A personalized system for scalable distribution of multimedia content in multicast wireless networks

Ismael de Fez[a], Miriam Gil[b], Joan Fons[b], Juan Carlos Guerri[a], Vicente Pelechano[b]

[a] *Institute of Telecommunications and Multimedia Applications (iTEAM),*
Universitat Politècnica de València, Camino de Vera, 46022, Valencia, Spain
[b] *Research Center on Software Production Methods (PROS),*
Universitat Politècnica de València, Camino de Vera, 46022, Valencia, Spain

e-mail: isdefez@iteam.upv.es, mgil@pros.upv.es, jjfons@pros.upv.es, jcguerri@dcom.upv.es, pele@pros.upv.es
phone: 34-963879588

Abstract. This paper presents a novel architecture for scalable multimedia content delivery over wireless networks. The architecture takes into account both the user preferences and context in order to provide personalized contents to each user. In this way, third-party applications filter the most appropriate contents for each client in each situation. One of the key characteristics of the proposal is the scalability, which is provided, apart from the use of filtering techniques, through the transmission in multicast networks. In this sense, content delivery is carried out by means of the FLUTE (File Delivery over Unidirectional Transport) protocol, which provides reliability in unidirectional environments through different mechanisms such as AL-FEC (Application Layer – Forward Error Correction) codes, used in this paper. Another key characteristic is the context-awareness and personalization of content delivery, which is provided by means of context information, user profiles, and adaptation. The system proposed is validated through several empirical studies. Specifically, the paper presents evaluations of two types that collect objective and subjective measures. The first evaluate the efficiency of the transmission protocol, analyzing how the use of appropriate transmission parameters reduces the download time (and thus increasing the Quality of Experience), which can be minimized by using caching techniques. On the other hand, the subjective measures present a study about the user experience after testing the application and analyze the accuracy of the filtering process/strategy. Results show that using AL-FEC mechanisms produces download times until four times lower than when no protection is used. Also, results prove that there is a code rate that minimizes the download time depending on the losses and that, in general, code rates 0.7 and 0.9 provide good download times for a wide range of losses. On the other hand, subjective measures indicate a high user satisfaction (more than 80%) and a relevant degree of accuracy of the content adaption.

Keywords: *Push content download service, Personalization, Context-awareness, File delivery over unidirectional transport – FLUTE, Wireless networks, Multicast*

1. Introduction

The use of smartphones is experiencing an exponential growth in the last years. According to a meta-analysis by eMarketer [1], the global smartphone penetration is around 33% in 2013 and it is expected to pass 50% by 2017. This percentage is much higher in developed countries, in which the smartphone penetration is expected to pass 75% by 2017. As it is well known, smartphones offer advanced features such as geolocation, connectivity, sensors or advanced interaction mechanisms that allow developers to provide new kind of applications with an added value for users.

Nowadays, users demand more and more services, contents and applications in order to satisfy their needs. This situation generates new opportunities as well as new challenges, since one of the main requirements of new services is the quality offered to the user, independently of the number of users. In this sense, there are some environments where it is rather difficult to offer certain kind of services with a proper quality. Solving these technical difficulties represents a challenge to open new business routes.

For instance, providing multimedia content delivery services for a high number of users (each one with different preferences) gathered in a relatively small area simultaneously is not an easy task. Current solutions need great investments in order to be capable of satisfying the demand of a large amount of users. In overcrowded public spaces, basically, the bottleneck is the wireless access, which becomes saturated due to the amount of simultaneous connections. This causes difficulties for users to access available services with an acceptable quality. In these situations, it results very interesting to explode the use of multicast networks, which represent a good mechanism for reducing the bandwidth in wireless networks. By means of multicast networks it is possible to send content to all interested users using a single transmission operation.

Also, it is very probable that a large number of users are interested in the same contents, the most relevant ones in each user context. The concept of context refers to those information items that can be used to characterize the situation of entities (people, places or objects), which are relevant for the interaction between the user and the application [2]. In this sense, context-aware systems adapt their services according to the context information received, without any interaction from the user. To achieve this, it is desirable that both devices and services have access to context information and react according to this information. In this way, depending on the context information, the system could deliver special offers in shopping centers, replays of the most interesting moments in a sport event or trailers while waiting in the cinema. In the case of mobile devices, the pack of sensors (GPS, digital compasses, accelerometers, and so on) enables the devices to witness their owners' context that the sensors can identify [3]. However, as identified in [4], there is a need of new content delivery systems that exploit the infrastructure of mobile services to determine and take into account the current context of the user to deliver intelligent contents.

However, not all contents are relevant for all users since each user has different preferences and needs. Thus, content delivery should be personalized for each user according to the user's profile in order to guarantee a good Quality of Experience (QoE). For example, a runner (user that likes running) in a shopping center may be interested in contents about sport sales. Consequently, content delivery systems should be able of accessing the individual user's profile stored on the mobile device. This allows a dynamic adaptation of content since user information is updated over time according to user preferences. Also, when a relevant content is downloaded, users should be informed about it to be aware of the content. However, these notifications can become overwhelming for users and interrupt them in inappropriate situations. Thus, another challenge is to prioritize the user's attention and consider how to interrupt the user by choosing the most appropriate interaction mechanisms from all the ones available in his/her mobile device (sound, vibration, dialog with text, etc.). This decision can be made according to the user's situation (e.g., surrounded with people, running, etc.).

In this way, considering the aforementioned challenges, the objective of this paper is to present and evaluate a general-purpose architecture for the scalable delivery of personalized multimedia contents adapted to the users' preferences and their context in multicast wireless networks. This system allows the content server to deliver multimedia contents to an unlimited number of users in an efficient manner. By means of the user preferences and the user's context, the receiver application is able to download automatically relevant personalized multimedia contents, so users do not need to look for contents manually. Also, the user interface is adapted to the user situation at each moment, minimizing the annoyance and therefore increasing the QoE of the user.

The rest of the paper is structured as follows. Section 2 presents the related work and Section 3 explains a brief background of content transmission in unidirectional environments. The system overview is presented in Section 4, where the architecture of the system as well as implementation details are explained. The proposal is evaluated in Section 5, which presents two kinds of evaluations to gather objective and subjective measures. Finally, Section 6 shows the main conclusions and future work.

2. Related work

In this work, we address the scalable distribution of multimedia contents personalized to each user by taking into account user's preferences and context. Thus, we present and discuss approaches that deal with content distribution, transmission, context awareness and personalization.

There are different proposals that use context information to improve the performance of content distribution systems. Firstly, it should be highlighted that [5] presents a state of the art of context-sensitive systems. Also, [6] proposes an architecture that uses operational data of the service (popular content, bandwidth consumption, peak hours…) to download content in user devices in an efficient way. Moreover, [7] presents a network protocol model that uses information related to social relationships and characteristics of users from social networks in order to improve the network performance. There are also different projects within the 7th Framework Programme of the European Union [8] with similar objectives. For instance, COST presents a network of intelligent nodes, where contents are classified and cached to improve the service availability, and another project called Notube focuses on TV content as a medium for personalized interaction. In this work, we propose a mobile content distribution system that uses the preferences of users and their context to filter and download the most relevant contents according to the user's situation.

Regarding the content transmission, as it is explained in the following section, this proposal is based on FLUTE, a protocol used by different standardization organisms, such as DVB-H (Digital Video Broadcasting – Handheld) [9], DVB-IPTV (DVB – Internet Protocol TV) [10] or MBMS (Multimedia Broadcast Multicast Service) [11]. In this sense, there are different research works where the use of FLUTE in DVB and MBMS is analyzed, such as [12, 13, 14, 15]. In addition to DVB and MBMS, FLUTE can be used with other technologies such as TESLA (Timed Efficient Stream Loss-Tolerant Authentication) [16] or for scalable DTN (Delay-Tolerant Networking) communications [17, 18]. Apart from file transmission, other related works propose the use of FLUTE to provide video services, such as

[19, 20]. Another research work related to FLUTE to be highlighted is [21], where the transmission parameters of the protocol are analyzed in detail; and [22], which studies how the FDT (File Delivery Table) transmission frequency affects the download time of files using FLUTE. As we will see in the following section, one of the key elements of the FLUTE protocol is the FDT, which is used to include metadata referred to the contents sent. Despite the importance of the FDT, in the bibliography hardly exist publications where the performance of the FDT is analyzed. In this context, in contrast to different works based on FLUTE, this paper aims to provide personalized content download services by means of the FDT. These metadata defined by the FDT are useful to filter the contents in reception.

In this sense, adaptation to individual users and tasks is designated as personalization [23, 24]. Several works deal with personalizing intelligent environments [25] and adapting services to the user [26] based on the occupants' presence, behavior, and intentions. In these works, the personalization is set manually by the user since it is the user who chooses what information is going to be displayed or hidden. Personalization is also carried out in recommender systems, for instance for online shops based on the items the user has bought before. An overview of recommender systems is given in [27]. In our approach, the personalization is made automatically by using the mobile user's profile and the user's context. Traditionally recommender systems deal with applications having only two types of entities, users and items, and do no put them into a context when providing recommendations [28].

With regard to context awareness, specifically in the area of context-aware mobile interaction, several proposals have been carried out to adjust the mobile configuration profile according to changes in the user's context [29, 30]. However, they focus on context recognition, not on the interaction adaptation. There are also proposals that allow users to manually adapt the mobile interaction depending on the context information [31]. Nevertheless, in contrast to this proposal, all these works provide adaptation at the device level and not at the application and content level. Also, they not address the problem of scalable content distribution in wireless networks.

3. Background of content transmission in unidirectional environments

As mentioned in the introduction, there are several applications that require the delivery of a large amount of data over wireless networks to multiple users simultaneously. In these situations, multicast networks are appropriate since they represent a good mechanism to reduce the bandwidth in wireless networks. One of the main drawbacks of this kind of networks is the lack of a feedback channel from clients to the server, so clients are not able to inform the server about the packet losses (which are typical in wireless environments). In this way, it is highly recommendable to use a protocol that offers reliability in file transmission over unidirectional networks.

One of the most used protocols by different standardization institutions for file delivery is FLUTE. In this sense, this proposal uses FLUTE over IP with broadcast capabilities (such as Wi-Fi or DVB) for file distribution.

FLUTE, defined in RFC 6726 [32], is a protocol for the unidirectional delivery of files over the Internet, which is particularly suited to multicast networks. FLUTE transmissions are based on file delivery sessions, which are uniquely identified by a source IP address and by an identifier called TSI

(Transport Session Identifier). Each session contains one or more associated channels, in which files are sent. Each channel sends files in a specific port with a certain transmission rate.

These parameters that identify the session and the channel are indicated in the Session Description. With the information provided by the Session Description clients are able to connect to a certain FLUTE session and channel. There are different ways to get the Session Description, one of most used is through the protocol SDP (Session Description Protocol).

Once clients are connected to a certain FLUTE channel, they know the files (and their characteristics) that the channel is sending by means of the File Delivery Table. The FDT is written in XML (eXtensible Markup Language) and is described by several elements and attributes. The FDT is sent together with the files and there is, at least, one FDT per FLUTE channel. Fig. 1 shows an example of an FDT. The RFC of FLUTE defines the different elements and attributes of the FDT (some of them are mandatory and others are optional) and establishes that it is possible to define new elements and attributes for specific purposes. This is particularly useful in order to offer personalized services, as we will see in Section 4.4.

```
<?xml version="1.0" encoding="UTF-8"?>
<FDT-Instance
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:fl="http://www.comm.upv.es/flute"
xsi:schemaLocation="http://www.comm.upv.es/flutefdt.xsd"
Expires="2890842807">
  <File
      Content-Location="http://www.comm.upv.es/file1.mp3"
      TOI="1"
      Content-Length="1258291"
      Content-Type="audio/mp3"
      FEC-OTI-FEC-Encoding-ID="3"
      FEC-OTI-Maximum-Source-Block-Length="64"
      FEC-OTI-Encoding-Symbol-Length="1428"
      FEC-OTI-Max-Number-of-Encoding-Symbols="1000"
      Content-MD5="+VP5IrWploFkZWc11iLDdA==">
  </File>
</FDT-Instance>
```

Fig. 1 Example of an FDT

As a summary, Fig. 2 shows an example of file delivery using FLUTE.

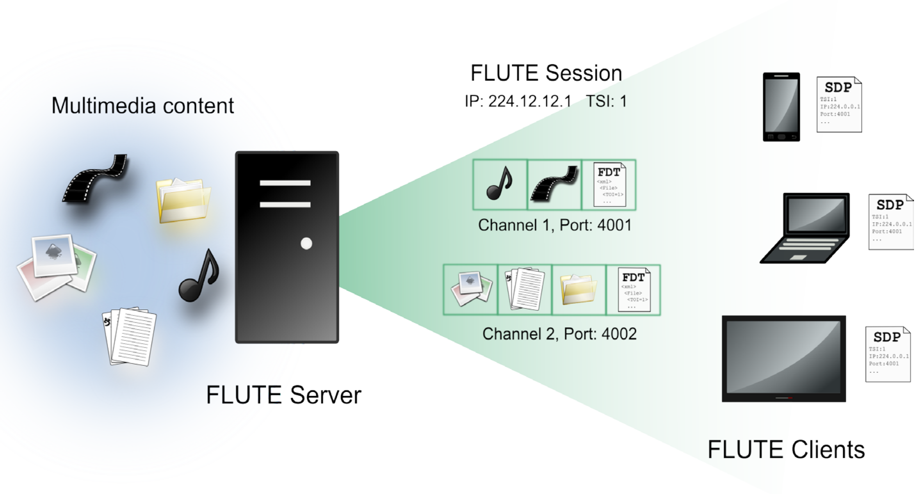


Fig. 2 File delivery using the FLUTE protocol

On the other hand, FLUTE uses different mechanisms for the reliable delivery of files, the two main are the use of carousels (where files are sent cyclically on a seamlessly endless group) and AL-FEC protection mechanisms against errors. AL-FEC mechanisms allow to reduce the number of retransmissions needed to rebuild the file, thus reducing the download time [33]. In this way, an appropriate configuration of the AL-FEC block can help to reduce additional investments in infrastructure [34]. In hybrid environments, which also offer a bidirectional link, AL-FEC can be combined with unidirectional repair mechanisms [35] or can be configured in a dynamic manner [36].

4. System overview

This section explains the overall architecture and functioning of the system proposed and how the conceptual ideas presented are translated into a technical implementation. The system has been designed with the objective to deliver scalable personalized contents to the users according to their context. To this end, we have followed the specifications of the FLUTE protocol and we have added a dynamic contextualization of contents in order to provide personalization and adaptation of contents to each user.

As a running example, we are going to use a scenario of personalized multimedia contents in a shopping center domain. This example describes a scenario where a user (e.g., Bob) that likes running, playing paddle, and fashion, is in a shopping center, and multimedia contents are delivered to him. For instance, when he enters and passes in front of a sport shop, content about a sale in shoes for running is delivered to him. This scenario is further described in Section 5. Scenarios like this can be found in the literature of location-based services and context-aware mobile computing [37, 38].

First the design considerations and the architecture are explained. Then, the building blocks that form the server side are presented, as well as the client side. Finally, the implementation of the system and how it works is explained.

4.1 Architecture

Fig. 3 represents a visual scheme of the main goal of this proposal: the use of context information to offer scalable delivery of multimedia content and adapting this content to each user.

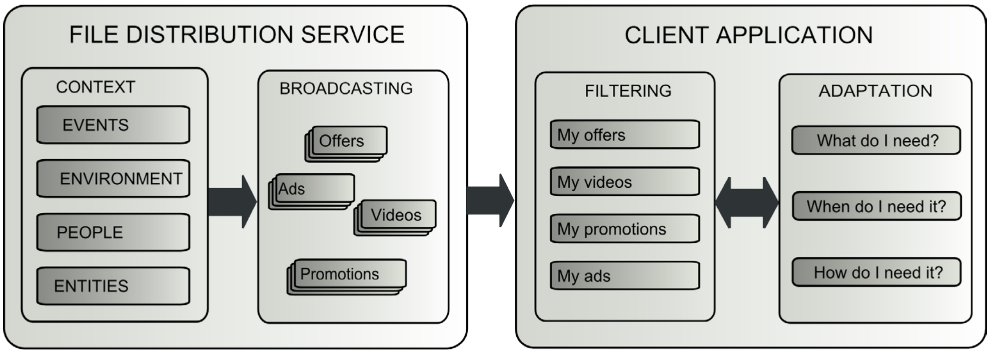


Fig. 3 Scheme of the system proposed

As figure depicts, context information is used for (1) deciding which information is delivered and (2) configuring the file delivery service in order to give more priority to the most relevant files. Among all the contents sent, the client application only downloads those contents that better fit the user preferences (*what do I need?*) –such as personal interests– and his/her context (*when do I need it?*) –such as geographical position, calendar information, time, etc.–. Also, the application interface is adapted to the user situation (e.g. with company, alone, busy) in order not to disturb the user (*How do I need it?*). Thus, context information is used directly in the whole content delivery process.

First of all, users register in the system so as to inform the server about the arrival of a new user. The framework allows the file distribution service to deliver a catalogue of contents to an unlimited number of users in an efficient way. The type of contents sent is heterogeneous, such as videos, documents, images, audios, etc. In the example shown in Fig. 3, the server sends offers, adds, videos and promotions. In this step, context information is used to estimate the utility of each content. Then, the catalogue of contents is delivered by means of a data carousel, where contents are sent cyclically. The delivery period within the carousel is not the same for all contents, but it is determined by the context information. In this way, the most "useful" files (for instance, the most popular files) are sent more frequently (shorter cycles), so clients who want to download one of the most useful files have to wait less time until that file is transmitted (a lower waiting time). Providing a contextual scalable delivery of multimedia contents is a novel contribution of this article.

On the other hand, the distribution system uses AL-FEC codes in order to reduce the download time. Specifically, LDPC (Low Density Parity Check) [39, 40] codes are used. These codes, supported by FLUTE, provide a good trade-off between performance (download time) and complexity (time required to generate parity and time required to do the decoding process) [41].

In this way, the architecture proposed is built over two different servers and a client application. The architecture has been defined in a modular and layered way in which each building block addresses a different issue. Fig. 4 shows the building blocks that compose the system as well as their interdependences. In the figure, we can see three main blocks: the content server, the content delivery server and the client. These building blocks are further explained in the following subsections.

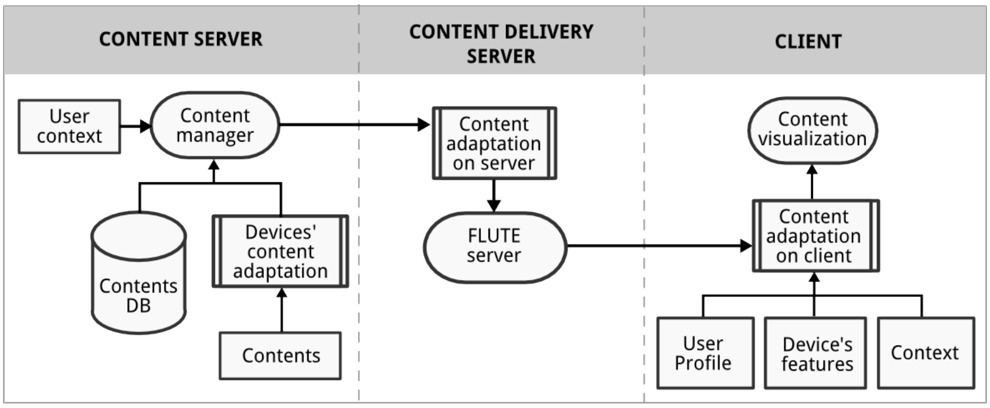


Fig. 4 Architecture overview

4.2 Server side

First, the *content server* manages and provides the different contents and the available metadata. The metadata represents the set of attributes characterizing the content and is used to determine the appropriateness of the content for the user. This metadata is described with keywords (pairs of property-value). For example, a content about a sale of sport shoes can contain the keywords:

Content(s) = (Category: sports, shoes, men; Store: SportLife)

Fig. 5 Example of keywords

For each content, this block stores in a database the information needed in order to access both the content and the associated metadata. This information is related to the users' context (e.g., location, time, etc.) and it is used by the *content adaptation* module so as to choose if a certain content is going to be sent or not depending on the users of the system. In this way, the *content adaptation* module adapts the contents to the different reception devices.

User's context is represented using an ontology-based context model [42] based on the SOUPA (Standard Ontology for Ubiquitous and Pervasive Applications) ontology. The coarse-grained concepts that we identify in our ontology are: Environment, System, Person, Policy, Time, and Event. Dividing them into finer-grained concepts, we obtain the classes needed to define the context information. An ontology-based approach for context modeling lets us to describe contexts semantically and share common understanding of the structure of contexts among users, devices, and services. The main benefit of this model is that it enables a formal analysis of the domain knowledge, such as performing context reasoning using first order logic. Thus, for example if a person wants to see a movie on Friday, and on Friday he/she only watches comedies, the system can select only comedies if the current day is Friday from the list of contents.

Finally, the *content manager* sends the most relevant information for the user's context to the *content delivery server*. Taking advantage of contextual information in this stage of the process constitutes a contextual pre-filtering where the context information is used to select a subset of relevant content for that specific user's context [28]. Following the diagram in Fig. 6, we start with all contents that have not been filtered within the matrix *User x Item x Context x Device*, where *User (U)* indicates the user preferences, *Item (I)* indicates the characteristics of the content (keywords), *Context (C)* specifies the contextual information associated with the user, and *Device (D)* specifies the characteristics of the device; and end up with a list of contextual contents for each user's device within the *User x Item* space (only remains those contents that are relevant for the current context of the user and supported by the user's device). One advantage of doing a pre-filtering of the content is that it allows relevant files to be sent to users in a short period time without overloading channels.

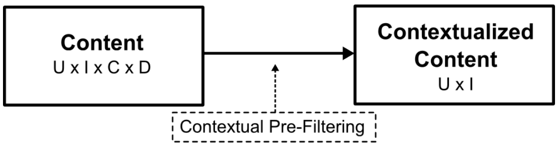


Fig. 6 Contextual pre-filtering paradigm

Finishing with the server side, the *content delivery server* is in charge of delivering the files by means of the *FLUTE server*. It should be noted that the *content adaptation* module on server receives the most appropriate files for the user's context at the moment from the *content server*. Specifically the *FLUTE server* is in charge of creating the corresponding session and channels. Once both sessions and channels are created, contents are assigned to each channel and the FDT of each channel is generated. Finally, the *FLUTE server* carries out the segmentation process by splitting the file into source and parity packets. The latter are generated using AL-FEC protection mechanisms. Contents are sent using file carousels. These carousels are dynamic, so the files sent and their metadata can change during the transmission.

4.3 Client side

As we have seen, in the server is carried out a contextual pre-filtering in order to send only the contents that result interesting for the users in the system for a certain moment. For instance, if in the shopping center there are no clients interested in sport contents, the server may not send contents related to sport. However, it results necessary to carry out a "fine" filtering in the client side. This second filtering allows showing to clients only those interesting contents according to their profile. This is done in the client side since the user profile is stored in the user's mobile device. This is a good solution to avoid overwhelming users with lots of contents (of which only a minority of them are interesting for users) and to increase the Quality of Experience. In this way, the second filtering results necessary. If only the first filtering was carried out, we could found two main problems: first, the information received by the clients could be excessive (which would get worse their Quality of Experience), as previously mentioned; second, the server, in order not to consume an excessive bandwidth, would send only the most popular contents according to the preferences of all users of the system without taking into account any specific user profile, therefore some clients would not receive contents related to their profile since their preferences would not match the preferences of the majority (which would get worse the Quality of Experience of these clients).

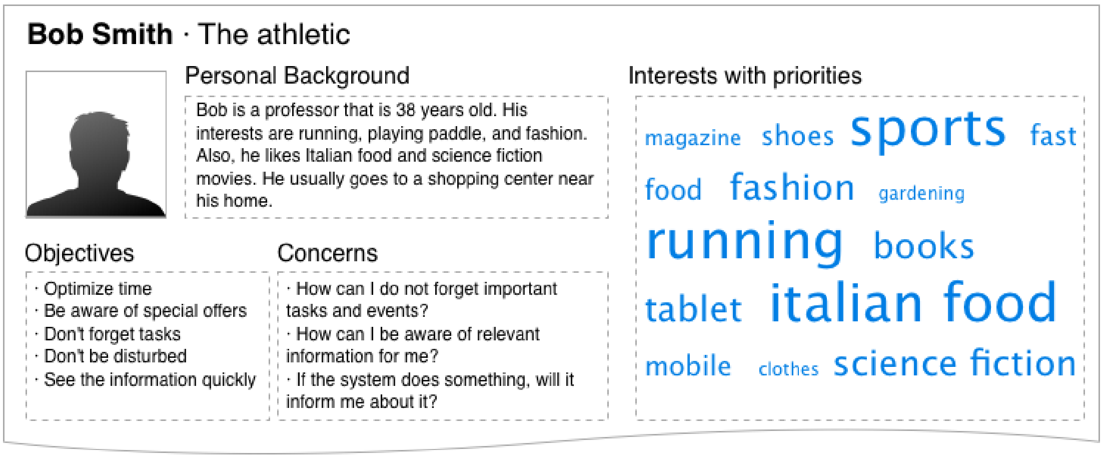
In the architecture presented, the *client* block receives all the contextual contents from the FLUTE server and filters this information according to the user's profile and preferences in order to personalize the contents for each user. Thus, in this step, the contextual contents are processed to decide whether these contents are downloaded or not. Also, this block filters the most interesting contents according to the devices' current features (e.g., memory size) and changes in user's context (e.g., user location, time). Finally, the personalized contents are downloaded automatically in the user's device. In this way, the access time to some contents (the most appropriate for each user) is minimized, thus improving the QoE.

Content adaptation is carried out by means of two mechanisms. On the one hand, a pre-caching is used, downloading useful contents for users without the need of being requested by the application. In order to determine the utility of files for a user and manage the local storage capacity, content-based filtering techniques are used [27, 43, 44]. On the other hand, the user interface is adapted to the user situation at each moment, minimizing the annoyance and therefore increasing the QoE of the user. For each situation, the system chooses the most appropriate interaction mechanisms for informing the user about a new relevant content downloaded. This adaptation is carried out by the *content visualization* module automatically using the interaction adaptation mechanisms presented in [45]. In a nutshell, this module receives the information of the downloaded content, chooses appropriate interaction resources (e.g., dialog, sound), and presents them to the user. Below, we elaborate on how the personalization and the adaptation are performed by this module, detailing the modules of *content adaptation* and *content visualization*, as shown in Fig. 4.

4.3.1 Content adaptation

In order to download the relevant (useful) content for users, the *content adaptation* module adapts the multimedia contents to each user. To that extent, it uses the information of the user's profile, device's features and context information.

The user's profile is initially created by means of user modeling techniques, using for example the "Personas" models as explained in [46]. Personas provide a framework for describing the target audience in a way useful to design systems and personalize them. In this way, content can be personalized according to these descriptions. This profiling information can be elicited from users explicitly (e.g., through questionnaires) or implicitly (e.g., learned from their behavior over time). Fig. 7 shows an example of the user profile for Bob, where personal background, objectives, concerns, and interests with their priority (representing weights) are specified. The users can change later their profile manually through an interface presented later.



Bob Smith · The athletic

Personal Background

Bob is a professor that is 38 years old. His interests are running, playing paddle, and fashion. Also, he likes Italian food and science fiction movies. He usually goes to a shopping center near his home.

Objectives

- Optimize time
- Be aware of special offers
- Don't forget tasks
- Don't be disturbed
- See the information quickly

Concerns

- How can I do not forget important tasks and events?
- How can I be aware of relevant information for me?
- If the system does something, will it inform me about it?

Interests with priorities

magazine shoes sports fast food fashion gardening running books tablet italian food mobile clothes science fiction

Fig. 7 Excerpt of a persona (user profile)

From all this information, the *content adaptation* module filters the relevant content for the user. In order to filter personalized contents, we have based on traditional content-based filtering algorithms using keywords and weights [27]. More formally, let *ContentBasedProfile(c)* be the profile of the user *c* containing tastes and preferences of the user, and let *Content(s)* be a contextual content. The "importance" of a content is determined with a *weighting* measure that is defined as follows:

ContentBasedProfile and *Content* are defined as a vector of weights ($w_1, \ldots, w_k$), where each weight denotes the importance of keyword k_i to user c and the presence of keyword k_i in content s, as Fig. 8 depicts.

$$ContentBasedProfile(c) = (wc_1, \ldots, wc_k)$$

$$Content(s) = (ws_1, \ldots, ws_k)$$

Fig. 8 Definition of the vector of *Content* and *ContentBasedProfile* based on weights

In this way, the relevancy of content s for user c is calculated by an utility function *u(c, s)* that is represented in information retrieval literature by some scoring heuristic defined in terms of vectors, such as the cosine similarity measure [43], shown in Fig. 9:

$$u(c, s) = score\ (ContentBasedProfile(c), Content(s))$$

$$u(c,s)=\cos(\vec{w}_c,\vec{w}_s)=\frac{\sum_{i=1}^{K} w_{i,c}w_{i,s}}{\sqrt{\sum_{i=1}^{K} w_{i,c}^2}\sqrt{\sum_{i=1}^{K} w_{i,s}^2}}$$

Fig. 9 The cosine value between two vectors

where K is the total number of keywords of the user. The content will be more relevant for the user if the similarity between the vectors is high (smaller angle and larger cosine value). Consequently, using the cosine will assign higher utility *u(c, s)* to those contents s that have higher similarity with the user profile of user *c*. Finally, if the utility exceeds a threshold level the *content visualization* module downloads it automatically and informs the user using the most convenient notification mechanisms; otherwise it is only added to the list of contextual contents. For example, in the case of the content of Fig. 5, that content is relevant for Bob because he is a *man* interested in *sports* and *shoes* and according to his profile these keywords have a high weigh. Thus, this content will be selected to download.

It should be noted that, apart from this automatic download mechanism, the user is able to download contents manually, by consulting a list of contents available in the channels in which the user is connected.

4.3.2 Content visualization

Then, in order to select the appropriate interaction mechanisms and minimize the intrusiveness when notifying the user about downloaded content, AdaptIO is used [47]. AdaptIO is a mobile adaptation framework that adapts service interaction obtrusiveness at runtime. It is a model-based approach where a service designer declaratively specifies the service's interaction adaptation behavior in knowledge models. In particular, there is an *obtrusiveness model* that contains a state machine. Each state corresponds to a degree of obtrusiveness (e.g., totally aware, slightly-appreciable, invisible), and the guard conditions of the state transitions reference a user situation. In Fig. 10 we show an example of the state diagram for Bob.

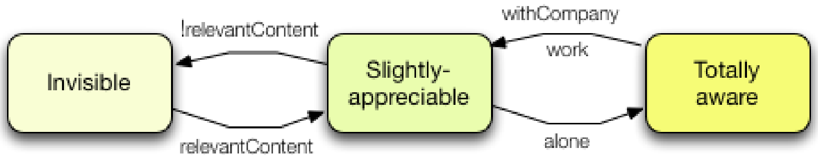


Fig. 10 Example of the obtrusiveness state diagram

This obtrusiveness state diagram specifies that when the user arrives at work or is with company, the service passes to the *slightly-appreciable* state, thus reducing notification obtrusiveness when the user is working or not alone. When the system determines that the user is no longer working or alone, the service notifications go back to the *totally aware* state, increasing the attentional demand. If the system determines the content is not relevant for the user, the service passes to the *invisible* state, making sure the user is not disturbed since the content is irrelevant.

Furthermore, each obtrusiveness state is supported by the appropriate interaction resources. Specifically, the designer associates interaction resources with one or more obtrusiveness states (e.g., totally aware: dialog, sound; slightly appreciable: status bar, vibration; invisible: list). This knowledge model is used by the *content visualization* module to adapt the interaction according to the user situation.

Back to the block diagram shown in Fig. 4, the blocks that form the *content visualization* module are explained in Fig. 11. In particular, the *reconfiguration engine* determines which interaction resources should be used for informing about the relevant content downloaded, based on the user's context and the obtrusiveness state diagram. When alerted by the *content adaptation* module about a new relevant content, the engine consults the *context* to check the current user context and calculate the interaction resources according to it.

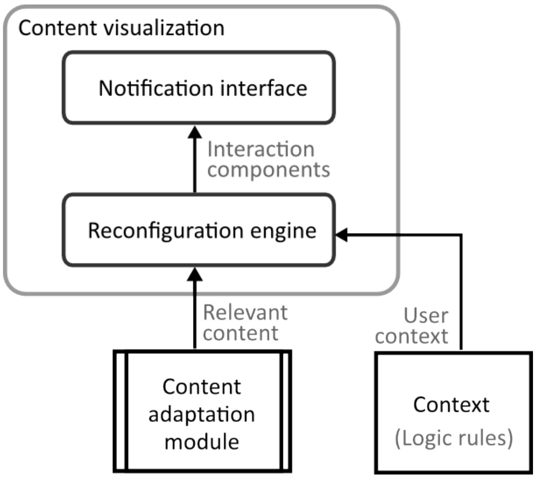


Fig. 11 Elements of the *content visualization* module

Specifically, the current user situation is calculated by means of logic rules that reason over the context information using the Androjena general-purpose rule engine [48]. For example, the logic rule to infer if the user is accompanied, is defined as Fig. 12 shows:

```
[withCompany: (?user rdf:type pros:Person)
              (?user pros:usersInLocation ?location)
              (?person1 rdf:type pros:Person)
              (?user pros:knows ?person1)
              (?person1 pros:usersInLocation ?location)
              (?user pros:socialRelationships ?person1)
              ->
              (?user pros:currentSituation pros:withCompany)]
```

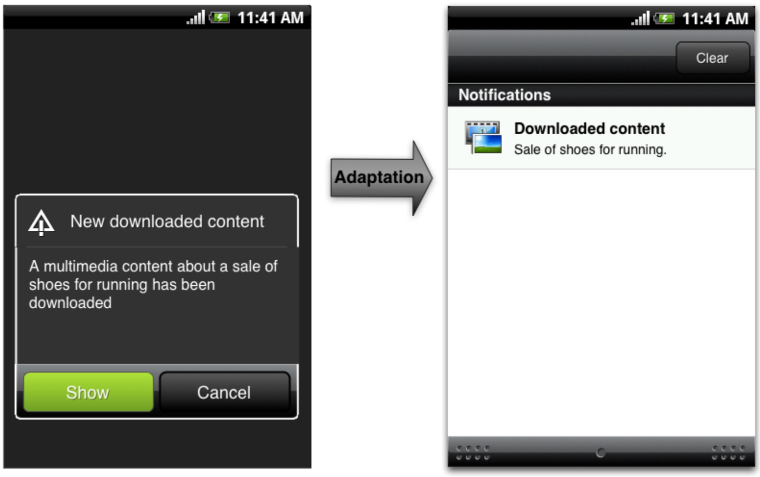
Fig. 12 Logic rule to check if the user is with company

This rule means that if two users are in the same location and have a social relationship, it implies that are together. Thus, depending on the user situation, the reconfiguration engine selects the appropriate interaction resources to notify the user. The adaptation is based on adaptation rules over the obtrusiveness diagram (see the transitions in the diagram) that indicate when the interaction should be reconfigured. An example of an adaptation rule is defined in Fig. 13:

T(totally aware)$_{withCompany}$ = {Slightly-appreciable: *Banner*, *Vibration*}

Fig. 13 Adaptation rule to notify the system in the slightly-appreciable level of attention

This means that when the user is with company, the system must notify the user in the slightly-appreciable state by using the *Banner* and *Vibration* mechanisms. In this way, when the engine detects a new user situation, it triggers the associated adaptation rule and adapts the interaction. This adaptation entails the activation of other interaction resources and the deactivation of the current ones. For example, if the user is alone, the interaction mechanisms to be used will be a dialog with sound. However, if the mobile detects that the user is with company, it will reconfigure the interface and use a banner and vibration. Finally, back to Fig. 11, the *notification interface* displays the notifications to the user, employing the suitable interaction components obtained from the *reconfiguration engine*. Fig. 14 shows a snapshot of the mentioned adaptation. It is worth noticing that only the relevant content downloaded is notified to the user. The specific operations that follow the *reconfiguration engine* to calculate the appropriate interaction resources are described in [45].



New downloaded content

A multimedia content about a sale of shoes for running has been downloaded

Downloaded content

Sale of shoes for running.

Fig. 14 Example of interaction adaptation to inform the user

4.4 Implementation

In this section we elaborate on the implementation and some working details of the main building blocks that form our system.

4.4.1 File delivery service

The implementation of the file delivery service has been developed according to the specifications of the RFC of FLUTE [32], and includes a coding/decoding module that implements LDPC codes. That implementation has been developed by the authors of this paper, presented in [33]. The implemented server allows the creation and edition of sessions and channels as well as the assignment of the files to the corresponding channel. Also, the application allows to associate metadata to each file sent. Thus, in each channel, apart from the files, the FDT is sent. As mentioned, the structure of the FDT is flexible, according to the specifications of the RFC 6726. In this way, it is possible to add new elements and attributes to the FDT (by means of new labels) in order to include additional information regarding the contents sent. In fact, the standard DVB-H [24] proposes the use of an additional "group" field in the FLUTE FDT to enable logical grouping of related files. This is useful, for instance, in software update packages (which are usually composed of several files) or in web pages (which are usually linked to each other). In this sense, in this paper we propose the use of a general attribute called "Metadata", which contains all the additional metadata of the file in a format dependent on the application, shown in Fig. 15. Therefore, only receiver applications interested in the metadata of the file would have to interpret these data. Thus, this label could contain several attributes, and receivers could parse the metadata as long as they know the structure of the label. In the structure proposed, the name of each attribute is followed by colon, the elements of each attribute are separated by commas, and the attributes are separated by semicolons.

```
<Metadata> ID1:value1,value2,value3; ID2:value1; … </Metadata>
```

Fig. 15 Format of the proposed element “Metadata”

In this way, using the proposed format for the example of the shopping center aforementioned, we could define, for instance, an identifier called “Category” to specify the category (or categories) in which a certain content is classified, and another one called “Store” to indicate the related store of the content sent (see Fig. 16).

```
<Metadata> Category:sports,shoes,men; Store:SportLife </Metadata>
```

Fig. 16 Example of the proposed element “Metadata”

4.4.2 Client application

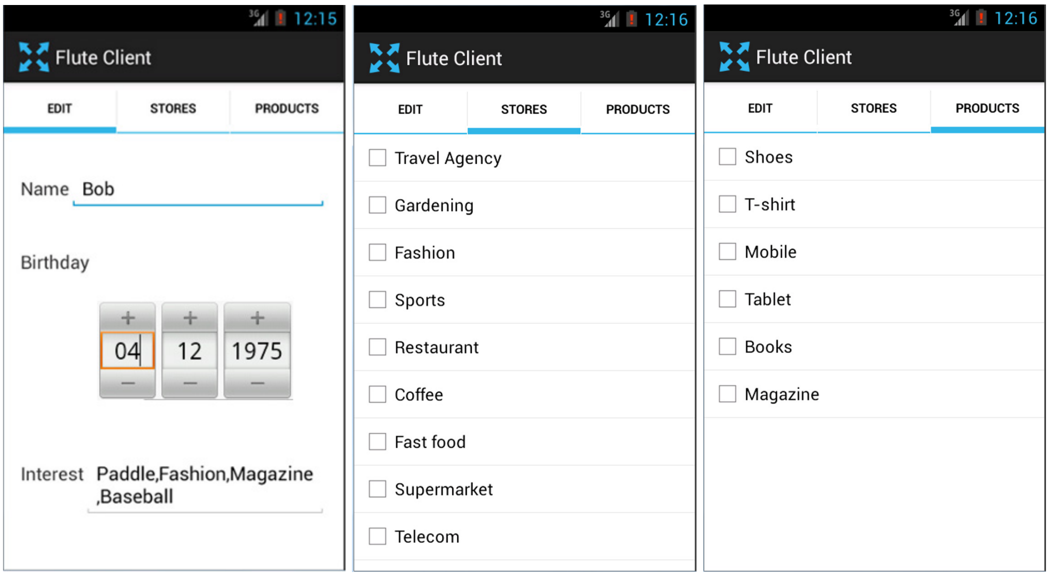
Regarding the mobile application, the client has been developed on the Android platform. Android provides an open application framework that supports advanced interaction techniques and easy communication mechanisms to integrate functionality of applications. The *user’s profile* module is implemented as an Android service that stores the information about the users such as age, sex, interests and likes, and preferences such as favorite shops and products. Also, each property has a weight that indicates the importance of that property for the user. This weight is used later for the filtering of the contents.

With regards to the *context* module, it detects context changes from the mobile sensors. The mobile services that control the virtual sensors are implemented as background services running on the mobile device that monitors changes in their own device sensors. For example, the user location is detected by the GPS service that controls the mobile GPS sensor. The context is represented by using an ontology-based context model based on the Web Ontology Language (OWL) [49]. OWL is a W3C standard that provides a markup language that greatly facilitates knowledge automated reasoning. The ontology-based context model used in this work is described as a collection of triples in the form of (*subject, predicate, object*). For example, (*Bob, personLocatedIn, Cinema*) means that Bob is located in the cinema. This context model keeps the updated context information to reason about the user’s context. A detailed description of this ontology is provided in [42].

Finally, it has been implemented a FLUTE client application in Android to receive contents. This client includes an LDPC module to decode the packets received. The client application is able to connect to different FLUTE sessions and channels, to parse the FDT of each channel and to download multimedia contents.

4.4.3 End-user interfaces

This section presents the mobile application developed. First, the user can change the values of his/her profile (properties and weights) by means of a mobile user interface. Fig. 17 shows some snapshots of the available screens that allow the end user to change his/her profile.



Name Bob

Birthday 04 12 1975

Interest Paddle,Fashion,Magazine ,Baseball

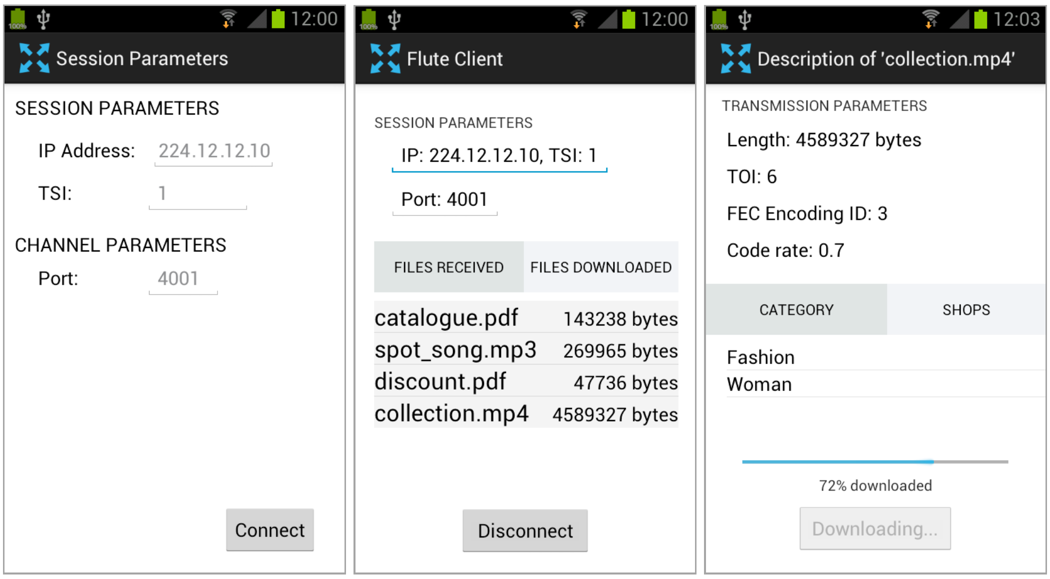
□ Travel Agency
□ Gardening
□ Fashion
□ Sports
□ Restaurant
□ Coffee
□ Fast food
□ Supermarket
□ Telecom

□ Shoes
□ T-shirt
□ Mobile
□ Tablet
□ Books
□ Magazine

Fig. 17 Screens to change the user's preferences

As shown in the figure, the user can edit his/her personal information and interests (see first screen), his/her relevant stores (see second screen), and the products he/she is interested in (see third screen).

On the other hand, regarding the connection to a certain session or channel, the connection can be manual or automatic. In the second case the application can connect automatically to those channels in which the client is interested (according to the users' preferences specified). For instance, if there is a FLUTE channel dedicated to send information about a certain shop and the user has specified that he/she is interested in that shop, the application connects automatically to that channel. Once the connection is established, the available files in the channel are shown to the client, who is able to decide the contents to download. When a file is downloaded, the client receives a certain notification according to their context and the client is able to consume that content. Fig. 18 shows some snapshots of the application: the first screen allows the user to connect manually to a certain session and channel; the second shows the files sent in a certain session and the files downloaded; after selecting a certain file, the third screen shows, for the file selected, the parameters related to the FLUTE transmission (such as the coding type or the code rate), the metadata of that file (in the example, metadata referred to category and shops) and the progress of the download.



Session Parameters

SESSION PARAMETERS

IP Address: 224.12.12.10

TSI: 1

CHANNEL PARAMETERS

Port: 4001

Flute Client

SESSION PARAMETERS

IP: 224.12.12.10, TSI: 1

Port: 4001

catalogue.pdf	143238 bytes
spot_song.mp3	269965 bytes
discount.pdf	47736 bytes
collection.mp4	4589327 bytes

Description of 'collection.mp4'

TRANSMISSION PARAMETERS

Length: 4589327 bytes

TOI: 6

FEC Encoding ID: 3

Code rate: 0.7

Fashion

Woman

72% downloaded

Fig. 18 Snapshots of the application

5. Evaluation

This section presents the evaluation of the system proposed. Two different evaluations have been carried out: an objective assessment where the access time to the contents is analyzed; and a subjective user evaluation that analyzes how the application adapts the contents to the user preferences by using a subjective survey. This survey captures the opinion of the users that have tested the developed application.

The objective of this evaluation is to analyze the Quality of Experience of users, a parameter that measures the satisfaction of users with the system. In content download services a good QoE is obtained when clients receive the contents without losses, with the minimum download time and offering a personalized service. In this sense, the use of the appropriate configuration parameters in FLUTE transmissions (such as the AL-FEC parameters) allows to minimize the channel losses and to reduce the download time, and this is what the objective evaluation presented in Section 5.1 analyzes. On the other hand, Section 5.2 presents the subjective evaluation, which analyzes the user satisfaction with the content personalization carried out by the system proposed.

To perform the different evaluations, we used an experimental setup that included: on the server side, an Intel Core i3 2100 PC, 3.10 GHz processor and 4GB RAM with Windows XP; and on the client side, the client application was running on a LG E400 mobile device with Android operating system 2.3 version, and on a Samsung Galaxy S2 with Android operating system 4.0 version. It should be noted that, in the scenario used to carry out the evaluation (which is detailed in the following section), in all empirical measurements carried out we have not perceived any computational problem in the mobile receivers.

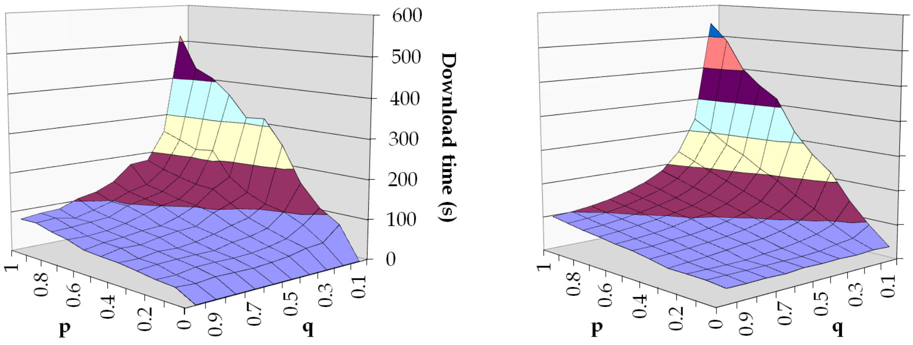
5.1 Objective evaluation

The objective assessment is carried out by evaluating the efficiency of the transmission protocol (in this case, FLUTE). To that extent, we calculate the download time of a file in environments with different losses. Specifically, we evaluate the advantages of the use of FEC mechanisms, showing that there is a code rate that minimizes the download time according to the channel losses.

The measures have been carried out in a controlled environment, simulating the losses in the channel with the two-state Markov model (also known as Gilbert model) [50]. This model is widely used in the literature and simulates well the burst losses, typical in wireless networks. The two-state Markov model establishes that the probability of losing a packet depends on whether the previous packet has been received or not.

On the other hand, in the studies we consider that clients download a particular file, which size is over 4 MB (that is, 3000 FLUTE packets with encoding symbol length of 1428 bytes). This size corresponds to a short video of quality 480p or a song of 3-4 minutes coded at 160 kbps, which are typical contents of the scenario presented in Section 5.2. Moreover, a transmission rate of 5 Mb/s is used, although the conclusions obtained are independent on the transmission rate. Among the two LDPC structures supported by FLUTE, that is, LDPC Staircase and Triangle, the studies consider LDPC Staircase codes. Finally, the number of measurements accomplishes good 99% confidence intervals in all scenarios.

The first study shows the advantages of using AL-FEC mechanisms. In AL-FEC, the parameter used to determine the amount of protection to a file is called code rate, which is in range [0, 1]. The less the code rate the more the protection. Fig. 19 shows the download time of a file based on the channel losses for two different FEC encodings: No-FEC (that is, no FEC protection is used) and LDPC Staircase (with a code rate of 2/3). It should be remembered that, in the Markov model, *p* indicates the probability that a packet is lost when the previous one was received, and *q* indicates the probability of the opposite transition.



(a) No-FEC

(b) LDPC Staircase (code rate = 2/3)

Fig. 19 Download time depending on coding, with 3000-packet size and transmission rate=5 Mb/s

Noting the scale of each graph, we can clearly see the convenience of using coding (for high losses the download time without FEC is more than four times the one obtained using LDPC Staircase). The tendency is the same in both codes, but the difference between them is higher when the losses increase. In low-loss environments (that is, when *p* is low and *q* is high), the graphs show that the download time using LDPC does not increase considerably, whereas when no coding is used the download time grows fast with a slight increase of the losses.

When AL-FEC is used, the download time depends greatly, apart from the channel losses, on the code rate used. In this way, in channel with high losses it is recommended to use a high protection (that is, low code rates) in order to recover the large amount of packets lost. As we have mentioned along the document, the proposal is based on multicast networks where contents are delivered to all users, thus saving bandwidth and providing scalability. In this way, there is no feedback mechanisms from clients to the server in order to request for lost packets, thus there is no problems regarding the congestion or latency due to retransmissions. Instead, clients can wait to the following transmission of the file in the carousel in order to recover it. Therefore, if not enough protection is provided to the transmission, clients have to wait several transmission carousel cycles to complete their downloads, thus increasing the download time. On the other hand, if too much protection is used in channel with low losses too much parity packets are sent through the network, which provokes that the download time increases. Fig. 20 shows a comparison among different code rates for LDPC Staircase codes, evaluating how channel losses affect the download time. In order to simulate a typical wireless channel, different channel losses between 0% and 30% (in steps of 5%) have been simulated. We have considered these percentages of losses since clients not excessively far from the access point hardly perceive losses higher than 30%. Also a 50% losses channel has been simulated to see the general tendency.

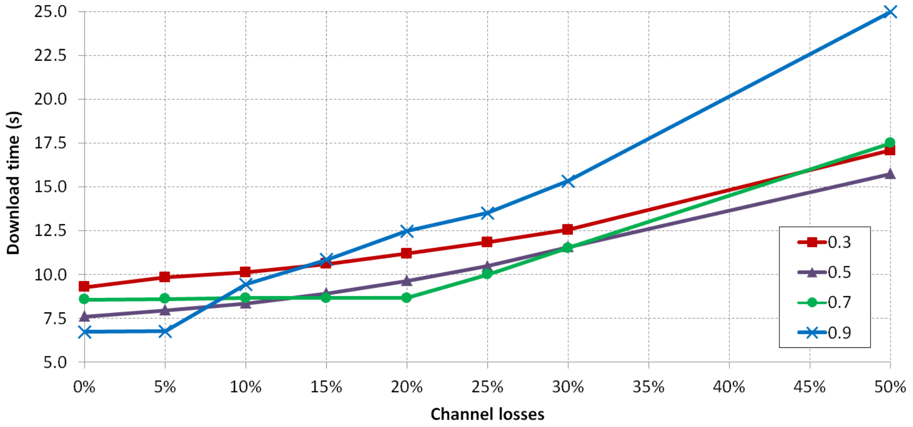
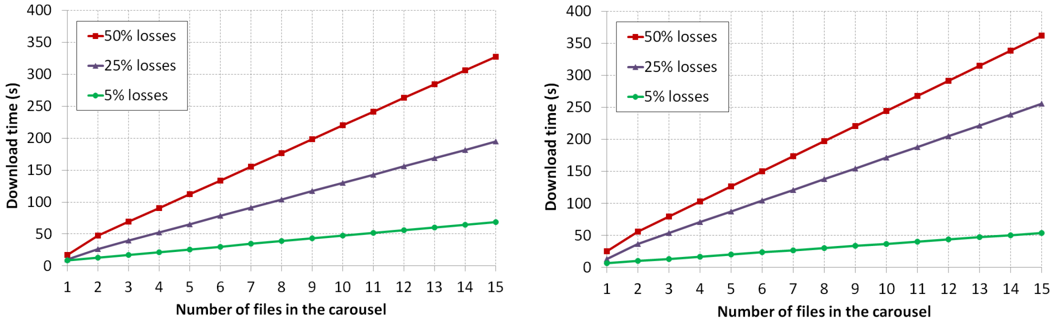


Fig. 20 Evaluation of the code rate of LDPC Staircase, with 3000-packet size and transmission rate=5 Mb/s

As we can see, the code rate that minimizes the download time depends on the losses. In this way, in channel with low losses the optimum code rate is 0.9 (few protection), whereas when losses are very high the code rate that minimizes the download time is 0.5 (high protection). For medium losses, the best code rate is 0.7, which behaves rather well for a wide range of losses. Also, when choosing the code rate it must be taken into account the bandwidth increase associated to each code rate. Using a high protection (low code rates) increases considerably the bandwidth. For instance, a code rate of 0.5 (one parity symbol for each source symbol) implies increasing the bandwidth a 100%, whereas the code rate of 0.9 produces an overhead of only 11%. Therefore there is a trade-off between the bandwidth and the download time. If the server has enough bandwidth, a good solution to reduce the download time of all clients would be to use two different FLUTE sessions, each one containing channels which transmit with code rates 0.7 and 0.9. In this way, depending on the losses perceived, clients would connect to those channels that send information with the code rate that better fits their losses. In this sense, a similar proposal is presented in [51], where it is proved that a file delivery service using only these two code rates provides a very good performance, since it offers low download times with a reduced bandwidth increase.

In sum, an appropriate configuration of the FLUTE transmission parameters can help to reduce the download time. This is very important when transmitting several files within the same carousel, since using too much protection increases the carousel size whereas, as mentioned, using insufficient protection provokes that clients have to wait several carousel cycles. In this sense, Fig. 21 shows the download time of a certain file for a different number of files within the carousel. The studies presented consider two code rates: 0.7 and 0.9. As we have seen in Fig. 20 and it is shown in [51], these code rates behave rather well for a wide range of losses.



(a) CR = 0.7

(b) CR = 0.9

Fig. 21 Evaluation of the number of files in the carousel, with files of 3000-packet size, coded with LDPC Staircase and transmission rate=5 Mb/s

Results reflect that the download time increases linearly with the number of files in the carousel. For 5% losses the code rate 0.9 provides lower download times, whereas code rate 0.7 results more convenient when losses are higher, as we saw too in Fig. 20.

Nevertheless, an excessive download time could not be affordable in some scenarios. In this sense, there are different solutions to minimize this problem:

- First, if the transmission rate increases, the download time decreases. However, in some scenarios it is not possible to increase the transmission rate. Also, some devices can have problems when receiving packets at higher rates.
- On the other hand, as we have seen in Section 3, the transmission in FLUTE is based on sessions and channels. Thus, it is possible to define several sessions and channels, each one containing different kinds of contents. For instance, a certain channel can contain contents related to sports whereas another channel can send fashion contents. Therefore, clients only connect to the channel in which they are interested. In this way, the number of files in each carousel is lower, thus decreasing the carousel size and therefore the download time.
- Finally, cache mechanisms represent a good solution to reduce the download time. In fact, the use of caches allows users to consume contents immediately, without waiting. Contents are downloaded before the client requests for a certain file. When using content pre-caching, the key element is the policy used to determine which contents should be downloaded. There are certain parameters to consider when deciding this, such as the user preferences or the context use. It should be noted that this pre-caching does not have to become a bottleneck in a typical content transmission in wireless networks. If the client perceives low or high losses the performance of the pre-caching is not affected, since packets will arrive to the clients independently if the client downloads or not those packets or uses a pre-caching mechanism. In this sense, next subsection evaluates the performance of these parameters.

5.2 Subjective evaluation

In order to evaluate the content personalization that our system offers to users, we conducted a user test. This test examines the user satisfaction with the system to check whether users are satisfied with the automated delivery of multimedia contents according to their preferences and context. To conduct the experiment, we have applied a case study based evaluation by following the research methodology practices provided by Runeson and Höst [51]. Specifically, the case study proposed illustrates how personalized contents are delivered to users according to their preferences in the context of a shopping center. The system takes into account user's preferences and the needed context information such as location, time, etc. All these factors have an effect on the delivery of personalized multimedia content. Next, we present the design of the case study, the procedure followed during the experiment and the results obtained.

5.2.1 Design of the experiment

The experiment is based on a case study that gives support to several scenarios of a user in a shopping center. This case study extends the running example used throughout the paper and it is described as follows: Bob is a professor that is 38 years old. His interests are running, playing paddle, and fashion and he likes Italian food and science fiction movies. Usually, he goes to a shopping center near his home. When he is near the shopping center, he receives multimedia content about new store openings. However, these contents are not automatically downloaded because they are not relevant for the user and he is outside the shopping center. Then, when he enters and passes in front of a sport shop, content about a sale in shoes for running is delivered to the user. Due to this content is relevant for Bob, the content is downloaded automatically and notified to the user in a notorious manner (by means of a dialog and sound) because he is near the shop. Some other shops have special offers and sales; however as these contents are not interesting for Bob they are not downloaded automatically. At lunchtime, Bob meets a friend and they go to have lunch together. As it is time to lunch and Bob likes Italian food, content about a menu in an Italian restaurant is delivered to Bob automatically. However, as he is with company (inferred by a logic rule), the interaction mechanisms chosen to inform Bob are gentle (e.g., a status bar and vibration are used). In the afternoon, the movie billboard is delivered to Bob. However, only the contents about science fiction movies (such as trailers) are downloaded automatically. This time, Bob is informed only by a dialog because he is in a noisy environment.

In this case study, the personalized multimedia contents are delivered to the user according to his preferences and context. For example, the running shoes sale is downloaded automatically because this content is more relevant for Bob. However, another sale is only shown to the user but not downloaded because it is less relevant. Also, the notifications about downloaded contents are presented with different interaction mechanisms according to the user's context. For example, if the user is alone and near the shop where it is the sale, the notification is presented using a dialog with sound. In contrast, if the user is with company the notification is presented in a slighter manner. Finally, if the user is in a noisy environment the sound is not used.

5.2.2 Test setup and procedure

In this experiment a total of 12 people participated (9 men and 3 women). Their age ranges from 25 to 41. Most of them were university students and daily mobile users.

For the evaluation of the experiment, users adopted Bob's profile and performed the scenarios earlier described. The study was conducted in our laboratory, simulating the different scenarios in which the experiment was based on. In-situ evaluation was possible since we only need the mobile device to interact with the user.

To measure the user satisfaction, we used an adapted IBM Post-Study Usability questionnaire (PSSUQ) [52]. This questionnaire measures the overall satisfaction (OVERALL) of the system, its usefulness (SYSUSE), its information quality (INFOQUAL), and its interface quality (INTERQUAL). To evaluate our system, we discarded the INFOQUAL questions since it was not our goal to measure the application information quality. Moreover, we extended this questionnaire to include questions regarding the personalization of the delivered multimedia content. Thus, two subjective dimensions were evaluated: user satisfaction and content personalization. We applied a Likert scale (from 1 to 5 points) to evaluate the items of the questionnaire. Some space was left at the end of the questionnaire for further comments.

We also evaluated the accuracy of the *content adaptation* module by comparing the multimedia content that users had downloaded manually and the automatically downloaded by the module. To measure this, we showed to users all the available contents to be delivered in the shopping center context and ask them to indicate which of them they wanted to be downloaded automatically according to the Bob profile. Then, we compared the selection made by users with the selected by our adaptation module and we calculated the percentage of accuracy.

5.2.3 Results

In this section, we present the evaluation results of the case study. Overall, the experiment showed that, by following our method, multimedia contents can be delivered to the users adapted to their preferences and context with a high user satisfaction.

Fig. 22 illustrates the table of the results obtained for the user satisfaction and content personalization measures. Regarding the user satisfaction, more than 80% of users considered the system simple to use, although one user found it a little confusing. Most of them (more than 90%) also answered that the system helped them to complete the tasks and scenarios effectively and quickly. Although most users (83.3%) felt comfortable using this system, some of them found it a bit difficult to learn at the beginning. However, all of them thought they could become productive quickly once they have learnt to use the system. Some users considered the interfaces unpleasant (33.3%) and they commented that the interfaces could be improved. However 75% of users liked using the interface and they considered that the system had all the functions and capabilities they expected for the scenario. Overall, most of them were satisfied with the system.

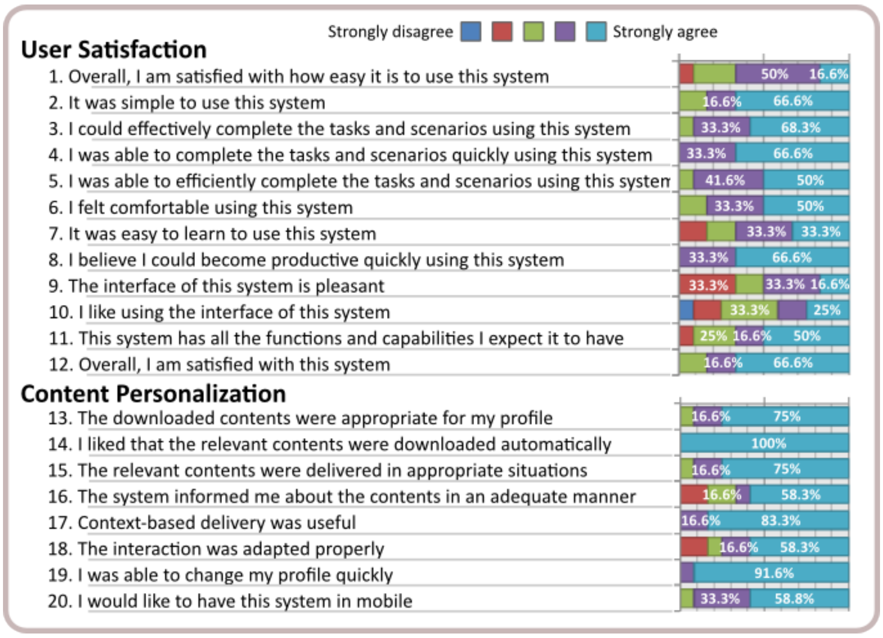


Fig. 22 Summarized results about user satisfaction and content personalization

Regarding the content personalization, more than 90% of users considered that the downloaded contents were appropriate for their preferences and they were delivered in appropriate situations. Moreover, all of the users liked the automatic download of the relevant contents. However, some users complained because they would like to be informed in another way (16.6%); for instance, some of them said that they would prefer not to be informed with sound. These users also considered that the interaction was not adapted properly. In spite of this, most users considered the context-based delivery was useful. Also, all users could change their profile quick enough. Finally, most of the users commented that they would like to have this system in their mobile.

With regard to the accuracy of the *content adaptation* module, Fig. 23 shows the results obtained. A total of 15 multimedia contents were delivered to the user through the case study. 6 users (50%) would have downloaded the same contents that the system downloaded automatically (four contents). Four users selected one more or one less, and one user selected two more relevant for him/her. Finally, one user downloaded three more contents. In this evaluation we can see that users tend to download more contents than the selected by our *content adaptation* module. Increasing the weights for selecting the contents can solve this issue.

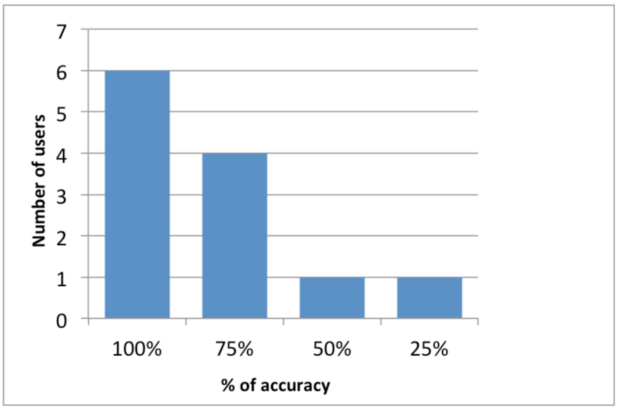


Fig. 23 Accuracy of the *content adaptation* module

6. Conclusions and future work

This paper has presented an architecture for scalable distribution of multimedia content in multicast wireless networks. The system offers a personalized service by considering both the user preferences and their context in order to deliver personalized contents to users. This system allows the content server to deliver the contents to an unlimited number of users minimizing the annoyance of new contents and increasing the QoE.

In the evaluations presented, we have seen the convenience of using AL-FEC error protection mechanisms and that it is recommended to estimate the channel losses when deciding the best code rate. This can help to reduce considerably the download time in reception. It should be noted that the download time is one of the main parameters that determine the Quality of Experience in a content download service. Also, the use of caching mechanisms it is highly recommended to reduce the download time.

On the other hand, we have evaluated the adaptation of the contents sent to the users in terms of user satisfaction and content personalization as well as the accuracy of the content adaptation module. To that extent, we have measured the subjective quality by means of the users experiences with the system. In this way, the subjective results have shown the validity of the system proposed for users.

For future work, we plan to extend the system in order to deliver the contents by means of different devices of the user's environment (not only their mobile phones). In this way, we want to calculate relevant contents for a group of users since it could be possible to show multimedia contents on a screen where multiple users are looking at.

On the other hand, in the caching process, when deciding the files to download, apart from considering the utility of a certain content for the user, it should be considered the total size of the cache at the client. In this way, given a set of files with certain size and utility, it is needed to determine the combination of files that better use the available cache space, considering the total size of the cache [53]. This is referred as the 0-1 knapsack optimization problem [54]. In this sense, there are different cache replacement policies to optimize the cache, some of them are presented in [55] and [56]. The implementation of a cache replacement policy is also part of the future work.

Acknowledgements

This work is supported in part by the Ministerio de Economía y Competitividad of the Government of Spain under project COMINN (IPT-2012-0883-430000) and by the project PAID/2012/313 from the PAID-05-12 program of the Vicerrectorado de Investigación of the Universitat Politècnica de València.

References

1. AdAdge (2013) A majority of U.S. mobile users are now smartphone users. Available at: http://adage.com/article/digital/a-majority-u-s-mobile-users-smartphone-users/241717. Accessed November 2013
2. Anind KD (2001) Understanding and Using Context. Personal Ubiquitous Comput. 5:4-7
3. Guillen J, Miranda J, Berrocal J, Garcia-Alonso J, Murillo J, Canal C (2014) People as a service: a mobile-centric model for providing collective sociological profiles. IEEE Software 31(2):48-53

4. Felfernig A, Jeran M, Ninaus G, Reinfrank F, Reiterer S (2013) Toward the next generation of recommender systems: applications and research challenges. Multimedia Services in Intelligent Environments 24(Chapter 5): 81-98
5. Baldauf M, Dustdar S, Rosenberg F (2007) A survey on context-aware systems. Int. J. AdHoc and Ubiquitous Computing, Springer-Verlag 2:263-277
6. Chen YFR, Jana R, Stern D, Wei B, Yang M, Sun H, Dyaberi J (2010) Zebroid: using IPTV data to support STB-assisted VoD content delivery. Multimedia System Journal 16(3):199-214
7. Zheng Q, Zhu P, Wang Y, Xu M (2010) EPSP: Enhancing network protocol with social-aware plane. Proc. of IEEE/ACM Int Conference on Green Computing and Communications (GreenCom) and Int Conference on Cyber, Ohysical and Social Computing (CPSCom), Hangzhou, China:578-583
8. Community Research and Development Information Service (CORDIS) – Seventh Framework Programme (FP7). Available at: http://cordis.europa.eu/fp7/home_en.html. Accesed October 2013
9. ETSI TS 102 472 (2009), Digital Video Broadcasting (DVB); IP Datacast over DVB-H: Content Delivery Protocols, v1.3.1, available online: www.etsi.org/deliver/etsi_ts/102400_102499/102472/01.03.01_60/ts_102472v010301p.pdf
10. ETSI TS 102 034 (2008), Transport of MPEG-2 TS Based DVB Services over IP based Networks (and associated XML), v1.4.1, available online: www.etsi.org/deliver/etsi_ts/102000_102099/102034/01.04.01_60/ts_102034v010401p.pdf
11. ETSI TS 126 346 (2013), Universal Mobile Telecommunications System (UMTS); LTE; Multimedia Broadcast/Multicast Service (MBMS); Protocols and codecs (release 10), v11.3.0, available online: www.etsi.org/deliver/etsi_ts/126300_126399/126346/11.03.00_60/ts_126346v110300p.pdf
12. Neumann C, Roca V, Walsh R (2005) Large scale content distribution protocols. ACM Computer Communication Review 35(5):85-92
13. Fraile F, de Fez I, Guerri JC (2009) Modela-TV: service personalization and business model management for mobile TV. Proc. of 7th European Interactive TV Conference (EuroITV), Leuven, Belgium:1-6
14. Hsieh CC, Lin CH, Chang WT (2009) Design and implementation of the interactive multimedia broadcasting services in DVB-H. IEEE Transactions on Consumer Electronics 55(4):1779-1787
15. Peltotalo J, Harju J, Saukko M, Väätämöinen L, Bouazizi I, Curcio I (2008) Personal mobile broadcasting based on the 3GPP MBMS System. Proc. of MoMM, Linz, Austria:156-162
16. Liang L, Cruichkshank H, Sun Z, Kulatunga C, Fairhurst G (2010) The integration of TESLA and FLUTE over satellite networks. Proc. of the IEEE Global Telecommunications Conference (Globecom), Miami, FL, USA:1-6
17. Kutscher D, Greifenberg J, Loos K (2007) Scalable DTN distribution over uni-directional links. Proc. of the SIGCOMM workshop on networked systems in developing regions (NSDR), Kyoto, Japan: article no. 6
18. Papastergiou G, Psaras I, Tsaoussidis V (2009) Deep-space transport protocol: a novel transport scheme for space DTNs. Computer Communications 32(16):1757-1767

19. Hrvoje J, Stockhammer T, Xu W, Abdel Samad W (2006) Efficient video-on-demand services over mobile datacast channels. Journal of Zhejiang University 7(5):873-884
20. Yetgin Z, Çelik T (2012) Efficient progressive downloading over multimedia broadcast multicast service. Computer Networks 56(2):533-547
21. Peltotalo J, Peltotalo S, Harju J, Walsh R (2007) Performance analysis of a file delivery system based on the FLUTE protocol. Int Journal of Communication Systems 20(6):633-659
22. de Fez I, Fraile F, Guerri JC (2013) Effect of the FDT transmission frequency for an optimum content delivery using the FLUTE protocol. Computer Communications 36(12):1298-1309
23. Streefkerk JW, van Esch-Bussemakers MP, Neerincx MA (2006) Designing personal attentive user interfaces in the mobile public safety domain. Comput Hum Behav 22:749-770
24. Weld DS, Anderson C, Domingos P, Etzioni O, Gajos K, Lau T, Wolf S (2003) Automatically personalizing user interfaces. Proc. of the Int Joint Conference on Artificial Intelligence (IJCAI), Acapulco, Mexico:1613-1619
25. Chatfield C, Carmichael D, Hexel R, Kay J, Kummerfeld B (2005) Personalisation in intelligent environments: managing the information flow. Proc. of the OZCHI Computer-human interaction, Canberra, Australia:1-10
26. Bright A, Kay J, Ler D, Ngo K, Niu W, Nuguid A (2005) Adaptively recommending museum tours. Proc. of the UbiComp Workshop on Smart Environments and their Applications to Cultural Heritage, Tokyo, Japan:29-32
27. Adomavicius G, Tuzhilin E (2005) Toward the next generation of recommender Systems: a survey of the state-of-the-art and possible extensions. IEEE Transactions on Knowledge and Data Engineering 17:734-749
28. Adomavicius G, Tuzhilin A (2010) Context-aware recommender systems. Recommender Systems Handbook (Chapter 7): 217–253
29. Valtonen M, Vainio AM, Vanhala J (2009) Proactive and adaptive fuzzy profile control for mobile phones. Proc. of the IEEE Int Conf on Pervasive Computing and Communications (PerCom), Galveston, Texas, USA:1-3
30. Assad M, Carmichael DJ, Kay J, Kummerfeld B (2007) PersonisAD: distributed, active, scrutable model framework for context-aware services. Proc. of Pervasive Computing, Toronto, Canada:55-72
31. Korpipaa P, Malm EJ, Rantakokko T, Kyllonen V, Kela J, Mantyjarvi J, Hakkila J, Kansala I (2006) Customizing user interaction in smart phones. IEEE Pervasive Computing 5:82-90
32. Paila T, Walsh R, Luby M, Roca V, Lehtonen R (2012) FLUTE – File Delivery Over Unidirectional Transport. IETF RFC 6726
33. de Fez I, Fraile F, Belda R, Guerri JC (2011) Performance evaluation of AL-FEC LDPC codes for push content applications in wireless unidirectional environments. Multimedia Tools and Applications 60(3):669-688
34. Barquero D, Bria A (2007) Forward Error Correction file delivery in DVB-H. Proc. of IEEE Vehicular Technology Conference (VTC), Dublin, Ireland:2951-2955

35. Lohmar T, Huschke J (2009) Radio resource optimization for MBMS file transmissions. Proc. of the IEEE Int Symposium on Broadband Multimedia Systems and Broadcasting (BMSB), Bilbao, Spain:1-7
36. de Fez I, Fraile F, Belda R, Guerri JC (2012) Analysis and evaluation of adaptive LDPC AL-FEC codes for content download services. IEEE Transactions on Multimedia 14(3):641-650
37. Schiller JH, Voisard A (2004) Location-based services. Morgan Kaufmann.
38. Chen G, Kotz D (2000) A survey on context-aware mobile computing research. Technical Report TR2000-381, Dartmouth Computer Science.
39. Gallager RG (1962) Low density parity check codes. IRE Transactions on Information Theory 8(1):21-28
40. Roca V, Neumann C, Furodet D (2008) Low density parity check (LDPC) staircase and triangle forward error correction (FEC) schemes. IETF RFC 5170
41. Paolini E, Varrella M, Chiani M, Matuz B, Liva G (2008) Low-complexity LDPC codes with near-optimum performance over the BEC. Proc. Adv Satellite Mobile Systems (ASMS), Bologna, Italy:274-282
42. Serral E, Valderas P, Pelechano V (2010) Towards the model driven development of context-aware pervasive systems. Pervasive and Mobile Computing 6(2): 254-280
43. Du R, Safavi-Naini R, Susilo W (2003) Web filtering using text classification (2003). Proc. of the Int Conf on Networks (ICON), Sydney, Australia:325-330
44. Kuppusamy KS, Aghila G (2012). A personalized web page content filtering model based on segmentation. Int Journal of Information Sciences and Techniques (IJIST) 2(1):41-51
45. Gil M, Giner P, Pelechano V (2012) Personalization for unobtrusive service interaction. Personal Ubiquitous Comput 16(5):543-561
46. Serral E, Gil M, Valderas P, Pelechano V (2013) Automating unobtrusive personalized services in ambient media environments. Multimedia Tools and Applications, Springer US, available online, doi: 10.1007/s11042-013-1634-2
47. van Woensel W, Gil M, Casteleyn S, Serral E, Pelechano V (2012) Adapting the obtrusiveness of service interactions in dynamically discovered environments. Proc. of MobiQuitous, Beijing, China:250-262
48. Androjena, Jena Android Porting (2013). Available at: https://code.google.com/p/androjena. Accessed December 2013
49. W3C (2012), OWL 2Web Ontology Language Document Overview, Recommendation 11. Available at: http://www.w3.org/TR/owl2-overview. Accesed: November 2013
50. Bai H, Atiquzzaman M (2003) Error modeling schemes for fading channels in wireless communications: a survey. IEEE Communications Surveys and Tutorials 5(2):2-9
51. de Fez I, Guerri JC (2014) An adaptive mechanism for optimal content download in wireless networks. IEEE Transactions on Multimedia, available online, doi: 10.1109/TMM.2014.2307155
52. Runeson P, Höst M (2009) Guidelines for conducting and reporting case study research in software engineering. Empir Softw Eng 14(2):131-164

53. Lewis JR (1995) Ibm computer usability satisfaction questionnaires: psychometric evaluation and instructions for use. Int J Hum Comput Interact 7(1):57-78
54. Fraile F, de Fez I, Guerri JC (2014) Evaluation of background push content download services to mobile devices over DVB networks. IEEE Transactions on Broadcasting, 60(1):1-15
55. Kellerer H, Pferschy U, Pisinger D (2004) Knapsack problems. Springer
56. Podlipnig S, Böszörmenyi L (2003) A survey of web cache replacement strategies. ACM Computing Surveys 35(4):374-398
57. Xu J, Hu Q, Lee W, Lee DL (2004) Performance evaluation of an optimal cache replacement policy for wireless data dissemination. IEEE Transactions on Knowledge and Data Engineering 16(1):125-139