H. Venkataraman and G.-M. Muntean, "Energy Consumption Analysis and Adaptive Energy Saving Solutions for Mobile Device Applications," in *Green IT: Technologies and Applications*, Jae Kim (Eds), Springer, August 2011.

Table of Contents

List of Abbreviations

AP:	Access Point
APSD:	Automatic Power Save Delivery
AVC:	Advanced Video Coding
BaSe-AMy:	Battery and Stream-aware Adaptive Multimedia Delivery
BLE:	Bluetooth Low Energy
CDMA:	Code Division Multiple Access
DASH:	Dynamic Adaptive Streaming over HTTP
DVS:	Dynamic Voltage Scaling
EASE:	Energy-aware Adaptive Solutions for Multimedia Delivery
EDGW:	Enhanced Data rates for Global Evolution
EDR:	Enhanced Data Rate
GPRS:	General Packet Radio Service
GPS:	Global Positioning System
GPU:	Graphics Processing Unit
GSM:	Global System for Mobile Communications
HEVC:	High Efficiency Video Coding
HLS:	HTTP Live Streaming
HS:	High Speed
HSCSD:	High-Speed Circuit-Switched Data
IoT:	Internet of Things
IrDA:	Infrared Data Association
KMP:	Key Management Protocol
LCD:	Liquid Crystal Display
LTE:	Long Term Evolution
MIMO:	Multiple Input Multiple Output
MOS:	Mean Opinion Score
MPEG:	Moving Picture Expert Group
NFC:	Near Field Communication
OLED:	Organic Light Emitting Diode
OS:	Operating System

P2P:	Peer-to-peer
PAN:	Personal Area Network
PDA:	Personal Digital Assistant
PSM:	Power Save Mode
PSNR:	Peak Signal-to-Noise Ratio
QoE:	Quality of Experience
QoS:	Quality of Service
RFID:	Radio-Frequency Identification
RoI:	Region of Interest
SD:	Secure Digital
SNR:	Signal-to-Noise Ratio
SSIM:	Structural Similarity Index
SVC:	Scalable Video Coding
UMTS:	Universal Mobile Telecommunications System
VoIP:	Voice over IP
VQM:	Video Quality Metric
VSNR:	Visual Signal-to-Noise Ratio
WLAN:	Wireless Local Area Network
WNIC:	Wireless Network Interface Card
WPA:	Wi-Fi Protected Access
WPAN:	Wireless Personal Area Network

List of Tables

List of Figures

Abstract

The functionality of smart mobile devices is improving rapidly but these devices are limited in terms of practical use because of battery-life. This situation cannot be remedied by simply installing batteries with higher capacities in the devices. There are strict limitations in the design of a smartphone, in terms of physical space, that prohibit this "quick-fix" from being possible. The solution instead lies with the creation of an intelligent, dynamic mechanism for utilizing the hardware components on a device in an energy-efficient manner, while also maintaining the Quality of Service (QoS) requirements of the applications running on the device.

This thesis proposes the following Energy-aware Adaptive Solutions (EASE):

1. **BaSe-AMy**: the Battery and Stream-aware Adaptive Multimedia Delivery (BaSe-AMy) algorithm assesses battery-life, network characteristics, video-stream properties and device hardware information, in order to dynamically reduce the power consumption of the device while streaming video. The algorithm computes the most efficient strategy for altering the characteristics of the stream, the playback of the video, and the hardware utilization of the device, dynamically, while meeting application's QoS requirements.
2. **PowerHop**: an algorithm which assesses network conditions, device power consumption, neighboring node devices and QoS requirements to decide whether to adapt the transmission power or the number of hops that a device uses for communication. PowerHop's ability to dynamically reduce the transmission power of the device's Wireless Network Interface Card (WNIC) provides scope for reducing the power consumption of the device. In this case shorter transmission distances with multiple hops can be utilized to maintain network range.
3. A comprehensive **survey** of adaptive energy optimizations in multimedia-centric wireless devices is also provided.

Additional contributions:

1. A custom **video comparison tool** was developed to facilitate objective assessment of streamed videos.
2. A new solution for **high-accuracy mobile power logging** was designed and implemented.

Chapter 1 - Introduction

This chapter documents the rise of the smart-phone and introduces the challenges associated with their power consumption characteristics. After that, the problem statement for this thesis is outlined and the solution proposed here is presented. The novel contributions of the solution in this thesis are introduced and then the structure for the remainder of the thesis is detailed.

1.1 Research Motivation

Sales of smart-phones have grown consistently each year since their first introduction and they have quickly become a ubiquitous technology. Smart-phone sales began gaining traction in the early 2000s with the introduction of devices to the market running Symbian and Blackberry Operating Systems (OS). These devices were primarily designed for productivity and functioned as mobile interfaces to the office. Things changed dramatically in 2007 however with the release of Apple's first iPhone. This device immediately stood out from its competitors because its style, its stylus-free multi-touch interface and, later, for the applications that were developed for it and released through the App Store. This device could perform the office related tasks of the Symbian and Blackberry devices but could also play music and video files and take photographs, making it a unified solution for the amalgamation of multiple devices: cameras, media players and phones.

Over the next 10 years, annual global smart-phone sales grew steadily from 122 million units in 2007 to 1.5 billion units in 2016 [1][2][3][4][5]. But while sales have sky-rocketed, there is an inherent flaw with all of the smart-phones available on the market today. Unfortunately, because of the device's energy requirements and the limited physical space for a device's battery in the phone design, the charge of a battery of a smart-phone will typically last for less than 48 hours. As a result, most smart-phone owners will experience a full battery depletion at some point while they do not have access to a phone charger. This thesis will introduce a novel solution for mobile device operation to reduce the power consumption of a smart-phone and increase its battery life but first let's start at the beginning and analyze the situation at hand.

1.2 Problem Statement

Both the functionality and the computational capability of mobile devices have developed rapidly in recent times. Importantly, modern smart-phones and tablets are now capable of interacting with multiple sensors, multitasking and communicating over various independent network interfaces. While these devices follow a functionality improvement rate similar to Moore's law, developments in battery capacity have lagged behind considerably. In the case of lithium-ion batteries, since its first commercial release in 1991 the energy density of a cell has increased from 200 Wh/l to 700 Wh/l in 2016 [6]. This equates to an improvement of approximately 5% per year [7]. A classic example of the gap between functionality and power-supply is the iPhone 7. When used continuously, for web browsing over a 4G LTE network, the battery charge will only last for "up to" 12 hours [8].

The battery life of a mobile device depends on both the hardware of the device and also the utilization of the hardware through the device's applications. One of the most popular and computationally intensive applications is video streaming. Mobile video streaming is an area which is experiencing an incredible growth rate. According to Cisco, video streaming currently accounts for 60% of all mobile data traffic in 2016. This percentage is predicted to grow to 78% of the total mobile data traffic in 2021 [9]. The specifications of the HTC Nexus One promote a battery life of 7 hours during talk time in 3G networks [10] but our measurements have shown that for the energy intensive application of video streaming, the battery depletes in as little as 4 hours. In order for a mobile phone to be effectively used on a day-to-day basis, it has to be able to function for at least 12 hours before requiring battery recharging. Currently, there are different energy conservation techniques included in mobile OS, such as background process control and ambient light-aware screen brightness adaptation. However, these approaches are not sufficient to provide the required energy savings in the device. More specifically, there are no comprehensive solutions that have been designed to conserve power on a device so that a video of a particular duration can be streamed in full. Video streaming applications are just one example of many applications that rapidly reduce battery life below acceptable levels.

1.3 Solution Overview

The built-in energy conservation techniques in modern smart-phone OSs include background process control and ambient light-aware screen brightness control. The background process-control functionality is a very flexible implementation. For example, the Android OS has a

process manager which manages applications that have completed their execution life-cycle. The only problem is that processes of the completed application can remain in memory on the device for hours, there are no time guarantees associated with the default process manager to dictate when dead processes will be removed from the system. To combat this, there are several solutions in the Android application market to take on this role. Whenever space in system memory becomes limited, these third-party applications clear out dead application processes automatically. In the case of the default screen brightness control, the ambient light sensor on the phone is used to automatically set the brightness of the device's screen. This works quite well for everyday usage but does not perform optimal adaptations of the screen brightness.

Currently, there is no sophisticated adaptive process that prolongs the battery life in real-time, depending on the nature of the application and current battery level. What is needed is a scheme which runs on the device. It should keep track of the applications running on the device and dynamically create an energy conservation strategy based on the resource requirements of the other applications. This strategy should be more aggressive if the battery level of the phone is low, thus ensuring the battery life is extended so that the objective of a particular application can be achieved.

1.4 Thesis Contributions

The *first contribution* of this research is the novel Battery and Stream-Aware Adaptive Multimedia Delivery (BaSe-AMy) algorithm. BaSe-AMy is an intelligent, context-aware algorithm for dynamically adapting the characteristics of a device in an energy and application-aware manner. This solution specifically targets different components on the mobile device and scales their functionality in order to save energy. Each type of application on a device can have significantly different Quality of Service (QoS) requirements and can also involve different methods for user interaction. For example, a browser application requires internet access but it is generally used to access web-pages and may not need network connectivity between the loadings of two sites. In contrast, constant network connectivity and sufficient bandwidth are highly important in a video streaming application. As a result, each application type will require different methods for achieving reductions in the power consumption. Each application type needs to be analyzed independently and subsequently incorporated into a unified mechanism for maximizing the energy savings while minimizing the effect to the QoS. BaSe-AMy has been proposed in the specific context of a video

streaming application. BaSe-AMy calculates the remaining battery duration and compares this to the remaining duration of the video stream. Based on this comparison, and on the network conditions, the brightness of the display and the quality of the video stream are adapted in real-time to attempt playback of the whole stream before depletion of the battery. If no adaptation is required for playback of the entire video sequence, then BasSe-AMy does not change any of the video or device parameters. BaSe-AMy is described in full in Chapter 5.

The *second contribution* of this research is the PowerHop algorithm which performs multi-hop data transmissions over a wireless network from the mobile device. PowerHop can dynamically switch a device's wireless transmissions from a single-hop transmission path to a multi-hop path, while scaling the transmission power of the device's network interface. In this way, the energy usage on the mobile device can be greatly reduced. In infrastructure mode IEEE 802.11 wireless networks, a mobile device on a wireless network communicates directly with an Access Point (AP). The further the wireless device is from the AP, the higher the transmission power necessary for successful communication over the network. In wireless networks however, a multi-hop route can exploit intermediary devices on the network between the mobile device and the network gateway. The mobile device can transmit data to the intermediary devices, which then relay the data on to the gateway. In this setup, the distance that the mobile device must transmit data is reduced; therefore the energy consumption of the device is also reduced. PowerHop is presented in full in Chapter 6.

The *third contribution* of this thesis is the related work chapter, a good portion of which was published in a paper in IEEE Transactions on Communications Surveys and Tutorials [11]. This paper provided a detailed overview of the research space and is enhanced further in this thesis.

In addition to these core contributions, the following minor contributions also supported the work presented in this thesis:

- A custom video comparison tool was developed to allow for robust calculation of the PSNR of a video against a reference video sequence. The tool also provides functionality for regenerating adaptive video sequences from the log files of a streaming server. These regenerated video files can then be used in the PSNR calculations, if necessary. This piece of work is described in full in Chapter 5.
- Power monitoring logger – a mobile Arduino-based circuit was developed. This circuit was used to log the power consumption of test devices to an SD card. This

circuit was necessary for the test scenarios related to the PowerHop algorithm because the tests needed to be performed outdoors. There is more information on this in Chapter 6.

1.5 Thesis Structure

This thesis is organized as follows:

- Chapter 1 – the motivation for the research presented in this thesis is introduced and the various challenges associated with the area are discussed. An overview is given of the solutions presented here and the novel contributions of the thesis are outlined.
- Chapter 2 – the constituent background technologies relevant to the research in this thesis are introduced. Wireless standards and protocols are described and video streaming mechanisms and codecs are compared.
- Chapter 3 – following that, some of the most effective state-of-the-art energy saving techniques for energy savings on smart mobile devices are investigated.
- Chapter 4 – the architecture of the Energy-aware Adaptive Solutions (EASE) is presented and explained in detail.
- Chapter 5 – the design, architecture and evaluation of the BaSe-AMy algorithm, as well as the design and implementation of the VideoCompare tool are described. The results from both simulation and experimental-based testing of the BaSe-AMy algorithm are analyzed here in detail too.
- Chapter 6 – describes the design, testing and the subsequent results from real-world experiments of the PowerHop algorithm.
- Chapter 7 – summarizes the thesis, the contributions, the effectiveness of the contributions and explains the tasks that remain as future work.

Chapter 2 - Technical Background

The existing background technologies that are particularly relevant in mobile devices and adaptive video streaming are discussed in this chapter. These include network standards, video codecs, video streaming protocols and video quality metrics.

2.1 Introduction

The research presented in this thesis focuses on improvements in the operation of mobile wireless devices in adaptive video streaming applications. In this context, there are a number of background technologies that are relevant to this research and these will be covered in this chapter.

Firstly, the different wireless networks that a modern mobile device can connect to are detailed. This includes information for all of the network interfaces that are included in the state-of-the-art smart phones and tablet devices. Following that, the different algorithms that are used for encoding videos for transmission over a network are described and compared. For any video stream, the biggest component in terms of bandwidth and energy consumption is the video component to the stream. This statement is backed-up in Chapter 4, where the different power consumption factors for video streaming to mobile devices are compared. While there are additional encoding algorithms for compressing the audio component of the multimedia streams, these are not analyzed in depth. In this thesis, the focus is on the adaptive control of the video component of the multimedia streams because there is more scope for energy-savings in this space, as seen in the results in Chapter 4. The protocols that are used for actually delivering the encoded video streams across the network are then explored. In this thesis, some energy consumption improvements are achieved by adapting the video stream during remote delivery and playback. For this reason, only adaptive video streaming protocols are discussed in this section of the thesis. Following that, the different metrics that can be used for assessing the quality of the video that is being presented to a user are detailed. Both objective and subjective methods are explained and the choice of the metrics that are used in the remainder of this thesis is outlined here.

2.2 Wireless Networks

There are generally up to three types of network interfaces on most modern smart devices that support data transmission and reception: a Near Field or Personal Area Network (PAN) interface, a Wireless Local Area Network (WLAN) interface and a cellular interface. Other wireless interfaces such as an IrDA (Infrared Data Association) interface or a GPS (Global Positioning System) interface can also be found on many smart devices, but they either do not support data transport or are not widely used for delivering multimedia content. The network technologies for each of the three interface types listed above will now be discussed in more depth.

2.2.1 Near Field and PAN

Near Field Communication (NFC)

NFC is a Radio-Frequency Identification (RFID) based technology that provides contactless communication between two devices or a device and a tag. NFC allows communication across very short distances, typically 10 cm or less. NFC is supported by a collection of standards **ISO/IEC 14443** [12], **ISO/IEC 15693** [13], **ISO/IEC 18092** [14] and **ISO/IEC 21481** [15]. In mobile devices, NFC is often used for payment authorization purposes or for managing automatic pairing between two devices on Bluetooth or Wi-Fi networks [16]. In the case of NFC being used for pairing assistance, the new Bluetooth or Wi-Fi link handles all data transfer because it offers a higher throughput than NFC.

IEEE 802.15 Family

A **Wireless Personal Area Network** (WPAN) allows communication with higher data-rates than NFC and/or over larger distances than NFC. A representative effort to standardizing WPAN is the **IEEE 802.15** family of standards. **IEEE 802.15.1** [17, p. 1] is synonymous with the original Bluetooth standard, which is discussed in more detail below.

IEEE 802.15.2 [18, p. 2] dealt with the Coexistence of networks in the same frequency bands in order to reduce interference between the WPAN and other network types. The IEEE 802.15.2 standard was published in 2003.

IEEE 802.15.3 [19] deals with the task of the standardization of High Rate WPANs and was published in 2003. Two amendments to the standard were released in 2005 and 2009. The first

of these amendments improved the implementation and interoperability of the Medium Access Control (MAC). The second amendment added a millimeter-wave-based alternative for the Physical Layer, boosting maximum transmission speeds up to 3 Gbps.

IEEE 802.15.4 [20] was first published in 2003 and consists of a standard for low rate WPANs where the main focus is to reduce the complexity and power consumption of the network implementation. There are many wireless network specifications based on the IEEE 802.15.4 standard, such as **ZigBee** [21], **6LoWPAN** [22], **WirelessHART** [23], etc. Each of them implement the core IEEE 802.15.4 standards and add more functionality in the upper layers on top of that. Of these networks, ZigBee is the most widely adopted but, all of them have a maximum throughput of 250 kbps, which makes them unsuitable for high quality multimedia delivery. Additionally, interfaces for these networks are not included in any wireless mobile devices in wide circulation.

IEEE 802.15.5 [24] deals with adding mesh networking support to the IEEE 802.15 family of standards. IEEE 802.15.5 is built on top of 802.15.4, for the low data-rate implementation of the protocol, and uses 802.15.3 for the high data-rate implementation.

IEEE 802.15.6 [25] was published in 2012 as a standard for Body Area Networks. These networks are short range networks, typically located around a human body, with low-power network devices.

IEEE 802.15.7 [26] was published in 2011 and deals with short-range wireless communication using visible light.

IEEE 802.15.8 defines advanced PHY and MAC protocols for improved peer aware communication in peer to peer and infrastructure-less networks [27]. This standard includes methods for discovering peer devices and calculating relative position of the device as well as a number of other improvements. A draft version of this standard is yet to be published.

IEEE 802.15.9 [28] was published in 2016 and defines a framework for the transport of Key Management Protocol (KMP) datagrams as information elements. This standard lays out a recommended practice but does not define a new KMP.

IEEE 802.15.10 [29] still has draft status. It defines a recommended practice for building and maintaining dynamic routes in IEEE 802.15.4 based networks. The described practice allows the coverage of a network to increase as its number of nodes increases.

Bluetooth

Bluetooth is a low-power short range WPAN based on the **IEEE 802.15.1** standard and is now maintained by the Bluetooth Special Interest Group [30]. It is predominantly used for device-to-device data transfers and for connecting the device to wireless headphones for audio playback and hands-free device operation. A Bluetooth network's range typically only extends 10 m from a mobile device, but the standard dictates no maximum range [31]. Bluetooth has a significantly lower maximum bandwidth than other wireless solutions at 3 Mbps for version **2.0 with Enhanced Data Rate (EDR)** [32]. **Bluetooth v3.0 + High Speed (HS)** [33] increased this up to 24 Mbps by utilizing an IEEE 802.11 link.

Bluetooth v4.0 (Smart) [34] introduced a new low energy incarnation of the standard, known as **Bluetooth Low Energy (BLE)**, specifically designed for operation on battery constrained devices. BLE sacrifices throughput in order to minimize the power consumption of the interface and as a result would not be relevant for multimedia streaming applications.

Bluetooth v4.1 [35] was adopted in December 2013. It builds on from v4.0 to improve coexistence with 4G cellular networks and is intended for use as a wireless solution for an **Internet of Things (IoT)** network.

Bluetooth v4.2 [36] added more features to enhance the standard for IoT applications. The new features include support for IPv6 addressing as well as increased security and packet length for BLE transmissions.

Bluetooth 5 [37] was released in December 2016. It quadruples the transmission range, doubles the transmission speed and increases the broadcasting capacity over Bluetooth 4.2 by 800%. This has a number of immediate benefits:

- Devices can be further away from each other while still communicating effectively
- Multiple devices can be supported at the same time; one phone could stream audio to two sets of Bluetooth headphones
- Increased transmission speed means more efficient networks

Another core component of Bluetooth 5 is that it supports additional contextual information to be sent from broadcasting devices, such as beacon devices. This means that location and navigational solutions can be implemented for mobile devices without the need for a custom application to be installed on the mobile device. The Samsung Galaxy s8 was the first mainstream mobile device released that supports Bluetooth 5 [38].

2.2.2 Wireless Local Area Networks

WLAN or Wi-Fi networks are based on the **IEEE 802.11** [39] standard. WLAN devices typically communicate together in either the 2.4 GHz band or the 5 GHz band. The IEEE 802.11 standard was first introduced in 1997 and allowed network devices to communicate at speeds up to 2 Mbps in the 2.4 GHz band. Since then there have been several improved versions of the 802.11 protocol that became widely available in implementations in consumer devices.

IEEE 802.11a [40] and **IEEE 802.11b** [41] were the first two of these updated protocols and were both released in 1999. The 802.11a standard can operate in the 5 GHz band at a bandwidth of up to 54 Mbps. The 802.11b standard is restricted to the 2.4 GHz band only and has a maximum bandwidth of 11 Mbps. Both of these standards employ 20 MHz wide channels for communication.

IEEE 802.11g [42] builds on the 802.11b protocol. 802.11g is also confined solely to the 2.4 GHz band and has the same channel width, but has a maximum bandwidth of 54 Mbps.

IEEE 802.11n [43] was standardized in 2009. 802.11n can function in either the 2.4 GHz or 5 GHz bands and it supports the use of multiple antennas (up to four) on a device, providing Multiple Input Multiple Output (MIMO) capabilities. Additionally, it introduced the use of channel bonding so that channels can be up to 40 MHz wide, in order to speed up communications. The combination of technology, MIMO and bonding functionality means that the maximum bandwidth is 600 Mbps.

IEEE 802.11ac [44] is the latest standard that builds upon the MIMO and bonding operations of 802.11n. 802.11ac pertains solely to the 5 GHz bands of the spectrum. It allows for up to eight antennas to function on a single MIMO device, allows for increased channel width, up to 160 MHz, and introduces a new modulation scheme (256-QAM). The transmission speeds

for the 802.11ac standard depend upon the configuration of the network device but can be as high as 866.7 Mbps per MIMO antenna on a device when using 256-QAM and a 160 MHz wide channel.

IEEE 802.11ad [45], also known as WiGig, operates in the 60 GHz band. The maximum throughput of 802.11ad is just less than 7 Gbps, but it has very different use cases to the other protocols listed here. The range of the network is confined to line-of-sight, small distance communications because of its operation in the 60 GHz band. Communication at this frequency does not have the same penetration as at 2.4 or 5 GHz. This is compensated for in 802.11ad with the use of beamforming and the protocol also has the option to switch to a channel in either the 2.4 or 5 GHz band.

IEEE 802.11ah [46], also known as Wi-Fi HaLow [47], is a standard that operates in the unlicensed bands below 1 GHz. This is primarily envisioned for Internet of Things networks and for network range-extension applications. This is because of the increased transmission range of the sub 1 GHz bands, in comparison to the 2.4 or 5 GHz bands.

IEEE 802.11ax [48], is the successor to IEEE 802.11ac. It is still in development but will operate in both the 2.4 and 5 GHz bands and is expected to achieve a network throughput of 10 Gbps by using OFDA for stream multiplexing.

The protocols or standard amendments discussed here can be summarized as seen in Table 2.1. Other amendments to the 802.11 standard are also listed in Table 2.2.

In the 802.11 standard, a Wireless Network Interface Card (WNIC) network can function in two discrete modes: *infrastructure mode* and *ad hoc mode*. *Infrastructure* mode operation is where the WNIC of a device connects to a network via an AP. The other nodes on the network must connect to the same AP in order to be able to communicate with each other and to other LANs. The AP manages the connection of all nodes and routes all packets to their destination nodes on the network directly. In *ad hoc* mode, there is no central device to manage the network and wireless nodes communicate between each other directly. In *ad hoc* networks there are many different routing protocols for handling the specific route a packet takes on its way to its destination. However, these routing protocols are beyond the scope of this thesis.

Table 2.1 – Summary of IEEE 802.11 Standard Amendments Discussed Above

Protocol	Date	Spectrum	Max. Data Rate
802.11 [39]	1997	2.4 GHz	2 Mbps
802.11a [40]	1999	5 GHz	54 Mbps
802.11b [41]	1999	2.4 GHz	11 Mbps
802.11g [42]	2003	2.4 GHz	54 Mbps
802.11n [43]	2009	2.4 GHz/5 GHz	600 Mbps
802.11ac [44]	2013	5 GHz	Just < 7 Gbps
802.11ad [45]	2012	60 GHz	Just < 7 Gbps
802.11ah [46]	2017	<1 GHz	347 Mbps
802.11ax [48]	2018*	2.4 GHz/5 GHz	10 Gbps

Table 2.2 – Summary of Other IEEE 802.11 Standard Amendments

Amendment	Date	Description
802.11c [49]	1998	Details the specification for wireless bridging
802.11d [50]	2001	Extensions to allow for additional regulatory domains
802.11e [51]	2005	Adds enhancements to the MAC layer for QoS
802.11F [52]	2003	Details an inter-access point communication protocol
802.11h [53]	2004	Spectrum and transmit power management extensions
802.11i [54]	2004	Introduces Wi-Fi Protected Access II (WPA2) security
802.11j [55]	2004	For Japanese market, for use in 4.9 – 5 GHz spectrum
802.11k [56]	2008	Radio resource measurement and management
802.11p [57]	2010	Enhancement for wireless vehicular networking
802.11r [58]	2008	Fast BSS transition, to enable continuous connectivity during handoffs between base stations
802.11s [59]	2011	Enhancements for wireless mesh networking
802.11u [60]	2011	Interworking with external networks (non-802 networks)
802.11v [61]	2011	Wireless network management and inter-client device communication
802.11w [62]	2009	Enhanced security for network management frames
802.11y [63]	2008	Allows for use of **802.11a** protocol in 3.7 GHz band in the USA
802.11z [64]	2010	Extensions to Direct-Link Setup (DLS)
802.11aa [65]	2012	MAC Enhancements for Robust Audio Video Streaming
802.11ae [66]	2012	Prioritization of Management Frames
802.11af [67]	2013	Operation in licensed TV bands in the sub 1 GHz space

* Still a draft standard

One core component of the 802.11 standard is the **Power Save Mode (PSM)**. PSM builds on the fact that an 802.11 devices can be in one of the following states: *transmit, receive, idle* or *sleep.* While the 802.11 device is not transmitting or receiving data over the network, the interface on the device can be switched from the *idle* state to the *sleep* state, in order to conserve energy. The PSM enables the automation of this process, in an infrastructure mode network, while maintaining a network presence. This presence can be maintained because the network's AP periodically (generally every 100 ms) broadcasts a beacon frame to the network. The device's WNIC wakes up to receive this beacon frame and can decipher from the content of the beacon frame whether or not the AP has buffered any data packets for it. If so, the WNIC stays awake and downloads the data from the AP before returning to the *sleep* state.

The **IEEE 802.11e** amendment [68] enhanced the default PSM by proposing the **Automatic Power Save Delivery (APSD)** mechanism. APSD reduces the congestion and overhead of PSM by reducing the signaling required for maintaining the wake/sleep timing between the AP and the wireless node. This mechanism has different implementations for two timing structures: *scheduled* and *unscheduled* [69].

- *Scheduled* timing: the AP and the wireless node agree a periodic schedule for delivery of bursts of buffered data.
- *Unscheduled* timing: the wireless node can transmit any type of data packet to the AP, which acts as a trigger to let the AP know that the WNIC on the node is active.

Another recent development in the 802.11 family is **Wi-Fi Direct** [70]. Wi-Fi Direct allows client devices to share content between themselves by creating peer-to-peer (P2P) wireless connections. One device is required to act as a temporary AP for the new Wi-Fi Direct network and then other devices can connect to it. As a result of the infrastructure-mode architecture of Wi-Fi Direct networks, the network can be secured with Wi-Fi Protected Access (WPA) security and the default 802.11 PSM is enacted (along with some extensions specifically for Wi-Fi Direct networks).

2.2.3 Cellular Networks

Cellular networks are used worldwide to provide wireless network connectivity to mobile telecommunication devices, for voice and data communications. Network access is provided to mobile nodes by national network carriers, who control a portion of the licensed cellular spectrum. The standards used for cellular communications have evolved hugely since their

first development. They are generally classified into different *generations,* based on the performance metrics of the standard. The first generation (**1G**) of cellular telecommunication standards used analog radio signals for carrying the transmissions. These networks were only used for voice transmission and did not cater for the transmission of other user data. From the second generation (**2G**) on, the standards used digital signals, which allowed for compression of signals and more efficient bandwidth utilization. This also made data transmissions possible too. The digital generations of cellular technology are now looked at in turn and analyzed in terms of their performance.

2G – GPRS and EDGE

The **General Packet Radio Service (GPRS)** is a packet-switching addition to the **Global System for Mobile Communications (GSM)** standard for 2G cellular communications. GPRS was designed by the European Telecommunications Standards Institute to cater for mobile data communication over cellular GSM networks [71]. Upon release, it immediately stood out from its predecessor, High-Speed Circuit-Switched Data (HSCSD), because of its packet-switching architecture. This architecture enables maximum speeds of up to 171 kbps to be achieved instead of 57.6 kbps with HSCSD [72]. These speeds are possible because of the packet switching nature of GPRS and because it allows the simultaneous use of up to eight GSM time slots and as such is best suited to bursty traffic, such as mobile data transmissions.

The **Enhanced Data rates for Global Evolution (EDGE)** standard [73] was the next version of mobile data network released for GSM. EDGE could offer data rates of up to 384 kbps because of modifications to the pre-existing physical layer of GSM. Irish mobile phone users can still connect to GPRS and EDGE networks but most modern phones on the market support 3G connectivity and some now support 4G networks. However, in areas of poor network signal strength, network providers use GPRS and EDGE as a fail-safe and users' data connections will automatically be switched over to use these 2G networks.

3G – CDMA2000, UMTS, HSPA and LTE

While EDGE fulfills the bandwidth requirements of the International Mobile Telecoms-2000 [74] classification of a 3G network, it is not generally considered a 3G network. There are however two main 3G network technologies. The first, **CDMA2000** or **IS-2000**, is a family of standards that perform **Code Division Multiple Access (CDMA)** to enable data-rates of up to 14.7 Mbps [75]. CDMA2000 networks are not widely deployed in Europe but are primarily

used by North American and South Korean network carriers. Additionally, plans for a 4G evolution of CDMA2000 to the Ultra Mobile Broadband (UMB) standard have been shelved. Qualcomm, who manufacture the CDMA2000 interface chips and were the leading sponsor of the development of UMB, decided that other 4G standards had brighter futures [76].

The second 3G network solution is the **Universal Mobile Telecommunications System (UMTS)** which is widely deployed worldwide [77]. UMTS is a core network that is based on GSM, but it has been completely re-engineered from top to bottom. In Ireland, UMTS has been rolled out with the High-Speed Packet Access (HSPA) protocol implemented on top of the core network. HSPA consists of two component protocols, one for the downlink and one for the uplink: **High-Speed Downlink Packet Access (HSDPA)** and **High-Speed Uplink Packet Access (HSUPA)**. HSDPA provides mobile devices with up to 14 Mbps throughput, and HSUPA allows a throughput of up to 5.8 Mbps [78].

3.5G – HSPA+, LTE and IEEE 802.16e

Building on the HSPA protocol, the **Evolved High-Speed Packet Access (HSPA+)** protocol was released in 2008. Subsequent revisions of the protocol have increased the maximum theoretical data rates of the downlink to 168 Mbps and of the uplink to 23 Mbps [79]. These improvements are achieved by combining enhanced MIMO, modulation and dual-band operation techniques. HSPA+ remains backward compatible with the hardware of previous HSPA networks, meaning that operators could deploy it without upgrading their antennas.

The **Long Term Evolution (LTE)** standard was built as a rethinking of GSM/UMTS networks [80]. It was initially designed to function as a 4G network, but does not meet the requirements for 4G networks laid out in the IMT-Advanced specifications released by the International Telecommunications Union (ITU) [81]. LTE offers a maximum throughput of 300 Mbps in the downlink and 75 Mbps in the uplink. This is achieved by using an all-IP architecture, different multiplexing schemes (frequency-based division multiple access schemes) to the code-based schemes used in UMTS and by also allowing MIMO operations. LTE is not backward compatible with other 3G protocols though and requires operators to install new hardware if they want to provide LTE network access to their customers.

The **WirelessMAN** standard, also known as IEEE 802.16 or WiMAX (from Worldwide Interoperability for Microwave Access) also has protocol implementations for operation on

mobile devices. This began with **802.16e** or Mobile WiMAX which supports data-rates of up to 63 Mbps in the downlink and 28 Mbps in the uplink, when using 10 MHz wide channels, MIMO antennas and advanced coding and modulation schemes [82]. Mobile WiMAX has had a number of commercial deployments worldwide and is known as WiBro in Korea.

4G – LTE Advanced and IEEE 802.16m

The IMT-Advanced specifications by the ITU lay out the performance requirements of 4G networks [81]. In order to be deemed a 4G network, the network must be able to provide a connection speed of at least 100 Mbps to mobile users and at least 1 Gbps to stationary users. Among the other criteria, the networks must be interoperable with existing networking standards, must be based on IP packet switching, be capable of dynamic channel bandwidth and have high spectral efficiency.

There are currently two protocols that have been accepted to conform to the IMT-Advanced specifications; **LTE-Advanced** and **WirelessMAN-Advanced**. LTE-Advanced [83] is an evolution of the LTE standard discussed above. It provides a maximum download rate of 3 Gbps and a maximum upload rate of 1.5 Gbps. In addition, LTE-Avanced also enables more consistent connections than LTE with new transmission protocols and improved MIMO technology. WirelessMAN-Advanced is based on the **IEEE 802.16m** standard [84] which is an evolution of the Mobile WiMAX standard discussed above. WirelessMAN-Advanced increases the maximum data rates to over 1 Gbps using MIMO technologies and channel aggregation.

4.5G – LTE Advanced Pro

LTE Advance Pro [85] is based on the LTE standard and is backward compatible with it. It is a step towards a 5G network but does not meet the required 10 Gbps throughput and less than 1 ms latency thresholds. It does bring LTE up to 3 Gbps throughput and reduces latency to less than 2ms. This is achieved by utilizing the 5 GHz spectrum as well, either with a Wi-Fi carrier or LTE in the unlicensed bands.

5G

No standard has been finalized for a 5G network as of yet. The technical requirements are approximately 10 Gbps throughput on the downlink to a device and less than 1 ms latency [85]. Upload speeds would most likely be less than half the download speed [86]. Even these

requirements have not been solidified fully but are on the agenda for an ITU meeting this year [87].

2.3 Video Codecs

There are many video compression standards in use for the transmission of video content across heterogeneous networks. These include MPEG-1, MPEG-2, MPEG-4, H.261, H.262, H.263, H.264, H.265, VP7, VP8, VP9, AV1, Ogg Theora, Xvid, Dirac, etc. Some of the more historical of these standards will now be discussed, to provide context for the newest standards in use today.

H.261 [88] was one of the first practical video compression standards. It introduced the use of *macroblocks*, which are used to break a video frame into smaller blocks for more accurate encoding of the motion sequences in video frames. H.261 formed the basis for all of the subsequent MPEG and H.26x standards. The **Moving Picture Expert Group (MPEG)** series of standards were developed by the ISO/IEC. **MPEG-1** [89] was the first of these standards and builds on from H.261. The MPEG-1 standard allowed for the compression of video resolutions theoretically up to 4095 x 4095 pixels and of bitrates up to 100 Mbps. Typically, videos were encoded at Source Input Format resolution (352 x 288 pixels), with a bitrate of approximately 1.5 Mbps. The **MPEG-2** [90] video encoding scheme, also known as **H.262**, is quite similar to that of MPEG-1 in terms of its implementation. MPEG-2 adds functionality to allow the encoding of interlaced video and provides a practical platform for encoding video at a large number of resolutions, even up to high definition content. MPEG-2 is used as the encoding standard for DVD and Blu-ray movies and is also used in some countries for broadcast television. **H.263** [91] was originally designed for videoconferencing applications and was used as a partial basis for the development of **MPEG-4 Part 2** [92]. Both H.264 and MPEG-4 Part 2 targeted producing low-bitrate video compression, for video transmission over a network. While H.263 is still used in some mobile video streaming applications, it has largely been superseded MPEG-4 Part 10/H.264.

For mobile video streaming systems, the complexity, the adaptation-capabilities and the widespread use of each video codec, play a huge part in the selection of a codec for the system. As a result of the consideration of these issues, H.264 and VP8 have become the most popular video codecs in use in modern video serving systems. The H.264 and VP8 codecs will now be analyzed in detail and compared in terms of their strengths and weaknesses. In addition, other

existing codecs that could be used for serving video streams are touched upon and emerging video codecs, High Efficiency Video Coding (HEVC) and VP9, are also investigated.

H.264 Advanced Video Coding (AVC) [93] is a video compression standard which was developed by ITU-T and ISO/IEC, and is synonymous with **MPEG-4 Part 10.** H.264 was designed to improve upon the previous MPEG and H.26x standards by reducing the bitrate required for video encoding by half or more, while keeping the algorithm complexity in check. Not only is it prevalent in the video streaming area, it is also the most popular video codec for video conferencing systems; it is used in Skype, Google+ Hangouts and Apple's Facetime. The standard caters for a wide range of video resolutions and qualities, up to 4096 × 2304 px, which can be set statically or adjusted dynamically during the encoding/decoding process. The dynamic adjustment of encoding and decoding parameters was introduced in Annex G of the standard and is generally referred to as **H.264 SVC** (where SVC stands for Scalable Video Coding)[94]. H.264 has gained significant traction in the mobile video consumption sphere in the last couple of years due to the availability of hardware encoder/decoder chips in most modern smartphones [95]. Not only does this offload the video processing from the CPU and make the device more responsive but it also results in an increase in battery-life. These dedicated H.264 processing chips have gone to market in Android, iOS, Blackberry, Windows Mobile and other devices, which means that the underlying technology is pushing developers to use the H.264 codec in their applications [95]. One unfortunate downside of the codec is that it is not open source.

VP8 [96] is part of the WebM project supported by Google, which is an open source attempt to develop a freely available, high quality web video format. While the standard was initially released in 2008, it has not penetrated the mobile device market as successfully as H.264 [97]. As a result, hardware encoders and decoders are not common amongst current mobile devices, meaning all video processing must be completed in software. In addition, the optimization process for the VP8 encoder and decoder is still underway. This means that the codec is not as computationally efficient as it could be and that the H.264 encoder can perform up to 350% faster than the VP8 encoder when creating bit streams of equal quality [98]. Extensive testing in [98], [99] and [100] shows that for video sequences of equivalent bitrate, H.264 consistently outperforms the VP8 codec in terms of video quality. While VP8 has potential to become a widely-used free and open solution for video over the web, the fact that H.264 became so popular before VP8 was available may prevent VP8 from gaining proper traction.

Ogg Theora [101] is another open source video codec that can be used for video streaming applications. However the use of Theora in this context has complications [102]: the stream initialization requires large headers to traverse the network; there is also no support for splitting frames into multiple slices. It has been tested and shown in [103] that the Theora codec cannot match up to H.264 either, in terms of video quality versus bitrate.

Dirac [104] is a wavelet based open source video codec that has also gained some interest in the research community. Unfortunately, tests have shown that the Dirac codec cannot compete with H.264 either, in terms of video quality versus bitrate([105], [106]).

High Efficiency Video Coding (HEVC) [107] is the H.265 standard which was released in April 2013. It is an evolution of the H.264 standard. While the encoding process for HEVC is significantly more computationally intensive than that of H.264, HEVC outperforms H.264 in terms of the **Mean Opinion Score (MOS)** versus bitrate [100]. The standard was aiming to be able to transmit the same quality of video as a stream encoded with H.264, while using only half the bit-rate.

VP9 [108] is an evolution of the VP8 standard. It was released in June 2013, with the goal of halving the bit-rate required for the same quality of video as its predecessor. VP9, like VP8 is part of the WebM Project and is completely open source.

Hardware manufacturers for mobile devices will now be considering which video codec to support in their future designs. This choice will be between the HEVC and VP9 codecs and the result of that decision will most likely determine how successful each of the codecs will be. To have energy and computationally efficient video stream consumption on a mobile device, the decoding of video needs to be performed in purpose-made hardware chips.

AOMedia Video 1 (AV1) is the successor of the VP9 standard proposed by the Alliance for Open Media, to have a video encoding scheme that is royalty free. The alliance consists of a number of different software and hardware companies including Amazon, AMD, ARM, Google, Intel, Microsoft and NVidia [109].

2.4 Adaptive Streaming Protocols

Adaptive streaming is the process whereby the quality of a multimedia stream is altered in real-time while it is being sent from server to client. This adaptation of quality is decided in decision modules on either the client or server. The adaptation may come as a result of weighing different network or device metrics, for example, with a decrease in network throughput, an adaptation to a lower quality of video may improve the playback. Energy relevant metrics can also be considered in order to decide whether an adaptation would be beneficial or not. Energy savings are achieved on the WNIC because less data is being received over the interface. Additionally, this opens up a larger window of free time that can be used to put the WNIC in sleep mode more frequently.

There are currently four main standards that have been widely adopted for implementing adaptive streaming videos to mobile devices. These standards are discussed here along with one other constantly evolving streaming technology.

2.4.1 HTTP Live Streaming

Apple has developed **HTTP Live Streaming (HLS)** [110]. HLS is an open standard which Apple has submitted to the IETF [111]. HLS works by taking a video input and encoding it at multiple different levels of bit-rate. Each of these levels of video are then segmented into multiple sections of uniform playback duration. A mobile client can request a section of video over a HTTP connection. Subsequent sections can be downloaded as required for playback. At any point, the client can switch over to request video sections from one of the other levels of bit-rate. The result is that an adaptive stream is delivered to the mobile device. HLS is currently available on iOS devices since version 3.0. One advantage of HLS is that it is not tied to any single platform for either delivery or consumption. Adobe recently added HLS to Flash Media Server (FMS) [112], so that FMS can also be used to delivery video streams to devices that are not Flash-enabled. Google has also added HLS to Android 3.0 [113] which makes it a ubiquitous solution.

2.4.2 HTTP Dynamic Streaming

As well as supporting HLS, Adobe has their own dynamic streaming solutions: **RTMP Dynamic Streaming** and **HTTP Dynamic Streaming**. RTMP Dynamic Streaming does not require any segmentation of the video streams but it does require the use of Flash Media Server

for the stream delivery. HTTP Dynamic Streaming [114] can be served from Flash Media Server or any other web server. This approach works in a similar way to HLS in that it requires segmentation of the video stream before transmission. Both these solutions would be played in a Flash player or an Adobe AIR application. While flash players are available on many platforms, requiring their utilization is quite a limiting restriction. Any development of streaming applications would have to be done using proprietary Adobe software. In addition, support for Flash on mobile devices is quite poor; it is no longer supported on the latest versions of Android and was never supported on iOS.

2.4.3 Smooth Streaming

Notably, Microsoft has developed a solution called **Smooth Streaming** [115] which is part of their Silverlight system. Smooth Streaming uses HTTP as the delivery protocol for its streams. Although this mechanism is not compatible with Android phones, Microsoft have released solutions for both iOS and Windows Mobile. One important thing to note is that developers do not have complete access to the configuration of the adaptation algorithms [116].

2.4.4 Dynamic Adaptive Streaming over HTTP

Dynamic Adaptive Streaming over HTTP (DASH) was developed by the Moving Pictures Expert Group (MPEG) and standardized as ISO/IEC 23009-1 [117]. It supports the adaptive transmission of segmented MPEG-4 files and MPEG-2 Transport Streams. The available quality levels for specific video sections are advertised in XML documents enabling the player to select a specific video quality or bit-rate to stream. There is also an open source, platform independent library available for interfacing with the DASH standard. This is called libdash [118].

HTML5 [119], relies on the DASH format for delivery of adaptive video content. This is still not hugely widespread with most DASH solutions being implemented in JavaScript to ensure browser compatibility. The primary reason for this is that no royalty-free video standard has been adopted in all modern browsers. There is still significant fragmentation in support for different video encodings in internet browsers. This could change once AV1 is finalized as this should fit the video encoding standard gap perfectly.

2.5 Video Quality Assessment

For any video delivery system, it is important to be able to quantify how well it performs. This can be done by assessing the **Quality of Service (QoS)** and the **Quality of Experience (QoE)** levels.

QoS in video streaming applications is defined as the application's level of ability to deliver the video data across the network from the server to the client device. In the ITU-T G.1010 Recommendation [120], it is explained that QoS is affected by the underlying network conditions in the system, e.g. throughput, delay, jitter, packet loss, etc.

QoE, on the other hand, refers to the user's perception of the service they receive while watching the video stream on the client device. QoE will be affected by the QoS but can also be affected by a number of external factors too, which would not affect the QoS, e.g. the user's environmental conditions, hardware limitations, the user's activities while watching the video, monetary cost of the service, etc. [121].

Considering that a video streaming application provides a service to a real user, the most important goal for the assessment of that application is to quantify the user's perception of how well the application performed, i.e. QoE. This can be achieved by two approaches: directly measuring the QoE with subjective methods and indirectly approximating the QoE using objective QoS methods. Subjective methods require asking users to rate their experience of the video sequence, while objective methods involve formulaic calculations of an assessment metric which can then be used to infer an approximation of what a subjective response might be. Subjective methods are more accurate for assessing the QoE because they allow real users to rate real experiences. The downside is that performing subjective tests is a lengthy process and as such is not viable for real-time applications. Objective QoS methods however, can be used to get an indication of the QoE instantaneously, making them useful for quick assessment of a video stream. Objective QoS methods do not have a 100% correlation with the users QoE though, as they cannot account for all of the factors that influence the QoE. However, they can give a good indication of how the QoE might be affected by certain changes in the service. The different metrics for objective QoE approximation are discussed here, followed by the subjective alternatives.

2.5.1 Objective Methods

Peak Signal-to-Noise Ratio (PSNR) is an objective metric for comparing two images (or video frames) [122]. PSNR is the most frequently used metric for approximating the user perceived quality of video streams in the absence of subjective testing. In the case of video streaming applications, the PSNR is calculated by comparing the original video from the server-side with the received video on the client-side. In this respect, PSNR is a full reference metric because it requires both the original and transmitted video for its calculations. The comparison is performed on a pixel-by-pixel basis in the corresponding frames of the reference video and the received video using Equation (2.1). The comparison can consider each of the color channels in the reference and received frames, but normally just the luminance of each pixel is used in the calculations. This is because the human eye is generally four times more aware of changes in luminance than in color [123]. In (2.1) MAX stands for the maximum intensity level of a pixel, which for an image with 8 bits per sample is 255. MSE stands for the Mean Square Error as defined in Equation (2.2). In (2.2) *m* and *n* refer to the number of rows and columns of pixels in a video frame respectively. *I* and *K* refer to the luminance values for each individual pixel in the reference and received video frames respectively. In the case where there are two identical video frames on both the sender and receiver side of the streaming application, the MSE is equal to zero. This means that the PSNR value cannot be defined because the formula attempts a division by zero. To combat this situation, the general practice is to set a cap of 100 dB on the PSNR value. In the case where a video frame is lost over the network, there is no frame to compare against. The general practice for this situation is to set the PSNR value to 0 dB.

$$PSNR = 10 \times \log_{10}\left(\frac{MAX^2}{MSE}\right) \tag{2.1}$$

$$MSE = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}\left[I(i,j) - K(i,j)\right]^2 \tag{2.2}$$

Structural Similarity Index (SSIM) is another full reference video quality metric [124]. It aims to improve upon PSNR by considering that errors will be more noticeable by the human eye in pixels that are spatially close to each other. The similarity index is compiled by assessing three features of the videos and combining the results: the luminance, contrast and structural similarities.

Multi-scale Structural Similarity Index (MS-SSIM) is an extension of the SSIM metric above [125] that allows for the abstraction of the contrast and structural comparison of the reference and received videos for multiple lower resolutions of video. This is then combined with the luminance comparison for the highest level resolution of the two videos in order to provide a more accurate approximation to how the user perceives the video. The increased accuracy comes because the multi-scale comparison adds realism to the calculations of the QoE, which in a subjective test, may be affected by the display size of the video, the user's distance from the screen or even the user's eyesight.

Visual Signal-to-Noise Ratio (VSNR) [126] is a two stage full reference video quality algorithm. It first uses a wavelet-based approach to detect distortions in a received video frame. If the level of these distortions is below a threshold, the distortions will not be visibly noticeable and the received video frame is "deemed to be of perfect visual quality". If the distortions are ruled to be visibly noticeable, the VSNR metric proceeds to classify the received frame by assessing its visual fidelity in terms of the perceived contrast and effect of the distortions.

Video Quality Metric (VQM) [127], also known as the National Telecommunications and Information Administration (NTIA) General Model, performs feature extraction and filtering in order to compare aspects of the videos that particularly affect the visual perception. The features analyzed by this algorithm are spatial loss, blurring, tiling/blocking artifacts, color differences and transmission errors.

Visual Information Fidelity (VIF) [128] is derived from the combined insights offered by three types of models. Statistical modeling of natural images is performed in the wavelet domain using Gaussian scale mixtures. Modeling of image distortions is also performed in the wavelet domain, by using signal attenuation and additive noise. Finally, modeling of the Human Vision System (HVS) is also performed.

MOtion-based Video Integrity Evaluation Index (MOVIE) [129] is a video quality assessment metric that analyses the spatial, temporal and spatio-temporal components of a video's distortion in order to approximate the QoE. MOVIE incorporates computing the motion vectors of components in a video sequence and quantifying the effect of any distortion in the video on the overall fidelity of the motion vectors.

2.5.2 Subjective Methods

The most common metric for subjectively assessing a user's QoE in video streaming applications is the **Mean Opinion Score (MOS)**. MOS is calculated by showing the user a video sequence and then asking them to rate different aspects of the video watching experience from 1 and 5, where 1 correlates to "bad" and 5 correlates to "excellent". These tests take time and results can vary widely depending on the external factors that also affect the QoE. In order to obtain results that can be compared between different tests and different researchers, there have been a number of recommendations published to standardize the testing procedures and environments.

ITU-T P.910 [130] and **P.911** [131] are two recommendations proposed by the ITU to meet the goal of standardizing the non-interactive subjective testing procedures for video quality assessment in multimedia applications. They make suggestions on a number of different aspects of the testing. For example, it is recommended that at least 4 different video sequences with varying spatial and temporal complexities be shown to the test subject in order to assess performance for different types of video and to keep the test interesting for the user. The video sequences should be approximately 10 seconds in duration. A number of recommendations are made about the testing environment too, for instance, the screen brightness, the ambient light level in the room, the listening level for the audio and the background noise level. It is also recommended that the viewing distance be kept the same for all test subjects. In addition to the video and environment recommendations, the other main aspect of the recommendations in these documents is that the whole testing procedure be explained to the test subjects before the test begins.

ITU-T P.920 [132] was published to recommend a standard approach to interactive test methods for audiovisual communications. This deals with primarily with bi-directional video communications, i.e. video conferencing, and outlines a procedure for assessing the quality of the terminal and communication link performance during communication using the system.

ITU-R BT.500 [133] is a recommendation for testing procedures and environmental configurations for assessing the quality of television pictures. It outlines similar environmental criteria and configurations as outlined above in ITU-T P.910 and P.911. These ensure consistent testing conditions in order to reduce the chance of bias in the results. Additionally, BT.500 outlines different comparative methods for the presentation of the videos to the test subject. These include the Double-Stimulus Impairment Scale, where the original reference

video is shown to the test subject, immediately followed by the test video sequence. This gives the test subject the ability to see the quality of the original video and compare the quality of the test video against this. BT.500 also lays out the grading scales that should be used by the test subjects for each of the different comparative methods.

2.5.3 Metrics used in this Thesis

In this thesis, the approximation of the QoE via objective methods is performed using PSNR. PSNR was selected for the following reasons:

- It is the most widely used objective video quality assessment metric in the research community
- It is an open standard, not proprietary like some of the other objective metrics. In addition, it is low in complexity which means that real-time computation is possible
- It has an acceptable correlation to MOS for video tests on mobile devices, as seen in Table 2.3
  - The data in this Table compares the computed values of seven objective video quality assessment metrics against user feedback from subjective tests. The higher the number, the higher the correlation of the two values.
  - While some other metrics achieve a higher correlation to the MOS from the subjective tests, the PSNR metric performs reasonable well for tests on mobile devices, outperforming some more complex video quality metrics.
  - The authors in [134] discovered that their test subjects were more forgiving of bad video quality levels on smaller screen devices and also that some of the single-scale algorithms (SSIM and VQM) may not predict the quality of the video as accurately on smaller display devices.
  - PSNR, on the other hand performed better on smaller screen devices.

Table 2.3 – Correlation of Objective Metrics to the MOS

Metrics	Performance SROCC LIVE Database [135]	Performance SROCC LIVE Mobile Database [134]
PSNR	0.3684	0.6780
SSIM	0.5257	0.6498
MS-SSIM	0.7361	0.7425
VSNR	0.6755	0.7517
VQM	0.7026	0.6945
VIF	0.5710	0.7439
MOVIE	0.7890	0.6420

- Considering that PSNR is so widely adopted and that its algorithm is implementable for real-time feedback in systems, it is a good choice for the tests required in this thesis.

For subjective tests, the recommendations from ITU-T P.910 and P.911, and ITU-R BT.500 were considered.

2.6 Summary

This chapter introduced some of the more important background technologies in the mobile device and video streaming space. In particular, the various wireless networks available on modern mobile wireless devices are explored. The different encoding and transmission protocols for adaptive video streaming are then described and the methods for video quality assessment are also detailed.

The next chapter moves on from the existing standards and examines the state-of-the-art research in energy efficient mobile device operations, video adaptation and multi-hop networking.

Chapter 3 - Related Work

In the last chapter, existing standards for multimedia streaming and teleconferencing applications were introduced. This chapter presents the state-of-the-art research projects in fields related to the work of this thesis. To tackle the problem outlined in Chapter 1, it is imperative to show the state of current research solutions before presenting the novel solutions of this thesis. To provide a detailed overview of the state-of-the-art in the literature, related research projects have been classified into four main categories: energy-aware mobile devices, energy-aware wireless communications, adaptive data delivery and intelligent algorithm design. These will now be discussed in the remainder of this chapter.

3.1 Energy-aware Mobile Devices

3.1.1 Specialized Hardware

In a smart phone/high end mobile device, there are several specific hardware configurations that can be exploited to achieve energy savings. Bahl *et al*. [136] investigated a technique using a low-power radio in conjunction with a regular 802.11 WNIC. One benefit of this mechanism is that it allows a mobile device and its radio to be powered off, while the low-power radio maintains a network presence that can be used to wake the device up, in the event of any data reception. One example where this approach would be effective is voice over IP (VoIP), where a user can maintain an online presence with minimal bandwidth and only requires a high-speed connection while interacting with another user. The issue with this approach is that it can only work if there is a low-power radio available on the mobile device. While the Bluetooth interface, which is common among mobile devices, could be used, the range for Bluetooth is typically about 10 meters. Other low-power radio interfaces would work instead, such as ZigBee, where the transmission range is between 10 m and 75 m. The fact that additional hardware is required for this solution rules out most existing mobile devices though.

Most newer smart phones now support Multiple Input Multiple Output (MIMO) wireless interfaces in either their IEEE 802.11n/ac WNICs or in their cellular interfaces (LTE has support for MIMO interfaces for example). This allows network presence to be maintained with these interfaces, using only one of the antennas in the WNIC, as in [137] described for IEEE 802.11n. The rest of the antennas in the WNIC can be powered down in this scenario

and only powered up for support high bitrate data transmission and reception. Tests in [137] measured the power consumption for each of the antennas in the WNIC to be 330 mW, so disabling some of these antennas dynamically can yield significant power savings.

Another example is the use of hardware acceleration in video decoding. Adobe Flash Player [138] has utilized this functionality since version 10.1. The player uses both hardware H.264 decoding and hardware graphics rendering using the Graphics Processing Unit (GPU) in the device. The main benefit is that by offloading this processing from device's CPU to purpose designed hardware, the GPU, the performance and energy efficiency increase dramatically. This is supported natively on the OS level on modern smart mobile devices. In Android, hardware decoding of multimedia streams can be performed with the MediaCodec [139] class. For iOS, the Apple's AV Foundation [140] enables the same functionality.

3.1.2 Dynamic Screen Control

The display screen is one of the largest energy consuming components on a mobile device. It also yields the largest range for energy saving possibilities [141]. Over the years, different techniques have been proposed to adjust display resolution, contrast, color-usage and screen brightness based on human factors to reduce display power consumption. Cheng *et al*. [142] proposed a Concurrent Brightness and Contrast Scaling (CBCS) technique that aims at conserving power by reducing the backlight illumination of LCD screens, while retaining the image fidelity through preservation of the image contrast. The proposed technique yields a 3.7X power saving in the device's screen with only 10% of contrast distortion. This does not cater for OLED-based displays however and only considers static images in its testing so would not be relevant for multimedia streaming applications.

In terms of video streaming, the initial research works considered the video stream to comprise of a series of image frames and dynamically changed the backlight by applying backlight-scaling techniques to each image frame individually. In [143], the authors managed to achieve a 16.4% power reduction (measured on the CPU, Image Processing Unit and screen of a device) during the playback of an excerpt of a movie. The side-effect of their approach is that the backlight level may change notably across consecutive frames, which in turn would result in flickering effects during video playback.

In order to solve this, the work in [144] determines the backlight level for an image by considering the preceding frames' pixel values and backlight levels. This technique allows the backlight level to be manipulated while reducing the user perception of these changes. In experimental tests with 6 different video sequences (from 3 movies), the authors found that the temporally-aware backlight scaling algorithm did not yield as high energy savings as the original approach. It did, however, reduce the temporal distortion of the video sequences by approximately 25%, in comparison to the original approach. Power savings of up to approximately 26% were achieved on the components used in the LCD screen, which only equated to approximately 6% of the total device power consumption. The work in this paper did not perform any other manipulations on the device or to the video sequence so has been extended in this thesis.

Cheng *et al.* subsequently revisited their earlier work to consider video playback applications in [145] and proposed a different solution for smoothing backlight transitions. The solution proposed by the authors there is to perform a low-pass filter on the video frame to prevent multiple backlight manipulations in the same scene. Additionally, the backlight is only changed to one of 5 levels. This prevents fine-grained changes happening more frequently. Experimental tests were performed using a Compaq PDA and the proposed technique resulted in power savings of up to 40% on the components used for the LCD display. This solution is not temporally-aware though, so could still introduce some flickering effects, given the correct test scenarios. This solution also requires additional computation on a proxy server to calculate the backlight level to use from the video sequence. This information is then sent on to the mobile device along with the video data, and must be interpreted there for dynamically changing the backlight level. The extra requirements on the server side increase the complexity of the solution and limit immediately it could be deployed.

Hsiu *et al.* proposed a similar mechanism for dynamic backlight scaling for mobile streaming applications, using an external server for processing the video content and calculating the backlight level [146]. Their dynamic programming algorithm reduces the power consumption of the backlight by up to 31% when browsing YouTube videos. This approach only targets LCD displays however so is not applicable to many modern mobile phones, which use OLED based screens.

In [147], Yan *et al.* designed a similar algorithm to [146] but the streaming server stores different versions of the video which have been encoded at different bitrates and also with

compensation for different levels of brightness. In this way, the streaming client requests a specific bitrate and brightness level of video from the server and then displays this with the appropriate brightness settings on the device. The authors measured total device power savings of 19% on average with comparable a QoE to that seen in basic adaptive video streaming applications.

Liu *et al.* [148] proposed a variation on the above techniques where the calculation of the backlight and luminance compensation is moved from the server-side to the mobile device itself. This means that there are no extra components required on the video streaming server or at a video proxy server, so the solution can be deployed relatively easily. The side-effect of moving these processing steps to the mobile device is that the inherent power consumption of each step is now coming from the mobile device's battery. This is offset by the authors by reducing the amount of the video that is used for their calculations and also by using the GPU, instead of the CPU, for performing luminance compensation during the video playback. Experimental tests on 4 Android tablet devices shows that their technique can yield power savings for some types of videos. The number of videos where power savings can be achieved using this technique changes greatly depending on the settings for the acceptable level of pixel distortion in the video and also depending on the device model being tested. It was found that power savings of over 10% could be achieved on some devices with specific video sequences.

Different screen technologies have very different energy consumption characteristics. Unlike LCD screens, OLED-based displays do not require a backlight as their pixels are light-emitting. In an LCD display, the backlight accounts for most the device's power draw. For OLED devices, energy consumption depends on intensity and chromaticity of each pixel being displayed. As a result, OLED devices generally consume very little power when displaying black pixels. However, they consume far more than an LCD screen when displaying white pixels at maximum brightness, on average. Of course, the exact performance will depend on the specific display, but this is the general trend of the OLED and LCD technologies.

In [149], while Shin *et al*. proposed a technique based on dynamic voltage scaling (DVS) of the OLED panel. The proposed method saves power in the driver transistor and the internal resistance with an amplitude modulation driver and a pulse width modulation driver, respectively. This technique elicits similar power-saving results as scaling the backlight on LCD based displays. In experimental tests, using two static images, instead of videos, power savings of up to 52.5% were achieved on the OLED panel. This work is complementary to the

techniques proposed and implemented as part of this thesis. One thing to note is that, support for DVS of the screen would need to be provided on a hardware/driver level within the mobile device.

Dong *et al*. performed significant research on manipulating the colors of pixels on OLED screens in order to conserve energy on mobile devices [150]. The first step was to create a model for an OLED device so that an optimal solution could be achieved. This involved devising a device independent mechanism for assessing, pixel-by-pixel, the energy required to display each of the available colors on the OLED screen. This is quite a long and computationally intensive process to be performed in an iterative fashion. Hence, the authors created a shortened and simplified approximation algorithm. The new algorithm decreased the computational cost of the calculation by 1600 times while still achieving 90% accuracy.

Having devised the algorithm above, Dong *et al*. then investigated using color transformations. In [151], the current colors of the different Graphical User Interface (GUI) themes are assessed in terms of their energy efficiency. They are then altered to different colors, which maintain the overall contrasts of the different colors on the page, while providing significant energy savings. Energy reductions of over 75% were achieved on the display while still showing a GUI to users with acceptable visual quality. In [152], the same authors succeeded in combining all of the work from their previous two papers in order to create a fully functional Android application. The application creates an energy-color model of the device's OLED screen and then uses it to perform color transformations to websites. The application, *Chameleon*, reduced the total system power consumption by over 41% during web-browsing. While Chameleon is very effective for static pages, it does not consider applying the same process to other types of media at a lower level in the OS. Color transformations would not be efficient in the application layer for computationally intensive multimedia streaming applications, so this solution would need a significant overhaul to add support for applying the same process to multimedia content.

In [153], Kim *et al.* have capitalized on how humans perceive color in order to reduce the power consumption of OLED-based screens on smart mobile devices. The authors found that the palette of colors to display on the device could be reduced without the average human eye being able to perceive the difference. In this way, the colors of each pixel could be changed imperceptibly in real-time in middle-ware on the device to be one of a subset of colors, that have a lower power consumption than the original color. The proposed color transform method

is called *Blind*. *Blind* can be performed with varying degrees of aggression, and for video streaming applications resulted in power savings of just over 2.5% at the most aggressive level. The approach used in Blind could be complementary to the work presented in this this thesis and would be interesting to investigate in future work, to see how it could be combined with the work in this thesis to provide further power savings.

Anand *et al.* proposed a non-linear tone mapping technique to dynamically increase the image brightness for gaming applications [154]. Though their focus was on adaptation while playing games, the same principle could be extended for video streaming as well. A significant insight that can be derived from these approaches is that the power consumption of OLED screens can be reduced adaptively, depending on many factors, e.g. gamma correction, screen brightness and chromaticity. An important point to be noted is that techniques like backlight optimization, dynamic voltage scaling, etc. can be performed dynamically in real-time, based on the application and the nature of the content.

3.1.3 Adaptive Decoding

Adaptive decoding refers to operations on the client device that alter the default decoding process in real-time in order to maximize energy efficiency. This can involve simplifying the decoding process or skipping specific video frames instead of decoding them. This process inherently lowers the QoS and QoE but also increases battery life.

Scalable Video Coding (SVC) is an extension to the H.264/MPEG-4 video standard. It details a mechanism for decoding an SVC video stream dynamically at one of multiple quality levels [155]. The dynamic scaling is achieved through any combination of three scaling mechanisms:

1. Temporal Scalability: Changing the frame-rate of the received video stream by dropping whole frames. In MPEG videos, B-Frames can be dropped without affecting any of the previous or following frames in a Group of Pictures. In [156], Yu *et al.* propose an algorithm for scaling the frame-rate of a video sequence during the video decoding process for reductions in decoding time and power consumption. The algorithm assesses the level of movement between the immediately preceding decoded video frames. If the level of movement between the previous frames is above a certain threshold, then the current frame is decoded as normal. However, if the level of movement is below the threshold, then the current frame can be discarded without decoding it and its reference frame is displayed again. This yielded a reduction in

decoding time by up to 35.9% while the PSNR of the video dropped from 37.44 dB to 33.12 dB

2. Spatial Scalability: Dynamically changing the resolution of the video. Functionality for decoding a video stream at a number of different resolutions is catered for in SVC. The client device that receives the stream decodes the lowest resolution version of the video first, decoding higher resolutions subsequently if required.
3. Quality Scalability: Changing the quantization parameter for each macro-block in the video decoder. Park *et al*. achieved a 42% decrease in energy consumption during video-decoding with a mere 13% quality degradation (measured with the PSNR metric) in the video by exploiting this method [157].

Dynamic video decoding is not included in the work presented in this thesis but these would all be complementary techniques to the techniques proposed here.

3.1.4 Dynamic Voltage Scaling (DVS)

In [158], Yang *et al*. proposed an algorithm for dynamically scaling the voltage supply to a mobile device's CPU. The decoding time for each video frame is predicted and used to select a frequency level on the CPU that will successfully decode a certain ratio of frames in time for presentation on the screen. This algorithm resulted in a reduction of system-wide energy consumption by up to 17% over other DVS mechanisms and up to 24% over non-DVS mechanisms. The testing for this paper was performed on a laptop with a 1.7 GHz Intel Pentium M processor, which may not provide a relevant picture for smart-phones and PDAs where the processors are less powerful to begin with.

Yang *et al.* present a different DVS scheme, the Low Overhead Optimal Schedule for Realistic CPUs (LO-OSRC), for mobile devices in [159]. The benefit LO-OSRC is that the scheme is actually tested on ARMv5 processors that were designed for smart-phones/PDAs. LO-OSRC periodically samples the instantaneous CPU usage and computes which of the processor's available clock rates would be best suited for handling the current tasks. The scheme considers, and minimizes, the overhead involved in implemented the algorithms too, which is another benefit of LO-OSRC. The energy savings achieved were up to 9.4% over another state-of-the-art DVS scheme and 15% over a non-DVS mechanism. While these results look promising, LO-OSRC is not analyzed in a number of different usage scenarios, e.g. for video

streaming or for web browsing, etc. As a result, it is not easy to say how complementary the proposed algorithm would be for the applications assessed in this thesis.

Another DVS algorithm is proposed in [160] for power management in a specific application-type, mobile games. Where DVS techniques for video applications try to predict the voltage level required for decoding a video frame from data collected while decoding previous video frames, a different approach is presented in this paper. Here the authors use the structural information that is available for each of the game frames (information that is not available for video streaming applications, without additional processing of the video on the server-side). In experimental tests on a Windows computer, a 50% improvement in the number of frames meeting their deadline was achieved with no increase in power consumption. Without the additional meta information being available for video frames, this approach would not be applicable for mobile video streaming applications.

3.2 Energy-aware Wireless Communications

3.2.1 Wireless Interface Sleep Mode

The IEEE 802.11 standard outlines a built-in Power PSM when operating in infrastructure mode. A simple energy saving technique is to put the WNIC of a device into sleep mode when it is not in use [161]. However, it is not always feasible and in fact, the savings depend on the application in use. Multimedia streaming and video-on-demand applications have different QoS requirements in comparison to traditional data transmission applications because of tight time constraints on both the network and the device. Another application that has some timing requirements similar to this is an RF localization app, where a device uses its Wi-Fi interface to calculate the position of a mobile device relative to some fixed devices. Saidi *et al.* [162] have proposed a battery-aware localization mechanism for wireless networks wherein the trade-off between saving energy in a wireless node by lengthening the sleep-cycle period is investigated while still allowing it to perform accurate localization calculations. The main issue with this scheme is that if it is employed in an application with stricter time constraints, such as a multimedia streaming application, it inherently leads to a significantly lower QoS unless the device knows exactly when it will receive data.

Namboodiri *et al.* proposed the GreenCall algorithm which allows the network interface of a client device to sleep during VoIP calls [163]. GreenCall dynamically calculates the sleep

interval of the WNIC by comparing the latency involved in the transmission and decoding of packets, to the packets' play-out deadlines. The duration between these two figures is the maximum possible sleep duration. The benefit of implementing this approach is that energy savings of up to 80% can be achieved in the network interface while keeping the loss rate below 2% for a real-time application. GreenCall does introduce an energy-overhead in measuring the network latency with ICMP packets and the calculations required for computing the sleep interval time. This is not explicitly accounted for in the results presented. The testing in this paper was also performed with a laptop. The implementation of GreenCall on a smart mobile device would be very interesting to test, particularly if it was enhanced to support video traffic too.

SleepWell [164] is a modified version of 802.11's PSM. It enables multiple APs in a network, which would normally have overlapping beacon-periods, to readjust their beacon intervals in order to eliminate unnecessary network contention. The result is that in a test-bed of 9 Nexus One devices, across a variety of real online applications, energy savings from 38% to 51% were achieved. SleepWell is compatible with existing 802.11 client devices (it only requires modification to the APs). Unfortunately, there is an inherent oversight in SleepWell which is that introduces latency and has not been optimized for any real-time applications, such as VoIP or Video Conferencing.

3.2.2 Energy Efficient Network selection and Handover

Most mobile devices are currently shipped with multiple heterogeneous wireless network interfaces, such as Wi-Fi, UMTS, GPRS and Bluetooth. Each of these networks has different energy consumption characteristics. For instance, in [165] measurements show that the energy consumption per unit time of communications over a UMTS and IEEE 802.11b/g are similar. However, the energy consumption as a function of the data transferred can be up to 300 times larger over the UMTS network interface. Trestian *et al.* [166] investigated new techniques in order to exploit these energy characteristics. In this paper, the authors selected the best available network for the client device to connect to, based on the energy, quality, monetary cost and user mobility characteristics for network. This approach is particularly useful for horizontal handover between two networks that a device has access to over a single network interface. However, in the case where a device is extremely limited in terms of battery-life, the authors do not consider the possibility that it may be more energy efficient for the device

to disable all but one of its network interfaces, instead of constantly scanning for the most energy efficient network with all its WNICs.

Mahkoum *et al.* take another approach and propose a power management framework which enables a device to maintain a network presence across multiple heterogeneous networks while powering-off all but one of the network cards on a device [167]. This is similar to that discussed in [136] but would work with the heterogeneous network interfaces that come with most modern mobile devices and does not require specialized hardware. The efficiency is achieved by utilizing proxies on each of the heterogeneous networks to feign the connectivity of the device's network interfaces. If a connection is made through the proxy for any of the sleeping network interfaces, the proxy contacts the device's active interface, which in turn wakes up the interface required.

In [168] a detailed comparison of the use of 802.11 wireless networks and of Bluetooth networks for energy efficient data communication is provided. For all transmissions in infrastructure mode networks, Wi-Fi was measured to be more power efficient than Bluetooth (efficiency was measured as KB/mWs) because of the PSM in the IEEE 802.11 standard. If a version of the PSM was implemented for *ad hoc* 802.11 networks, then Wi-Fi would always be more efficient than Bluetooth. The measurements were conducted on real devices with modern components and modern OS. This is a Cross-Layer Throughput-to-Power Optimization that limits the transmission speed from an AP to a mobile device so that packets need not be dropped on the device, which results in significant power-savings. It would have been interesting if Wi-Fi Direct and IEEE 802.11n had been included in this testing.

In [169], an algorithm is proposed for vertical handover between heterogeneous wireless networks in the case where a device has multiple network interfaces. The algorithm balances the load among the attachment points (base stations and access points) and also maximizes the network lifetime. The routing algorithm caters for operation in both infrastructure mode and *ad hoc* networks; however, in *ad hoc* mode, the route calculation does not use cooperative information in order to ensure that the minimum power route is selected.

3.2.3 Energy Efficient Multi-hop Transmissions

When a mobile device is transmitting data, the transmission power of its WNIC needs to increase as the transmission distance increases, in order to ensure successful transmission of

the data. This is caused by the attenuation associated with radio transmission. One method of lowering the power consumption of the WNIC is to simply transmit the data to an intermediary node which can then relay the information to the destination either directly or through other intermediary nodes. This is called multi-hop networking.

In [170] Ibrahim *et al*. proposed the novel Minimum Power Cooperative Routing (MPCR) algorithm which constructs a route from source to destination in an *ad hoc* wireless network. This MPCR route is the minimum-power route in the network which guarantees the required throughput. This algorithm yielded a 65.61% power saving over previously existing cooperative algorithms in a linear network and 29.8% in a grid network. However, MPCR does not consider that intermediary nodes in the route may also have energy constraints. This would need to be incorporated into the algorithm to ensure fairness in selecting a route and also to increase network lifetime.

Siam *et al.* presented a network protocol for multiple antenna devices, where transmission power is minimized for CTS and ACK packets [171]. The overall energy savings are increased by exploiting multi-hop routing, which yields up an additional 61% energy reduction over the direct transmissions. Transmissions can be sent further with lower transmission power rates as a result of the compensation of MIMO technology. Unfortunately, the scheme only considers devices with multiple antennas and does not investigate the use of other technologies apart from Wi-Fi for hops. For example, it would be interesting to measure the energy savings if the first hop from the mobile device was performed as a Bluetooth connection, provided the required throughput was low.

[172] illustrates that energy savings can be achieved in routing calls through a CDMA cellular network using multi-hop communication over single-hop communication. The energy was measured on the electronics of the network interface on all of the devices involved in the multi-hop transmission and energy savings are dependent on the number of hops in the communication. It was observed that the energy consumed decreases with the increase in number of hops until a certain point and then increases again. This happens because for a set total transmission distance, the power consumption involved in adding an extra network interface (hop) to the transmission chain outweighs the power that can be saved on each device. The multi-hop connection was shown to be most effective in environments with high path-loss and when power control is enabled. Additional latency as a result of the multiple

hops is not considered directly in this thesis, though any effect to the QoE of the video playback is measured.

Transmission Power Adaptation: The Transmission Power setting of a wireless card is used to configure the gain applied to a device's radio antenna, during data transmission. This gain level dictates the Signal to Noise Ratio (SNR) of the transmission and thus the range of successful transmissions. While setting the device to the maximum transmission power level will result in the data being successfully transmitted over a larger distance, this is not always an ideal solution. A high transmission power directly results in a high power consumption level on the device's battery. Setting the transmission power to the lowest level can also result in increased power consumption. This can occur when the SNR becomes so low that loss on the network rises. This in turn can require a higher level of packet retransmission, depending on the application and transport layer protocols.

Lu *et al.* [173] proposed a mechanism for optimizing the process of encoding and transmitting H.263 video over wireless links. This mechanism set the INTRA frame frequency in the encoding process, dynamically managed the channel coder and also adapted the transmission power on the device in response to information about the video communication link. The authors were able to prove that for successful transmission of video across a wireless network, the transmission distance greatly effects the power consumption of the sender device. For larger distances, the transmission power must be increased to ensure delivery of the data. This in turn increases the power consumption on the device. While this thesis provides very useful insights, it does not consider the use of multi-hop paths for saving power. Additionally, the tests were performed with low resolution video sequences and laptop computers, not smart-phones/tablet devices.

In [174], a novel cross-layer, state-machine based algorithm is presented for limiting the loss rate of important video frames in a H.264 stream while keeping the device power consumption stable. The algorithm aims to maintain a stable level of power consumption on the device while increasing the QoS of the stream received by the viewer. This is achieved by configuring a higher transmission power for the more important video frames (i.e. Intra frames) in the video sequence. In addition, a feedback mechanism is used to allow dynamic control of the transmission power so that it can react appropriately to packet loss on the network. The simulation results show that for a minor power-consumption overhead, the QoS of the received

video can be increased slightly. These test scenarios have not been attempted on real-world devices however and do not include any provision for multi-hop routing either.

Multi-hop Paths: When transmitting data over large distances, the transmission power of the wireless interface needs to be set high, so that the data will be received. Multi-hop routing can be used to circumvent this issue by allowing the sender to transmit its data to a nearby node, which then forwards the data on to the destination node. The benefit of this approach is that the transmissions take place over shorter distances and as a result, do not require as high a transmission power level. By extending this idea further, with more intermediary hops in the transmission path, the range of the wireless network can be increased greatly too.

Multi-hop routing creates some additional challenges however. For example, why would a mobile device agree to relay data for another device? The most common solutions to this issue are to provide a reputation-based [175] or credit-based [176] incentive to the relay device, so that they benefit from helping other devices with their transmissions. Some incentive schemes propose a combination of the reputation and credit-based approaches as seen in [177]. These incentivization issues are not addressed directly in this thesis but it is assumed that they would already be in place on the multi-hop network.

Another issue that is inherent in using multi-hop paths is the selection of the specific path and number of hops to use. In [178], Banerjee *et al.* prove that any energy-aware routing policy should consider the error rate and the probability of retransmissions in order to calculate the true cost of a multi-hop path. By considering the transmission error rate of all hops in their transmission path, the authors were able to achieve a 70% power reduction over other minimum-energy routing protocols. The proposed routing protocol does not consider any specific traffic type or delivery constraints, so does not assess how network latency would affect the playback of a video stream, for instance. This information is crucial for the selection of an appropriate energy-aware path.

3.3 Adaptive Data Delivery

3.3.1 Mobile Video Broadcasting Platform Architectures

There are number of different architectures used for live video broadcasting applications. The first is the direct streaming approach. IP Webcam [179], for instance, serves the video directly

from an Android device, without using a video server platform. This approach functions perfectly when transmitting video to one or two people, video conferencing for instance, but one limitation is that the system is not very scalable. As the number of people watching the video broadcast increases, more strain is placed on the mobile broadcasting device.

A solution to this issue is to have a cloud-based distribution system. Two of the largest video broadcasting platforms, Ustream.tv [180] and Veetle.com [181], have apps for iOS and Android devices. These enable users to broadcast live video content from their phones and tablet devices through online portals, where other people can view the streams. Both applications rely on the backend cloud-based systems provided by Ustream and Veetle respectively. These systems receive the video content from the mobile broadcaster and are then used to handle viewer requests and the distribution of the content.

BitTorrent Live [182] is a system which has addressed the issue of scalable video delivery without the need for a cloud-based backend. Using the Peer-to-Peer style of network that forms the backbone of Torrent file downloads, the video stream can be distributed between viewers. Each viewer can then pass on sections of the stream data to other viewers, so that all nodes on the Peer-to-Peer network can view the whole stream. The benefits of this architecture are that the cloud-based video server is not required for scalability issues and the broadcast becomes more stable when more people attempt to view it. In addition, this peer-to-peer architecture lends itself nicely to the multi-hop architecture of the PowerHop algorithm, proposed in this thesis.

3.3.2 Adaptive Streaming

The power consumption of the sender devices is directly proportional to the amount of data that it is sending. For video streaming applications, the bit-rate of the stream can be altered dynamically. This technique is included in some of the papers mentioned above to augment their power savings, [173] [174].

In [183] and [184] a video delivery model is proposed for transmitting video content to wireless mobile devices in an *ad hoc* network. The proposed model assesses the device characteristics and the battery life of the mobile device and then adapts up or down enhancement layers in an MPEG4 SVC video in order to send an appropriate stream. This changing of video quality changes the traffic on the network. In simulations, the lifetime of

the *ad hoc* network can be increased by up to 200%, when compared to streaming the maximum quality of video. While the proposed model is compatible with energy-aware routing protocols and multi-hop paths, these have not been incorporated into the model. Similarly, the residual power of a device is the only characteristic that is monitored periodically. This is a bit of a simplistic implementation.

Aupet *et al.* have presented a novel Web service for Automatic Video Data Flows Adaptation (Wava) which allows adaptive video streaming across heterogeneous networks [185]. The adaptation process is performed in two stages. Upon the initiation of a stream, information about the terminal device's CPU, GPU and screen resolution are sent to the server and assigned a "mark" between 1 and 5. Periodically, during the stream, the network bandwidth is also sent to the server and assigned a "mark" between 1 and 5. The lower of the two "marks" determines the video quality of the stream. This adaptation algorithm is quite basic and does not consider other important metrics relevant to video streaming, such as the residual power of a terminal device or available networks interfaces.

Di *et al.* proposed a technique for manipulating the size of the data chunks that are being downloaded during the playback of a video stream [186]. Their approach was to develop an algorithm that looks at the current bitrate of the video stream, as well as the stream duration in order to predict the duration of video that the user will choose to view and thus to decide the HTTP chunk size for delivery of the video. It was found that most users stopped watching videos before 220s of the video had elapsed. By predicting this value, the download rate and use of the cache can be controlled to match the length of video the user will see. The other effect is that with larger chunk sizes, the transport protocol overhead can be reduced and the time that the WNIC is asleep can be increased. In comparison with video streams from YouTube and Netflix, the proposed algorithm resulted in lower power consumption, but it does not consider the residual battery life of the device or the available network throughput in the in its selection of chunk size. This manipulation of the HTTP chunk size would be complementary to the work presented here and will be considered in future work.

Guruprasad *et al.* [187] proposed an approach which combines adaptive video streaming with residual battery life awareness. The authors dynamically controlled the data download rate (also seen in [186]) and the transceiver configuration, in terms of the number of antennas that are enabled on the MIMO WNIC, to reduce the power consumption of the device further. Simulations showed some increases in the maximum playback duration of a video stream over

basic adaptive video streams but this is to be expected when the download rate and transceiver configuration are also being controlled. One thing to note here is that the algorithm is not aware of the remaining stream duration, which is why the video quality would not be as high as the techniques presented in this thesis. Additionally, the authors have not implemented any savings on the hardware component that consumes the largest amount of power during video streaming, the screen of the device.

Ahmad *et al.* [188] proposed a scheme which considers the current network conditions and the residual battery level of the device for performing power-aware adaptive video streaming over LTE networks. The scheme requires an edge server, at the LTE Base Station, which is capable of transmitting the video sequence to the mobile devices at one of the available Scalable Video Coding (SVC) levels. This level is selected based on the battery level of the device but the remaining duration of the stream is not considered in this calculation. Power savings of up to 10% were measured in on-device tests.

3.3.3 Region of Interest

Existing adaptive solutions affect equally the whole viewing area of the multimedia frames in the content bitrate adjustment process. A noteworthy technique in optimizing energy consumption in a screen is to consider a region of interest (RoI).

An RoI, in video terminology, is classified as an area of a video frame that attracts the most amount of attention from viewers. As an example, while watching a football match, the viewer may be most interested in looking at the area around the football (though this may not be the case for everybody). There are certain regions in each frame of any video sequence on which the users focus more than the others. The authors in [189] found that when a high-resolution window is adapted at the point-of-gaze and the resolution in peripheral areas is degraded, the users had longer initial saccadic latencies in peripheral areas (the time taken to identify a visual target), than when a low resolution was uniformly displayed across the whole display window. Further, the authors in [190] found that if the degradation is increased in the peripheral areas, then in order to maintain the user's perceived quality of experience, the size of the adapted high-resolution window at the point of gaze also needs to be increased [191]. Using gray scale images, the five factors that were known to influence user-attention could be merged seamlessly. These include contrast with region background, region size, shape, location and determination of foreground and background areas. These factors were combined into an

overall “Importance map” (IM), which was used to classify the importance of image regions. Based on the IMs, the authors demonstrated a technique for controlling adaptive quantization processes in an MPEG encoder [192].

Agrafiotis *et al.* [193] presented a framework for model-based, context-dependent video coding, based on exploitation of characteristics of the human visual system. The system utilizes variable-quality coding, based on priority maps, which are created using mostly context-dependent rules. There has been considerable interest in RoI research, primarily based on the premise that where a user’s gaze rests corresponds to the location of the symbol currently being processed in working memory. Consequently, the idea has been to allocate screen area dynamically, with more resources being earmarked for the portion corresponding to the RoI. The authors in [194], propose a scalable RoI (SRoI) algorithm, which can support fine-grained scalability in region of interests with low computing complexity in order to achieve better objective and subjective video quality.

Figure 3-1 shows an illustration of typical smart-phone with pre-defined block of RoI. In this scenario, the display region of the phone is divided into 20 pre-defined blocks, though in reality, this could be any number. Further, the RoI could consist of one or more blocks in the screen. Each block selected as the RoI could be decoded with a very high quality while decoding the other regions in the frame using a lower quality. These blocks within the frame could be then superimposed to form a single RoI based adaptive encoded frame.

However, a major problem with using the RoI methods is that the bordering area between the RoI and the non-RoI needs to be handled carefully, without affecting the perceived quality. There are several methods for discovering the RoI in a video sequence, which include eye-tracking (with cameras), and video processing and analysis [192]. These could be broadly classified into two categories.

Passive setting of RoI: The passive method involves defining regions of interest and regions not of interest beforehand, and assigning Signal-to-Noise Ratio (SNR) elements accordingly to the regions. This method is used for Closed Circuit TeleVision (CCTV), for example, where the user wants higher resolution for a certain region [195]. This method handles the pre-set RoI with high resolution in the encoding stage.

Active setting of RoI: The active method does not set a pre-defined RoI, for it would change regularly based on the environment or contents. The RoI detection methods include using a Motion Vector (MV) to select the region with a large vector value as a RoI setting [195][196]. The vector value is derived from monitoring the number of differences of movement between consecutive video frames in the stage of movement prediction and is used to divide regions with Flexible Macroblock Ordering (FMO) [197]. Other methods include adjusting the quantization value in the encoding stage to improve resolution in just the regions of interest, and making it go through the high high (HH) filter in the wavelets conversion stage to improve resolution [198][199].

While a purely passive scheme would result in improper selection of RoI, an active and dynamic setting of RoI would not be very practical on a battery-powered device as it would consume too much energy for mobile devices. An alternative to this would be to find the RoI on the streaming server itself, which could then transmit RoI metadata alongside the video. Ji *et al*. have shown how RoI processing can be combined with quality scalability to maximize energy-savings and QoE [200]. Their scheme showed that a 15% energy reduction could be achieved with adaptive RoI processing with minimal degradation in video quality. In [201], RoI is also used to significantly reduce the video bit-rate while maintaining a high QoE. However, the algorithm proposed for implementing the decoding adaptations does not make use of temporal scalability or spatial scalability. It would be interesting to see if the adaptation

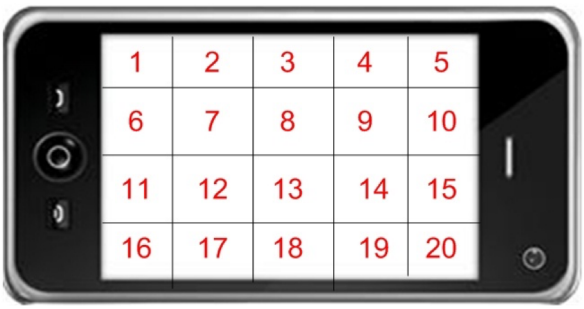

Figure 3-1 - Different pre-defined region-of-interest blocks in a smartphone

of the video stream on the server could be performed using SVC and RoI mechanisms to maximize the QoE-to-bit-rate ratio. This is a significant research challenge that would have to be addressed in the coming years.

3.4 Intelligent Algorithm Designs

3.4.1 Utility Functions

Utility functions are models for implementing an algorithm where each parameter is given its own weight based on its overall effect on the total system. By weighting the parameters in this way, the algorithm can be used to approximate the optimal outcomes in the system. For example, Trestian *et al.* investigated new techniques in [166] predict the energy consumption of the networks available to a mobile device. In this work, a utility function (3.1) was proposed for ordering the vertical/horizontal handover between different networks within range based on the predicted energy that will be consumed on each network. Other metrics are also accounted for in this utility function: the monetary cost of the network and the suitability of the network for quality and mobility requirements of the client device.

$$U^i = (u_e^i)^{w_e} * (u_q^i)^{w_q} * (u_c^i)^{w_c} * (u_m^i)^{w_m} \tag{3.1}$$

Where: i - the candidate network, U - overall utility for network i and u_e, u_q, u_c, and u_m are the utility functions defined for energy, quality, monetary cost and user mobility for network i. w_e, w_q, w_c and w_m are the weights assigned for each of the four considered criteria.

In [200], the authors propose the Energy Scalable Video-Decoding (ESVD) for the implementation of a RoI-aware video decoder. Each macro-block (MB) in the video is assessed under three criteria (L_1, L_2 and L_3 in (3.2)):

L_1: what type of MB it is (intra, inter or inter MB in a B-frame)

L_2: the partition information related to the MB

L_3: the level of interest that the user has in the MB (obtained from RoI calculations)

In order to compile these three criteria and enable the selection of an energy and QoS aware decoding profile for the MB, a novel function has been proposed (3.2).

$$f(MB) = a_1 \sum L_1(k) + a_2 \sum L_2(k) + a_3 \sum L_3(k) \tag{3.2}$$

The terms a_1, a_2 and a_3 are empirically set weights for each of the three consideration criteria and the $L_1(k)$, $L_2(k)$ and $L_3(k)$ terms are the values of each of the three criteria described above, assessed at multiple phases of processing for each assessment criterion. The k term refers to the effects on the total performance for each individual phase. The results from each phases of assessment are summed together. Following the selection of the energy profile for a MB using equation (3.2) each MB and its profile information are fed into the system utility function (3.3) which attempts to maximize the quality of the decoding of the MB but limits the energy consumption of the process based on whether the device is in one of four states: "maximum battery life mode", "battery optimized mode", "maximum performance mode" or "enhanced quality mode".

$$u_{i,j} = \frac{\Delta e_{i,j}}{\Delta PSNR_{i,j}} \tag{3.3}$$

In equation (3.3), i,j denotes the position of the MB within the frame. $\Delta e_{i,j}$ corresponds to the difference in energy between decoding the MB normally or as per the selected energy profile. Similarly $\Delta PSNR_{i,j}$ refers to the difference in PSNR of decoding the MB normally or as per the selected energy profile. While the initial test results yielded energy savings in the region of up to 40% between the highest and lowest ESVD decoding modes, the algorithm could have been even more aggressive for additional energy savings. For example, as well as the RoI-aware adaptive decoding, ESVD could have employed an adaptive frame-rate in the decoding process too.

In [202], a conceptual framework based on utility function (UF) is introduced, which models video entity, adaptation, resource, utility, and the relationship among them. Instead of modeling the UF through analytical models, the proposed approach performs UF prediction, based on the video content, and classifies the video clips into a number of clusters. Based on the predicted UF, the video trans-coding parameters are applied. However, neither the transmission aspects of the video delivery nor the device's energy are considered in the algorithm.

3.4.2 Context-aware computing

Context-aware energy-conscious computing is the process of recording inputs from various metrics and sensors available on a mobile device. These data in conjunction with past data are

then fed into a utility-function which adapts and learns from past experiences in order to maximize the user's QoE while maximizing the energy savings in a device. An important point about context-aware computing is that the aim is to find an optimal and complete solution by combining all available mechanisms. Vallina-Rodriguez *et al*. conducted a study where-by the day-to-day interaction of 20 subjects with their Android devices was recorded for analytical purposes [203]. The goal was to investigate whether the users' interactions with mobile devices could be modeled easily in order to algorithmically create a unified energy conservation mechanism. Statistics about the CPU, memory, battery, network interfaces, the screen and other sensors were logged on each user's device. The authors concluded that as a result of widely varying user-device interaction behavior, that static-algorithmic control of the device's resources is an insufficient approach.

There are three steps that must be taken in order to design a context-aware energy-conscious solution for mobile devices:

Firstly, the quality-of-experience-to-energy-consumption ratio of each mechanism and combination of mechanisms outlined above must be evaluated in a range of different real-life situations. This entails designing and conducting significant user testing in order to gather enough information to model how each mechanism and combination of mechanisms perform in relation to each other. This model will later be used for selecting the combinations of mechanisms to be used in order to achieve the energy savings required on the mobile device. The lower the ratio is, the higher the mechanism's efficiency in terms of reducing the energy consumption while maximizing the QoE. Testing similar to this as this can be seen in [152] where the authors describe the GUIs investigated in terms of user popularity and energy-reductions. The ratio calculation is an extension of this work.

Secondly, a utility-function for assessing the residual power of the mobile device, in the context of its current application, must be created. This involves considering application-specific QoS requirements, the current rate of energy consumption, past experience, user preferences and predictions formed from the collection and analysis of past data. The utility function will give the system utility, U, which can then be assessed (using thresholds) in order to select the most effective of the mechanisms/combinations above.

Finally, the algorithm must maintain a cache of device usage statistic and trends. This data is required for dynamic alteration of utility function weights and the threshold values in order to

ensure the optimality of the solution. This step is where the algorithm becomes context-aware by conducting real-time learning and behavior-monitoring. One example of a fully context-aware system is ErdOS [204], where the authors have built an energy-aware extension to the Android OS for mobile devices. ErdOS is designed to predict user-device interaction based on past profiling and location etc. This enables the system to predict when the user will next recharge their device and to predict the resource requirements of specific applications that are running on the device. The second main feature of ErdOS is to allow a device to take advantage of opportunistic access to computing resources in nearby devices. The example use of this might be to retrieve the location of a neighboring device over Bluetooth instead of enabling the GPS interface. Currently the functionality provided by ErdOS is simply to enable or disable hardware resources based on the predictions of whether they will be required in the near future or not. ErdOS cannot yet scale device functionality gracefully but adopts a hard-line "on or off" approach. While this does yield energy savings, the same effect could be achieved using the JuiceDefender [205] application, or something similar.

3.5 Summary

This chapter described state-of-the-art techniques for achieving energy savings on smart mobile devices and analyzed their inner workings in detail. As has been illustrated when describing each of the works, there has been no comprehensive or unified solution presented to date that successfully solves the problem highlighted at the start of this thesis. Each of the related research works falls short in some aspect of its implementation, which makes the development of EASE all the more important. In the next chapter, the architecture and design of EASE is presented to solve the problem statement of this thesis.

Chapter 4 - Background Testing and Solution Architecture

This chapter introduces EASE, the novel Energy-aware Adaptive Solutions for multimedia delivery to wireless devices, that can be used to solve the issues highlighted in Chapter 1. EASE is a context-aware umbrella-like collection of solutions which gathers information about a mobile device's residual power, usage, hardware components, Quality of Service (QoS) requirements and its neighboring network nodes. This information is then analyzed by the algorithms implemented within EASE and an efficient adaptation strategy is calculated in order to conserve energy on the mobile device, while maintaining the QoS requirements of running applications. In this chapter, the background tests that were performed to investigate the performance of different components on a mobile device are described. Following this, the architecture of EASE is described in detail, in relation to its constituent components, the devices it operates on, and its usage scenarios.

4.1 Introduction

Most modern smart mobile devices are equipped with an IEEE 802.11 wireless network interface card (WNIC) as well as some or all of the following interfaces too: cellular, Bluetooth, NFC, IrDA, ANT+ and GPS. Additionally, modern smart-devices can also have many sensors built-in, such as an accelerometer, gyrometer, barometer, digital compass, ambient light and proximity sensor, etc. All of these hardware components have different energy-consumption characteristics and allow different opportunities for energy savings on the mobile device. For example, many smart-phones and tablets come with ambient light sensors, which enable dynamic adjustment of screen brightness based on the measured brightness levels. Another example of different hardware allowing different energy savings is when a mobile device can switch from using a cellular interface to using a WLAN, for data transmission. In fact, the power consumption on a mobile device can be said to depend on a combination of three factors: the applications running on the device, the hardware components of the device and the user's interactions with the device.

To illustrate this statement better, Figure 4.1 lists some of the components and behavioral traits that cause significant energy usage for different types of mobile applications. This is not an exhaustive list of all the components required for the specific apps to work but shows the key energy draining components. For video streaming apps, for instance, the screen consumes a

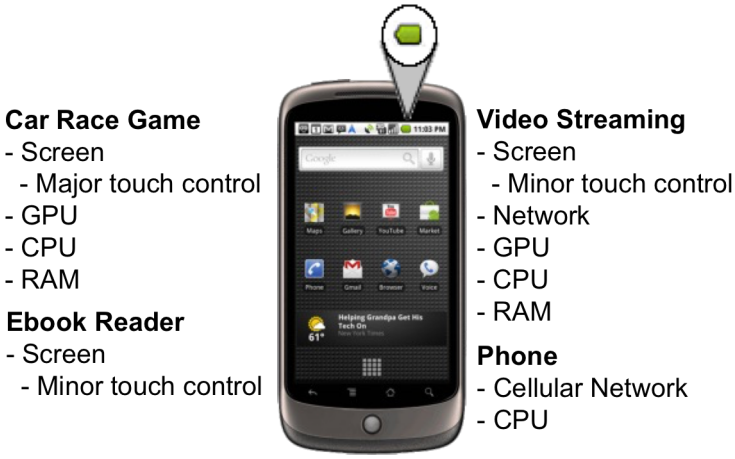

Figure 4.1 - Mobile app energy drains

lot of power because it is constantly updating its content. The user does not interact with the touch controls frequently though because they are watching the video. The video itself is being streamed, which means that network, CPU and RAM resources will be in high demand. The compressed video stream will also need to be decoded before play-back, so this means more work for the GPU, CPU and RAM. Making a phone call is very different in terms of the components used, with the Cellular interface and the CPU being the two main power draws. An eBook reader app will require the screen to be on again and will need to use the touch interface to control turning the pages in the book. A car racing app, on the other hand, is somewhat similar to the video streaming app, in terms of the components involved, but there is less network traffic involved. The behavioral pattern of the apps' users will be quite different though, with the racing game needing a lot of touch control for steering etc. There is a more detailed break-down of different energy consuming components in Section 4.3.

EASE, is a novel way to take advantage of the hardware and software settings available on a device, in order to save energy, while maintaining the QoS requirements of running applications. EASE can be used to dynamically monitor different device, application and network characteristics. For example, the following lists some of the metrics that could be monitored:

- Residual power of the device
- Brightness of the display
- Ambient light level
- Expected remaining duration of app execution (if available)

- Packet loss on the network
- Current QoS requirements of the apps running on the device
- CPU usage and clock frequency
- Neighboring nodes available for multi-hop *ad hoc* networking
- Energy cost of transmission route
- Latency
- Bandwidth

Having collected these data, energy saving strategies can be implemented in EASE by exploiting different energy conservation techniques. Some of these techniques were discussed in the related work chapter and will now be discussed in terms of how they would be beneficial for inclusion and further development within EASE.

4.2 Power Saving Opportunities for EASE

Where specialized hardware is available in a device, this could be exploited by EASE because it provides a unique opportunity to maintain complete device functionality while reducing the power consumption of the device. However, the offloading of processing from the CPU to the GPU is implemented by default in most modern mobile-device video decoders. Similarly, taking advantage of multiple network interfaces to maintain constant network presence will not be possible with devices that have already gone to market without the additional hardware specifications required. Control of a device's MIMO WNIC, where some of the WNIC's antennas are disabled dynamically would be beneficial for use in EASE. One challenge with this functionality is that there are no APIs exposing this level of control in the Android and iOS SDKs, so a custom middleware implementation would be required.

The Power Save Mode (PSM) of IEEE802.11 networks is implemented on all modern smart devices. For example in the Android operating system, the WNIC of the device has a beacon period of 300ms for keeping in touch with the network Access Point (AP) [164]. PSM can be utilized and extended in EASE to cater for *ad hoc* networks as well as infrastructure mode networks in order to increase the energy efficiency of *ad hoc* operation.

The time that the WNIC is asleep can be increased using intelligent download-rate and chunk-size control. This, in turn, reduces the overhead of receiving the video content on the client device. Another setting that can be manipulated is the width of the transmission channel,

which can also be used to speed up the download rate of the video content as required with a minor impact on the power consumption of the WNIC. EASE would benefit from being able to use these techniques.

Energy efficient horizontal and vertical network handover could also yield energy savings in the context of the unified EASE approach. An automatic scheme for determining the most energy efficient network available to a mobile device would be extremely beneficial in terms of overall device energy conservation. This is because the WNICs of a mobile device are some of the most power-hungry components on the device.

Adaptive video streaming and decoding are both essential parts of any adaptive scheme for energy savings on the modern mobile devices. This is because of the dramatic increase in popularity of mobile video streaming applications in recent years [206]. Adaptive encoded videos on the server side of the video stream can come in the form of adaptive frame-rate, frame-resolution, quantization parameters or color sampling-rates. Adaptive frame-rate and quantization parameters can also be employed on the client device to further increase the energy savings. Energy savings from this approach can be seen on the device's WNIC, CPU, GPU, and RAM. Similarly, Region of Interest (RoI) can be used to augment the adaptive encoding and decoding schemes in order to intelligently increase the QoS of a video stream.

EASE could also benefit from dynamically managing the brightness and supply voltage of a device's screen as the current screen brightness manager of mobile devices only considers ambient light for the management of the screen. This is simply not sufficient for an effective dynamic solution. The scheme should automatically become more aggressive the lower the battery level is and it should also consider the content that needs to be displayed on the screen at any given time.

The energy consumption of a mobile device very much depends on the hardware components and specifically how they are utilized in software. A key example of this is the CPU. The clock frequency of the CPU and the voltage supplied to it can be configured dynamically as required by applications on the device. This scaling of the CPU could be included in EASE as the CPU is also one of the largest consumers of energy in a mobile device.

For data transmissions from a mobile device, the distance of a device from the AP that it is associated with greatly affects the required transmission power for successful communication.

One approach for reducing the power required for transmission is to find an intermediary node to act as a relay for the data. This approach requires an *ad hoc* connection between neighboring devices in the WLAN but provides a big opportunity for energy savings. This would also be ideal for inclusion and refinement in EASE.

In order to know the most effective techniques to employ, and which apps/usage scenarios to target, the power consumption characteristics of a mobile device were analyzed for different usage scenarios. This work is presented in the next Section and provides the context for the selection of the testing scenarios used in this research.

4.3 System Based Modeling and Testing

Battery depletion in a wireless device depends on both the hardware and software of a device. The exact amount of energy consumed by each of these components in the device is dictated by the device characteristics and the nature of the applications running on the device and the interaction of the user with the device. In order to get a comprehensive analysis of the energy consumption behavior in a high-end wireless device, one specific device, the HTC Nexus One, was considered and various tests were performed. This device was selected because of its wide range of functionality and because it runs Android, an open source operating system well-studied and highly popular [207]. This is critical to our studies as it allows for a deeper understanding of the power distribution between the device components. For this reason, few other mobile operating systems would have been suitable at the modeling stage. It is worth mentioning here that power management studies in smart-phones have been done in [208],[141], but for different sets of devices: the Openmoko Neo Freerunner and the 1st Android HTC phone, the HTC Dream. However, the HTC Nexus One is a newer device with more modern hardware components: an OLED-based screen, a 1 GHz ARMv7 CPU, a dedicated GPU and many sensors. As a result, even though this device is no longer state-of-the-art, the trends that can be derived from utilizing it for testing purposes are relevant to the state-of-the-art devices now on the market because of common hardware technologies. The test system composed of the Nexus One phone (running Android 2.1, and subsequently Android 4.2.2) and a multimedia (video and audio) streaming server in a wireless IEEE 802.11g network (Wireless Local Area Network (WLAN) or Wi-Fi).

The phone is connected to an external measurement setup that monitors and logs its power consumption during the execution of various tests. These tests, specifically designed for the analysis of the consumption, include receiving and playing video streams over the wireless network as well as applications to monitor the CPU usage and to automatically change device settings to put the phone into different states (e.g. changes in the screen brightness). The aim of these tests was not just to measure the overall system power, but also to measure the exact breakdown of power consumption by the device's main hardware components.

Figure 4.2 shows a smart-phone and the potential major energy consuming components in a hand-held device. The power consumption is measured as shown in Figure 4.3. A shunt resistor (1.24 Ω; ± 1%) is inserted between the phone and the battery, in order to calculate the power consumption by measuring the voltage drop across the resistor. All these measurements were sampled by an Arduino micro-controller, which logged the instantaneous power-consumption of the device onto a computer. In order to break down the power consumption, experiments for each device component were performed by changing the parameters of one component, while leaving those of the other components constant [209]. The additional information was provided by Android's battery usage statistics, which gave a rough indication of the percent-age of battery usage attributed to each of the major consumers [210]. A detailed explanation on the measurement set-up and how the tests were performed is provided in [141],[209],[211], though the paper [209] in itself focuses on proposing new energy-saving mechanisms in the smart-phone. Further, in order to ensure that the resistor did not have an effect on the overall power measurements, two different high-precision resistors (0.22 Ω ± 1% and 1.24 Ω ± 1%) were considered in our work and the tests were repeated. However, apart from the slightly higher voltage drop across the 1.24 Ω resistor, there was no noticeable change

Figure 4.2 - Smart-phone and its different components

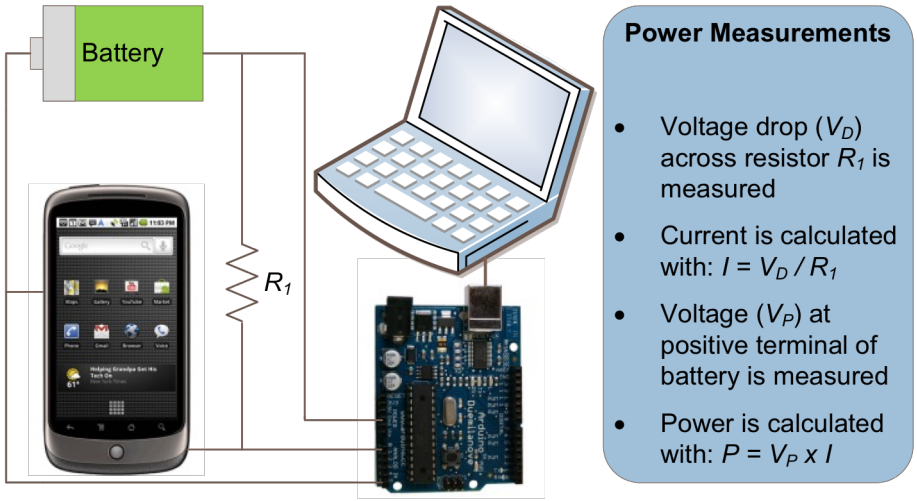

Figure 4.3 - Schematic of a Measurement Set-up

in the power consumed by the major components on the phone. Notably, it was observed that there were four major energy-consuming components - screen, CPU, speaker and network interface. The energy consumed in these four major components was measured as follows:

1. The screen was tested by measuring different pairs of brightness levels and pixel colors.
2. The CPU was tested by running an application that gradually changed the CPU activity usage in steps from as close to 0% as possible, up to 100%. The application did this by starting arbitrary background processes at dynamic intervals. Setting the interval between the background processes allowed the CPU usage to be increased or decreased as desired. The power consumption of the device was measured for the different CPU usage levels and recorded for that specific CPU level. The minimum CPU usage levels below are from tests where the CPU was at 3% usage.
3. The audio play-out was tested by playing the same song at different volume levels, using the speakers and the earphones.
4. The effects of the network interface and the audio-visual quality in multimedia streaming were tested in a single experiment, in which the same audio-visual content was played using different quality settings over a wireless network stream.

Figure 4.4 shows an important result in terms of energy consumption in the Android Nexus One device. It shows the minimum and the maximum energy consumption values of the four major energy-consuming components. The energy consumed by the screen was calculated by measuring the power consumption during a particular test with the screen enabled (at 100% brightness and displaying white, then once again with 1% brightness and displaying black). The same test was then re-run with the screen completely disabled and the result was subtracted from the power consumption measured in the two previous tests. Further, the power drawn by CPU was obtained by monitoring the CPU usage and subtracting the screen's power consumption. Once these values were obtained, the other components' energy consumption was acquired by calculating the power difference of the additional contribution that was caused while another component was in use. It can be observed from Figure 4.4 that in case of multimedia transmission, the display screen and the CPU consume the maximum amount of power followed by the wireless network interface. Further, the maximum power consumption in the display screen was eight times more than the minimum power consumption. This factor increased to sixteen times in case of CPU and was five times in case of wireless network interface. Hence, in order to support optimum quality and long lasting

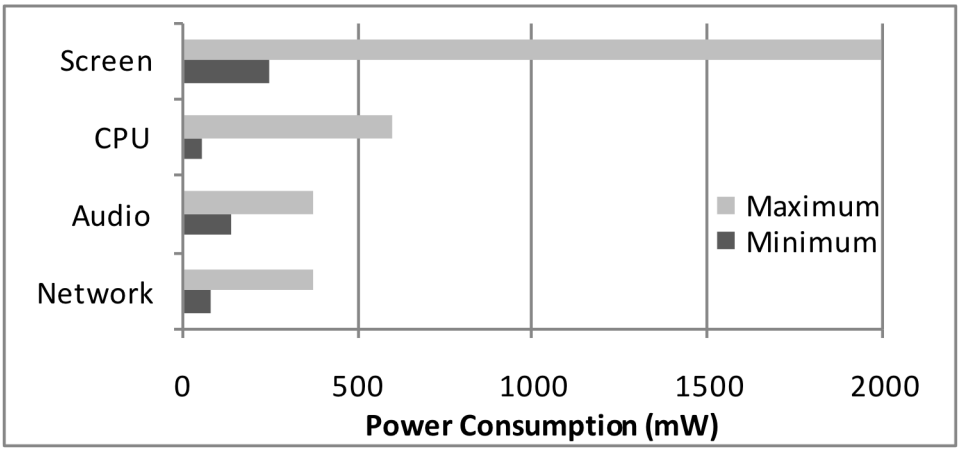

Figure 4.4 - Energy distribution of different components in Nexus One mobile device

Table 4.1 - Minimum and maximum power settings in Nexus One mobile device

	Minimum	***Maximum***
CPU	CPU usage as close to 0% as possible (recorded at 3%)	CPU usage at 100%
Screen	Brightness at 1%, black pixel color	Brightness at 100%, white pixel color
Audio	Audio playback muted	Audio playback at highest volume using speakers
Network	Connected to a WiFi network, idle	Connected and receiving a high quality video stream

video transmission to the end-user while on the move, the energy in the display, processor and in the wireless interface would need to be considerably reduced. Hence, it is imperative to thoroughly investigate and propose an energy-optimal adaptive scheme for multimedia-centric wireless devices. The settings used for measuring the minimum and maximum consumption values for the components are summarized in Table 4.1. The dependence of each component's consumption on device settings is discussed as follows.

4.3.1 Energy Consumption in Screen

The screen's power consumption ranges from about 0.25W to 2W. For the tests, the red, green and blue pixel components were kept at identical levels; the energy consumption of different colors on the display was not measured but is discussed in Section 3.1.2. The energy consumption was found to depend on both the brightness level of the screen as well as the brightness of the pixels' colors. In Figure 4.5, the screen brightness levels here were quantified in percent, from 0%-100%. These are the steps possible in the Android and iOS APIs. The pixel brightness is a numerical from 0 to 255 signifying the luminance of the pixels, i.e. the averaged value of the RGB color channels. From the measurements in Figure 4.5, it can be observed that the consumption increases approximately in linear fashion with the screen's brightness and exponentially with the color brightness. This is because of the energy characteristics of the particular OLED display. Hence, the higher the brightness level, the more important it is to know the average pixel brightness in order to accurately estimate the power consumption. Without taking the color into account, the error of an estimate can be as high as 300% as the power difference between a black and a white screen reaches 1.77W at the highest brightness level.

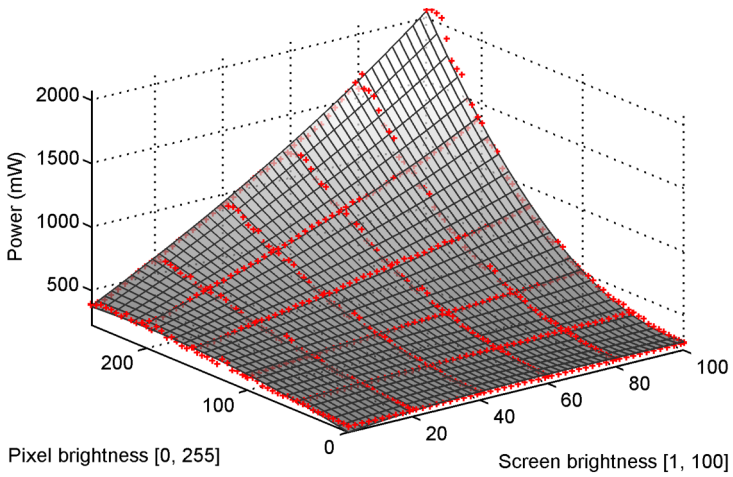

Figure 4.5 - Power measurements in the device screen

4.3.2 Energy Consumption in CPU

The CPU's power consumption depends on its usage and ranges from 49mW to 625mW. The results from four executions of the same test has been combined and can be seen in Figure 4.6. The two graphs are marked by a straight line and by a series of '+' symbols respectively. It can be observed that the power increased with the usage in a linear fashion but increased sharply at about 80% usage. This behavior was also noted in [208] and [141] on tests on a range of different Android devices with varying specifications and processors. This could potentially be caused by some dynamic voltage and frequency scaling on the devices as the CPU comes under more pressure, however, there has not been any comprehensive answer for this behavior. It still requires a significant amount of investigation to derive a definite answer for such a behavior. Notably, this manifests into the following: reducing the CPU usage from 70% and lower, yielded a relatively little power savings compared to a reduction that takes the usage from any value higher than 80% to any value lower than 70%. However, this assumes that the usage information provided by the Android API for the CPU level is always accurate. Nevertheless, the measurements gave an indication about the energy-saving potential of the CPU and put it into relation with the other components.

4.3.3 Energy Consumption in Audio

During the modelling stage, it was observed that while playing back audio on the speakers on a device, the power consumption increased from 135mW at the lowest to 375mW at the highest volume setting on the device. The device in question had 16 levels available for the

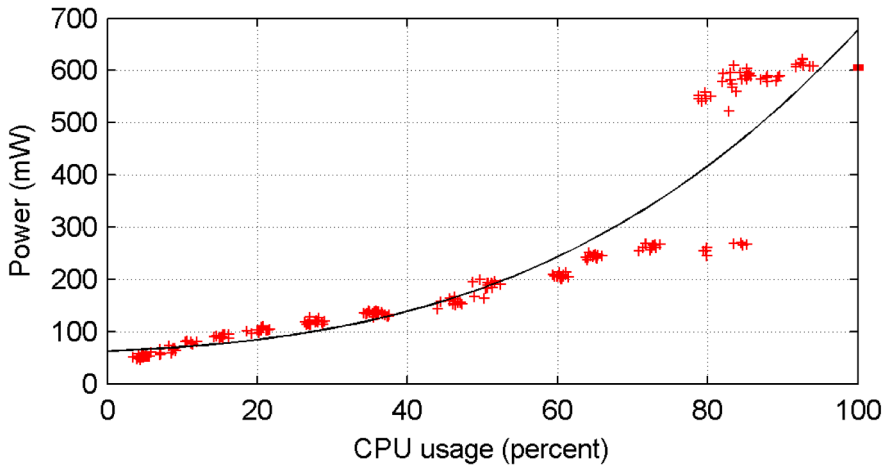

Figure 4.6 - Power consumption in CPU

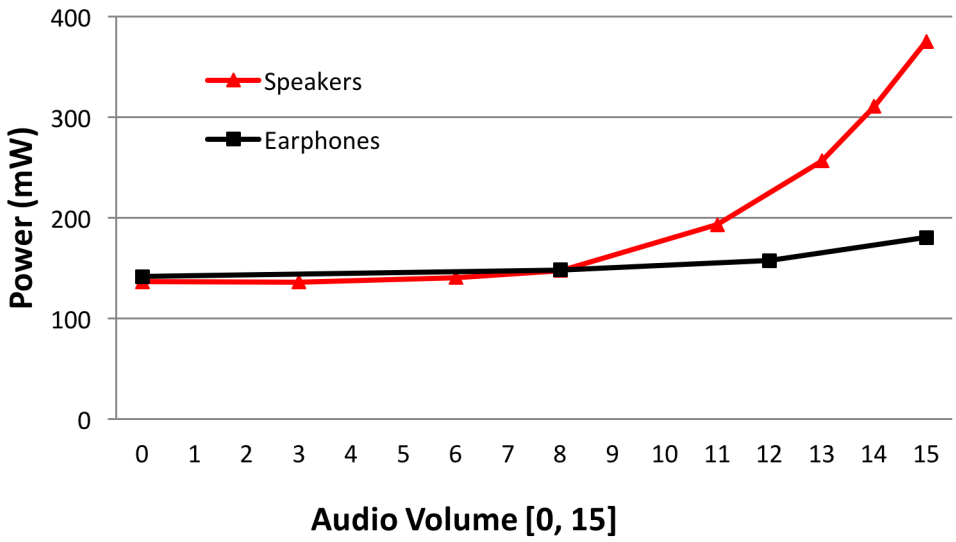

Figure 4.7 - Power consumption in audio playback

audio volume, so Figure 4.7 shows the power consumption for each of these levels. The power consumption increased as the volume setting on the device (i.e. the sound level) was raised, but the difference in energy consumption when playing audio through earphones was marginal while the consumption increased exponentially when using speakers; as can be observed in Figure 4.7. This shows that the audio interface offers almost no savings when earphones are used and reducing the audio volume in this case to save energy does not yield significant results. The main reason for the higher increase in power consumption in the case of the speaker is that the power required to drive the speaker is significantly higher than that required for the earphones. As a result, the jumps from one volume setting to another on the device involve larger amounts of power to drive the change, in the case of the speakers. This means that the amplifier components in the audio subsystem consume increased amounts of power as the volume is increased.

4.3.4 Network and the Effect of Video Quality during Streaming

In order to test the energy consumption in the network interface, a particular video clip was encoded using low and high quality settings as in Table 4.2. A continuous multimedia stream resulted in significant energy consumption across the network interface. As shown in Figure 4.4, the network interface consumes around 400mW of power during continuous streaming. Further, the average power consumed in the network interface was found to be around 350mW

while using 3G, around 300mW during Wi-Fi and less than 100mW while using Bluetooth. A separate study on the energy consumption analysis in the network interface in different traffic scenarios can be found in [210],[212]. This shows the amount of energy that could be potentially saved in the network interface.

4.3.5 Total Device Energy Consumption

In order to deduce information about both the multimedia quality and the network effects on the energy consumption, two different scenarios were considered.

Firstly, a particular video clip (based on the encoding scheme mentioned in Table 4.2) was played both locally and streamed using Wi-Fi. Figure 4.8 puts the measurements of the power consumption on a device for each video clip into relation. The power difference between the low quality (LQ) and high quality (HQ) video was 255mW for the local playback and 325mW for the stream. The difference was greater in case of streaming, as the HQ stream not only increases the computational power but also the data rate of the received stream. The power difference between the local playback and the stream is 305mW for LQ video and 372mW for HQ video. Further, it was observed that the network interface accounted for about 370mW

Table 4.2 - Video encodings used

	Low Quality(LQ)	***High Quality(HQ)***
Codec	MPEG-4	MPEG-4
Video bit-rate	128 kbps	1536 kbps
FPS	10 frames/second	23 frames/second
Dimensions	200 x 120 px	800 x 480 px
Audio bit-rate	32 kbps	128 kbps

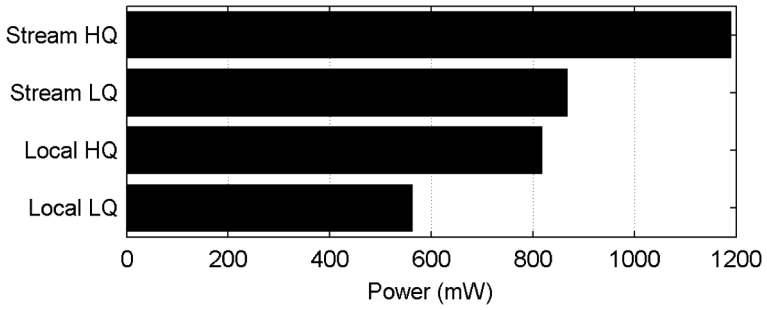

Figure 4.8 – Energy Consumption of Phone During Video Playback

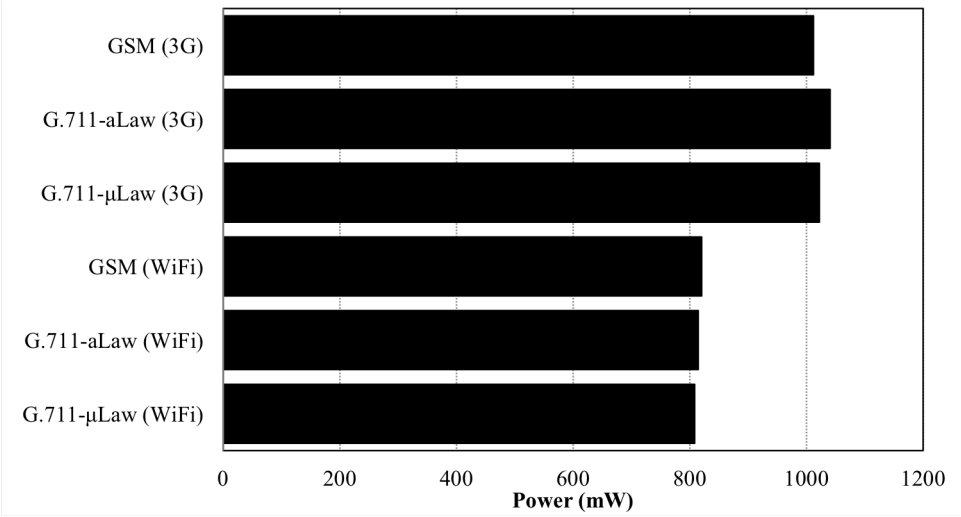

Figure 4.9 - Energy consumption of Phone during VoIP

while receiving HQ stream and offered savings up to 70mW when the quality was reduced. Notably, a quality reduction lowered the computational power, which resulted in savings up to 255mW.

Secondly, the power consumption of the device was measured for a VoIP call using the 3CXPhone application [213]. This call was repeated for three different audio codecs (GSM, G.711 a-Law and G.711 µ-Law) over Wi-Fi and 3G networks. As can be seen in Figure 4.9, the 3G cellular interface consumed, on average, 210mW more than the IEEE 802.11 wireless interface, for VoIP calls. Additionally, for VoIP calls over 3G, the GSM codec is the most power-efficient of the three codecs while for Wi-Fi, the G.711 µ-Law codec performs most efficiently. This behavior difference is due to the contrasting energy-per-bit characteristics of Wi-Fi and cellular networks. Notably, it illustrates that considerable power could be saved in VoIP applications based on the choice of network and audio codec.

4.4 Target Scenario Selection

It is clear that there are numerous options available for investigation in the context of a unified solution for energy savings on mobile devices. While the end-goal of EASE is to dynamically optimize the usage of each of these techniques for the QoS requirements of any mobile app, this thesis deals with a subset of this functionality.

The research presented here targets *energy-savings in IEEE 802.11 networks in the context of video streaming applications*. This subset of functionality was selected for four reasons:

- As illustrated in this chapter, video streaming apps are among the most battery-intensive and popular of mobile apps. Achieving energy-savings for video streaming apps has a big impact on total mobile device battery life. This will be explored in further detail in the next chapter too.
- The QoS requirements of these apps fluctuate in real-time. As a result, a successful solution for video streaming apps needs to be very dynamic in its adaptation strategies.
- Video streaming requires the transmission of a significant amount of data. Therefore, the solution for this type of app must achieve optimizations on these data transmissions.
- IEEE 802.11 networks are the most ubiquitous of all wireless networks across smart mobile devices, laptops, PCs, etc. Designing a solution that targets Wi-Fi networks specifically means that the scheme can be deployed on all of these devices.

For these four reasons, the need for an energy-conservation solution in video streaming apps is great. Other research groups are working in the same space as a result, which further validates the need for this research. In this thesis, the area of video streaming is tackled for both the down-link and the up-link, i.e. for watching a video stream and for broadcasting a video stream. The components of such a solution can subsequently be utilized for other app types. For example, there is significant overlap, in terms of behavior, between video streaming applications and video-conferencing applications.

4.5 EASE Architecture

In this thesis, EASE acts as an umbrella-like solution for reducing the power consumption of battery-constrained smart mobile devices in IEEE 802.11 networks. These devices comprise of smart-phones, tablets and laptops. As a result, EASE needs to cater for varied device hardware and usage requirements.

The full network topology of a scenario where EASE is in use is depicted in Figure 4.10. In this topology, there are three distinct types of device. The first are the *Mobile EASE Devices* that run EASE and its algorithms; these are the user's terminal devices. The second type of device is the *IEEE 802.11 Wireless Access Point*. This access point manages an infrastructure-

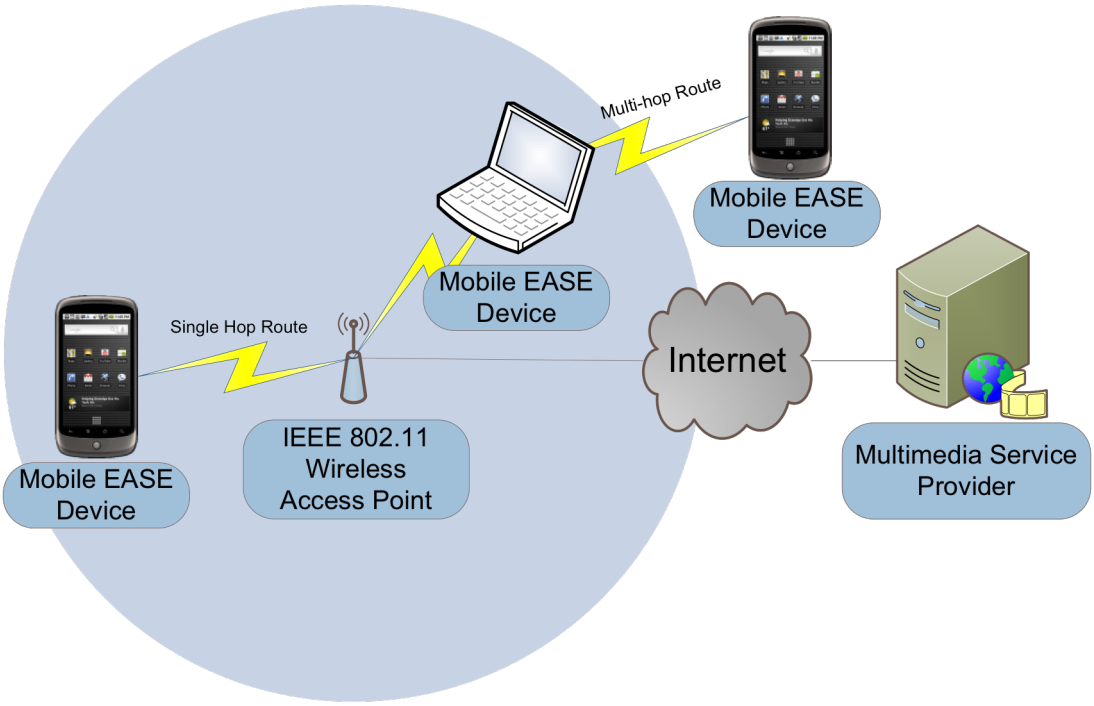

Figure 4.10 - EASE network topology

mode network and acts as the gateway for the network to the *Internet*. The final type of device is the *Multimedia Service Provider*; this is the server/cloud service that either serves or syndicates the adaptive video stream.

The *IEEE 802.11 Wireless Access Point* device does not perform any EASE specific operations. It can just be a generic AP device. As a result, this device is not described further. Both the *Mobile EASE Devices* and the *Multimedia Service Provider* will now be explained in more detail.

4.5.1 Mobile EASE Device

The block level architecture of EASE can be seen in Figure 4.11. All of the functionality of EASE sits on the *Mobile EASE Device* which is connected to an IEEE 802.11 wireless network. The EASE architecture is **Operating System (OS) and device independent**, so could be implemented on any mobile wireless device. EASE provides energy savings for both the down-link and up-link in video streaming applications, meaning that it can handle the play-back of video streams and the broadcast of video data too.

EASE operates as a middleware, to facilitate energy-aware optimizations between the hardware drivers in the OS and the applications running on the device. Figure 4.11 shows how EASE fits into the Hardware-Software stack on the *Mobile EASE Device*, with all of EASE's contributions and areas of power savings being highlighted in green stripes.

The first action that EASE undertakes is monitoring the performance of the device, the applications and the network. The *Performance Monitor* runs in the background on the device and performs this task. The collected data are then passed into the *Algorithm Modules*, where the data is processed by different modular algorithms in the over-arching EASE scheme. These algorithms can each address specific application, hardware or interaction use-cases and are discussed in more detail in Section 4.6. The algorithms calculate whether an adaptation should be performed or not and then pass their results on to the *Adaptation Module*. The *Adaptation Module* handles making changes to settings on the device, or to the content being displayed on the device, in order to save power without sacrificing user experience. The hardware components that are affected by the operation of EASE are highlighted in the figure to illustrate that it targets multiple different components. Specifically, for the selected target case in this thesis, power is saved on the *Display*, *Wi-Fi*, *CPU*, *GPU* and *RAM* components, and this is reflected in the device's *Battery* life.

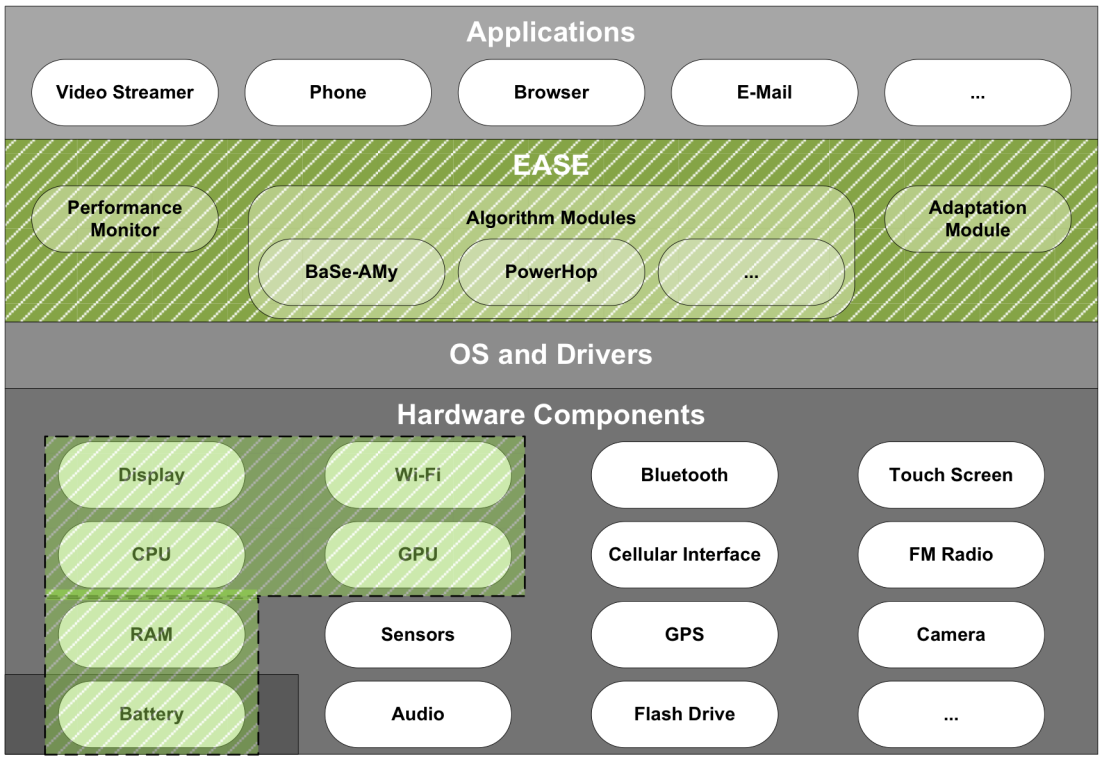

Figure 4.11 - Block Level Architecture of EASE on Mobile EASE Device

4.5.2 Multimedia Service Provider

The *Multimedia Service Provider* device is a streaming server which is connected to the *Mobile EASE Device's* LAN over the *Internet*. There are no components of EASE implemented on this device, instead all data collection and decision making occurs on the *Mobile EASE Devices*. The only requirements for this service provider device are that it is capable of serving adaptive video streams and taking orders from the client devices in order to adapt stream quality levels. This is a common ability in most modern adaptive video streaming protocols and standards. For instance, MPEG DASH [117, pp. 23009–1], Apple HLS [111] and Adobe HTTP Dynamic Streaming [114] all allow the video streaming client to order the server to switch the stream to a higher or lower quality level. For these protocols, the server application is just a simple HTTP server, where the video data of each stream is split into segments of a specific duration and then encoded at different quality levels. This means that the streaming client device just performs a HTTP REQUEST to download the next segment required for playback, at one of the available quality levels. The client video player appends this segment to the currently playing segment in order to have continuous play-back.

4.6 EASE Algorithms

In this thesis two novel algorithms are presented which fit under the EASE umbrella. The first is the Battery and Stream-Aware Adaptive Multimedia Delivery (BaSe-AMy) algorithm. BaSe-AMy is used to target energy-savings during the down-link of video stream reception and during on-device optimizations. The second is the PowerHop algorithm. PowerHop is used to target energy savings that can be achieved by performing multi-hop data transmissions on the up-link of sending data content from a mobile device. EASE constructively combines both of these contributions together in order to allow intelligent energy savings for both the down-link and up-link in video transmissions.

The BaSe-AMy algorithm addresses the first novel contribution of this thesis. BaSe-AMy returns energy savings in all of the hardware components highlighted in [11]. This is achieved by changing some settings on the device (e.g. screen brightness) and in the application (e.g. video quality). The algorithm has been successfully implemented and tested in previous publications ([214] & [209]) and is explained in further detail in Chapter 5.

When the client device is uploading content to a server (such as uploading a new video from a phone to YouTube), the PowerHop algorithm is used to determine whether to switch network

connectivity to an *ad hoc* multi-hop route, when it is deemed a more energy efficient path. This multi-hop algorithm addresses the second novel contribution of this thesis. The energy savings that result from this contribution are achieved on the WNIC of the client device. These savings come as a result of a need for a lower transmission power in order for successful delivery of data packets to a close intermediary node instead of an AP which is significantly further away. The PowerHop algorithm has been successfully implemented and tested in [215] and is discussed in detail in Chapter 6.

EASE dynamically combines both BaSe-AMy and PowerHop constructively in order to maximize the total energy savings on the device for both directions of video transmission.

4.7 Summary

In this chapter, the overall system architecture and usage scenarios of the Energy-aware Adaptive Solutions (EASE) are presented. Potential techniques for achieving energy savings on mobile devices were compared and a subset of these was selected for development and implementation in this thesis, within the umbrella EASE architecture. As has been explained, EASE is implemented as a middleware on mobile wireless devices. The constituent algorithms of EASE are also introduced and their locations in the hardware-software stack are illustrated. In the next chapter the BaSe-AMy algorithm is discussed in greater detail, from design right through to implementation.

Chapter 5 - BaSe-AMy

The overall goal of EASE is to prolong battery life by scaling and adapting the functionality of device and network settings dynamically without degrading the QoS beyond acceptable levels. In this chapter, the design of the novel BaSe-AMy algorithm for energy efficient video streaming is described. BaSe-AMy is implemented on a wireless mobile client device and specifically targets the key hardware components that consume most of the power on a mobile device for video streaming applications. The test tools and scenarios for evaluating the BaSe-AMy algorithm are then described and the results presented and analyzed in detail.

5.1 BaSe-AMy Algorithm

The novel Battery and Stream-aware Adaptive Multimedia Delivery (BaSe-AMy) algorithm has been proposed [214], [209]. BaSe-AMy is an algorithm for energy-aware device operation in the context of a video streaming client application. Two versions of the BaSe-AMy algorithm have been proposed to date, the first in [214] and the second in [209]. The first version of the algorithm receives battery level predictions, the remaining video stream duration and packet loss updates from the EASE performance monitoring system. The BaSe-AMy algorithm then analyzes this information to decide when or if the quality level of the video stream should be changed so that the multimedia stream can be played in full, where possible. The second version of the algorithm is an enhanced version of the original algorithm. In it, dynamic control of screen brightness is also performed in order to increase the energy reductions further.

The BaSe-AMy algorithm will now be discussed in further detail. The explanation of the algorithm will pertain to the second version of BaSe-AMy, which is the algorithm that will be utilized in the future work of this project.

Figure 5.1 shows the architecture of an EASE video streaming application which incorporates the BaSe-AMy algorithm. The application implementation consists of a number of modular blocks on both the client and server devices. The two forms of adaptation that are performed in BaSe-AMy are to the quality level of the video and to the brightness of the screen on the client. The adaptation to the video quality alters the bit-rate of the video stream that is being

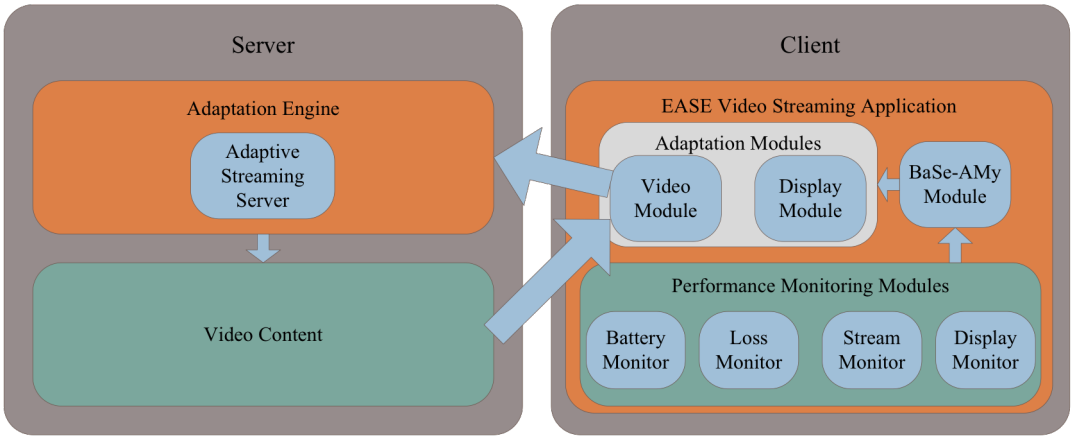

Figure 5.1 - BaSe-AMy Architecture

consumed on the client device. The result of this is that with lower bit-rates, the Wireless Network Interface Card (WNIC) of the mobile devices receives less data over the network, and reduces the amount of data to be handled in the RAM, CPU and GPU. This is where the energy savings were achieved. The brightness of the screen on the client device relates to a significant percentage of the total energy consumption of the client device (**Figure 4.5**). Real-time adaptations of the screen brightness are used to reduce the power-draw of the screen.

BaSe-AMy works in four separate phases, as part of EASE:

5.1.1 Streaming

The client opens the video stream connection to the *Adaptive Streaming Server*. The video stream is then displayed on the screen of the client device in full-screen mode with the *Video Module*. In the initial stages of the streaming connection, the highest video quality stream is played.

5.1.2 Monitoring

As the stream is playing, the remaining battery capacity, packet loss and stream duration are sampled periodically by the EASE *Performance Monitoring Modules*. The exact sampling period can be stipulated by the user or can be arbitrarily set in EASE before use on the device. Previous and current readings of the battery level are used to dynamically predict the remaining battery life in the *Battery Monitor* module. The loss rate is computed over each monitoring interval in the *Loss Monitor* module. The *Stream Monitor* module focuses on stream monitoring and determines the remaining stream duration. Finally, the *Display Monitor*

module keeps track of the screen brightness level. Unlike the other metrics in the monitoring phase, the brightness does not need to be sampled periodically as it is stored from the last time it was adapted.

5.1.3 Decision

The BaSe-AMy algorithm (Figure 5.2) is implemented in the *BaSe-AMy Module* on the client device. It analyzes the collected data from the monitoring modules and decides whether or not any adaptations are required. BaSe-AMy decides whether to adapt the video stream up or down one video quality level or to adapt the device's display brightness in order to achieve optimal results. BaSe-AMy decides on the adaptation to a lower quality of video and to a dimmer screen brightness setting if either the remaining stream duration exceeds the remaining battery-life or the packet-loss rate exceeds 25%. The loss rate threshold is based the threshold for the loss rate in Adobe's Dynamic Streaming Class [216], that triggers automatic adaptation if the packet-loss goes over 25%. The adaptations to lower values of video quality and brightness will only occur if the video quality and screen brightness are above their lowest possible values. In the assessment for adapting up again to higher video quality levels or a brighter screen, a smoothing mechanism is implemented which limits the ping-pong effect of adaptations up or down the different levels. This is achieved by multiplying the remaining stream duration by a scaling factor, to inflate it slightly, and then comparing the resultant value to the remaining battery life. The value of the scaling factor for the smoothing mechanism has been set empirically based on the testing results from the experiments that will be discussed in Section 5.2. Adaptations to higher quality levels and brightness levels will only be performed if the maximum levels are not in use already.

5.1.4 Implementation of Adaptation

There are two EASE *Adaptation Modules* that receive orders from the *BaSe-AMy Module*. These are the *Video Module* and the *Display Module*. When adaptations to the stream bit-rate are required, the *BaSe-AMy Module* lets the *Video Module* know that this is required. From there, an order is sent from the client to the multimedia server to request the new video quality level as directed by the BaSe-AMy algorithm. Adaptations to the level of brightness of the display are implemented in the *Display Module* on the client device based on the instructions from the *BaSe-AMy Module*. The application then continues monitoring its resources and the BaSe-AMy algorithm will decide if any additional adaptations are required.

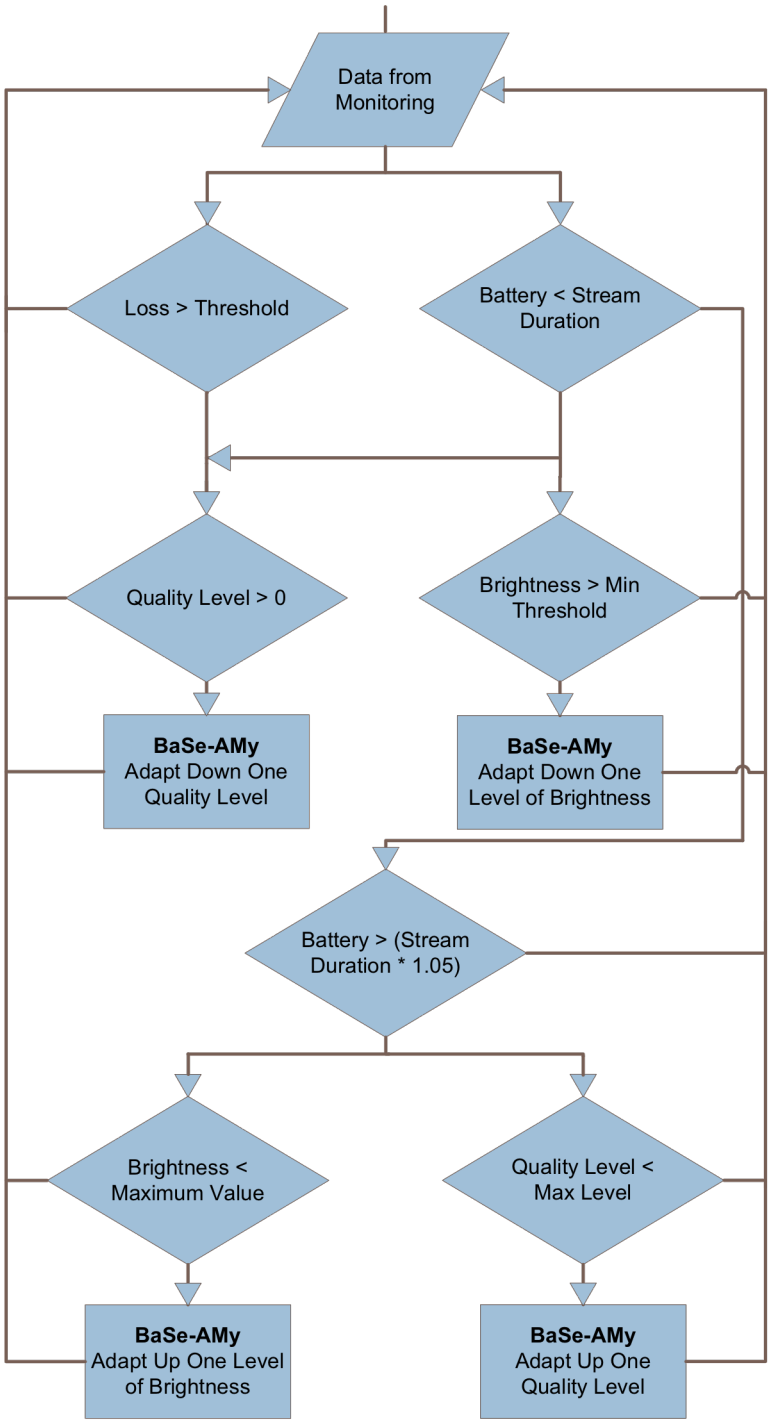

Figure 5.2 - BaSe-AMy Algorithm

5.2 Testing of BaSe-AMy Algorithm

5.2.1 Video Comparison Tool

As mentioned in Chapter 1, a custom video comparison tool was developed as part of the research presented in this thesis. One of the challenges that was faced early in the research presented here was to find a piece of software that would allow for the comparison of two video streams, even if their bitrates were different or if frames were missing from one of the videos.

At the time, 2010, the tool that was being used most for video comparison in the Performance Engineering Laboratory was the MSU Video Quality Measurement Tool [217]. This is a very powerful piece of software but there were significant limitations with this tool at the time:

- Some of the newer video codecs were not supported
- Additionally, for comparing a video that had been transmitted over a network with the original server-side version, if any video frames were lost during transit (UDP streaming for instance), this would throw off which frames were being matched and compared in the two videos. This would mean that every frame after that point in the received video file would be compared against the wrong frame in the reference video.

The second of these issues meant that a new tool was required which would be able to match up the relevant frames in each of the two videos correctly. The VideoCompare tool was created to solve this problem.

The VideoCompare tool has two screens side-by-side as shown in Figure 5.3; one screen shows the play out of the original error-free video, and the second screen shows the play out of the video stream as experienced on the user device with the losses, delays, jitter etc. The video playout of both files is synchronized so that they can compared visually too. Below the play-bar, a graph of the PSNR measurements (made between the two videos) is displayed to give the viewer an estimate of the quality of the received video in comparison to the original source file.

Java Media Framework (JMF) [218] was considered as a base for the design of the side-by-side synchronized video playout, because it enables audio, video and other time-based media to be added to Java applications and applet, but JMF cannot play MPEG-2, MPEG-4,

Figure 5.3 - VideoCompare Tool

Windows Media, Real Media, most QuickTime movies, Flash content newer than Flash 2, and needs a plug-in to play the MP3 format. Also, JMF does not get much maintenance effort from Oracle/Sun.

VLC [219] on the other hand is a media player that is open source, can handle most available codecs and is under constant development. As a result, it was decided that a Java interface for using this application would be ideal.

There were two libraries available for connecting to VLC from a Java application at the time: **JVLC** [220] was a Java binding of VLC that allowed integration of instances of VLC directly into Java programs very neatly. As it was a wrapping of VLC player, it had most of the functionalities that VLC provides including multimedia streaming and supporting of various format of media data (including MPEG2, MPEG4 …). JVLC had negative points too, in that its lack of documentation, examples and support was very problematic. The project ultimately ended in 2010.

Instead the **DJNativeSwing** project [221] was chosen for playback of the videos in VideoCompare. This library also provides a java connection to a VLC installation but this connection is made through the VLC Firefox plugin. The VLC Firefox plugin is normally used to play media from within the browser window. In the DJNativeSwing project, Java uses the API for the plugin to send commands to the VLC application. The reason the DJNativeSwing project was selected for use in this tool is that it was well maintained, documented and supported online.

In order to perform the PSNR calculations between the two videos, it was necessary to decode each frame in the two videos and then compare the raw data. The library that was used for this procedure was **Xuggle** [222]. This library allows for Java applications to connect to a native version of the FFMPEG video processing tool [223]. FFMPEG was used to expose the raw decoded data for each video frame in each stream which was then inserted into the formula for PSNR. Crucially, the frames could be matched based on their Presentation Time-Stamp (PTS) ensuring that the correct frames of each video were being compared with each other.

The VideoCompare tool can be used for calculating the objective quality of two Constant Bit Rate (CBR) streams but it excels at comparing adaptive video streams. Using adaptive streaming to dynamically cope with changing network conditions can improve the user perceived quality at the mobile device. Instead of the user experiencing severe packet loss in the video stream, the video quality is downgraded so that the user can view a continuous stream. This type of solution involves storing the video at a number of specific different quality levels on the server. The quality levels could be named from the highest quality level being called level 1 and the lowest quality level being level 5, for example. The server would begin streaming at one of these quality levels but would switch to another level, if necessary, as dictated by feedback from the client device.

In order to evaluate the objective quality of an adaptive stream received on a client device, additional functionality was added to the VideoCompare tool so that it could reconstruct the reference adaptive video sequence from the original video files, as seen in Figure 5.4. For example if 20 seconds of level 1 quality video is sent, followed by 15 seconds of level 2, and 30 seconds of level 5, then the adapted stream has to be reconstructed, so that it can be compared against the video that the client device received. This is achieved by providing VideoCompare with a log for reconstructing the adapted video file, with the exact timestamps and levels that are involved in each change in quality. These instructions are read into

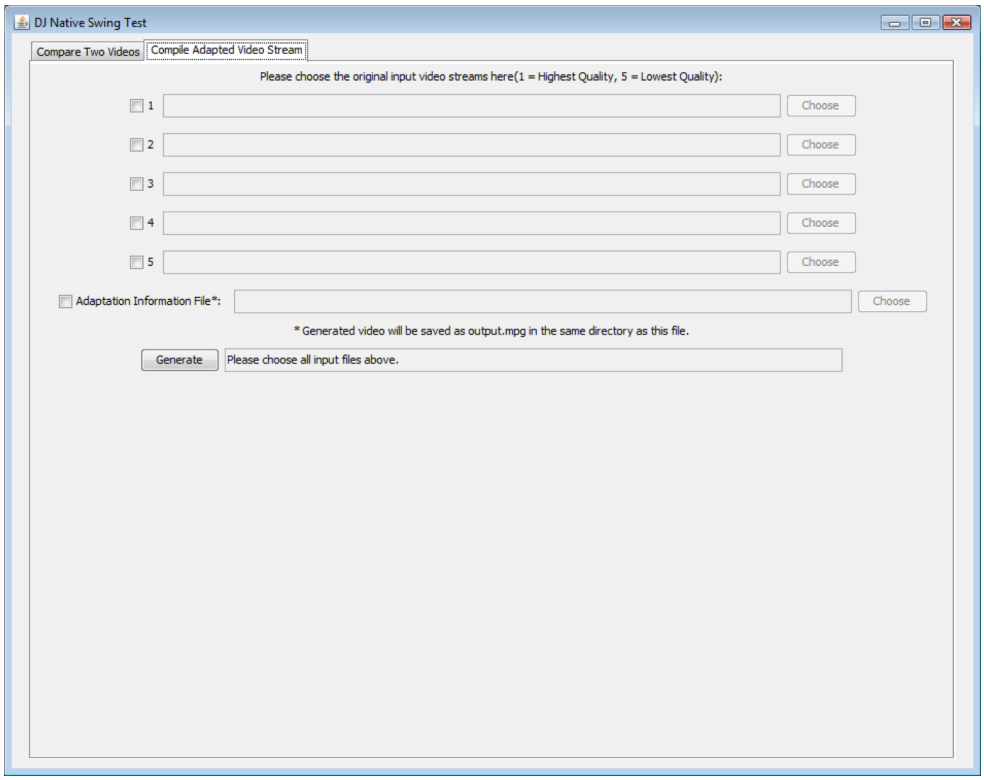

Figure 5.4 - VideoCompare: Adaptive Video Reconstruction

VideoCompare from a text file. For each adaptation order in the file, the tool selects the video frames from the relevant video level which have a PTS between the specified start and end times. These frames are then sent to a new video file which contains the composite data for the adapted stream. As a result of this selection process for gathering the required frames, no frames are lost in the changeover from one quality level to another. Similarly, no sections of any frames are lost when the adaptation occurs because changeover deals with whole frames.

The VideoCompare tool was used in all the video-quality and algorithm evaluations presented in this thesis because of its flexibility and reliability for comparing the correct frames between two video sequences.

5.2.2 Simulation Based Testing

Network Topology and Parameters

The OMNeT++ 4.1 simulator [224] was used for all modeling and simulations. OMNeT++ was chosen for the simulations because of its modular, object-oriented structure, because it has a comprehensive simulation framework available for modeling battery consumption and

because it is open-source. The energy model is available through the MiXiM framework [225], which also contains mobility and wireless networking models. The network topology for BaSe-AMy is that of a client-server, as seen in Figure 5.5.

In order to implement a simple infrastructure-mode network, it was decided that the server and the Wireless Access Point, seen in Figure 5.5, would be collapsed into a single node. The reason that it was desired to have an infrastructure-mode network is so the WNIC is able to use the 802.11 PSM functionality. This functionality is not available in *ad hoc* 802.11 networks, so the test results would not be reflective of a real-world pre-existing streaming application. Sleep mode is not fully catered for in the MiXiM framework so this was implemented manually. A single user network was targeted in the simulations as the video streaming transmission was realized by uni-casting.

Among other parameters, the network configuration involved setting the energy consumption of the WNIC while it is in its different states (sleep, reception and transmission). For this a low-powered Qualcomm WNIC was modeled. The background energy consumption of the Nexus One device during video streaming in strict conditions was measured and subsequently incorporated into the simulations. The nominal battery capacity was also set to 1400 mAh and the voltage to 3.7 V for the simulations. These settings were all chosen so that a real world device, the Nexus One, could be accurately modeled in the simulations.

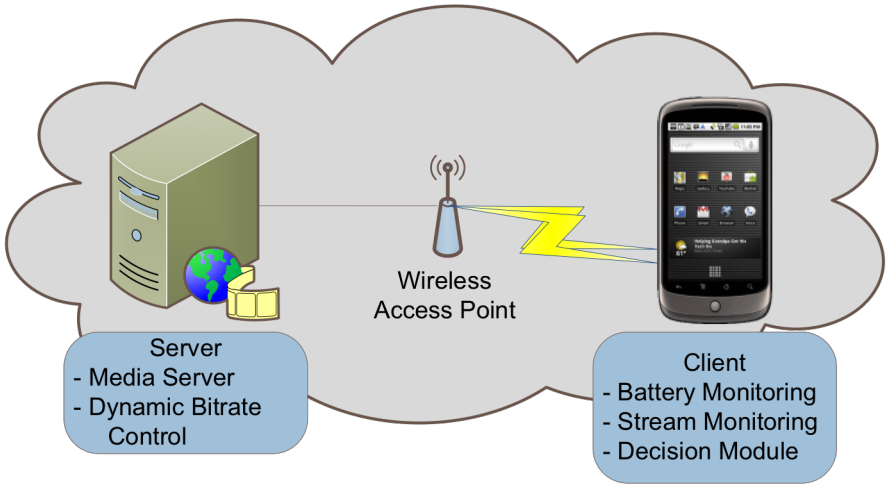

Figure 5.5 – BaSe-AMy Architecture

A simple path-loss model was implemented for the simulation scenarios, where the loss coefficient was set as 4, to mimic the loss in an urban environment. On top of this, the default Omnet++/Mixim channel settings were used for simulating fading on the channel. The configuration for this required the following variables to be set as follows:

- sim.*ver.nic.mac.snr2Mbit = 1.46dB
- sim.*ver.nic.mac.snr5Mbit = 2.6dB
- sim.*ver.nic.mac.snr11Mbit = 5.68dB

The devices in the simulation were portable, but the tests presented here investigate stationary scenarios only.

Video Quality Measurements

The video that was used to illustrate the user perceived quality was a 145 second long high-action video clip. This was transcoded into MPEG4 format to five target bit-rates which correspond to each of the five levels of video quality (400 kbps, 800 kbps, 1.2 Mbps, 1.6 Mbps and 2 Mbps). These levels were selected as they represent a good range of quality levels from poor to the best that can be displayed on a Nexus One device. PSNR is an objective metric for comparing two images (or video frames). PSNR is often used to approximate the user perceived quality of images and videos in the absence of subjective testing. The average PSNR for each quality level was computed with respect to the original video (with a bit-rate of 3 Mbps). This calculation was done on a pixel-by-pixel basis in the corresponding frames of the two videos using (5.1) using the custom video comparison tool created by the author during the course of this research. There is more information on this below. In (5.1) MAX_i stands for the maximum intensity level of a pixel (usually 255) and MSE stands for the Mean Square Error.

$$PSNR = 10 \times \log_{10}\left(\frac{MAX_i^2}{MSE}\right) \tag{5.1}$$

Having calculated the adaptation occurrences in the simulation it was then possible to compare an approximation of the QoS for BaSe-AMy with that of the non-adapting Constant Bit Rate (CBR) stream.

Results & Analysis

Simulations were performed for two different test scenarios.

1. For the first simulation, 16200 seconds of video were streamed to the client. The battery in the client device had enough capacity to receive the whole video clip when the BaSe-AMy mechanism was used. However, while streaming a 2 Mbps CBR version of the video, the client's battery depleted after just 15101 seconds. This happens because the high-quality video stream adds extra load on the Wi-Fi network interface, and raises the power consumption of other components on the device too.

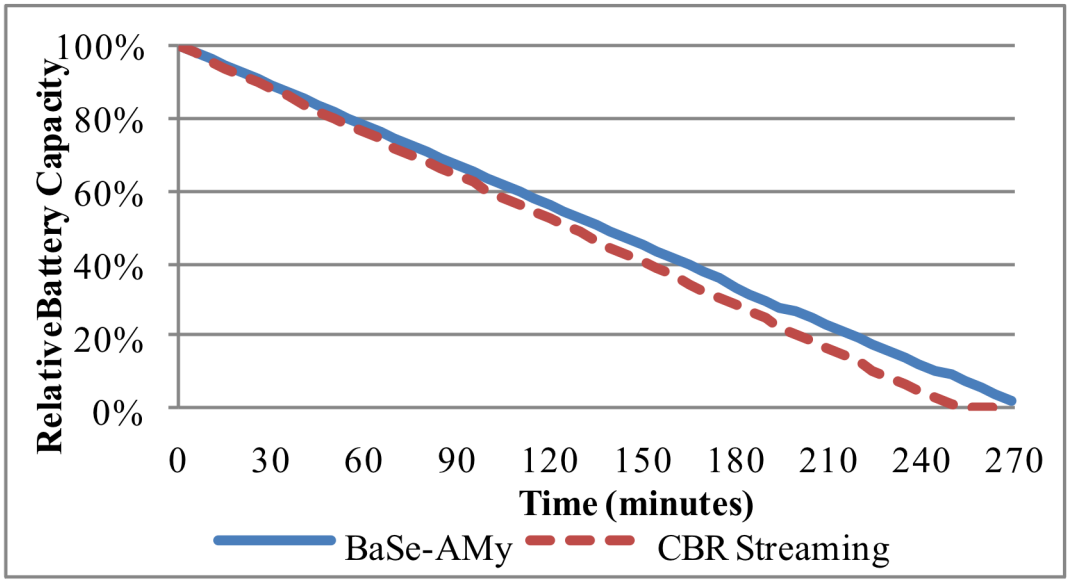

Figure 5.6 – Simulation 1 - Battery Capacity over Time

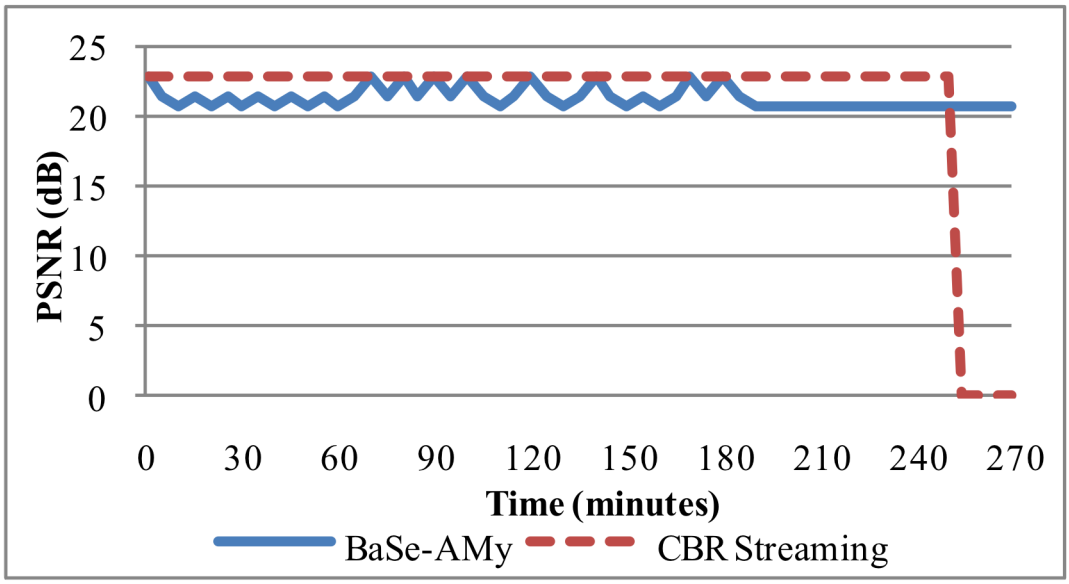

Figure 5.7 – Simulation 1 - PSNR (averaged for each quality level) over Time

In the simulation, the power consumption model reflects this in order to mimic the additional consumption from GPU/RAM on the device. FEC could also be used to help prevent retransmissions because of loss but would add extra network overhead. This first simulation scenario is illustrated in Figure 5.6 and Figure 5.7. Figure 5.6 plots the battery percentage level versus time. The solid blue line of the BaSe-AMy stream allows for a longer duration of video playback than that of the CBR stream because of the adaptations to the video quality. The resultant energy savings mean that the video can be played for an additional 1099 seconds, which could be just long enough for the user to see to the end of a movie or the end of a football match. Figure 5.7 shows the PSNR comparison between the received video file from the CBR stream and the BaSe-AMy stream. The PSNR value of the BaSe-AMy stream is constantly changing which reflects the adaptations between the different quality levels of video. As can be seen the PSNR never dips below 20dB which means that the user can still watch an acceptable quality of video while conserving playback energy.

2. The second simulation involved streaming an 18000 second video, which was too long to be fully received by the client using either the CBR streaming or BaSe-AMy. In this scenario, the CBR stream failed at the same point but the BaSe-AMy stream lasted for 17750 seconds. Figure 5.8 and Figure 5.9 illustrate these results. Figure 5.8 shows that because of a more aggressive quality level adaptation strategy as a result of a longer video stream, the playback duration of the BaSe-AMy stream has increased in this scenario. The benefit of this is that an addition 44 minutes of video can be played on the device over a CBR stream. In Figure 5.9 we can see that the quality level of the BaSe-AMy stream adapts down to the very lowest level during the course of the stream, before the devices battery died at approximately 300 minutes. While this means a reduced QoS to the user, the trade-off is that the user can view a significant amount of additional video footage.

The results in Table 5.1 show a comparison of the BaSe-AMy and CBR streaming mechanisms in terms of client battery duration, throughput and the PSNR of the received video. As expected, in almost every metric, for both simulation-scenarios, BaSe-AMy significantly outperformed the CBR streams, including an increase in battery life of over 17.5% for the second simulation-scenario. The only metric that suffers slightly as a result of the BaSe-AMy algorithm is the average PSNR of the video stream in the first test scenario. This is because the energy savings are achieved by reducing the video quality with the goal of increasing video playback duration. A 1.65dB decrease in video quality is acceptable, particularly when

considering the gains in the battery life of over 7%. Similarly, a 2.63dB decrease in the PSNR is more than acceptable for a video playback duration increase of over 17.5%.

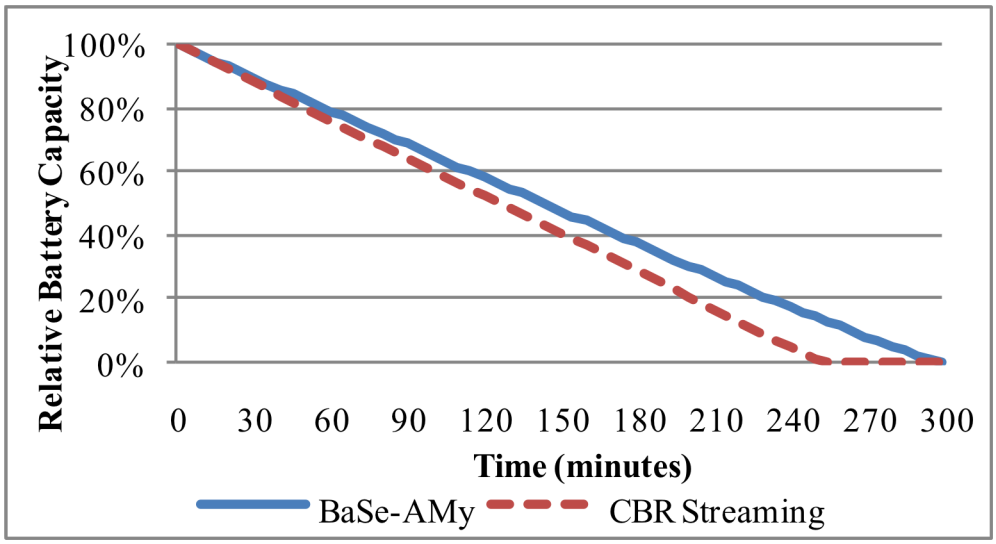

Figure 5.8 – Simulation 2 - Battery Capacity over Time

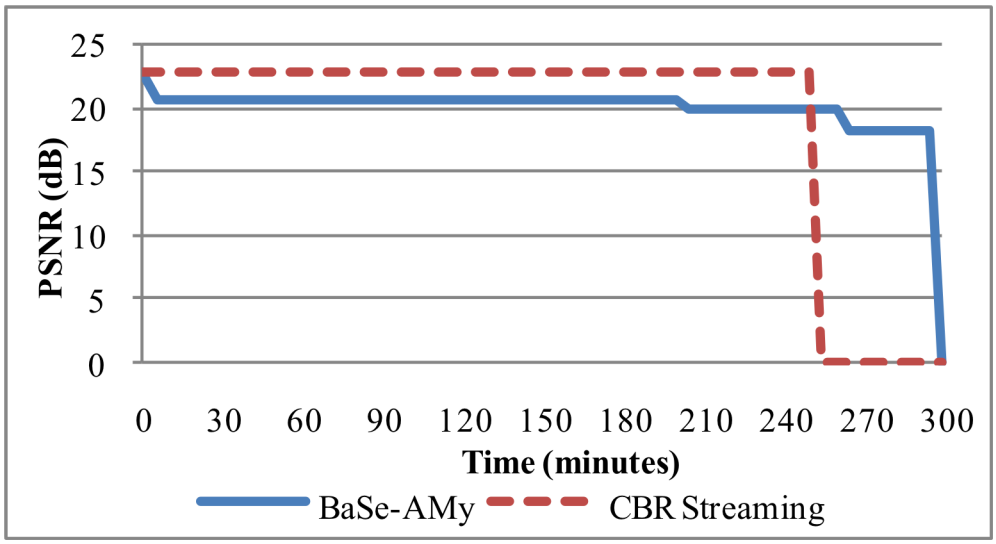

Figure 5.9 – Simulation 2 - PSNR (averaged for each quality level) over Time

Table 5.1 - Comparison of Simulated Streaming Mechanisms

Video Length (s)		CBR	BaSe-AMy	Benefit
16200	Battery Life (s)	15101	>16200	7.28 %
	Avg Throughput (Mbps)	1.86	1.44	22.58%
	Stdev. Throughput (Mbps)	0.50	0.28	-
	Min Throughput (Mbps)	2	1.2	-
	Max Throughput (Mbps	2	2	
	Avg PSNR (dB)	22.92	21.27	-1.65dB
	Stdev. PSNR (dB)	0	0.76	-
	Min PSNR (dB)	22.92	20.77	-
	Max PSNR (dB)	22.92	22.92	-
18000	Battery Life (s)	15101	17750	17.54%
	Avg Throughput (Mbps)	1.68	1.02	39.28%
	Stdev. Throughput	0.70	0.28	-
	Min Throughput (Mbps)	2	0.4	-
	Max Throughput (Mbps	2	2	-
	Avg PSNR (dB)	22.92	20.29	-2.63dB
	Stdev. PSNR (dB)	0	0.85	-
	Min PSNR (dB)	22.92	18.16	-
	Max PSNR (dB)	22.92	22.92	-

5.2.3 Prototype Device Testing

Equipment and Scenarios

As well as simulating the BaSe-AMy algorithm, it was also tested in an adaptive video streaming application on a real mobile device. The complete testing setup can be seen in Figure 5.10. This mechanism is evaluated using a HTC Nexus One running Android 2.3 as the client device in an 802.11g network. This specific device was selected because of its wide range of functionality, its OLED screen and because it runs Android, an Open Source platform. Additionally, Android has integrated functionality for logging the power consumption of the device, which is ideal for the use-case, considered in this paper.

Adobe's RTMP Dynamic Streaming mechanism is selected for the implementation of adaptive video-streaming to the client. This has been selected because it is a viable option across a wide range of mobile and laptop Operating Systems. MPEG DASH or even Apple HLS would be the better choices of mechanisms but unfortunately the adoption of these technologies by mobile Operating Systems is not very mature yet. For MPEG DASH, there is still no support for this in the core Android libraries by default but Google have released a library to enable this functionality on Android 4+ devices [226]. While HLS has been included from Android 3+, it is not as widely adopted as MPEG DASH in the research community. Therefore, Adobe's Flash Media Server 4 [227] is used to serve the adaptive H.264 video stream to an application written in Adobe AIR [228] and compiled for Android. The battery percentage of the Nexus One is measured periodically by the application and these values are used to predict the remaining battery-life. The remaining duration of the stream and the loss can be obtained from the stream metadata in Flash. Adaptations to the brightness of the device's display cannot be implemented in an Adobe AIR application as this functionality is not exposed in the Air SDK. In order to calculate the total energy savings on the device, this issue was circumvented by running the Adobe AIR application and measuring the energy

Figure 5.10 - Test Setup: Server (laptop), Access Point and two Client Devices

consumption and calculating the PSNR for each level of video quality and brightness combination. This information was logged and later compiled for analysis in the graphs shown in the Results and Analysis section here.

Two separate tests were performed to assess the value of the BaSe-AMy algorithm. The video that was streamed during these tests was a 113s high-action video clip that was played on loop for the desired stream duration. This clip was encoded at 15 frames per second, with a resolution of 800x480 pixels using the H.264 codec and an MP4 container. The video was transcoded to 5 different levels of bit-rate: 2 Mbps, 1525 kbps, 1.05 kbps, 575 kbps and 100 kbps. It is important to note that Adobe AIR on Android has poor performance for video stream playback with videos encoded above a bit-rate of 500 kbps. As support for MPEG DASH and Apple HLS increases for Android, there will be no need to use the Adobe mechanism as in this paper. However, these tests are performed here to illustrate the potential for energy savings using the BaSe-AMy algorithm.

Four streaming mechanisms are compared in the tests presented here. The first involves a Constant Bit-Rate (CBR) 2 Mbps stream delivered to a device whose screen brightness is set to 100%. The second makes use of the same CBR stream, but the device has the default OS automatic brightness control enabled. The third mechanism is the BaSe-AMy algorithm with the device screen brightness statically set to 60%. The final mechanism is the BaSe-AMy algorithm with dynamic screen brightness control. Each mechanism is compared in terms of remaining battery capacity and QoS.

Results and Analysis

The results of each of the test scenarios are plotted in Figure 5.11 - Figure 5.14. The labels CBR+ and BaSe-AMy+ are to signify when screen brightness control is enabled. For CBR+, this is the default system scheme but for BaSe-AMy+, the brightness is adaptively controlled in the BaSe-AMy algorithm.

The first test involves streaming 228 minutes of video from the media server to the server to the client device. The CBR and CBR+ mechanisms result in total battery depletion before the whole video sequence can be displayed, as seen in Figure 5.11. However, the BaSe-AMy and BaSe-AMy+ mechanisms enable the full video stream to be played completely, due to the quality adaptations. Additionally, BaSe-AMy+ slightly outperformed BaSe-AMy. It is not

surprising as dynamic brightness control reduces the power-draw of the device's screen. The significance of the results is that the BaSe-AMy algorithm successfully exploits this behavior to return energy savings. In Figure 5.12, it is clear that the PSNR of the CBR streams is higher than that of the BaSe-AMy streams. However, the BaSe-AMy streams retain PSNR values of approximately 50dB throughout the experiment. A PSNR value above 30 dB is considered to be close to visually lossless [229] which means that no difference of quality will be perceptible in the video. For smaller screen devices, such as smart-phones, the PSNR can be even lower before the video quality noticeably deteriorates.

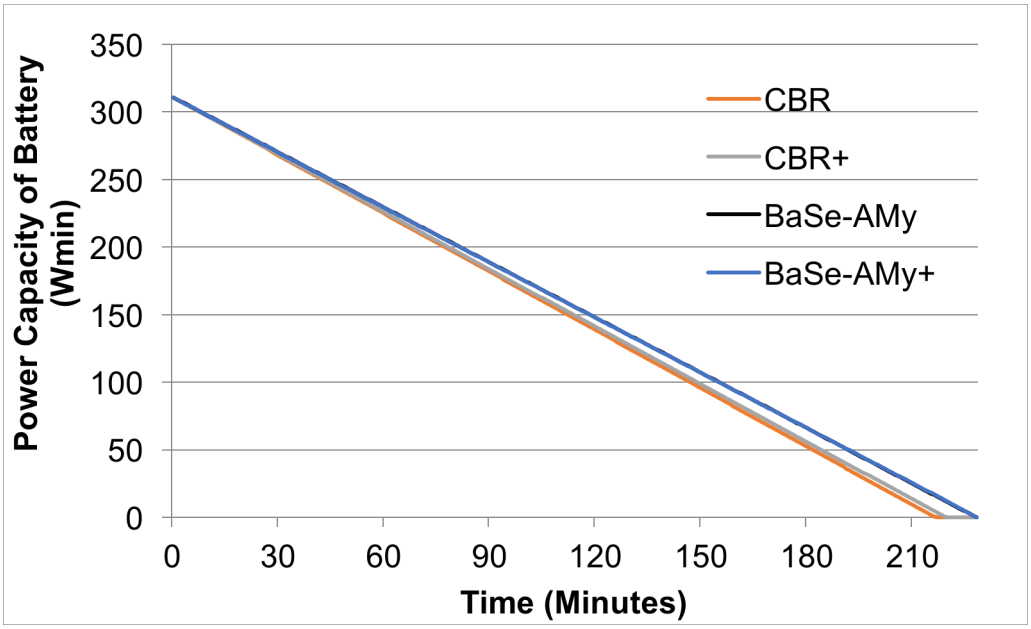

Figure 5.11 – Test 1 - Battery Capacity over Time

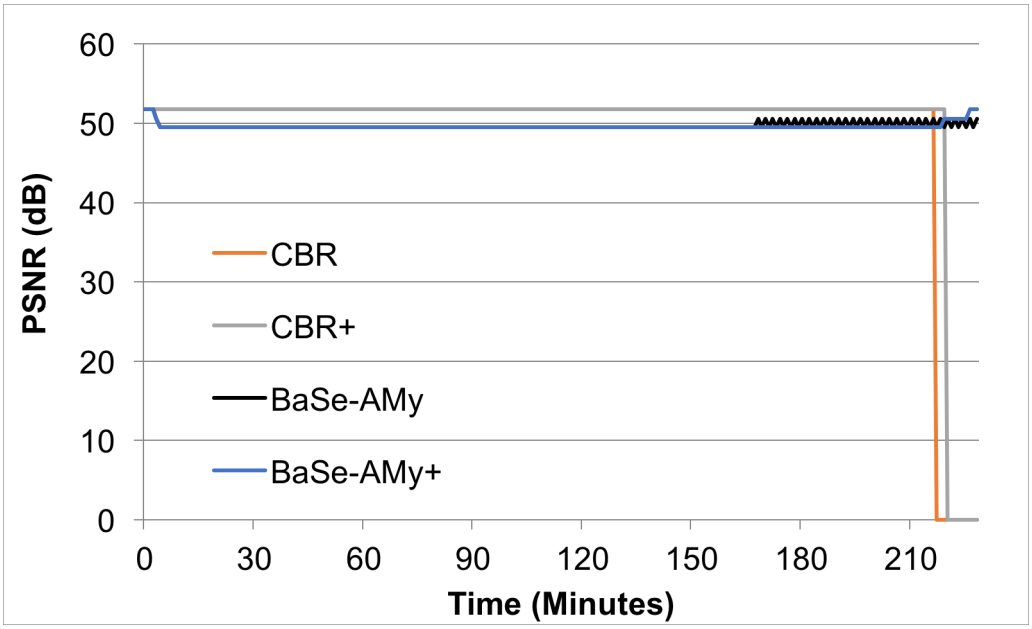

Figure 5.12 – Test 1 - PSNR (averaged for each quality level) over Time

In the second test, the duration of the video stream was increased to 235 minutes. While the CBR and CBR+ mechanisms resulted in total battery depletion in the same length of time as seen in the first test, the BaSe-AMy and BaSe-AMy+ algorithms successfully adapted the stream playback to increase the battery life by 7.6% and 10.1% respectively. Figure 5.13 looks quite similar to Figure 5.11 but on closer inspection of Figure 5.13, the dynamic brightness control of the BaSe-AMy+ stream results in a significant additional energy savings over the

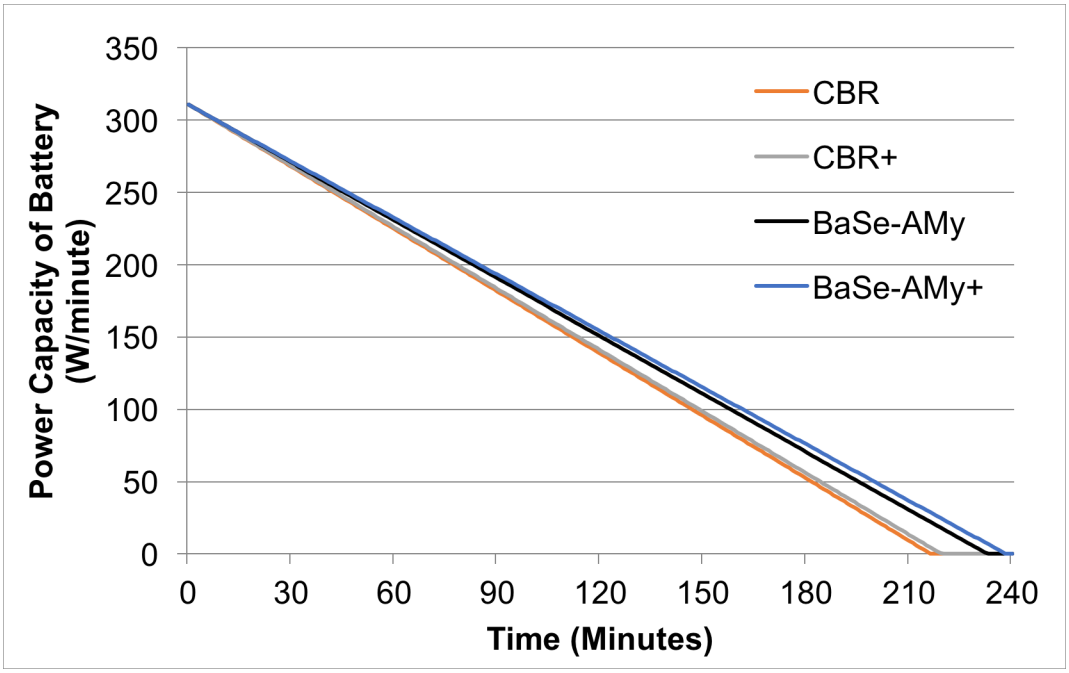

Figure 5.13 – Test 2 - Battery Capacity over Time

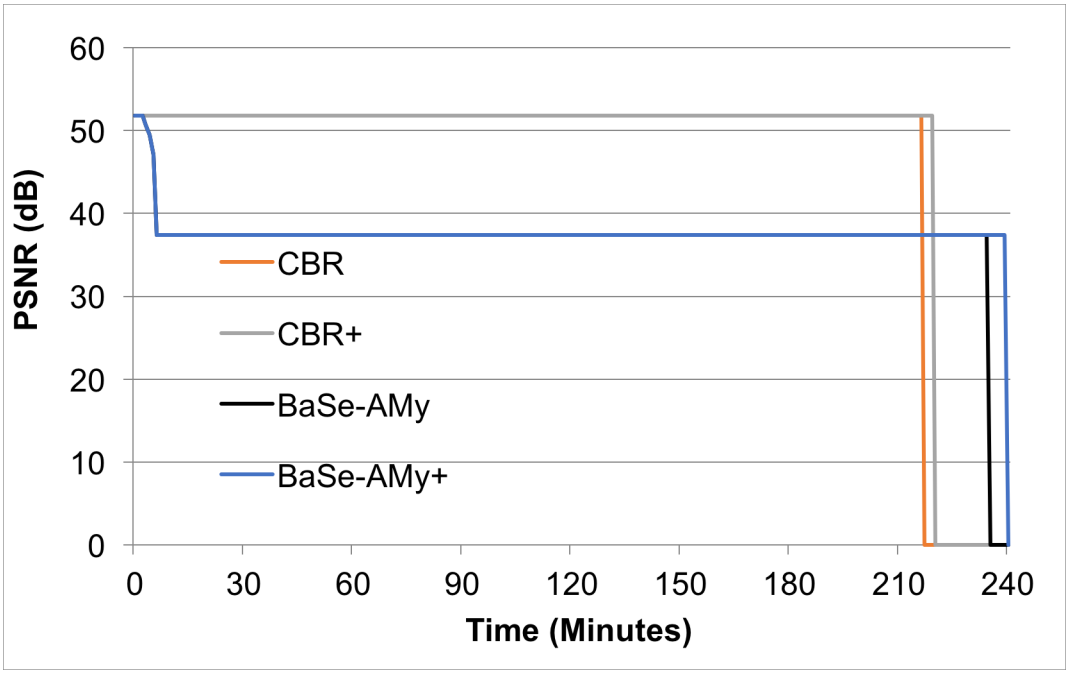

Figure 5.14 – Test 2 - PSNR (averaged for each quality level) over Time

BaSe-AMy stream. This highlights the benefit of adapting the brightness of the device's screen. Figure 5.14 shows that BaSe-AMy immediately adapts video quality to the lowest level. This occurs because the battery-life predictions of the algorithm reveal that the stream duration exceeds the remaining battery-life of the device. The user will not notice a big difference in video quality as the PSNR remains quite high at roughly 37.7dB. This would suggest that for the future work of this project, the BaSe-AMy algorithm should try to adapt the video quality before adapting the brightness of the device's display.

Table 5.2 provides a comparison of results from each of the two tests scenarios. The BaSe-AMy algorithm with the dynamic screen brightness control is the most energy-efficient algorithm analyzed, increasing the battery-life by up to 10% over the CBR stream. This is a significant energy saving with only a minor effect on the QoS, which may not even be noticed by the user.

5.3 Summary

In this chapter, the design of the BaSe-AMy algorithm for energy efficient video streaming is introduced and described in detail. This algorithm assesses the network, device and stream conditions to manipulate the video quality level and the screen brightness, so that as much of the video as possible can be viewed on the device at the highest possible quality level.

Table 5.2 - Comparison of Streaming Mechanisms

					Gain
Test 1 Video Length: 228 Mins	Battery Life	CBR vs CBR+	12978s	13159s	1.39%
		CBR vs BaSe-AMy	12978s	13680s	5.41%
		CBR vs BaSe-AMy+	12978s	13686s	5.45%
	PSNR	CBR vs CBR+	51.76dB	51.76dB	0dB
		CBR vs BaSe-AMy	51.76dB	49.66dB	-1.9dB
		CBR vs BaSe-AMy+	51.76dB	49.57dB	-2.19dB
Test 2 Video Length: 235 Mins	Battery Life	CBR vs CBR+	12978s	13159s	1.39%
		CBR vs BaSe-AMy	12978s	13965s	7.6%
		CBR vs BaSe-AMy+	12978s	14289s	10.1%
	PSNR	CBR vs CBR+	51.76dB	51.76dB	0dB
		CBR vs BaSe-AMy	51.76dB	37.75dB	-14.0dB
		CBR vs BaSe-AMy+	51.76dB	37.74dB	-14.0dB

Following this, the assessment of the BaSe-AMy algorithm is described. There were three separate sections as part of the assessment procedure:

- The first pertained to the custom video comparison tool that was developed as part of this research
- The second was related to simulations that were performed using BaSe-AMy
- The third was related to real-world experimental testing of the BaSe-AMy algorithm, where control of the screen brightness was also performed

For both the simulation-based and the experimental-based tests, BaSe-AMy achieved significant power savings: up to a 17.5% and a 10.1% increase in stream playback duration respectively.

In the next chapter another algorithm that is part of EASE is dealt with.

Chapter 6 - PowerHop

The second major contribution of EASE is the PowerHop algorithm. This algorithm tackles the problem of high power consumption on a mobile device when it is sending video data to another device or server. PowerHop is described in detail in this chapter, from the theory to the design to the evaluation in real-world tests using mobile devices. A new mobile power logger solution was created as part of this work to facilitate measuring the power consumption during the testing of the PowerHop algorithm. This solution is also described in more detail in this chapter.

6.1 PowerHop Theory

Different network technologies on a mobile device have significantly different energy consumption characteristics. For example, [165] shows that the energy consumption per unit time of communications over UMTS and IEEE 802.11b/g interfaces are similar. However, the energy consumption as a function of the data transferred can be up to 300 times larger over the UMTS network interface. Even transmissions on a single type of network can have drastically different energy consumption characteristics. This is because of different signal strengths between the source and destination in the wireless transmission. Such alterations in the signal strength are usually caused by either obstructions in the transmission path or by an increased distance between source and destination. In wireless transmissions, the greater the distance the greater the minimum transmission power required for successful data transfer.

Figure 6.1 shows a network topology with a mobile device that is implementing a multi-hop algorithm for data uploads. The original communication distance between the device and the network AP is denoted as d. By using an intermediary node as an *ad hoc* relay node, now the mobile device only has to transmit data the length of d_1. The intermediary node then relays the data on to the network AP by sending it the remaining distance, d_2. In order to justify the viability of utilizing multi-hop transmissions for conserving energy, we can look at the Friis transmission equation [230].

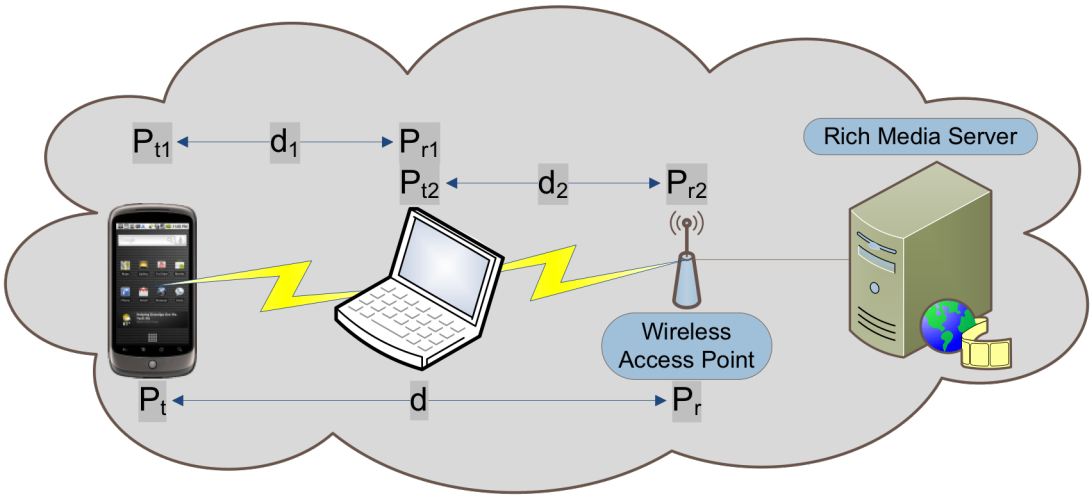

Figure 6.1 - Multi-hop Transmission Scenario

The Friis transmission equation (6.1) is a method of calculating the ratio of the received power, P_r, of a signal to its transmitted power, P_t, based on the gain of the antennas used for the transmission (G_t & G_r), the wavelength of the signal (λ), the transmission distance (d) and the attenuation coefficient of the medium (α).

$$\frac{P_r}{P_t} = G_t G_r \left(\frac{\lambda}{4\pi d} \right)^{\alpha} \qquad (6.1)$$

It is clear from this formula that the received power has an inverse power law relationship with the distance of the data transmission. This means that the more we can reduce the transmission distance, the lower the transmission power required for successful delivery of data will become.

6.2 Architecture

PowerHop tackles the problem of high power consumption in a smart-phone while it is transmitting video content, e.g. videoconferencing or live video broadcasting apps. PowerHop is a novel algorithm for balancing energy saving and quality during mobile video delivery in a wireless network environment. PowerHop, performs adaptations to the transmission power of a device, decides the number of hops to include in the communication route and dynamically scales the video quality. PowerHop assesses network conditions, neighboring

node devices and QoS requirements to decide whether to adapt the transmission power, whether to use a direct or multi-hop route for communication and whether to increase or decrease the video quality level.

The algorithm solves the problem of having to transmit data when the network signal strength is low. For example, when the distance from the source to the destination of the wireless transmission is large, the mobile device can be presented with a list of neighboring wireless devices that could act as intermediary nodes for the transmission. The mobile device can then make an *ad hoc* connection with one of these intermediary nodes and relay all its data transmission through it.

The PowerHop algorithm is implemented within a custom video streaming application. This application has both client and server components. Figure 6.2 depicts the network and three different transmission paths: a direct connection, a two-hop path and a three-hop path. In each case, the smart-phone at the end of the link is serving the video to the tablet in the center of the diagram. While the client device in this case is a tablet computer, this could be

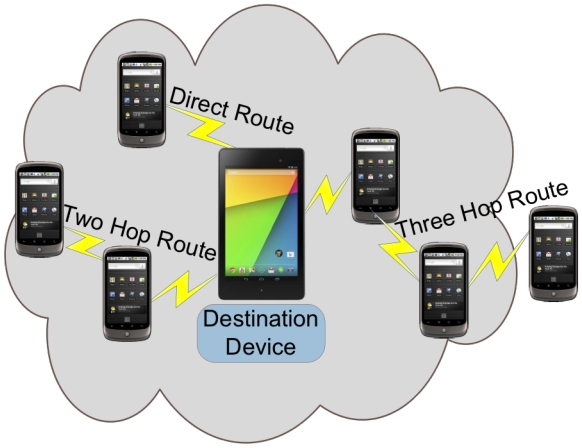

Figure 6.2 - Network Topologies

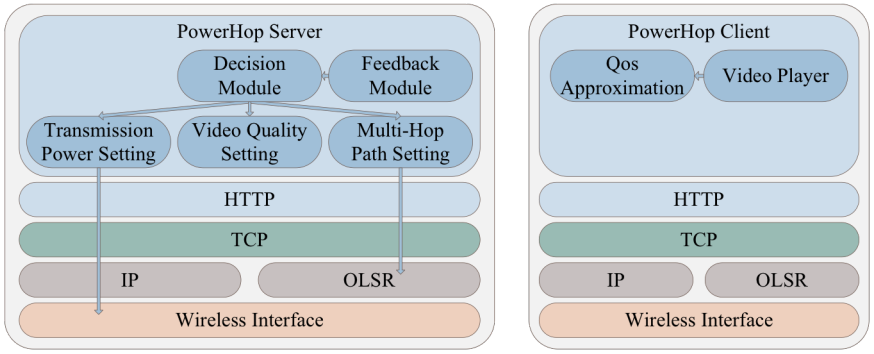

Figure 6.3 - PowerHop Block Diagram

implemented with any device capable of handling the video stream. Similarly, the client device could be on another network, provided there was a gateway to that network.

A block diagram of the individual components of the PowerHop system, and where they are located, can be seen in Figure 6.3. For both the client and the server devices, the base system is identical. Each device is running OLSR for *ad hoc* routing purposes and the video data is transmitted using HTTP. On the client device, there are just two additional modules in the PowerHop architecture. The Video Player module handles the playback of the video stream on the device. The Qos Approximation module records when an error occurs during the video playback. This could come in the form of the buffer being empty or a frame being decoded too late to be displayed. Additionally, using the ping command, the latency on the network is approximated and recorded here too. This information is then used to assess the loss on the network and thus approximate the QoS of the video stream. QoS estimation is performed with the formula as shown in Equation 6.2 [231]. This alternative definition of PSNR allows for calculation of the QoS, and thus approximation of the QoE, without requiring a reference video to compare against. If the video quality levels are changed up or down, the formula will continue to calculate the PSNR correctly for the new quality level, as the numerator will change with respect to each video quality level. The data from this module is sent back to the Feedback module on the PowerHop Server.

$$PSNR = 20\log_{10}\left(\frac{Max\_Bitrate}{|Exp\_Thr - Crt\_Thr|}\right) \tag{6.2}$$

In Equation 6.2, *Max_Bitrate* is the maximum data rate of the transmitted stream, *Exp_Thr* is the expected throughput and *Crt_Thr* is the actual average throughput.

On the PowerHop Server, additional modules provide functionality to read the feedback from the client application. This feeds directly into the decision module, where the PowerHop algorithm runs and decides what settings to use for the outputs. The output settings include changing the transmission power of the Wi-Fi interface, the quality of the video stream and the number of hops used in the transmission route.

The PowerHop algorithm considers the estimated PSNR which is made available to the device through the feedback module. There are two metrics that can affect the PSNR level. These are

the loss and the latency in the communication link. If either of those metrics rises, then the *Crt_Thr* drops. This in turn lowers the PSNR value. Loss and latency on the network can be assumed to have been caused by either a *low SNR* on the communication link or by *network congestion*. The PowerHop algorithm attempts to tackle both of these root issues. Equation 6.3 shows the core formula of the PowerHop algorithm. α is a normalization factor so that the value of U will always be between '0' and '1'. *U* is the utility of the function and is used in PowerHop's decision making.

$$U = \ln(PSNR^{\alpha}) \tag{6.3}$$

When the value of *U* goes below a threshold of '0.7', then PowerHop boosts the transmission power up a level. If *U* drops to below '0.65' then PowerHop switches down one video quality level. If *U* continues to drop and goes below 0.5, PowerHop looks to switch to using a direct route. If the value of *U* increases above '0.7' again, then PowerHop switches up one video quality level. If *U* increases above '0.75' then PowerHop lowers the transmission power by one level. If *U* increases above '0.8' then PowerHop looks for a neighboring node to add as a hop in the transmission path.

As this contribution works on the up-link from the mobile device, it is completely complementary to the BaSe-AMy mechanism.

Table 6.1 - Testing Parameters

			Server Device			Relay Device		
Single Hop			32m	16m	1m			
Two Hop			32m	16m	1m	32m	16m	1m
Tx. Power			32dBm	16dBm	1dBm	32dBm	16dBm	1dBm
Video Rate (Quality Level)	0.3 Mbps	0.9 Mbps	1.5 Mbps	H.264, Baseline				

6.3 PowerHop Testing

Testing is performed using three HTC Nexus One devices, running Android 4.2.2, as the server and relay nodes. The power consumption of the Nexus One devices is measured externally in real-time and logged to SD cards. This is achieved by using an Arduino microcontroller based solution that was designed specially for the purpose, as seen in Figure 6.4. An Asus Nexus 7, running Android 4.3, is used as the client device. The full test-bed setup can be seen in Figure 6.5. The adaptive transmission power control and the selection of the number of hops is implemented through the MANET Manager app on all of the devices [19]. This application performs all the ad hoc routing operations using the OLSR protocol and the PowerHop algorithm functions on top of that.

The testing scenarios involve a HTTP Live Streaming (HLS) video stream to be transmitted from the server device to the client device. This streaming is repeated for different distances, for both single and two-hop routes, for different video quality levels and for different

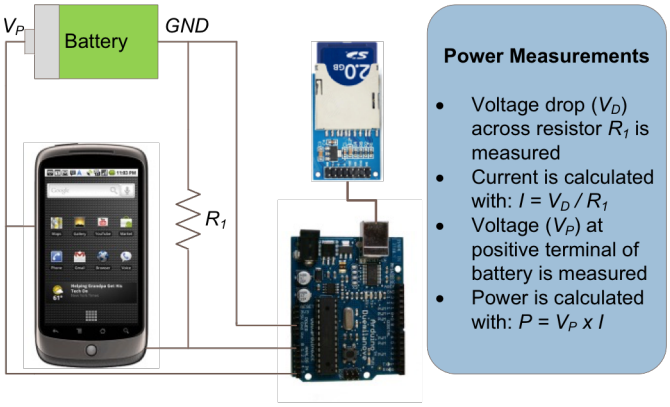

Figure 6.4 - Power Measurement Setup

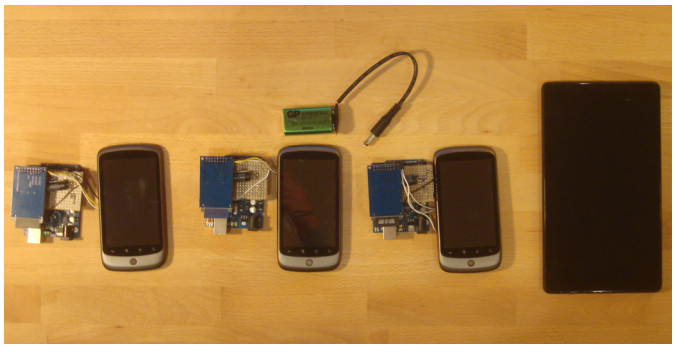

Figure 6.5 – Test-bed Devices

transmission power levels. For each scenario, the loss, latency and power consumption of the stream are recorded. The received video on the client device and the power consumption required for the whole transmission can then be analyzed in order to assess PowerHop's performance. In total, 270 streaming tests were performed, with all the possible permutations of each of the items in Table 6.1. The testing took place in a park that is surrounded by an urban area. Within the park, all tests were performed in an open space with a direct line-of-sight path between each of the devices. The mobile devices were arranged in straight lines for all test scenarios and were kept 1m above the ground for the duration of the tests in an attempt to stabilize the testing conditions as much as possible. Additionally, the wireless spectrum was scanned and the Wi-Fi channel selected specifically to prevent unwanted interference with other networks.

6.3.1 TEST RESULT

The results have been arranged here in such a way as to illustrate three different types of the tests: the first is a study on the effect of transmission power control, the second is a study on the effect of video quality adaptation and the third, analyzes the performance of the proposed PowerHop algorithm. For each of these three arrangements, a transmission distance of 32m was selected. This distance was measured to be the maximum reliable transmission range for the devices in our tests in a direct route. In some of the test cases, the total transmission distance was increased to 33m. This was performed so that the tests could capture the transmission behavior of a 32m hop as part of the multi-hop link. This information is desirable for the purposes of future work, where it could be used to build a model of transmissions over distances of 1m, 16m and 32m for different configurations of multi-hop links. Judging from experience gained while conducting these tests, reducing the testing distance by 1m would have a negligible impact on the measured results.

A Study on the Effect of Transmission Power Adaptation

This first set of results can be seen in Table 6.2. For one level of video quality, the same stream was sent across the network 9 times. For each of these repetitions, the transmission power of the server and relay device were changed and the transmission

Table 6.2 – Study on the Effect of Transmission Power Adaptation

	Server Tx. Distance	Relay Tx. Distance	Video Quality	Tx. Power = 32dBm					Tx. Power = 16dBm					Tx. Power = 1 dBm				
				Loss (%)	PSNR (dB)	Delay (ms)	Server Power (W)	Relay Power (W)	Loss (%)	PSNR (dB)	Delay (ms)	Server Power (W)	Relay Power (W)	Loss (%)	PSNR (dB)	Delay (ms)	Server Power (W)	Relay Power (W)
Single Hop	32m	N/A	0.9 Mbps	0	100	6.1	0.336 (0.509)		0	100	11.5	0.356 (0.529)		0	100	3.9	0.356 (0.529)	
Two Hop	1m	32m	0.9 Mbps	78	12.6	74.3	0.337	0.550	6.25	34.54	41.8	0.377	0.522	31.8	20.4	139.3	0.354	0.635
Two Hop	16m	16m	0.9 Mbps	6.25	34.5	20.8	0.389	0.373	9.1	31.3	32.6	0.407	0.362	6.25	34.5	35.8	0.361	0.336

Table 6.3 – Study on the Effect of Video Quality Adaptation

	Server Tx. Distance	Relay Tx. Distance	Tx. Power	Video Quality = High (1.5 Mbps, 720p)					Video Quality = Medium (0.9 Mbps, 480p)					Video Quality = Low (0.3 Mbps, 426x240px)				
				Loss (%)	PSNR (dB)	Delay (ms)	Server Power (W)	Relay Power (W)	Loss (%)	PSNR (dB)	Delay (ms)	Server Power (W)	Relay Power (W)	Loss (%)	PSNR (dB)	Delay (ms)	Server Power (W)	Relay Power (W)
Single Hop	32m	N/A	16dBm	59.5	10.5	101.4	0.618 (0.791)		0	100	11.5	0.356 (0.529)		0	100	11.0	0.347 (0.520)	
Two Hop	1m	32m	16dBm	18.9	20.5	52.2	0.383	0.604	6.25	34.54	41.8	0.377	0.522	0	100	7.8	0.310	0.382
Two Hop	16m	16m	16dBm	18.9	20.5	145.6	0.466	0.580	9.1	31.3	32.6	0.407	0.362	77.6	22.2	178.9	0.412	0.329

path was modified between two different two-hop routes and a direct route. The single-hop route is just a direct wireless connection between the server and client over a 32m wireless link. In the first of the two-hop paths, the server transmits the video to a neighboring node which is 1m away. This node then relays the data over the remainder of the link to the client device. In the second two-hop route, the relay device is exactly halfway between the server and the client devices. For simplicity of presentation, when changing the transmission power in the two-hop route, the server and relay devices are configured with identical transmission power levels. There is no reason that the two devices cannot be configured with different power levels for a real-life application and this is accounted for in the PowerHop algorithm.

Before looking at the results for these tests in-depth, there are a couple of important items to take note of. For the single-hop route, the tests were performed with no other nodes in the network. This is possible in an experimental setting, but is not practical in a deployment setting. Additional nodes on the network mean that more energy is spent on the device processing routing information. This is particularly true for proactive routing algorithms such as OLSR, which is used in these tests. The "Server Power" column shows the power consumption of the server device for each of the test scenarios. For the single-hop routes, there is an additional number in this cell in brackets. This number refers to the power consumption of the device for the transmission if there is another node on the network. The other node does not have to be involved in the communication, but drives up the power consumption on the server device anyway. There is some additional routing and beacon traffic that would be sent on the network with the addition of an extra node. There could also be a small amount of interference generated by the introduction of an extra node. To fully determine the root cause of this overhead additional experimentation would be required and this will be included as future work for this research. In order to compare like-with-like, comparisons between the power consumption of the single-hop and multi-hop routes will use the updated number which accounts for the overhead of other nodes on the network. Another aspect to note is that the PSNR has been capped at 100dB for identical streams with no distortions (i.e. using eq. (6.2) will result in an infinite value).

The most energy efficient option for this group of test scenarios is to transmit the video content directly, in a single hop. As we can see in Table 6.2, when there are no other devices in the network, this approach results in the lowest power consumption of all the scenarios, while also providing the highest PSNR value. Unfortunately, this option does not perform quite so well when there are other devices in the network. In these cases a power savings of up to 33% can

be achieved on the sender by switching to using a multi-hop route and offloading the long data transmissions onto a neighboring device. The negative aspects of using the two-hop route are that the delay and loss on the network are likely to increase. This can be combated by tuning the transmission power of the devices, as shown in the PowerHop algorithm, for example.

The transmission power is important to the power budget of a device but savings cannot be achieved by simply setting it low. The transmission power needs to be tuned in order to ensure that the loss rate (and thus the number of retransmissions) is kept low. Otherwise the power consumption of the device will increase, negating any benefit from reducing the transmission power. Configuring a high transmission power for devices that are very close to each other can introduce interference and loss on the network. Setting the power too low on the other hand means that the SNR of the communication may not be high enough to transmit over the distance required. Both issues result in the need for more retransmissions for successful delivery of the data (when using TCP at least), which introduces delays, lowers the PSNR and increases the power consumption. In the testing data in this table, this can be seen in the second row of results. When the server and relay devices are each transmitting at 32dBm, we can see the loss rate is high. This is caused by the transmissions from the two devices causing interference with the reception of the data at the destination device. Reducing the transmission power down to 16dBm for both devices immediately reduces the loss and as a result the PSNR increases by 170%. In a related way, the loss rate starts to increase again if the transmission power is reduced too far, as can be seen on the same line, when it is reduced down to 1dBm. This is because the transmission power is not high enough for at least one hop of the transmission.

Effect of Video Quality Adaptation

Table 6.3 shows a slightly different subset of the testing data. For the tests shown in this table, the transmission power is kept constant while, for each set of distances, different video quality levels, i.e. bit-rates, are exploited for reducing the power consumption during video transmission. The pattern here is clear, for each transmission path, the lower the number of bits being sent across the network, the lower the power consumption on both the server and relay device. The last result in the table bucks this trend slightly, but this is an anomaly most likely caused by external interference on the network. Interestingly though, this spurious result also indicates how the network behaves when the network is suffering from high levels of loss due to congestion or interference. In this situation, the low-quality video can still make it

through the network with an acceptable PSNR level. A high-quality video stream would only compound the network issues in this situation and because of the loss and required retransmissions, would have a significantly lower PSNR level upon delivery, while also consuming more power on the server and relay devices. Another example of this is visible on the first line in the same table where the loss rate is 59.5% for a high-quality stream, but when the quality level is reduced, then the loss reduces dramatically too. This highlights how crucial it is to consider the bit-rate of the communication as well as the device transmission power and the communication route, when targeting system-wide power savings. The test with the anomalous data was not repeated due to time constraints, but will be part of the future work for this research.

In Table 6.3, the most efficient option in terms of power consumption is to use the first of the two-hop routes with the lowest quality video. This achieves a power saving of approximately 19%, against streaming the high quality video across the same link.

6.3.2 Performance of PowerHop

The final set of results presented here shows the performance of the PowerHop algorithm and can be seen in Figure 6.6 – Figure 6.9. PowerHop was tested in two scenarios. For both scenarios, a video stream is sent wirelessly over a 32m distance for 2 minutes. Every 10 seconds the PowerHop algorithm repeats to decide whether or not to change the streaming parameters.

In the first scenario, the algorithm has a choice of using the direct path or following a multi-hop route where it transmits the data to a relay node that is 1m away. This relay node then sends the data the remainder of the distance. Figure 6.6 plots the power consumption over time of the system using PowerHop. Additionally, the power consumption of a static stream of the high quality video over the direct path is shown in the background. This consumption rate is shown for when there are only two nodes on the network and also in an adjusted format to include the overhead of other nodes on the network, as described above. In the graph, we can see that the power consumption is decreased by the operation of the PowerHop algorithm. A power savings of 20% is achieved using the PowerHop algorithm or 58% for the adjusted power consumption rate. Figure 6.7 shows the PSNR of the video over time for both the PowerHop system and the static stream. The PowerHop algorithm achieves an increase in the

PSNR of 138% over the static stream. The PSNR of the delivered stream is increased because the PowerHop algorithm reacts to the changing network conditions, and adapts the stream quality, the transmission power on the server device and the number of hops to include in the route dynamically. In comparison, the static stream just tries to send a single level of video with one group of settings, but if network conditions deteriorate, the loss and latency on the network start to affect the playback of the video. This manifests in playback delays and significant visual artefacts in the video stream. As a result, even by switching to lower video quality levels with PowerHop, the quality of service is increased. There is an additional overhead involved in the multi-hop path due to an increase in the power consumption of the relay device. For this scenario, the average power consumption on the relay device was 0.55W. As a result, the power consumption of the whole network increases by approximately 50%, but power can be saved on the server device.

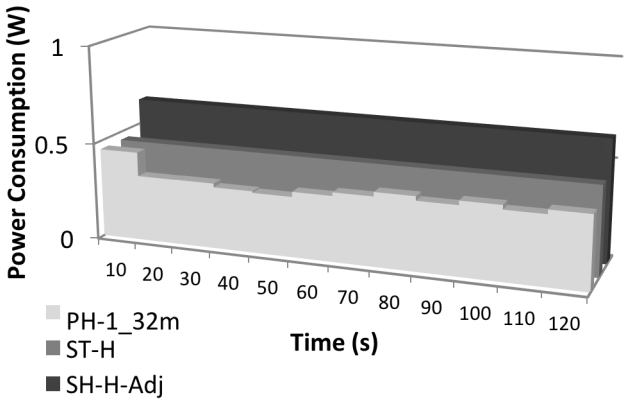

Figure 6.6 - Power v's Time - 1m - 32m Hop

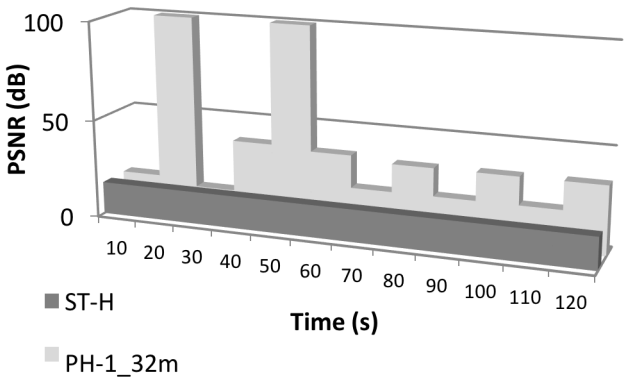

Figure 6.7 - PSNR v's Time - 1m - 32m Hop

In the second scenario, the distances are slightly different. The relay is exactly half way between the sender and receiver in this case. The rest of the parameters remain the same from the first scenario. Figure 6.8 plots the power consumption of the server device over time. As noted before, a clear reduction in the average power consumption is visible. The PowerHop algorithm saves 8% power on the server device over the static route. With the adjusted power consumption of the static route, this increases to 33% power savings. In addition to the power savings, the PSNR of the video stream is increased by 102% in comparison with the static stream case. A plot of the PSNR over time is illustrated in Figure 6.9. In this scenario, the relay device is used to achieve the savings on the server device. As a result, the average power consumption of the relay device increases to 0.44W. The overall network power consumption increases by approximately 50% in this scenario, too.

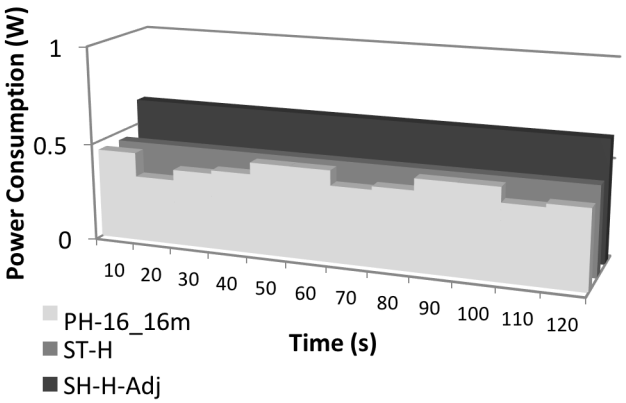

Figure 6.8 - Power v's Time - 16m - 16m Hop

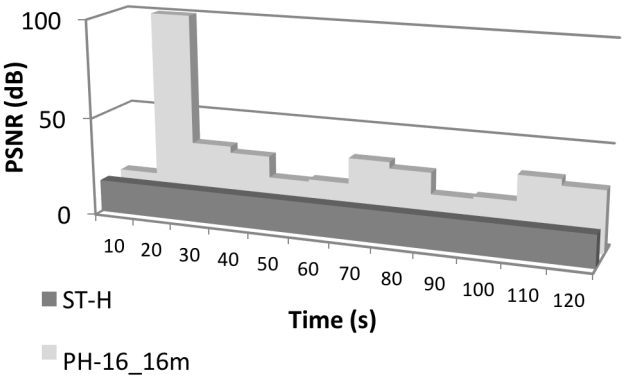

Figure 6.9 - PSNR v's Time - 16m - 16m Hop

6.3.3 CONCLUSION

PowerHop is proposed as an algorithm for increasing the energy efficiency of mobile video transmission applications. These energy savings are achieved by dynamically configuring the transmission power of the device's WNIC, selecting whether or not to use a multi-hop communication route and adapting the quality of the video stream. Real world tests are performed on smart-phones to assess the effect of the algorithm. The results show that power savings of up to 20% can be achieved by using the PowerHop algorithm. In fact a 58% savings can be achieved if there are more than two devices in the network. For this power saving, the PSNR of the transmitted stream has also increased by 138% in comparison with the static delivery of the same video stream.

This thesis demonstrates how power savings can be achieved for the video transmission device when using PowerHop. It is important to note that while using the PowerHop algorithm, although the power savings can be achieved on the video server device, the power consumption for the whole network increases. This occurs because additional burden is spread out onto other network devices.

6.4 Summary

In this chapter, we have seen how the PowerHop algorithm has been designed and implemented for achieving power savings while uploading video content from a mobile device. Additionally, the evaluation of the algorithm with a series of real-world tests on mobile devices was described. In the next chapter, we will look to recap on all of the contributions of EASE and look at what challenges are outstanding as future work.

Chapter 7 - Summary and Future Work

In this chapter a summary of the thesis is presented. This involves looking at all the contributions of the thesis again and recapping how successful each contribution was. Following that, the future tasks involved in the development of EASE are detailed.

7.1 Summary

This thesis introduced a novel Energy-aware Adaptive Solutions (EASE) architecture which is a collection of adaptive energy conservation solutions for smart mobile devices running power-hungry applications. The largest challenge that smart-phones users face regularly is quick battery depletion. Often users will not be able to use their phones for more than a day without recharging them. This issue becomes even more problematic for specific applications that are particularly energy intensive; e.g. video streaming, video conferencing, VoIP, etc. EASE targets one of these application-types, video streaming, in order to provide solutions that significantly prolong device battery-life for both downstream and upstream communications.

Existing approaches to energy-aware video streaming have been analyzed. Additionally, the energy characteristics of the components of a real device (Nexus One) have been measured and based on the aggregation of this information EASE was proposed. EASE combines two independent energy-conservation avenues:

1. The first comes in the form of the Battery and Stream-aware Adaptive Multimedia Delivery algorithm (BaSe-AMy) which functions when a mobile device is playing a video stream. In this scenario, the device is downloading data constantly, decoding it and then playing it back on its screen and speakers. BaSe-AMy reduces the energy consumption involved in this process by dynamically adapting the video quality and screen brightness. This results in energy savings on the WNIC, Display, CPU, GPU, and RAM of the device. In experiments on a Nexus One Android phone BaSe-AMy achieved an increase in battery life by up to 10%.
2. The second avenue is the PowerHop algorithm, which functions when the client device (smart-phone) is uploading content to a server or to another device. In this scenario, the device can scan for neighboring devices that are closer to the gateway

AP of the network. If and when such devices exist, then an *ad hoc* connection can be created between the two devices. Energy savings are achieved in the WNIC in this case because the transmission distance from the client to the intermediary device is less than the distance from the client to the network AP. With this in mind, a lower transmission power can be used on the device for the transmission of data to the intermediary device. Power savings of up to 20% were achieved in experimental tests of the PowerHop algorithm. In fact, savings of 58% were achieved when there were more than two devices in the network.

As well as the algorithmic contributions, this thesis also presented the following additional contributions:

- A comprehensive survey of related technical standards, procedures and the state of the art in the research world was presented in Chapters 2 and 3. A significant portion of this survey was published in the IEEE Communications Surveys & Tutorials Journal [11].
- A video comparison tool was designed and developed so that video sequences could be compared accurately even if some of the video frames were lost during transmission or discarded because they could not be rendered by their Presentation Time Stamp. This tool was used for helping to evaluate the performance of both EASE algorithms in this thesis.
- A battery-powered mobile power logging solution was built to allow for recording the power consumption of the mobile devices in the outdoor testing of the PowerHop algorithm. This solution used an audio amplifier Integrated Circuit to increase the resolution of the power measurement and wrote the logs to an SD card for processing later.

7.2 Future Work for EASE

7.2.1 Additional Innovations for BaSe-AMy

Evaluation of the BaSe-AMy algorithm has been very positive. The first area of focus for the future work will be to will focus on implementing an adaptive streaming solution on mobile devices using MPEG DASH. It has been found that the Adobe Air/Flash solution for the adaptive streaming application is very computationally intensive. It is also quite a limiting approach as many of the low-end android phones cannot support Flash. Adobe Flash for

Android requires an ARMv7 CPU as a minimum specification [138]. This means that devices like the Huawei U8160 (Vodafone 858) [232] and the Samsung Galaxy Europa i5500 [233], which are two of the most affordable Android options available on Irish mobile networks, will not be able to run Adobe Flash applications as they both have ARMv6 CPUs.

The second aspect of BaSe-AMy that could be developed further is the inclusion of extra energy saving techniques into the adaptive streaming solution. Currently BaSe-AMy incorporates adaptive video encoding and dynamic screen brightness control in the algorithm. This functionality could be augmented with download rate control, adaptive decoding and CPU/screen DVS to yield additional energy savings. These approaches were not included at this time because a priority list of components to target with energy savings was derived from the modeling work in Chapter 4. These extra techniques were lower priority than the techniques currently implemented in the BaSe-AMy algorithm.

7.2.2 Additional Innovations for PowerHop

The next step for PowerHop is to develop a model from the real-world test data so that simulations can be performed for larger numbers of devices. Some additional testing will be required as part of the development of this, which will also provide the opportunity to repeat some tests which had anomalous results.

The other remaining challenge for the PowerHop algorithm is to add in some extra techniques that are not currently possible on the application level. These include Wi-Fi channel width manipulation and dynamic disabling of some of the antennas in MIMO WNICs, in order to achieve some power savings. This will be a big task as it requires custom middleware to be developed for the mobile devices to control this behavior, which is not trivial.

7.2.3 Additional Innovations for EASE

The next step for all of the algorithms in the EASE collection is to perform some subjective testing to confirm the results with real test subjects. This was not performed at this time due to time constraints

References

[1] Gartner Inc., "Market Share: Smartphones, Worldwide, 4Q08 and 2008," Mar. 2009.

[2] Gartner Inc., "Competitive Landscape: Mobile Devices, Worldwide, 4Q10 and 2010," Feb. 2011.

[3] Gartner Inc., "Market Share Analysis: Mobile Phones, Worldwide, 4Q12 and 2012," Feb. 2013.

[4] Gartner Inc., "Market Share: Devices, All Countries, 4Q14 Update," Mar. 2015.

[5] Gartner Inc., "Market Share: Final PCs, Ultramobiles and Mobile Phones, All Countries, 4Q16," Feb. 2017.

[6] J. Janek and W. G. Zeier, "A solid future for battery development," *Nature Energy*, vol. 1, no. 9, p. 16141, Sep. 2016.

[7] "Moving Beyond Lithium-Ion," *OEM Off-Highway*. [Online]. Available: //www.oemoffhighway.com/electronics/article/12150133/post-lithiumion-battery-technologies. [Accessed: 28-Aug-2017].

[8] "iPhone 7 - Technical Specifications," *Apple (Ireland)*. [Online]. Available: http://www.apple.com/ie/iphone-7/specs/. [Accessed: 28-Aug-2017].

[9] "Solutions - Cisco's 2017 Visual Networking Index (VNI) Infographic," *Cisco*. [Online]. Available: https://www.cisco.com/c/en/us/solutions/service-provider/visual-networking-index-vni/vni-infographic.html. [Accessed: 28-Aug-2017].

[10] "HTC Google Nexus One - Full phone specifications." [Online]. Available: http://www.gsmarena.com/htc_google_nexus_one-3069.php. [Accessed: 28-Aug-2017].

[11] M. Kennedy, A. Ksentini, Y. Hadjadj-Aoul, and G.-M. Muntean, "Adaptive Energy Optimization in Multimedia-Centric Wireless Devices: A Survey," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 2, pp. 768–786, 2013.

[12] WG8, "ISO/IEC 14443 Standards," *Standing Document 1 (SD1): WG8 Projects*. [Online]. Available: http://wg8.de/sd1.html#14443. [Accessed: 20-Jan-2014].

[13] WG8, "ISO/IEC 15693 Standards," *Standing Document 1 (SD1): WG8 Projects*. [Online]. Available: http://wg8.de/sd1.html#15693. [Accessed: 20-Jan-2014].

[14] "ISO/IEC 18092:2013 - Information technology -- Telecommunications and information exchange between systems -- Near Field Communication -- Interface and Protocol (NFCIP-1)." [Online]. Available: http://www.iso.org/iso/home/store/catalogue_ics/catalogue_detail_ics.htm?csnumber=56692. [Accessed: 20-Jan-2014].

[15] "ISO/IEC 21481:2012 - Information technology -- Telecommunications and information exchange between systems -- Near Field Communication Interface and Protocol -2 (NFCIP-2)." [Online]. Available: http://www.iso.org/iso/home/store/catalogue_ics/catalogue_detail_ics.htm?csnumber=56855. [Accessed: 20-Jan-2014].

[16] "Near Field Communication | Android Developers." [Online]. Available: http://developer.android.com/guide/topics/connectivity/nfc/index.html. [Accessed: 09-Oct-2013].

[17] *IEEE Standard for Information Technology - Telecommunications and Information Exchange Between Systems - Local and Metropolitan Area Networks - Specific Requirements. - Part 15.1 Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specification.* [S.l.]: [s.n.], 2005.

[18] IEEE Computer Society, LAN/MAN Standards Committee, Institute of Electrical and Electronics Engineers, and IEEE-SA Standards Board, *IEEE recommended practice for information technology telecommunications and information exchange between systems-- local and metropolitan area networks-- specific requirements. Part 15.2, Part 15.2,*. New York, NY: Institute of Electrical and Electronics Engineers, 2003.

[19] *IEEE Standard for Information Technology - Telecommunications and Information Exchange Between Systems - Local and Metropolitan Area Networks - Specific Requirements Part 15.3 Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications f.* [S.l.]: [s.n.], 2003.

[20] IEEE Computer Society, LAN/MAN Standards Committee, Institute of Electrical and Electronics Engineers, and IEEE-SA Standards Board, *IEEE standard for local and metropolitan area networks. Part 15.4, Part 15.4,*. New York: Institute of Electrical and Electronics Engineers, 2011.

[21] "ZigBee Specification Overview." [Online]. Available: http://www.zigbee.org/Specifications/ZigBee/Overview.aspx. [Accessed: 20-Jan-2014].

[22] "The Wireless Embedded Internet." [Online]. Available: http://6lowpan.net/. [Accessed: 20-Jan-2014].

[23] "HART Communication Protocol -- WirelessHART." [Online]. Available: http://www.hartcomm.org/protocol/wihart/wireless_overview.html. [Accessed: 20-Jan-2014].

[24] Institute of Electrical and Electronics Engineers, *IEEE recommended practice for information technology telecommunications and information exchange between systems-- local and metropolitan area networks-- specific requirements. Part 15.5, Part 15.5,*. New York: Institute of Electrical and Electronics Engineers, 2009.

[25] Institute of Electrical and Electronics Engineers and IEEE-SA Standards Board, *IEEE standard for local and metropolitan area networks. Part 15.6, Part 15.6,*. New York: Institute of Electrical and Electronics Engineers, 2012.

[26] IEEE Computer Society, LAN/MAN Standards Committee, Institute of Electrical and Electronics Engineers, and IEEE-SA Standards Board, *IEEE standard for local and metropolitan area networks. Part 15.7, Part 15.7,*. New York: Institute of Electrical and Electronics Engineers, 2011.

[27] "IEEE 802.15 WPAN - Task Group 8." [Online]. Available: http://www.ieee802.org/15/pub/TG8.html. [Accessed: 04-Nov-2016].

[28] *IEEE Recommended Practice for Transport of Key Management Protocol (KMP) Datagrams*. Piscataway, USA: IEEE, 2016.

[29] "IEEE Draft Recommended Practice for Routing Packets in 802.15.4 Dynamically Changing Wireless Networks," *IEEE P802.15.10/D10, October, 2016*, pp. 1–150, Jan. 2016.

[30] "Home | Bluetooth Technology Special Interest Group." [Online]. Available: http://www.bluetooth.org. [Accessed: 09-Oct-2013].

[31] "Specification | Adopted Documents | Bluetooth Technology Special Interest Group." [Online]. Available: https://www.bluetooth.org/en-us/specification/adopted-specifications. [Accessed: 09-Oct-2013].

[32] Bluetooth SIG, "Specification of the Bluetooth System - Core Version 2.0 + EDR," Nov. 2004.

[33] Bluetooth SIG, "Specification of the Bluetooth System - Core Version 3.0 + HS," Apr. 2009.

[34] Bluetooth SIG, "Specification of the Bluetooth System - Core Version 4.0," Jun. 2010.

[35] Bluetooth SIG, "Specification of the Bluetooth System - Core Version 4.1," Dec. 2013.

[36] Bluetooth SIG, "Specification of the Bluetooth System - Core Version 4.2," Dec. 2014.

[37] Bluetooth SIG, "Bluetooth Core Specification v5.0," Dec. 2016.

[38] "What is Bluetooth 5? Samsung Galaxy S8's new wireless tech explained," *Trusted Reviews*, 14-Jun-2016. [Online]. Available: http://www.trustedreviews.com/opinion/what-is-bluetooth-5-2947121. [Accessed: 28-Aug-2017].

[39] *IEEE Standard for Information Technology- Telecommunications and Information Exchange Between Systems-Local and Metropolitan Area Networks-Specific Requirements-Part 11 Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications.* [S.l.]: [s.n.], 1997.

[40] IEEE Computer Society, LAN/MAN Standards Committee, Institute of Electrical and Electronics Engineers, and IEEE-SA Standards Board, *Supplement to IEEE standard for Information technology-- telecommunications and information exchange between systems-- local and metropolitan area networks-- specific requirements: part 11 : wireless LAN medium access control (MAC) and physical layer (PHY) specifications : High-speed physical layer in the 5 GHz band*. New York, N.Y., USA: Institute of Electrical and Electronics Engineers, 1999.

[41] IEEE Computer Society, LAN/MAN Standards Committee, Institute of Electrical and Electronics Engineers, and IEEE-SA Standards Board, *Supplement to IEEE standard for Information technology-- telecommunications and information exchange between systems-- local and metropolitan area networks-- specific requirements: part 11 : wireless LAN medium access control (MAC) and physical layer (PHY) specifications : Higher-speed physical layer extension in the 2.4 Ghz band*. New York, N.Y., USA: Institute of Electrical and Electronics Engineers, 2000.

[42] IEEE Computer Society, LAN/MAN Standards Committee, Institute of Electrical and Electronics Engineers, and IEEE-SA Standards Board, *IEEE standard for Information technology: telecommunications and information exchange between systems-- local and metropolitan area networks-- specific requirements. Part 11, Part 11,*. New York, N.Y.: Institute of Electrical and Electronics Engineers, 2003.

[43] Institute of Electrical and Electronics Engineers and IEEE-SA Standards Board, *IEEE standard for information technology - telecommunications and information exchange between systems - local and metropolitan area networks - specific requirements Part 11: wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Amendment 5: enhancements for higher throughput*. New York: Institute of Electrical and Electronics Engineers, 2009.

[44] IEEE Computer Society, LAN/MAN Standards Committee, Institute of Electrical and Electronics Engineers, and IEEE-SA Standards Board, *IEEE standard for information technology: telecommunications and information exchange between systems : local and metropolitan area network-- specific requirements. Part 11, Amendment 4, Part 11, Amendment 4,*. 2013.

[45] IEEE Computer Society, LAN/MAN Standards Committee, Institute of Electrical and Electronics Engineers, and IEEE-SA Standards Board, *IEEE standard for information technology: telecommunications and information exchange between systems : local and metropolitan area networks--specific requirements. Part 11, Amendment 3, Part 11, Amendment 3,*. 2012.

[46] *IEEE Standard for Information technology--Telecommunications and information exchange between systems - Local and metropolitan area networks--Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 2: Sub 1 GHz License Exempt Operation*. New York, USA: IEEE, 2017.

[47] "Wi-Fi HaLow | Wi-Fi Alliance." [Online]. Available: http://www.wi-fi.org/discover-wi-fi/wi-fi-halow. [Accessed: 29-Nov-2016].

[48] IEEE, "IEEE P802.11 – TASK GROUP AX." [Online]. Available: http://www.ieee802.org/11/Reports/tgax_update.htm. [Accessed: 29-Nov-2016].

[49] "IEEE SA - 802.11c-1998 - IEEE Standard for Information Technology - Telecommunications and information exchange between systems - Local area networks - Media access control (MAC) bridges - Supplement for support by IEEE

802.11." [Online]. Available: http://standards.ieee.org/findstds/standard/802.11c-1998.html. [Accessed: 21-Jan-2014].

[50] "IEEE SA - 802.11d-2001 - IEEE Standard for Information technology-- Local and metropolitan area networks-- Specific requirements-- Part 11: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 3: Specification for operation in additional regulatory domains." [Online]. Available: http://standards.ieee.org/findstds/standard/802.11d-2001.html. [Accessed: 21-Jan-2014].

[51] "IEEE SA - 802.11e-2005 - IEEE Standard for Information technology--Local and metropolitan area networks--Specific requirements--Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications - Amendment 8: Medium Access Control (MAC) Quality of Service Enhancements." [Online]. Available: http://standards.ieee.org/findstds/standard/802.11e-2005.html. [Accessed: 21-Jan-2014].

[52] "IEEE SA - 802.11F-2003 - IEEE Trial-Use Recommended Practice for Multi-Vendor Access Point Interoperability Via an Inter-Access Point Protocol Across Distribution Systems Supporting IEEE 802.11 Operation." [Online]. Available: http://standards.ieee.org/findstds/standard/802.11F-2003.html. [Accessed: 21-Jan-2014].

[53] *ISO/IEC Standard for Information Technology- Telecommunications and Information Exchange Between Systems- Local and Metropolitan Area Networks- Specific Requirements- Part 11 Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications Am.* [S.l.]: [s.n.], 2006.

[54] "IEEE SA - 802.11i-2004 - IEEE Standard for information technology-Telecommunications and information exchange between systems-Local and metropolitan area networks-Specific requirements-Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Amendment 6: Medium Access Control (MAC) Security Enhancements." [Online]. Available: http://standards.ieee.org/findstds/standard/802.11i-2004.html. [Accessed: 05-Mar-2014].

[55] "IEEE SA - 802.11j-2004 - IEEE Standard for Information technology--Telecommunications and information exchange between systems-Local and metropolitan area networks--Specific requirements--Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications--Amendment 7:4.9 GHz-5 GHz Operation in Japan." [Online]. Available: https://standards.ieee.org/findstds/standard/802.11j-2004.html. [Accessed: 05-Mar-2014].

[56] "IEEE SA - 802.11k-2008 - IEEE Standard for Information technology-- Local and metropolitan area networks-- Specific requirements-- Part 11: Wireless LAN Medium Access Control (MAC)and Physical Layer (PHY) Specifications Amendment 1: Radio Resource Measurement of Wireless LANs." [Online]. Available: http://standards.ieee.org/findstds/standard/802.11k-2008.html. [Accessed: 05-Mar-2014].

[57] "IEEE SA - 802.11p-2010 - IEEE Standard for Information technology-- Local and metropolitan area networks-- Specific requirements-- Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 6: Wireless Access in Vehicular Environments." [Online]. Available: http://standards.ieee.org/findstds/standard/802.11p-2010.html. [Accessed: 05-Mar-2014].

[58] "IEEE SA - 802.11r-2008 - IEEE Standard for Information technology-- Local and metropolitan area networks-- Specific requirements-- Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 2: Fast

Basic Service Set (BSS) Transition." [Online]. Available: http://standards.ieee.org/findstds/standard/802.11r-2008.html. [Accessed: 05-Mar-2014].

[59] "IEEE SA - 802.11s-2011 - IEEE Standard for Information Technology--Telecommunications and information exchange between systems--Local and metropolitan area networks--Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications Amendment 10: Mesh Networking." [Online]. Available: http://standards.ieee.org/findstds/standard/802.11s-2011.html. [Accessed: 05-Mar-2014].

[60] "IEEE SA - 802.11u-2011 - IEEE Standard for Information Technology-Telecommunications and information exchange between systems-Local and Metropolitan networks-specific requirements-Part II: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Amendment 9: Interworking with External Networks." [Online]. Available: http://standards.ieee.org/findstds/standard/802.11u-2011.html. [Accessed: 05-Mar-2014].

[61] "IEEE SA - 802.11v-2011 - IEEE Standard for Information technology-- Local and metropolitan area networks-- Specific requirements-- Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications Amendment 8: IEEE 802.11 Wireless Network Management." [Online]. Available: http://standards.ieee.org/findstds/standard/802.11v-2011.html. [Accessed: 05-Mar-2014].

[62] "IEEE SA - 802.11w-2009 - IEEE Standard for Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements. Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 4: Protected Management Frames." [Online]. Available: http://standards.ieee.org/findstds/standard/802.11w-2009.html. [Accessed: 05-Mar-2014].

[63] "IEEE SA - 802.11y-2008 - IEEE Standard for Information technology-- Local and metropolitan area networks-- Specific requirements-- Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 3: 3650-3700 MHz Operation in USA." [Online]. Available: http://standards.ieee.org/findstds/standard/802.11y-2008.html. [Accessed: 05-Mar-2014].

[64] "IEEE SA - 802.11z-2010 - IEEE Standard for Information technology-- Local and metropolitan area networks-- Specific requirements-- Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications Amendment 7: Extensions to Direct-Link Setup (DLS)." [Online]. Available: http://standards.ieee.org/findstds/standard/802.11z-2010.html. [Accessed: 05-Mar-2014].

[65] "IEEE SA - 802.11aa-2012 - IEEE Standard for Information technology--Telecommunications and information exchange between systems Local and metropolitan area networks--Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 2: MAC Enhancements for Robust Audio Video Streaming." [Online]. Available: http://standards.ieee.org/findstds/standard/802.11aa-2012.html. [Accessed: 05-Mar-2014].

[66] "IEEE SA - 802.11ae-2012 - IEEE Standard for Information technology--Telecommunications and information exchange between systems--Local and metropolitan area networks--Specific requirements Part 11: Wireless LAN Medium

Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 1: Prioritization of Management Frames." [Online]. Available: http://standards.ieee.org/findstds/standard/802.11ae-2012.html. [Accessed: 05-Mar-2014].

[67] IEEE Computer Society, LAN/MAN Standards Committee, Institute of Electrical and Electronics Engineers, and IEEE-SA Standards Board, *IEEE standard for information technology: telecommunications and information exchange between systems-- Local and metropolitan area networks-- Specific requirements. Part 11, Amendment 5, Part 11, Amendment 5,*. 2014.

[68] *IEEE Standard for Information Technology - Telecommunications and Information Exchange Between Systems - Local and Metropolitan Area Networks - Specific Requirements Part 11 Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications.* [S.l.]: [s.n.], 2005.

[69] *Emerging technologies in wireless LANs: theory, design, and deployment*. Cambridge ; New York: Cambridge University Press, 2008.

[70] Wi-Fi Alliance, "Wi-Fi CERTIFIED Wi-Fi Direct," *White Paper: http://www. wi-fi. org/news_articles. php*, 2010.

[71] "ETSI GPRS." [Online]. Available: http://www.etsi.org/WebSite/technologies/gprs.aspx. [Accessed: 11-Sep-2011].

[72] "ETSI HSCSD." [Online]. Available: http://www.etsi.org/WebSite/Technologies/hscsd.aspx. [Accessed: 11-Sep-2011].

[73] "ETSI EDGE." [Online]. Available: http://www.etsi.org/WebSite/Technologies/edge.aspx. [Accessed: 11-Sep-2011].

[74] "Radiocommunication Sector (ITU-R) - ITU global standard for international mobile telecommunications ´IMT-Advanced´." [Online]. Available: www.imt-2000.org. [Accessed: 29-Jan-2014].

[75] Qualcomm, "Rev. B - Enhanced Mobile Broadband for All," 17-Aug-2011.

[76] "Qualcomm halts UMB project, sees no major job cuts | Reuters." [Online]. Available: http://www.reuters.com/article/2008/11/13/qualcomm-umb-idUSN1335969420081113. [Accessed: 11-Sep-2011].

[77] "3GPP - UMTS." [Online]. Available: http://www.3gpp.org/article/umts. [Accessed: 11-Sep-2011].

[78] "3GPP - HSPA." [Online]. Available: http://www.3gpp.org/HSPA. [Accessed: 11-Sep-2011].

[79] K. Johansson, J. Bergman, D. Gerstenberger, M. Blomgren, and A. Wallén, "Multi-carrier HSPA evolution," in *Vehicular Technology Conference, 2009. VTC Spring 2009. IEEE 69th*, 2009, pp. 1–5.

[80] "3GPP - LTE." [Online]. Available: http://www.3gpp.org/Technologies/Keywords-Acronyms/LTE. [Accessed: 21-Oct-2013].

[81] "Radiocommunication Sector (ITU-R) - ITU global standard for international mobile telecommunications ´IMT-Advanced´." [Online]. Available: http://www.itu.int/ITU-R/index.asp?category=information&rlink=imt-advanced&lang=en. [Accessed: 21-Oct-2013].

[82] WiMAX Forum, "Mobile WiMAX - Part 1: A Technical Overview and Performance Evaluation," Aug. 2006.

[83] "3GPP - LTE-Advanced." [Online]. Available: http://www.3gpp.org/lte-advanced. [Accessed: 22-Oct-2013].

[84] Institute of Electrical and Electronics Engineers and IEEE-SA Standards Board, *IEEE standard for local and metropolitan area networks. Part 16, Amendment 3, Part 16, Amendment 3,*. New York: Institute of Electrical and Electronics Engineers, 2011.

[85] Nokia, "LTE Advanced Pro."

[86] "How fast is 5G?" [Online]. Available: https://5g.co.uk/guides/how-fast-is-5g/. [Accessed: 29-Aug-2017].
[87] "ITU towards 'IMT for 2020 and beyond,'" *ITU*. [Online]. Available: http://www.itu.int:80/en/ITU-R/study-groups/rsg5/rwp5d/imt-2020/Pages/default.aspx. [Accessed: 29-Aug-2017].
[88] "H.261 : Video codec for audiovisual services at p x 384 kbit/s." [Online]. Available: http://www.itu.int/rec/T-REC-H.261-198811-S/en. [Accessed: 30-Jan-2014].
[89] ISO/IEC, "Information technology - Coding of moving pictures and associated audio for digital storage media at up to about 1.5 Mbit/s - Part 2: Video," Jan. 1993.
[90] "H.262 : Information technology - Generic coding of moving pictures and associated audio information: Video." [Online]. Available: http://www.itu.int/rec/T-REC-H.262-201202-I/en. [Accessed: 17-Feb-2014].
[91] "H.263 : Video coding for low bit rate communication." [Online]. Available: http://www.itu.int/rec/T-REC-H.263-200501-I/en. [Accessed: 17-Feb-2014].
[92] "ISO/IEC 14496-2:2004 - Information technology -- Coding of audio-visual objects -- Part 2: Visual." [Online]. Available: http://www.iso.org/iso/iso_catalogue/catalogue_ics/catalogue_detail_ics.htm?csnumber=39259. [Accessed: 17-Feb-2014].
[93] I. Richardson, "White Paper: An Overview of H. 264 Advanced Video Coding," March, 2007.
[94] Alex Eleftheriadis, "SVC and Video Communications," Vidyo, White Paper, 2011.
[95] Brendan Eich, "Video, Mobile, and the Open Web," *Mozilla Hacks*, 18-Mar-2012. [Online]. Available: https://hacks.mozilla.org/2012/03/video-mobile-and-the-open-web/. [Accessed: 14-Nov-2012].
[96] "The WebM Project | Welcome to the WebM Project." [Online]. Available: http://www.webmproject.org/. [Accessed: 02-Nov-2012].
[97] Tsahi Levent-Levi, "H.264 vs VP8: Which is the Better Codec for WebRTC? : BlogGeek.me," *BlogGeek.me*, 18-Sep-2012. [Online]. Available: http://bloggeek.me/webrtc-h264-vp8/. [Accessed: 14-Nov-2012].
[98] C. Feller, J. Wuenschmann, T. Roll, and A. Rothermel, "The VP8 video codec-overview and comparison to H. 264/AVC," in *Consumer Electronics-Berlin (ICCE-Berlin), 2011 IEEE International Conference on*, 2011, pp. 57–61.
[99] K. Shah, "Implementation, Performance Analysis & Comparison of H.264 and VP8."
[100] F. De Simone, L. Goldmann, J. S. Lee, and T. Ebrahimi, "Performance analysis of VP8 image and video compression based on subjective evaluations," in *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, 2011, vol. 8135, p. 19.
[101] Xiph.Org, "Theora, video for everyone." [Online]. Available: http://www.theora.org/. [Accessed: 15-Nov-2012].
[102] "[Asterisk-video] theora." [Online]. Available: http://lists.digium.com/pipermail/asterisk-video/2011-January/003334.html. [Accessed: 14-Nov-2012].
[103] Till Halbach, "A performance assessment of the royalty-free and open video compression specifications Dirac, Dirac Pro, and Theora and their open-source implementations," Mar-2009. [Online]. Available: http://etill.net/projects/dirac_theora_evaluation/. [Accessed: 14-Nov-2012].
[104] "Dirac." [Online]. Available: http://diracvideo.org/. [Accessed: 15-Nov-2012].
[105] A. Ravi and K. R. Rao, "Performance Analysis and Comparison of the Dirac Video Codec with H.264/MPEG-4 Part 10 AVC," *International Journal of Wavelets, Multiresolution and Information Processing*, vol. 09, no. 04, pp. 635–654, Jul. 2011.
[106] S. Issa and O. O. Khalifa, "Performance analysis of Dirac video codec with H. 264/AVC," in *Computer and Communication Engineering (ICCCE), 2010 International Conference on*, 2010, pp. 1–6.

[107] S. SECTOR and O. ITU, “SERIES H: AUDIOVISUAL AND MULTIMEDIA SYSTEMS Infrastructure of audiovisual services–Coding of moving video - High Eficiency Video Coding.”
[108] “The WebM Project | VP9 Video Codec Summary.” [Online]. Available: http://www.webmproject.org/vp9/. [Accessed: 22-Oct-2013].
[109] J. Ozer, “A Progress Report: The Alliance for Open Media and the AV1 Codec,” *Streaming Media Magazine*, 20160412. [Online]. Available: http://www.streamingmedia.com/Articles/ReadArticle.aspx?ArticleID=110383. [Accessed: 30-Nov-2016].
[110] Apple Inc., “HTTP Live Streaming Resources – Apple Developer.” [Online]. Available: http://developer.apple.com/resources/http-streaming/. [Accessed: 09-May-2011].
[111] Apple Inc., “HTTP Live Streaming - draft-pantos-http-live-streaming-20,” 20-Sep-2016. [Online]. Available: https://tools.ietf.org/html/draft-pantos-http-live-streaming-20. [Accessed: 01-Jan-2017].
[112] Kevin Towes, “Sneak Peek: Future Adobe technology for HTTP streaming across multiple devices « Kevin Towes on Online Video at Adobe.” [Online]. Available: http://blogs.adobe.com/ktowes/2011/04/sneak-peak-future-adobe-technology-for-http-streaming-across-multiple-devices.html. [Accessed: 09-May-2011].
[113] “Android 3.0 Platform | Android Developers.” [Online]. Available: http://developer.android.com/sdk/android-3.0.html. [Accessed: 09-May-2011].
[114] Adobe Systems Incorporated, “HTTP Dynamic Streaming Datasheet.” [Online]. Available: http://wwwimages.adobe.com/content/dam/Adobe/en/products/hds-dynamic-streaming/pdfs/hds_datasheet.pdf. [Accessed: 01-Jan-2017].
[115] A. Zambelli, “IIS smooth streaming technical overview,” *Microsoft Corporation*, 2009.
[116] David Hassoun, “Dynamic streaming in Flash Media Server 3.5 – Part 1: Overview of the new capabilities | Adobe Developer Connection.” [Online]. Available: http://www.adobe.com/devnet/flashmediaserver/articles/dynstream_advanced_pt1.html. [Accessed: 09-May-2011].
[117] “ISO/IEC 23009-1:2014 - Information technology -- Dynamic adaptive streaming over HTTP (DASH) -- Part 1: Media presentation description and segment formats.” [Online]. Available: http://www.iso.org/iso/home/store/catalogue_ics/catalogue_detail_ics.htm?csnumber=65274. [Accessed: 01-Jan-2017].
[118] “bitmovin/libdash · GitHub.” [Online]. Available: https://github.com/bitmovin/libdash. [Accessed: 22-Oct-2013].
[119] “HTML5.” [Online]. Available: http://www.w3.org/TR/html5/. [Accessed: 22-Oct-2013].
[120] ITU Telecommunication Standardization Sector, “ITU-T Recommendation G.1010,” *ITU-T G-Series Recommendations*, Nov. 2001.
[121] K. ur R. Laghari, N. Crespi, B. Molina, and C. E. Palau, “QoE Aware Service Delivery in Distributed Environment,” 2011, pp. 837–842.
[122] Z. Wang, H. R. Sheikh, and A. C. Bovik, “Objective video quality assessment,” in *The handbook of video databases: design and applications*, 2003, pp. 1041–1078.
[123] “Peak Signal-to-Noise Ratio as an Image Quality Metric - National Instruments.” [Online]. Available: http://www.ni.com/white-paper/13306/en/. [Accessed: 29-Aug-2017].
[124] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image quality assessment: From error visibility to structural similarity,” *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.

[125] Z. Wang, E. P. Simoncelli, and A. C. Bovik, "Multiscale structural similarity for image quality assessment," in *Signals, Systems and Computers, 2003. Conference Record of the Thirty-Seventh Asilomar Conference on*, 2003, vol. 2, pp. 1398–1402.
[126] D. M. Chandler and S. S. Hemami, "VSNR: A Wavelet-Based Visual Signal-to-Noise Ratio for Natural Images," *IEEE Transactions on Image Processing*, vol. 16, no. 9, pp. 2284–2298, Sep. 2007.
[127] M. H. Pinson and S. Wolf, "A New Standardized Method for Objectively Measuring Video Quality," *IEEE Transactions on Broadcasting*, vol. 50, no. 3, pp. 312–322, Sep. 2004.
[128] H. R. Sheikh and A. C. Bovik, "Image information and visual quality," *IEEE Transactions on Image Processing*, vol. 15, no. 2, pp. 430–444, Feb. 2006.
[129] K. Seshadrinathan and A. C. Bovik, "Motion Tuned Spatio-Temporal Quality Assessment of Natural Videos," *IEEE Transactions on Image Processing*, vol. 19, no. 2, pp. 335–350, Feb. 2010.
[130] ITU-T, "Subjective video quality assessment methods for multimedia applications," 1999.
[131] ITU-T, "Subjective audiovisual quality assessment methods for multimedia applications," International Telecommunication Union, 1997.
[132] ITU-T, "Interactive test methods for audiovisual communications," May 2000.
[133] ITU-R, "Recommendation ITU-R BT.500-13 - Methodology for the subjective assessment of the quality of television pictures," Jan. 2012.
[134] A. K. Moorthy, L. K. Choi, A. C. Bovik, and G. de Veciana, "Video Quality Assessment on Mobile Devices: Subjective, Behavioral and Objective Studies," *IEEE Journal of Selected Topics in Signal Processing*, vol. 6, no. 6, pp. 652–671, Oct. 2012.
[135] K. Seshadrinathan, R. Soundararajan, A. C. Bovik, and L. K. Cormack, "Study of Subjective and Objective Quality Assessment of Video," *IEEE Transactions on Image Processing*, vol. 19, no. 6, pp. 1427–1441, Jun. 2010.
[136] P. Bahl, A. Adya, J. Padhye, and A. Walman, "Reconsidering wireless systems with multiple radios," *ACM SIGCOMM Computer Communication Review*, vol. 34, no. 5, pp. 39–46, 2004.
[137] D. Halperin, B. Greenstein, A. Sheth, and D. Wetherall, "Demystifying 802.11 n power consumption," in *Proceedings of the 2010 international conference on Power aware computing and systems*, 2010, p. 1.
[138] Adobe Systems Incorporated, "Rich Internet Applications | Adobe Flash Player system requirements." [Online]. Available: http://www.adobe.com/products/flashplayer/systemreqs/index.html#mobile. [Accessed: 01-Jan-2017].
[139] "MediaCodec | Android Developers." [Online]. Available: https://developer.android.com/reference/android/media/MediaCodec.html. [Accessed: 30-Dec-2016].
[140] "AV Foundation for iOS and macOS - Apple Developer." [Online]. Available: https://developer.apple.com/av-foundation/. [Accessed: 30-Dec-2016].
[141] Aaron Carroll and Gernot Heiser, "An analysis of power consumption in a smartphone," presented at the Proceedings of the 2010 USENIX conference on USENIX annual technical conference (USENIXATC'10), Boston, USA, 2010.
[142] W.-C. Cheng, Y. Hou, and M. Pedram, "Power minimization in a backlit TFT-LCD display by concurrent brightness and contrast scaling," in *Proceedings of the conference on Design, automation and test in Europe-Volume 1*, 2004, p. 10252.
[143] M. Ruggiero, A. Bartolini, and L. Benini, "DBS4video: dynamic luminance backlight scaling based on multi-histogram frame characterization for video streaming application," in *Proceedings of the 8th ACM international conference on Embedded software*, 2008, pp. 109–118.

[144] A. Iranli, W. Lee, and M. Pedram, "HVS-Aware Dynamic Backlight Scaling in TFT-LCDs," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 14, no. 10, pp. 1103–1116, Oct. 2006.

[145] L. Cheng, S. Mohapatra, M. El Zarki, N. Dutt, and N. Venkatasubramanian, "Quality-Based Backlight Optimization for Video Playback on Handheld Devices," *Advances in Multimedia*, vol. 2007, pp. 1–10, 2007.

[146] P.-C. Hsiu, C.-H. Lin, and C.-K. Hsieh, "Dynamic backlight scaling optimization for mobile streaming applications," in *Proceedings of the 17th IEEE/ACM international symposium on low-power electronics and design*, 2011, pp. 309–314.

[147] Z. Yan and C. W. Chen, "RnB: rate and brightness adaptation for rate-distortion-energy tradeoff in HTTP adaptive streaming over mobile devices," 2016, pp. 308–319.

[148] Y. Liu *et al.*, "GoCAD: GPU-Assisted Online Content-Adaptive Display Power Saving for Mobile Devices in Internet Streaming," in *Proceedings of the 25th International Conference on World Wide Web*, 2016, pp. 1329–1338.

[149] D. Shin, Y. Kim, N. Chang, and M. Pedram, "Dynamic voltage scaling of OLED displays," in *Design Automation Conference (DAC), 2011 48th ACM/EDAC/IEEE*, 2011, pp. 53–58.

[150] M. Dong, Y. S. . Choi, and L. Zhong, "Power modeling of graphical user interfaces on OLED displays," in *Design Automation Conference, 2009. DAC'09. 46th ACM/IEEE*, 2009, pp. 652–657.

[151] M. Dong, Y. S. . Choi, and L. Zhong, "Power-saving color transformation of mobile graphical user interfaces on OLED-based displays," in *Proceedings of the 14th ACM/IEEE international symposium on Low power electronics and design*, 2009, pp. 339–342.

[152] M. Dong and L. Zhong, "Chameleon: A Color-Adaptive Web Browser for Mobile OLED Displays," *Arxiv preprint arXiv:1101.1240*, 2010.

[153] S. Kim, S. Hyun, T. Heo, D. Im, and J. Huh, "Blind: Power saving color transform method for OLED displays," in *2016 IEEE International Conference on Consumer Electronics (ICCE)*, 2016, pp. 500–501.

[154] B. Anand *et al.*, "Adaptive display power management for mobile games," in *Proceedings of the 9th international conference on Mobile systems, applications, and services*, 2011, pp. 57–70.

[155] A. M. Alt and D. Simon, "Control strategies for H. 264 video decoding under resources constraints," in *Proceedings of the Fifth International Workshop on Feedback Control Implementation and Design in Computing Systems and Networks*, 2010, pp. 13–18.

[156] W. Yu, X. Jin, and S. Goto, "Temporal scalable decoding process with frame rate conversion method for surveillance video," *Advances in Multimedia Information Processing-PCM 2010*, pp. 297–308, 2011.

[157] S. Park, Y. Lee, J. Lee, and H. Shin, "Quality-adaptive requantization for low-energy MPEG-4 video decoding in mobile devices," *Consumer Electronics, IEEE Transactions on*, vol. 51, no. 3, pp. 999–1005, 2005.

[158] A. Yang and M. Song, "Aggressive dynamic voltage scaling for energy-aware video playback based on decoding time estimation," in *Proceedings of the seventh ACM international conference on Embedded software*, Grenoble, France, 2009, pp. 1–10.

[159] C. C. Yang, K. Wang, M. H. Lin, and P. Lin, "Energy Efficient Intra-Task Dynamic Voltage Scaling for Realistic CPUs of Mobile Devices," *Journal of Information Science and Engineering*, vol. 25, no. 1, pp. 251–272, 2009.

[160] Y. Gu and S. Chakraborty, "Power Management of Interactive 3D Games Using Frame Structures," 2008, pp. 679–684.

[161] R. Kravets and P. Krishnan, "Application-driven power management for mobile communication," *Wireless Networks*, vol. 6, no. 4, pp. 263–277, 2000.

[162] Ali Saidi, Chuntao Zhang, Silviu Chiricescu, Loren J. Rittle, and Yang Yu, "Battery-Aware Localization in Wireless Networks," presented at the 2009 IEEE 34th Conference on Local Computer Networks (LCN 2009), Zürich, Switzerland, 2009.

[163] V. Namboodiri and Lixin Gao, "Energy-Efficient VoIP over Wireless LANs," *IEEE Trans. on Mobile Comput.*, vol. 9, no. 4, pp. 566–581, Apr. 2010.

[164] J. Manweiler and R. R. Choudhury, "Avoiding the rush hours: WiFi energy management via traffic isolation," in *Proceedings of the 9th international conference on Mobile systems, applications, and services*, 2011, pp. 253–266.

[165] H. Petander, "Energy-aware network selection using traffic estimation," in *Proceedings of the 1st ACM workshop on Mobile internet through cellular networks*, 2009, pp. 55–60.

[166] R. Trestian, O. Ormond, and G. M. Muntean, "Power-friendly access network selection strategy for heterogeneous wireless multimedia networks," in *Broadband Multimedia Systems and Broadcasting (BMSB), 2010 IEEE International Symposium on*, 2010, pp. 1–5.

[167] H. Mahkoum, B. Sarikaya, and A. Hafid, "A framework for power management of handheld devices with multiple radios," in *Wireless Communications and Networking Conference, 2009. WCNC 2009. IEEE*, 2009, pp. 1–6.

[168] R. Friedman, A. Kogan, and Y. Krivolapov, "On power and throughput tradeoffs of WiFi and Bluetooth in smartphones," in *2011 Proceedings IEEE INFOCOM*, Shanghai, China, 2011, pp. 900–908.

[169] SuKyoung Lee, K. Sriram, Kyungsoo Kim, Yoon Hyuk Kim, and N. Golmie, "Vertical Handoff Decision Algorithms for Providing Optimized Performance in Heterogeneous Wireless Networks," *IEEE Trans. Veh. Technol.*, vol. 58, no. 2, pp. 865–881, Feb. 2009.

[170] A. Ibrahim, Zhu Han, and K. J. R. Liu, "Distributed energy-efficient cooperative routing in wireless networks," *IEEE Trans. Wireless Commun.*, vol. 7, no. 10, pp. 3930–3941, Oct. 2008.

[171] Mohammad Z. Siam, M. Krunz, S. Cui, and A. Muqattash, "Energy-efficient protocols for wireless networks with adaptive MIMO capabilities," *Wireless Netw*, vol. 16, no. 1, pp. 199–212, Jul. 2008.

[172] A. Radwan and H. S. Hassanein, "NXG04-3: Does Multi-hop Communication Extend the Battery Life of Mobile Terminals?," in *Global Telecommunications Conference, 2006. GLOBECOM'06. IEEE*, 2006, pp. 1–5.

[173] Xiaoan Lu, E. Erkip, Yao Wang, and D. Goodman, "Power efficient multimedia communication over wireless channels," *IEEE Journal on Selected Areas in Communications*, vol. 21, no. 10, pp. 1738–1751, Dec. 2003.

[174] C. Bouras, S. Charalambides, K. Stamos, S. Stroumpis, and G. Zaoudis, "Power Management Mechanism Exploiting Network and Video Information over Wireless Links," *Journal of Networks*, vol. 8, no. 1, Jan. 2013.

[175] M. T. Refaei, L. A. DaSilva, M. Eltoweissy, and T. Nadeem, "Adaptation of Reputation Management Systems to Dynamic Network Conditions in Ad Hoc Networks," *IEEE Transactions on Computers*, vol. 59, no. 5, pp. 707–719, May 2010.

[176] B. B. Chen and M. C. Chan, "Mobicent: a credit-based incentive system for disruption tolerant network," in *INFOCOM, 2010 Proceedings IEEE*, 2010, pp. 1–9.

[177] Z. Li and H. Shen, "Game-Theoretic Analysis of Cooperation Incentive Strategies in Mobile Ad Hoc Networks," *IEEE Transactions on Mobile Computing*, vol. 11, no. 8, pp. 1287–1303, Aug. 2012.

[178] S. Banerjee and A. Misra, "Minimum energy paths for reliable communication in multi-hop wireless networks," in *Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking & computing*, 2002, pp. 146–156.

[179] "IP Camera Adapter." [Online]. Available: http://ip-webcam.appspot.com/. [Accessed: 01-Jan-2017].
[180] "Ustream — The leading HD streaming video platform." [Online]. Available: http://www.ustream.tv/. [Accessed: 01-Jan-2017].
[181] "Veetle - Live Video Stream on Veetle." [Online]. Available: http://veetle.com/. [Accessed: 01-Jan-2017].
[182] "BitTorrent Live | People Powered Internet TV," *BitTorrent Live*. [Online]. Available: https://btlive.tv/. [Accessed: 03-Jan-2017].
[183] L. Kaddar and A. Mehaoua, "ESTREL: Transmission and Reception Energy Saving Model for Wireless Ad Hoc Networks," in *32nd IEEE Conference on Local Computer Networks (LCN 2007)*, Dublin, Ireland, 2007, pp. 784–785.
[184] L. Kaddar, Y. Hadjadj-Aoul, and A. Mehaoua, "EVAN: Energy-Aware SVC Video Streaming over Wireless Ad Hoc Networks," in *2011 IEEE 73rd Vehicular Technology Conference (VTC Spring)*, Budapest, Hungary, 2011, pp. 1–5.
[185] J. Aupet, R. Kassab, and J. Lapayre, "WAVA: a new web service for automatic video data flow adaptation in heterogeneous collaborative environments," *Cooperative Design, Visualization, and Engineering*, pp. 280–288, 2009.
[186] S. Di, Y. Zhao, C. Li, and Y. Guo, "An energy-aware chunk selection mechanism in HTTP video streaming," in *Wireless Communications & Signal Processing (WCSP), 2016 8th International Conference on*, 2016, pp. 1–5.
[187] R. Guruprasad and S. Dey, "Battery Aware Video Delivery Techniques Using Rate Adaptation and Base Station Reconfiguration," *IEEE Transactions on Multimedia*, vol. 17, no. 9, pp. 1630–1645, Sep. 2015.
[188] H. Ahmad, N. Saxena, A. Roy, and P. De, "Extending Video Playback Time With Limited Residual Battery," *IEEE Communications Letters*, vol. 20, no. 8, pp. 1659–1662, Aug. 2016.
[189] E. M. Reingold and L. C. Loschky, "Reduced saliency of peripheral targets in gaze-contingent multi-resolutional displays: blended versus sharp boundary windows," 2002, p. 89.
[190] B. Watson, N. Walker, L. F. Hodges, and A. Worden, "Managing level of detail through peripheral degradation: effects on search performance with a head-mounted display," *ACM Transactions on Computer-Human Interaction*, vol. 4, no. 4, pp. 323–346, Dec. 1997.
[191] L. C. Loschky and G. W. McConkie, "User performance with gaze contingent multiresolutional displays," 2000, pp. 97–103.
[192] B. Ciubotaru, G.-M. Muntean, and G. Ghinea, "Objective Assessment of Region of Interest-Aware Adaptive Multimedia Streaming Quality," *IEEE Trans. on Broadcast.*, vol. 55, no. 2, pp. 202–212, Jun. 2009.
[193] D. Agrafiotis, S. J. C. Davies, N. Canagarajah, and D. R. Bull, "Towards efficient context-specific video coding based on gaze-tracking analysis," *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 3, no. 4, pp. 1–15, Dec. 2007.
[194] J.-H. Lee and C. Yoo, "Scalable roi algorithm for H.264/SVC-based video streaming," *IEEE Transactions on Consumer Electronics*, vol. 57, no. 2, pp. 882–887, May 2011.
[195] J. Lee, S. Lee, and J. Paik, "Digital image stabilization based on statistical selection of feasible regions," *IEEE Transactions on Consumer Electronics*, vol. 55, no. 4, pp. 1748–1755, Nov. 2009.
[196] N. Staelens *et al.*, "Assessing Quality of Experience of IPTV and Video on Demand Services in Real-Life Environments," *IEEE Transactions on Broadcasting*, vol. 56, no. 4, pp. 458–466, Dec. 2010.
[197] Z. Xue, K. K. Loo, J. Cosmas, M. Tun, L. Feng, and P.-Y. Yip, "Error-Resilient Scheme for Wavelet Video Codec Using Automatic ROI Detection and Wyner-Ziv

Coding Over Packet Erasure Channel," *IEEE Transactions on Broadcasting*, vol. 56, no. 4, pp. 481–493, Dec. 2010.

[198] X. Lan, N. Zheng, W. Ma, and Y. Yuan, "Arbitrary ROI-based wavelet video coding," *Neurocomputing*, vol. 74, no. 12–13, pp. 2114–2122, Jun. 2011.

[199] Y. Y. Chung, W. J. Chin, X. Chen, D. Y. Shi, E. Choi, and F. Chen, "Content-based video retrieval system using wavelet transform.," *WSEAS Transactions on Circuits and Systems*, vol. 6, no. 2, pp. 259–265, 2007.

[200] W. Ji, M. Chen, X. Ge, P. Li, and Y. Chen, "ESVD: an integrated energy scalable framework for low-power video decoding systems," *EURASIP Journal on Wireless Communications and Networking*, vol. 2010, p. 5, 2010.

[201] D. Grois, E. Kaminsky, and O. Hadar, "Dynamically adjustable and scalable ROI video coding," in *Broadband Multimedia Systems and Broadcasting (BMSB), 2010 IEEE International Symposium on*, 2010, pp. 1–5.

[202] Y. Wang, Jae-Gon Kim, S.-F. Chang, and Hyung-Myung Kim, "Utility-Based Video Adaptation for Universal Multimedia Access (UMA) and Content-Based Utility Function Prediction for Real-Time Video Transcoding," *IEEE Trans. Multimedia*, vol. 9, no. 2, pp. 213–220, Feb. 2007.

[203] N. Vallina-Rodriguez, P. Hui, J. Crowcroft, and A. Rice, "Exhausting Battery Statistics," *Proc. ACM Mobiheld*, 2010.

[204] N. Vallina-Rodriguez, N. Vallina, and J. Crowcroft, "ErdOS: achieving energy savings in mobile OS," in *Proceedings of the sixth international workshop on MobiArch*, 2011, pp. 37–42.

[205] "JuiceDefender - battery saver - Android Apps on Google Play." [Online]. Available: https://play.google.com/store/apps/details?id=com.latedroid.juicedefender. [Accessed: 02-Jan-2017].

[206] Adobe Digital Index, "Digital Video Benchmark Report Q1 2016," Jun. 2016.

[207] K. Paul and T. K. Kundu, "Android on Mobile Devices: An Energy Perspective," 2010, pp. 2421–2426.

[208] G. P. Perrucci, F. H. P. Fitzek, and J. Widmer, "Survey on Energy Consumption Entities on the Smartphone Platform," in *Vehicular Technology Conference (VTC Spring), 2011 IEEE 73rd*, 2011, pp. 1–6.

[209] Martin Kennedy, Hrishikesh Venkataraman, and Gabriel-Miro Muntean, "Dynamic Stream Control for Energy Efficient Video Streaming," presented at the IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB), Nuremberg, Germany, 2011.

[210] R. Trestian, A.-N. Moldovan, O. Ormond, and G. Muntean, "Energy consumption analysis of video streaming to android mobile devices," in *Network Operations and Management Symposium (NOMS), 2012 IEEE*, 2012, pp. 444–452.

[211] A. Rice and S. Hay, "Decomposing power measurements for mobile devices," in *Pervasive Computing and Communications (PerCom), 2010 IEEE International Conference on*, 2010, pp. 70–78.

[212] R. Balani, "Energy consumption analysis for bluetooth, wifi and cellular networks," *Online]. http://nesl. ee. ucla. edu/fw/documents/reports/2007/PowerAnalysis. pdf*, 2007.

[213] "Free SIP Phone for Windows, Android and iPhone." [Online]. Available: http://www.3cx.com/VOIP/sip-phone/. [Accessed: 02-Jan-2017].

[214] M. Kennedy, H. Venkataraman, and G.-M. Muntean, "Battery and stream-aware adaptive multimedia delivery for wireless devices," in *Local Computer Networks (LCN), 2010 IEEE 35th Conference on*, 2010, pp. 843–846.

[215] M. Kennedy and G.-M. Muntean, "A study on the effect of transmission power adaptation and multi-hop path usage on power consumption and QoS in adaptive

mobile video delivery," presented at the IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB), Beijing, China, 2014, pp. 1–7.
[216] Adobe Systems Incorporated, "media server software tools and downloads | Adobe Flash Media Server." [Online]. Available: http://www.adobe.com/products/flashmediaserver/tool_downloads/. [Accessed: 13-May-2011].
[217] "MSU Video Quality Measurement Tool (PSNR, MSE, VQM, SSIM)." [Online]. Available: http://compression.ru/video/quality_measure/video_measurement_tool.html. [Accessed: 03-Jan-2017].
[218] "JMF 2.1.1 Software Documentation." [Online]. Available: http://www.oracle.com/technetwork/java/javase/documentation-138769.html. [Accessed: 04-Jan-2017].
[219] "VideoLAN - Official page for VLC media player, the Open Source video framework!" [Online]. Available: http://www.videolan.org/vlc/index.html. [Accessed: 04-Jan-2017].
[220] "Java bindings - VideoLAN Wiki." [Online]. Available: https://wiki.videolan.org/index.php?title=Java_bindings&direction=prev&oldid=15452. [Accessed: 04-Jan-2017].
[221] "The DJ project." [Online]. Available: http://djproject.sourceforge.net/ns/. [Accessed: 03-Jan-2017].
[222] "Xuggle." [Online]. Available: http://www.xuggle.com/. [Accessed: 04-Jan-2017].
[223] "FFmpeg." [Online]. Available: https://ffmpeg.org/. [Accessed: 04-Jan-2017].
[224] OMNeT++ Community, "OMNeT++ Community Site," 20-Apr-2010. [Online]. Available: http://www.omnetpp.org/. [Accessed: 04-May-2010].
[225] "MiXiM." [Online]. Available: http://mixim.sourceforge.net/. [Accessed: 04-May-2010].
[226] "ExoPlayer - Home." [Online]. Available: http://google.github.io/ExoPlayer/. [Accessed: 04-Jan-2017].
[227] "Media server for streaming video | Adobe Media Server family." [Online]. Available: http://www.adobe.com/ie/products/adobe-media-server-family.html. [Accessed: 04-Jan-2017].
[228] "Adobe AIR | Deploy applications." [Online]. Available: http://www.adobe.com/ie/products/air.html. [Accessed: 04-Jan-2017].
[229] N. Ponomarenko, S. Krivenko, V. Lukin, and K. Egiazarian, "Visual quality of lossy compressed images," in *CAD Systems in Microelectronics, 2009. CADSM 2009. 10th International Conference-The Experience of Designing and Application of*, 2009, pp. 137–142.
[230] Harald Friis, "A Note on a Simple Transmission Formula," presented at the Proceedings of the I.R.E. and Waves and Electrons, 1946.
[231] S.-B. Lee, G. Muntean, and A. F. Smeaton, "Performance-aware replication of distributed pre-recorded IPTV content," *Broadcasting, IEEE Transactions on*, vol. 55, no. 2, pp. 516–526, 2009.
[232] "Vodafone Smart Black." [Online]. Available: http://www.vodafone.ie/df/prepay/new/index.jsp?handset=SMRTB&manufacturerSelected=ALL. [Accessed: 09-Sep-2011].
[233] "Samsung Europa Black from Three.ie." [Online]. Available: http://three.ie/shop/products/samsung-europa-black.html?_selectedTariff=prepay&_defaultPriceplan=9#. [Accessed: 09-Sep-2011].