Information and Communication Technologies (ICT) Programme

Project No: FP7-ICT- 248036

COAST

Deliverable D6.1

D6.1: Tools for Network Awareness and Adaptation

Author(s):	Jeremy Zylberberg, Chayan Sarkar, Sven Wiethölter, Stephan Rein, Adam Wolisz (TUB), Roberto Baldessari, Stella Spagna, Saverio Niccolini (NEC), Vicente Cruz (TID), Ioannis Koufoudakis, Theodore Zahariadis (Synelixis)

Status -Version:	Final Frozen
Delivery Date (DOW):	30 May 2011
Actual Delivery Date:	10 June 2011
Distribution - Confidentiality:	Public
Code:	COAST_D6.1_TUB_V3.0_20110610.docx

Abstract:

This deliverable describes the underlying techniques and tools for the COAST context awareness, network awareness and content adaptation. It describes how COAST utilizes context information to improve network performance and user experience. Out of the general description, a specific example on context detection is described and evaluated via a prototype, which aims to deliver relevant content to the user. The COAST network awareness tools – the ALTO server and active probing methods to estimate the quality of the links, are detailed and evaluated. Network aware content adaptation is realized by utilization of multiple wireless interfaces in combination with the scalable video coding techniques, and demonstrated to achieve gains in the perceived quality and resource consumption. Last, methods are given to utilize the upcoming radio technologies to ensure the user's continuous connectivity in case of COAST streaming services.

Disclaimer

The COAST Consortium consists of the following companies:

No	Participant name	Participant short name	Country	Country
1	ST Microelectronics	STM	Co-ordinator	Italy
2	Synelixis Solutions Ltd	Synelixis	Contractor	Greece
3	Yahoo Iberia SL	Yahoo	Contractor	Spain
4	NEC Europe Ltd	NEC	Contractor	UK
5	Telefonica Investigacion Y Desarrollo SA	TID	Contractor	Spain
6	Fraunhofer HHI	HHI	Contractor	Germany
7	Politecnico di Torino	Polito	Contractor	Italy
8	Technische Universität Berlin	TUB	Contractor	Germany
9	Fundacio Barcelona Media	BM	Contractor	Spain
10	University of California, Los Angeles	UCLA	Contractor	USA
11	Seoul National University	SNU	Contractor	S. Korea

This page has been intentionally left blank

Table of contents

Abbreviations

ALTO	Application Level Traffic Optimizer
AP	Access Point
BHR	Byte Hit Ratio
CC	Connected Component
CCI	COAST Content Identifier
CCL	Content Cache Locator
CCO	Content Cache Optimizer
CDN	Content Delivery Networks
CEP	Content overlay Entry Point
COI	COAST Object Identifier
CPU	Central Processor Unit
DHT	Distributed Hash Table
DL	Downlink
DNS	Domain Name Service
EDGE	Enhanced Data rates for GSM Evolution
FCN	Future Content-Centric Network
GPRS	General Packet Radio Service
GSM	Global System for Mobile Communications
HD	High Definition
HRW	Highest Random Weight
ISO	International Organization for Standardization
ISP	Internet Service Provider
LAN	Local Area Network
LTE	Long Term Evolution
LFU	Least Frequently Used
LRU	Least Recently Used
MPEG	Moving Picture Experts Group
MS	Mobile Station
NAPT	Network Address and Port Translation
NAT	Network Address Translation
NIC	Network Interface Card
OFDMA	Orthogonal Frequency-Division Multiple Access
OpenCV	Open Source Computer Vision
P2P	Peer to Peer
PQoS	Perceived Quality of Service
PS	Power Save
PSNR	Peak Signal-to-Noise Ratio
QCIF	Quarter Common Intermediate Format
QoE	Quality of Experience
QoS	Quality of Service
REACH	Reflected Exponential Chirp
RFID	Radio-Frequency Identification
RTSP	Real Time Streaming Protocol
RPA	Replica Placement Algorithms
RTT	Round Trip Time

SD	Standard Definition
SDR	Software-Defined-Radio
SE	Search Engine
SLA	Service Level Agreement
SVC	Scalable Video Coding
UL	Uplink
UMTS	Universal Mobile Telecommunications System
URL	Unified Resource Locator
USB	Universal Serial Bus
VoIP	Voice over Internet Protocol
VRA	Video Retrieval Agent
WiMAX	Worldwide Interoperability for Microwave Access
WLAN	Wireless Local Area Network

Executive Summary

The Future Content-Centric Network (FCN) overlay architecture to be built by the COAST project envisions provide multimedia distribution with superior quality and efficiency. The underlying tools, techniques and methods for network and context awareness to enable this content-centric architecture are described in this deliverable. The network awareness drives the smart utilization of network resources, such as selection of the best media data path or the best content source, thus saving network operator costs or improving the Perceived Quality of Service (PQoS). The network awareness can further drive the adaptation of the new media formats, taking for instance limitations in the network resources due to user mobility into account. The context awareness can utilize the features of the adaptive media formats as well, providing a content format that is in line with the user terminal capabilities or the user need due to the current situation.

The deliverable first identifies the types of user context that will be employed in COAST, describes the involved components, and defines interfaces and content representation data format. In COAST, user context is not only utilized for content adaptation but further to automatically select content due to current user need or interest, accounting for the future trend in the information systems to alleviate manual user information retrieval. Consequently, the deliverable then reports a work for detection of user interest based on visual and mobile device sensory data, which aims to supply the user in advance with information on objects he is walking towards, such as information boards or exhibits.

In order to efficiently deliver multimedia content (when interest is detected), the COAST enabling network awareness system may be twofold: on the one hand, it can utilize supplied network provider information, and on the other hand it can rely on active network measurements to estimate the available bandwidth. The former is addressed in the deliverable by presenting network awareness tools and documenting their usage for highly distributed content delivery networks (i.e., the usage of the ALTO service for CDNs). In this respect, simulation results are given to analyse different content caching solutions in terms of user experience, network-aware content placement techniques, and partial caching techniques, in order to provide guidelines for the implementation of intelligent algorithms inside the COAST Cache Optimization (CCO) node. The latter form of network awareness is addressed in the deliverable by available bandwidth measurement tools. A PlanetLab measurement series is conducted to verify if active probing selects the most suitable COAST cache in time. While the previous kind of network awareness decides on the best cache for the content delivery, the last two reported works aim to provide the required connectivity for the mobile user. The former one utilizes multiple wireless interfaces with respect to the available bandwidth in each of them to stream scalable video, thus achieving improved user experience. The latter work describes the discovery of alternative wireless networks by the upcoming multi-mode radios. The discovery is performed during an ongoing streaming service and therefore prevents service interruption due to potential loss of connectivity.

The deliverable finally reflects on the deployment of the described tools and methods for network and context awareness, which will build the foundation of the envisioned overlay architecture.

1. Introduction

As described in the COAST "Description of Work" [40], "COAST aims to build a Future Content-Centric Network (FCN) overlay architecture able to intelligently and efficiently link billions of content sources to billions of content consumers, and offer fast content-aware retrieval, delivery and streaming, while meeting network-wide Service Level Agreements (SLAs) in content and services consumption." This report describes underlying algorithms, techniques and methods to ensure the content-centric network awareness for the envisioned overlay architecture. Specifically, techniques for user context discovery towards retrieval of suitable content and methods for the selection of most suitable options in the network infrastructure and their optimized utilization are detailed.

The described techniques and methods thus comprise the functional chain of the COAST optimized content delivery, starting with the discovery of user context for the selection of relevant content and content format due to situation, activity, user terminal, or user preference. The user context can assess the user mobility in order to find out the user's options for the wireless connectivity. When suitable content and options for connectivity have been discovered, suitable peers or content servers have to be identified. The best peer is selected taking the given network topology and the provider costs due to network utilization into account. The selection can be supported by active measurements to assess the link characteristics, thereby predicting the download time and user experienced quality of service. When the best server is selected, content can be delivered while being dynamically adapted taking the quality of the wireless links and changing options in connectivity into account.

The outline of this report is given as follows. Section 2 gives an overview of the context awareness in COAST, starting with a description of the definition of context and the possible options for context exploitation in the COAST future content-centric network. The analysis leads to guidelines for context information management, including the description of how each piece of context information shall be generated, represented, and interchanged between COAST components.

In Section 3, a work on user context detection for delivery of relevant content is described. The work gives an approach and the underlying methods to discover the user's interest on a nearby object, such as a museum exhibit or an information board. Smartphone sensory and camera data serve to automatically offer the content on the respective device. The methods are implemented on a server and mobile devices and evaluated for their computational complexity and network resource usage in a museum-like environment.

Section 4 presents algorithms and a related simulation study to gain insights in the selection of best caches for the content delivery. The decision is supported by network awareness, i.e. by topology information supplied by the Internet service providers. The given decision methods are evaluated for their reduction of costs, download rates and network resource consumption.

The selection of the best peer can be supported by active available bandwidth measurements. Section 5 surveys available freeware tools for bandwidth estimation. The tools are first tested for their applicability in the PlanetLab platform, which will serve to verify the elaborated COAST content-centric network. A suitable tool is selected and finally evaluated for its performance and network resource consumption.

As COAST envisions the content adaptation due to user mobility or network load, Section 6 presents an approach to utilize multiple wireless interfaces to stream scalable video content. The approach switches the individual codec layers to the available and most suitable wireless links, while it solves the signalling between the involved components and the rerouting of the data flows. The streaming system is extended by two new components, a proxy server for redirection of the data flows and a client software module that triggers the content adaptation. The implemented components are finally evaluated for their functionality and performance.

In Section 7, a new scheme for discovery of alternative networks when utilizing the upcoming multi-mode radio interfaces is introduced. Such multi-mode radios only employ a single transceiver chain, which serves to operate for multiple access networks. The presented scheme is verified to support ongoing WLAN communication while a WiMAX network entry is achieved.

The last section summarizes the findings and reflects about the resulting components to be implemented and finally integrated into the COAST architecture. The findings will serve for the COAST deliverable 7.1 “Specification of the COAST service platform”, where details about all the components to be implemented for the COAST architecture will be given, including their interfaces and the information flow.

2. Context Awareness in COAST

2.1. Introduction

In order to provide the user with the optimal Perceived QoS (PQoS), the COAST platform needs to incorporate some degree of adaptation to the variable circumstances that influences the content delivery process. These "circumstances" is what we are going to define as context for the scope of the COAST project.

For example, if a user tries to access a huge high definition video through a small screened device on a slow GPRS connection, on one hand s/he will not get good PQoS (i.e. the quality of the video will be the maximum the device can handle, while the delay will be huge and the film will freeze many times), while on the other hand s/he will produce a great waste of resources and it will be a source of frustration. COAST will be capable of optimizing the delivery of the content by taking into account the limitations imposed by the context.

Context will be considered here in a broad sense to take into account all circumstances that may have an impact on the delivery of content to the user, including not only the user context but also the terminals characteristics, the network conditions and the user preferences, as illustrated in Figure 1. The user situation includes the user location (office, living room), activity, and transportation mode (car, bus, boat). If the network resources are not available, the user can be offered the content at a later time. If the user is moving to another location, COAST can switch the content to an alternative device to provide a continuous multimedia experience. COAST may select the appropriate format (3D, 2D) for the user or select the content type upon the user situation, such as when she/he stops in front of an information board (airport, train or bus station), additional and up-to-date information can be displayed on the user's mobile device.

This section describes the context-aware features of the COAST platform, the context information relevant for COAST and the way this information is generated, consumed and represented. We first analyze in Section 2.2 which functions of the COAST platform need to implement context-aware features. Section 2.3 defines the COAST context parameters, how they can be generated and identifies the components that will make use of them. Section 2.4 is focused on the context representation format and defines the interfaces for accessing context information. The impact on COAST of various degrees of availability of context information is discussed in Section 2.5.

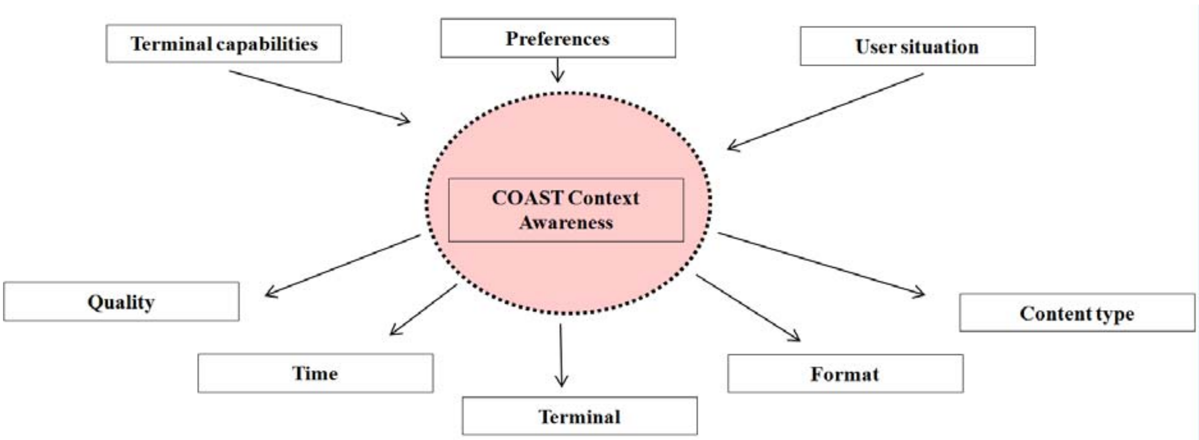

Figure 1: COAST context awareness. Information on the terminal type and capabilities (PDA, Laptop, Xbox, 3D TV), user preferences, and user situation (location, transportation mode) is utilized to improve performance and user experience.

2.2. Exploitation of Context Information in COAST

This section takes a look at the COAST delivery process to identify the parts that need to make use of context information, the type of information involved and the ways it will be exploited.

According to the COAST architecture defined in [18], the following elements are involved in the delivery of content:

- ***User:*** initiates the content delivery process by first searching for content and then accessing one or more search results. The context associated to the user can be literally any information related to him/her: location, country, language, current activity, preferences, tastes, etc. Therefore it is needed to clearly define the scope of the user context in COAST.
- ***Search Engine:*** allows the user to search for content. The SE returns links to the most relevant content for the keywords entered by the user. It is important to note that the SE is designed to return results sorted by relevancy in relation to the provided keywords. Therefore no user context information is involved in the search process with the exception of language and country as these parameters are important for a distributed search process.
- ***Video Retrieval Agent:*** the VRA is the application running on the user's terminal responsible for the actual download of the content. For SVC encoded content the VRA is able to download only the layers that produce the optimal PQoS taking into account the limitations of the terminal, the network and the user preferences.
- ***Terminal:*** it is the device used for accessing the content and for the presentation to the user. The capabilities of the terminal may impose restrictions to the type and quality of the contents that it is able to present. Also, some terminal parameters may be taken into account for content presentation such as battery level, ambient noise and light or actual screen orientation.
- ***Access network or last hop:*** it is the link used by the terminal to access the network. In many cases this part of the end to end path is the one limiting the overall available bandwidth. It is also the most probable responsible for bandwidth variations while the content delivery is in progress especially on wireless links. Also, it is possible that more than one link is available in the terminal, for example 3G and WLAN.
- ***CEP:*** the COAST Entry Point acts as a proxy that triggers the COAST delivery functionality when content is accessed by COI. As the COAST element in closest contact with the users it may play an important role in the optimization of network resources by caching the content better adapted to the characteristics of its clients (users and terminals).
- ***CCL:*** the COAST Cache Locator is responsible for redirecting the CEP to the best cache for the content. Therefore the CCL needs to have information about the network topology and performance.
- ***Cache:*** stores the content and makes it available to the users. Content is replicated across the COAST caches as directed by the COAST Cache Optimizer (CCO) with the purpose of optimizing network resources.

From this summarized view of the content delivery process we can identify some areas which may benefit from some degree of context awareness:

1. ***Content adaptation:*** a correct adaptation of the content to the terminal capabilities, current network performance and user context will enhance user satisfaction and reduce the waste of network resources.

2. ***Traffic localization:*** by placing replicas of the content closer to the users who are actively demanding it, the utilization of network resources is optimized and users receive a better QoS.
3. ***Access network selection:*** when the terminal supports more than one network interfaces, it is possible to select which one to use for content retrieval or even to use more than one simultaneously. By an intelligent selection of network interfaces it is possible to apply different optimization strategies based in user preferences, user activity and network availability. Possible strategies are maximizing video quality or reducing bandwidth costs.

Now, let's have a closer look to these opportunities for context awareness.

2.2.1. Content Adaptation

There exists a great variety of content in the internet. From low quality short videos recorded by camera phones to high definition/3D feature films. Also, there are many types of terminal providing internet access with very different capabilities and network interfaces. From STBs with gigabit interfaces connected to big HD TV sets to GSM mobile phones. Obviously, downloading a huge high bit rate video to a small screened device through a slow link is a waste of time and network resources as the device will not be able to present the video to the user with the original quality. On the other hand, watching a low resolution video on a big HD screen is a painful experience. In both cases, the user gets a frustrating experience either because of the unnecessarily long download times or the low quality. Also, current user activity may further limit the type of content that the user is able to enjoy. For instance, when the user is running or driving, she/he will only pay attention to the audio track of the content and thus there is no point on downloading and displaying the video.

The solution to this problem is achieved in COAST by using SVC encoded video. As explained in [18], an SVC encoded stream consists of several layers that add additional quality to the base layer. These layers can be downloaded separately making it possible to transfer no more data than the strictly required to meet the desired quality. The selection of layers is a function that needs to take into account some context information.

COAST content adaptation policies will take into consideration some or all of the following context parameters:

- ***Terminal capabilities:*** the characteristics of the terminal impose limitations in the resolution, frame rate, color depth and number of layers.
- ***Available bandwidth:*** limits the bit rate of the stream. Even if the terminal is capable of displaying high quality video, the bit rate must match the available bandwidth for avoiding hiccups.
- ***User preferences:*** the user may want to impose additional limitations due to personal tastes or other considerations such as the desire to avoid incurring on data overages on his/her mobile data plan.
- ***User activity:*** knowledge about the current activity of the user can be used to provide additional hints to the content adaptation engine.
- Additionally, ***other context parameters*** may be taken into account in order to achieve other optimizations such as preserving battery life. In this case, video quality could be reduced when battery is low.

The number of layers to download can be determined by applying all these limitations to the content's metadata. Only the layers whose combination meets the target resolution, frame rate, color depth and bit rate shall be downloaded. According to the COAST architecture, this decision is done by the VRA.

It is important to note that not all these context parameters are static. In particular, the available bandwidth may greatly vary while downloading the content. This may be due to variations in the load of the link or in the signal strength in the case of wireless links. The VRA must be able to dynamically change the number of layers in use in reaction to context variations.

2.2.2. Traffic Localization

In the actual internet content is identified by the location of the server that stores it. This implies that all users must access the content's location irrespective of their own location in the network. A better utilization of network resources can be achieved by placing content replicas as close as possible to the users. This is done in COAST by introducing a content object identifier (COI) for detaching the content itself from its location and by replicating the content in distributed caches in a smart way. When a user accesses some content by its COI the COAST platform is able to retrieve from the closest cache improving both performance and network usage.

The COAST components involved in traffic localization are the CEP, the CCL, the CCO and the caches. The information that these components need for achieving the intended optimizations includes:

- ***Network location of the user:*** depending on the location of the point of access to the network (DSL line, mobile cell, etc.) in relation to the network topology, user requests must be directed to the closest cache. This information is needed by the CCL for selecting the closest cache. It is also needed for the selection of the CEP that will be serving the user requests.
- ***Network topology:*** it is used by the CCL to determine which cache is the closest to the user. The CCO also needs to use this information when taking decision about where to replicate the contents.
- ***Network load:*** topology information, which is mostly static, must be adjusted with dynamic load information in order to obtain a more realistic "distance" estimate for each cache. The CCL and CCO use this information.
- ***Content demand:*** the CCO must obtain information about what content is in higher demand in every moment and also the distribution of this demand along the network in order to decide what and where must be replicated.

Information related to network location, load and topology, while needed to optimize the PQoS of content delivery, may be considered as internal to the COAST system and thus not part of the context. Also content demand information can be seen as an internal operational parameter. Traffic localization is further studied in Section 4.

2.2.3. Access Network Selection

Users are increasingly using mobile devices for accessing content. As described in [85], video consumption on mobile devices is being fuelled by the availability of more capable terminals and the ubiquitous presence of mobile broadband access at affordable prices. However, wireless bandwidth is a shared resource and congestion problems are starting to creep up. Most modern mobile terminals such as smart phones and tablets include more than one network interface. This circumstance can be exploited to select the most appropriate network at each moment.

Some optimizations are possible by selection of the access network:

- ***Quality maximization:*** the access link providing the higher bandwidth is selected in order to achieve the highest quality of service. Even more than one link can be used simultaneously for bandwidth aggregation.
- ***Cost minimization:*** link selection is driven by cost. Expensive links will only be used when no cheaper alternatives are available.

The access network selection policy may be defined as a weighted combination of these optimizations, with the user deciding what weights to apply. Some users may prefer quality over cost while some others may prefer a more balanced behaviour.

The element responsible for network selection must be included as part of the COAST user agent running on the user terminal. The VRA will download the content through the selected link(s).

The information needed for the selection of access networks includes:

- ***Access link availability:*** depending on the location of the user and the configuration of the terminal some links will be available and others will not. This information can be obtained from the terminal's operating system.
- ***Access link characterization:*** in order to apply the link selection policy some information is needed about each link. For example, link bandwidth and cost. Depending on the type of information, it may either be measured, as in the case of bandwidth, or pre-configured as in the case of cost.

Links availability and characterization may change very fast: network coverage may be lost or the link may get congested. The system must be able to react to these changes by selecting a different network and/or changing the content adaptation parameters.

Discovery of access networks is described in Section 7. The use of multiple wireless interfaces for content adaptation is discussed in Section 6.

2.3. Definition of Context in COAST

From the analysis of the COAST content delivery process and the identification of the opportunities for context awareness we may define now the scope of the context information for the COAST project.

Context information can be classified into three categories:

- ***User context:*** aspects related to user activity and preferences
- ***Terminal context:*** aspects related to the capabilities of the terminal and its state.
- ***Network context:*** aspects related to access network being used by the user for accessing the content. Network context can be supported by the user context, such as deducing available access networks from the user location.

The following sections describe each of these categories of context information.

2.3.1. User Context

This category includes the context information most directly related to the user that can be exploited by COAST. We will consider two different types of user context: user activity and user preferences.

2.3.1.1 User Activity

We can define user activity as the situation in which the user is immersed when consuming content. This concept comprises the place where the user is at the moment, what the user is doing and the time of day. All this information can be combined to obtain a classification of the user activity within a predefined set of possibilities such as running, working, eating, etc. Each of these predefined user activities imply a definite set of optimization hints that can be exploited by COAST.

While the techniques needed for the identification of user activity are beyond the scope of COAST we may envision how it can be performed:

- Sensory data are obtained from the user terminal or from other sensors worn by the user as part of a Body Area Network. User location and speed can be read from a GPS sensor. WLAN and Bluetooth signals can also be used to locate the user when no GPS is available. Accelerometer sensors can be used to track user movements.
- Sensory data are combined with other sources of information for a more accurate classification. For example, cartographic data can be used to know whether the user is at a restaurant, a theatre, an airport, on a highway or a rail track. User specific data can also be used such as the user's home and office location, usual time schedules, previous history, etc.
- All the data are fed to a classification algorithm that produces either a single output symbol or a list of them with associated probability figures. These symbols represent each of the predefined user activities.

An example of how user activity can be detected and exploited for the delivery of relevant content is described in Section 3.

As stated above, each user activity has an associated set of optimization hints. These hints can be used to modify the behaviour of the COAST context-aware functionalities in order to produce a more satisfactory result.

Many hints could be associated to each user activity. However we will limit them to those which can be exploited by the COAST context-aware functionalities. These functionalities, as described in Section 2.2 are content adaptation and access network selection. A definitive list of optimization hints is not available yet but these are some that are being considered:

- **high_performance**: no performance limitations need to be applied.
- **no_video**: video shall not be displayed as the user will not be able to see it.
- **no_audio**: audio shall not be reproduced as the user will not be able to hear it.
- **high_volume**: audio shall be played at a loud volume due to noisy environment.
- **low_volume**: audio shall be played at a reduced volume due to quiet environment.
- **high_brightness**: screen brightness shall be increased due to highly lit environment.
- **low_brightness**: screen brightness shall be reduced due to poorly lit environment.
- **no_cellular**: cellular radio interface shall not be used as it is not allowed to use it.
- **only_cellular**: only the cellular radio interface shall be used as no other option is possible.
- **no_wlan**: wireless LAN radio interface shall not be used as it is not allowed to use it.
- **only_wlan**: only the wireless LAN radio interface shall be used as no other option is possible.

The set of user activities that can be defined depends on the abilities of the sensors and algorithms used to identify them. Since they are out of the scope of the project we will define a number of common user activities that can be easily simulated or detected. Each activity will have some hints predefined but the user shall be able to change them to his/her needs.

The set of defined user activities and related optimization hints is the following:

User activity	Characteristics	Optimization hints
at_home	The user is at home. No limitations shall be applied.	**high_performance**
at_work	The user is at work. User defined limitations shall be applied.	Defined by the user
running	The user is jogging outdoors and is not able to watch video.	**no_video, no_wlan**
at_cinema	The user is at a cinema in a dark and silent environment.	**no_audio, low_brightness**
at_plane	The user is on a plane. No radio interfaces can be used and the environment is noisy.	**no_cellular, no_lan, high_volume**
at_train	The user is on a train. Only cellular access is possible and the environment is noisy.	**only_cellular, high_volume**
sitting_outdoors	The user is outdoors but not moving. Ambient light is high.	**high_brightness**

Table 1: User activities

User activity context information is foreseen to be generated at the user terminal by some specialized software agent. This element must provide a query interface to get the actual user activity and also provide an event notification service for informing about changes on the user activity.

User activity context information will be used by the VRA to adjust content reproduction and also by the access network selection component to comply with the constraints imposed by the user activity.

2.3.1.2 User Preferences

Some aspects of the COAST content delivery experience can be tailored to the needs and tastes of each particular user. This is done through the definition of a number of parameters that the user can set as she/he wishes.

The defined user preferences are related to the adaptation and presentation of content and the selection of access network:

- **no_hd**: High definition video shall not be downloaded or displayed.
- **no_3d**: 3D video shall not be downloaded or displayed.
- **always_hd**: Force the reproduction in HD quality if available.
- **always_3d**: Force the reproduction of 3D layer if available.
- **allow_link_selection**: Allows the COAST user agent to select the most appropriate access network

- **allow_multilink**: Allows the simultaneous use of several links when possible.
- **link_selection_policy**: Lets the user configure the compromise between cost minimization and quality maximization. This preference may be presented to the user as a slider that the user can move between these two extremes.
- **battery_threshold**: when battery level falls below this value video quality is reduced.
- **cpu_threshold**: when CPU load surpasses this value video quality is reduced.

User preferences shall be set through the user interface of the COAST user agent and stored at the terminal.

The VRA and the access network selection element shall access the user preferences stored in the terminal and take them into account.

2.3.2. Terminal Context

Terminal context represents the characteristics of the terminal that is executing the COAST user agent. Content adaptation depends on this information when selecting content layers for avoiding the download of layers that the terminal will not be able to reproduce properly.

Two types of terminal context information are considered: the static terminal capabilities and the dynamic terminal state information.

2.3.2.1 Terminal Capabilities

This information represents the hardware and software multimedia capabilities of the terminal. This context category includes:

- Display resolution
- Video frame rate
- Display colour depth
- 3D capability of the display
- Supported video codecs
- Audio output channels
- Audio output resolution
- Supported output audio codecs

Terminal capabilities information can be obtained in two different ways:

- It can be dynamically obtained by a software component running on the terminal as part of the COAST user agent. This method relies on the ability of the terminal's operating system to report the capabilities of the underlying hardware and software.
- It can be statically configured into the user agent application. This makes sense when a specific version of the user agent must be compiled for the terminal.

In either way, it shall be possible to obtain the terminal capabilities information through a query interface.

The VRA is the component that will use terminal capabilities information as an input to the content adaptation policy.

2.3.2.2 Terminal State

The state of the terminal may impose restrictions on the quality of the content that should be reproduced. In case of low battery level or high CPU load it may be necessary to reduce the video quality.

The terminal state information that is being considered in COAST is:

- Battery level
- CPU load

A software component running on the terminal as part of the COAST user agent shall generate this information. Internally, this component shall obtain the battery level and CPU load from the terminal's operating system. The information shall be available in two ways:

- Through a query interface it shall be possible to get the actual value of the terminal state parameters.
- An event notification service shall make it possible for a client application to receive an event when some terminal state parameter reaches a given threshold.

The VRA is the component that may use this information to modulate the resources used for video processing. It may set a battery level threshold and a CPU load threshold that when reached video quality is automatically reduced. Threshold values can be set by the user in the preferences.

2.3.3. Network Context

Network context information represents the characteristics of the access link being used for accessing content. Network conditions are variable and must be taken into account by the content adaptation policy, especially on wireless links. Also, in the case of mobile terminals and depending on the location of the user, different access networks may be available.

Network context information includes:

- Available link bandwidth: bandwidth must be continually monitored as link load may vary.
- Available access networks: in case of terminals having more than one network interfaces, this is the information about the links that can be used at any given moment. A cost function must be associated to each interface.

This information shall be generated by a software component running on the user terminal as part of the COAST user agent. The bandwidth monitor shall actively probe the link to obtain an estimation of the available bandwidth. The access network monitor shall rely on the terminal's operating system for detecting the available network. A query interface and an event notification service shall make this information available to client applications.

The components that exploit network context information are:

- The VRA shall use the bandwidth information for content adaptation decisions.
- The access network selection component shall use the information about the available networks in order to decide which ones must be used by the VRA to download the content.

In both cases the best way to use this information is through the event notification service. For bandwidth, several thresholds may be defined and that will trigger a change on the number of layers used. For access networks, an event may be generated when networks come and go.

2.4. Management of Context Information

The previous section describes which kind of context information is used in COAST and how it is generated and consumed. This section describes how context information is exchanged and represented.

2.4.1. Context-related Software Components

From Section 2.3 it can be concluded that all context information relevant for COAST is both generated and consumed at the terminal. This is a direct consequence of the fact that, in COAST, content adaptation is done at the terminal.

Therefore, the COAST user agent is an important part of the architecture. It must be able to generate the context information and then exploit it in the way of content adaptation and network selection. The following figure shows the block diagram of the user agent.

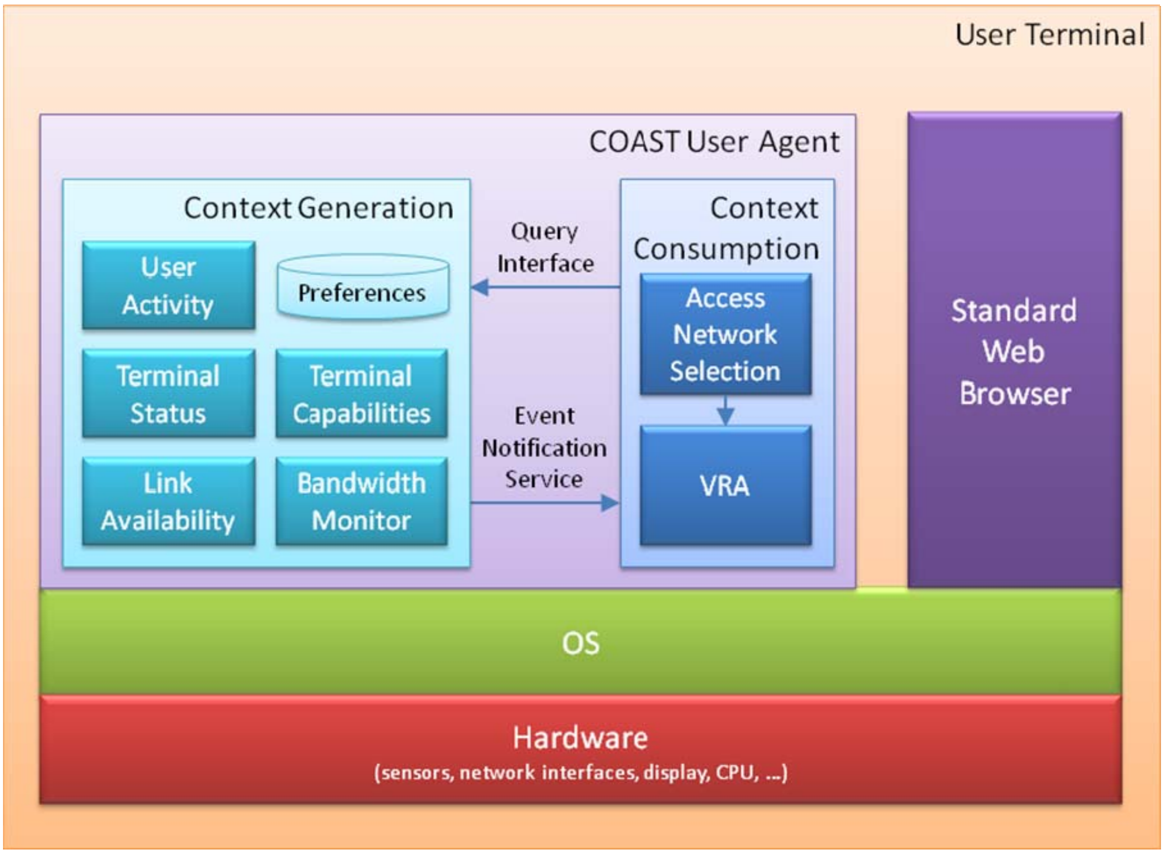

Figure 2: Context generation & consumption within the COAST user agent

The COAST user agent is composed of components that generate context information and components that consume context information. The latter ones are the responsible for the implementation of the COAST context-aware functionalities.

Context generation components take data from the hardware and software resources offered by the terminal. Depending on the characteristics of the terminal, some types of context information may not be available.

Context-aware components may retrieve the context information they need through the interface offered by the context generation components.

2.4.2. Context Retrieval Interface

As shown in Figure 2, the components responsible for the generation of context information may provide two types of interfaces:

- ***Query Interface:*** the client application shall be able to obtain the current value of any context parameter. A call to this interface will return the value of a single parameter or a group of parameters.
- ***Event notification service:*** this service will notify the client application about changes on a particular context parameter. This service will be implemented as a subscription service. The client application will subscribe to one or more context parameters and it will be notified when a change is detected. Depending on whether the context parameter takes discrete or continuous values the subscription is done differently:
    - For discrete parameters no additional information is needed. When the parameter changes the client will receive an event.
    - For continuous parameters the client needs to specify a threshold and a direction (up or down). The client will be notified when the value of the parameter goes above of the threshold (up direction) or below (down direction).

The representation of the context parameters exchanged through these interfaces is defined in the following section.

2.4.3. Context Representation Format

Representation of context information shall be done in a lightweight fashion as it is always exchanged between components running on the same terminal. In the case of mobile devices this is an important requirement since memory and processor resources are scarce.

The MPEG-21 Part 7 Digital Item Adaptation standard [39] contains a comprehensive set of context parameters that even surpasses the needs of COAST. However, the XML notation defined by the standard is too complex and verbose to meet the COAST requirements.

The proposed solution for COAST is to base context representation on key-value pairs. This format is ideally suited to pass information between components, with small requirements for processing and storage. The MPEG-21 standard shall be used as a reference for the definition of the keys and the allowed values.

While the key-value pair notation does not allow for much organization and hierarchization it is possible to group the context parameters by a introducing a specific notation for the keys. We will use a reverse DNS notation: lower levels in the hierarchy are appended to the right separated by a dot. The last part of the key, to the right of the last dot, identifies the context parameter.

```
ctx.termstate.battery_level
```

Figure 3: Example of context parameter key

The following table provides the complete list of keys and possible values defined at the time of the edition of this document:

Key	Allowed values	Multi-valued*
User activity group		
ctx.usract.activity	see Table 1: User activities	no
User preferences group		
ctx.usrpref.no_hd	boolean	no
ctx.usrpref.no_3d	boolean	no
ctx.usrpref.always_hd	boolean	no
ctx.usrpref.always_3d	boolean	no
ctx.usrpref.allow_link_selection	boolean	no
ctx.usrpref.allow_multilink	boolean	no
ctx.usrpref.link_selection_policy	integer 0-100	no
ctx.usrpref.battery_threshold	integer 0-100	no
ctx.usrpref.cpu_threshold	integer 0-100	no
Terminal capabilities group		
ctx.termcap.display.resolution.x	integer	no
ctx.termcap.display.resolution.y	integer	no
ctx.termcap.display.3D	boolean	no
ctx.termcap.display.colour_depth	integer	no
ctx.termcap.video.frame_rate	integer	no
ctx.termcap.video.codecs	string	yes
ctx.termcap.audio.output.channels	integer	no
ctx.termcap.audio.output.resolution	integer	no
ctx.termcap.audio.output.codecs	string	yes
Terminal state group		
ctx.termstate.battery_level	integer 0-100	no
ctx.termstate.cpu_load	integer 0-100	no
Network group		
ctx.net.available_links	string (link identifiers)	yes
ctx.net.bandwidth.{link_id}**	integer	no

Table 2: Context parameters key-value pairs

* Some context parameters may have multiple values like the list of supported codecs. In this case, the value is represented as an array of strings, integers or any other data type.

** The identifier of the link whose bandwidth is being reported/requested is included as part of the key.

2.5. Context Availability Scenarios

As described in Section 2.4, context is generated by some software components running on the user terminal as part of the COAST user agent. Hardware and/or software limitations of the terminal may not permit the generation of some context parameters or even no context information at all. Terminals that don't allow the installation of third party software will not generate any kind of context information.

Regarding the availability of context information, we can define two scenarios:

1. ***Maximum COAST integration:*** The user terminal allows the installation and execution of a COAST specific user agent supporting all or part of the context-aware functionalities described in this document.
2. ***Minimum COAST integration:*** The user terminal does not allow the installation of third party software. Content will be accessed and reproduced by using the standard browser and multimedia player supported by the terminal.

We will now describe the implications for COAST in each scenario.

2.5.1. Maximum COAST Integration Scenario

In this scenario a COAST user agent is running on the terminal. This implies that all context generation and context consumption components as depicted in Figure 2 are available. Therefore, all the optimizations provided by the COAST context-aware functionalities will be possible.

In the case of partial support for the generation of the COAST context information, the optimizations relying on the missing context parameters will not be possible but others can still be provided. For example, if no user activity detection is available, it is still possible to perform content adaptation to the terminal capabilities and network conditions.

2.5.2. Minimum COAST Integration Scenario

In this scenario there is no specific COAST user agent running at the terminal. Content is accessed using a standard web browser. Video is reproduced using the native multimedia application. No context is generated but at the same time there are no context-aware components to consume it.

With this type of terminal the user will be limited to access the contents supported by the terminal, which most probably will not include SVC encoded video. Therefore, no content adaptation will be possible and the availability of multiple access networks will not be fully exploited.

However, even in this case the user will still benefit from COAST. As the access to COAST content referenced by COI is backwards compatible with standard web browsers, the user request will be redirected to the closest cache. When the CEP detects the COI included in the request it will trigger the content-aware and network-aware functionalities built into the COAST platform on the network side.

2.6. Conclusions

Context information is used in COAST with the purpose of optimizing the PQoS of content delivery. Content fetching and reproduction is done accordingly to the limitations imposed by user activity, terminal capabilities and network conditions. This is achieved in COAST through a mechanism of content adaptation. Scalable video encoding makes it possible to download, process and reproduce only the amount of content data that produces the optimal quality in the actual context. Additional optimizations can be obtained by the dynamic selection of the most appropriate access network when the terminal allows for that.

In this section we have identified and described the context parameters that must be taken into account by the content adaptation and network selection features. We have also analyzed how these context parameters can be generated and exploited. As a result of this analysis we have concluded that all context information relevant for COAST is both generated and consumed at the terminal by specialized software components. The internal structure of the COAST user agent has been defined, including all components needed to implement the COAST context-aware features.

An interface has been defined for retrieving context information. It includes a notification mechanism that sends events when context parameters change. Such feature simplifies the implementation of dynamic context-aware adaptation.

A lightweight format for the representation of context information has been defined. This format has been designed taking into account the limitations imposed by mobile devices, which are the type of terminals more exposed to be operated under very different contexts.

Finally, it has been analyzed the impact of the terminal on the functionalities offered by COAST to the user. In this regard, two scenarios have been identified. In a maximum COAST integration scenario, the terminal is capable of executing a specific COAST user agent that implements all or part of the context-aware features described in this section. In a minimum COAST integration scenario the terminal does not allow the execution of a COAST user agent implying that no context-aware features will be available but still content delivery will benefit from content and network-aware access to the content provided by COAST.

3. User Context Detection to Deliver Relevant Content

3.1. Introduction and Problem Description

The COAST functionality includes delivery of content to mobile devices such as mobile smart phones. The COAST service requirement specification [19] describes the “On-the-move” scenario as a context sensitive delivery of content – that is, content is delivered to the user that relates to his current location, situation, and interest. In the corresponding scenario, tourists are moving towards an object of potential interest. When arriving at the object, content describing the (history of the) object may be streamed to the tourist’s smart phone. Similar examples can concern information boards, where the user is provided additional or more current information when standing in front of the board or a museum, where visitors are automatically delivered content on museum exhibits when they are close to or walking towards the exhibit. Such a detection of potential user interest is addressed in this section, specifically, a system is developed that recognizes if a user is close to an object and finds the respective user device to which the content will be delivered. A typical scenario is illustrated in Figure 4, where two users are walking through a museum hall while one of them shows interest in an exhibit by stopping in front of it.

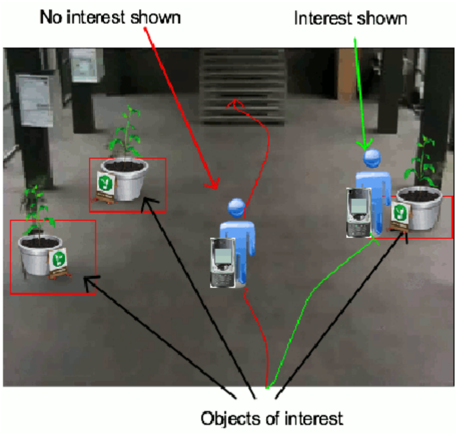

Figure 4: Scenario of a specific kind of user context detection in order to deliver relevant content. Interest is detected when a user moves towards a museum exhibit. When the user arrives at the exhibit, relevant multimedia content is displayed on his mobile device.

The challenge of this interest detection is first that there may be many exhibition objects close to each other, and there may be many users around that area – with only a few or none of them showing interest. The system needs to recognize if interest is shown in such a scenario with a reasonable reliability. Second, it needs to discover the respective user device, as there may be many wireless devices in the area, especially in the here considered case of public places, like museums, train stations, or airports.

The introduced system retrieves smart phone acceleration and camera data in order to find the concerning device and to detect potential interest by an estimation of user proximity to the object. To be more precise, the system maps a set of acceleration data from the discovered wireless devices to a set of camera data, which here contains 2d coordinates referring to the potential users of the automatic content delivery in the by the system covered area. The acceleration and camera data are partially transformed using statistical features thus allowing for the mapping between the visual and acceleration data. The mapping relates a wireless device to the 2d coordinates that are continuously collected by the camera. When the coordinates are within a pre-defined area relating to the object of interest, the relevant content can be delivered to the device.

Requirements of the system include a central server which runs the algorithmic for the detection and a cooperation between the mobile devices and the server. The system provides the following features:

- The interest detection is achieved using off-the-shelf hardware – that is, a single (USB) camera, wireless devices with acceleration sensor as provided by most of today's smart phones, and a small server which performs the needed computations. In many cases, a camera is already installed in the here considered sort of places.
- Privacy of the users is protected as only the users 2d coordinates are collected from the camera images. The images themselves are discarded.
- Different objects of interest can be very close to each other. With the employed hardware, software, and the selected experimentation setup the minimal distance that worked with reasonable reliability was lower than 0.5 meters.
- Content can be pre-fetched to mobile devices in order to make it available in time.
- A localization of the users in terms of absolute position or distance metrics is not needed.

The given features are verified on a working prototype by an experimentation series in a museum-like environment.

Section 3 is structured as follows. In the next section related work to the here considered type of interest detection is given. Section 3.3 describes the components and methods of the introduced system. In Section 3.4 the system is evaluated and the evaluation results are discussed. In the last section the contributions are summarized.

3.2. Related Work

There exists related work to the here introduced interest detection referring 1) to the application in general – that is, interest can be detected in several ways, using RFID tags or localization techniques due to signal strength or propagation time. Related work can refer to different solutions of the here involved components, which are 2) the technique to match the acceleration and visual sensor data, 3) the motion tracking component, and 4) the interest detection. In the following subsection related works to 1) and 2) are reviewed.

3.2.1. Related Applications with Different Approaches

The application of interest detection is addressed in [20] by usage of Radio-Frequency Identification (RFID) technology to provide the mobile web content from [21] in a museum setting. RFID tags were attached to exhibition objects to be detected by RFID readers carried by the registered users. Similarly, in [26] RFID tags are used to provide a system for location aware content delivery, which is proofed by experiments in the Expo 2005. In [22] custom hardware is employed using infrared sensors in order to visually augment the museum experience. The related applications on interest detection do generally require the user to carry customized hardware, installation of complex infrastructure, or do not provide the precision needed in case of very close (possibly small) objects to each other.

In [23] a tablet PC with a touch-screen and a web-cam takes carried by the user takes pictures of the environment. Via object recognition the environment is detected and to the user relevant information on the art objects are provided on the screen. In [24] Bluetooth technology is employed to measure proximity to objects and to deliver information to the devices. The Bluetooth proximity measurement does however not allow for a precise distinction between two objects being very close to each other. The work in [25] records the strength of WLAN signals received by wireless terminals from multiple access points at different locations in a building thus allowing for a user localization. For 92.87 percent of the considered cases an error range of 1.5 -2 m is measured.

QR codes (established as an ISO (ISO/IEC18004) standard) can serve for personalized content delivery but require user interaction and thus do not provide the features of automatic interest detection.

3.2.2. Related Work on Matching of Sensory Data

Camera and mobile device sensory data has been analyzed for a possible coherence in the literature in order to identify or localize users. In [27] normalized cross correlation between acceleration and image sensory data is computed and it is concluded that it is possible to identify one out of three persons wearing an accelerometer. The accelerometer can either be carried by hand or in a bag. The given method there does however not give a final result regarding the identification. Instead it is argued that the visible patterns in the result plots allow for a conclusion, which does not allow to build a readily applicable system.

The introduced method in [28] relies on a user gait detection – that is, the camera images are utilized to detect step patterns from cameras. As the acceleration sensory is required to gather the step motion frequency, sensory mounted to the foot would rather be appropriate than smart phones carried in pockets or bags, which we consider in the context of this section as more suitable devices. In [29] camera networks with cameras installed at the ceiling are employed to match camera and sensory data. The there required infrastructure is more expensive as each camera only covers an area of 6 square meters. Similarly in [30] a camera network is installed at the ceiling each camera observing only a space of 3x4 meters, while not only accelerometers but additionally magnetometers are considered. A real-time experimentation is not conducted. Instead the verification is performed on acquired traces. The required association between the sets of data is based on a probabilistic framework which employs a hidden Markov model and is computationally too complex for a consideration in our real-time framework.

The work in [31] again uses ceiling mounted cameras but accelerometers attached to the person's waist. The experimentation is based on pre-collected data sets and a) verifies the similarity measurement and b) the path disambiguation performance, thus similarly as for the work in [30] not providing a real-time verification of the system.

While the previous works introduce methods to find the corresponding sets of acceleration and camera sensory data, the given methods do not consider the real deployment of an interest detection system in a content distribution overlay architecture. The here considered system is designed to fulfil the real-time requirements and to minimize the deployment effort, such that reliable results are provided in time with a single camera without need for ceiling installation.

3.3. System Components and Employed Methods

The here introduced system has mainly three components,

- the motion tracking with a camera,
- the mobile devices with acceleration sensory, and
- a light-weight server, which performs the computations on matching and image processing and retrieves the data,

as illustrated in Figure 5.

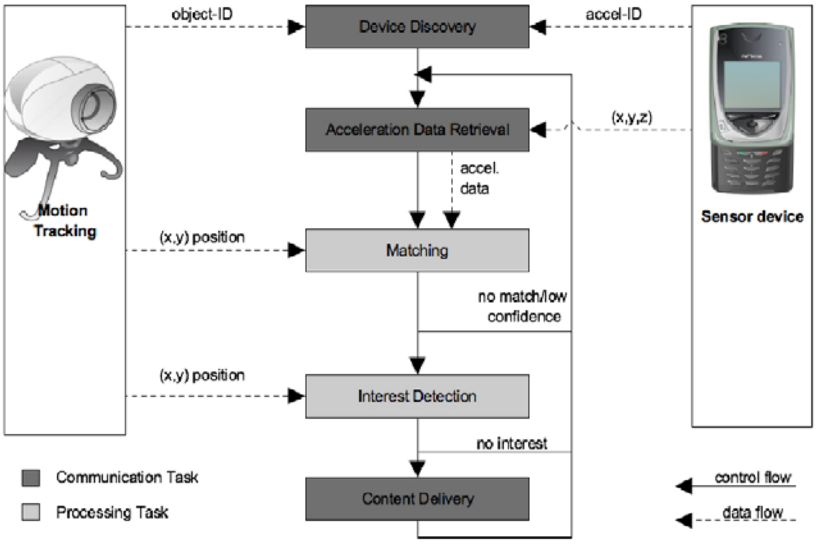

Figure 5: Overview of the introduced system's hardware components and the components' employed software modules. There exist a motion tracking, a small server which coordinates the data retrieval, and one or a set of smart phones, from which the user's acceleration data is retrieved.

Note that for ease of readability, in Figure 5, the motion tracking – that is, the estimation of 2d (x, y) position data, is performed in the camera itself, while in our setup the camera is connected to the server where the 2d position data is computed with an image library. The server employs the modules *discovery, acceleration data retrieval, matching*, and *interest detection*.

The rough outline of a primary system methodology is that first the mobile devices are discovered by the system in that the server detects connectivity to each of them. Then the motion tracking component is asked for the detected number of users that can be tracked (at this point there is not yet any need for computation of 2d (x,y) data). In case there is any user detected by the camera, acceleration data from the mobile devices is retrieved and position data is computed by the motion tracking. The two sets of sensory data (position and acceleration) are given to the matching component, which finds the coherent pairs of position and acceleration data. More specifically, the matching computations allow for a link between a detected user for whom position data is available and the coherent device.

While the matching computations are performed, the 2d position data is continuously estimated and collected. When the matching results are available, the interest detection module inspects the position data for the user proximity to the objects of interest. If the user is within a pre-defined area that relates to the exhibit, user interest is detected and relevant content can be delivered to the mobile device.

In the next sections, the software modules and installed methods are described. In Section 3.3.1 the motion tracking is shortly described. Section 3.3.2 gives more information on the device discovery and acceleration data retrieval, including a description of the here employed hardware. The matching method is based on correlation measures and is detailed in Section 3.3.3. A more refined system methodology for detection of user interest is given in the last section.

3.3.1. Motion Tracking with Camera and Tracking Library

The motion tracking module needs a) to detect an object – that is in our case, the silhouette of a user, and b) it needs to track the detected object. The focus of this work is more on the optimization of the communication aspects for the retrieval of sensory data and their processing – not on the design of image processing techniques. The interested reader is referred to the work in [34] for a survey on object tracking. The starting point here, however, concerned the selection of a freely available software library that provides the needed features.

In this work, the free OpenCV (Open Source Computer Vision) library [35] is selected for the motion tracking, as it contains more than 500 established and optimized algorithms for real time computer vision. Thus it allows for the choice of suitable image processing techniques in the here given context of interest detection.

From the available functions, the OpenCV Video Surveillance Facility[1] was selected to realize the motion tracking. The needed output of the library is for each detected object a set of position estimates and object sizes. The surveillance facility contains the set of modules as illustrated in Figure 6.

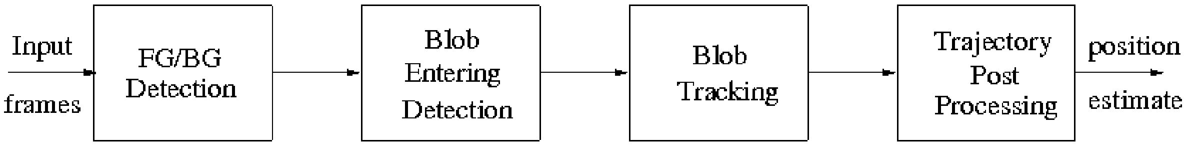

Figure 6: OpenCV Surveillance Facility pipeline. The modules include a foreground / background (FG/BG) detection to separate interesting parts of the image, an entrance detection to identify occurrences of new objects, a Blob Tracking to form a coherence of labeled objects with those in the previous frames by usage of trajectory estimation, and a trajectory post processing to filter out wrong estimations.

The module takes the series of raw camera images as an input (a USB camera is supported by given interfaces) and outputs a list of labelled tracked objects, including timely position estimates for each of them. Furthermore, the library provides a modified input picture output to allow for a visual inspection by the programmer, as illustrated in Figure 7. The 2d position estimates refer to the position of the detected user on the image – not to the real local position of the user in the scene.

Figure 7: Input and output picture of the OpenCV Surveillance Facility library for motion tracking. In the given example, objects of interest are represented by plants. The library detects the users present in the scene and gives 2d position estimates over time.

An exact localization of the user is not needed in the here considered approach.

Examples for the foreground / background detection and for two different algorithms of the blob tracking module are given in Figure 8 and Figure 9, respectively. The CCMSPF algorithm refers to a connected component (CC) particle filtering (PF) based on mean shift (MS) algorithm weight, while the MSPF algorithm does not employ the connected component tracking. For a detailed description of the functionalities and a performance evaluation of the modules and underlying algorithms, the interested reader is referred to the work in [36].

[1] http://opencv.willowgarage.com/wiki/VideoSurveillance

Figure 8: Example for a foreground / background detection. On the left-hand side, the raw image is given, while the result of the foreground / background detection is given on the right-hand side.

Figure 9: More complex scene processed by the blob tracking module using two different algorithms, i.e., the CCMSPF method (image on left-hand side) and the MSPF method (image on right-hand side). The CCMSPF method appears to deliver more useful results for the tracking of detected users.

3.3.2. Mobile Devices, Discovery, and Acceleration Data Retrieval

The user interest detection is based on finding a coherence between the camera and the acceleration data for each of the by the camera detected users. Therefore, the collection of user acceleration data is a fundamental step, which can be achieved with small wireless sensors or smart phones, both carried by the user.

As detection of user context is part of the COAST Deliverable 6.1, and Task 6.1 considers the instrumentation of the users with body area networks, a wireless sensor node is selected here as a mobile device. The suggested methodology for interest detection does however also support the application of smart phones with integrated sensory, as the communication protocols for data retrieval can work with them as well.

As a sensor node, the so called *Irene* node is employed, as it is a recent hardware development equipped with a solid case, see http://www.moteware.com. The Irene sensor node employs a Texas Instruments MSP430 microcontroller, has acceleration sensory, a flash chip, and a radio chip compliant to the IEEE 802.15.4, and is illustrated in Figure 10. In this work it is operated using TinyOS, which is an open source operating system designed for low-power wireless devices.

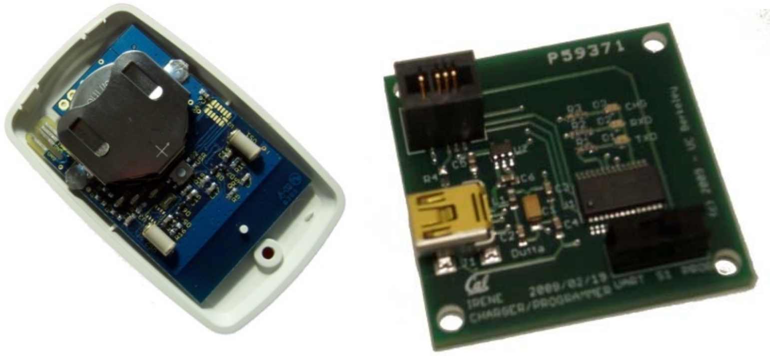

Figure 10: Employed sensor node *Irene* (image on left-hand side) and correspondent programmer module (image on right-hand side) as part of the user context detection. The nodes can be carried in the user's pocket and allow to gather the acceleration over an IEEE 802.15.4-compliant radio interface. A similar functionality can be carried out by smart phones.

The user sensor nodes can send their data to a base station node. The base station is a general wireless (Irene) sensor node that however runs a different software (than the nodes tagged to the users) and is connected via USB cable to the server (or laptop), thus allowing to collect the acceleration data from the sensor nodes. Before a node can deliver acceleration data it has to be discovered as defined in the IEEE 802.15.4, which usually happens when a node enters the connectivity range of the network. The discovery information is made public to the server.

To allow the server the acceleration data retrieval, a sensor node service is implemented that delivers acceleration data due to the requested features:

- Number of 3-d acceleration samples
- Acceleration sample rate

The central server can thus request at any time a segment of acceleration data from a discovered sensor node. As time synchronization between server and sensor nodes is not established, the exact timing of the data in reference to the request time is not known. In the experimentation the delay between request and the moment of data reception was in the range of a few milliseconds and is compensated by a time warping algorithm as part of the matching algorithmic.

3.3.3. Matching of Camera and Acceleration Data

The camera tracking module computes a set of object Ids and for each object the respective 2d position data over time. Concurrently, a set of acceleration data frames is retrieved. (The set of object Ids and the set of acceleration data frames do not need to be of the same dimension, as for instance, a user can be in the camera's field of view but his mobile device might not have been discovered yet, and vice versa.) In this section, the methodology for the matching procedure – that is, finding a match between an identified object's position data and a frame of acceleration data, is shortly outlined.

The matching module can perform computations for many combinations of the available sets of position and acceleration data. The by the module selected combinations are mainly based on the assumption that the camera data is the ground truth, that means, for each of the users detected by the camera the available frames of acceleration data are browsed. An acceleration frame is taken out of the list of available frames if it has been assigned to a user's mobile device, or more precisely, to the acceleration data of a user's mobile device. The assignment of data sets to be compared is part of the matching module. In the following, however, we analyze the module's method to find out if the position data frame of an identified user relates to a given frame of acceleration data. The method uses findings of the work in [31].

1) The position frame data is first transformed to a velocity frame. (Note that the velocity frame has less samples than the position data frame.) Velocity is estimated using small segments of the frame, more specifically, the velocity equals the change in the object position (calculated by the Euclidian distance between the two x,y positions at the beginning and the end of the frame) divided by the dime duration of the position frame.
2) The velocity data is smoothed by computing its moving average, as illustrated in Figure 11.
3) The acceleration data signal magnitude is computed, that means, for each set of x, y, z data only one value remains, as illustrated in Figure 12.
4) From the acceleration magnitude, the standard deviation and subsequently the moving average of that standard deviation is computed, see Figure 13.
5) For the resulting video and acceleration data (Figure 14) the correlation is computed. As the data frames do correspond, a high correlation over the data frame time span is obtained, as visible in Figure 15 (red line).

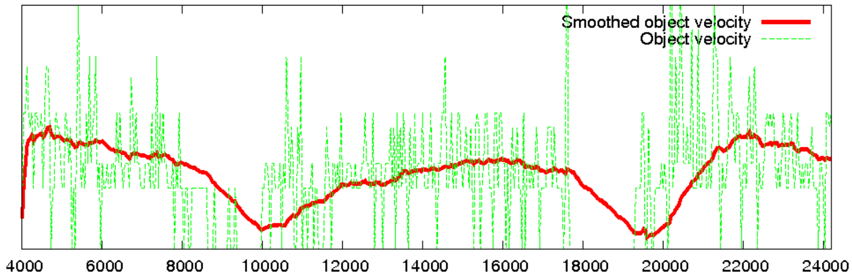

Figure 11: Velocity data estimated from the motion tracking position data and its moving average.

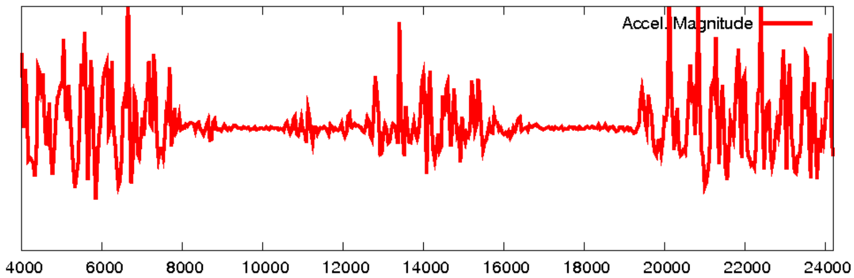

Figure 12: Acceleration data signal magnitude.

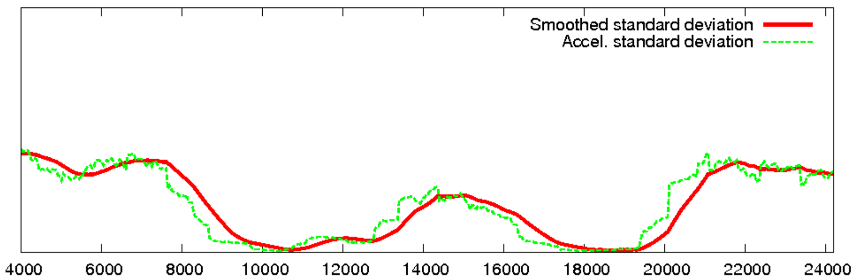

Figure 13: Standard deviation of acceleration data magnitude and the smoothed version of the magnitude.

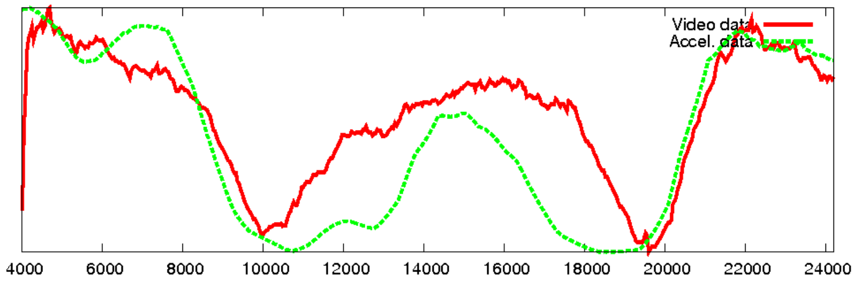

Figure 14: Combined video and acceleration data for the same time frame.

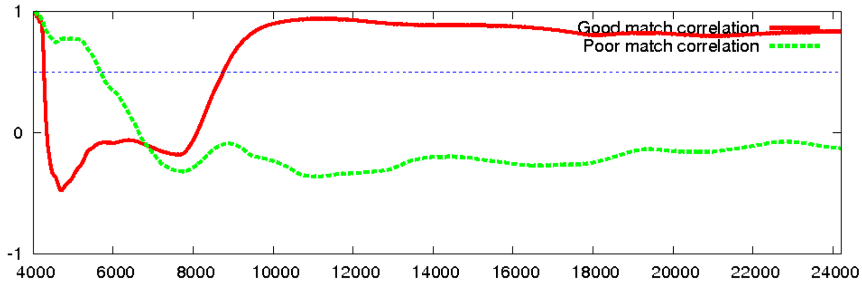

Figure 15: Correlation of the video and acceleration data for the here analyzed data frames (red line), which correspond to each other, and for a video and acceleration frame that do not relate to each other (green line).

The timely coherence of a video data and acceleration frame is established by a time warping algorithm.

3.3.4. Detection of Interest due to Defined Proximity to Objects

When the individual objects recorded by the motion tracking module have been assigned to their respective mobile device (by analysis of the acceleration data), it has to be detected if a user shows interest in a nearby object. Different interest detection methods are possible, such as the prediction of the motion path due to collected position data. A more precise interest detection measurement can estimate head pose [32] or even the user's emotions [33]. These methods are however computationally more expensive, and might not fulfil the real-time requirements.

In this work, we detect interest by a) defining an area of interest which encloses the object of interest, and b) by estimating if the tracked user is within this area, as illustrated in Figure 16.

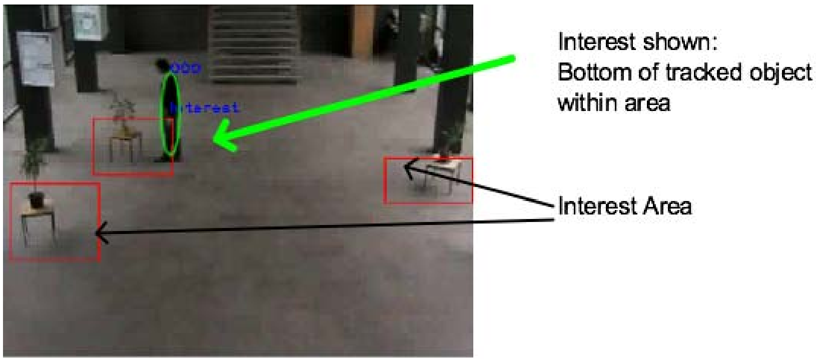

Figure 16: Method for detection of interest in an object. After a device has been assigned to a detected user (for which 2d position data is now available), the system has to estimate if the user shows interest, which is here realized by a definition of an area enclosing the exhibit (here a plant). Interest is indicated if the bottom of the tracked object is within the defined area.

The red rectangles define the areas of interest (here each referring to one of the three plants). If a user enters such an area with his foots, the assumption is made that he has interest in the respective object, and relevant content is delivered to his mobile device.

The method can be refined by definition of a second, larger area of interest enclosing the smaller area. This would allow to take more extensive precautionary measures, such as pre-fetching of multimedia content, thus eliminating the problem of delayed content playback.

3.4. Evaluation of the System

The idea of the in this work introduced system of interest detection is to automatically deliver context relevant (multimedia-) content to the mobile device of the user. In case of false estimations, the system might deliver content which is not relevant to a user or overlook a potential content delivery. In this section some primarily measurements are conducted to evaluate the system's reliability in some example situations.

In order to evaluate the proposed system in a museum-like environment, a study with five participants was performed within an indoor area with the dimension of 15x7 meters. The camera was installed at a height of 4 meters at the entrance side of the hall (not at the ceiling) and angled downwards. Three objects of interest were placed within the area, with a minimum object distance of 3 meters.

The employed digital video camera is connected to a central server, which runs the motion tracking algorithmic to deliver sets of positions with timestamps for each of the detected object. Each participant of the study is equipped with one sensor node, and the accelerometer data is instantaneously gathered and transmitted to the server's base station application.

3.4.1. Scenarios in a Museum-like Environment

We have defined five scenarios to be performed by the participants. The participants were given visual instructions to make clear where they have to move, as illustrated in Figure 17, and a screenplay (script) for each of the scenarios. Regarding the participant's movements the general assumptions are as follows:

- The participants walk on the shortest path to a given destination following the instructions.
- User interest is shown when a participant stops within the pre-defined area of interest.
- The participants maintain their natural and comfortable walking pace.
- Each of the five defined scenarios takes about 2-3 minutes.

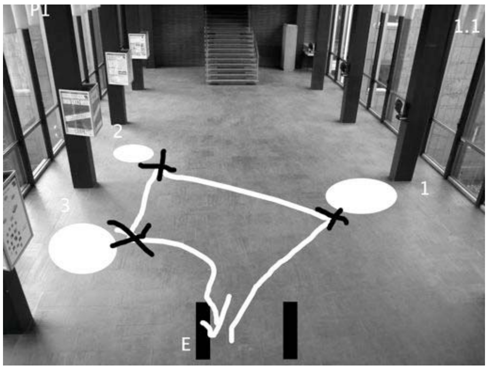

Figure 17: Example for a visual instruction provided to a participant of the study. The participant first has to move to object 1, stop there, then move to object 2, and so on. The visual instructions were complemented by textual explanations (scripts).

Each scenario contains a set of sub-scenarios, which are related to each other by a common technical aspect of the system that is to be evaluated. In the following an example for the scenario 3.2 description is given. The aim of that scenario concerns the evaluation of the system ability to differentiate between multiple people with different movement patterns, where the majority of the participants do not show interest. The actions are described as follows:

- Participant 1 enters room and walks slowly nearby object 3, but does not stop.
- Participant 2 enters the room and walks to object 1, showing interest in it and remaining there for 5 seconds.
- Participant 1 walks further around, but is not showing interest.
- Participant 2 shows interest in object 2.
- Participant 3 enters the room, stands in middle looking around.
- Participant 1 decides to leave.
- Participant 3 decides to leave.
- Participant 2 leaves.

The scenarios 1-4 are detailed in the appendix.

3.4.2. Results for System Precision

In this subsection, results on the system *precision* and system *recall* are given. Precision and recall are basic measurements to evaluate a system's strategy to deliver results. We define the precision here as follows:

Precision = number of correct deliveries / total number of deliveries = number of correct deliveries / (correct + incorrect deliveries)

The precision gives an indication of how accurate the performed deliveries have been. Recall is defined as

Recall = number of correct deliveries / total ground truth deliveries,

where total ground truth deliveries equals the number of delivery opportunities based on scripted scenarios. This gives an indication of a) how relevant the performed deliveries have been or b) of missed opportunities to deliver relevant content.

To give an example, we consider precision = 100% and recall = 50%. This indicates that all deliveries are accurate, but only half of the opportunities to deliver content were used. This can happen due to too slow matching or due to wrong interest or motion detection.

If recall=100% and precision = 50%, all opportunities to deliver content were recognized, however, there were additional delivery attempts that were inaccurate. This can happen if in a first attempt the content is delivered to a device from a person not showing interest, possibly due to an inaccurate matching procedure. The system however manages to send the content to the corresponding device in a second attempt.

Table 3 shows the performance results for the scenarios 1-5.

Sample Rate		Number of People									
		1	2				3			4	5
25 Hz	Precision	1,00	0,50	1,00	1,00	0,80	0,80	0,50	1,00	0,63	0,58
	Recall	1,00	1,00	1,00	1,00	0,67	1,00	1,00	1,00	0,71	0,64
12.5 Hz	Precision	1,00	1,00	0,86	1,00	0,67	1,00	0,67	0,83	0,57	0,63
	Recall	1,00	1,00	1,00	1,00	0,67	0,75	1,00	0,83	0,57	0,45
8.33 Hz	Precision	1,00	1,00	0,86	1,00	1,00	0,75	1,00	1,00	0,43	0,57
	Recall	1,00	1,00	1,00	1,00	0,67	0,75	1,00	1,00	0,43	0,36

Table 3: Results on the interest detection system performance using the basic measurements Precision and Recall for the scenarios 1-5. For the scenarios 2 and 3 there exist 4 and 3 sub-scenarios, respectively , each of them given in one column.

The results show that the system is generally reliable for 1-3 people over the complete range of sample rates, as errors only occur occasionally. For the 4 and 5 people scenarios, however, the results are less accurate, and the performance decreases with smaller sample rate. (The lower performance for the scenarios with four or more people is mainly due to the not precise results of the motion tracking module.) A lower sample rate thus can be sufficient when only few people are in the scene, while a sample rate equal or larger than 25 Hz should be selected if more than three devices are present. A lower sample rate can however enlarge the time duration of the matching procedure. In the next subsection the time consumption of the system's operations is measured.

3.4.3. Results on the System's Time Consumption

Figure 18 illustrates the time duration results for selected scenarios with 2 and 3 participants. The underlying data is a sub-set from the data collected throughout Section 3.4.2, with the intention to get information on the time duration of the involved data retrieval and matching procedures for a successful detection of interest. (Scenarios 1, 2, 3, and 4 here relate to the previous scenarios 2.1, 2.2, 3.2, and 3.3, respectively.) The figure gives the time durations for the discovery, acceleration data retrieval, and the matching procedure. Each bar indicates the calculated correlation for a person's acceleration data (where a person x is indicated as P_x, x=1,2,3) and the camera data of a detected person. A successful match – given here by a cumulative correlation larger than 0.5, for the first person who enters the scene can be achieved in less than 3.3 seconds. Subsequent matches can take more time, and new data must be retrieved from unmatched devices. Data retrieval time is based on the transmission time of individual packets, which grows as matching time grows. The time required for discovery is smaller than 1 millisecond.

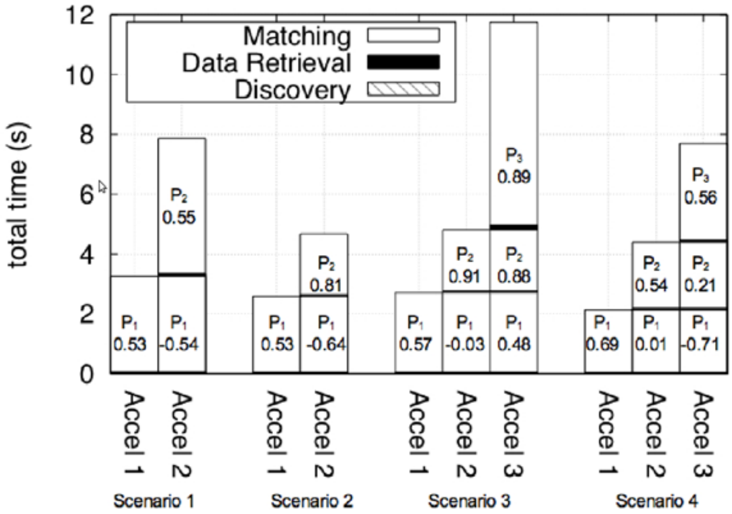

Figure 18: Time durations needed for the discovery of sensor node devices, data retrieval, and the matching for four different scenarios with two or three persons. Each person carries one sensor node measuring acceleration. The persons enter the scene simultaneously. The numbers in the bars reflect the correlation result of the video and acceleration data. The matching computation can take several seconds.

3.5. Conclusions and Future Work

The requirements of the COAST deliverable 6.1 include the description of techniques and methods for detection of user context. User context can be exploited a) to optimize the content delivery (by detection of transportation modes, activity, or local destination, and taking preparations to optimize the performance or resource consumption) or b) to select multimedia content that might be relevant to the user in the current situation. The form of context detection in b) requires a more thoroughly assessment of the user context in that not only the activity (standing, walking, etc.) but also the environment of the user – that is, objects close to the user, is considered. Such a form of context detection, where the meaning of a situation is estimated in order to find out potential user interest, is considered here for the selection and delivery of relevant multimedia data.

Similarly as described in the COAST service requirement specification given in [19], we base our investigations on the assumption that user interest does relate in certain environments to the user proximity to objects or movement towards nearby objects, which for instance can be exhibits in a museum or information boards at airports. In case of detection of interest, the relevant multimedia content is delivered to the user’s smart phone or displayed on a nearby display. In order to achieve this functionality we have identified the need for establishing a link between a user’s movements and his mobile device. While this can be achieved by a localization technique based on radio signals, the limitations in the precision and required installation of infrastructure lead to the consideration of acceleration sensory and visual data, as explained in the related work Section 3.2. The here considered approach thus analyzes the user acceleration and 2d image position coordinates computed from camera data, and thereby constructs a link between the visual data and the mobile device of the user – without requiring a user identification, thus protecting the user’s privacy. After the link is constructed, the user movements are further analyzed in order to detect the interest, with the intention to trigger content delivery when he moves close to the object of interest and stops in a defined area of proximity.

We have introduced a system for interest detection and have first described its main components, i.e. a visual module composed by a camera and software for motion tracking algorithmic, a (set of) mobile device(s) to allow acceleration data retrieval, and a central entity realized by a small server to coordinate and perform the sequence of system operations, including calculation software to find a match between camera and acceleration data (by correlation and significance computation). For the motion tracking we have selected and appropriately parameterized by experimentation a library that estimates 2d position coordinates for people on a stream of input images. Importantly, the position estimate only refers to the position on the input images, not on the real location in the environment, thus only requiring light-weight computations which here are performed in real-time. The acceleration data is retrieved from small sensor nodes carried in the user's pockets. Sensor nodes are employed due to the descriptions in the COAST Task 6.1, which considers the use of body area networks for context detection. The sensor nodes do not only serve for the interest detection, but can further detect the activity as suggested there.

A central server coordinates the retrieval of acceleration data and the subsequent computations to find a match between acceleration and visual data. Through experimentation we have identified a low-complexity technique to find a correlation between the data frames, which consists of a time warping, a data transformation of acceleration and video data to velocity data, and a correlation measure. For detection of user interest from the image position data we suggest the definition of an area of interest enclosing the object. Thereby interest can be easily detected if a user enters that area and stops within it.

We have evaluated the system by a performance analysis for a set of five main scenarios, including related sub-scenarios for some of them. The scenarios relate to a museum environment with three objects of interest, and were conducted by five users emulating typical movements described in the scenarios' scripts within an experimental study. The results show that the introduced system is reliable for up to three people in the scene, while for more people a higher sample rate and a more precise motion tracking module can be advisable. In addition, the total time span needed for interest detection of each user in the given scenarios is verified to be sufficient to proactively trigger multimedia content delivery in time. Parts of these results were recently published in [37].

Future work on interest detection can concern the improvement of the interest detection performance for more users or the reduction of resource consumption on the server or the wireless body area network. A more refined operational sequence, where acceleration data is gathered successively with different frame sizes and sample rates regarding the intermediate correlation results and visual information, such as the number of people or local proximity of the people to each other (which can be related to the probability of wrong motion tracking estimates), can give improvements, and a method for appropriate parameterization has to be developed. A further performance evaluation and user study can provide insights about the time from person appearance in the camera field of view to finding a match between the person's motion tracking identifier and the respective acceleration data. The study can consider different sizes of interest areas or different distances between objects of interest.

4. Traffic Localization through Network Awareness

This chapter is organized as follows: Section 4.1 introduces the network awareness tools and provides a problem statement for the traffic localization problem in CDNs. The following three sections introduce the three enhancements introduced by COAST to content delivery systems based on network awareness, namely network-aware cache selection (4.2) network-aware chunk-level content placement (4.3) and network-aware full content placement (4.4). Section 4.5 shows the simulation environment used for the evaluation of the proposed enhancements and describes the simulations results.

4.1. Introduction to Network Awareness Tools

4.1.1. Problem Statement

There is a class of Internet applications, such as peer-to-peer file-sharing, Content Delivery Networks (CDN), or peer-to-peer video delivery. These applications typically operate on the fact that a single information entity (e.g., a file, video) is distributed in multiple replicas across the network. If an application instance is retrieving such a replica it usually does take into account the availability of the replica and the current achievable throughput between this instance and the remote instances. Other factors, such as resulting costs (either monetary or caused load on servers and network paths), are not taken into account by the applications.

However, the costs caused by applications may be an issue for an operator, depending on the type of application. For instance, peer-to-peer file-sharing applications cause a considerable load on an operator's network and this over a longer duration.

Application usually cannot judge the costs caused by an action, such as loading data from or another location, as these costs are transparent to the applications. There are some methods to guess the resulting costs [83] [84], but they are at best rudimentary. Only the actual operator of a network can have the knowledge about the costs for transferring data within a network or any other information related to network topology.

Figure 19 depicts the regular operation of an example peer-to-peer application, without network awareness. In this figure, peer 2 is retrieving a file from peer 3, crossing the borders from ISP1 to the tier 1 ISP and tier 1 ISP to ISP 2, potentially causing costs to ISP1 and ISP 2. However, peer 2 could cost-effective retrieve the same file from peer 1 which is located in the same ISP 1.

4.1.2. The Application Layer Traffic Optimization (ALTO) Protocol

There are two parties involved in making applications network aware: the application itself and the network operator. The main assumption is that each (or at least some) network operators would offer a service to the applications, where those can query for guidance about which other application instances on other hosts should be preferably contacted first. There is a protocol required between the applications and the network operator's service. The Internet Engineering Task Force (IETF) is working on such a protocol in the Application-Layer Traffic Optimization (ALTO, http://datatracker.ietf.org/wg/alto/charter/) working group.

The goal of the ALTO protocol is to allow applications with an ALTO client to send IP addresses to an ALTO server. The ALTO server will sort the list of IP addresses according to the network operator's preferences and return it to the client. The client *may* respect the guidance of the server and contact preferred IP addresses first. However, there is no guarantee that the client is exactly doing this.

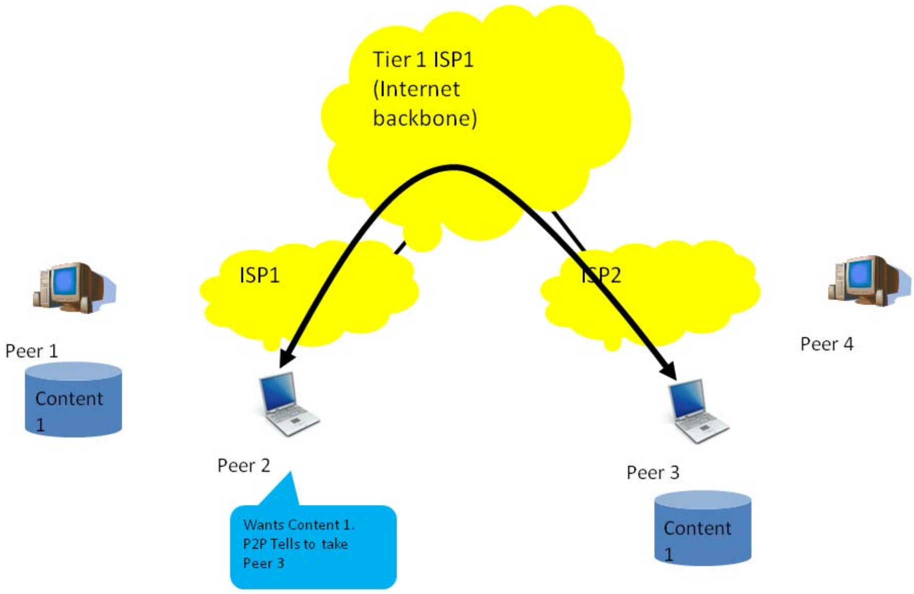

Figure 19: Regular peer-to-peer application operations

Figure 20 depicts the operation of a peer-to-peer application with ALTO. Peer 2 sends the list of IP addresses of possible peers to the ALTO server. In this case, peer 3, peer 4, and peer 1. The ALTO server will return a sorted list according to its own preference. We assume that peer 1 is to be preferred and that peer 3 and peer 4 are getting a low preference value. As a result, peer 2 will connect to peer 1 and retrieve data from this peer. The assumption is that the resulting costs for the traffic exchange between peer 1 and peer 2 are more favourable for the ISP as any other combination. It should be noted that operator's preference could also give peer 1 a lower preference, and other peers outside the network of ISP 1 a higher preferences. The choice of what is to prefer is solely at the command of the network operator.

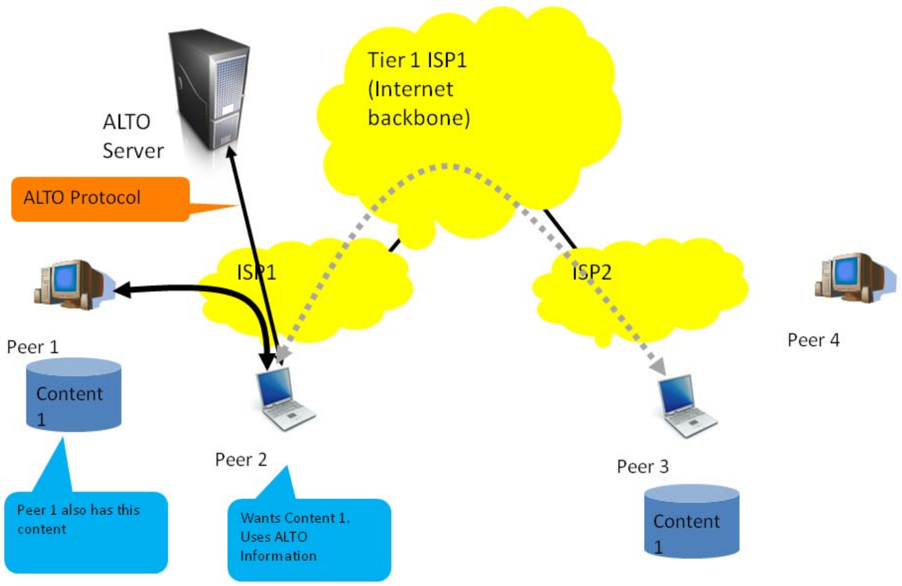

Figure 20: Peer-to-peer application operations with ALTO

4.1.3. Application to Content Delivery Networks

Section 4.1.2 describes the usage of ALTO for peer-to-peer applications. This class of applications was the primary intended use case of ALTO. However, in the course of the ALTO protocol development, the interest also included ALTO-application for Content Delivery Networks (CDN).

Figure 21 depicts the standard operation of a CDN how the closest edge cache is selected by the CDN control logic (simplified view). The biggest challenge for a CDN is to guess the topological location of an IP address and the closest edge cache for this particular IP address. Typically, CDN systems make use of the DNS server IP address, where this DNS server is the network operator's DNS server. Depending from which DNS server the request for a content of the CDN is coming; the CDN selects an edge server.

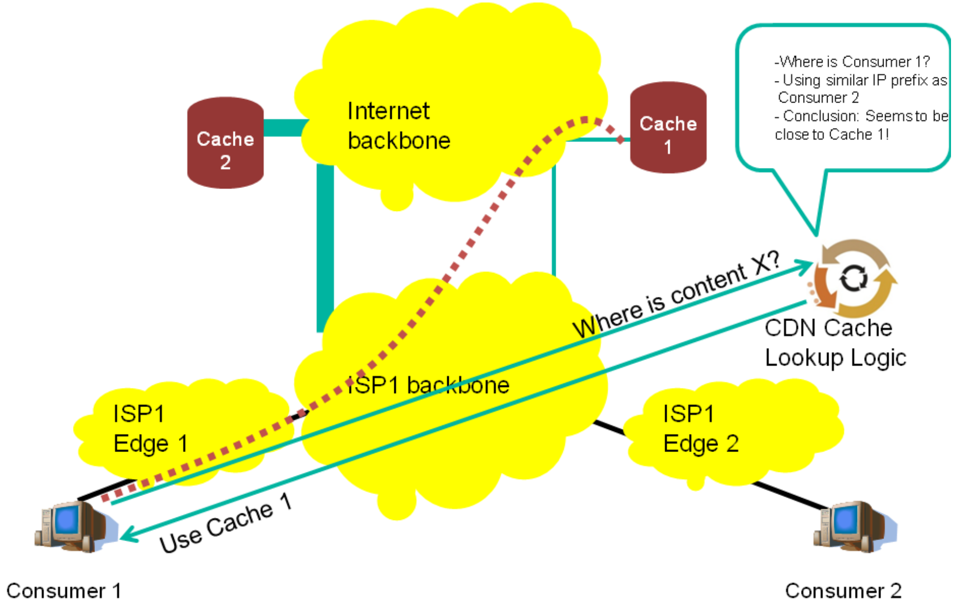

Figure 21: Standard CDN operation (simplified)

However, the relation between the DNS server used and the actual location of the IP is somewhat fuzzy. The CDN operator has to guess which DNS server is serving which region of an ISP. Since this is an educated guess at best, some error may happen.

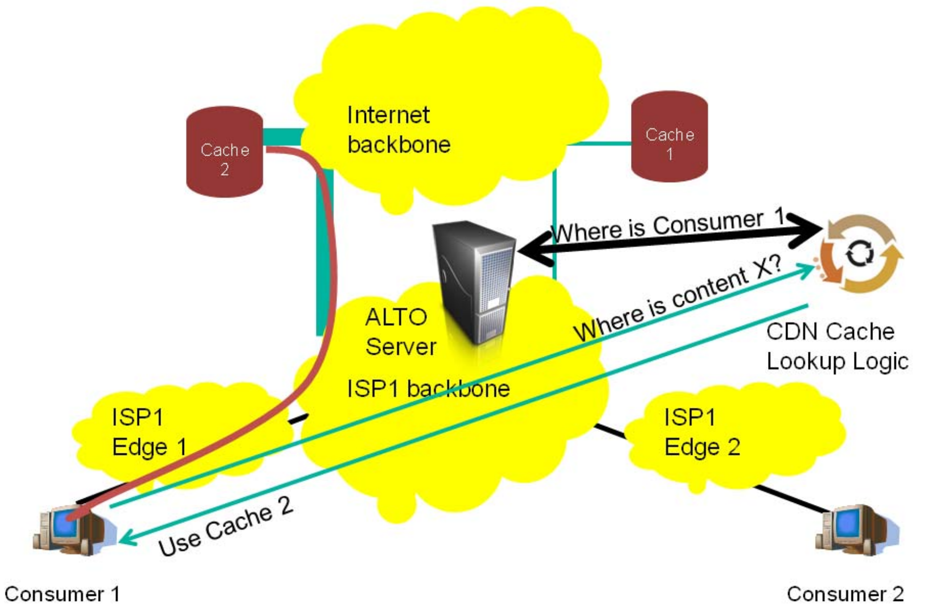

Figure 22: CDN operations with ALTO

Therefore, to release the CDN operator from guessing the location of an IP address, an ALTO server can be provided from a network operator to the CDN. Instead of guessing the location of an IP address, the CDN can simply ask the ALTO server for guidance (Figure 22).

4.2. Network-aware Content Surrogate Selection

Two of the fundamental functions of a Content Distribution Network are replica placement and the already mentioned cache selection. Later in this deliverable, the replica placement problem will be formulated and a network cost-aware heuristic will be presented.

Cache selection or surrogate server selection takes place when a user request is intercepted by a CDN entry point, like the COAST Entry Point (CEP) and consists of the resolution of a the request into a CDN server that can serve the request. This section introduces the usage of ALTO to achieve network-aware content surrogate selection. Further in this document, simulations results are presented that quantify the benefit of adopting network-aware cache selection.

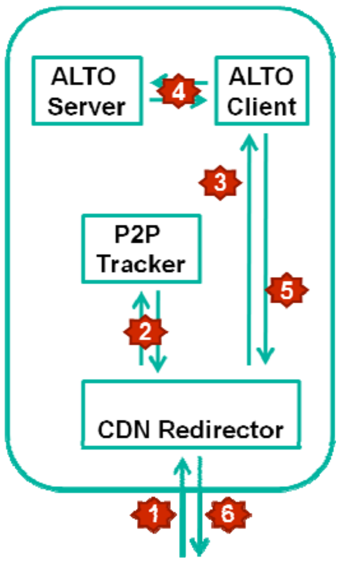

Figure 23: ALTO interaction in the simulated cache selection process

In the implemented simulator introduced in Section 4.5.1, a module simulates all the functions depicted in Figure 23 and here briefly described. The user requests are intercepted by an entry point and forwarded to a CDN redirector (1). The CDN redirector obtains the list of cache servers currently holding the requested content by querying a P2P tracker which implements a content directory (2). Then, the CDN redirector uses an ALTO client (3) to provide the list of candidate cache servers to an ALTO server (4) and obtain back the list ordered based on the network cost. Then, the ALTO client provides the ordered list back to the CDN redirector (5). The CDN redirector merges the network awareness information with other information like the current load on each cache server and selects the cache server that will serve the request (6).

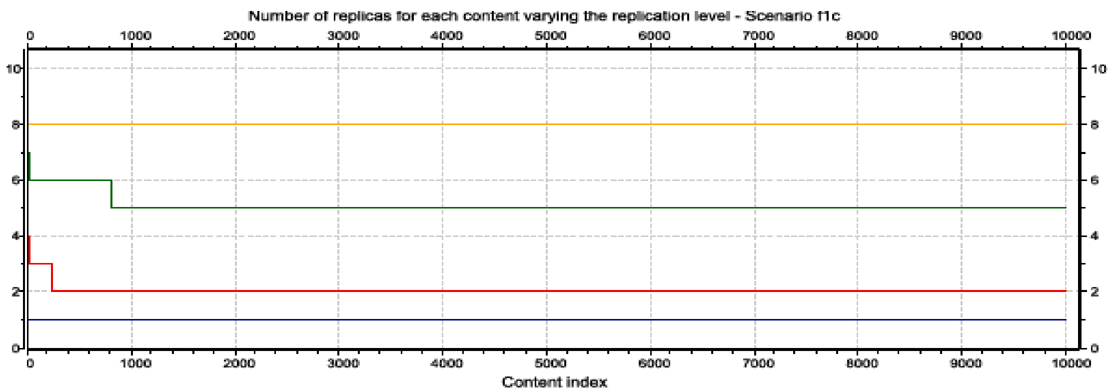

Figure 24: Simple replica number selection based on content popularity

Figure 24 depicts the method used to determine the number of cache servers that should host a certain content. Content indexes are equivalent to their ranking. The 4 lines correspond to 4 levels of replication and will be explained later.

4.3. Network-aware Chunk-level Placement

4.3.1. Taxonomy of Existing Approaches

The widespread penetration of broadband Internet has lead to the access of a great variety of contents. Users can access media object as well as conventional Web objects (i.e. HTML pages and images). It has contributed to a significant increasing of the Internet traffic and to the necessity to design new solutions for Content Delivery Network, to overcome the new challenges coming from the wide contents garden that users can access to.

Historically, Content Delivery Networks (CDNs) have evolved to overcome the limitations of the Internet when accessing Web content. Web caching and replication tune capacity with performance and they have become essential components of the Web. From the traditional Web object, soon the evolution of Web contents has required the design of new approaches for content delivery. As a first example, let's image the evolution of Web site from static to dynamic contents. Most of the modern Web sites include information that constantly changes. It has introduced the cache consistency problems; if content can change during the time, it is important to design a mechanism of caching that can guarantee that the available replica is the updated one. Another issue highlighted by the evolution to dynamic contents, is the need to apply replacement techniques that avoid the update of the whole content but assuring an effective updating.

In this framework, authors of [41] have proposed a fragment-based approach for efficiently creating dynamic web content. The basic idea is to construct complex Web pages using simpler fragments. Fragments represent parts of Web pages that change together; when a change to a set of fragments occur, such fragments are easily identified and updated in the cache server.

This basic idea of partial cache techniques has been applied when the media objects have become popular to end users representing the new challenges for the CDNs. Given the huge size of media objects compared with the conventional web objects, caching them entirely at a web proxy is clearly impractical. The large files are a qualitatively different problem for CDNs; as an example, Akamai has a service called EdgeSuite Net Storage to face with it, where large files reside in specialized replicated storage and are served to clients via overlay routing.

Another new challenge is related to the great variety of media objects that have to be guaranteed to the users. Appropriate placement techniques and replacement mechanism need to be designed to guarantee efficiency and consistence of delivered contents through the CDN architecture. These approaches break away from the ''whole-file'' data transfer model; author of [42] have proposed a partial caching technique for continuous media streams.

Instead of caching entire video or audio streams, the proxy should store a prefix consisting of the initial frames of each content. Upon receiving a request for the stream, the proxy immediately initiates transmission to the client, while simultaneously requesting the suffix (i.e. the remaining frames) from the server.

Recently, driven by the huge success of peer-to-peer file-transfer application such as BitTorrent, a chunk-based model has been applied even to the content distribution in the CDN architecture. The basic idea is that clients download pieces of the file and exchange them with each other to form the complete file. The use of chunk as a basic transfer unit for big file deliver has been applied to the CoBlitz service [43], which runs on the top of CoDeeN content distribution network, an HTTP based CDN. To make this approach as transparent as possible to clients and servers, the dynamic fragmentation and reassembly of the chunks is performed inside the CDN, on demand. Each CDN node has an agent that accepts clients' requests for large files and converts them into a series of requests for the chunks of the file.

The chunk-based content deliver opens the possibility to adopt a collaborative behavior among different nodes participating to the content deliver (i.e. peers). Based on the inter-peers communication overlay topology, it is possible to define two different models; the swarm system and the stream system. The first is an all-to-all communication model (i.e. BitTorrent) while in the second, peers are organized in a tree-like topology. The collaboration among cache servers (peers) in the CoBliTz solution is a swarm system. The request routing among the cache servers is performed using the Highest Random Weight (HRW) strategy. It is similar to Consistent Hashing; for each URL, CoDeen generates an array of values by hashing the URL with the name of each node in the peer set (i.e. cache servers). The obtained values are sorted by load and the final candidate is chosen randomly from the set of the less loaded caches.

CoralCDN [44] is another example of a CDN that exploits the experience of peer-to-peer systems using the chunks as transfer unit and applying a structured model of communication among cache servers. Compared to traditional DHTs, Coral introduces a novel technique in the key-value indexing layer. To improve locality, each Coral node (cache server) belongs to several distinct routing structure called cluster. Each cluster is characterized by a maximum desired network round trip time (RTT). The system is parameterized by a fixed hierarchy of clusters with different RTT thresholds. In this way, a node first queries others in its cluster level before going up in the cluster hierarchy if no nodes hold the required content.

Although the chunk-based distribution system leads to new collaborative models among cache servers, it opens new challenging issue in the replacement techniques. When a cache server has no more available memory to store new chunks, a suitable replacement technique has to be performed to select the more suitable victim to evict from the cache.

An exhaustive literature about replacement techniques for the traditional Web caching exists [45], [45]. Traditional replacement policies are the Least Recently Used (LRU) and the Least Frequently Used (LFU); the first evicts the object which was requested the least recently, the second evicts the object which is accessed the least frequently. Other solutions can be classified as Key-based replacement policies and Cost-based replacement policies.

The replacement policies in the first category evict object based upon a primary key (i.e. the size of the content, the download latency) and ties are broken based on secondary keys, tertiary keys, and so on. Most of these solutions are obtained by modifying the well-known LRU and LFU approaches. Replacement policies belonging to the second category generalize the problem and assign to each content a cost function derived from different function as time since last access, entry time of the object in the cache, transfer cost time and so on. After associating a cost to each content, the object with the lowest cost value is evicted.

So far, the most used techniques are the LRU and the SIZE although solutions belonging to the Cost-based category lead to design appropriate replacement policy that can fit to any particular application.

The recent chunk-based distribution techniques open new issue for the replacement mechanism. Authors of [47] have shown that using a chunk-based replacement technique where chunk is the granularity for the replacement procedure, the memory is slowly freed and it means a slower reaction to popularity changes. To cope with this problem, the segment-based buffer management approach is widely used.

In such approach [48], blocks (i.e. chunks) of a media object received by the proxy server are grouped into variable-sized segments. The segmentation process is transparent to the media content provider or to client; it is the artifact introduced by the proxy server to make cache management more effective. To quickly discard a big portion of a cached media object that was once hot but has turned cold, the segment size increases exponentially from the beginning segment. An exponential number of chunks are grouped together to form a segment (i.e. chunk 0 for segment 0, chunk 1 and chunk 2 for segment 1, chunk from 3 to 6 for segment 2 and so on). Then, when the content is selected to be evicted, using a modified LFU strategy, later segments of it are removed. The exponentially segmentation mechanism allows to free a big portion of the memory in few moves. Referring to the prefix/suffix [42] approach, authors of [48] have proposed a cache admission policy where chunks with a sequence number less that a threshold are always cached (the dimension of the threshold determines the length of the prefix); it leads to cache the beginning part of a popular content. A number of solutions that slightly modify the segmentation scheme are present in literature; for example authors of [49] differentiate their solution using constant-size segments and performing the segmentation when the content is selected to be removed defining a suitable segment size using the statistical values for that content collected so far. Of course, it is possible to combine each solution with one of the existing replacement policy.

To conclude, a wide literature about existing solutions for traditional Web object and successful system as for example the peer-to-peer paradigm can be source of inspiration to design effective strategies that can cope with the challenges presented by the new variety of contents that users can access to.

4.3.2. Design Principles in COAST

Before introducing the design principles, let’s introduce the particular features of a video service, both Live Streaming or Video on Demand.

- Huge Size: for conventional Web object a binary decision works well for the proxy caching (i.e. caching an object in its entirely or not caching). In contrast, a media object has a high data rate and a long duration that lead to a huge data volume. For this reason caching the entire object is clearly impractical.
- Intensive bandwidth used: the object needs to be streamed for the server that holds it to the users and it requires a significant amount of bandwidth. Hence, the bandwidth bottleneck limits the number of clients that a proxy can simultaneously support.
- High interactivity: the inner nature of a media object leads to various client interactions. Above all, different studies as fox example [78] found that nearly 90% media playbacks are terminated prematurely by clients.

Given these unique features of media object, novel caching algorithms need to be defined. Partial caching technique is the solution for the new challenges but to manage which portions of which objects has to be carefully managed. This section addresses the following techniques:

- Content Eligibility Policy: it answers the question about which portions of which objects to cache.
- Content Replacement Policy: for each new content (i.e. portion of content) to cache, it determines the victim for the replacement.

Let’s first consider the traditional way of placing content in caches, as shown in Figure 25, before detailing the partial caching with suffixes and prefixes. The figure shows that for each content a full file is stored in the cache (C1, C2, C3, and C4).

In an extreme of the prefix/suffix caching case, only prefixes of the content are stored in a cache, as shown in Figure 25, by using dotted lines for the suffix parts. The partially cached contents are denoted by C1´, C2´, C3´, and C4´. This partially caching of contents allows to either reduce the disk storage size of a cache, if only prefixes are cached, or to increase the number of cached prefixes while retaining the same disk storage size. The latter point is sketched in Figure 27, where additionally to the prefixes shown in Figure 26, the prefixes C5´ and C6´ are cached.

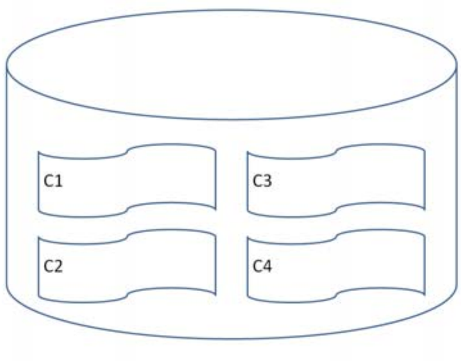

Figure 25: Traditional Content Caching

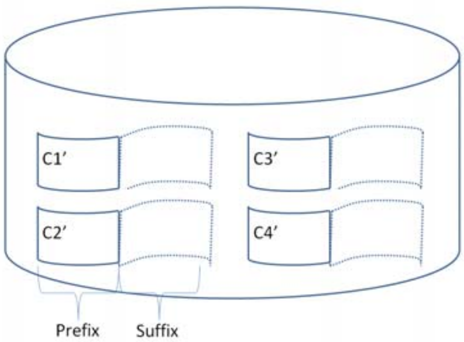

Figure 26: Partial Content Caching

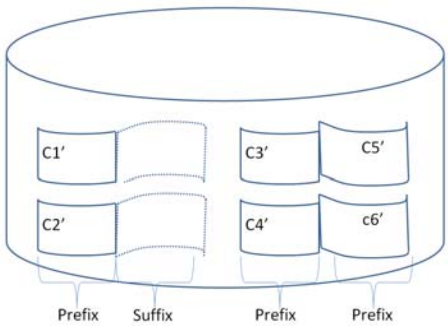

Figure 27: Partial Content Caching (Prefixes)

The basic idea of partial caching technique is to increase the number of contents that the network can suddenly provide to users. Caching the whole contents is the most suitable solution for the users but is highly resource consuming. For this reason, intuitively, it is desirable to cache the whole content (i.e. prefix and suffix) if the number of requests for the content is increasing taking advantage by replication from both users and content provider point of view. Of course, the content provider is interested in a cost-effective decision.

For this reason, the idea is to perform a prefix-based placement and then to cache the remaining part of a video (i.e. suffix) on demand if the popularity for that content increases over a given threshold.

Of course, the decision whether or not to cache the whole content (prefix and suffix) can combine the popularity of the content with same economical consideration to obtain a cost-effective decision.

The second aspect of effectiveness of proxy caches is a content replacement that can yield high hit rate. In general, we can describe the replacement problem using a cost/benefit model. For each content, we want to evaluate the benefit to cache it and the cost to do it. Then, the ratio among cost and benefit determines a score for such content defined as utility value. The goal of a well designed replacement algorithm is to maximize the overall Utility of the cache and it means to remove from the cache the content with the lowest utility value.

In general, as a reference strategy, we will take into account two different Least Recently Used (LRU) stacks, for prefix and suffix respectively. We can imagine the utility value in the LRU scenario as:

$$U_i = 1/T_{last}$$

where T_{last} is the time since the last request for the object. LRU is a traditional and widely used replacement policy that evicts the object which was requested the least recently.

The decision on the two stacks (i.e. prefix and suffix) is taken independently. It is worth mentioning that, in order to maintain caching contiguous media for object N, this caching algorithm is invoked only if chunk (k − 1) of object N has been cached.

A key aspect for a fast reaction to popularity changes is a well designed granularity replacement. The basic transmission unit is the chunk. It increases the modularity of the replacement strategy because it is possible to decide the portion of the memory to free.

In the prefix stack, prefix are replaced as a unit; in the suffix stack a replacement algorithm defines how many chunks of the victim suffix to delete. The mechanism is threshold based; when the content is judged to be stale, the whole suffix is evicted otherwise only a percentage of it is evicted.

4.4. Network-aware Content Placement

4.4.1. Taxonomy of Existing Approaches

The existing state of the art in CDN literature has classified the content distribution according to:

- *content selection and delivery based* on specific user requests*;*
- *placement* of cache servers into some strategic positions so that the edge servers are close to the clients;
- *content outsourcing* to decide which outsourcing methodology to follow.

4.4.1.1 Content Selection and Delivery

The core concept of content delivery network consist to smartly choose which are the contents to put close to the end users and which ones can stay more far away. Every approach for content selection affects the download time from clients perspective and servers load.

Content selection approaches can be divided in *empirical, popularity, object, and cluster*-based. In an empirical-based approach [50], is the administrator the one undertaken to selects the content to replicate to the edge servers. Heuristics algorithms are used to make such decisions. In a popularity-based approach, the most popular objects are replicated to the surrogates. In this technique the video objects are replicated according to their popularity estimation which is not really accurate since it changes dynamically as the time passes by and it's quite difficult to foresee, above all for newly introduced contents.
Object-based approach is a greedy approach, i.e. each object is replicated under certain storage and network constraints [50] [51]. The greedy nature itself of this approach allows to achieve high

performances gain, but it also has problems related to the complexity of its implementation on real applications.
In the cluster-based approach contents is replicated into clusters of users which show similar characteristics. This approach was used by the authors of [52] through web log files. They also turned out the beneficial aspects of this method, indeed is easier to both define groups of users with similar traffic patterns and groups of web pages holding the same kind of contents.
Experimental results show that content replication based on such clustering approaches reduce client download time and the load on servers. However, these schemes suffer from the complexity involved to deploy them.

4.4.1.2 Surrogate Placement

The phase in which the best locations for each replica are chosen is closely related to the content delivery process and it was a really wide explored field. Reducing the user perceived latency for access a content together with minimizing the total bandwidth usage for transferring replicated objects from caches to clients, is the main goal of a well designed surrogate placement mechanism. This means allowing a CDN provider to save infrastructure and communication costs and offer high quality services at low CDN prices [53].

There are different existing surrogate server placement strategies. Theoretical approaches such as *minimum k-centre problem* and *k-Hierarchically well-Separated Trees (k-HST)* model the surrogate placement problem as *the centre placement problem* which is defined in [54] [55].

These kinds of approaches are also very complex to implement, therefore some heuristics as *Greedy Replica Placement* and *Topology-Informed Placement* have been proposed in order to provide reasonable solutions with lower computation costs.

In the **Greedy Replica Placement [56]** method a subset M of N potential sites is chosen in an iterative way. First the cost associated to each location is calculated and the one with the minimum cost is picked up. Then it searches for a second location which has the next lowest cost in relation with the already chosen site and so on like this until all M location are found. The greedy algorithm works well even with imperfect input data even if it needs the knowledge of the clients locations in the network and all pair wise inter-node distances.

The **Topology-Informed Placement** [58] approach starts placing surrogates on those sites which have a high number of connected nodes, assuming that those nodes can reach more locations with a smaller latency. In order to direct clients to the closest surrogate, we need to know the distances between each client and all of the surrogate server. If the network topology is known, then the cache closest to any particular client can be identified by computing the shortest path from the client to all locations, using Dijkstra's shortest path first algorithm, for example.

In [59] the authors presented another server placement algorithm called **Hot Spot** which places replicas close to most traffic generating clients, while in the **Tree-based** replica placement [60] the assumption is that the underlying topologies are made up of trees.

To a CDN administrator is also very important to decide a smart reasonable number of the surrogates to deploy since from it depends how fast and how easy the client can retrieve a content from the network. The main approaches are called *single-ISP* and *multi-ISP [61]*. The policy in the single-ISP method is to put typically one or two really large cache servers in each major city within the ISP covering area, even if the drawback could be that the surrogates may be placed far away from the clients of CDN provider.
In the latter approach instead, numerous surrogate servers are placed close to end users allowing a a reliable delivery of contents – i.e. Akamai manage more than 25000 servers globally [63][64].

The main disadvantage of the multi-ISP approach is that some caches could receive just few content requests resulting in a overall poor CDN performance [65]. Related studies on the performances showed the single-ISP method behave better for sites with low-to-medium traffic volumes, while the multi-ISP approach is better for high-traffic sites [62].

4.4.1.3 Replica Placement Algorithms

Designing replica placement algorithms (RPA) is one of the most challenging tasks in Content Delivery Networks (CDN). These algorithms decide what data to replicate on which storage nodes, in order to achieve improved performance[2] with low infrastructure cost. A number of RPAs have been proposed in the CDN literature.
Replica placement is typically formulated as a NP-complete problem definition that approximates performance, the workload and the target system. To be solved these problems, need heuristics to search for suboptimal solutions within a feasible amount of time and computational resources.

RPA algorithms have to decide where object replicas in the system should be, such that either the client perceived latency and the infrastructure's cost can be minimized improving performances.

Heuristics can be characterized along three axes: *metric scope*, *approximation method* and *cost function simplification*.

Metric Scope. With metric scope we mean the set of all input data and initial conditions like sets of clients, objects and network topology configuration. The heuristic performance strongly depends on the chosen scope. Let's consider for example that there is just one node, in such a case the heuristic would be executed independently on everyone resulting in a decentralized approach; while by the other side, having more nodes involved, will be better to have the intelligence on a single central entity that controls all the other nodes in a centralized way.

Approximation Method. Affects the decision to place which objects into the caching system. Not all the contents have the same characteristics and probably for a specific application, some of them would have a higher priority compared to the others, influencing in this way the selection and the order of the delivery. The approximation methods used in the heuristics are:

- *Ranking:* the ranking approach basically inspects all the possible combinations given the metric scope and calculate the associated costs, sort them and chose the minimum one that doesn't violate any constraints. The cost function is re-computed with a greedy method after the placement of each content.
- *Fixed Threshold*: in this approach the placement decision is up to a specified threshold. If the cost function result is lower than this threshold the content is placed.

Improvement: consists in introducing some slight changes which may improve the current solution. In [73] for example is described the *m-distance improvement* method, that evaluates alternative results permuting at most some bits. The improvement methods are independent of the problem definition.

- *Relaxation Techniques*: are a set of mathematical optimization techniques for relaxing a strict requirement, by either substituting it with a more easily handled one or excluding it completely. Some examples are *Lagrangian relaxation* and *Linear relaxation* [74] which relax the constraints of the problem definition by including them into the cost function.
- *Dynamic Programming:* The key idea behind dynamic programming is quite simple. In general, to solve a given problem, we need to solve different parts of the problem itself (subproblems), then combine the solutions of the subproblems to reach an overall solution.

[2] Performance could be, for example, latency, throughput or availability.

- So intermediate results are saved in order to avoid recalculation. This method is specific to the problem definition.
- *Parametric Pruning:* excludes those neglectable parts of the search space based on a conservative estimation of the optimal cost [74].
- *Hierarchical:* All the aforementioned approximation methods can be used in a hierarchical fashion. For example, ranking can be used on each node at the leaf level. The results from the ranking are then aggregated by the nodes on the next higher level that perform another ranking using this aggregated data, and so on up to the root node [75].
- *Multi-phase:* Some of the techniques described above can be combined in certain ways. For example, greedy ranking can be followed by an improvement heuristic. Multi-phase techniques can also be combined with hierarchal methods.

Cost Function Simplification. It is applied to the original cost function of the problem definition, and depending on the context, it might work well anyway.

4.4.1.4 Topology-Informed Internet Replica Placement

As we said before, the efficacy of a replica placement algorithm is strongly related to the metric scope given as input. The knowledge of the topology of the network for example could result in a better placement decision together with the minimization of client access latency. There have been several studies in the past about replica placement problem and its impact on network performances – two of them [76] [77] considered a greedy placement, which compared to the unfeasible optimal solution, still achieve great results in practice.
In this section we give a description of the existing topology aware replica placement methods.

Greedy placement. It takes all pairwise inter-node distances, links properties and client locations information as inputs and tries to place all the replicas one by one. In a greedy manner one node is evaluated after the other and the one yielding the minimum network overhead is chosen, until all the nodes in the topology have been checked.

Max-router fanout placement. The basic idea behind this method is that there would probably be some nodes in the topology with a larger fanout[3] and (on average) they are eventually the closest to all other nodes, so they might be the better locations to hold content replicas.
Max router fanout placement so, starts placing these replicas to those nodes with higher fanout and then following the decreasing order of the node degree until everyone have been chosen.

Max-AS/max-router fanout placement. Really similar to max-router fanout placement, this method just assumes that each node is included in some AS and that all ASs are connected to each other. First a number of ASs with the greater fanout is chosen, then within each AS, the router with the largest router level fanout is chosen. In this way the selected nodes will be closer to the other ones.

Max-AS/min-router fanout placement. This method is similar to the max-AS/max-router fanout placement, except that instead of selecting the router with largest fanout within each of the chosen ASs, we select the router with the smallest fanout. This placement may not make sense for practical purposes, but there is still need to consider it in order to evaluate the sensitivity of network performance.

[3] We will use interchangeably the terms node degree and node *fanou*t to represent the number of links connecting a node with its neighbors.

Random placement. Is actually considered as an "upper-bound" method in a sense that an efficient replica placement should always perform better than random. It choses replica locations randomly with uniform probability among all nodes in the topology.

Choosing the most suitable replica placement approach is not an easy task to achieve, as we said deciding how many replicas to create and where to put them, in order to have an aimed performance goal, minimizing certain costs is an NP-hard problem. Of course this decision strongly influences the results in terms of cost of required infrastructure (cache server capacity, network bandwidth, computational power etc.).
Moreover, every heuristic is designed on the basis of its particular application scenario, this means that a heuristic which shows better in one case may be completely inappropriate in a different context, either be too much expensive or even unable to reach the minimum performances goal.

4.4.2. Design Principles in COAST

The problem of replica placement consists of finding an assignment of objects replicas to a set of cache servers such that a certain metrics is maximized. Commonly used metrics include network cost, latency, caching cost, replacement cost. In our approach, the metrics used to indicate the quality of an assignment is a cost function C(X) defined by (1).

$$\mathbf{C(X) = C_{NW}(X) + C_C(X)} \qquad \textbf{(1)}$$

For a certain assignment X, C(X) defined in (1) takes into account both network costs $C_{NW}(X)$ and caching costs $C_C(X)$.

- The network cost $C_{NW}(X)$ is expressed as a function of the amount of transmitted data. The cost per unit of transmitted data c^u_{NW} is assumed to take into account network operational expenditures (OPEX), since network infrastructure is assumed to be already in place. However, the same variable c^u_{NW} can also be used to include capital expenditures (CAPEX) in the case that network upgrade or installation is planned.
- The caching cost $C_C(X)$ contains both the cost of storage space $C_{STO}(X)$ and the cost of serving requests $C_{SER}(X)$, being the first one a function of the cost of the physical per-unit storage space c^u_{STO} and the latter a function of the cache server resource cost per unit of data being server c^u_{SER}. C^u_{SER} is mostly due to I/O resources occupied per unit of data, being CPU resources occupied by a content transfer insignificant.

$$\mathbf{C_C(X) = C_{STO}(X) + C_{SER}(X)} \qquad \textbf{(2)}$$

It should be noted that the ratio

$$\boldsymbol{\alpha = c^u_{NW} / c^u_{STO}} \qquad \textbf{(3)}$$

is a critical factor that determines the optimal assignment. In [79], the authors analyse the impact of this ratio on the optimal location of cache servers in a 3G network. It should be noted that this ratio depends on the network operator and it is impossible to assign values that are universally correct. As references, cloud storage providers apply different pricing ratios: Amazon [80] charges $\alpha = 0.71$, Rackspace Hosting [81] charges $\alpha = 1.09$ and Nephoscale [82] charges $\alpha = 1.33$.

The other important unitarian cost ratio is

$$\boldsymbol{\beta = c^u_{sto} / c^u_{ser}} \qquad \textbf{(4)}$$

According to [79] $\beta = 1.3$ based on cloud services. However, no details are provided as to how this ratio was determined. We believe this value is overestimated because it takes into account huge computational power, whereas for CDNs, cache servers have to bear a load that is comparable to a file system get operation. We found that both Amazon [80] and Nephoscale [82] charge 0.01$ per get/put request. Assuming an average video size of 30 MB, we found that both Nephoscale and Amazon charge $\beta \approx 4.5\text{x}10^{17}$. Therefore, we can conclude that $c^u_{SER} << c^u_{STO}$.

An assignment X is a K × J matrix. Since each content $k \in K$ cannot have more than one replica hosted on a single cache $j \in J$, the matrix X is made up of binary values $x_{k,j} \in \{0, 1\}$. The total number of possible assignments corresponds to the total number of possible binary matrixes of size K × J which is 2^{KJ}, therefore the space of all possible assignments has size:

$$S = 2^{KJ} \tag{5}$$

If we assume that the origin server is part of the set of cache servers J, each content must be hosted at least by one cache server, in order for the content to be retrievable.

Therefore, the set of valid assignments $Sv \subset S$ does not contain assignments that do not assign every content k at least once. In other words, valid assignments have at least K "1" values and, among those with at least K "1" values, only those that assign at least one cache to each content are valid assignments. Without determining the exact size of the space Sv, we can say that:

$$Sv < 2^{KJ} - \left(\sum_{n=1}^{K-2} \sum_{i=1}^{K-n} i + KJ + 1 \right) \tag{6}$$

where the second term is the number of assignments with less than K "1" values. For example let's consider K = J = 2. While S is given by 16 matrices, Sv contains the 9 matrices below:

$$\begin{pmatrix}1&1\\0&0\end{pmatrix}\begin{pmatrix}0&0\\1&1\end{pmatrix}\begin{pmatrix}1&0\\0&1\end{pmatrix}\begin{pmatrix}0&1\\1&0\end{pmatrix}$$

$$\begin{pmatrix}1&1\\1&0\end{pmatrix}\begin{pmatrix}1&1\\0&1\end{pmatrix}\begin{pmatrix}1&0\\1&1\end{pmatrix}\begin{pmatrix}0&1\\1&1\end{pmatrix}$$

$$\begin{pmatrix}1&1\\1&1\end{pmatrix}$$

The set of invalid matrices S_{inv} contains the 5 matrices with less than 2 "1" values and 2 additional matrices with 2 "1" values but with one content not assigned to any cache server:

$$\begin{pmatrix}0&0\\0&0\end{pmatrix}$$

$$\begin{pmatrix}1&0\\0&0\end{pmatrix}\begin{pmatrix}0&0\\0&1\end{pmatrix}\begin{pmatrix}0&0\\1&0\end{pmatrix}\begin{pmatrix}0&1\\0&0\end{pmatrix}$$

$$\begin{pmatrix}1&0\\1&0\end{pmatrix}\begin{pmatrix}0&1\\0&1\end{pmatrix}$$

According to Equation (6):

$$Sv < 2^4 - (4 + 1) = 11 \tag{7}$$

In order to simplify the problem and formulate the cost function (1) as a function of an assignment X we introduce the following assumptions:

- Each network path between an aggregation point i and a cache server j is associated to a normalized path cost $c_{i,j}$ which is provided by the network operator.

- The origin server is associated to the highest path cost.
- The content popularity distribution, including the number of requests for the k content NR(k) is known for each k.
- The content popularity distribution z_k is Zipfian.

Based on the above assumptions, $C_{NW}(X)$ can be expressed as:

$$C_{NW}(X) = c^{u}_{NW} \sum_{i \in I} \left(NR_i \sum_{k \in K} \sum_{j \in J} r_{i,j}(k) x_{k,j} \, s_k \, z_k \, c_{i,j} \right) \quad (8)$$

and $C_C(X)$ can be expressed as:

$$C_c(X) = c^{u}_{STO} \sum_{k \in K} \sum_{j \in J} x_{k,j} \, s_k + c^{u}_{SER} \, NR \sum_{k \in K} \frac{s_k z_k}{\sum_{j \in J} x_{k,j}} \approx c^{u}_{STO} \sum_{k \in K} \sum_{j \in J} x_{k,j} \, s_k \quad (9)$$

Where s_k is the size of the content k, $i \in I$ is the generic aggregation point, $j \in J$ is a cache and $r_{i,j}(k)$ is the number of request for the content k that flows from i to j.

The simulated content placement algorithm minimizes the cost function defined in (1); it is a greedy heuristic to obtain a network cost-driven approach. At each iteration of the algorithm, the space of all the possible assignments is reduced to satisfy a given cost constraint at that iteration. Such constraint is defined on a per-content and per-aggregation point basis.

The described heuristic is greedy because it tries to minimize the storage cost and is network cost-driven because the solution space is reduced by imposing a budget distribution and a constraint on the network cost.

4.5. Evaluation

4.5.1. Simulation Framework

OMNeT++ is a modular, component-based simulation library and framework for event driven simulations. Smallest components specify the fine granular simulations and are implemented in C++ programming language. Simple modules can be combined to compound modules to build network nodes and complete network topologies, which are defined in the Network Description NED language. The OMNeT++ simulator has been chosen because it provides a modular and flexible framework as well as a sufficiently large set of functionalities in its library to achieve fast results. It avoids complexity which is not required at this time of the evaluation tasks in the CNSI++ project, such as mobility and Internet Protocol layer support. The simulation framework comprises the simulation kernel library, a compiler for the NED topology description language, and IDE based on Eclipse, various tools to process simulation results and good documentation.

The reference topology is representative of a mobile network of an operator that works on a national level with caches co-located with the Packet Gateway.

In order to keep the simulation time within a range that allowed handy operation, the simulation parameter are the following:

- Number of requests per hour: 10.000

- Number of contents: 5.000
- Duration of content: 1-3 mins
- Prefix Duration: 30 sec
- Chunk Size 64 KB
- SD format fmt 34, 640x480, 800 Kbps
- HD format fmt 22, 720p, 2300 Kbps
- HD contents ratio 50%
- Dynamic Model Popularity: the popularity of each content evolves over time

In the simulation study, both the network-aware placement and retrieval and the partial caching techniques have been considered. The following parameters have been taken into account:

- Byte Hit Ratio (BHR) per cache: the ratio between the number of cached bytes over the number of requested bytes calculated on each cache.
- Overall Byte Hit Ratio (BHR): the overall BHR on all the caches.
- Per-session Max Chunk Delay: the maximum end-to-end chunk latency among all chunks within one single session. One session is the transmission of an entire content.
- Aggregated Network Cost: the cost accumulated by one chunk during its delivery from the source (origin or cache server) to the destination (terminal). This represents the cost of the delivery of one chunk. It is calculated by adding the per-chunk cost of each link that is traversed by the chunk. The average chunk cost throughout the whole simulation is then calculated and displayed.

It is important to note that the network cost due to the ingestion of content from the origin server to the cache is accounted for.

4.5.2. Simulations Results

4.5.2.1 Network-aware Content Surrogate Selection

Instead of a random cache selection, the ALTO guidance can be used. When ALTO guidance is used, the Tracker accesses the network topology (OMNeT++ library functions) and computes the cost of the delivery from each candidate cache server. To compute this cost, the shortest path algorithm between the aggregation point and the candidate cache server is used.

The simulation results presented in this section aim at quantifying the benefits of ALTO-based content caching solutions in terms of user experience.

The following performance parameters are taken into account:

- Per-session Max Chunk Delay: the maximum end-to-end chunk latency among all chunks within one single session. One session is the transmission of an entire content.
- Number of Requests: the total number of requests per server in the entire simulation.

The input parameters for the obtained results are the following:

- Cache Selection (c): indicates how the cache server is selected among the cache servers containing the desired content. The selection can be performed network-unaware (c=1) or with ALTO (c=2).

- Cached Content Ratio (ccr): indicates the fraction of the total number of contents that are cached in the cache servers. how the cache server is selected among the cache servers containing the desired content. It can be 20% (ccr=0.2) or 40% (ccr=0.4).
- Replication Level (r): a factor between 0 and 1 that indicates the level of replication with which contents should be cached. With r=0 one content is stored in one single cache server. With r=1 each content is stored in each cache server. With intermediate values, the more popular content is replicated more than the less popular content.
- Load Balancing (lb): when random cache selection is used, if load balancing is enabled a random cache among those with the current lowest number of served requests is selected. If ALTO is used, when load balancing is enabled a cache with the lowest number of served requests is selected among those with the lowest cost.
- Source Routing Strategy (sr): Source routes are used to control routing of packets via gateways, even though a destination may be reached on a shorter path.

Table 4 resumes the default parameters for the simulation study.

Parameter Name	Default Value
Cache Selection (c)	1
Replication level (r)	0.6
Cached content ratio (ccr)	0.2
Load balancing (lb)	false
Source routes (sr)	1

Table 4: Default Simulation Parameters

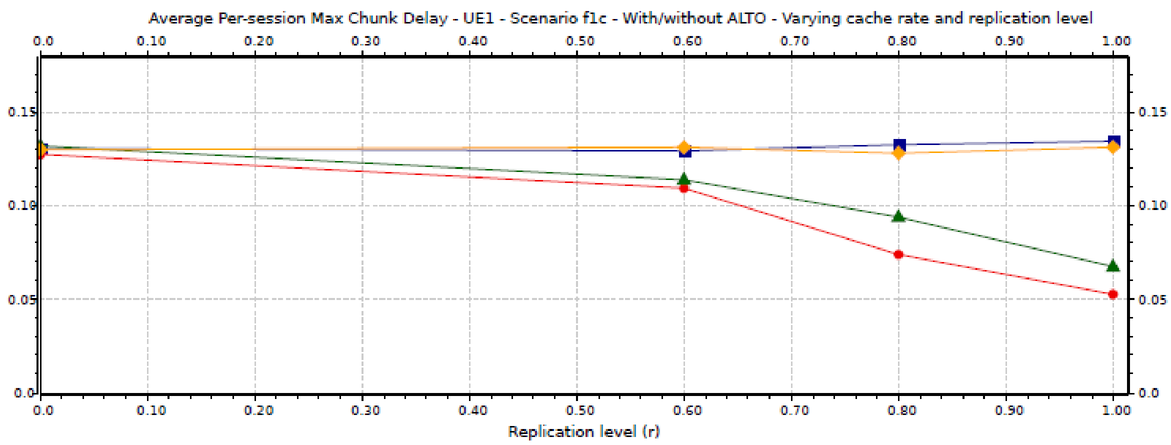

Figure 28: Average per session Max Chunk Delay Delay, network-unaware vs ALTO as a function of replication level and varying cached content ratio

Figure 28 depicts the average per-session maximum chunk delay with random cache selection (blue and yellow) and with ALTO (green and red) varying the replication level (x axis) and the cached content ratio (ccr=0.2 for blue and green lines, ccr=0.4 for yellow and red lines).
It can be observed that increasing the amount of cached content is not as beneficial as increasing the number of replicas for the more popular content. In particular, ccr=0.4 requires double storage

space both brings at most a 30% reduction of the delay (for r=0.8 and ALTO). Instead, increasing r from 0.6 to 0.8 brings almost 60% reduction of delay with the same increase of required storage size Figure 29 depicts the cumulative probability function of the per-session maximum chunk delay with random cache selection (red) and ALTO-based cache selection (blue).

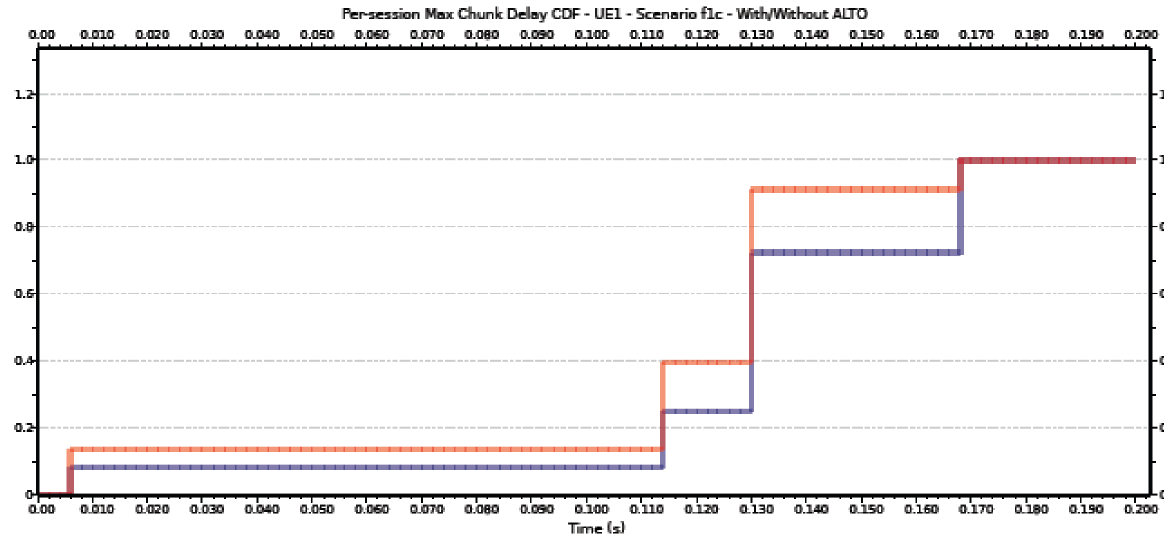

Figure 29: CDF of Per-session Maximum Chunk Delay terminal 1, random vs ALTO with default values for other parameters

Figure 30 depicts the number of requests per cache server with ALTO-based cache selection and without load balancing for different values of the replication level: r=0 (blue) r=0.6 (yellow) r=0.8 (green) r=1 (red). Figure 31 shows the same indicator but with load balancing and a different color map. It can be observed that load balancing is very rarely used with ALTO, due to the low number of cache servers in the considered scenarios.

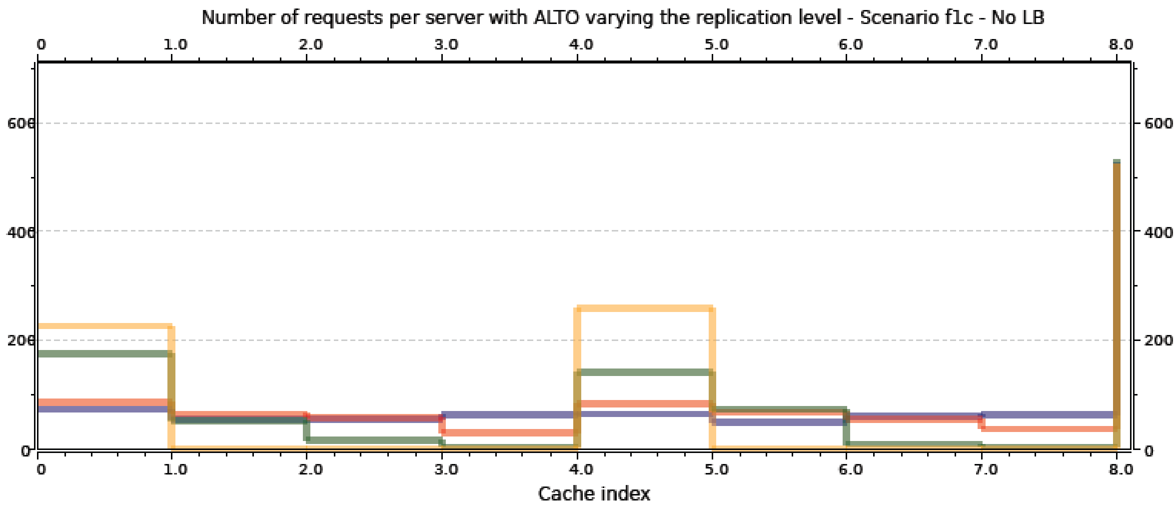

Figure 30: Number of requests per server with ALTO cache selection and without load balancing for different values of replication level

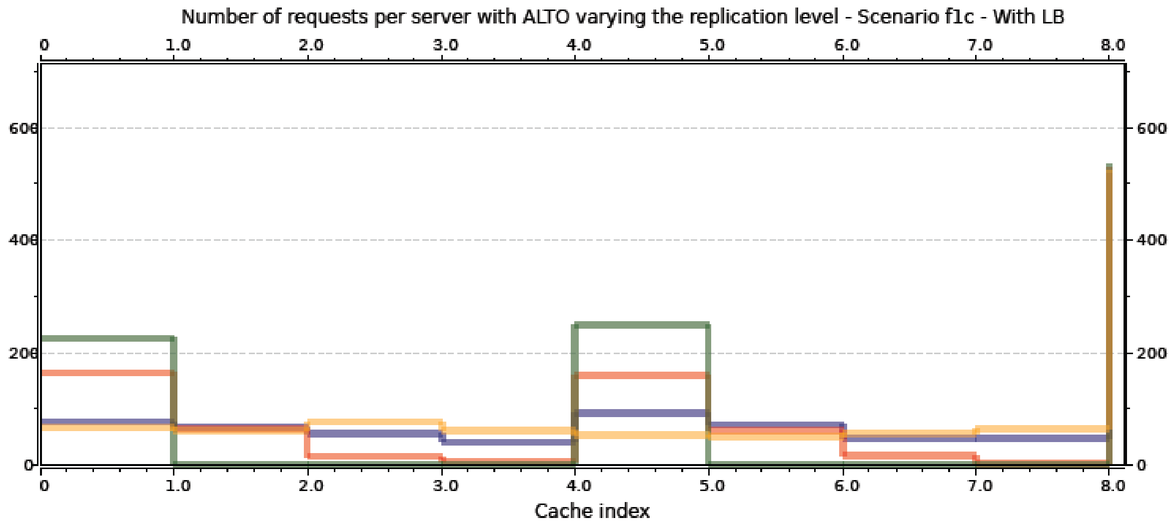

Figure 31: Number of requests per server with ALTO cache selection and with load balancing for different values of replication level

4.5.2.2 Network-aware Content Placement

The second simulation study is based on the performance of the network-aware placement algorithm.

The following strategies will be compared:

- Network-driven heuristic (RPA algorithm): it is the heuristic obtained by the problem of replica placement defined as finding the assignment X of contents k to cache servers j such that the sum of network costs and caching costs is minimized. In other words, it is a cost aware placement algorithm.
- Network-unaware algorithm: this algorithm simply places a number of replicas of a certain content that is proportional to the popularity of the content. In other words, it is popularity-aware but not network-aware. Replicas are distributed on randomly chosen cache servers.

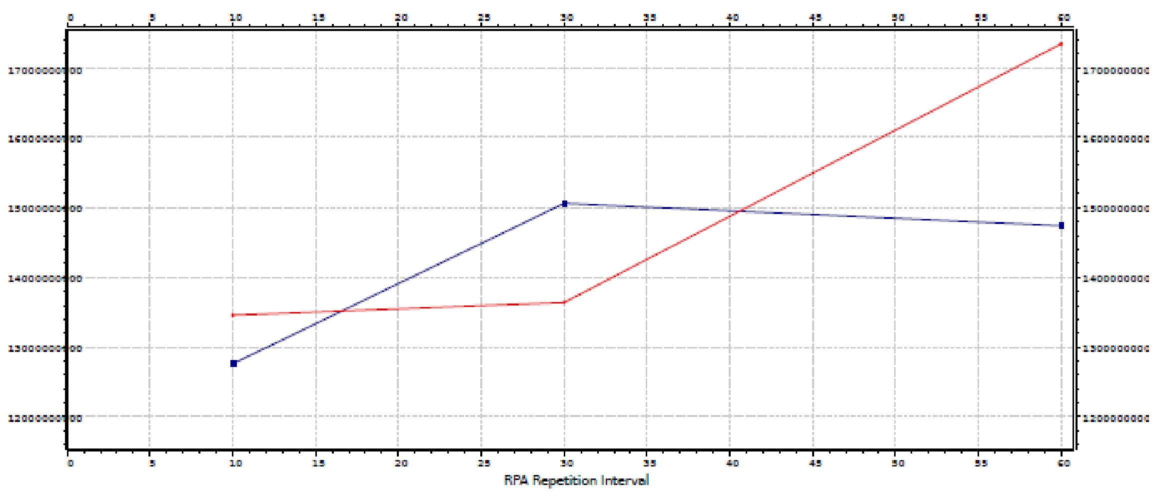

Figure 32: Total Aggregated Network Cost for Network-driven heuristic (blue) and the Network-unaware algorithm (red) varying the placement repetition interval

Figure 32 depicts the total aggregated network cost for the Network-driven heuristic (blue) and the Network-unaware algorithm (red) for different values of placement repetition interval within the 1h-long simulation.
The following observations can be made:

- Given the quickly varying popularity model, the Network-driven heuristic performs well only when executed frequently. It should be noted that the popularity model is way too dynamic as compared to reality (all 5000 contents experience the full Gaussian popularity curve within the 1h-simulation time).
- For repetition interval equal to 10 minutes, the Network cost-drive algorithm reduces network cost by 6% as compared to the popularity-aware Network-unaware algorithm

Figure 33 and Figure 34 depict the average per-session chunk delay experienced by each terminal aggregator for the Network-unaware Algorithm and the Network cost-driven Algorithm respectively. The network cost-driven algorithm reduces the average latency.
However, it should be noted that the network simulated is over-provisioned, i.e. no congestion occurs. In reality, bigger differences in the latency are expected.

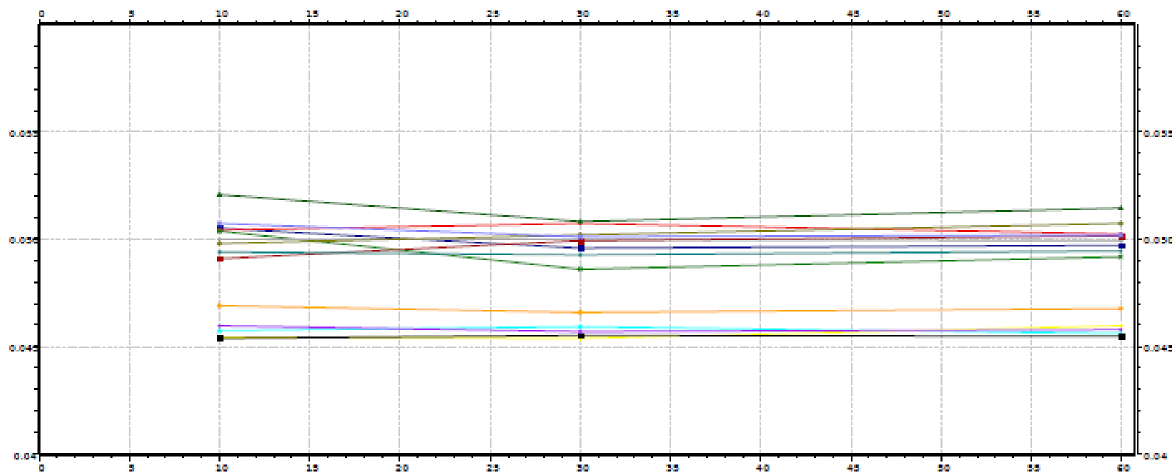

Figure 33: Per-session Average Chunk Delay with Network-unaware Algorithm

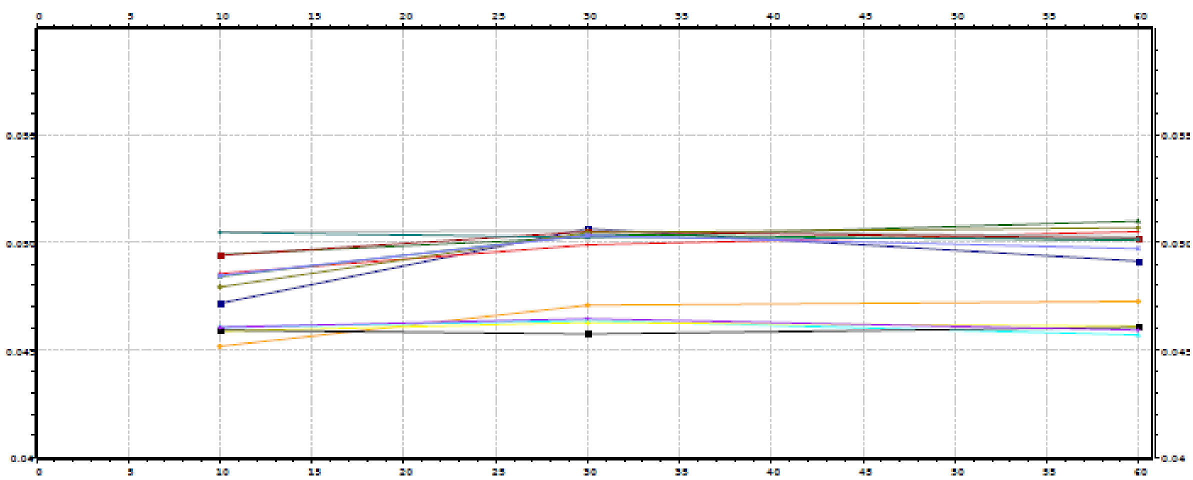

Figure 34: Per-session Average Chunk Delay with Network cost-driven Algorithm

4.5.2.3 Network-aware Chunk-level Placement

The last simulation study is focused on the partial caching techniques.

The strategies that will be compared are the following:

- A : all the contents are fully cached;
- C : there are no contents initially cached, contents are cached on demand;
- A/C : a percentage l1% of the total number of contents is initially fully cached, the remainder contents are cached on demand;
- B_{push} : the contents are initially partially cached (i.e. the prefix of each content is pre-fetched), suffixes are cached following a threshold based mechanism to determine if to cache a suffix or not;
- B_{pull} : there are no contents initially cached, prefixes are cached on demand while suffixes are cached following a threshold based mechanism to determine if to cache a suffix or not;
- A/B_{push} : a percentage l1% of the total number of contents is initially cached, the remainder contents are initially partially cached (i.e. the prefix of each content is pre-fetched), suffixes are cached following a threshold based mechanism to determine if to cache a suffix or not;
- A/B_{pull} : there is a percentage l1% of the total number of contents initially cached, the remainder contents are partially cached on demand, while suffixes are cached following a threshold based mechanism to determine if to cache a suffix or not;

It is worth mentioning that the prefix placement algorithm is random while the whole contents placement for strategy A considers the Network-driven heuristic (RPA algorithm); the repetition interval is equal to the simulation duration (i.e. the placement is performed only once at the beginning) if the contrary is not explicitly told.

The statistics are collected each minute, neglecting the first 5 minutes of simulation.

It is worth mentioning that the maximum storage size is set using the output value obtained by fully caching all the contents using strategy A. Then the maximum storage is partitioned to assign the storage size to each cache. It guarantees that the caches always work in a situation where the storage resources are fully utilized and the LRU mechanism will activate.

When the partial caching strategies are active, the total storage assigned to the cache is equally divided among the prefix and the suffix stack.

Figure 35 shows the obtained performance when the dynamic popularity model is used.
The legend is the following:

- blue square: C
- red circle: A
- green triangle: A/C
- yellow diamond: Bpush
- green cross: Bpull
- gray plus: A/Bpush
- red square: A/Bpull

As expected, the BHR when all the contents are fully cached reach the maximum value of 1 that means that for each request there is a cache hit. This result could be taken for granted but it is useful to validate our simulation model.

On the contrary, if there are no pre-fetched contents (i.e. strategy C), the BHR is initially zero but increases to a final value of 0.6 because contents are requested and then entirely cached on demand.

If the contents are partially cached on demand (i.e. strategy B_{pull}), the final BHR value is less than 0.3. This final value is highly influenced by the threshold chosen to determine whether or not to cache the suffix.

To tune this parameter, an empirical analysis has been followed; the value has been set to 10. It means that if the content to cache has been requested more that 10 times, then both prefix and suffix are cached. If the threshold is lower, more contents are fully cached and the BHR tends to the observed result of the C strategy.

Strategies A/C and A/B_{pull} show an initial increasing of the BHR value if compared with C and B_{pull} due to the initial pre-fetching of the l1% of the total contents.

The value of l1% has been fixed at 0.2 to focus on partial caching; higher value of such parameter leads to higher initial BHR.

It is possible to observe that the strategies A/B_{pull} and A/B_{push} have a worsening of performance. It shows that the LRU replacement mechanism is not able to cope with a high popularity change and a phenomenon of cache pollution occurs; replacement techniques based on popularity estimation need to be investigated.

The worsening in performance of strategies A/B_{push} and A/B_{pull} and the cache pollution effect is also due to the fact that RPA algorithm has been run only once and was based on the popularity of contents at the very beginning of the simulation. Given that the popularity varies over time, the contents fully cached become stale and waste cache resource decreasing the overall available resource storage for partial stacks. To reduce such effect another simulation study has been carried out.

The dynamic model of popularity has been chosen and the RPA algorithm has been performed periodically, each 10 minutes. It means that each 10 minutes, the popularity of contents is evaluated and then the l1% of the most popular contents are fully cached (removing copies from prefix/suffix stacks if any) to reduce the number of stale contents and increasing the storage resource in prefix/suffix stacks. Figure 36 shows that the performance of A/C, A/B_{push} and A/B_{pull} strategies improve. The BHR of A/C strategy improves from 0.7 to near 0.8 while the improvement is much more evident for strategies A/B_{push} and A/B_{pull}. The legend for Figure 36 is the following:

- blue square: C
- red circle: Bpull
- green triangle: Bpush
- yellow diamond: A/C
- green cross: A/Bpull
- gray plus: A/Bpush

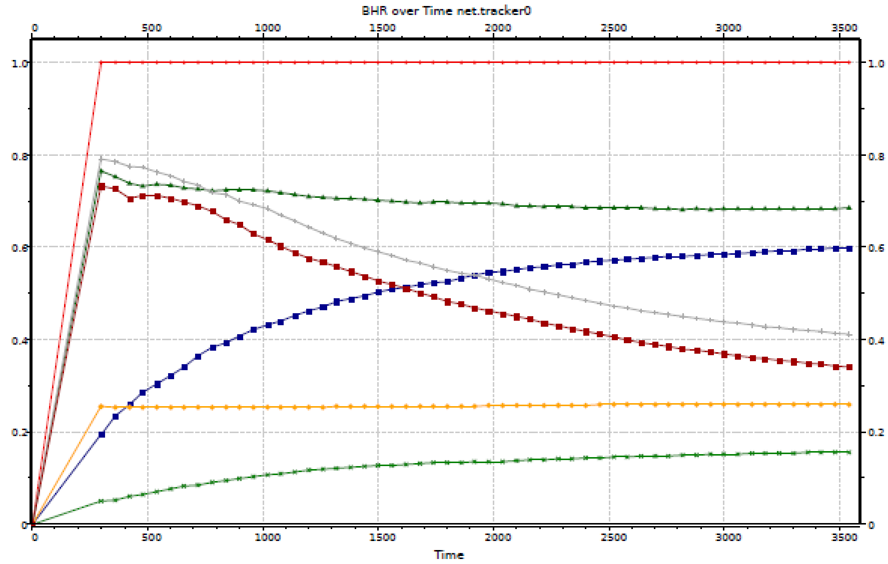

Figure 35: Overall BHR without periodic RPA

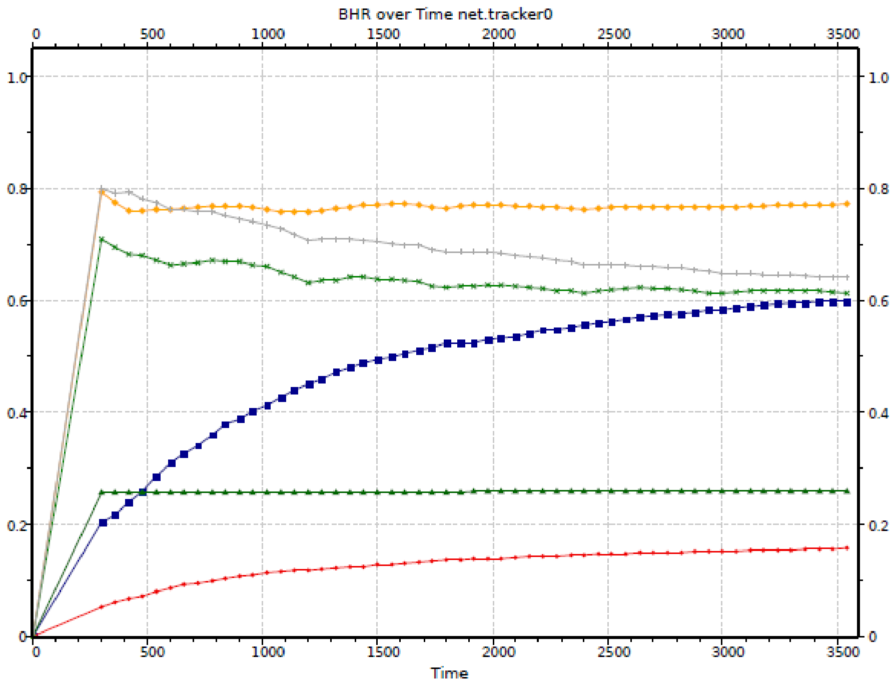

Figure 36: overall BHR when periodic RPA is performed

Taking into account the A/B_{push} mechanism with RPA without repetition, the Figure 37 - Figure 39 show the BHR at three caches. It shows that that there are some full hit but most of the hit are related to partial cached contents.

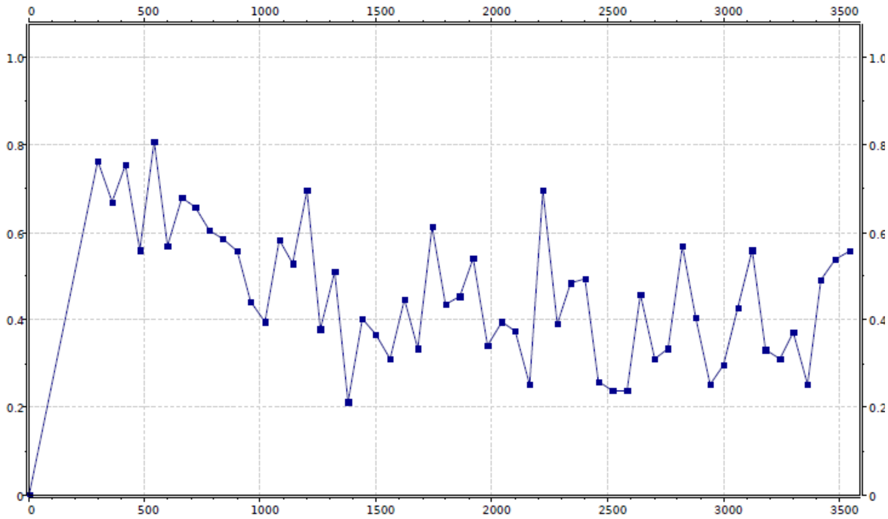

Figure 37: BHR at cache 1

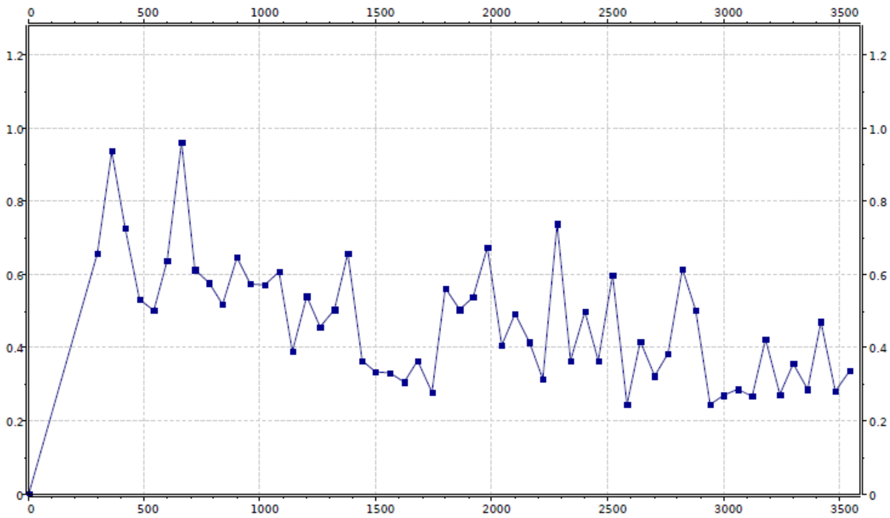

Figure 38: BHR at cache 2

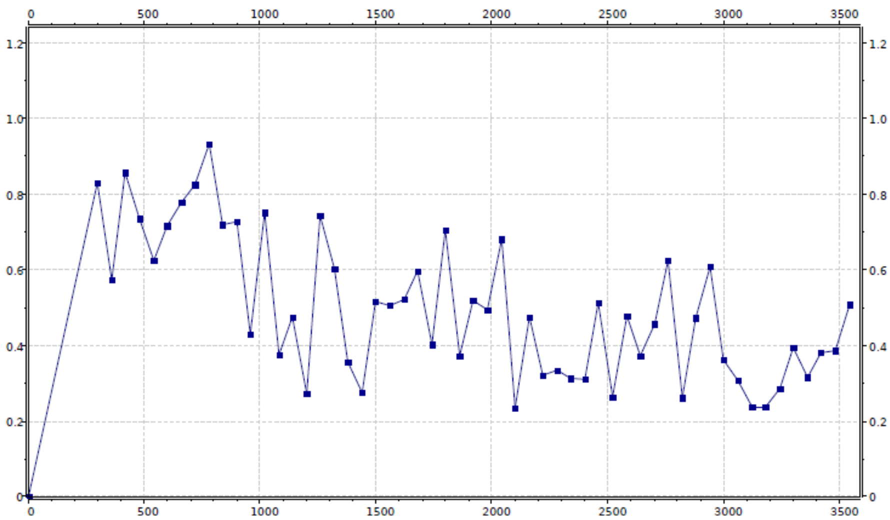

Figure 39: BHR at cache 3

4.6. Conclusions and Future Work

This section has shown, by means of simulations, the potential improvements of adopting a network-aware strategy for content distribution. The simulations have shown how both monetary costs and network utilization can be decreased while improving user experience by selecting the "best" replica and adopting a network-aware content placement strategy. Future work will be the extensions of simulations in order to cover massive scenarios (e.g., a Tier-1 operator network with millions of users) and the improvements of content placement strategies based on probabilistic meta-heuristics.

5. Tools for Available Bandwidth Estimation

5.1. Introduction and Problem Description

The COAST architecture aims to deliver content in an efficient and network-friendly way, while providing improved services, such as better QoS, faster response to a request, and better streaming quality (due to adaptation). A fundamental requirement to provide the data in such a manner is the network awareness, which partially concerns the quality of the involved links. Examples for network-aware content retrieval can concern

- the selection of a most suitable server or cache for the content delivery,
- the placement of content among the set of available servers (regarding potential user needs),
- the selection of a set of servers each of them providing a part of the content (P2P),
- and the distribution of different versions of the content on selected servers, such as the assignment of scalable video layers to servers with appropriate connectivity.

The COAST Network Monitor component (introduced in [18]) establishes the COAST core functionality “CF1.4: Network traffic discovery” and thus provides information on the available bandwidth of the links. We define the available bandwidth of a path as the unused bandwidth on a tight link. In this section, an analysis on available bandwidth will be performed over a set of PlanetLab servers to answer the questions

- of the feasibility of active bandwidth measurements within the COAST envisioned content-centric network overlay architecture, specifically, regarding the selection of appropriate tools and the requirements on the content servers,
- of the meaning, reliability and use of the obtained results of such measurements,
- of the network intrusiveness, specifically the amount of traffic generated by the tools,
- of the time needed to obtain a result.

In Section 5.2 the considered techniques and tools are shortly reviewed and a suitable tool is selected for the evaluation of its applicability in the COAST architecture. Section 5.3 explains the employed PlanetLab setup and the methodology for the conducted measurements. Results and the main findings are given in Section 5.4 and 5.5, respectively.

5.2. Consideration of Available Tools for the COAST Network Monitor

Available bandwidth of a path concerns the unused bandwidth on a tight link along that path, and can be estimated using passive and active estimation techniques. While the former ones do only inspect the ongoing traffic for their estimation, the latter ones send probing packets and measure their arrival time. The active techniques allow for a point in time bandwidth estimation, as the measurements can be triggered when required. The passive techniques however only give results on past traffic obtained from services. A basic requirement for these techniques is thus that data traffic has to be present which flows through the link of interest and can be inspected by the employed method. Ideally, the passive estimation technique has information on the timing of the data flow and all competing connections, thereby being able to give estimates within a limited time.

The COAST network monitor does not have this information, it further needs to estimate bandwidth upon request and independently of ongoing traffic, which in case of HTTP streaming data is controlled by congestion algorithms that make the estimation difficult. We therefore here consider only active estimation techniques.

The active tools send probe packets to measure the end-to-end packet delay, which is dependent on the active cross traffic. Generally, these tools require installation of the algorithmic on receiver and sender side of the nodes. In the context of their applicability in the PlanetLab Architecture, we have been considering and testing the bandwidth estimation tools

- pathChar [11],
- Pathrate [16],
- capprobe [17],
- Spruce [14],
- IGI [13],
- BART [15],
- Pathload [9],
- pathChirp [10],
- abget [12], and
- Assolo [8].

The pathChar, Pathrate, and capprobe do estimate available path capacity – not available bandwidth. The available capacity of a path refers to the available bandwidth of that path in case no cross traffic exists – that is, the maximum throughput that a flow can get in the path from sender to receiver. While available capacity can be a part of the available bandwidth estimation, we here do only evaluate self-contained tools for available bandwidth estimation in order to alleviate the possible integration into the COAST architecture. Similarly, the Spruce tool does not allow for a self-contained measure, as it requires for its bandwidth calculation the knowledge of path capacity, which is not estimated by the tool.

The IGI tool gave not give consistent results in our PlanetLab setup, and therefore we did not look further into it. For the BART method we could not find a publicly available implementation.

The tools Pathload, pathChirp, abget, and Assolo are all based on the same concept of self-induced congestion by sending packet trains, while the Assolo (Available-bandwidth Smart Sampling On-Line) tool is a more recent development. Compared to previous tools Assolo features a new probing traffic profile called REACH (Reflected Exponential Chirp), which tests a wide range of rates and thus provides more accurate results. The REACH packet stream increases the instantaneous packet rates from a lower bound to a maximum rate, while the probing rate does not increase linearly. In our primary PlanetLab setup the tool gave more stable results in a shorter time and is therefore used in the following evaluation.

5.3. PlanetLab Setup for Available Bandwidth Estimation

PlanetLab is a free platform for research on computer networking, content distribution, or distributed systems. It consists of 913 nodes at 460 sites worldwide. Researchers can register for a *slice* on the PlanetLab platform. A slice is a set of PlanetLab nodes, on which the user has access to a private virtual server (running on the respective node) via UNIX shell.

For the bandwidth measurements, a slice of nine servers has been created:

Server name	Operator	Short name
1. csplanetlab2.kaist.ac.kr	Korea Advanced Institute of Sciencee and Technology	kaist
2. gschembra4.diit.unict.it	University of Catania – Computer Science and Telecommunications Engineering	unict
3. host3-plb.loria.fr	INRIA Nancy – Grand Est	loria
4. dannan.disy.inf.uni-konstanz.de	University of Konstanz	konstanz
5. planet01.hhi.fraunhofer.de	Fraunhofer-Institute for Telecommunications – Heinrich-Hertz-Institut	fraunhofer
6. planetlab1.poly.edu	Polytechnic University	poly
7. planetlab2.ls.fi.upm.es	Universisad Politecnica de Madrid	upm
8. planetlab2.upc.es	Universitat Politecnica de Catalunya	upc
9. ait05.us.es	University of Sevilla	us

Table 5: Servers for bandwidth measurements

The servers have been selected due to provision of open ports required for the measurements. On each of the servers, programs can be executed using an ssh-command with the program name as an option.

The measurement methodology takes care that the bandwidth between each possible pair of the nine servers is estimated. For each node,

- the available bandwidth to each of the eight remaining nodes is estimated using the Assolo tool, and
- second the time to transfer a file (10 MByte, 100 kiloByte, 10 kiloByte) to each of the remaining nodes is measured.

The available bandwidth measurement is initiated before each file transfer and can take a few seconds. To perform the available bandwidth measurement between two servers, on both of the servers a daemon program has to be active. These programs are the *assolo_snd* to be started on the sender and *assolo_rcv* to be started on the receiver. The bandwidth measurement is triggered by an *assolo_run* command which takes the addresses of sender and receiver server as an option. It can be executed on a third server that controls the measurement series.

To estimate the transfer time the *netcat* (nc) tool is started on the sender side to listen on a given port. Netcat can read and write from network connections using TCP or UDP. The receiver node triggers the file transfer by setting up a TCP connection to the sender and writing the data into a file while the file transfer time is measured.

5.4. Results

The following figures give the available bandwidth measurements compared to the transfer rate in Mega Bit per second (MBit/s). The transfer rate is deduced from the file size in MBits divided by the transfer time in seconds . Available bandwidth is given on the left-hand y-scale while the transfer rate is given on the right-hand y-scale. For each plot there are one source node and eight destination nodes – that means, for the bandwidth estimation probe packets are sent from the source node to the destination node and similarly for the file transfer the file is sent to the destination node. File sizes of 10 MByte, 100 kiloByte, and 10 kiloByte are considered, and for each of the file sizes a number of four to five selected plots are given. Note that in order to facilitate a comparison, each plot has a x-axis with nine nodes including the plot's sender node (which results a transfer time of zero seconds).

5.4.1. 10 MByte file transfer

Figure 40, Figure 41, and Figure 42 give the available bandwidth and 10 MByte transfer rate measured for the loria and the konstanz node, for the upm and the upc node, and for the us node, respectively. The available bandwidth and data rates are highly divers. The available bandwidth is generally overestimated regarding the total file transfer time which gives the average file transfer bit rate. The available bandwidth and the transfer rate are generally correlated.

We analyze available bandwidth and average file transfer rate for the unict receiver node. In case of the loria sender node (Figure 40, left-hand plot) the available bandwidth (indicated in the plot with the –bw suffix) is estimated as 60 MBit/s, while the transfer rate is 9.2 MBit/s. For the konstanz sending node the available bandwidth is 41 MBit/s, and the sending rate is 9.1 MBit/s. For sending data from the upm node to the unict node (Figure 41, left-hand plot), available bandwidth is 50 MBit/s and sending rate is 15 MBit/s. For the upc sending node (Figure 41, right-hand plot) and the us sending node (Figure 42), the available bandwidth and sending rate are 42 and 6.7 MBit/s, and 55 and 9.5 MBit/s, respectively. The resulting over-estimation factors for the 5 sending nodes are 60/9.2=6.5, 41/9.1=4.5, 50/15=3.33, 42/6.7=6.23 , and 55/9.5=5.79.

For the fraunhofer node the performance is the lowest – that is, approximately 5 MBit/s for available bandwidth and in the range of 0 to 0.2 MBit/s for the transfer rate.

To some extend bandwidth and transfer rate symmetry are visible, as for instance regarding the konstanz and the upc nodes (right-hand plots in Figure 40 and Figure 41). The konstanz sending node achieves a bandwidth of 56 MBit/s, while the upc node achieves 60 MBit/s in the reverse direction. For the transfer rate the konstanz node gives 9 MBit/s for the transfer rate, while the upc node gives 8.8 MBit/s.

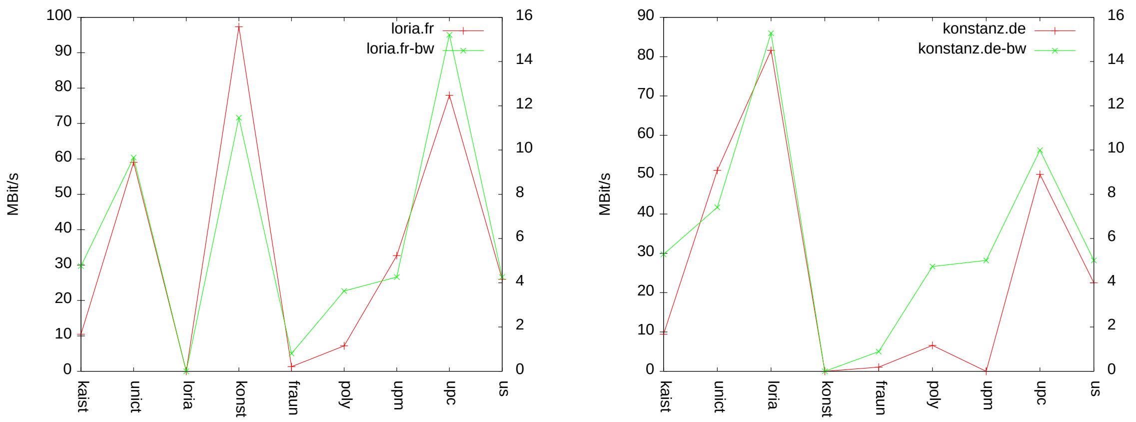

Figure 40: Available bandwidth and average data rates due to 10 MByte file transfer for loria (left-hand plot) and konstanz (right-hand plot) sender node. The left-hand scale refers to the measured available bandwidth and the curve indexed with *-bw*, while the right-hand scale refers to the average file transfer bit rate. To alleviate comparison of the plots, the sender node is included in the x-axis even if it does not account for a result.

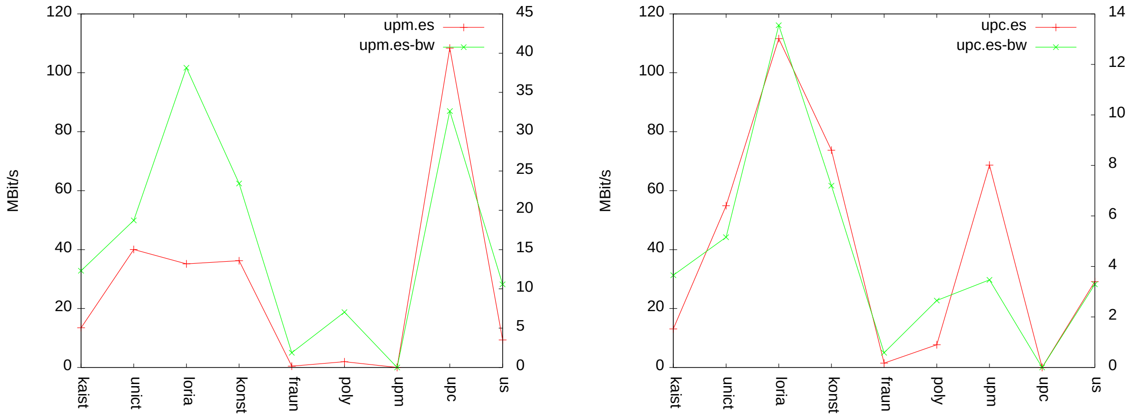

Figure 41: Available bandwidth and average data rates due to 10 MByte file transfer for the upm (left-hand plot) and the upc (right-hand plot) sender nodes.

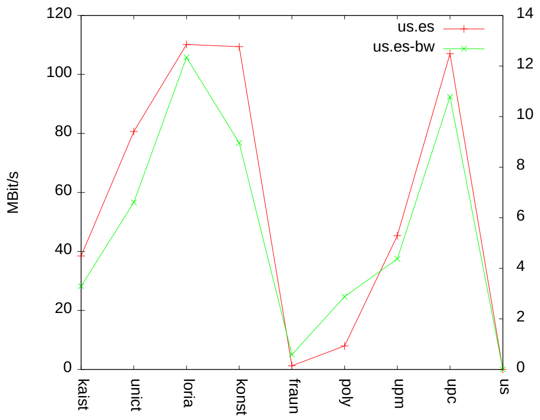

Figure 42: Available bandwidth and average data rates due to 10 MByte file transfer for the us sender node.

5.4.2. 100 kiloByte file transfer

Figure 43 shows the results for the kaist and the fraunhofer node, and Figure 44 shows the results for the poly and the us node, both Figures referring to the 100 kiloByte file transfer. Due to the smaller file size, the estimate on average file transfer rate is less meaningful, thus the correlations between available bandwidth and transfer rate are lower than for the larger file transfer of 10 MByte in the previous analysis. Due to timely decoupling, estimated bit rates appear to be much different than previously estimated. When comparing for instance the results from the measurements of the sending node us (Figure 44, right-hand plot) for 100 kiloByte file transfer and for 10 MByte file transfer (Figure 42) available bandwidth is in the range of 27 to 36 MBit/s and in the range of 5 to 102 MBit/s, respectively.

Unlike in the 10 MByte evaluation series, the fraunhofer node does not drift much compared to the other nodes. Comparing the over-estimation factors for the unict node, we get 31/2.4=12.917, 5/9=0.56, 27/12=2.25, and 29/10=2.9.

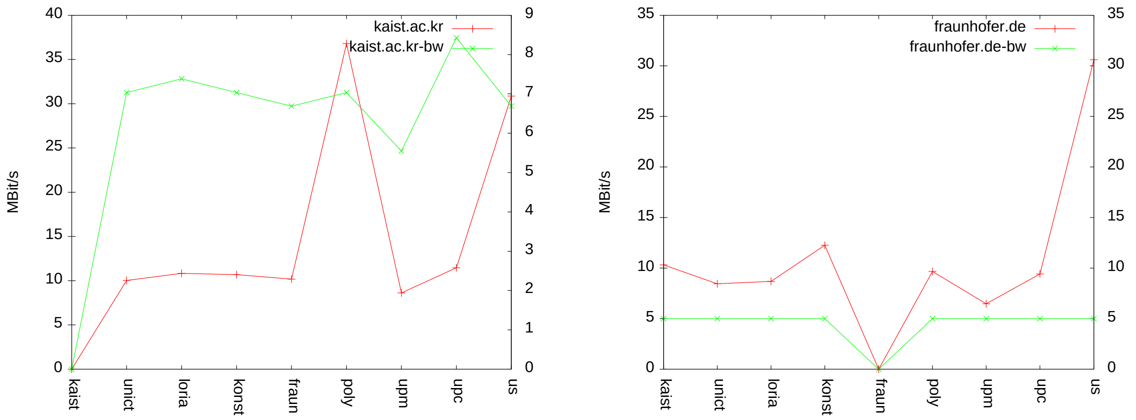

Figure 43: Available bandwidth and average data rates due to 100 kiloByte file transfer for the kaist (left-hand plot) and the fraunhofer (right-hand plot) sender nodes.

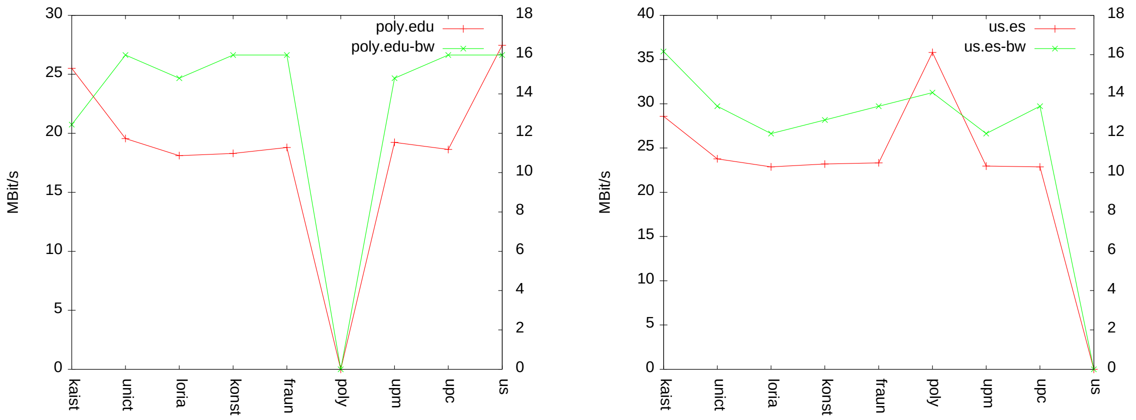

Figure 44: Available bandwidth and average data rates due to 100 kiloByte file transfer for the poly (left-hand plot) and the us (right-hand plot) sender nodes.

5.4.3. 10 kiloByte file transfer

Figure 45 and Figure 46 show the results for the 10 kiloByte file transfer for the kaist and fraunhofer nodes and the poly and us nodes, respectively. The results in Figure 46 are similar to the respective results for the 100 kiloByte file transfer evaluation. Regarding the 10 kiloByte results for the kaist sending node, the upm node failed for the file transfer, and for the nodes unict, loria, konstanz, and fraunhofer the transfer rate is less stable, specifically it is in the range of 1 to 2.3 MBits/s compared to an almost constant 2 MBit/s for the 100 kiloByte transfer. For the fraunhofer sender node the transfer rate is remarkably different for the kaist, the unict, and for the konstanz receiving node, specifically, the transfer rate for these nodes is 3 to 4.5 times smaller than for the 100 kiloByte evaluation. The less stable transfer rates can result from the smaller transfer file size.

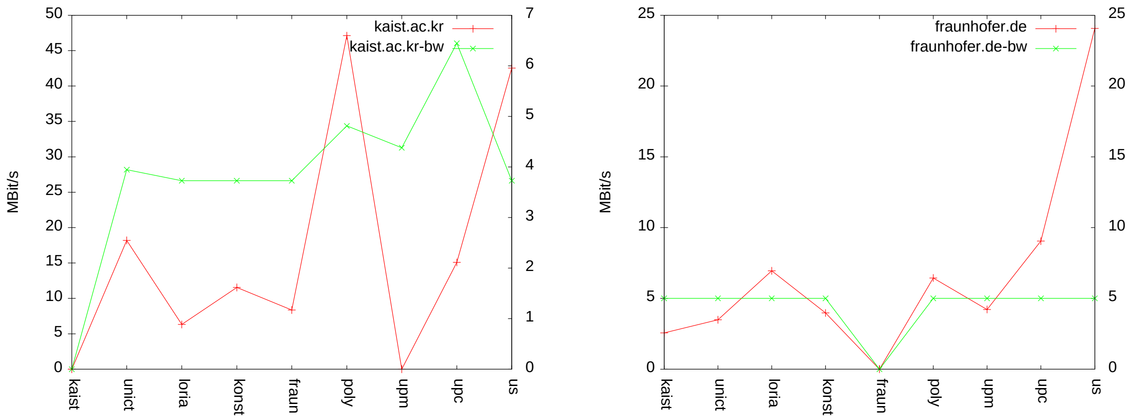

Figure 45: Available bandwidth and average data rates due to 10 kiloByte file transfer for the kaist (left-hand plot) and the fraunhofer (right-hand plot) sender nodes.

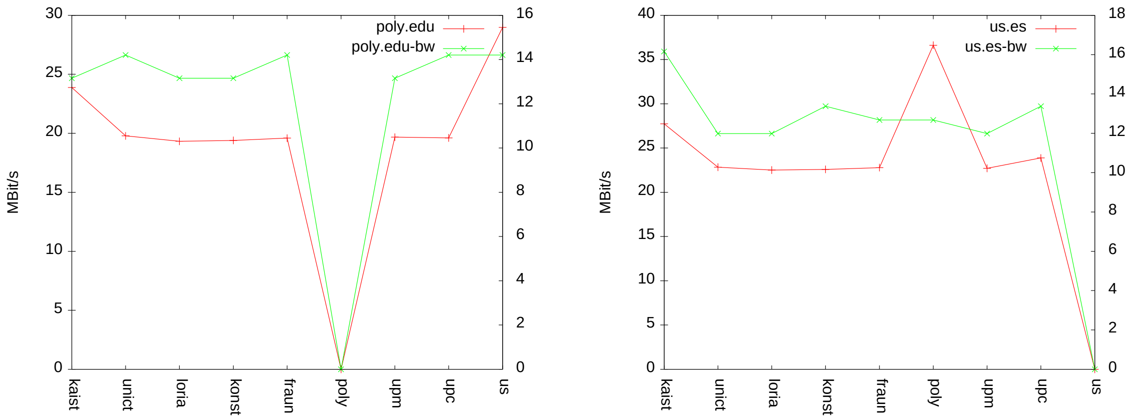

Figure 46: Available bandwidth and average data rates due to 10 kiloByte file transfer for the poly (left-hand plot) and the us (right-hand plot) sender nodes.

5.5. Conclusions

Available bandwidth measurements are an option to allow for data delivery that uses the network resources more efficiently and to make sure that the data is delivered to the user in time including improved PQoS. As described in [18], these measurements are performed within the COAST architecture supporting the core functionalities of Network traffic discovery (CF1.4), Caching (CF2), and Content and services delivery path finding (CF3.2).

The measurement technique of self-induced congestion is considered in this section as a possible candidate to provide the available bandwidth estimation. In this technique, a train of packets with increasing delays is sent from a sender to a receiver node, while the distribution of packet arrival times is analyzed. The recent tool Assolo based on the self-induced congestion is employed in the here provided measurement series, which is conducted on the PlanetLab architecture using nine servers, each of them selected to be suitable for the operation of Assolo (in the PlanetLab testbed, many nodes are configured to not allow traffic on specified ports). Software needs to be installed and started on both ends of the link to be evaluated for available bandwidth, and the measurement can be triggered from a third node.

The following main findings are concluded from the measurements:

- The estimated available bandwidth is highly diverse, specifically, it can range from 100 MBit/s to 0 MBit/s (in case of temporarily not available node). Thus the selection of a "best" server makes sense. Such a selection can for instance make sense in case of existing content duplicates on the COAST caches.
- The estimated available bandwidth does not reflect the achieved average TCP throughput due to file transfer. This is due to the design of the estimation technique, which makes the assumption that there exists only one bottleneck node in the path where the packets get stuck. While this assumption holds true in the lab environment for which the tool is verified in [8], the tool gives not accurate estimates for the more complex paths in the Internet backbone considered in this analysis. Additional discrepancy between available bandwidth estimate and average file transfer throughput results from the TCP congestion control algorithm and the timely decoupling.
- However, available bandwidth and TCP throughput do correlate, thus the bandwidth measurement can be used to find out the "best" server, to indicate weak links, or to find alternative paths in the COAST architecture.
- The available bandwidth is highly dependant on time (morning, afternoon, night, week-day etc.). Cache optimization operations – that is for instance, moving large amounts of content between the caches, can be performed in times of less network cross traffic.
- The measurements impose little network traffic. In the default case the used tool sends 18 packets of 1000 Bytes. The number of sent packets depends on the parameterized lower and upper bandwidth bounds, which gives the range within the available bandwidth is estimated (a function to automatically adjust the range if the respective available bandwidth is not found is integrated in the Assolo tool, however the function will enlarge the estimation time). As the measurement principle is however based on injecting traffic at a rate higher than the network path can process, the induced congestion can delay cross traffic that goes over the bottleneck node.
- Measurement time was longer than reported in [8], specifically it was not lower than 1 second but sometimes in the range of several seconds. This might result from temporal PlanetLab node unavailability, as one node can be shared by many users. The measurement time does not include the time for starting the daemons on the sender and receiver node.

The findings confirm the employment of available bandwidth measurements as part of the COAST Network / Traffic Monitor as envisioned in [18], supporting the modules of Distributed Caching, Content Cache Locator (CCL), and Cache Optimizer (CCO) by providing information on the quality of the involved links. The respective COAST components where the bandwidth estimation tool is integrated can be the COAST Content overlay Entry Point (CEP) and the COAST cache nodes, concerning both download or upload of content.

The probing time in the range of 1 to a few seconds has to be taken into account in case the PQoS can suffer by the introduced latency. In that case, the potential gain due to selection of a best path has to make up the induced probing latency. Proactive measurements from the CEP to a set of COAST content nodes can alleviate the latency due to online probing.

6. Content Adaptation for Scalable Video over Multiple Wireless Interfaces

6.1. Introduction and Problem Description

The emerging scalable video standard (Part 10 extensions to H.264/AVC/MPEG-4 standard) enables encoding of a high-quality video by a set of individual data flows from which a selected subset can be received to achieve temporal (frame rate), spatial (resolution), or quality scalability. The scalability feature allows to adapt the video content to the client requirements, taking into account characteristics such as terminal screen size or available bandwidth. As the mobile devices are equipped with multiple wireless interfaces, the smart utilization of available access technologies can lead to improved performance or to more efficient usage of the network resources. The problems to be solved here are the signaling of the switch and the respective rerouting of the data flows. The rerouting has to take care of the network address translation (NAT), which hides the IP address space used by the wireless interfaces. Furthermore, a switch of a flow between two wireless interfaces can induce additional delays, which have to be balanced such that the codec data frames still are received in time.

In this work we propose a method for smart utilization of multiple wireless client interfaces, which is carried out by the integration of two new components – one component for redirection of the data flows, and the other one taking care that the redirection is signaled properly upon the client's decision. The introduced solution does not require any change of the streaming server application or the client player for scalable video. In the next sections, we describe the underlying method and the resulting introduced components.

6.2. Methods, System Architecture and Implemented Components

The system architecture to be introduced here shall exploit the available wireless interfaces for scalable video streaming, thus improving the perceived QoS. As a starting point to build the system, we use the Darwin streaming server (see http://dss.macosforge.org/) and the VLC media player (available at http://www.videolan.org/vlc/). A scalable video plug-in for the VLC player (provided by our COAST partners from Fraunhofer HHI) allows for setting up a session and streaming the scalable content to the client. (A command line encoder to produce scalable video content that can be streamed from the Darwin streaming server has been provided to us as well.) A streaming session is set up using the Real Time Streaming Protocol (RTSP). It communicates to the client the available codec layers (using specific media description files) and the client can request from the available options content upon his needs. Importantly, the RTSP communication uses the TCP protocol, while the media data is streamed via UDP.

The given components, i.e., the streaming server and the extended VLC player, allow for streaming of scalable content to a client over one (wireless) interface. In order to realize scalable streaming over multiple wireless interfaces, new components have to be introduced to switch the data traffic. The intended approach switches the single but complete codec layer streams to a suitable interface, and neither modifies the given streaming server software nor the VLC media player. Two new components are introduced here, a) the *proxy component* and b) the *client module*, as illustrated in Figure 47. The proxy component reroutes incoming streaming data packets to the respective IP addresses of the wireless interfaces, or it can drop them if the client does not require them. It is located close to the server or can be part of the server. The client module is a part of the client's software, and forwards the on the wireless interfaces incoming media data packets to the respective VLC player application ports.

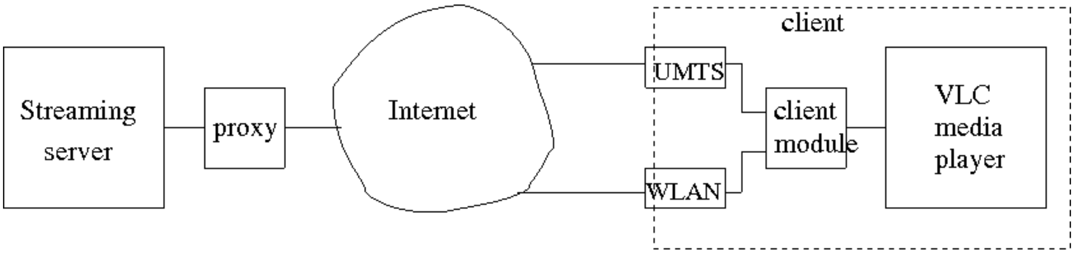

Figure 47: Overview of the main components of the scalable video streaming architecture, which exploits the two wireless client interfaces UMTS and WLAN. The newly introduced components – which are the proxy component and the client module, do make sure that the video codec layer flows are switched to the suitable wireless interfaces. The given approach does not require any modification of server or media player software.

In general, a streaming session is set up using one of the two given wireless interfaces. The client module then informs the proxy by sending control messages to take the needed rerouting actions for future incoming streaming data packets. Thereby it can receive flows of single codec layers over selected wireless interfaces.

In the following subsections, the proxy component and the client module will be described in more detail. In the last subsection the decision making module is discussed, which controls the content adaptation upon available bandwidth.

6.2.1. Proxy component

The proxy component filters all incoming and outgoing packets and

a) can modify its internal IP table rules for rerouting packets based on from the client incoming control messages,

b) can take modifications on the from the server side incoming packets' IP headers to reroute them to the appropriate interface based on its IP table settings, and

c) can drop from the server incoming packets to allow for content adaptation.

Figure 48 illustrates the functionality of the proxy, where the left-hand side refers to the incoming control messages and the right-hand side to the outgoing packets. Incoming packets are stored in a *netfilter* queue (NFQ). (The netfilter queue is a part of the *netfilter* software, which is a packet filtering framework inside the linux kernel (http://www.netfilter.org/), and employed here to intercept all incoming packets.) The proxy forwards TCP and UDP packets that refer to the streaming session setup or to other general message exchange between server and client. It however inspects the *UDP control packets*, which are introduced here to signal from the client side the required behaviour of the proxy. The UDP control packets can signal the proxy to reroute packets from a codec layer to a different client interface, i.e., it sets a route changing info. Similarly, such a route changing info can be removed. Another control message instructs the proxy to drop packets from a codec layer. In general, the signalling can happen

- when the streaming session is set up, because the client might need to be delivered the content through multiple interfaces and codec layers might need to be dropped as part of the content adaptation or
- during the streaming session when the codec layers need to be shifted / sent to different interfaces or to be dropped at the proxy.

The actions that have to be taken upon the signalling request are illustrated on the right-hand side of Figure 48. It relates to the from the server side incoming packets. TCP and UDP packets that do not refer to UDP streaming data are forwarded without modification, and the UDP streaming packets are inspected for their respective codec layer to be optionally rerouted.

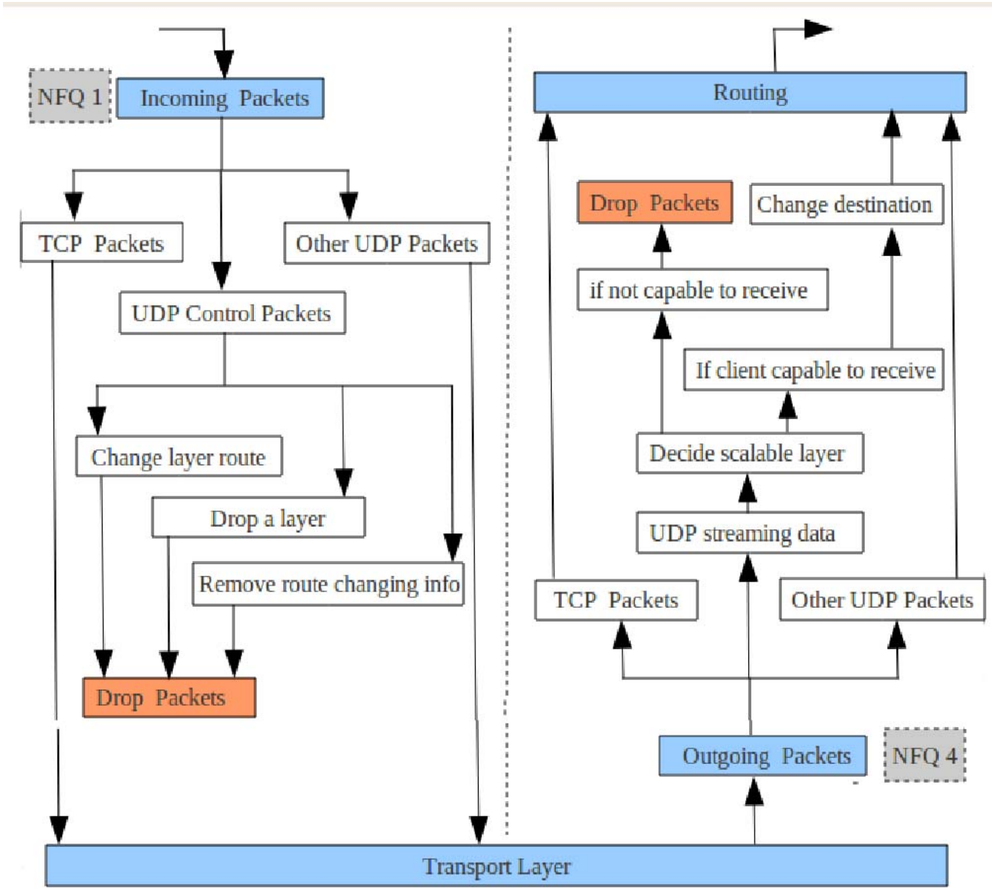

Figure 48: The proxy component intercepts all incoming (left-hand side) and outgoing (right-hand side) packets. The incoming packets are inspected for signaling content, which sets the internal routing tables. The outgoing packets are inspected for their respective codec layer and either can be dropped or rerouted to a different client interface.

The rerouting of UDP packets does not only refer to the change of IP address (and packet checksum) but includes the application port number. As the wireless interfaces in our setup (and in general) are behind a Network Address and Port Translation (NAPT) router, the proxy component needs to be informed by a signalling method about the IP address and port number which it should use for the needed flow routing. This signalling is needed due to the general RTSP protocol session setup, where for each codec flow individual application port numbers are negotiated between server and client. The negotiation is based on plain text in the TCP/IP based RTSP session setup. Packets sent to the client are generally modified by the NAPT router in that the public destination address and port number are translated on the fly to the private ones. Similarly, the NAPT translates the private address and port number from the incoming client packet to the public address and port number. To make sure that the codec flows sent by the server are received by the client with the proper application port, we use specific signalling messages sent by the client (using the negotiated client port number) from which the proxy component extracts the sender port number and IP address.

6.2.2. Client Module

The client module is part of the client's device network layer software. Similarly as the proxy component, the client module filters all incoming and outgoing packets using a separate net filter queue (NFQ) for each of the flows. While the VLC player has an internal queue which accounts for changing short-term characteristics of the link (jitter, temporal changes in available bandwidth), the here introduced additional netfilter queue for incoming packets is needed to alleviate the potential delays introduced by switching from one wireless interface to the other, while it accounts for the different characteristics of each of the wireless links. Furthermore, as the packets in the queue have different decoding timelines, the queue does not operate in a first-in-first-out (FIFO) modus but sends the packets to the transport layer in an order such that the decoder receives packets with higher priority in time.

As illustrated on the left-hand side of Figure 49, incoming UDP packets are forwarded to the transport layer without inspection.

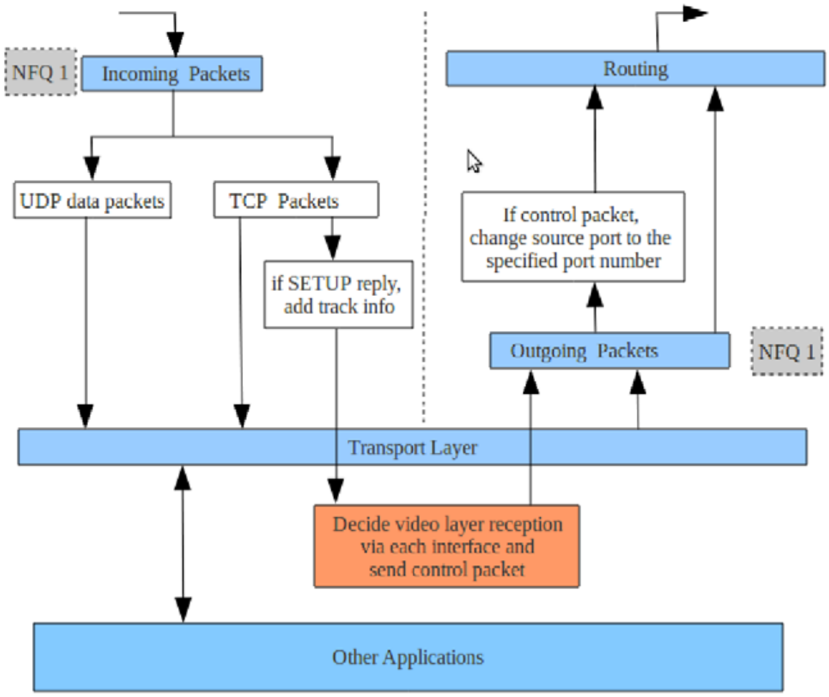

Figure 49: The client module is a part of the client side networking stack and

RTSP / TCP Packets that relate to a reply from the server due to a streaming session setup initiated by the client, are however inspected for their negotiated port numbers, which are employed for the client's UDP control messages. Specifically, the client decides about the application port numbers on which it expects the respective codec layers to be received and informs the server about it. The server then replies with its employed source ports for each of the layer flows. The source and destination port are used by the UDP control messages to appropriately signal the proxy component a switch of a layer data flow. The previous destination port is then used in the UDP control message as a source port to take into account the NAT translation from a private to a public address.

In order to enable the sending of a UDP control message from the appropriate client source port, the following workaround is employed here, as the client source port is already employed by the VLC media player: The UDP control message is sent from a dummy port number, and the IP header is modified using the netfilter tools to contain the needed source port number.

6.2.3. Decision Module (Wireless Interface Selection and Content Adaptation)

The decision module is part of the client's software and makes the decisions

a) when to use which wireless access interface (UMTS, WLAN, both),

b) which data flows, more specifically, which codec layer data to switch to which wireless interface,

c) which available enhancement codec layers to use, and

d) which options of the base layer in terms of supported quality, frame rate, and frame resolution to use.

The decision module is interfaced to the client's module, and the taken decisions trigger the appropriate signalling via own UDP control messages to the proxy. The decision strategy can be based on user context, available options for connectivity, or user terminal properties, such as user mobility and resulting availability of WLAN access points, characteristics of the wireless links (estimated available bandwidth, packet delay and jitter) or battery requirements for the single wireless interfaces. In our current implementation i) the mobility is informed to the client's module by manual user input (pressing keys on the laptop) and ii) the available bandwidth in the WLAN cell is estimated using a given tool. It is planned to refine the estimation of i) through usage of smart phone sensory, such as accelerometer and compass, and to refine the given bandwidth estimation technique to be less data intrusive. UMTS is assumed to be always available and to supply a defined minimum of available bandwidth.

For the available WLAN bandwidth estimation the so-called WBEST tool is selected, because it is evaluated in [38] to be more precise and less intrusive than related available bandwidth estimation tools. The WBEST tool uses a two-step technique to find out the available bandwidth. It first estimates the effective capacity of the wireless cell by sending a packet pair, and second it employs a packet train to estimate the achievable throughput and deduce the available bandwidth from that. An analytic model eliminates the possible measurement error sources and optimizes the measurements by finding a trade-off between accuracy, traffic intrusiveness, and total measurement time. The better performance of the WBest tool is explained in that its estimation does not rely on search algorithms but on statistics which infers the available fraction from the link capacity, thus reducing the measurement delays and random errors in the wireless channel.

In the next section the current state of the implementation of the here proposed system – including the newly introduced proxy component and client module – is evaluated for its functionality, while the intended final decision strategy will be concluded from the measurements as a part of the ongoing work of the COAST task 6.1. The measurements will employ a pre-defined script to take the decisions on interface selection and switch of codec layer data flow, and the results on available WLAN bandwidth are verified in a controlled setup.

6.3. Verification of the System Functionality

In this section we evaluate the proposed approach and the resulting implemented components for streaming of scalable video over the multiple wireless client interfaces. The streaming server is installed on a PC that is connected via Ethernet to the TU Berlin LAN, while the client device is a laptop (Intel Core2 Duo CPU T6400 with 2 GHz) equipped with an UMTS wireless card connected via USB and with an integrated WLAN interface. For the WLAN connectivity the TU Berlin campus network is utilized. The intended measurement shall evaluate the PSNR video quality for a streaming session where the UDP data flows are shifted several times from one interface to the other due to a pre-defined script. Thereby a) the system functionality is verified, specifically the redirection of the data flows to an alternative IP address, triggered by the client module and enforced by the proxy component, and b) it is evaluated if the introduced queuing techniques do account for the delays due to switching and the different link characteristics of the alternative wireless link.

Figure 50 gives the PSNR quality for the *foreman* video sequence (available at http://media.xiph.org/video/derf/) for 300 frames using 30 frames per second (fps), which gives a video duration of 10 seconds. The encoded video has two layers – one uses Quarter Common Intermediate Format (QCIF) resolution with 15 fps and the other uses a CIF resolution with 30fps. The PSNR quality is calculated comparing the decoded (not by the wireless link(s) distorted) and the received video. A switch from WLAN to UMTS and vice versa is triggered at frame numbers 40 and 180. From the plot it is visible that the switch does not result into a change in PSNR quality. Measurements for more complex sessions including the switch of single enhancement layers while using the both interfaces instantaneously gave similar results, thus demonstrating the system's ability to compensate for the packet delays due to switching.

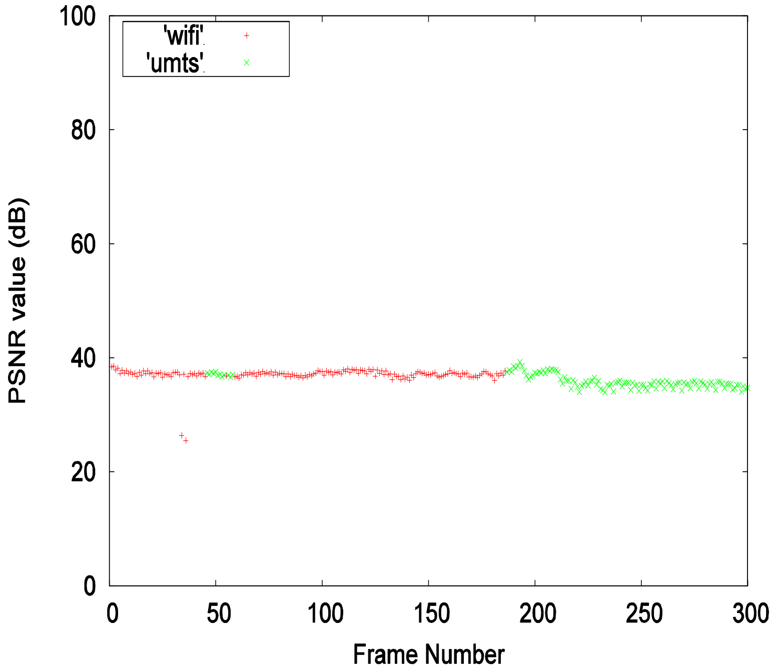

Figure 50: PSNR quality over video frame number for a video streaming session where the flows of the scalable video are switched between WLAN (in the plot indexed by 'wifi') and UMTS. The PSNR quality stays stable, thus demonstrating the proper streaming system functionality.

Figure 51 illustrates the PSNR quality for a scalable video streaming session using the *Highway* sequence (available at http://media.xiph.org/video/derf/), here with the intention to verify the proper simultaneous usage of multiple interfaces. Similarly as for the previous measurements the video is encoded with two layers, one with QCIF resolution with 15 fps and the other with CIF resolution with 30 fps. The length of the video is 66 seconds.

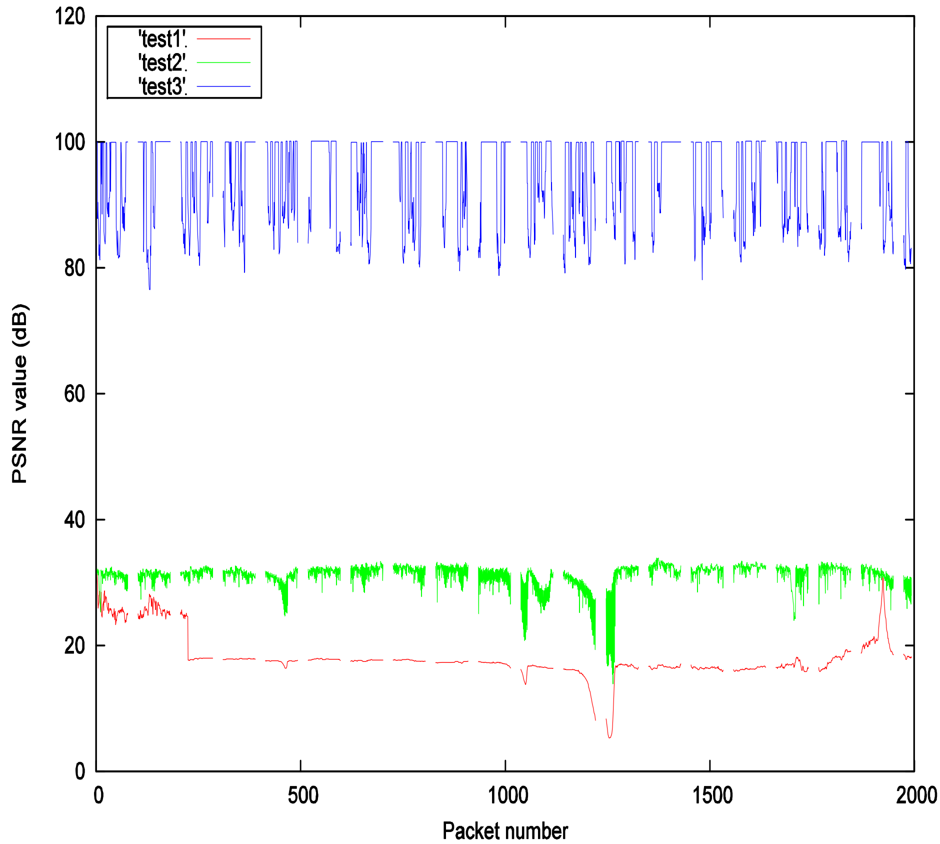

Figure 51: Comparison of PSNR quality for a video encoded with two layers for usage of WLAN only (test1), WLAN only but dropping the enhancement layer (test2), and using WLAN for the base layer and UMTS for the enhancement layer (test3). Dropping a layer and usage of multiple interfaces improve the PSNR by roughly 10 and more than 50 dB, respectively.

The figure shows the PSNR results for three tests, each test compromising a streaming session of the Highway video. In the first test base and enhancement layer are conveyed through the WLAN interface, resulting for the given setup into a PSNR quality is in the range of 20 dB. (It is part of the ongoing work to perform the measurement in a controlled environment with specified WLAN load.) The second test the content is adapted to the available WLAN bandwidth by dropping the enhancement layer, thus the PSNR quality is improved by 10 dB. The third test employs the UMTS interface to convey the enhancement layer. The resulting quality varies between 80 and 100 dB. To review the measurements we plotted one representative example image of each test in Figure 52.

Figure 52: Usage of multiple interfaces (test3) can result in superior PSNR image quality in comparison to usage of only one interface that conveys base and enhancement layer (test1) or one interface that conveys the base layer only (test2).

6.4. Conclusions

We have introduced an approach for streaming of scalable video over usage of multiple wireless client interfaces. The approach solves the problem of redirecting the video codec flows to the appropriate wireless interfaces upon the client's decision. The redirection is achieved by the introduction of a proxy component located close to the server and an additional client software module, which makes its pairs of IP address and application port number on which it expects the respective codec layers to be received public to the proxy. The signaling is achieved by sending customized UDP control messages such that the network address translation (NAT) problems due to UDP streaming to interfaces behind a NAT are solved.

Our implementation of the proposed approach is verified to switch the codec layer flows to alternative interfaces while resulting packet delays are compensated by integration of additional client queues. The PSNR results on simultaneous transmission of individual codec layers over multiple interfaces with different link characteristics further illustrate the potential gain of the here followed approach of splitting the video stream to the available wireless interfaces.

Ongoing work in Task 6.1 concerns the development of a decision method that automates the switches of codec layers due to available bandwidth and a customized available bandwidth measurement that reduces the periodic probing resource consumption in the wireless network.

7. Discovery of Alternative Networks over Multi-Mode Radios

Task 6.1 “Network awareness and adaptation” targets, among other aspects, the efficient and dynamic discovery of the underlying network infrastructure and the identification of potential alternative paths. Thereby, it has remained a challenge, how to support a sufficient quality of experience (QoE) for delay-sensitive applications on the hand, while discovering alternative networks simultaneously over the same wireless interface on the other hand.

7.1. Introduction and Problem Description

Applications like Video-Streaming, Audio-Streaming, or even VoIP Sessions have in common that the end-to-end delay as well as packet losses have a great impact on the QoE of the end user. For example, with maximum tolerable delays of around *150-250ms* and a maximum packet loss of about 5 percent, VoIP traffic is probably the most sensitive application.

In order to allow a seamless connectivity of such applications anywhere and anytime, future wireless mobile devices will have to support a variety of heterogeneous access technologies. Although today’s smart phones or handheld devices are equipped with a bunch of technologies, e.g., GSM/GPRS/EDGE/UMTS/HDPA/WLAN/Bluetooth, it is expected that the number of technologies will further increase.

If the trend towards increasing number of technologies within one handheld device remains, the higher will be the demand for more network interface cards (NICs) being incorporated. However, such mobile devices are constrained regarding size, weight, and energy consumption.

In recent years, researchers have been working on pure Software-Defined-Radio (SDR) approaches with the goal to map multiple technologies onto a single, physical network interface card (NIC). However, beside the computational complexity, pure SDR has strong requirements regarding filters and amplifiers to support a broad range of frequency bands. Thus, it is questionable, when pure SDR solutions will be available for cheap, energy-constrained devices with limited computational resources.

In order to maximize technological coverage with a limited number of NICs within a device, a hybrid approach is common today: multi-mode, reconfigurable radios are able to do MAC as well as some PHY (base-band) processing in pure software, but still apply transceiver chains with analog parts (amplifier, filters). These chains are specifically designed and adopted to support *a group of similar technologies* and a certain range of frequency bands (e.g. joint WLAN/WiMAX and 2G/UMTS/LTE transceiver chains) [1],[2].

As a result, such multi-mode radios can support only access to one wireless technology over a specific transceiver chain, thus, no simultaneous, parallel usage of the same chain for another technology is possible.

Additionally, the potential number of transceiver chains within an end device is limited to a few in practice due to design, energy and cost issues. Compared to the capabilities of the digital part, the limited number of transceiver chains thus imposes a bottleneck for solutions of many parallel communication links with different technologies.

Consequently, this work deals with the problem how to enable a usage of heterogeneous links over the same transceiver chain, while still meeting hard QoS constraints of a connection being highly sensitive to network delay and packet losses.

A trivial solution may simply switch the transceiver chain from one access technology to the other, imposing a hard vertical handover, but obviously problems arise if the interruption for neighbor discovery and link setup is so significant that the on-going connection may face a severe degradation of the quality.

Following the spirit of [3],[4], we advocate the concept of alternating the on-going RAT (WLAN) transmission with communication preparation phases on the second RAT(WiMAX) on properly chosen short time scales.

This work presents analysis results that identify the limits of this novel approach allowing a smooth support of a connection with hard QoS constraints in WLAN concurrently with the WiMAX network entry process over a single transceiver chain. This network entry process is known to be lengthy and highly dependent on the parameterization of the access cell [5] just creating a serious challenge.

In order to identify the limits of the approach, we focus on the application which is known to have the highest requirements regarding network delay and losses., i.e., a VoIP session (G.711, *20ms* packetization).

7.2. Heterogeneous Opportunistic Approach

Following the spirit of [3],[4], we pause WLAN by means of the power save mode and switch to WiMAX in the gaps (Figure 53). Let us analyze the "quanta" in which the WiMAX entry process has to proceed. The first step consists of finding the WiMAX downlink channel and to adapt to the strict timing of the WiMAX frames afterwards.

For all WiMAX frames, the Mobile Station (MS) has to be present during the downlink (DL) part. In case there is no pending action for the uplink (UL), MS switches to WLAN and returns back for the start of the next WiMAX frame.

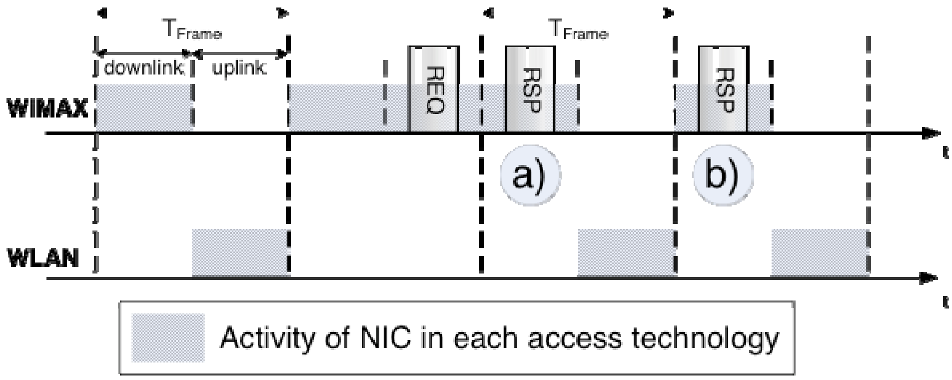

Figure 53: WLAN-WiMAX alternation principle

Beside neighbor discovery, all further steps of the network entry process for WiMAX [5] are based on request (REQ) / response (RSP) two-way handshakes, where MS issues the REQ and waits for the RSP of the Base Station (BS). For each of these steps, MS has to spend the complete frame plus the following DL subframe in WiMAX mode (in order to send out REQ and potentially receive RSP, if transmitted by BS immediately in the subsequent DL subframe, Figure 53 case a)). If RSP will be sent later in one of the following frames, it will be received by MS anyway since it always spends the DL-part of the frame within WiMAX (Figure 53 case b)).

7.2.1. Assumptions

The mobile device has a WLAN/WiMAX transceiver chain being able to switch among the RATs within insignificant time of a single clock cycle [1]. It is in the coverage of WLAN Access Point (AP) and WiMAX BS both being connected to the Internet. In both RATs, we assume to have a good wireless channel, i.e., no packet losses due to fading are considered.

The analysis starts with one active device (MS) in the WiMAX cell and considers multiple active MSs in the second step. Effects of different background load levels in WLAN were studied in [4].

In WiMAX, we consider the TDD mode as well as the parameters of the mobile profiles specified by the WiMAX Forum [6]. QPSK-1/2 is applied as Modulation and Coding Scheme (MCS) for DL-/UL-MAP, and DL/UL channel descriptor (DCD, UCD) messages.

7.2.2. Timing Issues

The duration of the WiMAX DL subframe limits the available time for VoIP transmissions in WLAN and vice versa. The analysis takes into account the maximum duration of communication patterns in each technology, such that MS can be still present for WiMAX DL subframes as well as transmit VoIP without any quality distortions in WLAN.

7.2.2.1 WiMAX

The duration of the DL part takes its maximum for the network entry process if DL/-UL-MAP, UCD, DCD (within DL-burst 1) and (the largest) RSP message (DL-burst 2) are transmitted together in one DL subframe. Eq. 1 gives the duration of the DL subframe.

$$t_{WiMAX-DL} = t_{symbol}\left\{S_{FCH,DL-MAP} + S_{preamble} + S_{DL-PUSC} + S_{DL-FUSC}\right\} \qquad (1)$$

where the number of symbols $S_{DL\text{-}PUSC/FUSC}$, the number of occupied slots N_{slots}, and the symbol duration t_{symbol} are specified below:

$$S_{DL-PUSC} = 2\left\lceil \frac{N_{slots}}{N_{DL-PUSC}} \right\rceil \quad \text{[symbols]} \qquad (2)$$

$$S_{DL-FUSC} = 2\left\lceil \frac{N_{slots}}{N_{DL-FUSC}} \right\rceil \quad \text{[symbols]} \qquad (3)$$

$$N_{slot} = \left\lceil \frac{8 * L_{burst\,X}}{48 * c * m} \right\rceil \quad \text{[slot]} \qquad (4)$$

$$t_{symbol} = (1+G)\frac{N_{FFT}}{n * BW} \qquad (5)$$

Selected parameters and their values according to IEEE 802.16e OFDMA [7] and the mobile profiles from the specification of the WiMAX Forum [6] are given in Table 6: **Parameters according to IEEE 802.16e and WiMAX Forum**

frame duration [ms]	T_{frame}	5
bandwidth [MHz]	BW	3.5, 5, 7, 8.75, 10
Cyclic prefix ratio	G	1/8
FFT size	N_{FFT}	512 (3.5, 5MHz), 1024 else
sampling factor	n	28/25 (5, 10MHz), 8/7 else
#PUSC subchannels	$N_{DL-PUSC}$	15 (512FFT), 30 (1024FFT)
#FUSC subchannels	$N_{DL-FUSC}$	8 (512FFT), 16 (1024FFT)
code rate	c	1/2, 2/3, ¾
Modulation level	m	2, 4, 6
Modulation		QPSK, 16 & 64QAM
	$S_{preamble}$	1
#Symbols	$S_{FCH,DL-MAP}$	2
DL burst #1 [Bytes]	$L_{burst-1}$	301
DL burst #2 [Bytes]	$L_{burst-2}$	163

Table 6: Parameters according to IEEE 802.16e and WiMAX Forum

7.2.2.2 WLAN

The maximum duration of VoIP transmissions depends on whether packets are awaiting their transmission in UL as well as DL or in one of the directions only as the power save (PS) signaling and its duration changes. Figure 54 **Figure 54: WLAN PS signaling with up- and downlink transmission**

shows the worst case, which consists of the wakeup, the exchange of one VoIP packet in UL and DL, and finally the sleep signaling.

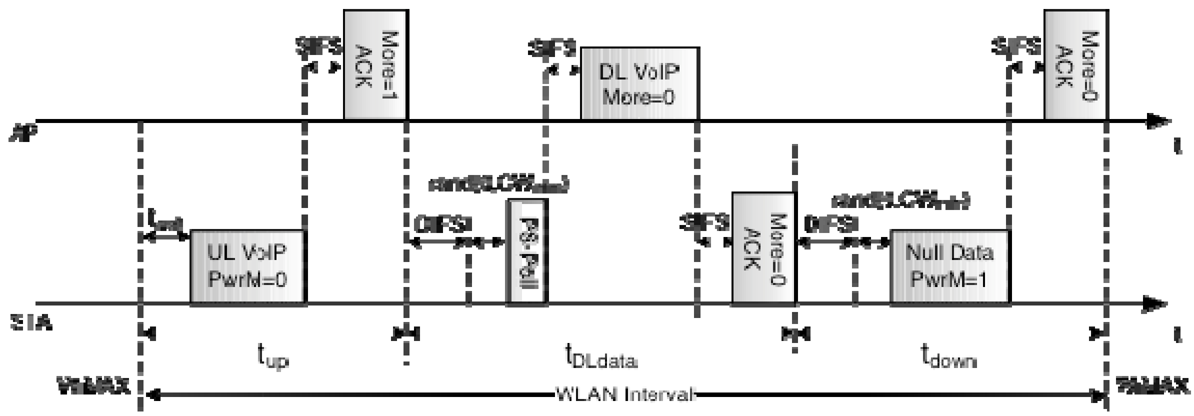

Figure 54: WLAN PS signaling with up- and downlink transmission

Eq. 6 specifies the maximum active duration in WLAN for this case:

$$t_{WLAN} = t_{up} + t_{DLdata} + t_{down} \quad (6)$$

where:

$$t_{up} = t_{wait} + t_{SIFS} + t_{ACK} + \begin{cases} t_{VoIP} & \text{pkt in UL} \\ t_{null} & \text{else} \end{cases} \quad (7)$$

$$t_{DLdata} = \begin{cases} 0 & \text{no pkt in DL} \\ t_{DIFS} + t_{PSPoll} + t_{SIFS} + t_{VoIP} + t_{SIFS} + t_{ACK} & \text{else} \end{cases} \quad (8)$$

$$t_{down} = t_{DIFS} + t_{Null} + t_{SIFS} + t_{ACK} \quad (9)$$

Table 7 gives the maximum active communication duration for IEEE 802.11g ERP OFDM (parameters as in [3], [4]), for the cases of no traffic, a packet in each direction only, and for both up -and DL. For the cases with present VoIP traffic, the highest values for the most robust MCS with 6Mbps have been selected as thresholds (highlighted in grey).

(Mbit/s)	Null data	1 VoIP packet In DL	1 VoIP packet In UL	1 VoIP packet In UL & DL each
54	1.230	1.430	1.258	1.458
6	1.350	1.875	1.618	2.142

Table 7: t_{WLAN} (ms) for IEEE 802.11g OFDM

7.3. Discussion of Results

This study firstly derives optimal values for the discovery of the WiMAX network. Secondly, it analyses the applicability of our approach for mobile WiMAX and, thirdly, evaluates how background load in WiMAX influences those results.

7.3.1. WiMax Neighbor Discovery

This paragraph derives constraints for the selection of the scanning interval similar to the work in [3].

To speed up the WiMAX discovery process, the scanning duration is maximized. In order to not induce additional delay for VoIP traffic, we shall stay below the packetization interval of *20ms*. Additionally, multiples of the WiMAX frames sizes must be avoided and the interval size should be a prime number [3]. Overall, this leads to an optimal scanning interval size of *19ms*.

This allows to find a WiMAX network in a specific frequency band in just 1 scanning interval if the WiMAX frame size (T_{frame}) is equal or smaller than *12.5ms*. Applying Eq. 4 in [3] for T_{frame} of *20ms*, results in a 5-percent probability for finding WiMAX in either 2, 3, or 4 scanning attempts, while a single attempt is only required in 85 percent of the cases.

Compared to the task of finding another WLAN AP presented in [3], the number of required scanning attempts for WiMAX is very low and can be seen as an uncritical part of the network entry process.

7.3.2. Single MS: Feasible Parameter Space for Mobile WiMAX

When the MS stays only in WiMAX for the duration of the DL subframe, a residual duration occurs, which is analyzed in the following. Since we assume that this time span is used for WLAN communication, $t_{residual}$ has to be greater that the WLAN thresholds defined in Section 7.2.2.2.

$$t_{residual} = T_{frame} - t_{WiMAX-DL} > T_{WLAN} \tag{10}$$

The residual time values were computed numerically for all combinations of channel bandwidths and MCSs (for DL-burst 2) listed in Table 6. Figure 55: Available residual time of WiMAX frame shows the results for the case of no other DL-load in WiMAX: the residual time of the WiMAX frame stays far above the WLAN thresholds for all parameter combinations. Thus, our approach is theoretically feasible if no other traffic is present.

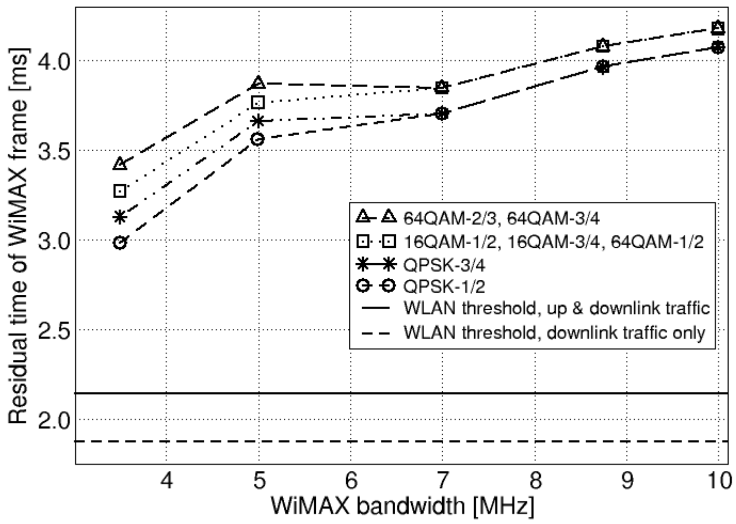

Figure 55: Available residual time of WiMAX frame

7.3.3. Multi-MS Case: Load-Dependency

The last part deals with the influence of background traffic in the WiMAX DL subframe, i.e., BS serves also other MSs. This further reduces the time span that is utilized to switch to WLAN. We identify the performance limits for this duration as a function of present traffic in the WiMAX DL subframe.

Eq. 11 specifies the maximum size of WiMAX DL subframe (as fraction of T_{frame}), under which the timing constraints of our solution still work.

$$load_{\max} = 1 - \frac{t_{WLAN-threshold}}{T_{frame}} \tag{11}$$

Table 8 gives the results for various frame sizes and both WLAN thresholds. If T_{frame} is far below the VoIP packetization interval, it is pretty likely that there is only one packet waiting in UL or DL. In this case, the smaller WLAN-threshold applies. For larger T_{frame} above *10ms*, the second WLAN threshold is likely.

Overall, for 802.16e OFDMA with *5ms* frames, our solution is applicable if the DL part consumes not more than *62.5* percent of the frame duration (or *3.1ms*).

Figure 56 finally connects the results with and without other DL traffic graphically: the edge with the step-wise characteristic of the inclined plane represents the case with no other background traffic (and QPSK-1/2 MCS for all messages). Dependent on the WiMAX parameter combination, there is still space for other *16* percent (*3.5 MHz* bandwidth) and *36* percent (*10 MHz*) of the WiMAX frame for the DL load.

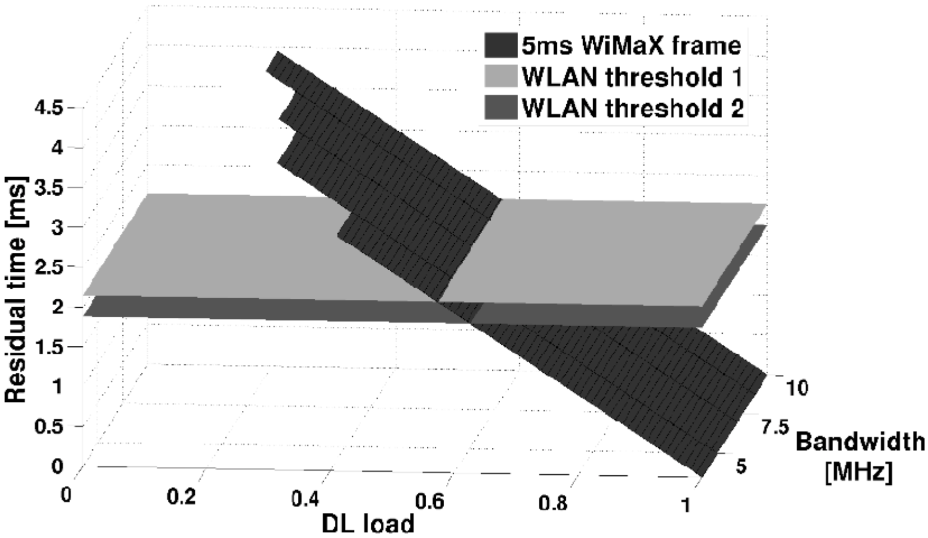

Figure 56: Residual time with different levels of WiMAX DL load

t_{WLAN}	Frame Duration [ms]						
	2.5	4	5	8	10	12.5	20
1.874ms	25.0	53.2	62.5	76.6	81.3	85.5	90.6
2.142ms	14.3	46.4	57.2	73.2	78.6	82.9	89.3

Table 8: Maximum WiMAX DL load (in percent of frame size)

7.4. Conclusions

This work presented an analysis of a novel scheme that enables ongoing WLAN communication as well as WiMAX network entry over a single transceiver chain by alternating both on small time scales. Timing constraints were derived and numerically evaluated for the mobile profiles of the WiMAX forum and WLAN. It turns out that the proposed solution is applicable for the broad range of these parameters under the assumptions of perfect timing and good wireless channels. Lastly this work gives bounds for the maximum duration of the DL subframes under which this scheme is still applicable.

The results further motivate research regarding the support of traffic with high QoS requirements over two RATs via single transceiver chains, thereby focusing on the influence of different wireless channel conditions as well as bursty traffic.

8. Final Conclusions

This deliverables has described the tools and methods for network and context awareness to drive the content adaptation of the new media data formats, the efficient network resource usage and the strived superior PQoS for the envisioned COAST content-centric network architecture. The network and context awareness both are key attributes to facilitate the deployment of the new multimedia services envisioned in the COAST project.

The deliverable has first introduced the COAST context awareness in general. Specifically, it has identified the types of context information to be considered in COAST. The analysis of the COAST delivery process has revealed the functionalities that need to exploit context information in order to meet the COAST service requirements. These context-aware functionalities are the ones responsible for content adaptation and access network selection. Content adaptation policies are more effective when more context parameters are taken into account. Context parameters for content adaptation include the user activity, user preferences, terminal capabilities, and access link characteristics. Additionally, an intelligent selection of the most convenient access network at each moment provides an improved PQoS. Therefore, the availability and performance of access networks are considered as part of the context in COAST. The deliverable provides the definition and representation of the COAST context parameters and discusses how they can be generated and consumed. An interface for retrieval of context information has been defined, including a notification mechanism for monitoring dynamic context parameters. Finally, the impact on the COAST service of the total or partial unavailability of context information is analyzed, and it is concluded that COAST will utilize the context parameters independently, such that even if only partial context information is available the content delivery will be improved as well.

After the COAST context awareness has been described in general, a specific use-case for utilization of context is detailed. As the current trend in multimedia delivery implies to more intelligently link the huge amount of content sources to the user, the deliverable introduces a method to detect user interest due to user movements towards an object of potential interest. If a user stops in front of an object (museum exhibit or information board) he is offered multimedia content on his mobile device that gives further information about that object. The potential user interest is detected by a utilization of smart phone sensory and visual data. The introduced system requires a small server connected to a USB camera and a wireless interface to the mobile device such as Bluetooth. If interest is detected the server can communicate the COAST content object identifier to the user's device, which can retrieve the content via the COAST video retrieval agent. The evaluation results verify a reasonable performance for up to three people in the scene with a successful detection of interest for 75 to 100 % of the conducted scenarios using a sample rate of 8.33 Hz. The employed software will be optimized to achieve shorter computing times for the correlation computations and integrated into an Android smart phone in Work Package 7.

The network awareness required for the COAST architecture has been addressed in this deliverable by a simulation of the ALTO-based caching solutions. As the measurement results verify the suitability of ALTO for the placement and retrieval of COAST content, the ALTO server and client software will be integrated into COAST to provide the improved PQoS and to reduce costs.

As the COAST network monitor shall estimate the load of the available links to deliver the multimedia content, a selected tool is analyzed for provision of available bandwidth. The tool fulfills the requirements for applicability in the PlanetLab architecture, which is considered in COAST for evaluation of the overlay of content-aware nodes. The bandwidth estimates are verified to be suitable to decide for the currently best cache. The tool will be integrated into the COAST CEP component and can complement the ALTO guidance.

While the previous works on selection of the best COAST cache do regard the wired link characteristics between the CEP and the potential COAST content aware node to be employed as a content source, the last works discover and utilize the options for the wireless connectivity. Specifically, an architectural concept and methods for delivery of multimedia content using scalable video over multiple wireless interfaces were developed and implemented, which solve a) the signaling for selection of the appropriate wireless interface for a codec layer flow and b) the NAT component problems due to the UDP based media flows. The given solution is demonstrated to allow for improved PQoS due to content adaptation, and interruptions due to the shifting of codec layers were addressed by introduction of additional queues in the network layer. The final COAST architecture will demonstrate scalable video on a laptop, where stable PQoS is achieved by utilization of the available wireless links even if the wireless networks or the provided link characteristics do change due to the user mobility.

The given work on discovery of alternative wireless networks for multi-mode radios is a starting point to account for the future trend of transceiver chain circuitry in the mobile devices, and needs further investigations before it can be integrated in a content distribution network. The work serves for the detailed description of the COAST service platform in work package 7 in that it will be made sure there that the architecture in principle can utilize the upcoming radio technologies as well to provide improved services.

9. References

[1] J. G. Atallah and M. Ismail, "Future 4G front-ends enabling smooth vertical handovers," *Circuits and Devices Magazine, IEEE*, vol. 22, no. 1, pp. 6 – 15, 2006.

[2] U. Ramacher, "Software-defined radio prospects for multistandard mobile phones," *Computer*, vol. 40, pp. 62–69, 2007.

[3] M. Emmelmann, S. Wiethölter, and H.-T. Lim, "Opportunistic scanning: Interruption-free network topology discovery for wireless mesh networks," in *IEEE WoWMoM*, Kos, Greece, Jun. 2009, pp. 1 – 6.

[4] M. Emmelmann, S. Wiethölter, and H.-T. Lim, "Influence of network load on the performance of opportunistic scanning," in *IEEE LCN*, Zurich, Switzerland, Oct. 2009, pp. 1–8.

[5] M. Hollick, P. Mogre, C. Schott, and R. Steinmetz, "Slow and steady: Modelling and performance analysis of the network entry process in IEEE 802.16," in *IEEE IWQoS 2007*, jun. 2007, pp. 126 –134.

[6] WiMAX Forum, "Mobile System Profile Specification, Release 1.5, Common Part," WiMAX Forum, Tech. Rep. WMF-T23-001-R015v01, August 2009.

[7] 802.16-2009: IEEE Standard for Local and metropolitan area networks, Part 16: Air Interface for Broadband Wireless Access Systems, IEEE Std., May 2009.

[8] E. Goldoni, G. Rossi, A. Torelli,"Assolo: an Efficient Tool for Active End-to-end Available Bandwidth Estimation" , Int. Journal on Advances in Systems and Measurements, Volume 2, Number 4, 2009

[9] M. Jain and C. Dovrolis, "Pathload: A measurement tool for end-to-end available bandwidth", Proc. Passive and Active Measurement Conference (PAM 2002), March 2002, pp. 14-25

[10] V. Ribeiro, R. Riedi, R. Baraniuk, J. Navratil, and L. Cottrell, "path-Chirp: Efficient available bandwidth estimation for network paths", in Proc. Passive and Active Measurement Conference (PAM 2003), Apr. 2003

[11] A. B. Downey, "Using pathchar to estimate Internet link characteristics",Proc. Of the conference on Applications, technologies, architectures, and protocols for computer communication (SIGCOMM'99), 1999

[12] D. Antoniades, M. Athanatos, A. Papadogiannakis, E.P. Markatos, C. Dovrolis, "Available bandwidth measurement as simple as running wget", Proc. Passive and Active Measurement Conference (PAM 2006), March 2006

[13] N. Hu and P. Steenkiste, "Evaluation and characterization of available bandwidth probing techniques", IEEE Journal on Selected Areas in Communications, Volume 21, Number 6, pp.879-894, August 2003

[14] J. Strauss, D. Katabi, and F. Kaashoek, "A measurement study of available bandwidth estimation tools", Proc. ACM SIGCOMM Conference on Internet Measurement (IMC'03), pp. 39-44, October 2003

[15] S. Ekelin, M. Nilsson, E. Hartikainen, A. Johnsson, J.-E. Mang, B. Melander, and M. Bjorkman, "Real-time measurement of end-to-end available bandwidth using kalman filtering", Proc. IEEE/IFIP Network Operations and Management Symposium (NOMS 2006), Apr. 2006, pp. 73-84

[16] C. Dovrolis, P. Ramanathan, and D. More, "Packet dispersion techniques and a capacity estimation methodology", IEEE/ACM Transactions on Networking (TON), Volume 12, Issue 6, December 2004

[17] R. Kapoor, L.-J. Chen, L. Lao, M. Gerla, and M.Y. Sanadidi, "CapProbe: A simple and accurate capacity estimation technique", Proc. of the ACM SIGCOMM'04, August 2004

[18] COAST consortium, "Deliverable 2.2: End-to-End Future Content Network Specification", August 2010

[19] COAST consortium, "Deliverable 2.1: Service Requirements Specification", May 2010

[20] S. Hsi and H. Fait, "RFID enhances visitors' museum experience at the Exploratorium", Comm. Of the ACM, pp. 60-65, September 2005

[21] S. Hsi, "The electronic guidebook: A study of user experiences using mobile web content in a museum setting", IEEE International Workshop on Wireless and Mobile Technologies in Education, December 2002

[22] F. Sparacino, "The museum wearable: real-time sensor-driven understanding of visitors' interests for personalized visually-augmented museum experiences", in Proceedings of Museums and the Web (MW 2002), pp. 17-20, April 2002

[23] H. Bay, B. Fasel, and L. V. Gool, "Interactive museum guide: Fast and robust recognition of museum objects", in Proceedings of the International Workshop on Mobile Vision, 2006

[24] J.-C. Cano, P. Manzoni, and C. K. Toh, " Ubiqmuseum: A Bluetooth and Java based context aware system for ubiquitous computing", Journal of Wireless Personal Communication, Springer Verlag, 2006

[25] C.-Y. Tsai, "Location-aware tour guide systems in museums", Scientific Research and Essays, Vol. 5(8), pp. 714-720, Academic Journals, April 2010

[26] T. Yamashita, D. Takaoka, N. Izumi, A. Sahima, K. Kurumatani, and K. Hasida, "Context-awareness of content delivery service based on location estimate with an active RFID system", 18th Int. Workshop on Database and Expert Systems Applications (DEXA 2007), 2007

[27] O. Shigeta, S. Kagami, and K. Hashimoto, "Identifying a moving object with an accelerometer in a camera view", 8 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 3872-3877, September 2008

[28] T. Teixeira, D. Jung, G. Dublon, and A. Savvides, "PEM-ID: Identifying people by gait-matching using cameras and wearable accelerometers", Proceedings of the Third ACM/IEEE International Conference on Distributed Smart Cameras (ICDSC'09), August - September 2009

[29] T. Teixeira, D. Jung and G. Dublon, "Identifying People in Camera Networks using Wearable Accelerometers", Proceedings of the 2nd International Conference on Pervasive Technologies Related to Assistive Environments, June 2009

[30] T. Teoxeora, D. Jung, and A. Savvides, "Tasking Networked CCTV Cameras and Mobile Phones to Identify and Localize Multiple People", In Proceedings of the 12th ACM international conference on Ubiquitous computing (Ubicomp'10), pp. 213-222, 2010

[31] D. Jung, T. Teixeria, and A. Savvides, "Towards cooperative localization of wearable sensors using accelerometers and cameras", Proceedings of the 29th conference on Information communications (INFOCOM'10), 2010

[32] J. Ren, M. Rahman, and N. Kehrarnavaz, "Real-time head pose estimation on mobile platforms", Proceedings of the 13th World Multi-conference on Systemic, Cybernetics and Informatics, July 2009

[33] K.I. Benta, M. Cremene, N. R. Giba, U.X. Eligio, and A. Rarau, "Secondary emotions deduction from context", Innovations and Advances in Computer Sciences and Engineering, pp. 165-170, Springer Netherlands, 2010

[34] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey", ACM Journal of Computing Devices, December, 2006

[35] G. Bradski and A. Kaehler, "Learning OpenCV: Computer Vision with the OpenCV Library, Computer Vision with the OpenCV Library", O'Reilly, 2008

[36] T. P. Chen, H. Haussecker, A. Bovyrin, R. Belenov, K. Rodyushkin, A. Kuranov, and V. Eruhimov, „Computer vision work-load analysis: Case study of video surveillance system", Intel Technology Journal, volume 9, pp. 109-118, May 2005

[37] J. Zylberberg, S. Rein, A. Wolisz, "Recognition of Personal Interest using Cooperation of Smartphone and Camera Sensors", Proceedings of European Conference on Wireless Sensor Networks, EWSN'2011, February 2011

[38] Li Mingzhe, M. Claypool, R. Kinicki, "WBest: A bandwidth estimation tool for IEEE 802.11 wireless networks", IEEE conference on Local Computer Networks, October 2008

[39] ISO/IEC, "ISO/IEC 21000-7 Information Technology- Multimedia Framework (MPEG-21) – Part 7: Digital Item Adaptation", Second edition July 2007

[40] COAST Consortium, "COAST Description of Work," Approved by EC on 13 November 2009.

[41] J. Challenger, P. Dantzig, A. Iyengar, and K. Witting, "A fragment-based approach for efficiently creating dynamic web content" ACM Trans. Internet Technol., vol. 5, pp. 359–389, May 2005.

[42] S. Seny, J. Rexfordz, and D. Towsley, "Proxy prefix caching for multimedia streams" in INFOCOM '99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE, pp. 1310–1319, 1999.

[43] K. Park and V. S. Pai, "Scale and performance in the CoBlitz largefile distribution service" in Proceedings of the 3rd USENIX/ACM Symposiumon Networked Systems Design and Implementation (NSDI), p. 2006.

[44] M. J. Freedman, "Experiences with CoralCDN: a five-year operational view" in Proceedings of the 7th USENIX conference on Networked systems design and implementation, NSDI'10, (Berkeley, CA, USA), pp. 7–7, USENIX Association, 2010.

[45] V. K. Gurbani, V. Hilt, I. Rimac, M. Tomsu, and E. Marocco, "A survey of research on the application-layer traffic optimization problem and the need for layer cooperation" Comm. Mag., vol. 47, no. 8, pp. 107–112, 2009.

[46] C. Aggarwal, J. L. Wolf, and P. S. Yu, "Caching on the world wide web" IEEE Transactions on Knowledge and Data Engineering, vol. 11, pp. 94–107, 1999.

[47] E. Balafoutis, A. Panagakis, N. Laoutaris, and I. Stavrakakis, "The impact of replacement granularity on video caching" in Proceedings of the Second International IFIP-TC6 Networking Conference on Networking Technologies, Services, and Protocols; Performance of Computer and Communication Networks; and Mobile and Wireless Communications, NETWORKING '02, (London, UK, UK), pp. 214–225, Springer-Verlag, 2002.

[48] K.-L. Wu, P. S. Yu, and J. L. Wolf, "Segment-based proxy caching of multimedia streams" in

Proceedings of the 10th international conference on World Wide Web, WWW '01, (New York, NY, USA), pp. 36–44, ACM, 2001.

[49] S. Chen, S. Wee, B. Shen, and X. Zhang, "Adaptive and lazy segmentation based proxy caching for streaming media delivery" in Proceedings of ACM NOSSDAV, pp. 22–31, 2003.

[50] Chen, Y., Qiu, L., Chen, W., Nguyen, L., and Katz, R. H. Efficient and adaptive Web replication using content clustering. IEEE Journal on Selected Areas in Communications, 21(6), pp. 979–994, 2003.

[51] Wu, B. and Kshemkalyani, A. D. Objective-optimal algorithms for long-term Web Prefetching. IEEE Transactions on Computers, 55(1), pp. 2–17, 2006.

[52] Fujita, N., Ishikawa, Y., Iwata, A., and Izmailov, R. Coarse-grain replica management strategies for dynamic replication of Web contents. Computer Networks: The International Journal of Computer and Telecommunications Networking, 45(1), pp. 19–34, 2004.

[53] Sivasubramanian, S., Pierre, G., van Steen,M., and Alonso, G. Analysis of caching and replication strategies for Web applications. IEEE Internet Computing, 11(1), pp. 60–66, 2007.

[54] Bartal, Y. Probabilistic approximation of metric space and its algorithmic applications. InProc. of 37th Annual IEEE Symposium on Foundations of Computer Science, 1996.

[55] Jamin, S., Jin, C., Jin, Y., Raz, D., Shavitt, Y., and Zhang, L. On the placement of Internetinstrumentation. In Proc. of IEEE INFOCOM, Tel-Aviv, Israel, pp. 295–304, 2000.

[56] Krishnan, P., Raz, D., and Shavitt, Y. The cache location problem. IEEE/ACM Transactionon Networking, 8(5), 2000.

[57] Radoslavov, P., Govindan, R., and Estrin, D. Topology-informed Internet replica placement. In Proc. of Sixth International Workshop on Web Caching and Content Distribution, Boston, Massachusetts, 2001.

[58] Jamin, S., Jin, C., Kure, A. R., Raz, D., and Shavitt, Y. Constrained mirror placement on the Internet. In Proc. of IEEE INFOCOM, Anchorage, Alaska, USA, 2001.

[59] Qiu, L., Padmanabhan, V. N., and Voelker, G. M. On the placement of Web server replicas. In Proc. of IEEE INFOCOM, Anchorage, Alaska, USA, pp. 1587–1596, 2001.

[60] Li, B., Golin, M. J., Italiano, G. F., Xin, D., and Sohraby, K. On the optimal placement of Web proxies in the Internet. In Proc. of IEEE INFOCOM, NY, USA, pp. 1282–1290, 1999.

[61] Vakali, A. and Pallis, G. Content delivery networks: status and trends. IEEE Internet Computing, 7(6), IEEE Computer Society, pp. 68–74, 2003.

[62] Douglis, F. and Kaashoek, M. F. Scalable Internet services. IEEE Internet Computing, 5(4),pp. 36–37, 2001.

[63] Akamai Technologies, 2007. www.akamai.com

[64] Dilley, J., Maggs, B., Parikh, J., Prokop, H., Sitaraman, R., and Weihl, B. Globally distributed content delivery. IEEE Internet Computing, pp. 50–58, 2002.

[65] Pallis, G. and Vakali, A. Insight and perspectives for content delivery networks. Communications of the ACM, 49(1), ACM Press, NY, USA, pp. 101–106, 2006.

[66] Kangasharju, J., Roberts, J., and Ross, K. W. Object replication strategies in content distribution networks. Computer Communications, 25(4), pp. 367–383, 2002.

[67] Fujita, N., Ishikawa, Y., Iwata, A., and Izmailov, R. Coarse-grain replica management strategies for dynamic replication of Web contents. Computer Networks: The International Journal of Computer and Telecommunications Networking, 45(1), pp. 19–34, 2004.

[68] Johnson, K. L., Carr, J. F., Day, M. S., and Kaashoek, M. F. The measured performance of content distribution networks. Computer Communications, 24(2), pp. 202–206, 2001.

[69] Cooper, I., Melve, I., and Tomlinson, G. Internet Web replication and caching taxonomy. Internet Engineering Task Force RFC 3040, 2001. www.ietf.org/rfc/rfc3040.txt

[70] Freedman, M. J., Freudenthal, E., and Mazi` res, D. Democratizing content publication with e Coral. In Proc. of 1st USENIX/ACM Symposium on Networked Systems Design and Implementation, San Francisco, CA, USA, 2004.

[71] Tse, S. S. H. Approximate algorithms for document placement in distributed Web servers. IEEE Transactions on Parallel and Distributed Systems, 16(6), pp. 489–496, 2005.

[72] Pallis, G., Vakali, A., Stamos, K., Sidiropoulos, A., Katsaros, D., and Manolopoulos, Y. A latency-based object placement approach in content distribution networks. In Proc. of the 3rd Latin American Web Congress (La-Web 2005), IEEE Press, Buenos Aires, Argentina, pp. 140–147, 2005.

[73] J. Current, M. Daskin, and D. Schilling, “Discrete Network Location Models,” in Facility Location Theory: Applications and Methods, Z. Drezner and H. Hamacher, Eds., 2001, Forthcoming.

[74] V. Vazirani, Approximation Algorithms, ISBN: 3-540-65367-8. Springer-Verlag, 2001.

[75] A. Venkataramanj, P. Weidmann, and M. Dahlin, “Bandwidth Constrained Placement in a WAN,” in ACM Symposium on Principles of Distributed Computing (PODC’01), August 2001.

[76] Lili Qiu, Venkata N. Padmanabhan, and Geo rey M. Voelker. On the Placement of Web Server Replicas. In Proceedings of the IEEE Infocom 2001, Anchorage, Alaska, USA, April 2001.

[77] Sugih Jamin, Cheng Jin, Anthony R. Kurc, Danny Raz, and Yuval Shavitt. Constrained mirror Placement on the Internet. In Proceedings of the IEEE Infocom 2001, Anchorage, Alaska, USA, April 2001.

[78] Songqing Chen, Bo Shen, Susie Wee, and Xiaodong Zhang. Designs of high quality streaming proxy systems. In Proc. of IEEE INFOCOM, 2004.

[79] Jeffrey Erman, Alexandre Gerber, Mohammad Hajiaghayi, Dan Pei, Subhabrata Sen, and Oliver Spatscheck. To cache or not to cache: The 3g case. IEEE Internet Computing, 99(PrePrints), 2010.

[80] Amazon Web Services. Amazon simple storage service (amazon s3), 2011.

[81] Rackspace Hosting. Cloud files pricing, 2011.

[82] NephoScale. Object-based cloud storage, 2011.

[83] D.R. Choffnes and F.E. Bustamante, "Taming the Torrent", in Proceedings of ACM SIGCOMM, Seattle, WA, USA, August 2008.

[84] F. Dabek, R. Cox, F. Kaashoek, and R. Morris, "Vivaldi: A decentralized network coordinate system", in ACM SIGCOMM Computer Communication Review, Volume 34, No. 4, 2004.

[85] COAST consortium, “Deliverable 8.1: Initial Assessment and Market Analysis”, December 2010

10. Appendix

10.1. Description of User Context Detection Scenarios

10.1.1. Scenario 1 (scenarios with single but different participants)

Aim: To have a base comparison to demonstrate the data alignment process.

Scenario 1.1: Participant 1 enters and browses all 3 objects in order of 1, 2, 3. Stop time is approximately 5 seconds for each object.

Scenario 1.2: Participant 1 enters room and browses object 1. Walks around room, but exits

without further browsing.

Scenario 1.3: Participant 1 enters in and browses no objects, but walks towards each of the objects without stopping in front of them.

10.1.2. Scenario 2 (two participants)

Aim: Differentiate between a person moving and a person not moving, while also avoiding possible confusions arising from motion detection (i.e. path-crossing).

Scenario 2.1

- Participant 1 enters room, selects object 1 to browse for up to 5 seconds. Moves to another object to browse for up to 5 seconds. After this the participant leaves the room.
- Participant 2 meanwhile enters room, walks around towards objects, but chooses not to show interest, as in scenario 1.3. Subsequently leaves the room.

It has to be ensured that both participants do not walk near each other or cross paths.

Scenario 2.2

Aim: Differentiate between two people who both show interest by stopping at an object.

- Participant 1 enters room and selects object 1 to browse for up to 5 seconds. Following this, participant selects another object 2 to browse. Following this, participant shows interest in a final object 3.
- Meanwhile after participant 1 has finished browsing object 1, participant 2 enters room and selects object 1 to browse, without crossing paths with the other participant. This would require the participants to browse objects in a circular motion.
- Participant 2 continues browsing further objects, following the same path as participant 1.

Scenario 2.3

Aim: Differentiate between two people, while one does not show interest. The participants, however, do cross paths.

- Participant 1 enters room and browses object 1 for up to 5 seconds.
- Participant 2 enters room and begins to walk nearby object 1, 2 and 3. Chooses not to view anything and leaves.
- Meanwhile, participant 1 chooses an additional object 2 to browse, causing the two to cross paths. Participant 1 subsequently views object 2 for up to 5 seconds and leaves the scene.

Scenario 2.4

Aim: Differentiate between two people showing interest while crossing paths

Same scenario as 2.2, but paths in a circular pattern is not enforced. Therefore, participant 1 views objects in order of 1,2,3 while participant 2 views objects in order of 3,2,1.

10.1.3. Scenario 3 (three participants)

Aim: Analyze the effects of path crossing.

Scenario 3.1

Aim: Differentiation between multiple people with different movement patterns. Also provides an example of proximity to objects, but no interest detection.

- Participant 1 enters room and walks slowly nearby object 3, but does not stop.
- Participants 2 & 3 enter shortly following each other.
- Participant 2 selects object 2 to show interest in, and browses for up to 5 seconds.
- Participant 1 meanwhile selects object 1 to browse.
- Participant 3 decides not to show interest and leaves while remaining participants are still browsing.
- Participant 2 chooses object 1 to browse.
- Participant 1 leaves area.
- Participant 2 leaves area.

Scenario 3.2

Aim: Differentiation between multiple people with different movement patterns, where the majority of participants do not show interest.

- Participant 1 enters room and walks slowly nearby object 3, but does not stop.
- Participant 2 enters room and walks to object 1, showing interest in it. Browses for up to 5 seconds.
- Participant 1 walks further around, but is not showing interest.
- Participant 2 shows interest in object 2.
- Participant 3 enters room, stands in middle looking around.
- Participant 1 decides to leave.
- Participant 3 decides to leave.
- Participant 2 leaves.

Scenario 3.3

Aim: Differentiation between multiple people with different movement patterns, but all show interest, particularly in a single object at the same time.

- Participants 1 & 2 enter room and view object 2. Participant 2 remains for a shorter period of time at object 2 and moves to object 1.
- Participant 1 remains.
- Participant 3 enters and shows interest in object 2.
- After arrival of participant 3, participant 1 views object 3.

- Participant 2 views object 3, while participant 1 is still there.
- All participants leave at roughly the same time.

10.1.4. Scenario 4 (four participants)

Aim: Test the effects of varied movement and varied levels of interest.

- Participant 1 enters room to browse object 1.
- Meanwhile, participant 2 enters room to browse object 2.
- Participant 3 and 4 enter at the same time. Both head towards object 3.
- Participant 3 and 4 stop at object 3 to show interest, however participant 3 stays for less than 1 second. Participant 4 remains.
- Participant 3 walks to object 1 and shows interest.
- Participant 4 walks to object 1 and shows interest.
- Participant 1 meanwhile walks to object 3, remains only for a short period of time (1 second) and continues to object 2.
- Participant 4 walks to object 2 to show interest
- Participant 2 passes object 1 before stopping at 3 to inspect it.
- Participant 3 and participant 4 meet in middle of room and leave together.
- Participant 1 leaves.
- Participant 2 leaves.

10.1.5. Scenario 5 (five participants)

Aim: Test the full effects of varied movement and varied levels of interest over a longer, single period of time.

- Participant 1 enters room to browse object 1.
- Meanwhile, participant 2 enters room to browse object 2.
- Participant 3 and 4 enter at the same time. Both head towards object 3.
- Participant 3 and 4 stop at object 3 to show interest, however participant 3 stays for less than 1 second. Participant 4 remains.
- Participant 3 walks to object 1 and shows interest.
- Participant 4 walks to object 1 and shows interest.
- Participant 3 walks to object 2 and shows interest.
- Participant 1 meanwhile walks to object 3, remains only for a short period of time (1 second), and continues to object 2.
- Participant 5 enters room.
- Participant 5 and participant 2 stop in the middle of the room and talk.
- Participant 2 walks to object 1, participant 5 to object 3.
- Participant 2 and participant 5 change places simultaneously and view alternating objects.

- Participant 3 and participant 4 meet in middle of room and leave together.
- Participant 1 leaves.
- Participant 5 leaves.
- Participant 2 leaves.