CONVINcE

D4.2.2

Description of prototypes of energy saving mechanisms for resource usage and collaborative architectures

Editor: Oguzhan Iren Erol (Vestel)

Authors: Rickard Ljung (Sony Mobile)

Siwar Ben Hadj Said (CEA)

Anders Plymoth (TelHoc)

Björn Landfeldt (Lund University)

EXECUTIVE SUMMARY

With the advancement of the mobile Internet, video delivery is being considered as one of the most energy consuming applications in ICT. Every second, nearly a million minutes of video content is expected to cross the network by 2019. Video delivery forms a big portion of the Internet traffic. One of the most challenging problems of video delivery to mobile devices is the energy consumption and limited battery life. The problem is that accessing multimedia over wireless networks (like Wi-Fi, 3G, and 4G) is power consuming and mobile devices have limited battery life. So energy optimization of Video delivery over wireless networks is an interesting research topic. To reduce the energy consumption of wireless interfaces, various power saving mechanisms are introduced for both Wi-Fi and cellular networks (LTE).

Power consumption of fixed terminals is also increasing every year. The total number of IPTV subscribers is about 81 million and the number of WebTV users dramatically increases every year. Not only preparation and delivery of the IP video content to IPTV subscribers and WebTV users but also transforming IP packets into video on the screen is very crucial which depends on the fixed terminals, i.e., TV sets and Set-Top Boxes (STBs). According to the Natural Resource Defense Council (NRDC), in 2010, STBs in the United States consumed approximately 27 billion kilowatt-hours of electricity, which is equivalent to the annual output of nine average (500 MW) coal-fired power plants. Therefore, power consumption reduction in fixed terminals is very vital. In this document, energy consumption of both set-top box and TV set hardware is considered.

We first start by reviewing the state of the art in energy savings in both fixed and mobile terminals and then proceed to energy measurements. Moreover, the initial design of an energy efficient interface selection mechanism for multi-interface terminals is introduced. The following aspects are considered and discussed:

- The main power consuming entities in a mobile terminal include: wireless modem entities (e.g. Wi-Fi, LTE etc), application entity (e.g. application entity running system operating system, and specific hardware/software units handling specific tasks, e.g. graphics), display, multimedia content creation entities, e.g. video camera
- Power Saving Mode (PSM) is introduced in the IEEE 802.11 standard to reduce the energy consumption of Wi-Fi interfaces by putting devices into sleep mode when they do not have any data to send or to receive.
- A mobile terminal can choose the network interface to use to send/receive data. In some cases, it can use several interfaces at the same time by simultaneously assigning different application sessions to different network interfaces. Based on this, an energy-efficient interface selection mechanism is introduced.
- As the processor is a large constraint for mobile devices, approaches to offload computation from the devices to servers have emerged. The mobile computation offloading systems either aim to save energy of the mobile device or make it possible to accomplish tasks that are not normally possible solely using the mobile device.
- Network and fountain coding randomly combines a set of packets or pieces of data using a code. The major difference between network coding and fountain coding is that, in network coding packets from several different sources are combined, while for fountain coding packets from the same stream or file are typically combined.
- Regarding fixed terminals, SoC BCM7252 is used in our project as set-top box hardware. The BCM7252 implements Dynamic Power Management with four different power states, deep standby, passive standby, active standby, and active, which can reduce the energy consumption.
- To achieve power efficiency, Vestel uses different backlight algorithms and optical design of the backlight unit. Backlight algorithms are considered to both provide energy efficiency and increase contrast perception. Eco backlight and auto backlight algorithms are discussed in this document. Furthermore, new optical design is another solution that significantly changes the power consumption of the backlight unit. New optical design is studied in the project.

Table of Contents

Table of Figures

Table of Tables

1 DOCUMENT HISTORY AND ABBREVIATIONS

1.1 Document history

Version	Date	Description of the modifications
0.1	2016-09-03	Structure of the document (Vestel)
0.2	2017-05-24	All contributions are integrated by Vestel
0.3	2017-06-08	Document reviewed by Lund
1.0	2017-06-16	Document reviewed by Lund

1.2 Abbreviations

RNC	Radio Network Controller
RRC	Radio Resource Control
RLC	Radio Link Control
DCH	Dedicated Channel
FACH	Forward Access Channel
PCH	Paging Channel
CAM	Constant Awake Mode
PSM	Power save mode
AP	Access Point
TCP	Transmission Control Protocol
TIM	Traffic Indication Map
AM	Active Mode
BMI	Beacon Monitoring Intervals
APSM	Adaptive PSM
A-MSDU	aggregate MAC service data unit
A-MPDU	aggregate MAC protocol data unit
CRC	cyclic redundancy checks
FCS	frame check sequence
TID	traffic identifier
BA	block acknowledgement
RNC	Radio Network Controller
RRC	Radio Resource Control
RLC	Radio Link Control
DCH	Dedicated Channel
FACH	Forward Access Channel
PCH	Paging Channel
CAM	Constant Awake Mode

PSM	Power save mode
AP	Access Point
TCP	Transmission Control Protocol
TIM	Traffic Indication Map
AM	Active Mode
BMI	Beacon Monitoring Intervals
APSM	Adaptive PSM
A-MSDU	aggregate MAC service data unit
A-MPDU	aggregate MAC protocol data unit
CRC	cyclic redundancy checks
FCS	frame check sequence
TID	traffic identifier
BA	block acknowledgement

1 INTRODUCTION

CONVINcE addresses the challenge of reducing the power consumption in IP-based video networks with an end-to-end approach, from the Head End, where contents are encoded and streamed to the terminals where they are consumed, embracing the CDN and the core and access networks.

The general objective of WP4 is to explore how to minimize the energy consumption of video delivery networks on the terminal side. The WP does not consider the terminals only as video consumption units, but also looks at elements of video created, encoded and delivered from the terminal itself, as well as terminal elements of network connectivity.

2 ENERGY SAVING MECHANISMS

2.1 Fixed Terminals

The total number of the IPTV subscribers is about 81 million in and the number of the WebTV users dramatically increase every year. Not only preparation and delivery of the IP video content to the IPTV subscribers and WebTV users but also transforming IP packets into shows, movies, and sports on the screen is crucial and depends on the fixed terminals, namely TV sets and Set-Top Boxes (STBs). According to Natural Resource Defense Council (NRDC), in 2010, STBs in the United States consumed approximately 27 billion kilowatt-hours of electricity, which is equivalent to the annual output of nine average (500 MW) coal-fired power plants. Therefore, power consumption reduction in fixed terminals is vital.

Analyses show that in 2010 STBs used to consume almost as much power when not in use as they do when in use as shown in Figure 1. However, leading European service providers have begun to solve this problem in their newest boxes. The most efficient IPTV boxes, draw approximately 18 watts when operating (On mode) and 12 watts in light sleep state. European IPTV HD-DVRs demonstrated impressively low On mode power levels of less than 10 watts.

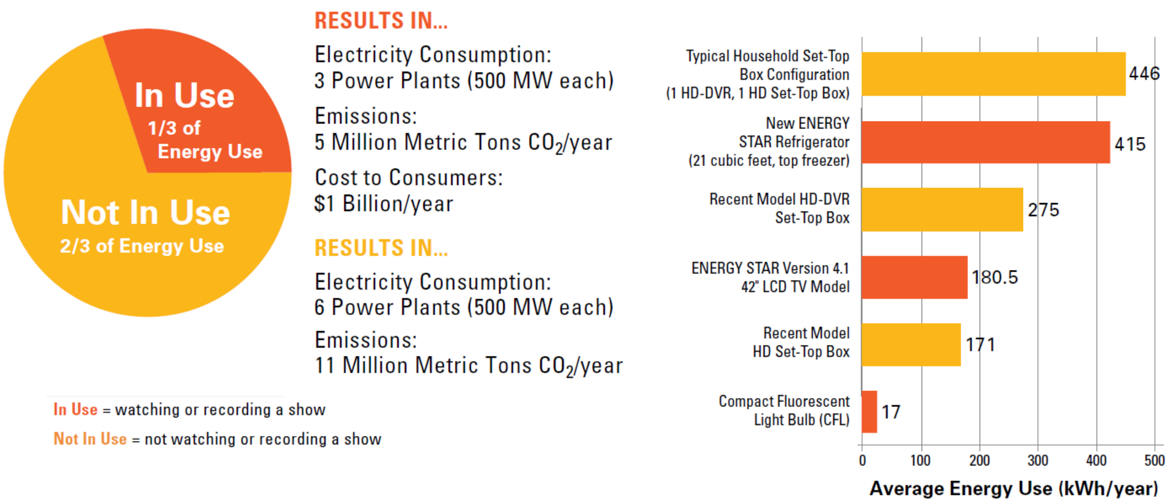


Figure 1:STB energy consumption statistic

The total number of TV sets in the world reached up to 1,5 and the number of the TV sets with Internet connectivity, namely connected TVs, dramatically increases parallel to the growth in the IP network infrastructure. The number of connected TV connections to the Internet will hit 596 million by 2017, up from 105 million at end-2010 and the 212 million expected at end-2012. The Connected TV Forecasts report states that this translates to 20% of global TV sets by 2016, up CONVINcE confidential CONVINcE D4.2.1 Initial design of energy-efficient terminals V1.1 Page 17/35 from only 6% at end-2010

Since the total number of connected TV sets is enormous and the amount of the power consumed is high for each TV set (in the order of hundred watt), the power consumption efficiency becomes a critical issue. TV manufacturers have been investing a lot of their resources to produce energy efficient TV sets using energy efficient display technologies such as LED backlight LCD and OLED.

2.1.1 Green software/hardware

There are two sections in terms of power consumption in STB and TV hardware which are power supply efficiency and Silicon efficiency. As stated in deliverable D4.1.1, the SoC BCM7252 is used in this project.

The overall circuit is supplied by different voltage levels and the power transformation part of the STB circuit conducts this operation. This part is idle and provides operation power for STB. The

BCM7252 implements Dynamic Power Management with the understanding that devices based on the BCM7252 face increasing power environment requirements including (among other sources) those from Energy Star®, the European Commission Institute for Environment and Sustainability (IES), and the National Resources Defense Council (NRDC). Consistent with these specifications, the BCM7252 implements four power modes:

• Deep Standby: This is the lowest power mode, fully powered down, where ONOFF and DRAM devices are powered down. A full reboot is needed to exit from this mode.
• Passive Standby: This is a low-power mode, where the product is connected to power but has no active functionality. The device may be "woken up" by external stimuli (or timers). Passive Standby supports Suspend to DRAM mode, where DRAM devices remain powered during deep standby and their content is recovered at boot time ("warm boot").
• Active Standby: In this mode, the product is connected to power but the functionality is limited, which may affect sending and receiving data from the front end and/or network interfaces. The device may be woken up by external stimuli (or timers) and/or in response to certain network data.
• Active: The device is fully functional in this mode. The BCM7250 Dynamic Power Management block controls power management transitions and is specifically designed so that power management and/or power management transitions do not introduce security vulnerabilities. The BCM7250 supports automatic voltage scaling, where on-chip process sensors are used to automatically scale down the supply rails.

2.1.2 Energy savings for displays

The most energy consuming part of a TV is its panel. 60-80% of all energy consumed by the TV is used by panel backlight. For this reason, backlight algorithms and optical design of the backlight unit is very important for achieving power efficiency.

2.1.2.1 Enhanced backlight algorithm

In the scope of the project, Vestel uses two different backlight algorithms both to provide energy efficiency and to increase contrast perception.

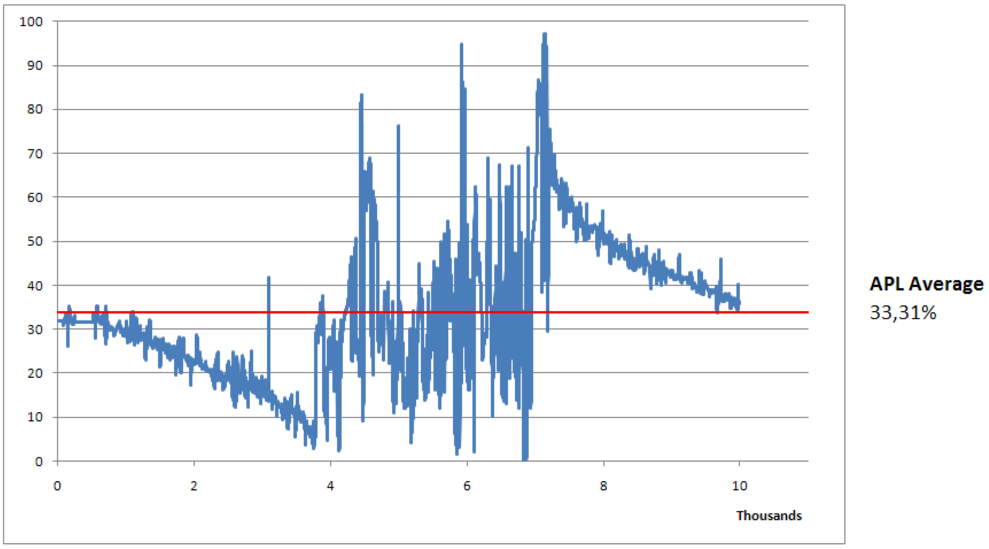


Figure 2. Average APL (Average Picture Level)

Research show that most TV programs and movies usually have around %33 APL (Average Picture Level which represents the average brightness of content). This means they are not always bright at all. So, applying full PWM, thus consuming maximum power every time is unnecessary. Besides, on LCD TV technology, because of the backlight leakages, applying max backlight at dark pictures leads to a low contrast feeling. Thus adaptively changing backlight power according to picture's APL is sensible both for good contrast and energy efficiency.

The standard energy consumption measurement video (which is around 10 minutes) has an APL change as shown in the graph above. This video actually simulates a standard video. So we designed our algorithms according to this data:

- Eco Backlight Algorithm: This algorithm provides power consumption reduction and also keeps the luminance of the panel most suitable for the input video. Basicly, there are high and low limits of PWM and backlight oscillates between these limits according to video content's APL, black pixel percentage and white pixel percentage. So, dynamic algorithm is used instead of constant PWM and it reduces the power consumption. Vestel TVs use this algorithm at the opening condition of the TV.

- Auto Backlight Algorithm: This algorithm is suitable for customers who want more vivid picture. This one also provides energy efficiency and increases contrast perception. The difference is this algorithm becomes active when APL of the content drops below 20%. It calculates new PWM using the black pixel percentage, APL percentage and current PWM. When there is no video, PWM duty cycle is set to 0% and the TV consumes very little power. The basic working mechanism graph can be seen below. Vestel TVs use this algorithm at Dynamic (vivid) picture mode.

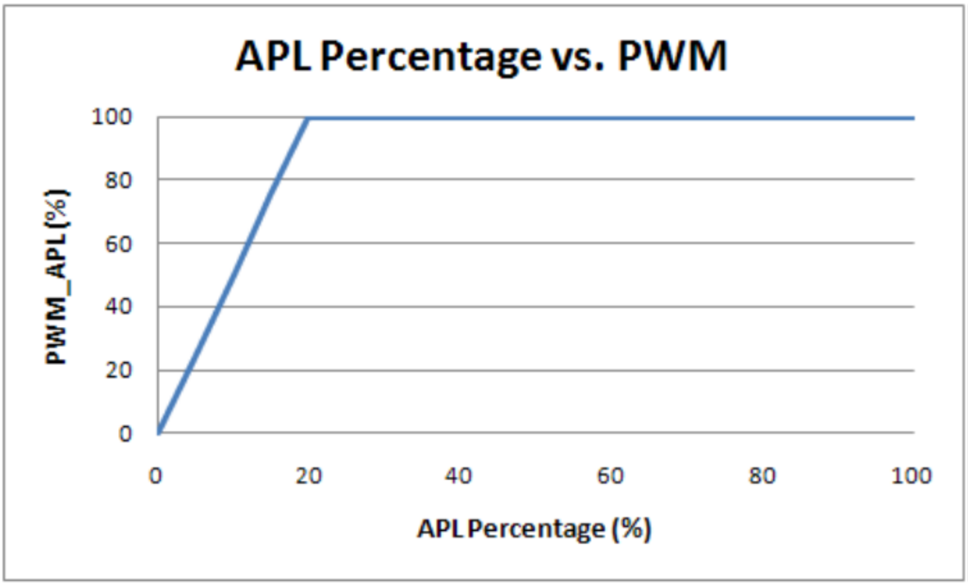


Figure 3. PWM vs APL Percentage graph

2.1.2.2 Optical design

A new optical design significantly changes the power consumption of the backlight unit. Standard luminance levels of TVs on the market is 350 cd/m^2 (nit). A new optical design is studied in this project. In the scope of these new design concepts, terminal luminance levels is re-considered and new luminance varieties such as 400nit, 450nit and 700nit defined regarding consumer needs. Normally a higher luminance level causes inefficiency in power. However, with the advantages of design and production capabilities, optical structure of the Backlight Units improved by designing new light guide plate and LED bar and studying optical film structures and remarkable energy efficiency increase (by 60%-85%, depending on terminal size) is achieved. Below, Table 1 shows Energy levels of non-improved and improved terminals. New terminals, including high bright versions, have advantages in power consumption compared to state of art terminals.

Size	Luminance Target	#LEDs	Standart BLU Power (W)	Improved BLU Power (W)	Power Ratio
43	350nit	116	56	36	64.29%
	400nit			38	
	450nit			34	
49	350nit	136	68	40	58.82%
	450nit			42	
55	350nit	148	68	59	86.76%
	400nit			59	
65	350nit	168	100	83	83.00%
	400nit			89	
	450nit			100	
	700nit			122	

Table 1 Energy levels of non-improved and improved terminals

2.2 Mobile Terminals

2.2.1 Collaborative architectures

2.2.1.1 Energy savings by network and terminal information sharing

2.2.1.1.1 Background architecture

Sony has developed a method for energy savings, using collaborative architectures when conducting video streaming between a mobile terminal and a content server using adaptive streaming such as e.g. MPEG DASH or HLS. We assume a mobile terminal which uses adaptive streaming as a video client to adjust video content quality level when accessing a live or on-demand video content delivery service on the Internet, where the transport network includes a mobile communication network. This concept was described in the CONVINcE deliverable D4.2.1, and a high level illustration of the involved communication link is shown in the Figure 4 below.

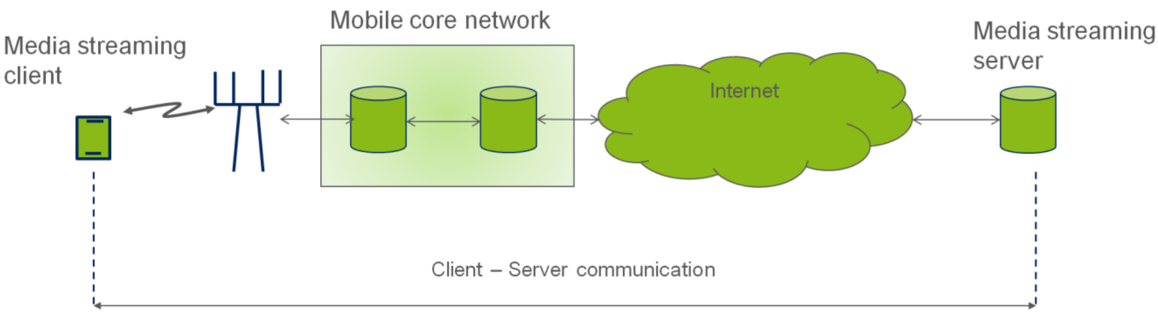


Figure 4 Video streaming scenario via a mobile radio access network

The media streaming client in the mobile terminal is responsible of requesting video segments from the media streaming server. The requests should ideally be made so that the local media content buffer in the mobile terminal always have video content to feed the video player in the terminal. Also, ideally the video content should have as good video quality level as possible. However, if requesting a high video quality level, this will put larger requirements on the mobile network delivery capability.

In a scenario of adaptive video streaming (e.g. MPEG DASH or HLS) over a mobile communication network it is clear that one important aspect of client selection of video quality from the media streaming server is the adaptation to the varying performance of the mobile network throughput capabilities.

2.2.1.1.2 Prototype setup

Within task 4.2 in CONVINcE project, Sony has developed the solution of network assistance where a network assistance function residing within the mobile network may indicate a recommendation of the video quality level to be used for coming video segment(s) to be delivered from the Internet server. The media streaming client within the mobile terminal may therefore take this information into account within its media buffer filling strategy.

In order to save energy within the terminal Sony has developed a prototype client. The prototype client consists of a Sony Xperia mobile phone with a tailored video streaming client software defined to modify its buffer filling strategy. The modifications include targeting to create inactivity periods in-between buffer filling occasions (media segment requests towards the media server), while still maintaining the same media playback quality as without this modification. The inactivity periods results in cellular modem inactivity, which enables the modem to enter low-power inactive states in-between buffer refill occasions. The result is a lower terminal energy consumption.

Without network assistance (NA) this modification could not be done providing the same video playback quality. The reason is that since same video quality shall be used the longer inactivity periods are created by lowering the minimum buffer level in the mobile terminal. Without network assistance such lowered buffer level would result in higher risk for buffer under-run. However, with the network assistance function, the terminal client will have better knowledge about mobile network data rate variations, and can therefore maintain same video playback quality with a lower minimum buffer level.

The principle software solution is illustrated in Figure 5 below.

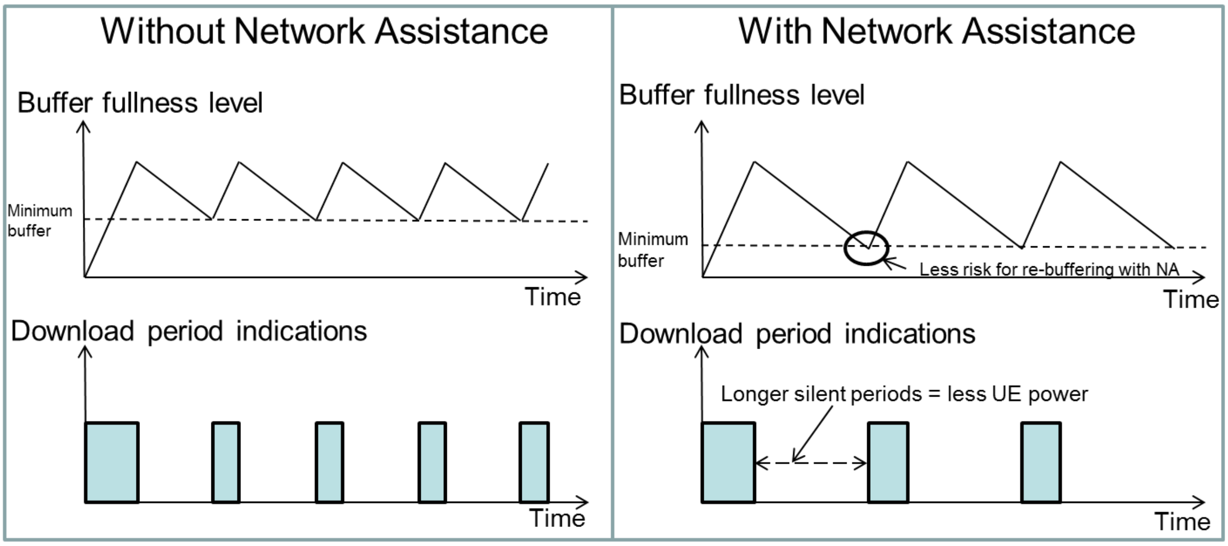


Figure 5 Terminal buffer filling schemes for network assistance prototype test

2.2.1.1.3 Prototype test results

The prototype Sony Xperia phone has been tested by Sony Mobile, running in a lab environment where the mobile network data rate can be controlled.
The mobile phone energy consumption was continuously measured during the tests, using the Sony Mobile energy consumption measurement tool developed within WP5 of CONVINcE project. Results from the measurements are shown in Figure 6 below.

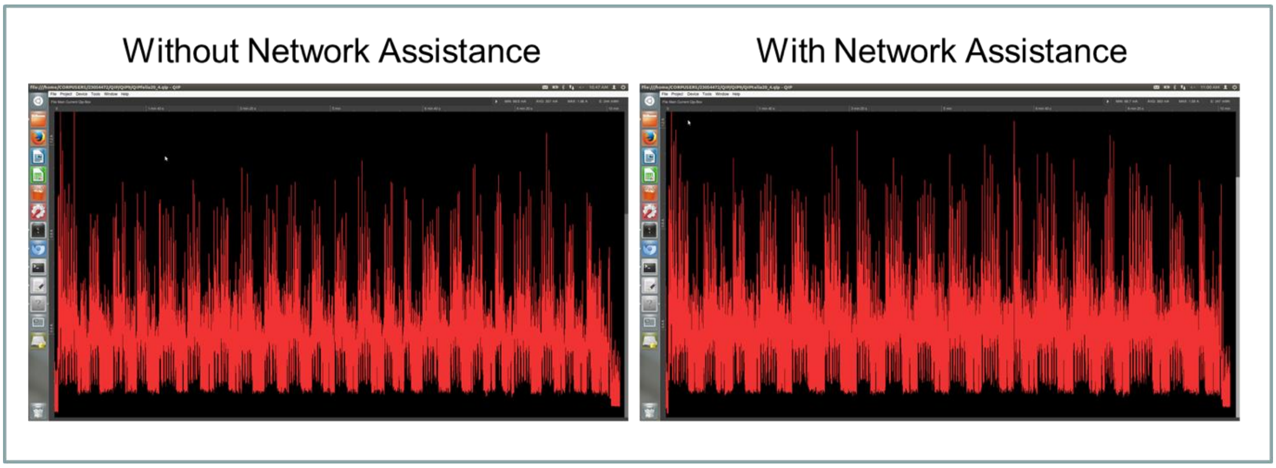


Figure 6 .Terminal continuous energy consumption during network assistance prototype tests

Figure 6 shows the variations in energy consumption during a relatively long period, and no details on the absolute values are given in these high level result figures. However, as can be seen in the figures there are more periods of high and low energy consumption without network assistance. Hence network assistance functions allows the terminal to stay longer periods in inactivity (the periods of low energy consumption), and correspondingly each period of video download are also longer.

The results, as measured by the Sony measurement tool developed in WP5 of CONVINcE show that the silent periods increased by 10% using the network assistance function.
The total energy consumption reduction on the complete Sony Mobile Xperia Smartphone during video streaming and playback was measured to 1.5% (a reduction from 1.35W down to 1.33W). The video playback quality was the same for both scenarios.

The results indicate that the network assistance functionality is working, and it may lead to reduced terminal energy consumption. However still For a Smartphone the screen and application processor are still dominating the power drain, limiting the total energy consumption to a relatively moderate level.

2.2.2 Resource Usage

2.2.2.1 Energy-efficient Network interface selection for the Terminal (CEA)

2.2.2.1.1 <u>Overall description:</u>

CEA has designed and developed an energy-efficient interface selection mechanism for multi-interface terminals. The principle of this mechanism is summarized as follows. When a terminal is connected to the network through multiple interfaces (e.g. Ethernet and WiFi), the proposed mechanism selects the best egress network interface through which the end-to-end path presents the least energy consumption.

We advocate that, while selecting the network interface to communicate, the terminal should not only consider the energy consumption related to the network interface, but also the estimated energy consumption of the end-to-end path (i.e. from the sender to the destination). The example depicted in Figure 7 explains why we need to consider the energy consumption of the whole path while selecting the egress interface. In the example, we have two terminals (Terminal 1 and 2) and two routers (Router 1 and 2). Terminal 1 is connected to the network via its interfaces Interf-1 and Interf-2 which are connected to Router 1 and Router 2, respectively. Terminal 2 is reachable through Router 2. The number over each link represents the corresponding energy cost. Let us assume that the Terminal 1 will send traffic to Terminal 2. As we can see, Terminal 1 is connected to two different routers (router 1 and router 2) through two heterogeneous interfaces (Interf-1 and Interf-2). The link energy cost in Interf-2 (i.e. 6) is higher than in Interf-1 (i.e. 4). If the interface selection is only based on the link energy cost, Terminal 1 will select Interf-1 to send the data traffic. To reach Terminal 2, the Router 1 needs to relay the data traffic to Router 2. Therefore, the end-to-end path will have an energy cost of 12. However, if Terminal 1 uses Interf-2, the end-to-end path will only have an energy cost of 9.

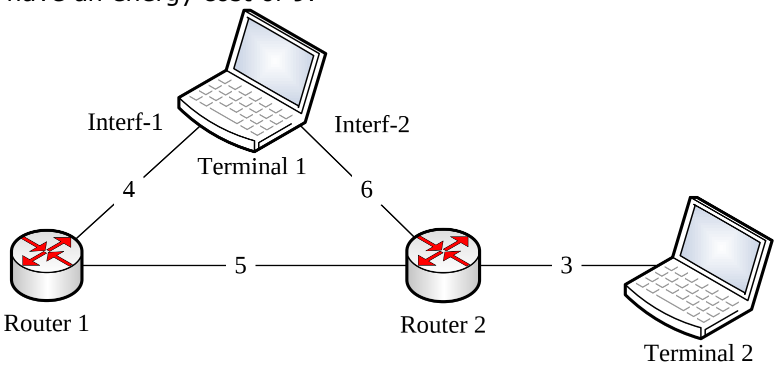


Figure 7: Network topology example

In order to enable the terminal to select the network interface that leads to the least power consumption, each reachable access router should inform it about the power consumption related to the path from this access router to the destination. A simple way to implement this mechanism consists in extending the Neighbor Discovery Protocol (NDP) mechanism that is natively implemented in all terminals and access routers [1].
The NDP enables terminals to discover, determine the reachability of their neighboring nodes, and perform various configurations (e.g. obtaining and configuring IPv6 address). According to NDP, routers periodically advertise their presence as well as other configuration parameters (e.g. the link MTU, the IPv6 prefixes, the presence of DHCPv6 server or proxy, etc.) by broadcasting Router Advertisement (RA) messages. The RA message can be sent upon receiving a Router Solicitation (RS) message from the host. The information provided by the RA messages are used by the terminal to perform interface auto-configuration. The RS and RA messages are only valid on-link and are never forwarded.
In order to support the energy-efficient interface selection in the terminal, we extended NDP RS and RA messages with the two following new options:

- **Energy Link Cost Option (ELCO**): It is used by the access router to advertise the link energy cost to the host. That is, the ELCO options are added in all RA messages broadcasted by the access router. Upon receiving the ELCO option, the host extracts the link energy cost and assigns it to the network interface on which it received the RA. The operation is executed on each of the active network interfaces of the host. Once all the network interfaces are configured with their corresponding energy cost, the host selects the interface with the lowest energy cost to send the data traffic.
- **Energy Path Cost Option (EPCO):** It is used by the terminal to request the end-to-end path energy cost to a specific destination. It is added to the RS message. Therefore, the terminal sends the RS message on each of its network interfaces. Upon receiving the EPCO option, the router extracts the destination address or prefix and determines from its routing table the most energy-efficient path to reach this destination. Then, the corresponding energy path cost is filled in the EPCO field in the RA message and sent back to the terminal. Upon receiving the responses to all its RS messages, the terminal calculates the end-to-end path energy cost related to each interface. The host selects the interface that ensures the path the least energy consuming to send the data traffic.

Let us consider the previous example (in Figure 7**Error! Reference source not found.**) to illustrate how the interface selection is performed according to our solution. Figure 8 shows the messages exchange diagram according to our solution. In this procedure we have 4 steps:

- Step1: Terminal 1 receives two RA messages coming from Router 1 and Router 2, respectively. Each one of these messages includes ELCO option that informs Terminal 1 the link energy cost related to the corresponding interface (i.e. in the example, the cost 4 for Interf-1 and cost 6 for Interf-2).

- Step 2: Terminal 1 decides to exchange traffic with Terminal 2. Instead of directly sending traffic through its default interface, Terminal 1 uses the EPCO option in RS messages to ask Router 1 and Router 2 the energy cost of their best path to reach Terminal 2 according to their routing tables.

- Step 3: Each router uses the EPCO option in the RA message to reply to Terminal 1 request. Router 1 announces to Terminal 1 that the path to reach Terminal 2 from Router 1 has an energy cost of 8. Router 2 announces that the path from Router 2 has an energy cost of 3. At this stage, Terminal 1 computes the end-to-end path energy cost for the two cases, compares between them and selects the interface that ensures the most energy-efficient path. The end-to-end path through Interf-1 has an energy cost of 12 and through Interf-2 has an energy cost of 9. In that case, Terminal 1 selects the interface Interf-2 as the egress interface and Router 2 as the next hop to send data traffic to Terminal 2.

- Step 4: Terminal 1 sends data traffic to Terminal 2 through the network interface Interf-2.

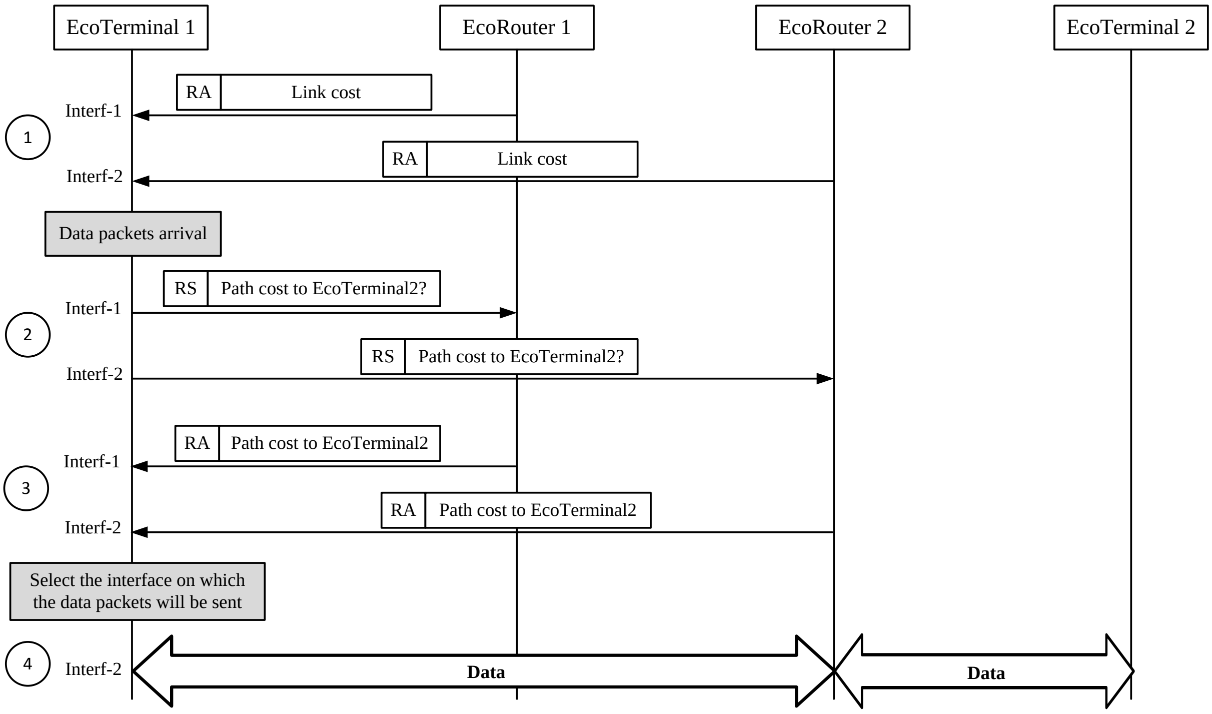


Figure 8: Energy-efficient network interface selection flowchart

2.2.2.1.2 Testbed setup

The testbed is composed of two EcoTerminals and three EcoRouters as depicted in Figure 99. The EcoTerminals represent the host or user terminal (e.g. smartphone, tablet, etc.) or the video server. They have one or more network interfaces and that have IP forwarding capability disabled. Part of the proposed energy efficient network interface selection mechanism is implemented in the EcoTerminal that represents the user terminal. An EcoRouter represents the devices with routing functionality and where we implemented the proposed extensions that are needed for an energy efficient network interface selection. The energy-efficient routing that was proposed in WP3 is implemented in these routers in order to enable routers to calculate the least power consumption path towards each destination. In the testbed, the EcoRouters are connected to each other via heterogeneous link technologies such as WiFi, Bluetooth, and Ethernet.

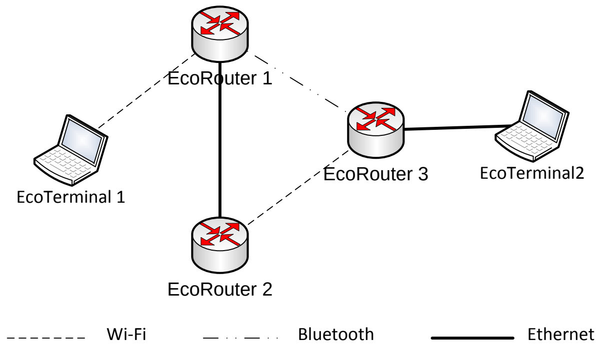


Figure 9: Testbed setting

For Ecorouters, we used the IONICS Stratus plug computer platform [2]. This device has MARVELL Kirkwood 1.2 GHz as a processor, 512 MB of flash memory, 512 MB of RAM and a Linux kernel 2.6 with Debian (Squeeze) distribution. The implementation of the distributed routing approach is based on QUAGGA [3] and RADVD [4] softwares. QUAGGA is a routing software suite that provides an implementation of several routing protocols including OSPFv3 (i.e. the IPv6 version of OSPF). RADVD (i.e. Router ADVertisement Daemon) is an implementation of the Router Solicitation (RS) / Router Advertisement (RA) messages used by Network Discovery Protocol (NDP). Both softwares are available on Linux repositories. We modified them according to our energy-efficient routing protocol.

We used conventional laptops as EcoTerminals. They run a Linux kernel 3.2 with Ubuntu 12.04 LTS distribution. EcoTerminal1 represents the user's laptop whereas EcoTerminal2 represents a server in the Internet.

The main issue with empirical approach in measuring power consumption related to a given network interface is that the obtained results are inevitably hardware-specific. To cope with this problem, we propose to rather refer to theoretical values. The European Commission EC recently published a Code of Conduct (CoC) about the energy consumption of broadband equipment [5]. This document defines the recommended power consumption that network equipment should reach for the year 2014. We argue that the use of such theoretical values is a suitable alternative to empirical measurements. Therefore, the computation of the energy consumption of each link will be based on values provided by the CoC and are summarized in Table 2.

Link type	***Power idle (W)***	***Power on-state (W) (tranmit/receive)***
Ethernet	0.2	0.6
WiFi	0.7	1.5
Bluetooth	0.1	0.3

Table 2 Power consumption of network interfaces according to the Code of Conduct

In this testbed, the energy-efficient routing protocol that was developed in WP3 and described in deliverables D3.2.1 and D3.2.2 is implemented and run in each router. This will enable each router to compute the least power consumption path towards the each destination. Moreover, we extended the NDP software in EcoRouter2 and EcoRouter1 with ELCO and EPCO options. These options are inserted in the RA messages and used to announce to EcoTerminal1 the cost of the path from itself to the destination in terms of power consumption. In addition, we extended the NDP software in EcoTerminal1 in order to make it understand the ELCO and EPCO options announced by the routers. The RS message is extended with a field where the EcoTerminal1 asks the routers about the path cost towards the destination. We implemented in EcoTerminal1 a mechanism that compares between the received RA messages in terms of path power consumption and selects the router that offers the least power consuming path.

2.2.2.1.3 Mechanism validation results

Initially, the EcoTerminal1 is connected to the network via its WiFi interface only (see Figure 10).

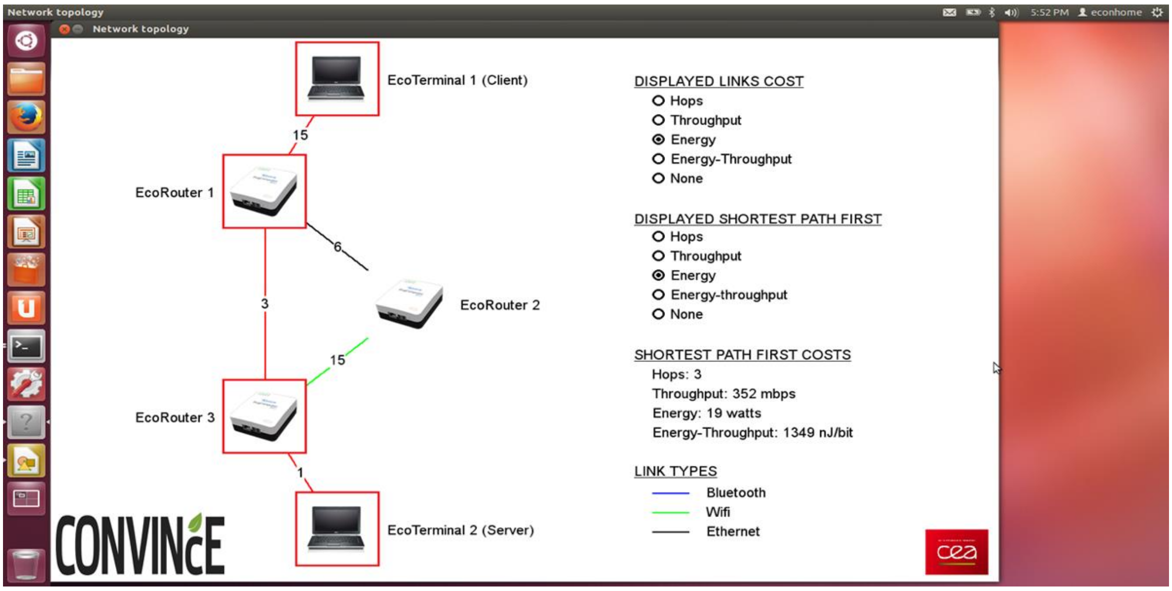


Figure 10: Selected data path in testbed

We activate the Ethernet interface in EcoTerminal1 and attach it to EcoRouter2. The EcoTerminal1 is now connected to the network by both of its Wi-Fi and Ethernet interfaces (see Figure 11). As detailed in , the selection of a default interface on a host depends on the operating system policy. For instance, on Ubuntu Operating System the first enabled network interface is selected as the default one and remains unchanged until it is disabled. That is, in this experimentation and without the use of our energy-efficient routing protocol, the activation of the Ethernet interface on EcoTerminal1 does not modify its default interface: the Wi-Fi interface remains the default one. Therefore, the data path between EcoTerminal1 and EcoTterminal2 remain the same ($EcoTerminal1 \leftrightarrow EcoRouter1 \leftrightarrow EcoRouter3 \leftrightarrow EcoTerminal2$)eading to the same energy consumption results.

However, as EcoTerminal1 runs the edge part of the routing protocol, it is aware of the fact that sending data to EcoTerminal2 via its Ethernet interface would lead to a better end-to-end energy path cost than using its Wi-Fi interface. Indeed, the data would follow the data path $EcoTerminal1 \leftrightarrow EcoRouter2 \leftrightarrow EcoRouter1 \leftrightarrow EcoRouter3 \leftrightarrow EcoTerminal2$

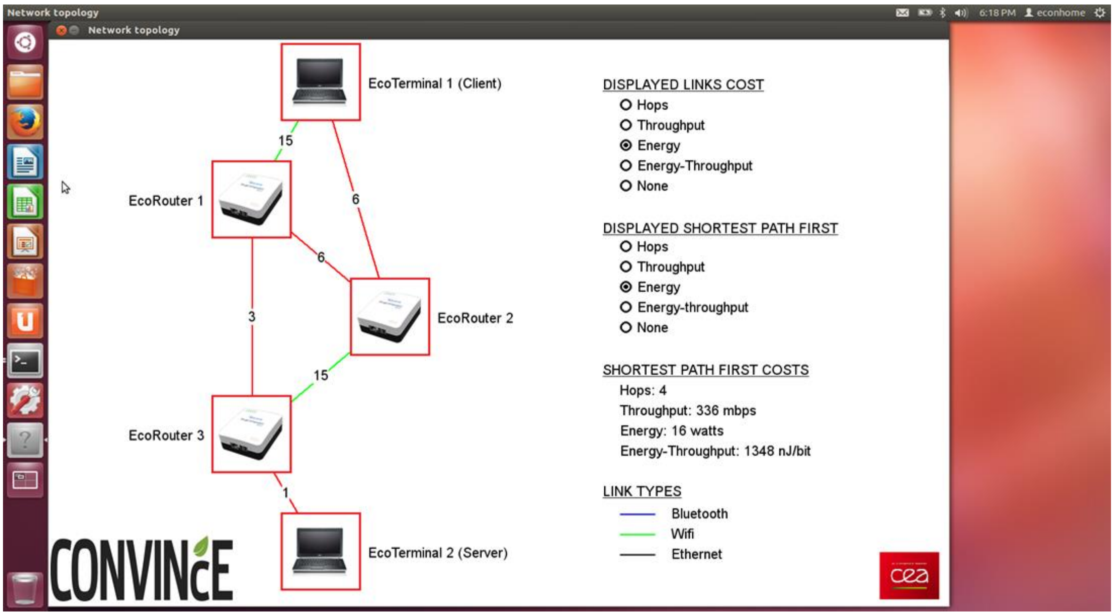


Figure 11: Selected data path after activating Ethernet interface in EcoTerminal1

Figure 12 and Figure 13 shows the energy consumed in each of network device before and after Ethernet interface activation in EcoTerminal1. In Figure 12, we note that the energy consumption of EcoTerminal1 has dropped to the same level as EcoTerminal2 as it uses its Ethernet interface instead of Wi-Fi interface. We also notice that EcoRouter1 hardly consumes energy when we compare between Figure 12Figure 12 and Figure 13. Indeed, it forwards traffic between EcoRouter2 and EcoRouter3 using Ethernet and Bluetooth interfaces. The WiFi interface in the EcoRouter1 is no more used leading to a reduced power consumption.

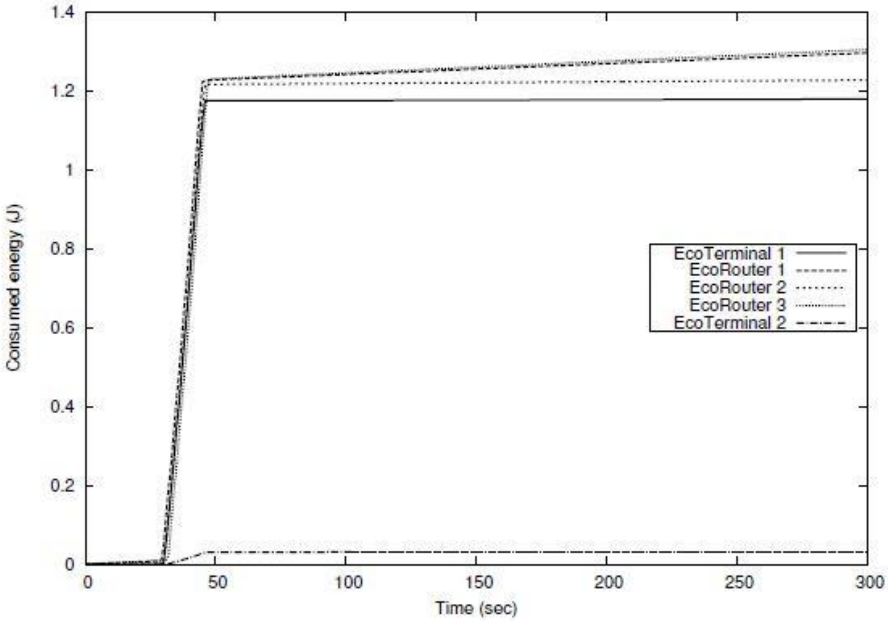


Figure 12: Power consumption in each equipment initially

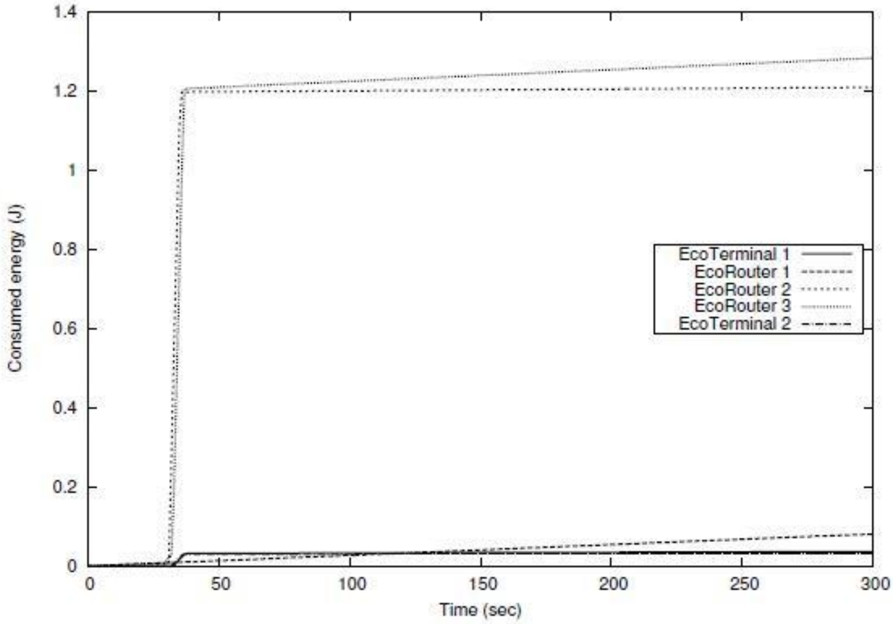


Figure 13: Power consumption in each equipment after activating the Ethernet interface in EcoTerminal1

2.2.3 Fountain and network coding for mobile terminals

2.2.3.1 Background

An important source of energy loss in wireless networks comes from retransmissions of lost packets. Many of the transmission protocols in use today were designed mainly to be used in wired IP networks, and therefore performs bad in wireless scenarios. An area that currently experiences rapid growth is video streaming, and Cisco reports that video will amount to about 80% of all Internet traffic by 2019 [6]. Cisco further predicts that 67% of all IP traffic will be consumed by WiFi connected devices.

While it is generally well know that WiFi connections experiences packet loss, it is less known that these losses are typically much higher than most people realize. For unicast packets, WiFi employ up to four retransmissions and perform dynamic link rate adaptation to better cope with bad channel and signal propagation conditions. One study [7] found that during an important computer science conference as much as 28% of all transmissions failed. In residential areas, many wireless networks and devices coexist in a small frequency spectrum and experiences sustained noise and interference. However, retransmissions are very efficient and even with a loss rate as high as 50%, the resulting packet loss rate is only 6.25%.

Depending on the type of service, video streams can either be transported using TCP or UDP, but often rely on use the UDP protocol in order to reduce latency and for its support of multicast and broadcast streaming services. However, UDP multicast and broadcast packet are not retransmitted, and are therefore much more vulnerable to adverse wireless channel conditions.

A topic that is currently receiving a lot interest in the wireless research community is fountain codes [8], because their properties are particularly efficient in broadcast and multicast scenarios and for improving retransmission performance. Recent studies [9] also indicate that they may also save energy.

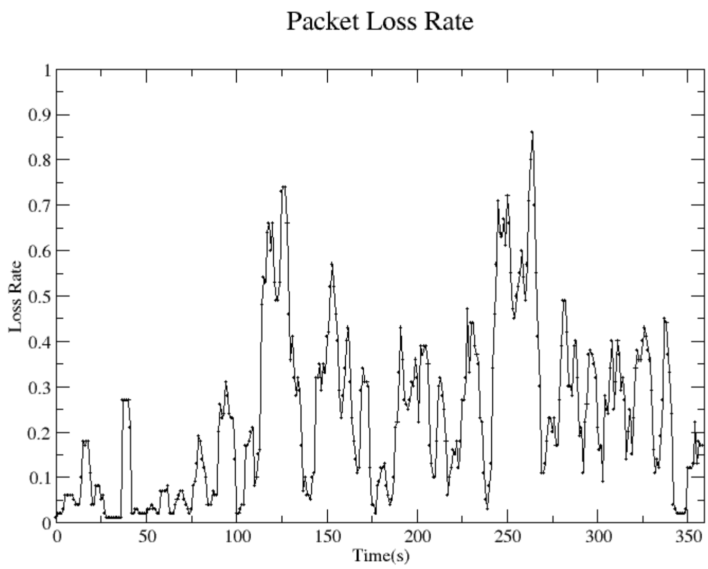


Figure 14. Example of packet loss rate for a WiFi connection over time.

2.2.3.2 Fountain Coding

Fountain codes have been proposed recently for video streaming [10] [11]. Fountain codes are rateless erasure codes in the sense that the encoder can create as many encoded symbols This is an advantage for wireless channels in which the channel conditions vary frequently or are unknown. Moreover, Fountain codes has low complexity both on the encoder and decoder sides compared with other Forward Error Correction (FEC) coding algorithms such as Reed-Solomon codes.

While much work has been performed on the study of fountain codes, including on how it can be used for video streaming [10], and improve the performance of video codecs [12], very little has been done in terms of assessing the impact it has on energy and power consumption. Our presented method also make its deployment much more straight forward, by instead of embedding the coding in the codec or network protocols, it can easily be dropped into applications. It can also be easily deployed at video servers, or at wireless access points.

In network and fountain coding, intermediate nodes may send out packets that are linear combinations of a set of other packets. There are two main benefits of this approach: potential throughput improvements and a higher degree of robustness. Robustness translates into loss resilience and facilitates the design of simple distributed algorithms that improves performance, even if decisions are based only on partial information.
The main difference between network and fountain coding, is that in network coding any packets may be combined, while in fountain coding only packet from a particular source or stream are combined. In network coding [13], a router or a set if routers may identify that multiple paths are available through the network, and that by combining some packets the number of transmitted packets in the network can be reduced.

Fountain codes typically operate on a set of data such as a file, or a piece of a file, and randomly combines these pieces so that the order in which they are received becomes unimportant as long as the number matches at least the number of pieces in the source data. Due to its similarity to erasure coding [14], fountain codes are sometimes also referred to as rateless erasure codes. As a Forward Error Correcting (FEC) mechanism used for Automatic Repeat Request (ARQ) operations, it means that a sender and receiver do not have to consider the correct arrival of individual packets, which greatly simplifies retransmission operation and the needed amount of signaling.
For multicast or multiple receiver operations, especially in wireless scenarios such as in WiFi networks, it has even higher benefits. If individual receivers independently loose different packets in a stream, each of those individual packets needs to be retransmitted. In the case of fountain codes, only a single packets would be needed for all of the receivers, thus greatly reducing the number of needed transmissions and thereby the energy consumed.

In Convince, we have considered two types of fountain coding, LT codes [15] and RaptorQ [16] codes.
Luby Transform (LT) codes are the first class of efficient practical Fountain codes. Potentially, a LT code can generate an unlimited amount of encoded data from the source, where

the source data can be efficiently and completely recovered from reception of any combination of encoded data essentially equal in size to the source data.
The LT (Luby Transform) encoder produces packets from a block of source data as follows: Randomly choose the degree, d from a degree distribution, which depends of the size, K of source data. Here K represent the number of packets needed to represent the source data. The encoder then uniformly at random, chooses d out of these K packets and bitwise modulo 2 combines these packets to a new packet of equal size. In [15] the chosen distribution is a Soliton distribution that enables fast encoding and decoding operations by creating a mix of low and high degree packets. The decoder then reverses this operation by considering the operation as a linear equation system and solves this through Gaussian elimination.
However, LT codes do not have linear decoding properties, and in order to improve upon this, Raptor Codes have been proposed. Raptor codes uses a compound coding structure, which usually includes a high-rate outer LDPC code and an inner LT code, which is able to nearly optimally minimize the needed number of packets for successful decoding. The difference is that in LT codes there is always a slight chance that a newly received packet is linear dependent in the de- coder equation matrix, and thus provides no new information. The outer code greatly reduces this probability and thereby increases the effectiveness of the code.
A superior form of Raptor codes have been proposed, i.e., RaptorQ codes. RaptorQ is more efficient than Raptor coding in terms of flexibility and efficiency. It uses an enhanced two-step pre-coder and a superior LT encoding algorithm. Moreover, it supports a larger range of the size of source symbols and encoding symbols and can deliver huge chunks of data at a time.
RaptorQ is currently the most efficient known fountain code, although it is covered by heavy IPR protection.

2.2.3.3 Energy savings for video services using fountain coding.

Video services delivered using UDP as the transport protocol are vulnerable to packet loss as no explicit acknowledgments or retransmissions are performed. While modern video codecs can accept some packet loss, this comes at the price of lower video quality. In [17] the authors studied how the quality of the video is impacted by packet loss, and found that the quality of the video degrades quickly as the packet loss ratio increases. By 10% it is so bad the video is almost unwatchable.

While it is reasonably well known that commercial and residential WiFi networks periodically experiences packet loss, it is less know how severe this loss actually is. This is because the WiFi protocol performs link adaptation and that this in combination with retransmissions is very efficient. The default is to perform 4 retransmissions, which means that even with a loss rate as high as 50%, the resulting perceived packet loss rate is only 6.25%. However, retransmissions are only performed for unicast packets, not for multicast or broadcast transmissions.

Figure 15 shows a 5 min measurement of the packet loss rate performed in a residential area in central Stockholm. The capacity fluctuates heavily due to variations in noise and interference, and the average packet loss was 23% with peaks up to 80% loss. This would have severe impact on video streams over UDP multicast or broadcast streams.

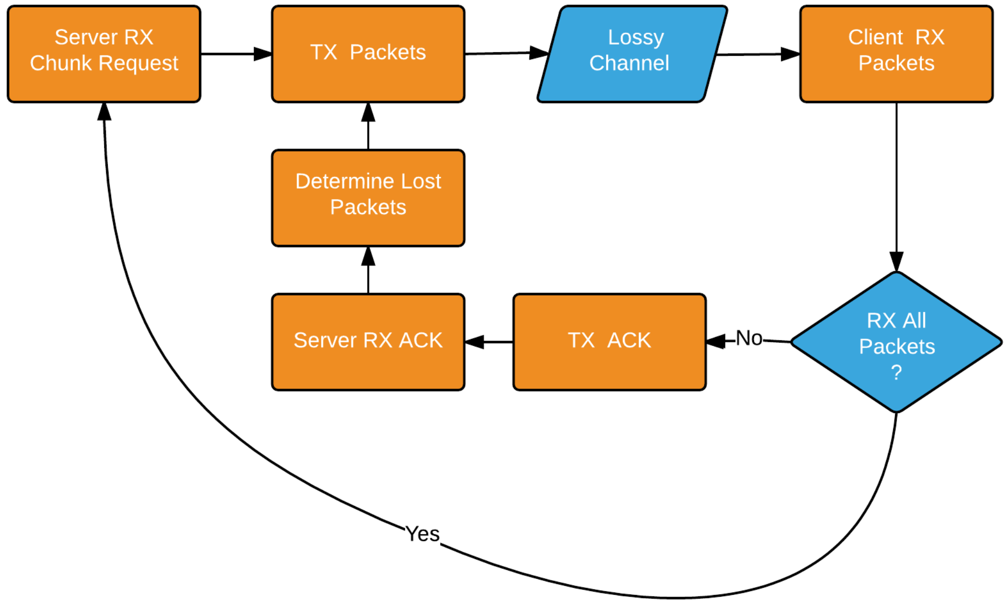


Figure 15. Optimized protocol for reliable UDP for video streaming.

In order to protect video streams from these severe conditions while still supporting multicast and broadcast services, we developed and tested a protected streaming protocol for UDP video packets, that operates similar to the Reliable UDP (RUDP) protocol [18]. This protocol transmits chunks of packets after which individual packets within the chunk are acknowledged using a bitfield in an acknowledgment packet. A server trans-mits these chunks upon receiving chunk requests from clients. When the server receives an acknowledgment, it transmits the packets indicated in the bitfield. See Figure 16. This makes the protocol very flexible and ideal for supporting video streams from both a sender and receiver perspective. A receiver may choose to receive all the packets in the chunk by indicating this in the bitfield, or it may use available video encoding information to only request as much data as it needs for successful decoding. The receiver can then also choose to prioritize packets that include data of high importance such as keyframes within the video. The server on the hand, can also choose to aggregate acknowledgments from several receivers in a multicast scenario, and simultaneously support ensured reliable delivery to all of them.

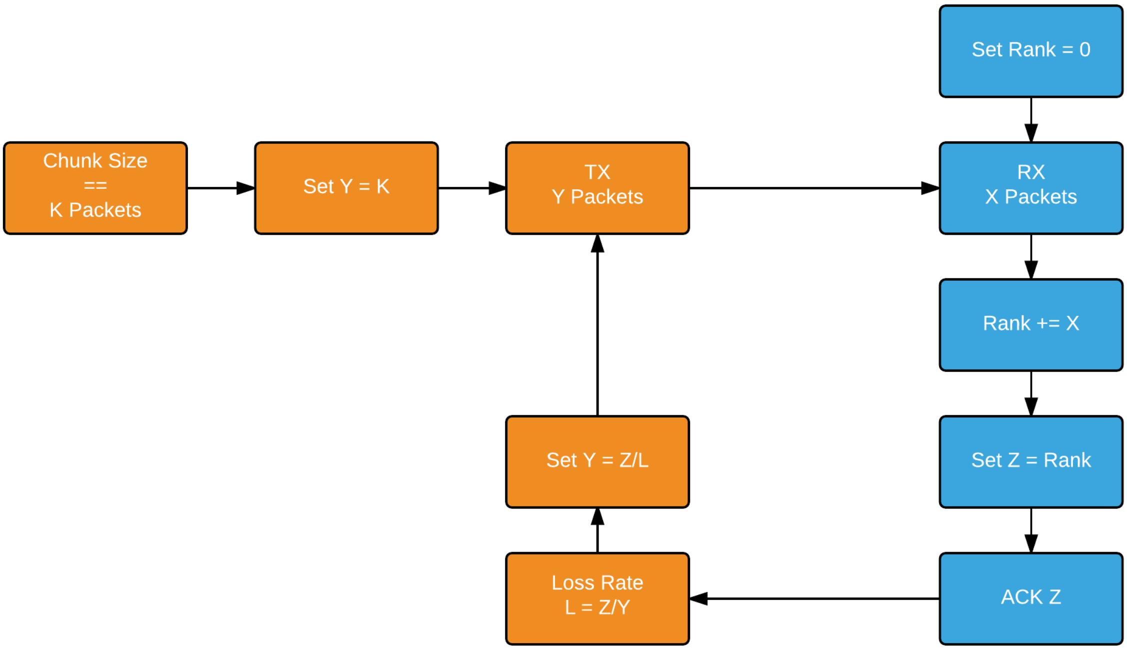


Figure 16. Optimized ARQ using fountain coding.

As mentioned above, fountain coding is an efficient method for transmitting packets over lossy wireless channels. Our protocol therefore also support LT and RaptorQ coding of packets. This version of the protocol operates the same way as the RUDP version, but a fountain code is used for each transmitted packet. The symbols from the fountain code are generated from the requested chunk of data and inserted into the transmitted packets. Instead of using a bitfield in the acknowledgments, only the rank of the decoding matrix is included, which therefore decreases the size of the acknowledgment packets. The server therefore only needs to know how many packets that needs to be transmitted, not exactly which packets, and therefore determines the number of packets to transmit as difference between full rank and current rank. When the server receives acknowledgments from several clients for the same video stream, it uses the lowest rank among the clients.

In a basic fountain coding scenario a sender keeps transmitting symbols until a receiver decodes the message and signals the sender to stop. In a wireless network situation this may become inefficient because between the time the sender sends a packet, and a receiver decodes the message and sends the stop signaling packet, the sender may already have sent several unneeded packets. This leads to unnecessary packet transmissions and a waste of energy. The other option is that sender only sends a certain amount of packets equaling the rank of the message, and then waits for feedback acknowledgments from the receivers. The sender then sends another set of packets equaling the difference between full and current rank. The problem with that approach is that it becomes inefficient in lossy networks as some packets will be lost in each transmission phases, and therefore typically requires several cycles. In this protocol we instead estimate the packet loss rate of the channel by looking at the difference between the number of sent and requested packets. Using this loss rate, an extra number of packets can be transmitted equal to the expected number of lost packets, see Figure 17.

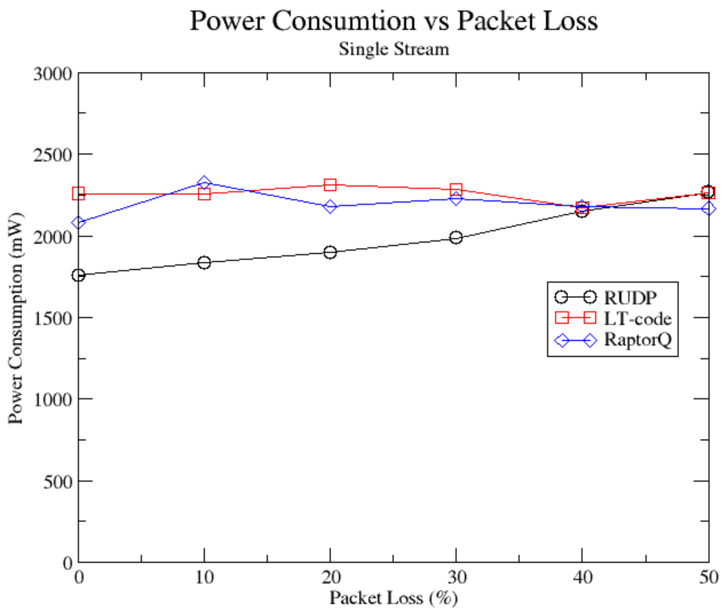


Figure 17

As can be seen from Figure 17, the power consumption of LT and RaptorQ coding not change very much as the packet loss rate increases, while for RUDP it increases almost linearly. That the power consumption should increase should be an obvious conclusion, because as more packets are lost there is a greater need for packet transmissions which therefore increases power consumption. This should also apply to the coding cases, but here you must also take into account the power consumption of the decoding process. The decoding process consumes power, which is why the fountain codes consume more power for the lower loss rates. However, as packets starts dropping, packets will arrive more sparsely allowing the CPU to spread out its work a bit more, which in turn also cools it down a little bit. That is, the energy consumed is spread out more in time meaning the average power goes down. So this decrease in power consumption for LT coding matches very closely the increase in power caused by the extra packet transmissions.

For RaptorQ the same argument can be applied, but its encoder and decoder process is a little bit more complicated. As RaptorQ consists of an inner and outer code, it is also typically decoded in two steps. While Raptor Codes have linear decoding time compared to the size of source data, the decoding effort is less linearly spread out in time than for LT codes. For LT codes, decoding progresses a little bit for each newly received symbol. Because of the two codes, Raptor Codes spends a bit more effort after receiving the last symbol in order to decode the whole message. This results in a more uneven distribution of energy over time, which is further complicated by the video decoder operating on the newly decoding data and the streaming software's data prefetch operation and bandwidth estimation policies. In summary, the LT codes even distribution of work in the decoding process translates into a more even process than RaptorQ, even though packets are arriving less often due to packet losses.

2.2.3.4 Multiple Clients and Live Streams

In the first experiment, only a single client was considered. Because fountain codes are rateless and consumers can receive packets in any order they are ideal for multi user and multicast scenarios, such as for live streams. As we saw in the single client scenario, the extra overhead of the fountain codes results in higher energy consumption compared to RUDP for lower loss rates, while being more efficient for extremely lossy channels.

With fountain coding, if several clients were to consume the same video stream at the same time, the sender wouldn't need to consider which individual packets of the stream each client have received, and the need for explicit feedback is essentially eliminated. This is especially important when multiple clients are considered over lossy channels. Depending on the size of the stream, and the size of the video chunks, it becomes likely that the different clients need different parts of the stream, and that the sender therefore needs to send a separate packet to each of them. Using fountain coding, this need is eliminated and a single packet will be sufficient for all of them which improves both power consumption and throughput.

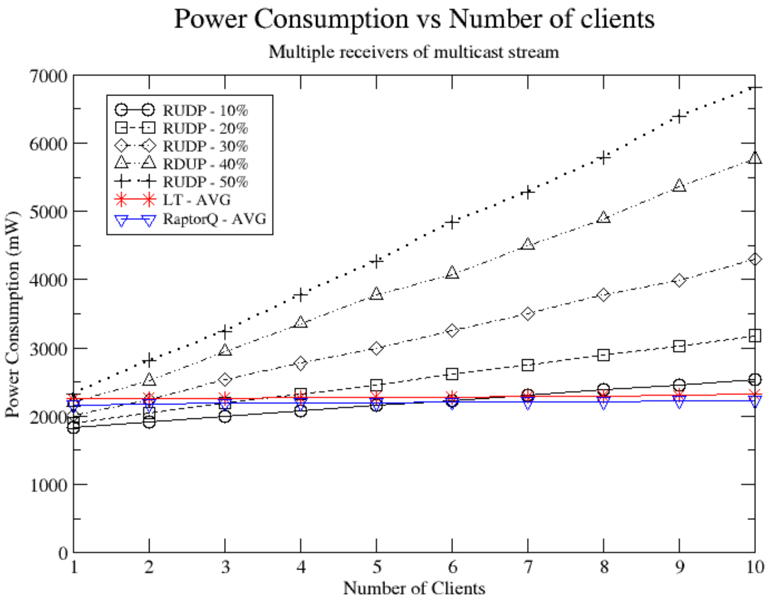


Figure 18

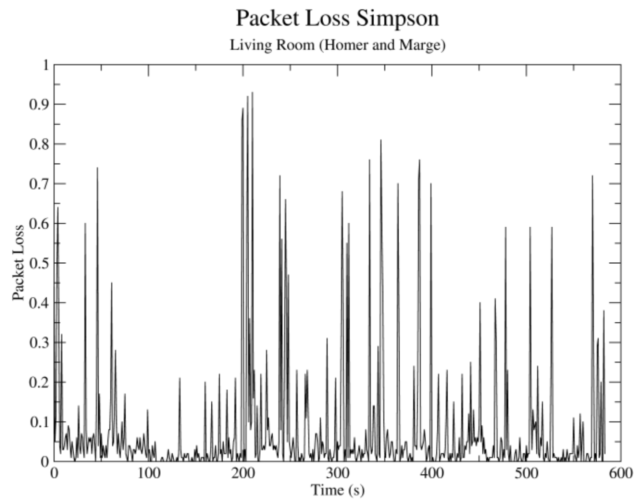
If we look at Figure 18 we can see the results for power measurements when the number of clients increases. The figure shows the power consumption for different packet loss rates for RUDP, but only the average for LT codes and RaptorQ because as saw above, this is not exactly dependent on the packet loss rate.

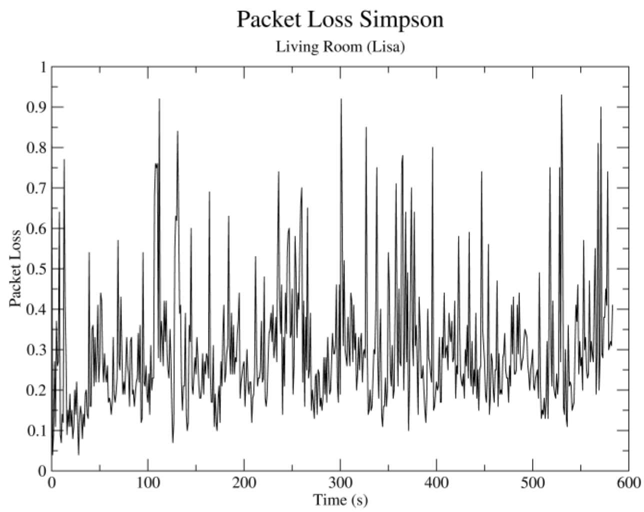
All clients are receiving the same video stream, but as the individual wireless channels are different (independent), clients drop different packets which needs to be retransmitted. The fountain codes maintain the same power consumption even as the number of clients increases. This means that for 10 clients the power consumption can be reduced between 14% and 205% compared to RUDP. For 2 clients, the consumption increases by 13% for 10% packet loss, to being reduced by 30% for 50% packet loss.

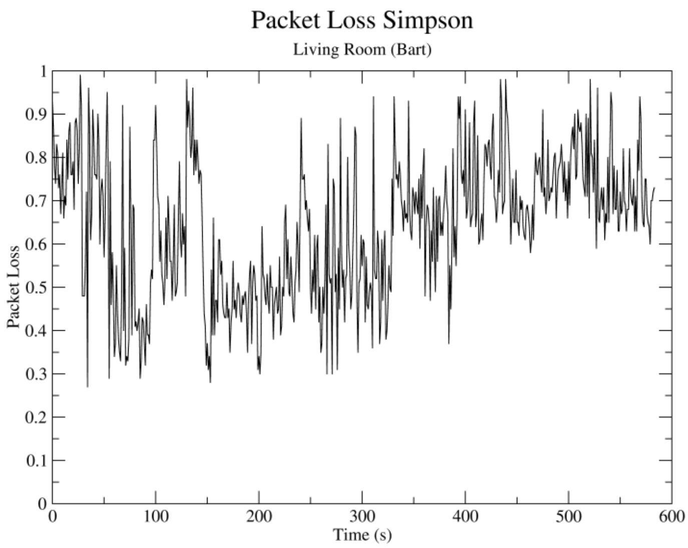
In this setup, all the clients experience the same level of packets loss rate, while in a more non experimental setup this might not be the case. The consumption would then be constrained by the client with the worst channel as the server needs to adapt to its need. It also means that other clients with better channels still receives packets which they don’t need, although these packets will just be discarded. This is a classical problem, called the near-far problem, but which we will not specifically address in this paper. Client devices closer to the source server though, should be able predict this as packets are received in cycles depending on the chunk size. They could thus predict the remaining length of the cycle and turn off their radios and save more energy.

2.2.3.5 Wifi Packet loss variation depending on distance from access point

As we have seen, the effectiveness of fountain coding in terms of reducing the power consumption greatly depends on packet loss behavior and loss rates. Although it can increase capacity and throughput in most situations that experience some loss, the loss rate needs to reach a certain level before it becomes energy efficient. An important question is therefore how these loss rate vary in typical situations. We therefore conducted several measurements where we measured the loss rate in an apartment in central Stockholm, in three different rooms. As can seen in the figures below, the loss rate depends greatly in which room the measurement is conducted, and depends on the distance from the Access Point.


Packet Loss Simpson
Living Room (Homer and Marge)
Packet Loss
Time (s)


Packet Loss Simpson
Living Room (Lisa)
Packet Loss
Time (s)


Packet Loss Simpson
Living Room (Bart)
Packet Loss
Time (s)

As a typical use case, where a family, in this case the Simpson family, is watching several video streams over a WiFi connection. Each member of the family is watching from a different room or at different rate, 1,2 or 6Mbps. In the table below, these values are represented as STO.

As a comparison, we also looked at a rural scenario where there is much less interference from other nearby access points, and where walls also typically tend to be thinner. In these rural case the packet loss rate only varies by a few percent. . In the table below, these values are represented as rural.

Because fountain coding achieves best performance when several users are watching the same stream, such as a live video stream, we also performed studies where Random Linear Network Coding (RLNC) is used that codes several different stream together. After decoding, the stream of interest is extracted while other data is discarded. The reason why this becomes efficient even though lots of uninteresting data is received, is because the wireless channel in WiFi is a broadcast channel, and devices will receive this via their wireless receivers anyway. Even though this mean increased software processing, using RLNC the overall number of transmissions can be significantly reduced.

	RUDP	LT	RQ	TCP	RLNC	RLNC-O
STO Same	2899	2266	2280	8996		
STO Different	8505	9477	8913	9268	8432	7688
Rural Same	1862	2260	2097	7915		
Rural Different	7467	9253	8530	8207	8321	7677

Figure 19 Power consumption values (mW) for LT coding (LT), Raptor Q coding (RQ), TCP, RLNC and optimized RLNC.

Today, it is very common that TCP is used as a transport mechanism for video, especially for OTT video services. For the Simpson use case of watching the same live video stream, if Raptor Q coding is used in the Access Point instead of TCP, power can be reduced by 295%. If the family is watching different independent streams, the use of RLNC can reduce power consumption by up to 21%.
If UDP is used as the transport protocol combined with some feedback to reduce losses that keep video codec operating properly, fountain coding can still save up to 22%. However, in the rural case, fountain coding actually increases consumption by 13%, so the amount savings depends on the conditions of the wireless channel. Note though, that fountain coding improves throughput and capacity in all of these cases, but for good channels this comes at the price of increased consumption.

In conclusion we can say that eliminating the use of TCP from the Access Point to the device always reduces the energy consumption. In typical city environments, RLNC is recommended for parallel independent streams while fountain coding could be used for live streams.

2.2.4 Optimizing DRX for Video delivery over LTE by Utilizing Channel Prediction and In-network caching

Lund University has investigated, through simulations, the energy saving potential sophisticated DRX schemes hold, compared with the currently used static method. We jointly optimize Discontinuous Reception (DRX) cycle length and LTE scheduling to minimize mobile devices' energy usage for video delivery, utilising the now well-established potential to predict future channel conditions in cellular networks. Assumption of in-network caching, allows to set a strict buffer constraint which provides zero buffer underflow to improve Quality of Experience. To this end, two novel DRX approaches are proposed and studied through simulations. The problem is formulated as an integer programming problem which gives the performance bound on energy saving. The results show that more sophisticated DRX schemes (with variable DRX cycle length) can potentially save up to 69 percent energy for mobile devices.

2.2.4.1 Proposed DRX schemes

It is assumed that future data rates for mobile users are known for a future period. We compare the performance of conventional DRX (which we will henceforth refer to as Static DRX (SDRX)), and two novel DRX schemes called Variable DRX (VDRX) and DRXset in the presence of channel prediction and in-network caching.

VDRX allows UEs to utilize any sleep opportunity. This requires modification of DRX parameters more frequently. Considering the signaling overhead associated with changing DRX parameters more often, we introduce DRXset, which incurs less signalling overhead in practice. The DRXset approach utilizes the knowledge of future channel states of the UEs and selects the best DRX cycle length for each UE from a set of DRX cycle lengths. The best DRX cycle length is the one that minimizes the energy usage, and the selected DRX cycles cannot be changed later. Our aim is to minimize energy usage while satisfying smooth streaming with zero buffer underflow.

Figure 20 below illustrates the SDRX and DRXset (a), and VDRX mechanisms (b). In SDRX all UEs have the same constant DRX cycle length, predefined on-duration and inactivity timer. We stress that in this study we assume in-network caching and channel prediction. This assumption is the best case for SDRX because without channel prediction and caching, the eNB needs to consider the packet arrival rate and current channel states to configure DRX, which degrades the efficiency of DRX. Moreover, in practice channel prediction is associated with some error that is not considered in the current study.

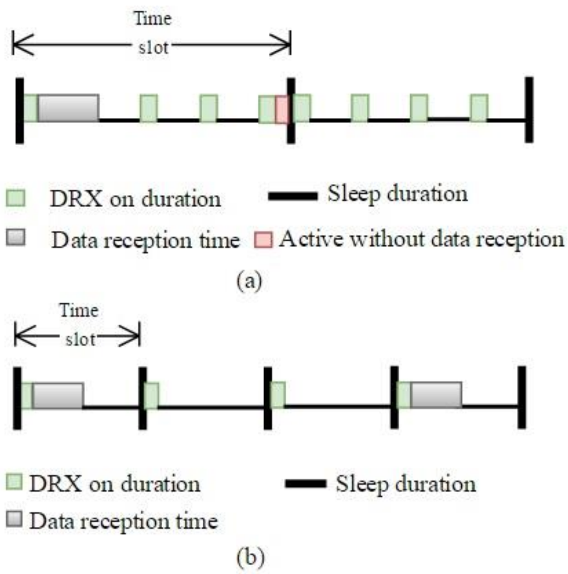


Figure 20 (a) SDRX and DRXset, (b) VDRX

In the absence of prediction errors, unnecessarily frequent DRX on-durations and inactivity timers are considered to be a waste of energy. Further, there are situations where the remaining time to the end of a time slot is less than one DRX cycle. In this case, in our proposed schemes, the UE must stay awake even though it is not receiving data. Although this represents another type of waste, it simplifies scheduling and reduces the complexity of the optimization problem.

By using in-network caching and channel prediction, it becomes redundant to set short cycles or an inactivity timer since we can calculate a sufficient reception time in advance and need not consider new packet arrivals. Therefore these are set to 0. For the same reason we can set the on-duration to 1 ms, as this is the smallest possible value for the on-duration.

In our first proposed configuration, which we call DRXset, DRX cycle length for each UE is chosen from a set of possible values. Figure 20(a) illustrates the DRXset mechanism for one UE. In this case, each UE can be assigned a different DRX cycle length in order to minimize its energy usage. Once the DRX length is chosen, the UE must then keep the same value in all slots. Our second approach, variable DRX cycle length (VDRX) shown in Figure 20(b), allows UEs to change their DRX cycle length every slot if necessary. This means that at each slot each user can receive data and can then switch to sleep mode for the remainder of the slot. Sleep duration can vary in accordance with the length of the reception time. This reduces unnecessary transitions due to short DRX cycles during the slots when the UE is not receiving data. In addition, this approach, due to its variable length, utilizes all available sleep opportunities. We emphasize that proposed DRX settings are possible due to the assumption of channel prediction and in-network caching.

2.2.4.2 Resource allocation and scheduling strategy

The channel prediction window is composed of time slots and each slot is composed of t mini slots, X. Figure 21 below illustrates the relation between resource block (RB), mini slot, and time slot. During each mini slot we allocate all RBs to one user. Therefore if a mini slot is allocated to a user it means that the total number of RBs during that TTI are dedicated to that user. In each time slot either all mini slots or a fraction of them can be allocated to each user, but once a mini slot is allocated to a user, no other users may share it.

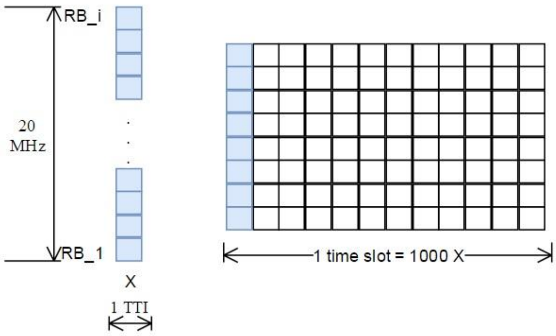


Figure 21 Mini slot and time slot

2.2.4.3 Implementation and Results

The energy minimization problem is formulated as an integer programing problem. It is then implemented in AMPL and solved by CPLEX for different numbers of UEs and video bit rates. The objective of the optimization problem is to minimize the total UE energy usage during data reception periods, active periods with no reception, and DRX cycles. It is subject to zero buffer under-run by setting a lower threshold for the UE's buffer occupancy.

Figure 22 below shows the mean energy usage for different video bit rates for 30 UEs, with 95 percent confidence intervals. The energy usage reduction is the same for 10 and 20 UEs, so we do not show the other cases tested. As shown in Figure 23 (and is also the case for 10 and 20 UEs), VDRX outperforms DRXset and SDRX in terms of energy saving in all scenarios. Although the mean energy usage of DRXset is shown as lower than SDRX, in some cases these two approaches have overlapping confidence intervals. However, VDRX is certainly decreasing energy usage significantly. In fact, although DRXset may be easier to implement due to lower signaling overhead, these results show that VDRX is much more efficient and may thus be worth this overhead cost. Besides which, the computational complexity of DRXset is much higher.

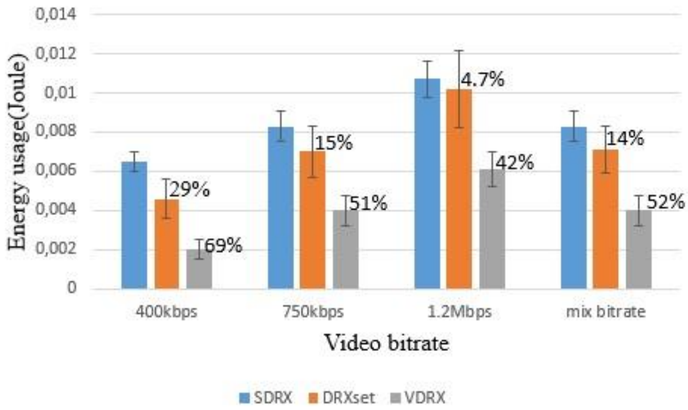


Figure 22 30 UEs- energy usage and percentage of energy usage reduction compared to SDRX

Figure 23below shows the average DRX cycle length for our two proposed approaches compared to SDRX. The bars show that when it is allowed to have a variable DRX cycle length, longer cycles are preferable. In the case of DRXset, in all scenarios 500 ms was chosen as the optimal value. This is because by utilizing the knowledge of future channel conditions, our optimization problem minimizes the energy usage on UEs by scheduling them during their best channel conditions and otherwise letting them sleep.

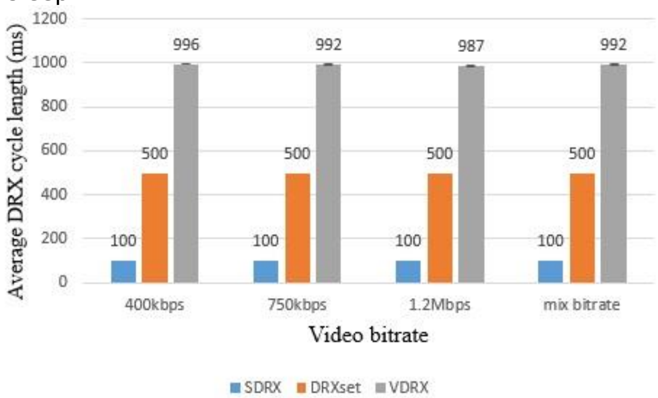


Figure 23 Average DRX cycle length chosen by three approaches (30 UEs)

The scheme also has two other important effects. First, by putting transmission during the best channel states for each UE, the eNB resources are occupied for less time. In other words, the optimization problem releases resources more frequently, which increases cell capacity. In addition, energy usage on the eNB is also reduced because UEs need resources less frequently.

Figure 24 below shows the percentage of empty slots during which no UE receives. These times can be considered as eNB airtime (energy) saving. Better performance of DRXset in this case occurs because our proposed approaches manage the buffer and energy at the same time. If a UE is about to experience buffer underflow it needs to receive data even though the channel is going to improve in the following slots. In this situation, VDRX due to its flexibility allows a UE to receive only as much data as needed in order for the buffer to last until a good channel state. The UE can then switch to sleep mode for the rest of the slot. In contrast, using DRXset, short reception times can result in large time waste (because the remaining time to the end of the time slot would be less than one DRX cycle). For instance, with a time slot duration of 1000 ms, if the UE receives data for 200 ms and the DRX cycle is 500 ms, the UE will experience a waste of 300ms. Therefore it is more energy efficient to send more data when the UE is scheduled, even if the channel is going to be better in the upcoming time slots. This is why VDRX saves more energy, but DRXset uses fewer slots.

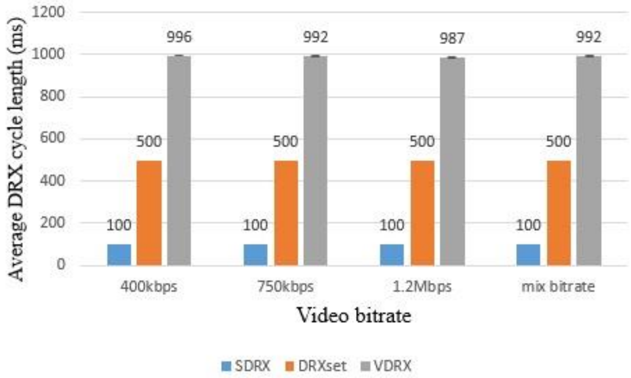


Figure 24 Percentage of eNB airtime saving(1.2Mbps)

3 CONCLUSIONS

In this document, we first started with review of energy saving mechanism in both fixed and mobile terminals proceeded to energy measurements for Wi-Fi and cellular networks. Moreover, we introduced resource usage for multi-interface terminals. The followings points are considered and discussed in this document:

- the main power consuming entities in a mobile terminal include: Wireless modem entities (e.g. wi-fi, LTE etc), Application entity (e.g. application entity running system operating system, and specific hardware/software units handling specific tasks, e.g. graphics), Display, Multimedia content creation entities, e.g. video camera
- Power Saving Mode (PSM) is introduced in IEEE 802.11 standard to reduce the energy consumption of Wi-Fi interface by putting devices into sleep mode when they do not have any data to send or receive.
- A mobile terminal can choose which network interface to use to send/receive data. In some cases, it could use several interfaces at the same time by simultaneously assigning different application sessions to different network interfaces. Based on this, an energy-efficient interface selection mechanism is introduced.
- As the processor is a large constraint for mobile devices, approaches to offload computation from the devices to servers (with fixed power supply) have emerged. All of the mobile computation offloading systems either aim to save energy of the mobile device or make it possible to accomplish tasks that are not normally possible solely using the mobile device.
- Network and fountain coding are two technologies that are currently receiving a lot of attention within both the academic research community as well as in the industry. Both network coding and fountain coding randomly combine a set of packets or pieces of data using a code. The major difference between network coding and fountain coding is typically that in network coding, packets from several different sources are combined, while for fountain coding packets from the same stream or file are typically combined.
- As regards fixed terminals, SoC BCM7252 is used in the project as set-top box hardware. The BCM7252 implements Dynamic Power Management with four different power states, deep standby, passive standby, active standby, and active, which can reduce energy consumption.
- To achieve power efficiency, Vestel uses different backlight algorithms and optical design of backlight unit. Backlight algorithms are considered to provide energy efficiency and to increase contrast perception. Eco backlight and auto backlight algorithms are discussed in this document. Furthermore, new optical design is another part that significantly changes the power consumption of the backlight unit. New optical design is studied in the project.

4 REFERENCES

[1] T. Narten, E. Nordmark, W. Simpson et H. Soliman, «Neighbor Discovery for IP version 6,» IETF - RFC4861, 2007.

[2] «IONICS stratus plugcomputer,» [En ligne]. Available: http://www.ionics-ems.com/plugtop/stratus.html.
[3] «QUAGGA routing software suite,» [En ligne]. Available: http://www.nongnu.org/quagga/.
[4] «Linux IPv6 Router Advertisment Daemon (radvd),» [En ligne]. Available: http://www.litech.org/radvd/.
[5] European Commission, «Code of Conduct on Energy Consumption of Broadband Equipment version 4.1,» 2013.
[6] Cisco. (2015, May) Cisco visual networking index predicts ip traffic to triple from 2014-2019; growth drivers include increasing mobile access, demand for video services. [Online]. Available: http://newsroom.cisco.com/press-release-content?articleId=1644203
[7] M. Rodrig, C. Reis, R. Mahajan, D. Wetherall, and J. Zahorjan, "Measurement-based characterization of 802.11 in a hotspot setting," in Proceedings of the 2005 ACM SIGCOMM Workshop on Experimental Approaches to Wireless Network Design and Analysis, ser. E-WIND '05. New York, NY, USA: ACM, 2005, pp. 5–10. [Online]. Available: http://doi.acm.org/10.1145/1080148.1080150
[8] D. J. MacKay, "Fountain codes," in Communications, IEE Proceedings-, vol. 152, no. 6. IET, 2005, pp. 1062–1068.
[9] A. N. Plymoth, P. Johansson, R. L. Cruz, O. Chipara, and W. G. Griswold, "Grapevine: hybrid cooperative opportunistic routing for challenged wireless networks using fountain coding," ACM SIGMOBILE Mobile Computing and Communications Review, vol. 17, no. 1, pp. 61– 70, 2013.
[10] Q. Xu, V. Stankovic, and Z. Xiong, "Wyner ndash;ziv video compression and fountain codes for receiver-driven layered multicast," Circuits and Systems for Video Technology, IEEE Transactions on, vol. 17, no. 7, pp. 901–906, July 2007.
[11] T. Schierl, S. Johansen, A. Perkis, and T. Wiegand, "Rateless scalable video coding for overlay multisource streaming in manets," Journal of Visual Communication and Image Representation, vol. 19, no. 8, pp. 500–507, 2008.
[12] S. Ahmad, R. Hamzaoui, and M. Al-Akaidi, "Unequal error protection using fountain codes with applications to video communication," Mul- timedia, IEEE Transactions on, vol. 13, no. 1, pp. 92–101, Feb 2011.
[13] C. Fragouli, J.-Y. Le Boudec, and J. Widmer, "Network coding: an instant primer," ACM SIGCOMM Computer Communication Review, vol. 36, no. 1, pp. 63–68, 2006.
[14] S. B. Wicker and V. K. Bhargava, Reed-Solomon codes and their applications. John Wiley & Sons, 1999.
[15] M. Luby, "Lt codes," in Foundations of Computer Science, 2002. Proceedings. The 43rd Annual IEEE Symposium on, 2002, pp. 271–280.
[16] M. Luby, A. Shokrollahi, M. Watson, and T. Stockhammer, "L. minder," raptorq forward error correction scheme for object delivery," RFC 6330,
August, Tech. Rep., 2011.
[17] J. S. Mwela, "Impact of packet loss on the quality of video stream transmission," Ph.D. dissertation, Blekinge Institute of Technology,
2010.
[18] T. Bova and T. Krivoruchka, "Reliable udp protocol," draft-ietf-sigtran- reliable-udp-00. txt, 1999.