N° d'ordre :

THÈSE

PRÉSENTÉE A

L'UNIVERSITÉ BORDEAUX 1

ÉCOLE DOCTORALE DE MATHÉMATIQUES ET D'INFORMATIQUE

Par Soraya AIT CHELLOUCHE

POUR OBTENIR LE GRADE DE

DOCTEUR

SPÉCIALITÉ : INFORMATIQUE

Délivrance de services média suivant le contexte au sein d'environnements hétérogènes pour les réseaux média du futur

Context-aware Media Services delivery in heterogeneous environments for Future Media Networks

Directrice de recherche : Mme Francine Krief

Soutenue le : 09/12/2012

Devant la commission d'examen formée de :

Mr Zoubir Mammeri	Professeur, IRIT, Toulouse III	Rapporteur
Mr Michel Banâtre	Directeur de recherche, INRIA, Rennes	Rapporteur
Mr Damien Magoni	Professeur, LaBRI, Bordeaux1	Examinateur
Mr Claude Perron	CTO Thomson Video Networks, Rennes	Examinateur
Mme Francine Krief	Professeur, ENSEIRB	Directrice de Thèse
Mr Daniel Négru	Maître de conférences, ENSEIRB	Co-encadrant de Thèse

Abstract- Users' willingness to consume media services along with the compelling proliferation of mobile devices interconnected via multiple wired and wireless networking technologies place high requirements on the Future Internet. It is a common belief today that Internet should evolve towards providing end users with ubiquitous and high quality media services and this, in a scalable, reliable, efficient and interoperable way.

However, enabling such a seamless media delivery raises a number of challenges. On one hand, services should be more context-aware to enable their delivery to a large and disparate computational context. On another hand, current Internet media delivery infrastructures need to scale in order to meet the continuously growing number of users while keeping quality at a satisfying level.

In this context, we introduce a novel architecture, enabling a novel collaborative framework for sharing and consuming Media Services within Future Internet (FI). The introduced architecture comprises a number of environments and layers aiming to improve today's media delivery networks and systems towards a better user experience. In this thesis, we are particulary interested in enabling context-aware multimedia services provisioning that meets on one hand, the users expectations and needs and on another hand, the exponentially growing users' demand experienced by these services.

Two major and demanding challenges are then faced in this thesis (1) the design of a context-awareness framework that allows adaptive multimedia services provisioning and, (2) the enhancement of the media delivery platform to support large-scale media services. The proposed solutions are built on the newly introduced virtual Home-Box layer in the latter proposed architecture.

First, in order to achieve context-awareness, two types of frameworks are proposed based on the two main models for context representation. The markup schemes-based framework aims to achieve light weight context management to ensure performance in term of responsiveness. The second framework uses ontology and rules to model and manage context. The aim is to allow higher formality and better expressiveness and sharing. However, ontology is known to be complex and thus difficult to scale. The aim of our work is then to prove the feasibility of such a solution in the field of multimedia services provisioning when the context management is distributed among the Home-Box layer.

Concerning the media services delivery enhancement, the idea is to leverage the participating and already deployed Home-Boxes disk storage and uploading capabilities to achieve service performance, scalability and reliability. Towards this, we have addressed two issues that are commonly induced by the content replication: (1) the server selection for which we have proposed a two-level anycast-based request redirection strategy that consists in a preliminary filtering based on the clients' contexts and in a second stage provides accurate network distance information, using not only the end-to-end delay metric but also the servers' load one and, (2) the content placement and replacement in cache for which we have designed an adaptive online popularity-based video caching strategy among the introduced HB overlay.

Keywords- Future Media Internet; Context-awareness; Ontology; Video-on-Demand provisioning; Request redirection, Load Balancing; Content Caching; Home-Boxes Evolution.

Résumé- La généralisation de l'usage de l'Internet, ces dernières années, a été marquée par deux tendances importantes. Nous citerons en premier, l'enthousiasme de plus en plus grand des utilisateurs pour les services médias. Cette tendance est particulièrement accentuée par l'avènement des contenus générés par les utilisateurs qui amènent dans les catalogues des fournisseurs de services un choix illimité de contenus. L'autre tendance est la diversification et l'hétérogénéité en ressources des terminaux et réseaux d'accès. Seule la valeur du service lui-même compte pour les utilisateurs et non le moyen d'y accéder.

Cependant, offrir aux utilisateurs un accès ubiquitaire à de plus en plus de services Internet, impose des exigences très rigoureuses sur l'infrastructure actuelle de l'Internet. En effet, L'évolution de l'Internet devient une évidence et cette évolution est d'autant plus nécessaire dans un contexte de services multimédias qui sont connus pour leur sensibilité au contexte, dans lequel ils sont consommés, et pour générer d'énormes quantités de trafic. L'Internet doit donc évoluer dans ses fonctionnalités, sa taille et sa capacité afin d'offrir aux utilisateurs finaux des services médias personnalisés et de qualité capables de passer à l'échelle.

Dans le cadre de cette thèse, nous nous focalisons sur deux enjeux importants dans l'évolution de l'Internet. A savoir, faciliter le déploiement de services médias personnalisés et adaptatifs et améliorer les plateformes de distribution de ces derniers afin de permettre leur passage à l'échelle tout en gardant la qualité de service à un niveau satisfaisant pour les utilisateurs finaux.

Afin de permettre ceci, nous introduisons en premier, une nouvelle architecture multi environnements et multi couches permettant un environnement collaboratif pour le partage et la consommation des services médias dans un cadre des réseaux média du futur. Puis, nous proposons deux contributions majeures que nous déployons sur la couche virtuelle formés par les Home-Boxes (passerelles résidentielles évoluées) introduite dans l'architecture précédente.

Dans notre première contribution, nous proposons un environnement permettant le déploiement à grande échelle de services sensibles au contexte. Deux approches ont été considérées dans la modélisation et la gestion du contexte. La première approche est basée sur les langages de balisage afin de permettre un traitement du contexte plus léger et par conséquent des temps de réponse très petits. La seconde approche, quant à elle est basée sur les ontologies et les règles afin de permettre plus d'expressivité et un meilleur partage et réutilisation des informations de contexte. Les ontologies étant connues pour leur complexité, le but de cette proposition et de prouver la faisabilité d'une telle approche dans un contexte de services médias par des moyen de distribution de la gestion du contexte.

Concernant notre deuxième contribution, l'idée et de tirer profit des ressources (disque et connectivité) déjà déployées dans les Home-Boxes, afin d'améliorer les plateformes de distribution des services médias et d'améliorer ainsi le passage à l'échelle, la performance et la fiabilité de ces derniers et ce, à moindre coût. Pour cela, nous proposons deux solutions pour deux problèmes communément traités dans la réplication des contenus : (1) la redirection de requêtes pour laquelle nous proposons un algorithme de sélection à deux niveaux de filtrage, un premier filtrage basé sur les règles afin de personnaliser les services en fonction du contexte de leur consommation suivi d'un filtrage basé sur des métriques réseaux (charges des serveurs et délais entre les serveurs et les clients) ; et (2) le placement et la distribution des contenus sur les

caches pour lesquels on a proposé une stratégie de mise en cache online, basée sur la popularité des contenus.

Mots clés- Réseaux médias du futur; Sensibilité au contexte ; Ontologie; Distribution des services de vidéo à la demande; Redirection de requêtes ; Equilibrage de charge ; Caching; Evolution de la passerelle résidentielle.

Table of Content

Liste of figures

List of tables

Remerciements

Mes premiers remerciements s'adressent tout à mon encadrent, Daniel Négru, pour sa grande disponibilité tout au long de ces trois dernières années. Ses conseils et ses remarques, toujours pertinentes, ont été indispensables à l'accomplissement de ce travail.

Mes profonds remerciements vont également à ma directrice de thèse, Francine Krief, pour ses conseils, son temps, mais par-dessus tout, pour sa bienveillance.

Je voudrai également remercier tous les membres du jury de m'avoir fait l'honneur et le plaisir de prendre de leur temps pour juger mes travaux.

Je tiens aussi à remercier, tous mes amis et collègues du LaBRI pour la bonne ambiance qui y régnait. Je citerai notamment ceux avec qui j'ai partagé mes bureaux successifs, Julien, Willy, Yiping, Simon, Preston, Cendrine, Florent et Pierre, mais je pense que les autres se reconnaitrons également dans mes propos. Travailler en leur compagnie à été pour moi un vrai plaisir.

Un grand Merci également à tous mes autres amis hors du LaBRI, dont je ne citerai les noms de peur d'en oublier quelqu'un. Leur présence dans ma vie, ces dernières années, m'a été d'un grand réconfort, particulièrement dans un contexte où ma famille était loin.

Je ne pourrai finir ces remerciements sans exprimer ma profonde reconnaissance à toute ma famille au sens le plus large, et particulièrement à mes parents, ma sœur, mes frères et mon tonton préféré qui se reconnaitra. Leur amour, leur soutien indéfectibles et la confiance qu'ils ont toujours placée en moi n'ont cessé de me porter. Merci de m'avoir toujours accompagné dans tous mes choix.

.

Chapter 1

Introduction

Internet is becoming part of our daily life. Billions of users, all over the world, are using it for information search, retrieve and exchange, business, administration, education and social purposes, to name a few. One major trend in the Internet usage is users' willingness for multimedia services (e.g. Video-on-Demand, IPTV, IP telephony, etc.). Indeed, these services have experienced a major breakthrough in the recent years. According to Cisco VNI [1], video traffic alone will exceed 91 percent of global consumer traffic by 2014. This trend is particularly accentuated by the advent of User Generated Contents (UGCs) that have brought an unlimited choice of videos to the Service Providers libraries by enabling users to be not only passive consumers but also producers. Typical examples of such popular services that exist thanks to a significant users' participation are YouTube, Facebook, Flickr, etc. In addition, as the broadband connectivity becomes more and more pervasive, consumers' expectations for higher services' quality and better experience increase. According to [1], it is expected that, by 2014, 46 percent of consumer Internet video traffic will be composed of HD/3D Internet video.

Another important trend in Internet usage is the high users' expectation for seamless access to services. Indeed, with the compelling proliferation of mobile devices (e.g. smartphones, tablets, laptops, etc.) and the rapid growth in mobile networking technologies (mobile network connection speeds doubled in 2010: globally, the average mobile network downstream speed in 2010 was 215 kilobits per second (kbps), up from 101 kbps in 2009 [2]), users are willing to access high quality services anywhere, anytime and through any device or network. At the end-user point of view, only the service value counts, not the networked device or software components that implement it.

If these new trends in Internet usage seem to be very attractive; they also impose stringent requirements on the current Internet infrastructure in terms of scalability, reliability and performance. It is especially true, while dealing with video services which generate huge traffic that heavily loads on the Internet. Moreover, they are known to be time-critical and loss-sensitive. These services increase the pressure on network infrastructure typically for high capacity, low latency and low-loss in the communication paths.

In such a computing context, it is then obvious that Internet has to growth in its functionality, capability and size to enable the creation and efficient distribution of novel advanced rich Media Services. Indeed, many limitations have been identified in the current

Internet [3] such as facilities for large-scale service provisioning, management and deployment, facilities for network, device and service mobility, Facilities to seamlessly use context information to enhance and improve existing services and deploy new ones, Facilities to support Quality of Service (QoS) and Service Level Agreements (SLA), Trust Management and Security, Privacy and data-protection mechanisms of distributed data and so on.

In this context, this thesis introduces a novel architecture, enabling a collaborative framework for sharing and consuming Media Services within Future Internet (FI). Our architecture proposal aims to constitute an integrated environment where End-Users and Service Providers share and distribute both legacy and novel multimedia-based services. The proposed architecture relies on two main pillars: (1) an innovative Service Environment (SE), involving a multifaceted service management, an overlay of interconnected media-centric Home Gateways ("Home-Boxes") and (2) a radically enhanced Network Environment (NE), featuring inherent Content Awareness enabling the creation of virtual overlay networks tailored for media transport. On top of the SE, a flexible User Environment (UE) is foreseen to enable ubiquitous service access in various usage scenarios using different wired and wireless terminals and featuring a unified graphical interface, a context modeling and management solution and real-time Quality-of-Experience (QoE) monitoring, all in strong cooperation with the new Home-Box entity.

In this thesis, the focus is put on the User and Service Environments and more specifically on the virtual Home-Box (HB) layer, shared between the two environments. We particularly consider two important challenges: (1) providing large-scale context-aware framework that enables personalized and context-adaptive multimedia services and (2) enhancing the current multimedia distribution platform to achieve scalability and performance.

Indeed, context-awareness is a key technology enabling ubiquitous access to services. This feature is all the more important in the case of multimedia services that, as mentioned previously, are very context-sensitive. The promise of context-awareness is to provide computing frameworks that track context information from different sources in the network, model it in a way that enable their processing by software entities, understand enough on it and finally draw conclusions on it. The latter will trigger adaption decision on the applications behavior. The aim is to provide end-users with contents, resources and services that suit their needs and preferences without explicit intervention from them. Context could be a user location, preferences and activity, device capabilities, network conditions, environment information such as time, light intensity, motion, sound noise level, etc.

Although different context-aware systems have been proposed, the design and deployment of such systems that scale to the size of Internet and support different multimedia services is still a challenging issue. In most current systems, the set of commonly used context information is still limited to identity and location. In addition, these systems are usually tackling specific applications and domains making their extension difficult and almost impossible. An efficient context-aware framework requires two main careful designs. First, the context has to be represented in a formal model that will allow it to be on one hand processed from software entities and on the other hand to be flexible, extensibility and interoperable. Second the context management (context acquisition, processing and dissemination) should also be designed with keeping in mind scalability, extensibility and framework responsiveness issues.

In this context, we propose a context-aware framework that will enable the context-awareness feature for adaptive multimedia services. To achieve scalability the latter feature is distributed among the virtual HB layer. We have then proposed two approaches to model and manage context information based on the tow mostly used context modeling approaches. In the first contribution, the context model is based on XML and its dialects and the context management relies on XML related tools. Based then on markup scheme models, this contribution aims to provide a context-aware multimedia framework with high performance. In another contribution, the context is modeled using ontology and rules for more expressiveness and formality. The aim was to prove the feasibility of such modeling approach in the field of large-scale multimedia services. Since the context model support reasoning, this function is incorporated in the context management to ensure context consistency checking, high-level context inference and context-aware decisions triggering.

Concerning the delivery of multimedia services, two candidates and competitive used approaches are Content Delivery Networks (CDNs) and P2P systems. In CDNs, the contents are pushed from origin servers to multiple powerful servers – so-called surrogates – with high storage and upload connectivity, deployed at strategic locations of the Internet edges, thus allowing Service Provider to handle much more End-Users requests and avoid typical congestion caused by flash crowds (i.e. a larger than anticipated set of users attempting to access a just published content). However, if CDNs enhance services scalability and network latencies, they also induce heavy scaling cost, especially with the continuously growing multimedia services popularity. More over if this approach avoids congestion on the core network, it still have no control on the last mile network (aggregation and access network).

On the other hand, P2P systems can be considered as the logical extreme of the distributed approach for content delivery and, accordingly, are highly scalable. However, if the P2P systems represent a promising low cost approach for highly scalable video content distribution, they also present some weaknesses such as lack of control, high peer churn and unfairness and significant imbalance between the uplink and downlink capacity. These weaknesses may rapidly result on system saturation and poor quality.

A hybrid approach called also peering CDN or adaptive CDN in some literatures consists in combining the two approaches to benefit on one hand on the reliability of CDN and on another hand on self-scalability of P2P networks. This solution brings the content closer to end users than CDNs could do and this with a much lower cost. However the efficiency of this type of solution is highly dependent on how the contents are cached among peers and from where users access the contents they request. Efficient and well designed caching and server selection strategies are then required.

Our proposal for multimedia services distribution is in line with the last approach. The idea is to leverage the participating and already deployed Home-Boxes disk caching and uploading capabilities to achieve service performance, scalability and reliability, especially in current context where the broadband providers are heavily investing to build out their high speed last mile networks. The proposed architecture keeps the high control on traffic design and management of CDNs in the core network while taking advantage of the User Environment capabilities (e.g. in terms of bandwidth, processing, availability, and storage), as P2P solutions do. This solution brings the contents closer to End-Users decreasing thus path latency towards a better experience. An efficient online popularity-based content placement and replacement in the HBs' caches is also proposed. The caching strategy aims to efficiently spread the most popular contents among the HBs' caches in order to make them available to much more users while keeping lower cost distribution. The caching is combined with an anycast-based request redirection strategy that aims to always provide users with contents from the optimal locations. The request redirection strategy consists on a two-level filtering: (1) the policy-based filtering that aims to select the contents that fit best the user context and (2) the metric-based filtering based on the server load and network delay that aims to select the most closer non overloaded servers. Since we target the VoD service that is loss-sensitive, our goal with this two-level filtering request redirection is to avoid congestion and this in all the environments. At the User Environment, by considering his computing context, at the Service Environment, by considering the servers' load and finally in the Network Environment, by always selecting the paths with the lowest delays.

To efficiently introduce our contributions and the context in which they were proposed, the reminder of this thesis is organized as follows:

Chapter 2 *"State of the art"* presents the state of the art of the two Future Media Internet challenges targeted by this thesis. We start then by presenting the state of the art of context-awareness. In this part, we give first an overview of the initiatives that tried to define the notions of context and context-awareness, and then present the different context modeling approaches by illustrating each of them by many examples and highlighting for each of them its advantages and limits and finally present the composition of a context-aware framework and describe and compare some representative context-aware systems that have been proposed in telecommunication field, and Internet field and academic research. The second past is dedicated to the state of the art of multimedia distribution networks. We start by an overview of managed networks including the proprietary ones and the standardization effort made in this field. Then, we present the best effort networks, in which we are particularly interested, namely the Content Delivery Networks (CDN) and the Peer-to-Peer (P2P) networks. For each approach we give the main technologies, protocols and strategies used in the content delivery process.

Chapter 3 *"Architecture Proposal for Future Media Networks"* presents our proposal for context-aware future media Internet architecture and the different environments and layers on which it is composed. The proposed architecture is compared to the FMIA-TT reference model [4] as well as to the most representative Future Internet architecture and particularly to Future Media Internet architectures proposed in literature.

Chapter 4 *"Framework enabling Context-Awareness in Future Media Networks"* presents our contribution to context awareness. In this chapter we start by presenting how context-awareness can be achieved in the Future Media Internet architecture presented in chapter 3. Then we detail the two contributions that we have proposed to achieve context-awareness: the markup-scheme based context-aware framework and the ontology- and rule-based framework. For each of them we detail the context model as well as the context management functionality and we highlight its strong points and limits.

Chapter 5 ***"A VoD Content delivery solution over IP-based media distribution systems***" presents our contributions to video content delivery for Video-on-Demand (VoD) services. This chapter presents first our contribution on request redirection strategy in case of replication among surrogate servers, then the application of this strategy to the proposed architecture in chapter 3. In this contribution, the request redirection is combined with a replication among the HB virtual layer. The used offline replication strategy based on the video content popularity is then detailed. Finally, we present our proposal for online caching, the online adaptive popularity-based caching strategy that is used to dynamically place video contents on HBs' caches. The video spread among the HBs' caches and the replacement strategy in cache as well as the derivation of the system cost are detailed. For each contribution we present its evaluation. All the contributions are evaluated by simulation and we describe for each of them the environment in which they were simulated and comment the simulation results.

Finally the ***conclusion*** summarizes the proposed solutions in this thesis and brings out some ongoing and future works.

Chapter 2

State of the art

2.1. Introduction

In the current computing context in which both user enthusiasm for multimedia services and their expectations for better quality in a disparate context are increasing more and more, enabling seamless, personalized and efficient Internet media services delivery in a scalable and low cost manner is a challenging issue. Therefore, the design of context-aware and scalable Internet media delivery frameworks that provide ubiquitous and high quality multimedia services to a large and continuously growing end user population has received a wide concern from academic and industrial research.

In the following, we first present the related work to context-awareness through the last two decades. The state of the art of both context modeling and context management are considered. Then, we introduce most relevant Internet multimedia content delivery solutions that are classified in two major categories: the managed delivery solution and the non-managed best effort one. For each solution, we explain the different techniques used to achieve scalability, reliability and performance.

The presented state of the art, in these two domains, constitutes the basis for our proposal for a context-aware framework for media delivery solution (including context-awareness features) for Future Media Internet Architecture. While presenting our architecture and solutions, we will also go further in the more-specific related works for each part in order to clearly justify, compare and evaluate our approaches towards other existing proposed solutions.

2.2. State of the art: Context-aware systems

2.2.1. Context & Context-awareness

2.2.1.1. Context definition

The context is a generic term that, until now, has not been given a clear and standard definition. However, in the context of ubiquitous computing, several definitions have been provided in the literature.

A widely used definition is: “a list of information able to be included”. In 1994, Shils et al. [5]-[6] already associated the context with the concepts of location, identity of people and objects surrounding the user and changes that may impact him. The authors then identify three categories of context: computing context (resources used in physical access to the service), the user context (location, profile, etc.) and the physical context (temperature, light, noise, etc.). Chen and Kotz extend this classification by including the “time” feature (hour, date, season, etc.), allowing to capture the temporal aspect of the first three categories and get a historical past that can be very useful for applications. Also, in a similar approach, Brown et al. [7] identifies context information such as location, orientation, time, season, temperature, etc. In [8], Held et al. identifies, as relevant, context information associated with the user terminals (e.g. memory, CPU power, peripheral I/O, etc.), to its network connectivity (e.g. bandwidth, delay, error rate, etc.) and to user himself (e.g. profiles and preferences). Ryan et al., in turn, sets the context such as user location, identity, environment and time [9]. Dey provides the context as the user emotional state, its interests, location and orientation, date and time and the surrounding objects and people [10].

Another approach consists in defining the context by using synonyms such as the environment or the situation. Hull and al. describes the context as aspects of the current situation. Brown, in [7], defines the context as all the components of the user's environment that are known by the user terminal. Ward et al. sees the context as the environment state of an application [11].

In fact, these definitions are very difficult to apply. Based on the first approach, the definitions are very specific. It is therefore difficult to say, using a definition, if new information should be considered as part of the context or not. As for the second approach, it is too broad and vague. By defining the context, for example by environment, some works identify it with the user environment, whereas others assimilate it more to an application environment.

Other works have attempted to define the context in a more formal way. Schmidt et al. defines it as knowledge that could be on the user, its terminal status and its surroundings, its situation and like the minimal extension, its location [12]. Abowd and Mynatt define, as necessary, the context information to meet the five W's, Who, What, Where, When and Why [13]. Chen and Kotz, for their part, define the context as the set of states and environmental configurations that determine the behavior of an application or where interesting events to the user occur [14]. Another generic definition, widely adopted by the research community, is that of Dey [15]-[16] which defines the context as any information that could be used to characterize the situation of an entity. An entity can be a person, place or object considered relevant for the interaction between a user and an application, including the user and applications themselves.

“Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves”. [15]-[17]

In [15], Dey derives the notion of situation and defines it as the description of states of relevant entities. Indeed, applications are often interested in the aggregation states of relevant entities for their implementation. The notion of situation has been taken in most systems sensitive to context.

We relied on this last definition and tried to identify the relevant entities for the various actors (users and services providers) of a large scale multimedia system. Our understanding of the context and the information that we consider relevant and useful in multimedia domain are presented in Chapter 4.

2.2.1.2. Context categories

As context covers a wide range of heterogeneous information that continues to grow with the advent of new services and technologies, it is useful to classify this information. A very popular approach in the classification of context information is the one that distinguishes the different dimensions. In [18], both internal and external dimensions are identified.

Other studies have also targeted physical and logical context, measured and conceptual context [19] or physical and social context [20]. The physical (external) dimension refers to the context information measured by sensors such as location, brightness, temperature, etc. The logical dimension (internal), on the other hand, refers to information derived from monitoring of user interactions such as the user goal, its activity, its emotional states, etc.

Chen and Kotz, in [14], classify context information into two categories. The active context that influences the behaviour of an application and the passive context that is relevant, but not necessarily critical, to the performance/efficiency of an application. So, the classification of context information depends on the use case. Context information can be considered as active for a given application and passive for another.

Another classification of context, widely used in context-aware-systems, consists in distinguishing the dynamic context information that changes over time, from static context information. This does not necessarily mean that the latter ones never change but they do not change during the service consumption. This classification is mainly used in multimedia systems where time is critical.

According to [15], addressing the context, three main entities can be distinguished:

- Places (region or geographical location e. g. office, building, street, etc.);
- Persons (individuals or groups of individuals being in the same location or different locations);
- Objects (physical object or software component e.g. an application, a document, etc.).

Each of these entities is described by a set of attributes that, in turn, can be classified into four different categories:

- Identity: each entity has an unique identifier in the space of names used by the application;
- Location: in addition to the position in a plane, the location includes information such as orientation, or altitude as well as any information that can be used to infer spatial relationships between entities as co-location, the capacity or proximity;
- Status: it is any intrinsic characteristic of an entity that can be captured. For the entity "places", these features represent the temperature or noise that can be measured. For the entity "person", they correspond to its state or its business;

- Time: it is used to timestamp situations. This context information provides track-records that are often very useful to applications. The information “time” is often used with other context information such as stamp or to set the time interval in which they remain valid.

2.2.1.3. Definition of context-awareness

Once the context information is identified, an important task is to define how information will be useful for applications. The notion of Context-Awareness, has also been given a number of definitions in the literature. In [5], Schils and Theimer define the notion of context-awareness as the ability for an application to discover the changes that may occur in the environment in which it runs and to react accordingly. From the perspective of adaptation to the context, the above definition is included in [6], defining context-aware applications as the examination of the environment and adaptation according to the changes that may occur, such as the location of use or nearby people or resources. The authors identify four categories of applications as sensitive to the context:

- Selection depending on the proximity: it is a user interface technique, exploiting the location information to highlight the closest resources thereby facilitating their selection;
- Automatic reconfiguration based on the context: it is the process of adding a new component, removing existing components or modification of relations between different components;
- Information and contextual commands: they consist in producing different results depending on the context in which they are issued;
- Actions triggered at the context change: they are simple IF THEN rules, which role is to guide the adaptation process.

On the same approach, Dey and Abowd propose, in [16], the classification, into three categories, of the functionalities of a context-aware application. These are:

- The presentation of information: this category refers to applications that either present the context information to users or uses the context to offer to users actions according to it;
- The automatic execution of services: it describes applications that trigger commands or reconfigure the system, transparently to the user, depending on the context information;
- The storage and enrichment of context information: it refers to applications that enrich the context information with metadata and which perpetuate it for future use.

According to Razzaque et al., the context-awareness is a term that comes from computing to designate terminals having the knowledge of the circumstances in which they are used and, consequently, reacting accordingly. The authors add that the sensitivity to context implies the development of applications able to acquire the context and having a dynamic behaviour depending on it.

In [15], Dey summarizes the above definitions in a simple and generic definition, adopted by the research community, which states that a system is context-aware if it uses context to produce information and / or services relevant to the user. The relevance depends on the tasks requested by the user.

2.2.2. Context models

Context modeling consists in a fundamental field in context-awareness concept. Indeed, to be processed by computational entities, context information should be formally described in a contextual model. Strang and Linnhoff-Popien classified context models based on the data structure used to represent and exchange context between the system entities. They identified six models: key-value models, Markup scheme models, graphical models, object-oriented models, logic-based models and ontology-based models. These models are detailed and illustrated through examples in the following.

2.2.2.1. Key-value models

The key-value models are undoubtedly the models with the simplest data structure. The context information is represented as key-value pairs (name of the context information and its actual value). The simplicity of the key-value models led to its adoption in several systems. Schilit et al. [21] were already exchanging the context information formatted as environment variables in their Active Badge system. The key-value models are also used in the context toolkit [15], where the context information is managed by the context widgets. Each widget presents a state composed of a set of parameters characterizing the context information, and the associated behavior that can notify the applications of the parameter variation. For example, the state of IdentityPresence widget models the presence information using three key value pairs: the managed location, the identifier of the last detected user and the time when the detection occurred. As to the associated behavior, the widget will notify the application of the arrival or departure of a user through the three pairs.

This model is also used in the IETF Media Feature Set [22] standard, which aims at describing the terminal characteristics and the user preferences. Boolean expressions formed by key-value pairs are used to describe this information.

The simplicity of the key value data structure facilitates the management of context information. Unfortunately, its lack of expressiveness prohibits any deduction from the considered context information and its flat structure does not support the relationship definition among parameters. The absence of data schema and meta-information on the considered context makes this type of models very difficult to reuse. Indeed, the key-value models are usually tightly coupled with the systems for which they have been built for.

2.2.2.2. Markup scheme-based models

These models are characterized by a hierarchical data structure. The context information is organized into elements identified by their tags, which are associated with attributes and contents. Recursively, an element can itself contain other elements. These models are often used in the standard for user profiling. Below are some examples of user profiles defined in different networking areas:

A. 3GPP Generic User Profile (GUP)

The GUP [23-25] has been defined by the 3GPP for harmonizing the usage of user-related information coming from different entities. The standard does not impose any classification of the information to be included in the profile but recommendations consist in the following:

- Information on the user subscription such as service offering and accessibility ;
- General information on the user himself, including his name, address, preferences and currently active profile;
- Information related to PLMN such as the GPRS parameters and the favorite access technology;
- Security policies for some services, for example the localization service;
- User information specific to a given service, such as relevant information for service personalization, and service access (key, certificate, password, etc.) ;
- Information related to the terminal device such as its hardware characteristics, interfaces and supported services;
- Other information, for example accounting.

A profile consisting of several components is defined for each user. The components can be managed and stored in different network nodes to which the user is subscribed, as well as in external service providers. All profiles must have the same structure, defined in a W3C XML Schema [26-28]. The component can be defined in external schemas, referenced by their namespaces in the element <xsd: schema>, and then imported into the global schema using element <xsd: import>.

B. MPEG standards

In order to ensure multimedia interoperability, two standards are defined by the Moving Picture Experts Group (MPEG), namely the MPEG-7 [29] also known as Multimedia Content Description Interface for MPEG and the MPEG-21 [30]. The structure of these standards is based on the W3C XML Schema.

MPEG-7 provides tools for describing the multimedia resources. The MPEG-7 description includes various information on the multimedia content such as its classification, creation (title, creators, etc..), usage (history of use, copyright, etc..), storage (format, encoding, etc..), as well as structural aspects (spatial components, temporal or spatio-temporal content), conceptual aspect (objects, events, etc..) or some low-level characteristics (colors, textures, etc.). Thanks to these descriptions, MPEG-7 allows the indexation of these multimedia resources strongly based on the content, and, consequently, more effective search and discovery based on criteria such as the content type and the involved person/object.

From a structural point of view, a MPEG-7 description is composed of basic elements called descriptors (D) used to describe the characteristics of multimedia content. The relationships between these descriptors are then described by Description Scheme (DS). The syntax of description tools is defined by the DDL (Description Definition Language), which is an extension of the W3C XML Schema. In order to support efficient storage, exchange and processing of MPEG-7 descriptions, the standard also offers several tools for encoding these descriptions in binary form or for synchronizing the descriptions with the content they describe, etc.

MPEG-21 aims to allow users to access, consume, share or more generally to manipulate multimedia content in an efficient, transparent and interoperable way. For this, the standard defines an open architecture that covers the entire distribution and consumption chain of multimedia contents. MPEG-21 relies on two innovative concepts: the Digital Item (DI) which is the basic unit involved in a transaction (audio, video, image, etc.) and the User that interacts (publishes, issues, etc.) with the MPEG-21 architecture. The MPEG-21 standard covers several independent aspects such as the identification, declaration, adaptation and streaming of a Digital Item, the protection of intellectual properties, session mobility, etc.

The Part 7 of the standard, called Digital Item Adaptation (DIA), provides the necessary tools to adapt the DI and enable their universal access. These tools are classified into eight categories as shown in Figure 2.1, namely UED (Usage Environment Description), BSD (Bitstream Syntax Description), BSD Link, Terminal and Network Quality of Service, UCD (Universal Constraints Description), Metadata Adaptability, Session Mobility and DIA (DI Adaptation) configuration.

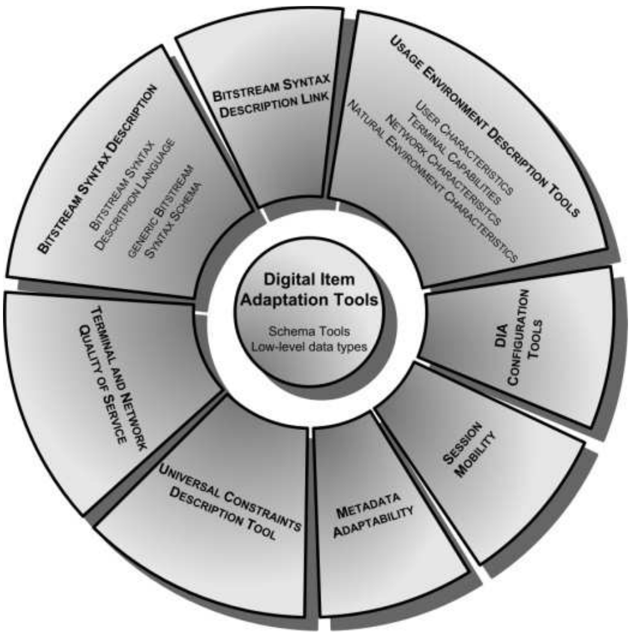

Figure 2.1. General view on the functional classification of MPEG 21 DIA tools.

The syntax of these description tools is defined as a collection of schemas. These schemas are then regrouped into a single document that defines the adopted namespaces and their respective prefixes. Two base types are defined by the standard:

- *DiaBaseType*: provides an abstraction of the types hierarchy. The majority of tools are defined as an extension of this type;

- *DIADescriptionType*: extends the DiaBaseType and abstracts a subset of types that can be used for describing a component such as the characteristics of a terminal. The types that extend DIADescriptionType are called high-level types.

C. Composite Capabilities/Preference Profiles (CC/PP)

The Composite Capabilities / Preference Profiles (CC/PP) [31] is a standard proposed by the W3C for the representation of user profiles. The CC/PP profiles allow the description of user preferences and terminals, especially mobile devices that are used to access the service. For this, the standard defines necessary tools for modeling the context of service consumption in order to enable the service adaptation.

The CC/PP profiles are described by RDF triples [32], serialized as XML documents for communication purposes. A profile is then represented as a two levels-tree structure in: one or more components associated with one or more attributes, with the possibility for each component to reference by default a set of external attributes. These components are represented by resources *ccpp:Component* and are connected to the root node of the profile using the property *ccpp:component*. At the second level, each component is then described by a labeled sub-tree. The labels represent the attributes describing the characteristics of the component and the leaves represent the values of these attributes. The default attributes (usually defined in external RDF documents and identified by their URIs) are referenced using the property *ccpp:default*.

For the extensibility purpose, the components and attributes contained in the CC/PP profiles are not defined by the standard. The vocabularies that are specific to different areas are defined in separate standards based on the RDF Schema [33]. An example of the vocabulary for describing the WAP terminal profiles is the WAG UAProf [34] proposed by the WAP Forum.

As shown in several studies [35], [36], the CC/PP profiles have several limitations due to their structure defined in the standard, and due to some missing features in RDF such as the cardinalities. Different profiles are constructed based on CC/PP in order to overcome these limitations and to complete the vocabulary defined by CC/PP and UAProf, for example, Comprehensive Context Profiles (CSCP) [8] or CC/PP Context Extension [35].

Markup scheme-based model are the mostly adopted models to represent context. There exist many standards and tools (parsers, validation tools, etc.) that are based on it especially with the generalization of the use of XML within web services tools and standards. In addition, the herarchical structure on which this model is built fits well the decomposition nature of content information.

However, existing standards are always targeting a specific type of applications and context domain. For example, MPEG is specific to User and Content, CC/PP is user and device orientied, etc. None has reached a full generic solution capable of meeting all requirements for a generic User Profile. Another limit of these models is that they only model the syntax of context information and lack for semantic. There is no way to represent meta-information or to model the relationship that may exist between the context information. Consequently, there are no possibilities to reason on context data.

2.2.2.3. Graphical models

A generic and well known modeling instrument that is also appropriate to model context is the Unified Modeling Language (UML). The context entities and their processing are represented in the UML diagrams (class diagram, use case diagram, sequence diagram, etc.). An example of UML-based graphical model is the air traffic control presented in [37].

Another graphical model for modeling context is the one introduced in [38], in which Henricksen et al. extend the Object Role Modeling (ORM) [39] to allow context facts types to be categorized according to their persistence and source. In ORM, the basic modeling concept is the *fact*, and the modeling of a domain using ORM involves identifying appropriate fact types and the roles that entity types play in these roles. Facts are classified as either static or dynamic. The latter ones are in their turn classified as profiled, sensed or derived. As illustrated in Figure 2.2, the facts, roles and constraints annotations of ORM are extended to capture:

- Different classes of context (static facts or dynamic facts that in turn classified as profiled, sensed or derived);

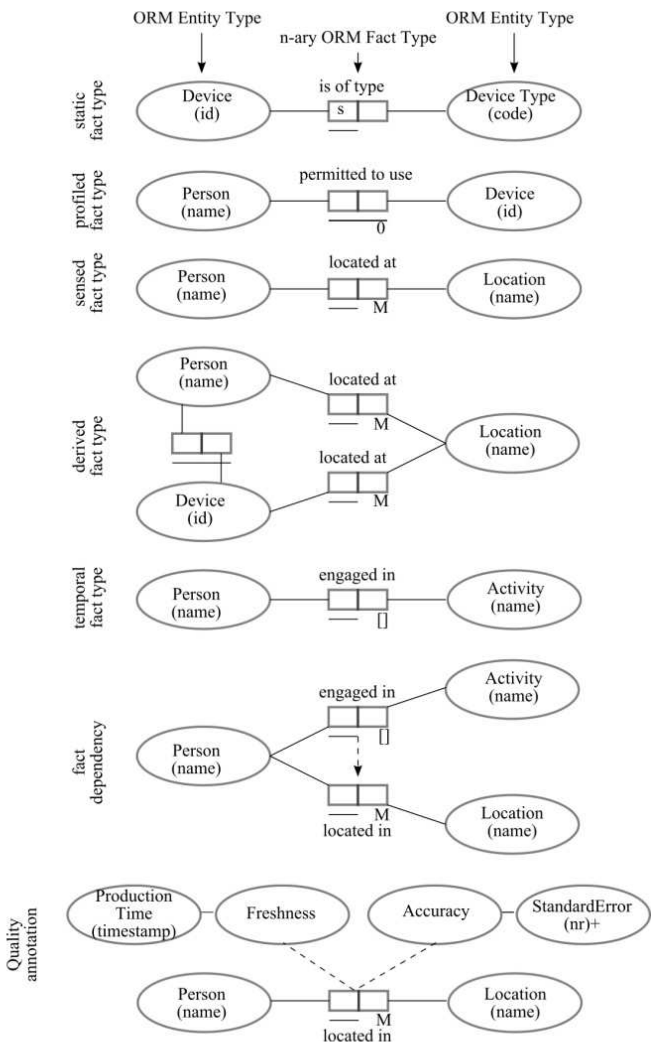

Figure 2.2. *Contextual Extension ORM.*

- Histories: by representing the start and end time as roles that participate in all uniqueness constraints of the fact type;
- Dependencies: by representing the relationship between facts by the binary, transitive dependOn relation;
- And quality: by associating facts with quality indicators such as accuracy and certainty.

The strengths of graphical models are their efficiency in representing the structure of context information. These models are intuitive and easy to integrate to the UML model of the rest of the system. Some code can also be derived from the context diagrams. However, since the graphical models are commonly used for human structuring purpose, they present a low level of formalism.

2.2.2.4. Object-oriented Models

This modeling approach encapsulates the representation and process details of context entities (such as location, identity, etc.) in context objects. The latter are accessed via well-known interfaces. The advantage of using such an approach in context modeling is to benefit from the full power of the object oriented approach (e.g. encapsulation, inheritance, reusability).

An illustration of object-oriented models can be seen in the Information model standardized by 3GPP to support User Data Convergence (UDC) [40]. This information model denotes an abstract formal representation of entity type, including their properties and relationships, the operations that can be performed on them, the related rules and constraints.

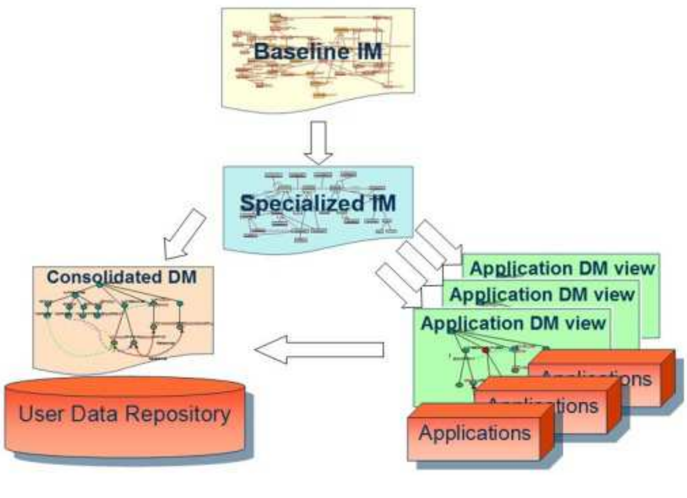

Figure 2.3. ***Information and Data Model of UDC.***

As illustrated in Figure 2.3, The UDC information model infrastructure is based on the Common Baseline Information Model (CBIM) [41]-[42]. CBIM describes the basic Information Object Classes (IOC) of UDC which constitutes the baseline for any given application. CBIM provides support for Subscription, Service Profile, End User, Identifier, End User Group and End Device. The UML representation of the CBIM Core View and Identifiers are shown in Figure 2.4 and Figure 2.5.

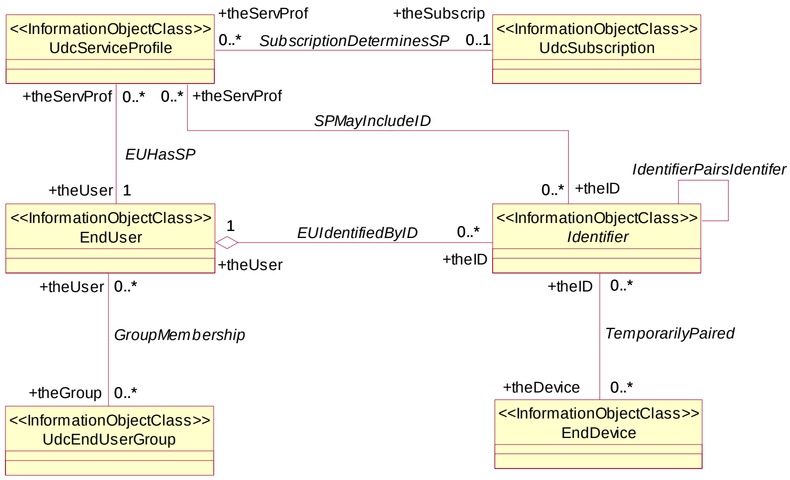

Figure 2.4. *UDC CBIM Core View.*

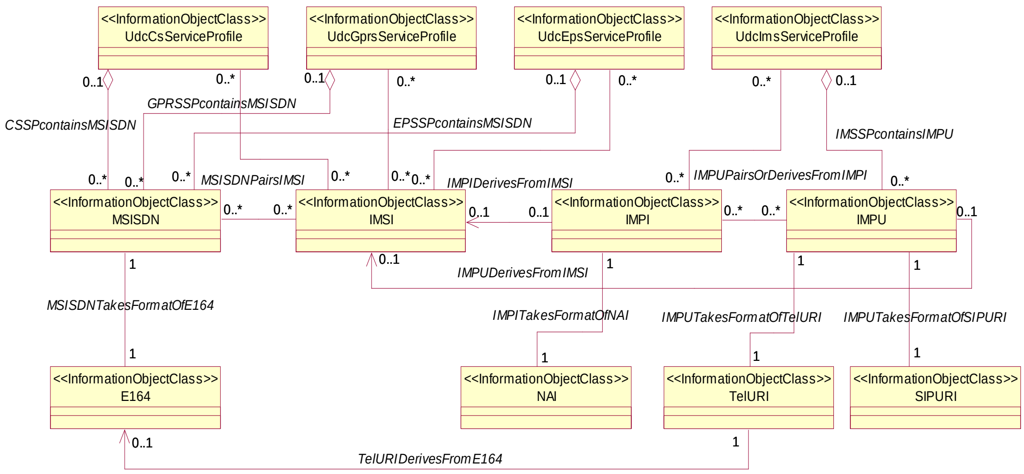

Figure 2.5. *UDC CBIM - Identifiers.*

CBIM can be further extended by means of specialization according to the user and service structure of a given operator. The Specialized Information Model (SpIM) is operator specific and contains all information to be stored in the User Data Repository (UDR). Each application only interfaces with the UDC for the data that it deals with. This leads to the concept of Application Data Model View (implementation of a SpIM for a given application). Since a single SpIM can serve different application, several Application DM Views are available. The implementation of the whole SpIM in the UDR leads to a Consolidated DM. Since the DM is a particular implementation of IM, it shall be possible to derive one or more DMs from a Common IM.

Another example of the use of object-oriented models is the *Hydrogen* project [43], which objective is to allow context sharing in a peer-to-peer manner between devices located in the same space, via WLAN, Bluetooth, etc. The *Hydrogen* project distinguishes between the local (knowledge that our own device knows about) and the remote Context (knowledge that another device knows about). Both local and remote context are modeled as context objects related to the superclass *ContextObject*. Extensibility is ensured by means of specialization. The superclass *ContextObject* is then extended by different context types such as *LocationContext* and *DeviceContext*. Each context type object has to implement the methods *toXML()* and *fromXML()* from the *ContextObject* class in order to enable context sharing through XML streams.

2.2.2.5. Logic-based models

In logic-based models, context is defined using facts, expressions and rules. The high level formality of this type of model allows high-level reasoning or inference. Context information can then be added to, modified and deleted from the logic-based system in terms of facts or derived from the system rules.

One of the early works that model context based on logic is the one published as Formalizing Context by McCarty [44]-[45]. The context is introduced as abstract mathematical entities with properties useful in artificial intelligence. The aim of this formalization is to allow simple axioms for common sense phenomena, e.g. axioms for static blocks world situations, to be lifted to context involving fewer assumptions, e.g. to contexts in which situations change. Various ways of getting new contexts from old ones by specialization are mentioned (specializing the time or place, specializing the situation, making abbreviations, specializing the subject matter, making assumptions and by specializing the context of a conversation). The basic formula in this approach is *ist(c,p)* which is to be taken as assertion that the proposition *p* is true in the context *c*.

2.2.2.6. Ontology-based models

The Ontology-based models represent the context based on ontologies [46]. Originally defined in philosophy as the study of the nature of *being*, *existence* or *reality*, as well as the basic categories of being and their relations. Ontologies are used in Computer Science for formal system representation using concepts, attributes and relations. According to [47], ontologies are mainly applied in the following three domains:

- Knowledge sharing and exchange: ontologies provide a common vocabulary which can be used by different entities (human and software-based entities);
- Logic-based reasoning and deduction: ontologies can be used to deduce implicit knowledge based on logic rules;
- Knowledge re-use of: general usage of ontologies, such as ontologies describing temporal or spatial concepts, can be further re-used when defining an ontology for a specific domain.

By providing an explicit conceptualisation giving a description of data structure and semantics, ontologies are perceived as promising tools towards adequate description and representation of context data. Their relation with the semantic Web also constitutes an important factor in the fact that different languages have been defined for ontologies, e.g. Ontolingua [48], LOOM and Ontology Web Language (OWL) [49]. Three examples of OWL-based ontologies are presented below: CONON [50], COBRA-ONT [51] and SOUPA [52].

A. CONtext ONtology (CONON)

CONON [50] is an ontology designed for the representation of context in pervasive environments. Given that services provided in such environments are generally gathered within a collection of sub-domains set for different intelligent spaces (i.e. home, vehicule, etc.), the underlying aim of this ontology is to first represent generic context information, suitable to all these sub-domains but also to provide enough extensibility so as to represent context information specific to different suitable domains. Towards this, CONON has a two-level construction: one high-level ontology modelling basic context information and related general properties, and one collection of ontologies describing concepts in details, linked to specific domains, and their related properties.

Figure 2.6 introduces the high-level ontology, in which the authors choose to model the context around four abstract entities: *CompEtity* (including all software or hardware entities allowing access to the service), Person, *Activity* and *Location*.

Figure 2.7 brings out a partial representation of an ontology specific to smart homes. Here, the number of abstract classes of the high-level ontology increases with new classes specific to the modelled domain. For example, in Figure 2.7, the *IndoorSpace* class is extended with four new sub-classes: *Building*, *Room*, *Corridor* and *Entry*.

Based on OWL, CONON allows to reason and so, to either infer new implicit information from explicit ones, or detect context inconsistences due to capture errors. The inference is made possible thanks to descriptive logic rules integrated in OWL semantics. CONON also allows more flexible reasoning process, through the End-Users-defined rules based on first order logic in order to deduce context situations as shown in the example below, wherein the activity of the End-User is deduced from a certain number of facts included in the ontology.

$$(?u\ locatedIn\ LivingRoom)\ \hat{}(TVSet\ locatedIn\ LivingRoom)\hat{}\ (TVSet\ status\ ON) \\ => \ (?u\ situation\ WACHINGTV)$$

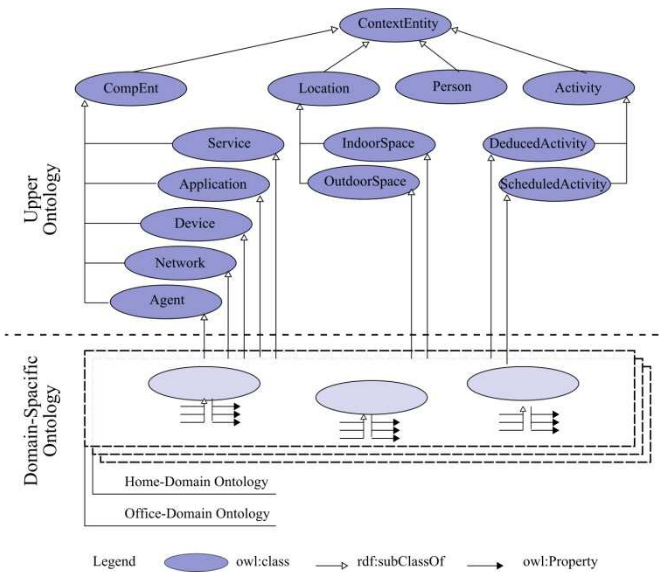

Figure 2.6. *CONON : high-level ontology*.

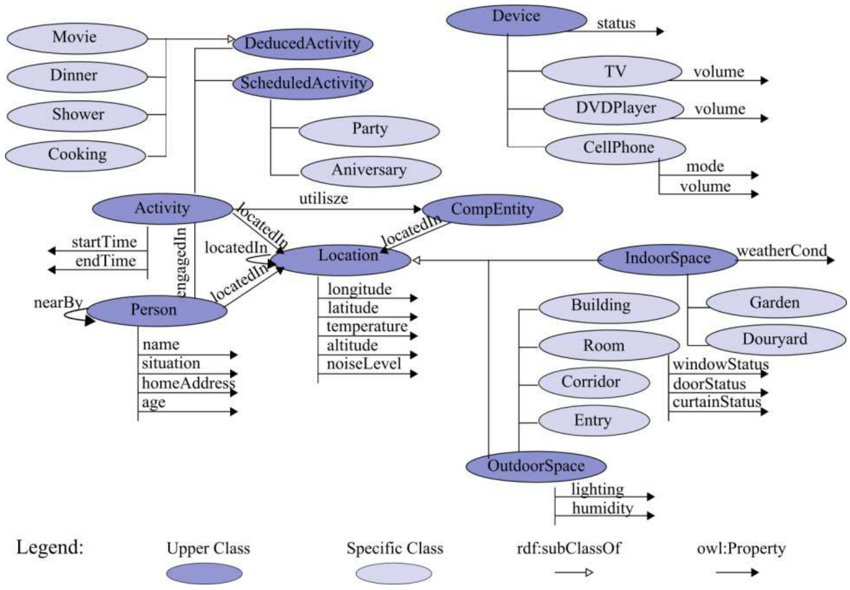

Figure 2.7. *CONON : smart home specific ontology extension.*

B. *COBRA-ONT*

COBRA-ONT [51] is a collection of ontologies, expressed in OWL, defined in the frame of Context *Brocker Architecture*, an architecture/framework designed to support context-sensitive pervasive systems. COBRA-ONT has been designed for the context representation of a smart meeting-room and consists in four sub-ontologies: *Place*, *Agent*, *Agents's Location* and *Agent's Activity*. Concepts defined in COBRA-ONT, as well as related properties and relations between these concepts are presented in Table 2.1.

The top level class in COBRA-ONT is *Place*, which represents an abstraction of the physical location of the modelled smart space. A location is described with a name, a longitude and latitude. The *Place* concept is the union of 2 concepts: *AtomicPlace* and *CompoundPlace*. A location may include other locations. For instance, the Campus and Building locations include the *Room*, *Hallway* and *Stairway* locations. The notion of capacity is represented by the *spatiallySubsumes* and *isSpatiallySubsumedBy* relations.

The ontology presenting the agents and able to act on the system is built around the Agent concept which inherits from two distinct concepts: Person and SoftwareAgent. An agent can be described by different attributes such as its name or email address. Roles, such as SpeakerAgent or AudienceAgent, are assigned to agents, through the fillsRole property. In order to deduce actions that an End-User intends to, the relation intendstoPerform of the Role concept is defined and takes as value an instance from the IntentionalAction class.

The ontology Agent's Location contains the basis for dynamic knowledge allowing to describe the localization of an agent. To do so, the LocatedIn relation is added to the Agent concept. This relation points on Place concepts. Considering that a location can be specialised in AtomicPlace and CompoundPlace, the following axioms are defined:

CoBrA Ontology Classes		CoBrA Ontology Properties	
“Place” Related	Agent's location Context	“Place” Related	Agent's location Context
Place	ThingInBuilding	Latitude	locatedIn
AtomicPlace	SoftwareAgemtInBuilding	Longitude	LocatedInAtomicPlaces
CompoundPlace	PersonInBuilding	HasPrettyName	LocatedInRoom
Campus	ThingNotInBuilding	IsSpatiallySubsumedBy	LocatedInRestroom
Building	SoftwareAgemtNotInBuilding	SpatiallySubsumed	LocatedInParkingLot
AtomicPlaceInBuilding	PersonNotInBuilding	AccessRestricted- toGenderer	LocatedInCompoundPlace
AtomicPlaceNotInBuilding		LotNumber	LocatedInBuilding
Room			locatedInCampus
Hallway	Agent's Activity Context	“Agent” Related	Agent's Activity Context
Stairway	PresentationSchedule	HasContactInformation	ParticipatesIn
OtherPlaceInBuilding	Event	hasFullName	starttime
Restroom	EventHappingNow	hasEmail	endtime
Gender	PresentationHappingNow	hasHomePage	location
LadiesRoom	RoomHasPresentationHappingNow	hasAgentAddress	hasEvent
MensRoom	ParticipantOfPresentation-	fillsRole	hasEventHappingNow
ParkingLot	HappingNow	isFilledBy	invitedSpeaker
	SpeakerOfPresentationHappingNow	intendsToPerform	expectedAudience
“Agent” Related	AudienceOfPresentationHappingNo	desiresSomeone-ToAchieve	presentationTitle
Agent	w		presentationAbstract
Person	PersonFillsRoleInPresentation		presentation
SoftwareAgent	PersonFillsSpeakerRole		eventDescription
Role	PersonFillsAudienceRole		eventSchedule
SpeakerRole			
AudienceRole			
IntentionalAction			
ActionFoundInPresentation			

Table 2.1 *COBRA-ONT concepts and related attributes and relations.*

- No agent can be present at two AtomicPlace locations within the same timespan ;
- And one agent can be present at two different *CompoundPlace* locations within the same timespan only if one location includes the other. This type of reasoning process is important for inconsistency detection when it comes to find/know the location of the agent.

It is also possible to build an agent-focused classification based on their location: *PersonInBuilding* and/or *SoftwareAgentInBuilding*. Further related classes are *PersonNotInBuilding* and *SoftwareAgentNotInBuilding*. Relying on OWL, COBRA-ONT can therefore propose reasoning options on OWL semantics. It becomes also possible to integrate in the language rules specific to the domain represented, so as to allow the detection and solving of inconsistencies, but also the interpretation of sensors-transmitted context information.

C. Standard Ontology for Ubiquitous and Pervasive Application SOUPA

SOUPA, standard ontology for OWL-based ubiquitous and pervasive applications [52], was also proposed by the same authors as CORBA-ONT. For interoperability aim, SOUPA includes different vocabularies stemming from external ontologies. Examples of ontologies from which SOUPA is using the vocabulary are *Friend-Of-A-Friend ontology (FOAF)* [53] [54], *DAMLTime* [55], COBRA-ONT [51], MoGATU BDI ontology [56] and Rei policy ontology [57]. Intended to be an ontology common to different pervasive applications, SOUPA is based on a modular structure. As Figure 2.8 shows, SOUPA is composed of 2 parts: SOUPA Core which contains generic ontologies common to all applications and SOUPA Extension which contains additional ontologies specific to applications for which they were primarily defined.

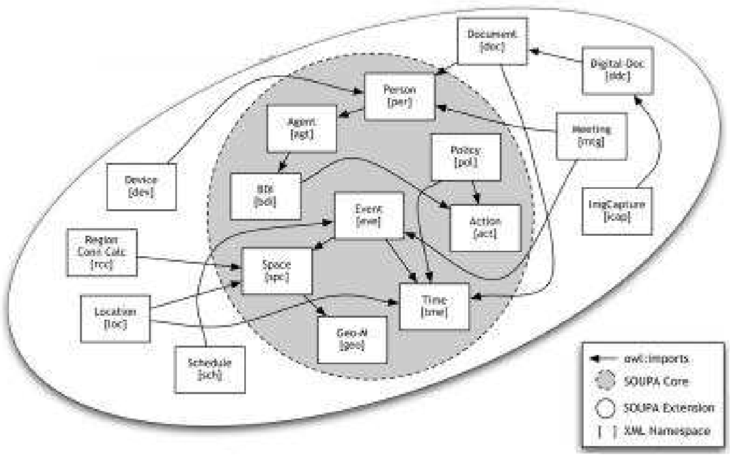

Figure 2.8. *SOUPA Ontology*.

SOUPA Core is composed of the following ontologies:

The SOUPA Person ontology is built around the Central Person concept. This ontology presents the End-User profile and related contact, social and professional information;

- The Agent & BDI ontologies present the system agents (human or software-based entities) through the explicit definition of their goals, plans, wants, intentions, convictions and duties.
- The Policy ontology provides the vocabulary needed to the definition of security and confidentiality rules. The descriptive logic that the ontology is based on makes it possible to reason based on these rules. Rules can be defined by the system administrator in order to set the access rights or by the End-User to insure the confidentiality of concerned context information.
- The Action ontology describes actions launched by system agents. To each Action concept are associated attributes describing actors of this action, entities impacted by this action, where and when the action is executed, the necessary tools to the execution of the action, etc. Access/Non-Access rights over the execution of the actions are determined by the rules set up in the Policy ontology.
- Relying on DAML Time and Entry sub-ontology of Time ontologies, SOUPA provides a set of ontologies to describe the timing and timing-related properties and relations of the major two concepts: tme:TimeInstant and tme:TimeInterval.
- The Space ontology provides the basis for reasoning on spatial relations between geographical areas, mapping geo-spatial coordinates of a geographical location upon its

symbolic representation and vice-versa. It represents geographical measurements of a location. This ontology is built in two documents: Space and Geo-Measurement. The first one presents the symbolic representation of a location and related spatial relations; the second defines the geo-spatial vocabulary (i.e. longitude, latitude, distance, etc.).

- The Event ontology presents events with spatial and temporal extensions. Events may be occurrences of different activities, forecasts and capture events.

Ontologies that compose the SOUPA extension are built based on the SOUPA Core. These ontologies aim to allow the context description for particular applicable scenarios. The two prototypes, CoBrA and MoGATU, on which SOUPA was tested, illustrate these scenarios [52].

2.2.2.7. Discussion

An evaluation of the six surveyed models is presented in [59]. The evaluation is based on six requirements that are defined as fundamental in ubiquitous computing:

- *Distributed composition (dc):* since computing systems are usually distributed, this feature is important due to the lack of a central instance being responsible for the creation, deployment and maintenance of data and services, in particular context descriptions;
- *Partial validation (pv):* as a result of the distributed composition requirement, the partial validation of contextual knowledge is particularly important;
- *Richness and quality of information (qua):* a context model should inherently support some quality and richness indicators (e.g. uncertainty, accuracy, etc.) for context information;
- *Incompleteness and ambiguity (inc):* being usually incomplete and/or ambiguous, this limit should be covered by the context model;
- *Level of formality (for):* It is important that the different computational entities composing the system have a shared understanding of a domain vocabulary;
- *Applicability to existing environments (app):* From the implementation perspective, it is important that a context model is applicable within the existing infrastructure.

Approach/requirements	dc	pv	qua	inc	For	App
Key-Value Model	-	-	-	-	-	+
Markup scheme Model	+	++	-	-	+	++
Graphical Model	-	-	+	-	+	+
Object-oriented Model	++	+	+	+	+	+
Logic-based Model	++	-	-	-	++	-
Ontology-based Model	++	++	+	+	++	+

Table 2.2. Context models' *appropriateness indication.*

The results of the study are summarized in Table 2.2. The conclusion is that ontology-based models are the ones that suit best the requirements of context modeling in ubiquitous computing, especially the formality, distributed composition and partial validation. Ontology-based models have also received much attention in the last years due to their linkage with the semantic web.

2.2.3. Context-Aware Systems and Framework

Context-awareness began to be investigated in the early of the 90s with the emergence of mobile computing. Two pioneer investigations were the Active Badge system [60], developed at the Olivetti Research Lab and the active map [61] at Xerox PARC. Since then, a wide range of context-aware systems have been reported. As location is considered as a common piece of context used in application development, most of these systems, focus on it neglecting other context information. Location-aware systems targeted different applications such as tour guide applications [62] and advertisement systems in smart spaces [63]. Location-aware systems use different technologies to acquire location information such as Global Positioning System (GPS), underlying communication infrastructures, cameras, card reader, etc.

Because context is more than location, some works combine different context information such as time, user location, activity and interest to build high-level context towards more adaptive context-aware systems. Examples of these systems are the GUIDE system [64] and the conference assistant [17]. These systems are typically proprietary and depend on the applications for which they are built and optimized to. A survey of mobile location and context-aware systems is presented in [14].

For more extensibility and flexibility, other works focus on providing a framework for context-awareness that enable easy and rapid prototyping of context-aware applications. Such generic infrastructures not only provide context consumers to retrieve context data, they also permit a simple registration of new context sources. A selection of context-aware frameworks are introduced and compared according to different design criteria in the following sub sections.

2.2.3.1. Context-aware systems Architecture

The design of a context-aware system depends on the system requirements and characteristics such as the location of context sources, the system scale, the consideration of further extension, etc. How the system’s applications acquire context is very important to determine the system architecture. Three different approaches are presented towards this [51]:

- *Direct sensor access*: in which applications software access context information directly from sensors internally located in the device in which they are deployed. This tightly coupled approach is not very used;
- *Middleware infrastructure*: this approach introduces an intermediate layer between the sensing and application layers with the intention of hiding sensing details for applications and allowing extensibility and reusability;

- *Context server*: this approach extends the last one by enabling concurrent multiple clients to access remote context sources. Another advantage of this approach is that it permits to relieve resource-limited client devices of intensive context management operations.

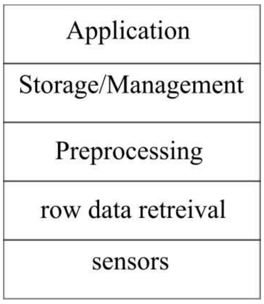

Figure 2.9. *Layered conceptual framework for context-aware systems.*

A. Sensing

Context information can be retrieved from different sources, thus sensors can be of different types. In [65], sensors are classified in three categories:

- Physical sensors: they consist in the most used sensors to capture physical data. Table 2.3 [66] presents different physical sensors that can be used in context-aware systems and frameworks.
- Virtual sensors: they consist in source context data from applications or services (e.g. electronic calendar, weather service, emails, etc.).
- Logical sensors: their role is to combine information provided by physical and virtual context sources to resolve higher task. For example, combining device location and user logins to devices in order to detect user location.

Type of context	***Available sensors***
Light	Photodiodes, colour sensors, IR and UV sensors, etc.
Visual context	Cameras
Audio	Microphones
Motion, acceleration	Mercury switches, angular sensors, accelerometers, motion detectors, magnetic fields
Location	Outdoor: Global Positioning System (GPS), Global System for Mobile Communications(GSM); Indoor: Active Badge system, etc.
Touch	Touch sensors implemented in mobile devices
Temperature	Thermometers
Physical attributes	Biosensors to measure skin resistance, blood pressure

Table 2.3. *Commonly used* physical *sensors.*

B. Row data retrieval

This layer makes use of appropriate drivers for physical sensors and APIs for logical ones to retrieve context data in a transparent and reusable manner. This layer also allows system extensibility since context sensors can be added, changed, and removed without affecting the upper layers.

C. Pre-processing

The processing layer is responsible for (1) interpreting raw-level context information and (2) deriving high context information from it by reasoning, composition and aggregation. This layer is not always implemented in context-aware systems but may be very useful since it provides a situation-level abstraction. This way, it is able to enhance significantly the relevance and the accuracy of context information and to allow reusability. Another important function implemented in this layer is conflict checking.

D. Storage and Management

This layer organises, models and stores context information in order to allow their access from client applications via public interfaces. The access is generally allowed in both synchronous and asynchronous way. As illustrated in section 2.2.2, different context models can be used and these models can be stored in different databases types.

E. Application

The last layer in composed of various context-aware applications, that use context information to adapt their behaviour and thus offer users better experience.

2.2.4. Most representative Context-Aware frameworks

In the following we give some details on representative context-aware systems. We present two 3GPP standards: the Generic User Profile Architecture and the User Data Convergence to be used with IMS, three representative works published in research papers: the Context Toolkit, CoBrA and SOCAM and the C-CAST European project proposal that deals with context-awareness and multimedia services distribution. A more in-depth survey and comparison of more context-aware frameworks are provided in [67].

2.2.4.1. 3GPP Generic User Profile (GUP)

The 3GPP Generic User Profile [23] aims to provide a conceptual description to enable harmonized usage of the user-related information located in different entities. Technically, the 3GPP Generic User Profile provides an architecture, data description and interface with mechanisms to handle the data. In this section, we focus on the GUP architecture and interfaces (the GUP content and structure is presented in section 2.2.2.2).

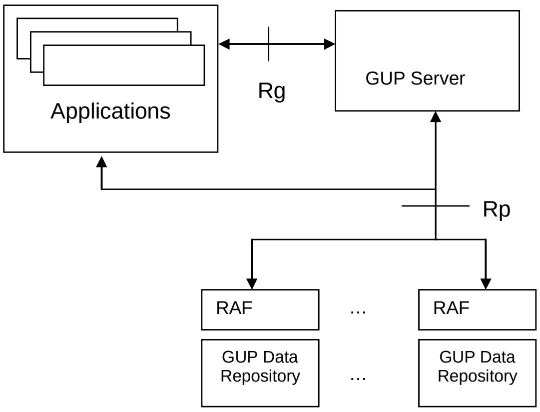

Figure 2.10. *GUP reference architecture.*

The GUP architecture [24] consists of the following functional entities, as shown in Figure 2.10:

A. GUP Server

The GUP Server is a functional entity providing a single point of access to the Generic User Profile data of a particular subscriber. The GUP Server includes the following main functionalities:

- Single point of access for reading and managing generic user profile data of a particular subscriber.
- Location of Profile Components.
- Authentication of profile requests.
- Authorization of profile requests.
- Synchronization of Profile Components.

In addition to proxying the requests (or handling them by itself), the GUP Server may also apply the redirect mode of operation for applications that support it. This implies that the GUP Server responds to the request with the redirection information such as redirection address and authorization assertions. Redirection can be made with Create, Delete, Modify, Query and Subscribe procedures.

B. Repository Access Function (RAF)

The Repository Access Function (RAF) realizes the harmonized access interface. It hides the implementation details of the data repositories from the GUP infrastructure. In addition, the RAF may take part in the authorization of access to such GUP information, under the control of the RAF.

C. GUP Data Repositories

Each GUP Data Repository stores the primary master copy of one or several profile components. It is assumed that the RAF and the GUP Data Repositories are usually co-located in the same network element. The GUP Data Repository may also contain the authorization data depending on the authorization model and architecture.

D. Rg and Rp reference points

The Rg reference point allows applications to create, read, modify and delete any user profile data using the harmonized access interface. This reference point supports also third party profile access. There are means to authorize all requests and protect the user's privacy in all operations. The defined procedures applied in the Rg reference point between the applications and the GUP Server are: Create, Delete, Modify, List, Query, Subscribe, Unsubscribe and Notify.

The Rp reference point shall allow the GUP Server or operator's own applications to create, read, modify and delete user profile data using the harmonized access interface. Rp is an intra-operator reference point. External applications and third party GUP data repositories shall be connected to the GUP Server only using the Rg reference point.

The Rg and Rp reference points carry user related data, and therefore shall be protected by security mechanisms to protect user's privacy.

E. Applications

These are the Application Servers and applications that need access to GUP data components. They may host some GUP data components themselves and may act as RAFs. Third party applications belonging to external security domains shall use a discovery service in a secured way to discover the GUP Server.

2.2.4.2. IMS - User Data Convergence (UDC)

User Data Convergence (UDC) [40] concept supports a layered architecture separating the data from the application logic, so that user data converge from where it belonged to a logically unique User Data Repository (UDR), where it can be stored according to the Information Model presented in section 2.2.2.4, managed and accessed in a common way. Convergence in data model avoids data duplication and inconsistency, overcome the data capacity bottleneck of a single entry point, simplify the overall network topology and interfaces and consequently simplify the development and deployment of new integrated services through a common and unified set of user data that are up now, scattered in several domains (e.g. PS, CS, IMS) and different network entities (e.g. HLR, HSS, Application Servers) of the current 3GPP system, as shown in Figure 2.11.

The UDR is the functional entity that acts as a single logical repository of user data and is unique from application Front End's perspective. Applications Front Ends (FE) are functional entities such as HLR/HSS/AUC, Application Servers, Access Network Discovery and Selection Functions in the Home Network (H-ANDSF), etc. that access the user data stored in UDR according to the UDC Information Model detailed in section 2.2.2.4. Application FEs are only able to access user data after authentication and authorization. Application FEs should then support common security algorithms and keys.

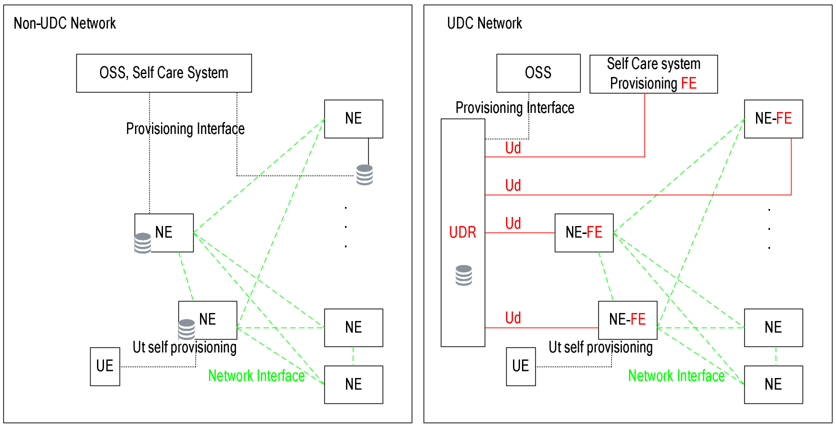

Figure 2.11. Comparison *Non-UDC Network to UDC Network [68]*.

UDR provides unique reference point called Ud that allows different FEs to create, modify and delete user data. Ud should also support the subscription/notification functionality which allows a relevant FE to be notified about a specific event which may occur on a specific user data in UDR and transaction. More information on the Ud's procedures and flow can be found in [68].

2.2.4.3. Context Toolkit

Inspired from the GUI Toolkit, the Context toolkit [69]-[15] aims to provide a reusable solution that makes easier the development of context-aware applications by insulating them from context sensing. The Context Toolkit relies on the concept of context widget. As the GUI widgets mediate between the application and the user, context widgets mediate between the application and its operating environment. However, dealing with context rather than user inputs raises additional issues due to (1) the heterogeneity and distribution of context sources, (2) the need to abstract context information to suit the expected needs of applications, (3) the fact that context widgets do not belong to the application as the GUI widgets do.

As shown in Figure 2.12, the Context Toolkit is composed of three main components:

- Widgets that provide reusable and customizable building blocks of context sensing. Each widget encapsulates information about a single piece of context such as location or activity. They also provide a unique interface to context information, hiding thus the context sensing complexity for applications;
- Aggregators that can be seen as meta-widgets. In addition to the capabilities of widgets, aggregators can also aggregate context information of real world entities;
- And finally, interpreters that are in charge of abstracting raw or low-level context information into higher level information.

The components of the Context Toolkit are deployed in a distributed architecture and run independently of any single application allowing thus their use by multiple applications. To support transparent distribution, all components share a common communication mechanism (XML over HTTP).

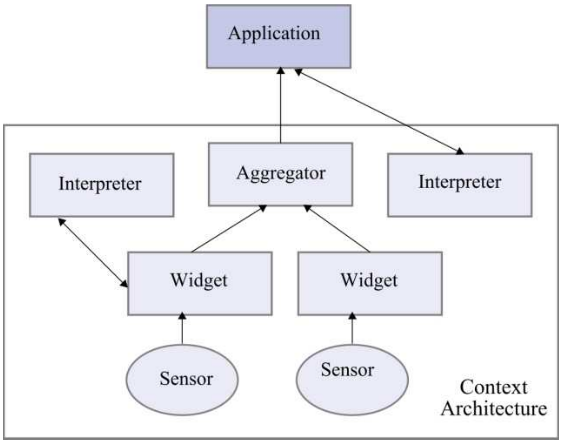

Figure 2.12. *Sample configuration of context component.*

In [15] authors discuss five applications that have been implemented with the Context Toolkit, including the Conference Assistant, In/Out Board, Context-Aware Mailing List, Dynamic Ubiquitous Mobile Meeting BOard (DUMMBO) and Intercoms.

2.2.4.4. Context Broker Architecture (CoBrA)

CoBrA [51] is a broker-centric agent architecture for supporting context-aware computing in smart spaces. Intelligent spaces are physical spaces (e.g. offices, meeting rooms, vehicles, etc.) that are populated with intelligent systems providing pervasive services to users. The key component of the CoBrA is the context broker, an intelligent agent in charge of maintaining a shared model of context on behalf of a community of agents and devices. It is also in charge of protecting the privacy of users by enforcing the user-defined policies when sharing information with other agents. All computing entities in a smart space are assumed to be aware of the presence of a context broker, and the high-level agents are presumed to communicate with the broker using the standard FIPA Agent Communication Language [22].

The design of the context broker and its relations with other agents, as shown in Figure 2.13, comprises the following four functional components:

- Context Knowledge Base: it ensures a persistent storage of the context knowledge (COBRA-ONT introduced previously).
- Context Reasoning Engine: a prototyped OWL inference engine called F-OWL. This inference engine is implemented using Flora-2. Key features of F-OWL include the ability to reason with the OWL ontology, the ability to support knowledge consistency checking using axiomatic rules defined in Flora-2, and an open Application Programming Interface (API) for Java application integrations.
- Context Acquisition Module: a library of procedures that form a middleware abstraction for context acquisition.
- Policy Management Module: a set of inference rules that deduce instructions for deciding the right permissions for different computing entities to share a particular piece of contextual information and for selecting the recipients to receive notifications of context changes.

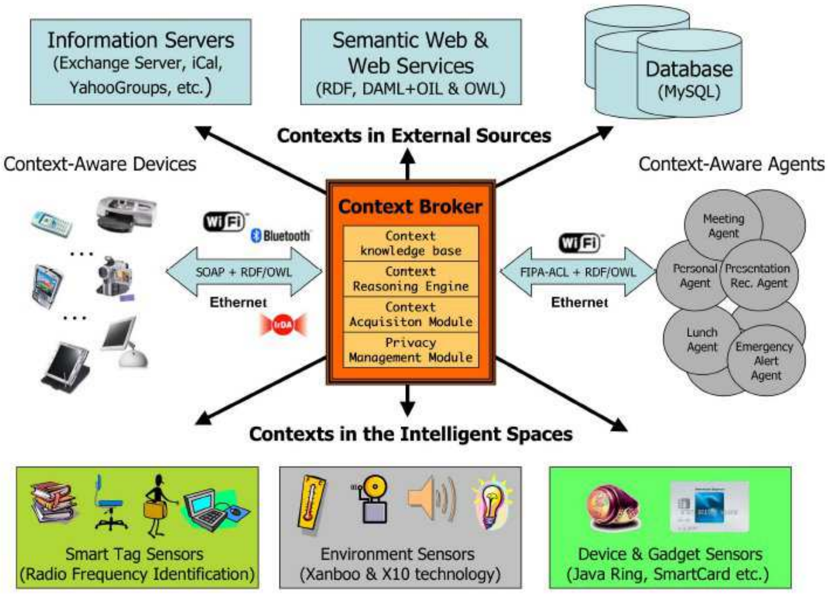

Figure 2.13. *Context Brocker Architecture design.*

CoBrA addresses two key issues in pervasive computing: (1) supporting resource-limited mobile computing devices and (2) addressing the concerns for user privacy. With the introduction of a context broker that operates in a resource-rich stationary computer, the complexity of acquiring and reasoning over contextual information is shifted away from the resource-limited mobile devices to the resource-rich broker. In addition, the complications inherent in establishing, monitoring and enforcing security, trust, and privacy policies are simplified in the presence of a centralized manager. However, the centralized design of a broker could create a “bottleneck” situation in a large scale intelligent space. To overcome this problem, multiple brokers could be grouped together to form a broker federation. Each broker is responsible for a part of the intelligent space. The federated brokers are organized according to some communication structure (e.g. peer-to-peer or hierarchical), and they periodically exchange and synchronize contextual knowledge.

2.2.4.5. Service-oriented Context Aware Middleware (SOCAM)

The Service-oriented Context Aware Middleware (SOCAM) [70] aims to provide an efficient infrastructure support for rapid prototyping of context-aware services in pervasive computing environments. SOCAM is a distributed middleware that allows an easy share and access of context information by context-aware services. The SOCAM architecture, presented in Figure 2.14, consists of the following components:

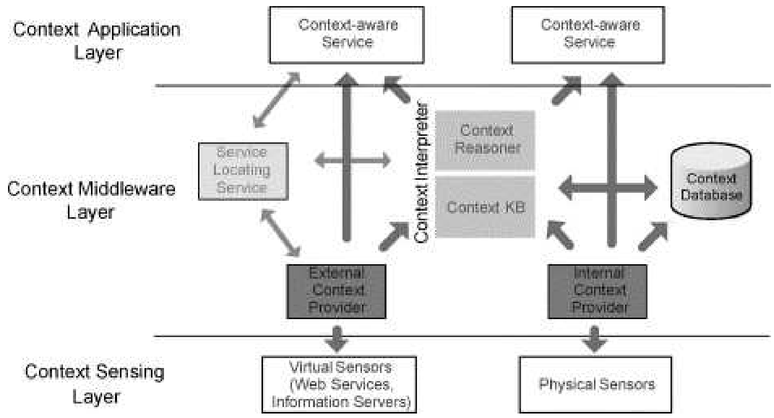

Figure 2.14. *SOCAM architecture overview.*

A. Context providers

Context Providers abstract low-level context from heterogeneous sources that could be external (e.g. a weather information server) or internal (e.g. RFID-based indoor location server). Context information is then converted to OWL representations so that it can be shared and reused by other service components.

B. Context interpreter

The context interpreter consists of a context reasoner and a context Knowledge Base (KB). The context reasoner is responsible for deriving high-level context from low-level context, querying context knowledge, maintaining the consistency of context knowledge and resolving context conflicts. It also acts as a context provider as it can provide deduced contexts. Two kinds of reasoning are supported: ontology reasoning (RDFS reasoning and OWL reasoning) and user-defined rule-based reasoning (first-order-logic rule-based reasoning). The context KB contains a domain specific context ontology and its instances that are pre-loaded into the context KB during system initiation. The context KB provides a set of APIs for other service components to query, add, delete or modify context knowledge.

C. Service locating service

The service locating service allows users and applications to discover contexts and locate context providers and context interpreters that have already advertized themselves.

The service locating service is able to track and adapt to the dynamic changes of context providers.

D. Context-aware services

Context-aware services are agents, applications and services that make use of context and adapt their behavior accordingly.

SOCAM components are designed as independent service components which may be distributed over heterogeneous networks and can interact with each other. The communication between these components is based on Java RMI. Context dissemination is done in both push and pull modes. A set of procedures and APIs to support both context query and context event subscription mechanisms are provided. The SOCAM middleware is built on top of the OSGi service platform to provide a middleware level support for context-aware systems and tested with the vehicle-domain ontology [70].

2.2.4.6. EU Project Context CASTing (C-CAST)

The EU project Context CASTing (C-CAST) [71] aims to evolve mobile multimedia multicasting to exploit the increasing integration of mobile devices with our everyday physical world and environment. The objective is to provide an end-to-end context-aware communication framework specifically for intelligent multicast-broadcast services. The framework has two facets; the first one, which we are interested in, is the creation of context-awareness, the other is the service (or content) transport and delivery. These two facets are tied together by service enablers and adaptation functions.

Figure 2.15 illustrates the functional architecture of the C-CAST context management system that comprises the following functional components:

A. Context Provider (CxP)

A Context Provider (CxP) is the component that supplies the system with context data. CxP achieves this by gathering data from a collection of sensors, network, services (e.g. web services) or other relevant sources. Each provider is tailored to provide a particular type of context. Moreover, a CxP can produce new high level context information from row level sensors or other source data. Towards this, it includes the following functionalities:

- Filtering: select the requested context information;
- Fusion: derive stable measurement values, e.g., by averaging;
- Aggregation: combine heterogeneous context information to derive new, higher level context, e.g. location + temperature + humidity = weather;
- Low level reasoning: use logical rules to derive conclusions about user situation from fused and aggregated sensor data and network context.

Every CxP advertises its availability and capabilities to the Context Brocker and exposes interfaces to provide context information to both the Context Consumers and the Context Brocker either synchronously or asynchronously.

B. Context Consumer (CxC)

The Context Consumer (CxC) is the architectural component that uses context data. A CxC can retrieve context information by requesting the CxB either in explicit or implicit (through subscription to context update event) way or by directly invoking a CxP over a specific interface. Usually, applications, services and the service enablers are pure CxCs. The main CxCs, considered in the system, are: Applications, Group Management Enabler, Content Selection Enabler, Context Cache and Context History.

C. Context Brocker (CxB)

The Context Broker (CxB) is the key component of the architecture. It works as a handler of context data and as an interface between other architectural components. The CxB provides a CxP Lookup Service. It contains a registry of all Context Providers and their capabilities. The related information is obtained through an advertising process. To accelerate the request

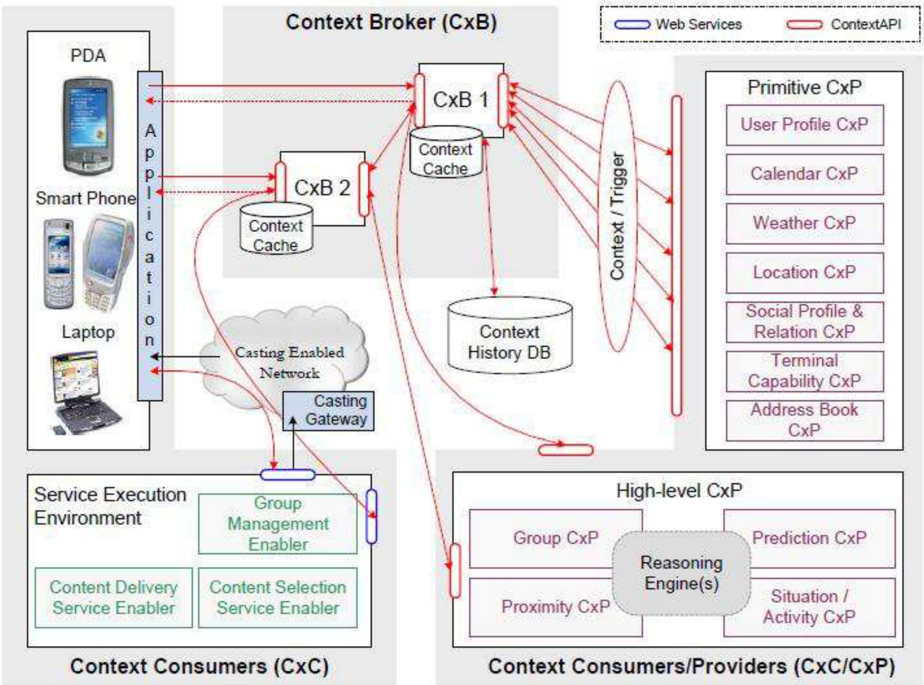

Figure 2.15. *The functional architecture of the C-CAST context management system.*

process, the CxB maintains a Context Cache to store context which has not expired yet. The expired context is moved from the Context Cache to a Context History database.

The C-CAST context Model uses the entity – scope association. An entity is the subject, which context data refers to, and it is composed of two parts: a type and an identifier. A type is an object that identifies a set of entities. An identifier specifies a particular item in a set of entities belonging to the same type. All context information set within C-CAST is defined as "scope" which is a set of closely related context parameters.

The schema for context representation and communication between components is defined in ContextML, a XML-based language. ContextML allows for the representation of context information, context meta-information, context data access messages and control messages exchanged between CxC, CxP and CxB.

2.2.4.7. Summary

The summary and comparison of the surveyed frameworks in this section is given in Table 2.4. The systems are compared based in different design criteria that strongly affect their performance such as their architecture, the resources discovery technique, the sensing technologies that are used to retrieve context information, the consideration of context information history and finally the security and privacy techniques that the system implement.

Design criteria/ frameworks	Architecture	Sensing	Context model	Context processing	Resource discovery	Historical Context data	Security & privacy
GUP	Distributed data repositories with a centralised access point	Repository Access Function (RAF)	XML Schema of the GUP + GCL	Data validation	Rg &Rp reference points	Available	Liberty ID-WSF
UDC	Centralized logical repository (UDR)	Front ends (FE)	Comon Baseline Information model	Context data convergence and federation	Ud reference point	Available	to be handled with SA3
Context toolkit	Widget based	Context widget	Attribute / value model	Interpretation and aggregation	Discoverer component	Available	Context ownership
CoBrA	Agent based	Context acquisition module	Ontologies (OWL)	Inference engine and knowledge base	n.a.	Available	Rei Policy language
SOCAM	Distributed with centralized server	Context providers	Ontologies (OWL)	Reasoning engine	Services locating services	Available	n.a.
C-CAST	Distributed context Brockers	Context Providers	ContextML	Low-level reasoning engine	Context Provider Lookup Service	Available	n.a.

Table 2.4. *Summary of context-aware frameworks.*

In this first section of the state of the art, we have surveyed all the existing and foreseen solutions for performing context-awareness. Based on them, our objective is to create one fitting the most adequately Future Media Networks, in order for them to take full benefit of contextual information and provide EUs with advanced features, capabilities and services, such as adaptation, personalization, seamless device and network mobility, etc. We will now present a survey on multimedia content delivery techniques used today and foreseen in future networks.

2.3. State of the art: Internet multimedia content delivery

Besides classical client-server content delivery method, we can classify the Internet multimedia delivery platforms into two main categories according to the level of management they are able to provide: (1) managed platforms integrating QoS functionalities and (2) unmanaged best effort delivery platforms with no control on resource availability, service quality and user experience. The second category includes essentially Content Delivery Networks (CDNs) and P2P networks.

2.3.1. Managed delivery platforms

Advances in access networks technologies along with the improvement of media coding algorithms led to the deployment of the Internet Protocol based Television (IPTV) over different broadband access networks. Digital Television, started as satellite and terrestrial, can now be delivered over fixed and mobile broadband networks. As of today, most of the major European telecom Service Providers provide triple-play (telephone, Internet access and IPTV) services. Even though open and standardised approaches emerge, the majority of managed delivery platforms rely on proprietary solutions only deployed and accessible on the operator’s network.

2.3.1.1. Proprietary solutions

IPTV refers to different video services such as Broadcast services - BC (live TV and radio channel feeds broadcasted or multicasted over the network), Content on demand – CoD - services (generally unicast services provided on subscriber’s demand, e.g. VOD) and Personal Video Recorder - PVR - services (services which allow recording, pause or time shift capabilities for live content).

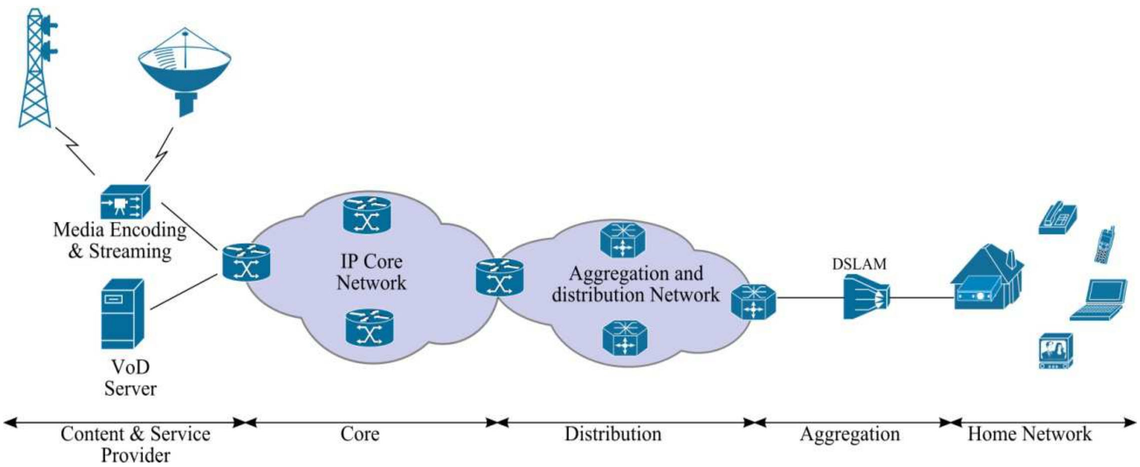

Figure 2.16. *Managed IPTV delivery network.*

As depicted in Figure 2.16, IPTV services are delivered over different network segments. At the Core, it is usually transmitted over fiber optic backbones using Multiprotocol Label Switching (MPLS) to ensure QoS. The distribution and aggregation network can be ATM (Asynchronous Transfer Mode) or Ethernet with a strong move to Ethernet for cost reasons.

Initially, broadband access used Asymmetrical DSL (ADSL) but it moved rapidly to Very high speed DSL (VDSL) and VDSL version 2 (VDSL2). Other Broadband access technologies such as Fiber-To-The-Home (FTTH) and coaxial cable are also used to support HD content distribution or interactive TV. The broadband connectivity is shared by all the home terminals (TVs, PCs, IP phones, etc.). The IP connectivity is established by the Residential or Home Gateway RG that can be or not integrated with the DSL modem.

Multicast communication scheme is often used to deliver live IPTV streams to multiple home networks. The list of subscribers is managed by Internet Group Management Protocol (IGMP) [72]. Audio and video are multiplexed in an MPEG2 Transport Stream (MPEG2-TS), and can be directly transmitted over User Datagram Protocol (UDP). However, an RTP [73] encapsulation is recommended especially for enabling quality monitoring. In the case of VoD services, if RTP is used, it is most often complemented by RTSP [74] for streaming control functions like play, pause, stop, etc.

An example of such solutions is Microsoft Mediaroom IPTV Platform [75] used by over 40 of the world's leading operators, delivering services to more than 7 millions consumer households.

2.3.1.2. Standardised solutions

For enabling easier deployment and better inter-operability, many stadardization efforts have been made in different organizations and fora such as ITU-T, ETSI TISPAN IPTV, 3GPP MBMS, etc. The standardization works aim to integrate IPTV in NGN architectures, enabling thus IPTV functions to interact with relevant NGN subsystems and use the capabilities they provide, namely dynamic network attachment and management of transport resources with quality of service control. Two directions are taken in standardization bodies. The first aims to integrate the current IPTV solutions in NGN architectures and the other one is the IMS-based IPTV that consists of using the IP Multimedia Subsystem (IMS) [76] as a control plane for IPTV services.

IMS, introduced first by the wireless standards body (3GPP/3GPP2) as an architectural framework dedicated to deliver IP multimedia services over packet based core network within third generation mobile networks and extended by European Telecommunications Standards Institute (ETSI)/Telecoms Internet converged Services Protocols for Advanced Networks (TISPAN), is adopted in the NGN standardization and integrated in the NGN IMS-based services platform. Different NGN IMS-based IPTV platforms have then been proposed, exploiting the IMS functionality (e.g. subscription and session management, fixed and mobile convergence, handover between devices, etc.) to support IPTV services. IMS-based IPTV is expected to provide not only basic IPTV but also quadruple play and enhanced services [77].

In [78], the authors identify four steps in the migration towards NGN-based IPTV architecture:

- *Non NGN-based IPTV architectures*, as highlighted previously, are the currently deployed solutions. They use proprietary IPTV middleware for service control and delivery but still some interworking with NGN subsystems can be achieved.

- *NGN-non IMS-based IPTV architectures* that enable interaction and interworking over specified reference points between IPTV dedicated functions and some existing NGN elements such as transport control elements for the Resource Admission and Control Subsystem (RACS), the Network Attachment Subsystem (NASS) or NGN user profiles. In this step, a dedicated IPTV subsystem within NGN is used to realize personalized, value-added IPTV features and to use network resources more efficiently.
- *IMS-based IPTV architectures* that specify IPTV functions based on the IMS subsystem and enable reusing of IMS functionality and SIP-based service initiation and control mechanisms.
- *NGN non-IMS and IMS-converged IPTV architectures* that are seen as a combination and a convergence of non-IMS and IMS-based IPTV architectures in a common configuration to provide converged types of IPTV services.

In the following, we introduce some standadization works done in the TISPAN, ITU-T and the Open IPTV Forum regarding the aforementioned IPTV architectures.

A. TISPAN IPTV

Several specifications of TISPAN NGN Release 2 and 3 address the integration of IPTV in NGN. The specifications consider IPTV in terms of service requirements, functional architecture, including functions definition, communication protocols, reference points and implemented procedures and communication flows. The specification in [79] is more generic and considers NGN subsystems in general and the one in [80] is IMS-specific and addresses the IMS-based IPTV architecture that relies on IMS for the session control.

Figure 2.17 depicts the different functional entities that compose the IMS-based IPTV architecture. Most of them fit the ones defined in NGN integrated IPTV subsystem. IPTV services execution involves the IPTV Media function that includes the Media Control Function (MCF) and the Media Delivery Function (MDF), as well as the Service Control Function (SCF) that is in charge of service management. The main function of the latter consists in service authorization during session initiation and session modification, including checking IPTV users' profiles in order to allow or deny access to the service. The IMS user profile and the IPTV specific profile data are held by the UPSF. The Service Discovery Function (SDF) and Service Selection Function (SSF) are functions for providing information necessary to the UE to select an IPTV service.

The communication between the UE and the SCF for session management purpose is transferred via the Core IMS. The Ut reference point can also be used for the purpose of service profile configuration. Media Control messages are exchanged between the UE and the MCF via the Xc reference point in order to control the media flow, and Media Data is exchanged between UE and MDF via the Xd reference point to deliver it.

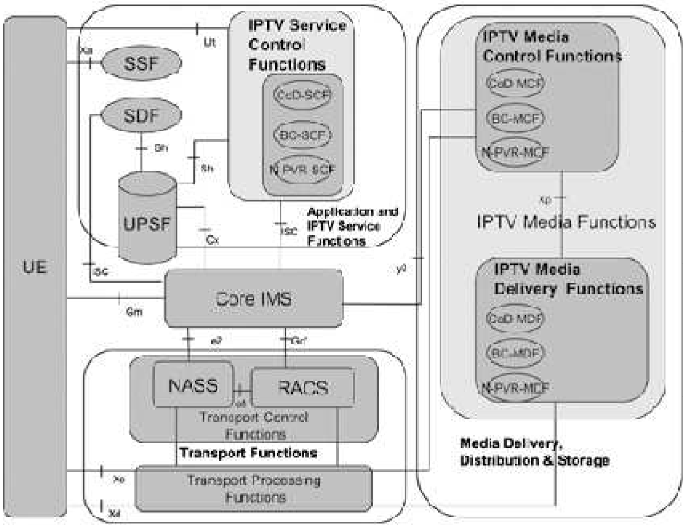

Figure 2.17. *ETSI/TISPAN Functional architecture for IPTV services.*

B. ITU-T FG IPTV

The standardization, under the ITU-T Focus Group IPTV [81]-[82], investigates IPTV regarding to IPTV roles, services, architecture, middleware, security, etc. Four functional domains are considered while designing the IPTV architecture and functions: content providers, service providers, network providers and end users or customer domains. Based on the defined roles of the latter domains, different components to be included in the functional architecture such as End-User Functions are defined. The identified functions are:

- End-User Functions: include those functions normally provided by the IPTV terminal and the End-User Network.
- Application Functions: provide the End-User Functions for IPTV services, enable the End-User Functions to select, and purchase if necessary, an item of content.
- Content Delivery Functions: provide content which is prepared in the Application functions via the Network Transport Functions. Contents may also be stored/cached in Content Delivery Functions. The latter also provide the capability to facilitate interaction between the End-User Functions and selected content, such as playback control (trick play functionality with VoD and Network PVR). Physically, the Content Delivery Functions employ the resources of Transport network Functions for the content delivery.
- Service Control Functions: provide the functions to request and release the network and service resources required to support the IPTV services.
- Management & Monitoring Functions: manage overall system status monitoring and configuration. This set of functions may be deployed in a centralized or distributed manner.
- Content Provider Functions: provided by the entity that owns or is licensed to sell content or content assets.

- Network Transport Functions: are the combination of network transport and control functions

The IUT-T IPTV functional architecture is defined in different approaches depending on the underlying network architecture. Three architectures are designed: the non NGN architecture, the NGN-based non IMS architecture and the NGN IMS-based architecture. However, the defined architectures share several functional entities and similarities. Figure 2.18 provide a functional decomposition of the IPTV Architecture in which the high-level introduced functional components are expanded. The red and green lines represent the connectivity between the functional components in the case of NGN IMS-based and the NGN non IMS-based architectures respectively.

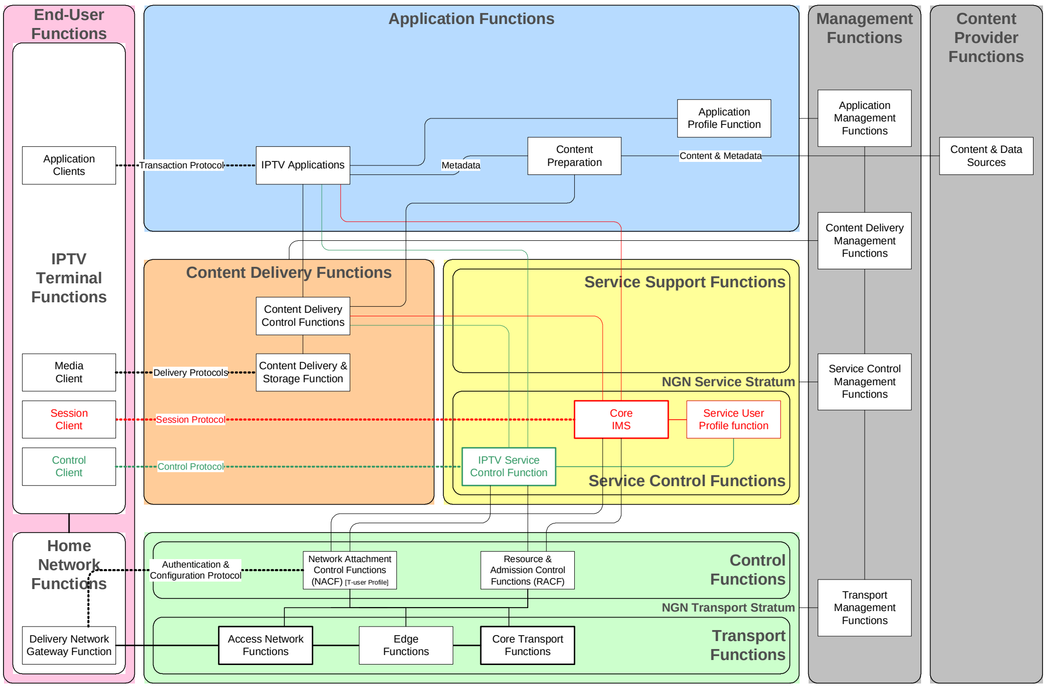

Figure 2.18. *ITU-T NGN Combined IPTV Architecture*

C. Open IPTV Forum

The Open IPTV Forum has developed an end-to-end solution to allow any consumer end-device, compliant to the Open IPTV Forum specifications [83], to access enriched and personalized IPTV services either in a managed or a non-managed network. To that end, the Open IPTV Forum focuses on standardizing the User-to-Network Interface (UNI) in both cases (managed and non-managed).

Figure 2.19 shows a high-level logical view of the scope of the Release 2 Solution in terms of networks and functional entities in the residential network. Managed Network IPTV Services are provided from within an operator’s core network, enabling the Service Provider to make use of service enhancement facilities like multicast delivery and QoS provision. Open Internet IPTV Services are accessed via an independently operated access network, with or without QoS guarantees. Open Internet IPTV services may be accessed via a service platform (e.g., a portal) that provides supporting facilities for multiple Service Providers.

The Open IPTV Forum specifications identify the different services to provide and the different functions to enable attractive and innovative ways to deliver them, such as service provisioning, service access and control, service and content navigation, interactive application platforms, content and service protection where applicable, and interworking with DLNA-compliant home network devices. The complete set of IPTV protocols to be used, the reference point interfaces (User Network Interfaces, Home Network Interfaces, Network Provider Interfaces and interfaces to external systems, e.g. the DLNA home network), the media format and content metadata, the application environment and the authentication and service and content protection are also specified.

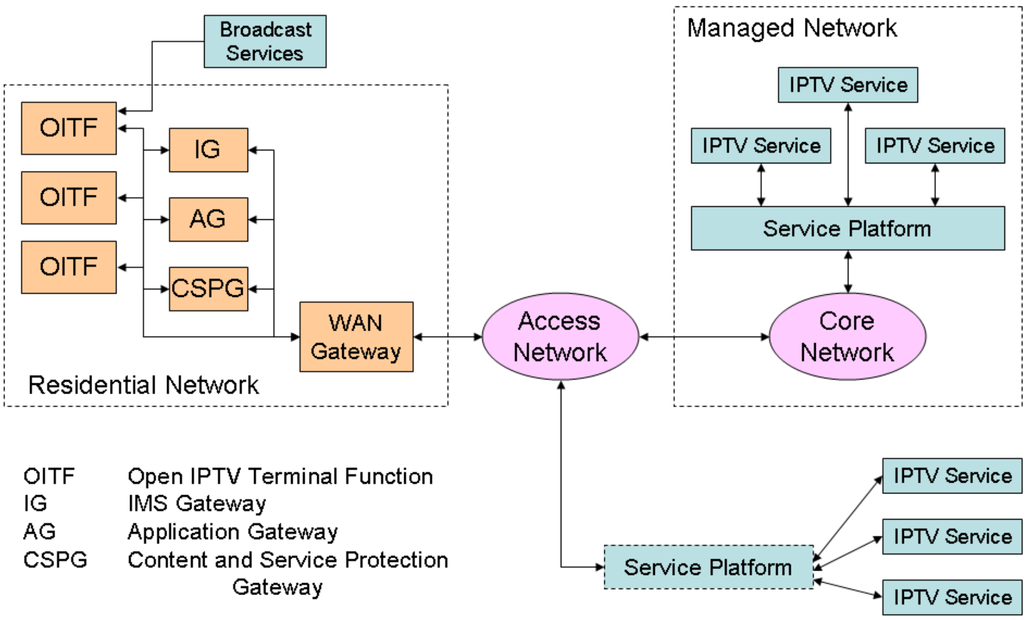

Figure 2.19. *Open IPTV Forum solution scope*

Most of current IPTV services are built on vendor-specific platforms without integration with Next-Generation Network (NGN) subsystems, usually quite heavy and costly to deploy. These proprietary closed solutions are caracterized with a tight collaboration between the Service Provider and the Network Provider. They are commonly exploited, in a non-standardised way, by telecommunications operators having their network deployed, for their own IPTV services.

2.3.2. Best-effort content delivery networks

With the increasing Internet growth, the content delivery infrastructure scalability, reliability and performance is becoming critical. A key challenge lies on delivering more and more complex and personalized contents to rapidly growing end user population. Such a delivery context led Service providers to further rely on delivery-dedicated infrastructures such as Content Delivery Networks (CDN) and peer-to-peer (P2P) networks for assuring the delivery of their contents.

Caching and replication are the key technologies used by the latter delivery infrastructure to improve service performance. Multiple copies of contents are maintained in geographically distributed nodes organized in several clusters. The aim is to bring contents closer to end users. The higher is the distribution, the more scalable and efficient is the content delivery. Indeed, a high distribution strategy means short distances between content servers and consumers and, consequently, evolves latencies and reduces network bandwidth consumption and opportunities that packet loss occurs, all these towards a better user experience.

CDNs and P2P networks have the same objective of distributing contents to end users in a more scalable way. However, there is a significant difference in their design that affects the delivery performance, workload and the way that caching and replication techniques may be implemented. In the following sub-sections, we present an overview of these two solutions.

2.3.2.1. Content Delivery Networks (CDN)

Content Delivery Networks (CDNs) have emerged at the end of the 90's to overcome the scalability and performance problems of Internet services. They are now considered as the primary solution for content delivery over Internet. By mean of replication, CDNs maximize bandwidth, improve accessibility, and maintain correctness [84]. The main concept is the delivery at edge points of the network, bringing thus contents closer to end users and consequently improving end users perceived quality while minimizing the delivery cost.

To deliver their contents in a reliable and timely manner, more and more service/content providers are then contracting with commercial CDNs providers to host and distribute their contents; offloading thus their servers to the CDN infrastructure. The use of CDNs has several advantages:

- Offloading the origin servers;
- Reducing the service/content providers' investments in the delivery infrastructure deployment and the management;
- Bypassing traffic jumps and avoiding peering traffic by bringing the contents closer to end users;
- Improving content delivery quality, speed and reliability;

Many services, such as web-based services, file transfer services, streaming media services, etc., can take advantage from CDNs infrastructure to deliver their contents. To deliver these services and contents in an efficient way, the design of CDN requires some important features such as replica placement, cache organization, surrogate selection and request routing mechanism. Since the delivery of live and on-demand streaming are more challenging due to the large size of delivered contents and to the long life of the streaming sessions, CDN replica servers should implement additional functionalities such as large and shared caching capabilities, peering capabilities, transcoding capabilities and streaming session handoff. The CDN features are described in detail in the following subsections in terms of architecture and functionalities.

A. Architecture

Figure 2.20 shows the layered architecture of a Content Delivery Network [85], which consists of the following layers:

- The *basic fabric* consists of the infrastructural hardware resources such as file servers, clusters, network infrastructure and the system software that they run such as operating systems, content indexing and management systems, etc.
- The *communication & connectivity* functions provide the core Internet protocols (e.g. TCP/UDP, FTP, etc.) as well as the CDN-specific Internet protocols (e.g. ICP, HTCP, CARP) and security protocols (e.g. PKI, SSL). This layer includes also application specific overlay structures to provide efficient content search and retrieval.
- The *CDN* services layer provides the core CDN functionalities such as surrogate selection, request routing, caching, geographic load balancing, SLA management, etc.;
- At the top of the CDN architecture, the *End-User* is typically a web user who is willing to access a service or content by specifying the service/content provider URL.
- Related to streaming services, the CDN architecture can be completed by an intermediate layer between the *End-User* and the *CDN* services, which is the *Online Video Platform (OVP)*. It is in charge of the additional media streaming application related functionalities such as transcoding, content annotation, etc. Examples of OVP providers are Brightcove, Ooyala, Stream OS, etc.

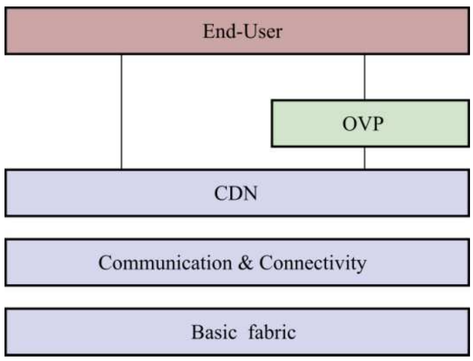

Figure 2.20. *Layered architecture of CDN.*

B. Surrogate placement

One key issue in Content Delivery Networks is replica placement. The problem consists of optimizing the placement of edge servers across the Internet with the aim of reducing the user perceived latency while optimizing the necessary bandwidth to maintain the replicas caches.

To resolve the replica placement problem, two graph theoretic approaches are used to both determine the number and the placement of replicas: the *k-hierarchically well-separated trees* (k-HST) [86] and the minimum k-center problem [87]. The k-HST algorithm consists of two phases. In the first phase, the graph is recursively partitioned as follows: A node is arbitrarily selected from the current (parent) partition, and all the nodes that are within a random radius from this node form a new (child) partition. The value of the radius of the child partition is a factor of k smaller than the diameter of the parent partition. This process is recursively reproduced for each partition, until each node is in a partition of its own. The tree of partitions is then obtained with the root node being the entire network and leaf nodes being individual nodes in the network. In the second phase, a virtual node is assigned to each of the partitions on each level. Each virtual node in a parent partition becomes the parent of the virtual nodes of the child partitions and together the virtual nodes form a tree. The greedy algorithm is further used to determine the number of centers needed when the maximum center-node distance is bounded. The minimum k-center problem is defined in the following steps:

- Given a graph $G(V,E)$ with all its edges arranged in non-decreasing order of edge cost c: $c(e_1) \leq c(e_2) \leq \cdots \leq c(e_m)$ construct a set of square graphs $G_1^2, G_2^2, \dots, G_m^2$. Each square graph of G denoted by G^2 is the graph containing nodes V and edges (u,v) wherever there is a path between u and v in G.
- Compute the maximal independent set M_i for each G_i^2 . An independent set of G^2 is a set of nodes in G that are at least three hops apart in G and a maximal independent set M is defined as an independent set V'such that all nodes in $V - V'$ are at most one hop away from nodes in V' .
- Find smallest i such that $M_i \leq K$, which is defined as j.
- Finally, M_j is the set of K centers.

For the static case where the network topology and the workload pattern are a priori known, some heuristics have been proposed. Li et al. proposed a tree-based placement algorithm, in [88], relying on the assumption that the underlying topologies are trees, and modeled it as a dynamic programming problem. The algorithm was originally designed for Web proxy cache placement, but it is also applicable for Web replica placement. At a very high level, they divide a tree T into several small trees T_i, and show that the best way of placing $t \geq 1$ proxies in the tree T is to place t_i' proxies the best way in each small tree T_i, where $\sum_i t_i' = t$. In [89], Jamin et al. present the greedy algorithm which chooses M replicas among N potential sites. One replica is chosen at a time. In the first iteration, each of the N potential sites is evaluated individually to determine its suitability for hosting a replica. The cost associated with each site, under the assumption that accesses from all clients converge at that site, is computed and picks the site that yields the lowest cost. In the second iteration, the second replica site which, in conjunction with the site already picked, yields the lowest cost is selected. In this iteration, the cost is computed under the assumption that clients direct their accesses to the nearest replica. The process is iterated until choosing M replicas. The hot spot algorithm [89] differs from the greedy algorithm only in selecting the best candidate in each step. It places replicas at the top sites that generate maximum traffic. The fanout algorithm, proposed in [90], selects a replica site with the maximum out-degree at each step and calculates the total cost. The process stops when the optimality condition is satisfied. In [91], authors investigate the problem of replica placement and formulate a new model for the problem that accommodates the characteristics of multimedia content delivery.

C. *Cache organization*

The replication is the commonly used technique to improve performance and reliability of distributed systems, such as CDNs. In the latter, replication is performed trough caching contents in the edge server to ensure their delivery in a timely manner with bandwidth consumption as low as possible. CDN performance is highly dependent on the cache organization, including the caching techniques used, the server cache update and the integration of caching policies in the CDN infrastructure. Taxonomy of caching techniques is presented in [85]. They are classified in intra-cluster caching and inter-cluster caching. The intra-cluster caching techniques consist of query-based scheme [92], digest-based [93], directory-based [94], and hashing-based schemes [95]-[96]. However, the commonly used one is the digest-based technique. In the inter-cluster caching, the most used technique is the request-based one.

Another issue that affects the CDN performance is the insurance of the cache consistency. To update the servers' caches, different caching techniques are deployed by CDNs. The most common one is the *periodic update* in which the origin servers periodically provide instructions to caches about what contents are cachables and for what time they do not need to. For this, the expiration times are associated to contents. The *update propagation* is triggered with a change related to the contents. This approach consists of delivering the content to CDN caches whenever this content is changed at the origin server side. In an *On-demand* update, the latest copy of content is delivered to CDN servers when the content is requested. Finally, in the *invalidation* update mechanism, an invalidation message is sent to CDN servers to inform them that the content has changed. The surrogate servers are blocked from access when the content is being changed. Later, each server needs to individually fetch the new version of the content.

To ensure the consistency of CDN servers' caches, content providers usually deploy organization-specific caching policies by either specifying their own caching policies in the CDN-specific format or by employing some heuristic. In the later approach, CDN servers learn about contents changes and tune their behavior accordingly.

D. *Request routing*

The aim of CDNs is to serve the client requests from closest servers in order to reduce both network bandwidth consumption and latency. Toward this, a request-routing system composed of a request routing algorithm and a request routing mechanism are used [97]. The request routing algorithm, also called server selection, is invoked by the request routing mechanism on receiving the client request to select the edge sever that will serve it and further inform the client about the selection result. Figure 2.21 [85] illustrates the high level view of the request redirection process in the CDN environment. The interaction flow involved in this process is the following:

- The client requests a content by providing its URL;
- When the origin server receives the client request, it responds with the basic content (e.g. index page);
- To serve the embedded objects, the origin server redirects the client request to the CDN environment;

- Using the request routing algorithm, the CDN provider selects the best edge replica (using some metrics as it will be detailed further) to provide the client with the requested embedded objects;
- Finally, the selected replica provides the client with the embedded contents if available in its cache. Otherwise, it retrieves them from the origin server, serves the client and keeps them in cache for further requests.

Because the nearest edge server is not always the server that meets best the current system condition or the one that optimizes the perceived quality at the client side, a variety of metrics such as the servers load, geographical and network proximity, client perceived latency are combined to determine the most appropriate server that will serve the client request. According to the used metrics, the request-routing algorithms can be classified in two categories: the non-adaptive request-routing and the adaptive request-routing.

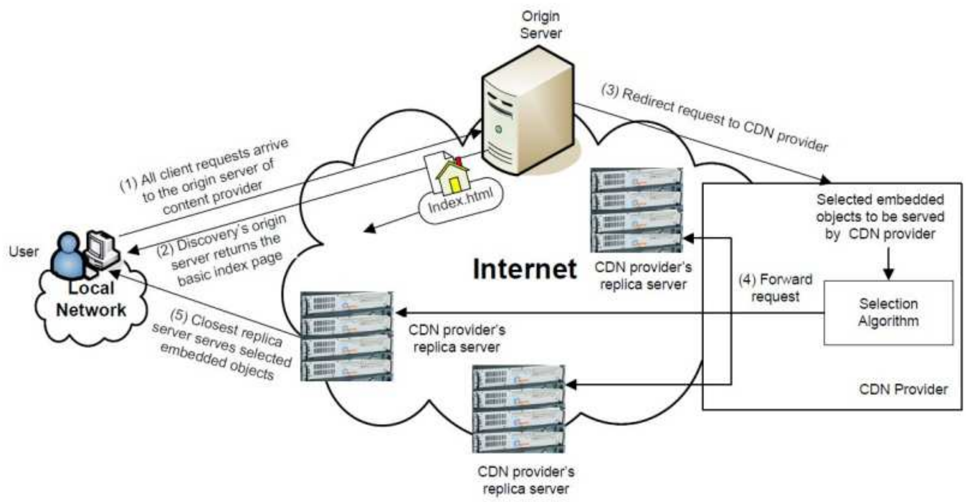

Figure 2.21. *Request-routing mechanism in CDN.*

The selection in the first category is based on some heuristics rather than the current system state. An example of such algorithm is the *Round Robin*, which distributes the requests to the CDN servers for balancing the load among them [98]. This algorithm, easy to implement, is efficient in the case of homogeneous servers located at the same place but does not perform well for distributed systems in wide areas. Indeed, requests may be directed to more distant servers and different requests can involve different costs, especially when dealing with multimedia contents. Another non-adaptive request-routing algorithm is the one proposed in [99], which is based on consistent hashing. A hashing function *h*, calculated from the servers identifiers space, based on the requested content URL is used to redirect the client requests to the server with the smallest ID from the same space, greater than *h*.

The adaptive request-routing algorithms are more complex algorithms which are able to change their behavior to cope with the current system state. These types of algorithms achieve better performance, however they require collecting and maintaining some statistics on the system state. In [100], Bartolini et al. identified three approaches to collect such information: (1) passive measurement by analyzing the traffic, (2) active probing methods such as the ICMP ECHO, and (3) feedback information that are periodically obtained from agents located at the surrogates. In [101] and [102], the anycast resolver decides which server among the replicated servers is the best one based on the server information. For the purpose of server information maintenance, four approaches are identified: (1) remote server performance probing, (2) server push, (3) probing for locally-maintained server performance, and (4) user experience.

Additionally to the request routing algorithms that select the best server, various request-routing mechanisms have been proposed in literature. A classification of such schemes according to the variety of request processing is given in [85]. These mechanisms are the following:

Global Server Load Balancing [103]: this approach consists in including Global Server Load Balancing (GSLB) in service Web switches. In the approach taken by Nortel/Alteon [104], service nodes, consisting of a GSLB-enabled Web switch and a number of real Web servers, are distributed in several locations among a wide area network. Two new capabilities extend these service nodes to allow global server load balancing: the global awareness, and the smart authoritative DNS. In local server load balancing, the Web switch in each service node is aware of the state of the real Web servers attached to it. In GSLB, the Web switches making up each service node are globally aware, each knowing the addresses of all the other service nodes. They also regularly exchange performance information among the Web switches in the GSLB configuration. This allows each switch to estimate the best server for any request, choosing not only from its pool of locally connected real servers, but the remote service nodes as well. To make use of this global awareness, the GSLB switches act as intelligent authoritative DNS servers for certain managed domains. In [105], an estimation of client proximity is included in the GSLB. Each Web switch uses the natural traffic flow between the client's browser and itself to measure the round-trip latency.

DNS-based request routing [85]-[100]: relies on a modified DNS to perform the mapping between the domain name and IP records of the pool attached to it. This approach is used in both full and partial site CDN providers. The DNS resolution is done as follows [84]: (1) when a client wants to request a content, it sends a DNS query to its local DNS server which forwards it in turn to the CDN's Request Routing Infrastructure (RRI); (2) the latter probes each surrogate server to get information about the route between it and the local DNS server; (3) the surrogate servers respond with measurements to both the local DNS server and the RRI. Some other selection criteria may be sent to the latter; (4) Based on the received measurements, the RRI selects the most appropriate server to deliver the content and sends a response to the local server DNS which forwards it to the client [84].

The DNS-based request routing presents the advantages to be simple, transparent and independent from any replicated service. However, it also presents some limitations such as scalability issues, non-resilience to failures on the targeted surrogate server and domain-level granularity. In addition, this approach does not take care of the client location. Only the location of the local DNS server is considered, which limits the ability of the request routing to determine the client's proximity to the surrogate. Furthermore, users that share the local DNS server are redirected to the same surrogate server during the TTL interval, which might lead to an overloaded situation during flash crowd.

Header inspection: this approach is based on the header inspection of the client requests. Protocols such as HTTP or RTSP allow a web server to respond to the client with a 302 response redirecting him to the delegate serving server. The client resubmits then its request to this server. The advantage of this approach is its simplicity. Its disadvantages are the lack of transparency and the significant overhead involved by the introduced extra message round-trip in the request process.

URL rewriting or Navigation hyperlink: in this scheme, the content provider directly communicates to the client the serving surrogate server by rewriting the dynamically generated pages' URL links. For example, the references to embedded objects within a main HTML page can be modified by the origin server so that the client could fetch them from the best surrogate. To automate this process, CDNs provide special scripts that transparently parse the Web pages content and replace embedded URLs.

URL rewriting can be proactive (called a priori URL rewriting) or reactive (called on-demand HTML rewriting) [106]. In the proactive URL rewriting, the content provider formulates the embedded URLs before the content is loaded in the origin server and made available to the client. In the reactive scheme, the content is modified on demand when receiving client requests. The latter scheme is then more flexible since it can take in consideration the client information such as identity, location and priority of the client. To not bind the client to a single surrogate, this approach is combined with the DNS request routing. Indeed, the DNS names contained in the embedded URLs point on a group of surrogates. Another advantage of URL rewriting is its fine granularity since the granule is the embedded object. However, the URL rewriting involves additional delay for URL parsing.

Anycasting: Anycast can be considered at two levels: at IP level and at the application level. The IP-level Anycast was introduced by Partridge et al. in [107] within a specific IPv4 class of addresses. The IP anycasting was then defined as a stateless best effort routing service able to deliver the anycast datagram to at least one of the hosts of the anycast address. An anycast IP address is then assigned to a group of servers that provide the same service. A client willing to access some services sends a datagram with the anycast address as a destination address. The sent datagram will be delivered to the "nearest" server (according to the routing protocol metric) identified by the anycast address. However, the IP-layer anycasting approach presents some limitations such as the routers necessity to allocate IP address space for anycast address. These limitations led the researchers to define the anycast paradigm at the application layer. In [101]-[108], Bhattacharjee et al. examined the definition and support of the anycasting paradigm at the application layer, providing a service that maps anycast domain names into one or more IP addresses using anycast resolvers. Application-layer anycasting appears then as a good solution for distributed Internet services provisioning, especially when it requires no modification in the existing infrastructure. Another motivation to use application-layer anycasting is the ability to manage QoS and define service requirements on a per-service basis by. Some relevant metrics that impact service quality such as server load, access delay and so on can then be evaluated and used in the selection process.

E. CDN Deployment example: Akamai

Akamai [109] is the leading content delivery service provider with more than 2700 customers. Evolved out of an MIT research effort aimed at solving the flash crowd problem, Akamai network includes over 84000 servers, deployed in 72 countries around the world. Akamai's deployments vary in size and scope, from just a few servers at small local ISPs, to hundreds of servers in high-traffic networks. Akamai's distributed network deployment places Akamai servers within one network hop of 90% of Internet users. The strong presence of Akamai is illustrated in network map of Figure 2.22 [110].

Akamai handles the flash crowd problem by means of high distribution and replication to serve end users from nearby servers. Toward this, the users request are redirected to the nearest available server using a two-level DNS system. This redirection, referred also as mapping, resolves a hostname to the IP address of the most appropriate edge server based on the service and content requested, users' location, network status and server health and load [111]. To achieve an efficient redirection, Akamai relies on an overall system monitoring. The edge servers and data centers publish load reports to the DNS servers that perform the mapping. A centralized reporting is also provided for costumer and content servers. For the monitoring of the end-to-end system performance, Akamai uses agents that simulate end users behaviour to measure the downloading times and failure rates.

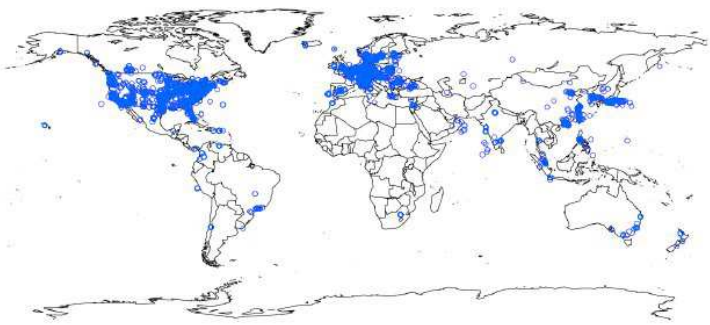

Figure 2.22. *Akamai network deployment in 2009.*

Akamai provides different services to its customers including Live and on demand streaming for Windows Media, Adobe Flash, Apple QuickTime, and real media formats, globally distributed replicated storage for digital media libraries, HTTP Downloads, rich media management, digital rights management, scalable reporting, etc. Akamai focuses on HD video as it considers the delivery cost of HD video a key differentiator for CDN providers.

To provide streaming services, Akamai technologies are combined with their proprietary solution Stream OS that simplifies rich media management and offers flexible tools to control content and enforce business policies. In 2008, 100 of the top US media companies trusted Akamai to manage and deliver their rich media assets.

The Akamai streaming delivery network is supplied with contents as follows: first, the content provider sends an encoded stream to an entry-point server of the Akamai network; then, the entry-point server sends the content to several edge servers that will serve in turn to end users. The delivery between entry-points and edge servers should be loss-resilient in order to replicate the content correctly. Further, the edge servers must deliver packets without delay and jitter toward a better playback quality. When necessary, Akamai uses information dispersal technique to let the entry point server send data on multiple redundant paths, letting the edge server to construct a clean copy of the stream when some paths are down or lossy.

Figure 2.23 illustrates how the media streaming is initiated, using Stream OS combined with the Akamai delivery network. The end user connects with the Stream OS OVP and gets a description file called BOSS. The client uses the links inside this BOSS file to get the streaming content from an edge server.

CDNs are currently the mostly used content delivery platforms. However their scalability is limited and one of the challenges in CDN research today is the design of peering CDN also called adaptive CDN which consists in the extension of the CDN platform with Peer-to-Peer networks. The next section will present the latter networks.

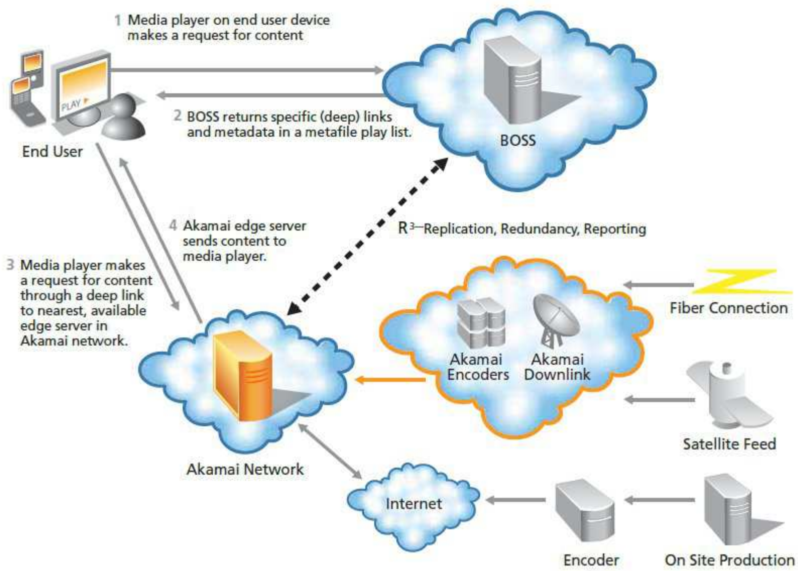

Figure 2.23. *Media delivery scheme in Akamai.*

2.3.2.2. Peer-to-peer (P2P) networks

Peer-to-Peer (P2P) overlay networks are large-scale distributed systems in nature without any hierarchical organization or centralized control. Milojicic et al. define P2P networks as a class of systems and applications that employ distributed resources to perform a function in a decentralized manner. The resources encompass computing power, data (storage and content), network bandwidth, and presence (computers, humans and other resources) [112]. The latter definition highlights the concepts of resource sharing, decentralization and self-organization. In contrast with the client/server model, P2P networks introduce a symmetry in roles by letting the client to act also as a server. Peers form a self-organizing overlay over the IP network combining several features such as redundant storage, massive scalability, fault tolerance, efficient search of contents, etc. P2P networks became popular since the emergence of file sharing systems, such as Bittorrent [113] and Napster [114]. Today, many other applications such as content distribution, Internet telephony and video conferencing are taking advantage of P2P network architecture.

The following subsection present some important aspects of P2P networks, such as the architecture of these networks, their classification according to their logical topology and the content searching used methods to locate contents.

A. Architecture

Figure 2.24 [115] illustrates the layered architecture of P2P overlay networks, which consists of the following layers: the Network Communications layer, the Overlay Nodes Management layer, the Features Management layer, the Services Specific layer and finally the application-level layer. The layers are defined in the following:

- The lowest Network communications layer describes the network characteristics of the participating nodes to the overlay. Nodes could be machines connected over the Internet.
- P2P networks are very dynamic in nature and peers benefit from a significant autonomy. Some additional functions are thus needed in order to let peers join the overlay, search for resources and retrieve them from optimal location and finally leave the overlay by announcing their departure, in the ideal case. Toward this, the Overlay nodes management layer covers the management of peers, which include discovery of peers and routing algorithms for optimization.
- The Features management layer deals with the P2P features such as fault resiliency, reliability, security and aggregated resource availability aspects of maintaining the robustness of P2P systems.
- The Services-specific layer supports the underlying P2P infrastructure and the application-specific components through scheduling of parallel and computation intensive tasks, as well as content and file management. Meta-data describes the content stored across the peers and the location information.
- The Application-level layer is the top layer of the architecture and represents the applications and services that are implemented with specific functionalities on top of the underlying P2P overlay infrastructure.

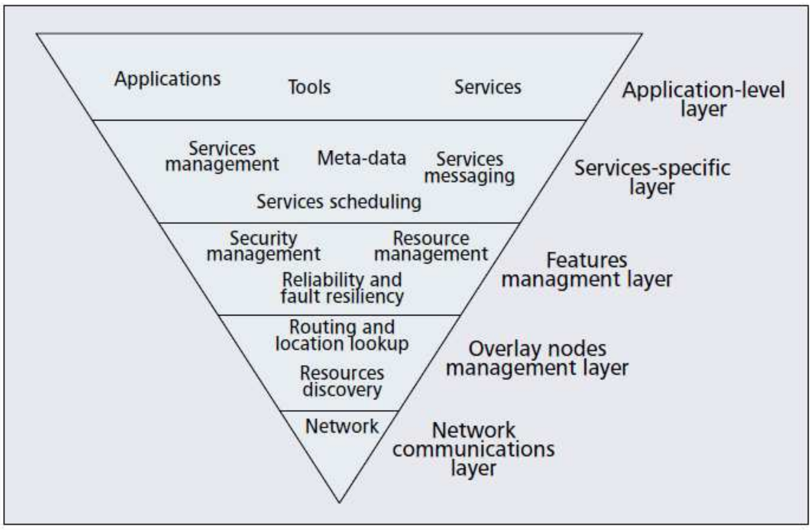

Figure 2.24. *Layered P2P overlay network architecture.*

B. Locating Content

To ensure an efficient content distribution, a P2P network should be able to locate content, scale to a useful size, and provide reliable operation. Three basic algorithms are commonly used to locate content: the *centralized directory* model, the *flooded request* model, and the *document routing* model [112].

In the ***centralized directory model,*** peers connect to a centralized directory where all peers publish content that they share with others. When the directory receives a request, it replies with information about the peer that holds the requested content. Several criteria such as network proximity, highest bandwidth connection, highest capacity, least congestion, or least recently requested content, might be used to select the best peer and achieve load balancing. The requesting peer then directly contacts the peer it has been referred to and begins the content transfer. Since this approach depends on a centralized directory, it suffers from two limits: (1) it is subject to a single point of failure and (2) it is limited to the capacity of the directory.

In the ***flooded request model***, after connecting to the network, the requesting peer broadcasts a query to its directly connected peers, which each in turn broadcasts it to their directly connected peers, and so on. This process continues until the request is answered, or some broadcast limit is reached. This approach generates a lot of ineffective network traffic and requires a lot of bandwidth, thus significantly limiting its scalability.

In the ***document routing model,*** an authoritative peer is asked for referral to get the requested content. Each peer has helpful, but only partially complete referral information. Each referral moves the requester closer to a peer that can satisfy the query. The great advantage of this approach is that the systems can reliably complete a comprehensive search in a bounded number of steps. The resulting systems can achieve scalability while still providing good performance.

C. Classification

According to P2P logical topology and the approach that is used to locate contents in the overlay, the P2P networks can be classified in three categories: the *centralized* overlays, the *decentralized but unstructured* overlays and the *decentralized and structured* overlays [116].

In the ***centralized P2P systems***, the objects index is kept in a centralized server in a form of <object-key, node address> items. Each node that joins the overlay has to publish the information about the resources that it wants to share to this server. Concerning the resource discovery, the peer just needs to retrieve from the server the addresses of other peers that provide the wanted resource. This type of architecture is simple and easy to deploy. The self-scaling property is achieved by the centralized search facility, thus it does not require much bandwidth for resource discovery. However, the central server constitutes a point of failure. The pioneer P2P file sharing system, Napster [114], is an example of such architecture.

The ***decentralized but unstructured P2P systems*** make use of unstructured overlay networks in which the search and download capabilities are distributed among peers. Peers are organized in a random graph which they join some loose rules without any prior information about its topology. Contents are discovered using a flooding-based technique. A peer requesting content sends a flood query with a limited scope across the overlay. The query is executed hop-by-hop until success / failure or time-out. If the flooding-based technique used in these networks is resilient to peers dynamicity and effective to locate highly replicated contents, it is clearly not scalable as the load on each peer makes the network size and the numbers of requests grow. Moreover, this technique is not effective in locating rare contents. Such systems are the most commonly deployed over the internet today and examples are Gnutella [117] and Freenet [118].

In the ***decentralized and structured P2P systems,*** the resource discovery is also distributed but the overlay topology and content placement are tightly controlled to make subsequent queries more efficient. These systems use Distributed Hash Table as a substrate. Keys are assigned to contents and are mapped to peers in a graph. Such structured systems allow an efficient content discovery. However, they do not support complex queries and need to store a pointer to the content at the peer that is responsible for the associated key. Content Addressable Networks (CAN) [119], Chord [120] and Tapestry [121]-[122] are examples of decentralized structured P2P networks.

P2P networks could be seen as the extreme of distribution which confer them an infinitely scaling. However the networks present also many weaknesses such as high peer churn, heterogeneous resources capabilities and that rapidly result on poor quality. We will see further in chapter 5 how these resource at the user environment could be exploited to extend the CDN server without endangering the service quality.

2.4. State of the art Conclusion

Throughout this chapter, we have presented different works that deal with the context-awareness. We have surveyed both context models that are used to describe context information and frameworks that allow running applications and services to benefit from the context-aware feature. Different representative works in both telecom and internet fields have been described.

We have also surveyed the content delivery networks that are currently used to distribute media contents over Internet. For each solution, we have presented the basic concepts on which it is built, its advantages, drawbacks and limitations. Two important conclusions can be drawn: first, these delivery solutions do not consider the context-aware feature that is essential for the expected service personalization by end users, and second, these solutions need to scale in order to provide the growing user population with the expected high quality video services while keeping low deployment and operational costs.

The following chapter will present a context-aware future media Internet architecture that, by introducing new virtual layers, allows delivery of scalable context-aware multimedia services while assuring quality at the most satisfying level for users.

Chapter 3

Architecture Proposal for Future Media Networks

3.1. Introduction

The exponential growth that experience multimedia services along with the increasing users' expectation in term of ubiquity, personalization, mobility and better experience creates stringent demands and requirements on the current Internet.

It is now a common belief that Internet has to growth in its functionality, capability and size to enable the creation and efficient distribution of novel advanced rich Media Services. Indeed, many limitations have been identified in the current Internet [3] such as facilities for large-scale service provisioning, management and deployment, facilities for network, device and service mobility, facilities to seamlessly use context information to enhance and improve existing services and deploy new ones, facilities to support Quality of Service (QoS) and Service Level Agreements (SLA), trust management and security and others.

In this context, we introduce a novel architecture proposed for Future Media Internet (within the ALICANTE - MediA Ecosystem Deployment through Ubiquitous Content-Aware Network Environments – European project) that aims to facilitate the deployment of an integrated Media Ecosystem within Future Internet, where all involved actors (Service/Content Creators and Providers, Network Operators/Providers and End-Users) collaborate for the efficient distribution of rich Media Services, both provider- and user-generated ones and addressing the aforementioned challenges.

In the following, we introduce the ALICANTE architecture proposal, its components and the functions that they implement. A comparison of our architecture to representative Generic or Media-oriented Future Internet Architecture is also given.

3.2. The proposed Future Media Internet Architecture (ALICANTE)

3.2.1. The high-level architecture overview

This section gives an overview of our architecture proposal for Future Media Internet, named ALICANTE (MediA Ecosystem Deployment through Ubiquitous Content-Aware

Network Environments). The architecture comprises a number of environments and layers aiming to improve today's media delivery networks and systems and to enable enhanced QoE and additional services for End-Users. This architectural proposal has been accepted as the main European research project in the Networked Media domain[1]. For the research work presented in this thesis, the focus is put on the User and Service Environments to enable (1) high quality services and personalization towards a better experience from the End User point of view and, (2) advanced media delivery features (including scalability, reliability, context-awareness, low cost, etc.) from the Service/Content Providers (S/CPs) perspectives.

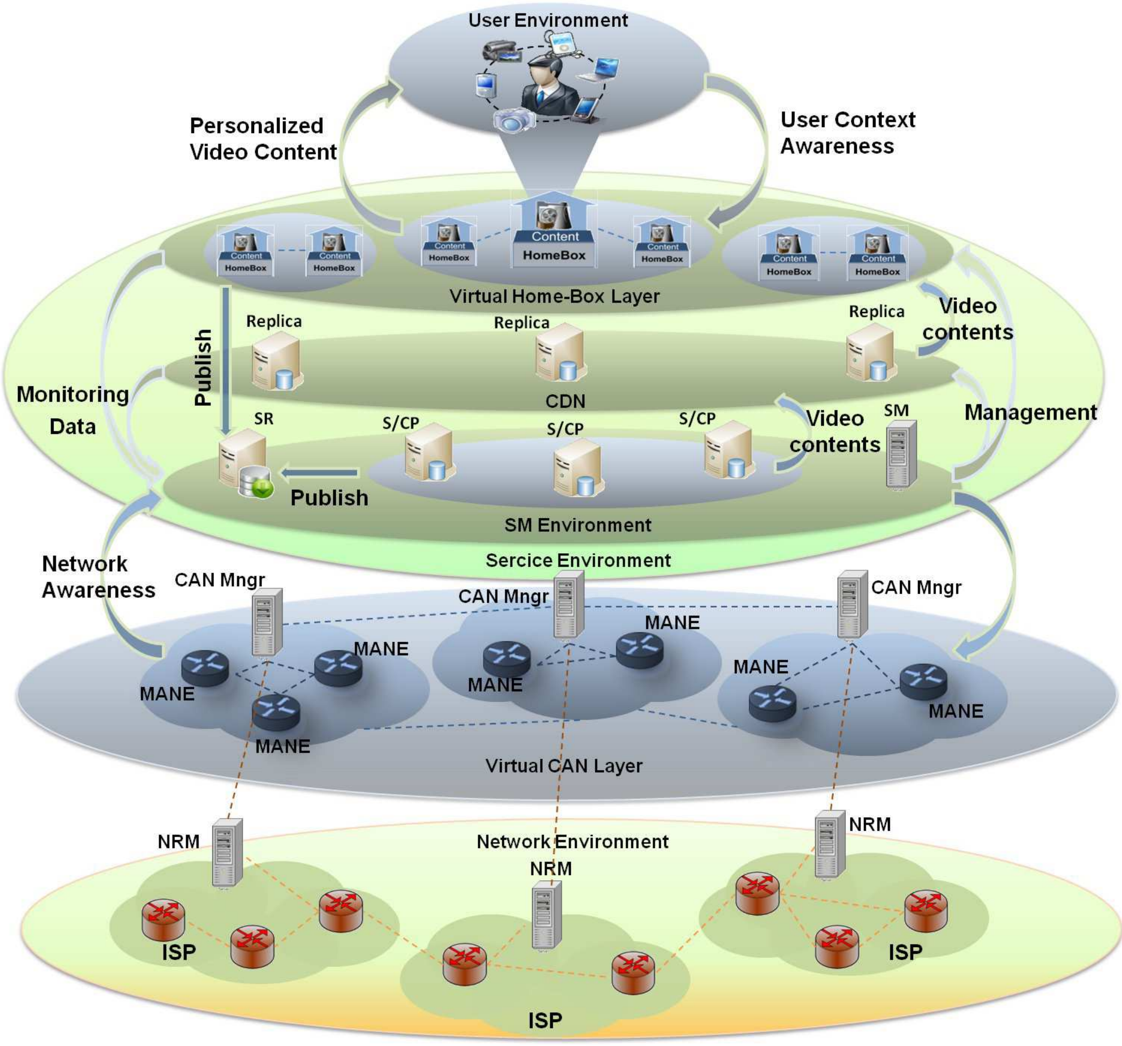

Figure 3.1. *High level Architecture overview*

Aiming to achieve a managed Media Ecosystem, ALICANTE proposes an architectural solution, illustrated in Figure 3.1, broken down to architectural layers and environments, as follows.

The **User Environment** (UE) allows the End-Users (EUs) to consume and/or generate contents and to exploit different services delivered by components of the Service Environment.

[1] ALICANTE is a funded EU FP7 research project under the Networked Media and Systems programme. It consists of 20 partners. http://www.ict-alicante.eu

Most of the UE functions are to be included in the EU terminal(s), which are connected to the user's residential gateway (Home-Box). The User Environment is the architectural part of the **ALICANTE** system dedicated to alleviate current EU's limitations. Towards this, the UE offers to the EU the potential to have several roles, such as Content and Service Consumer, Provider or Manager. This is made possible through a generic multi-platform, user-friendly graphical interface permitting the EU to access/deliver/manage any service/content on any device from anywhere and at any time. A User Profile is foreseen, characterizing the static and dynamic parameters of the user and his/her context in order to be exploited by Service Environment elements for the delivery of context-adapted services (Context Awareness). For the dynamic part, the User Profile relies on Quality of Experience and QoS monitoring.

The **Service Environment** (SE) offers enriched networked Media Services and manages their whole lifecycle (creation, provisioning, offering, delivery, control). In the management and control plane, the SE receives user context information from the UE and network information from the Network layer, to achieve a context- and network-aware service provision. In the data plane, the SE uses the overlay connectivity services of the CAN layer and is also involved in the process of adapting the services/content according to the EU context. Service composition, security and privacy are other functionalities considered at SE level. End-to-end Service Management is supported in an integrated manner, by the Service/Content Provider infrastructure, as well as evolving capabilities for combination and interoperability to existing platforms (especially standardised ones, such as the IP Multimedia Sybsystem (IMS) or Open IPTV ones). The Service Environment is the architectural part of the **ALICANTE** system dedicated to alleviate SPs limitations. Towards this, it allows providers to offer enhanced services and content through:

- A cluster of media servers (*SP/CP servers*);
- A *Service Registry* functionality that maintains metadata on all available services. It maintains updated information from all providers (including End-Users publishing user-generated content), and End-Users rely on it to discover available services and contents;
- Service Composition, Service Management, Service Monitoring and Adaptation features, supporting the entire service lifecycle.

The **Home-Box (HB) layer** is a newly proposed layer composed of virtually interconnected Home-Boxes, capable of advanced ways of service/content provisioning. The Home-Box layer is the architectural part of the **ALICANTE** system, within both User and Service Environment, dedicated to alleviate EU and SP limitations. Towards this, the *"Home-Box"* entity is a new media-centric Home Gateway, featuring advanced functionalities, such as native service provision (to other HBs), content caching, context management, service adaptation and redistribution (to the Home network and the associated User terminals), user/service mobility and security, as depicted in Figure 3.2.

The HBs are organized in an overlay offering a flexible logical infrastructure, configurable in hierarchical unicast/multicast distribution mode, in a distributed mode (P2P) or in a combination of both.

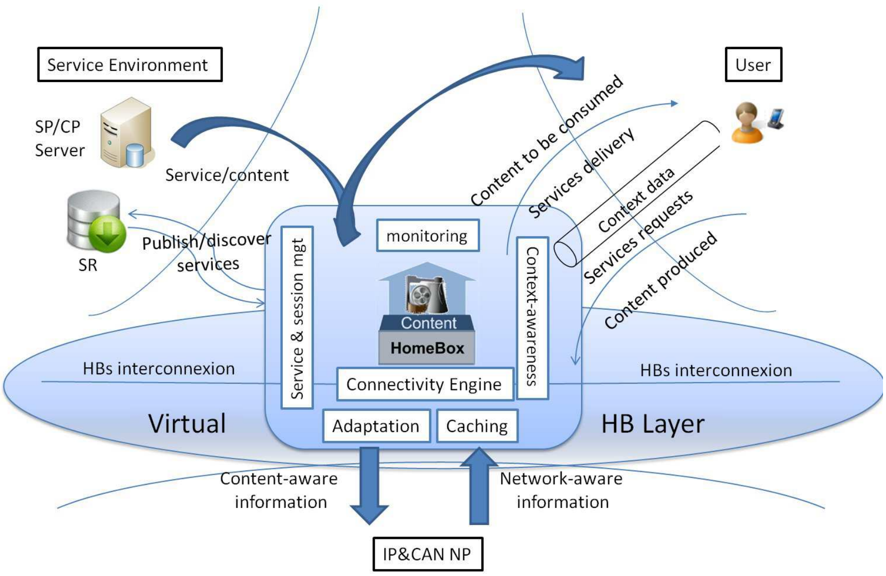

Figure 3.2. *Home-Box functional architecture.*

For the P2P mode, the idea is to leverage the participating and already deployed Home-Boxes disk caching and uploading capabilities to support the Content Delivery Network in the delivery of most popular contents, achieving thus service performance, scalability and reliability, improving the content availability and service responsiveness while saving aggregation networks' bandwidth and thus featuring low cost delivery. The proposed architecture keeps the high control of managed networks while taking advantage of the self-scaling property of P2P solutions. The HBs are managed by the SE components. The Service Manager (SM) acts then as a P2P bootstrap server and the Service Registry as a P2P tracker for the HBs peers. As it will be detailed in Chapter 5, the HB overlay also features an adaptive popularity-based caching strategy and a context-aware request redirection.

The virtual HB layer promotes distribution of both Provider- and User-Generated services among HBs in a flexible and optimised way. Towards this, the HB includes facilities that enabe End-Users to compose and publish and distribute their contents and services.

The **Network Environment (NE)** (including CAN and Network layers) provides to upper layers a rich and virtualized networked space, which can be customized and exploited by all business media sectors for delivering networked media content. The CAN & Network layers of the NE are the architectural parts of the **ALICANTE** system dedicated to alleviate NPs' limitations. Toward this, the NE includes:

- A new **Virtual CAN layer**, offering connectivity services on top of the IP infrastructure by constructing mono- or multi-domain Virtual Content-Aware Networks (VCANs). The VCANs are virtual networks offering enhanced support for packet/flow processing in network nodes. They can improve data delivery via content-aware traffic classification and appropriate processing (filtering, routing, forwarding, QoS-processing, adaptation-dynamic, aggregated or per flow, security, monitoring). The CAN layer is managed by dedicated modules (CAN Managers – CANMngr), one per network domain. The decision to have one CANMngr per network domain ensures seamless deployment of **ALICANTE** system in real environments;

- **IP network infrastructure layer**, instantiating the CANs via its Intra domain Network Resource Managers (Intra-NRM) at request of the CAN Manager. The **ALICANTE** advanced Network- and CAN-layer traffic handling capabilities are supported by enhanced/upgraded network elements: the Media-Aware Network Elements (MANEs).

Spanning all environments and layers, a **cross-layer monitoring system** supports end-to-end monitoring functions at all levels, for services and for resources especially useful for the adaptation system. **Media adaptation** is an important feature of the proposed **ALICANTE** architecture. The adaptation process itself can occur either at the Content Server side (at the SP/CP premises), inside the HB, in a network element (targeted by CAN decision) or in a combination of places. It can be launched on a unicast service but also on a multicast one, through the use of SVC. It is triggered by an Adaptation Decision Taking Framework (ADTF) based on monitoring system information and inter-actor established agreements (SLAs). The Adaptation system is the architectural part of the **ALICANTE** system clearly dedicated to improve the overall media consumption of the EU. It enables service personalization considering context-awareness features.

3.2.2. ALICANTE architecture compliancy to Future Internet standardisation efforts

The Future Internet is an emerging area where already a lot of on-going work is done. Through its Internet Architecture Task Force, FIArch has highlighted the main limitations of the current Internet architecture, which the Future Internet design should take into account [3]. A detailed up-to-date FIArch work can be found in [123]. It has started working at architectural level for the improvement of the whole Internet architecture and principles, incorporating, among others, media content aspects to which **ALICANTE** has one of the primary roles [124].

The Future Internet is expected to be a communication and delivery ecosystem. The Future Media Internet Architecture – Think Tank group (FMIA-TT) aims to specify a high-level reference model of a "Future Media Internet Architecture" which covers delivery, adaptation/enrichment and consumption of media within the Future Internet ecosystem. The objective is to reach compliancy between architectural works in this domain by relying on main concepts. Consequently, the FMIA-TT has defined [4] a high-level FMI network architecture based on three layers: *Information Overlay*: comprising intelligent nodes and servers with knowledge of the content/web-service location/caching and the network instantiation/conditions (the nodes can vary from P2P peers, secure routers or even Data Centers); *Distributed Content/Service Aware Overlay*: containing nodes which filter content and Web services; *Service/Network Provider Infrastructure*: acting as the traditional layer of services offered by the ISPs. The Users can be providers and/or consumers (*prosumers)* of the services offered by this layer. The Future Media Internet Architecture (FMIA) Reference Model is represented in the [124] ALICANTE is part of the FMIA-TT and has driven directions in the reference model.

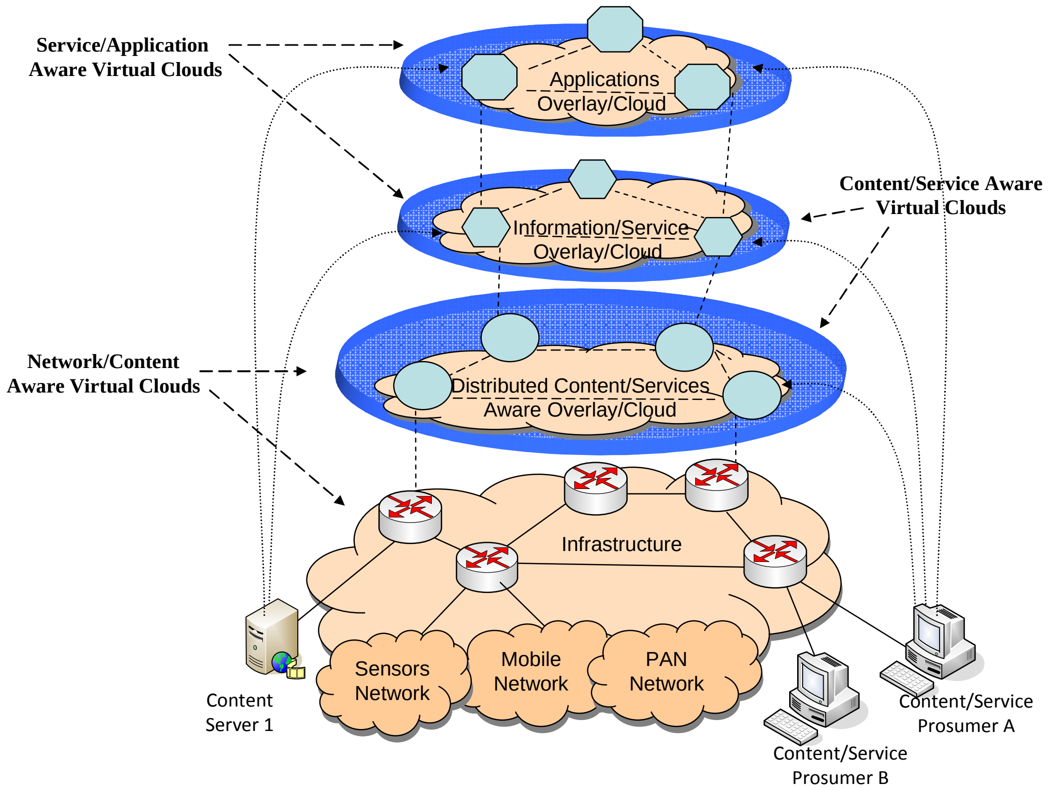

Figure 3.3. *FMIA Reference Model.*

The roles of the macro-layers defined in the FMIA architecture (or strata) are presented in the following:

The ***Service/Network Infrastructure (SNI)***:

- Executes content-aware operations in the network nodes;
- Deals with active content objects, and unstructured bit-streams whose type is known;
- Assures transport, congestion-control, policy, signalling and processing protocols in a distributed manner;
- Handles QoS and SLA;
- Enables mobility of terminals services, users and portability of content;
- Achieves virtualisation and self-organisation/self-management, as well as inter-domain connectivity and interworking, Classic (FCAPS) functionality,
- Monitors and controls network resources, security and privacy of the infrastructure;
- Assures robustness, stability, and survivability;
- Hides network complexity to the higher layers (abstracts information).

The ***Distributed Content/Services Aware Overlay (DCSAO)*** contains the logical Content Aware Network (CAN) Nodes (instantiated in core routers, edge routers, home gateways, terminal devices). CAN nodes have means to recognize and qualify the content. Part of this information may be stored locally and/or reported to the higher layer (Information Overlay).

They can filter the content and Web services flows via DPI, or identify streaming sessions and traffic via signalling. More specifically, DCSAO provides:

- Content (media and services) awareness (inspection, crawling, recognition, categorisation, indexing);
- Content (media and services) adaptation and personalisation; content caching and "findability";
- Mechanisms for content (media and services) extraction, combination, creation, orchestration;
- Name resolution, *route by name* and *route by type* capabilities;
- Mechanisms for accountability and billing of content (media and services);
- Mechanisms for protection of content (including beyond DRM, lightweight management of the content).

The ***Information Overlay (IO)*** contains intelligent nodes or servers knowing content/web-service location/caching. It has NAA-capabilities info on network instantiation/conditions. Logical node types range: ordinary P2P peers up to secure corporate routers, or even data centers in distributed carrier-grade cloud networks. The IO is aware of the content or services location/caching and the network information. However the content may be actually stored or cached at IO level or lower. IO decides on the way that content will be optimally retrieved and delivered to the subscribers or inquiring users or services. The IO provides mechanisms for:

- Controlling the content and service "findability", construction, orchestration, representation and deployment;
- Supervising where the *content or services* and the network resources information are located, cached or deployed;
- *Services*, *content*, *media* publishing or subscription, pushing accountability and billing of services, content and media;
- Supporting *networking tussle* and new business models.

The ***Applications Layer (AL***) includes:

- Applications which may use different services;
- The information delivered by the IO;
- The media/content themselves.

The **ALICANTE** architecture can be seen as a solution for the Future Media Internet [4]-[125] directly deriving from the FMIA-TT reference model. The **ALICANTE** environments are naturally mapped onto the strata presented in it.

The mapping between **ALICANTE** architectural proposal and FMIA-TT reference model strata is not exactly one-to-one since **ALICANTE** has not in its scope all functions cited above but the correlations are the ones presented in Table 3.1.

FMIA-TT	*ALICANTE*
SNI	Network Environment+Service Environment
DSCAO	CAN Layer
IO	HB Layer
AL	User Environment + Service Environment

Table 3.1. *FMIA and ALICANTE mapping.*

Therefore, **ALICANTE** architecture, as a media oriented one, has been designed inline with the most recently proposed evolutionary architectural reference models.

3.2.3. ALICANTE vs other existing/proposed Future Internet architectures

This section briefly overviews current and past research efforts, which address (parts of) the challenges for a Future Media Internet architecture.

The FP6 project **MESCAL** "Management of End-to-end Quality of Service Across the Internet at Large" project, [126]-[127] proposed an evolutionary, scalable, incrementally deployable architecture, enabling flexible deployment and delivery of inter-domain QoS across the Internet at large. The MESCAL business model actors are: Service Providers (SPs), IP Network Providers (INPs), Physical Connectivity Providers (PCPs) and Customers. MESCAL developed a generic, multi-domain, multi-service functional architecture, and a complex management system mainly focused on resource management and traffic engineering (offline and online) intra- and inter-domain. However, several MESCAL limitations exist with respect to **ALICANTE**: the End-Users cannot act as content providers; it is focused mainly on networking aspects and does not consider the service; it does not have a multimedia orientation as a main design direction. Content-aware networking, P2P mode, HBs are missing. While **ALICANTE** may use the MESCAL concepts of QoS classes (local, extended, meta-QC) in a multi-domain environment, it brings additional features mentioned above.

The FP6 project **ENTHRONE** "End-to-End QoS through Integrated Management of Content, Networks and Terminals" [128]. proposed an evolutionary complex architecture on top of IP, to cover an entire Audio/Video (A/V) service distribution chain, including content generation and protection, distribution across QoS-enabled heterogeneous networks and delivery of content at user terminals. ENTHRONE targeted primarily multimedia distribution services. Although it is focused on high-level Media Services and also some networking aspects, ENTHRONE User and Service Environment have limited capabilities in terms of complex services management, composition and mobility. Content-aware networking, P2P mode, HBs are missing. **ALICANTE** addresses such aspects.

The FP6 project **AGAVE** "A liGhtweight Approach for Viable End-to-end IP-based QoS Services" [129], tries to solve the end-to-end provisioning of QoS-aware services over multi-domain IP networks. The business model defines the Service Provider (SP) and the IP Network Provider (INP) business roles. The architecture is based on the novel concept of Network

Planes, allowing multiple INPs to provide Parallel Internets (PI) tailored to E2E service requirements. However, AGAVE lacks a complete chain of complex services management: definition, creation, offering, exploitation, composition, etc. It also does not consider home networking and content-aware aspects. **ALICANTE** will benefit from the AGAVE concepts of PIs, by offering the VCAN as enhanced equivalent of Network Planes, but in the framework of a more complete architecture, of the proposed Media Ecosystem.

The FP7 project **4WARD** "Architecture and design for the future Internet" [130], is a large research project clearly oriented towards Future Internet. It proposes new Architecture Concepts and Principles (NewACP) based on a plurality and multitude of network architectures: the best network for each task, each device, each customer, and each technology. Networks coexist and complement each other, each of them addressing individual requirements such as mobility, QoS, security, resilience, wireless transport and energy-awareness. The business players are: Physical Infrastructure Provider (PIP), Virtual Network Provider (VNP) (assembling virtual resources from one or multiple PIPs into a virtual topology), Virtual Network Operator (VNO) (installation/operation of a VNet over the virtual topology provided by the VNP for a tailored connectivity service), Service Providers (SPs) (use the virtual network as a support for their services - these can be value-added services and then SPs act as application service providers, or transport services and then SPs act as network service providers). Full network virtualisation is considered in 4WARD to solve the interoperability and is considered a main concept for a clean slate FI approach. **ALICANTE** will benefit from the virtualisation aspects investigated in 4WARD. However, it will aim at an evolutionary, backwards-compatible approach rather than a clean-slate one, in order to maximise adoption possibilities and accelerate market penetration.

The FP7 project **PSIRP** "Publish-Subscribe Internet Routing Paradigm" [131]-[132], claims to be a clean slate Future Internet approach. It aims to develop, implement and validate an Internet working architecture based on publish-subscribe paradigm, as a promising approach to solve many of the biggest challenges of the current Internet (alternative to the commonly used Send-Receive paradigm). The business model is composed of: publishers, subscribers, and a network of brokers. As a follow-on, the recent EU FP7 project **PURSUIT** "Publish-Subscribe Internet Technologies" [133], further explores and expand PSIRP's vision, targeting a more complete architecture and protocol suite, more performing and scalable. The PSIRP is revisited to produce and evaluate alternative designs, as well as to expand system dimension (hundreds of nodes), dissemination and exploitation. PURSUIT architecture maintains the PSIRP flat-label-based information identification. Every information item in PURSUIT is associated with – at least – one scope, and the information organization follows the same PSIRP principles. **ALICANTE**, although it does not fully adopt the publish-subscribe paradigm, will introduce personalized media discovery (cf. section 5.3.3.1) and distribution mechanisms beyond traditional mechanisms.

To summarize, Table 3.1 presents a main features comparison between the aims of **ALICANTE** and the projects presented above. This table does not aim at a direct competitive comparison; instead, it lists all aspects related to Media Ecosystems deployment, which are partially covered by the aforementioned projects and need to be fully fulfilled within **ALICANTE**.

Main Features	MESCAL	ENTHRONE	AGAVE	4WARD	PSIRP	PURSUIT	ALICANTE
Media (including real-time) services oriented	No	Yes	Yes	No	No	No	Yes
Full high-level services management	No	Yes	No	Yes	Yes	Yes	Yes
Multi-domain	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Network Virtualisation	No	No	Yes	Yes	No	No	Yes
CAN/NAA concepts	No	No	No	No	No	No	Yes
Network Resource Management	Yes	Yes	Yes	Yes	No	No	Yes
P2P capabilities	No	No	No	No	No	No	Yes
Home-Box concept and HB virtual layer	No	No	No	NA	NA	NA	Yes

Table 3.2. Comparison of ALICANTE with other FI oriented projects.

3.2.3.1. ALICANTE vs existing/proposed Networked Media delivery architectures

This section shortly presents architectures definitely oriented to media/content awareness, thus having a larger overlap with **ALICANTE** main stream.

The FP7 project ***C-CAST*** “Context CASTing” [71] aims to propose an evolve mobile multicasting to exploit the increasing integration of mobile in the everyday physical world and environment. The objective is to provide an end-to-end context-aware communication framework specifically for intelligent multicast-broadcast services. C-CAST is based on two main competence areas: the first one is the creation of context-awareness (detailed in section 2.2.4.6) and, the other is the multicasting technologies. These two project facets are tied together by service enablers and adaptation functions. Concerning the context-awareness feature, the C-CAST project focuses on the use of the situational context of the user in group and service management. However, **ALICANTE** is more media delivery-oriented and the focus is put on the management of dynamic user context toward ensuring to user better experience. Further, our context-awareness framework allows logic reasoning (cf. Section 4.4) since our context model uses ontology contrary to C-CAST that based its context model on ContextML (an XML-based language).

The FP7 project ***OCEAN*** “Open ContEnt Aware Networks” [134], designs a new open content delivery framework that optimizes the overall QoE to End-Users by caching content closer to the user than traditional CDNs do and by deploying network-controlled, scalable and adaptive content delivery technique. OCEAN aims to find solutions to the imminent problem of multimedia content traffic clogging up the future aggregation networks, when the offering of online video of high quality over the Open Internet continues to increase. OCEAN builds innovative self-learning caching algorithms that meet the specifics of the highly unpredictable location and time-dependent consumption patterns and dynamically adapt to the rising popularity of future delivery services. Media-aware congestion control mechanisms based on slight, but controlled quality degradation is suggested rather than blocking of user requests. **ALICANTE** goes beyond the CDN caching solutions proposed by OCEAN, by extending the CDN with the Home-Box overlay caching capabilities and designing an adaptive on-line popularity-based caching strategy that will be deployed in the HBs caches (cf. section 5.4).

The FP7 project ***P2P-Next*** [135] aims to build a next generation Peer-to-Peer (P2P) content delivery platform. The objectives of P2P-Next are the distribution of radio and television programmes, movies, music, ring tones, games, and various data applications to the general public via a variety of dedicated networks and special End-User terminals. P2P-Next bases its solution on the Bittorrent protocol to deliver SVC encoded content to the End-Users. Furthermore, P2P-Next provides adaptation at the client side via layer switching at Instantaneous Decoding Refresh (IDR) frames. The P2P delivery is one of the media delivery schemes in **ALICANTE** and collaboration with P2P-Next is planned to use their P2P connectivity engine for our HB overlay. However, in ALICANTE, the P2P overlay is formed by the HB nodes which overcome many limits of the P2P scheme (e.g. high peer churn, limited heterogeneous caching capabilities, lack of control, network unfriendliness) since the HB is a stable and managed node. In addition, we also consider the design of adaptive popularity-based video replication among the P2P overlay (cf. section 5.4). Furthermore, the ALICANTE approach not only offers adaptation (i.e., transcoding, rewriting) at the client side but also in the network by dropping SVC layers on MANEs (i.e. CAN Nodes).

The FP7 project **COMET** "COntent Mediator architecture for content-aware nETworks" [136]-[137] defines a novel content-oriented architecture that aims to provide a unified interface for content access whatever the content characteristics are: temporal nature (pre-recorded or live), physical location (centralised or distributed), interactivity requirements (elastic or real-time), or any other relevant features. It also aims: (1) to apply the most appropriate end-to-end transport strategy by mapping the content according to its requirements and user preferences to the appropriate network resources; (2) towards best quality of experience for End-Users; (3) to support unicast, anycast and multicast. The CURLING architecture [138] is proposed within the COMET project. It entails a holistic approach, supporting content publication, resolution and, content delivery and provides to both CPs and customers high flexibility in expressing their location preferences when publishing and requesting content, respectively, through *scoping* and *filtering* functions. The business entities are: Content Consumers (CC), Content Providers (CP), COMET-capable ISPs and carriers. COMET aims to be a flexible framework to accommodate several possible current or future content-related business models. It is identified that COMET and **ALICANTE** have overlapping domains. However, the COMET business model is only partially sufficient for **ALICANTE** needs; it does not consider fully the cooperation between network overlay and network resources but is focused mainly on mediation activities. Also, there is no complete chain in terms of complex services management and adaptation.

Within the Content Oriented Networking (CON) approach, a revolutionary solution, ***Content-Centric Networking (CCN)*** is proposed in [139]. The idea stems from the fact that the IP networks are increasingly used for content distribution and retrieval, while networking technology are still based of connections between hosts. CCN replaces the traditional "where" paradigm used for IP routing with "what" – identifying the content by taking the content as a primitive – decoupling location from identity, security and access, and retrieving content by name, using new approaches to content naming and name-based routing. CCN used a hierarchical naming and the unstructured routing in which the advertisement is mainly performed by flooding. Two research topics are identified for Content-Oriented Networking [140]: the multisource dissemination and the in-network caching. The last one, also treated in ALICANTE (cf. section 5), poses a real challenge in CON regarding routing scalability since popular contents are usually placed in several locations. More routing information have then to be advertized to enhance the network-wide caching performance.

3.3. Conclusion

This chapter described the ALICANTE approach for a novel architecture aiming at the deployment of Media Ecosystems within the context of the Future Internet. This approach is seen to effectively address current identified limitations and weaknesses in the current media delivery chain.

As mentioned previously, the presented contributions in this thesis focus on the upper part of the ALICANTE architecture and more specifically on the User and Service Environment, including the newly proposed HB layer. The new added value proposals for these parts are (1) a new context-aware framework (fully described in section 4) and (2) a new media delivery solution that benefit from the caching capabilities of the HB layer for scalability and consider the context-aware feature for personalization and adaptation (fully described in section 5). It should be mentionned that the proposed solutions are not dependent on a fully deployed ALICANTE architecture but have been elaborated as generic as possible in order to be possibly adopted by other NGN architectures (i.e. the popularity-based caching and the server selection algorithms are architecture- and protocol-independent). The HB design, for example, can be fully mapped with an underlying IMS subsytem. However, their evaluation have been logically made in the context of the proposed ALICANTE architecture

Chapter 4

Framework enabling Context-Awareness in Future Media Networks

4.1. Introduction

With the compelling proliferation of smart devices, the rapid growth in wireless networking technologies and the diversification of services, ubiquitous computing [141] becomes reality. The idea of ubiquitous computing is to put the software components and the networked computing devices that implement the services into the background towards focusing on providing end users with high quality service anytime, anywhere and through any device or networking technology. Indeed, from the End-User point of view, only the service value counts, not the networked device or software components that implement it.

One key aspect enabling ubiquitous services is Context-Awareness. The promise of context-awareness is to provide computing frameworks that, thanks to different sensing technologies, track the user context, understand enough on it and adapt applications behaviour accordingly to provide the End-User with contents, resources and services relevant to his current situation without explicit intervention from him. Context could be a user location, preferences and activity, device capabilities, network conditions, environment information such as time, light intensity, motion, sound noise level, etc.

The motivation for context-awareness is gaining importance with the vision of future media Internet that is foreseen to fully handle a wide range of multimedia services (VoD, IPTV, VoIP, gaming applications, and many others to come). Indeed, multimedia services are known to be very sensitive to diverse context information related to different context entities. Constraints induced by user, device, network and environment context have a strong impact on the efficiency and appropriateness of their provisioning. Multimedia services, more than others, need then to benefit from a context-aware framework that enables dynamic and automatic service personalization and content adaptation toward a better End-User experience. However, with the explosive demand experienced by Internet multimedia services and the wide context diversification inducing huge processing capabilities needed in such a framework, it is obvious that scalability is one of the most important requirements of the design of such a framework.

This chapter presents our frameworks proposals to achieve context-awareness. As mentioned in the state of the art of context modeling, (cf. Section 2.2.2.7) two models are suitable for representing context, the markup schemes-based models and ontology-based model. The first ones are the mostly used in benefits from the mature XML-based languages, technologies and tools. The second ones are more expressive and formal enabling advanced functionalities such as reasoning. However this comes with more complexity which results on less responsiveness. The latter limits make this type of models difficult to apply to multimedia frameworks that are known to be time-constrained. We have then proposed two context-aware frameworks based on these tow models. In the first contribution, the context model is based on XML and its dialects and the context management relies on XML related tools. Based then on markup scheme models, this contribution aims to provide a context-aware multimedia framework with high performance. In the second contribution, the context is modeled using ontology and rules for a more expressiveness and formality. The aim was to prove the feasibility of such models in the field of large-scale multimedia services. Since the context model support reasoning, this function is incorporated in the context management to ensure context consistency checking, high-level context inference and context-aware decisions triggering.

4.2. Context-awareness in the proposed ALICANTE architecture

As aforementioned in Chapter 3, the proposed (ALICANTE) Internet Media architecture aims to provide the context-awareness and the associated media content personalization and adaptation features on top of the Home-Box equipment so as to enable ubiquitous access to context-adaptive multimedia services without endanger the scalability and performance features. Within this thesis, we do not consider the virtual CAN layer of the architecture in our contributions, as illustrated in the functional architecture of Figure 4.1.

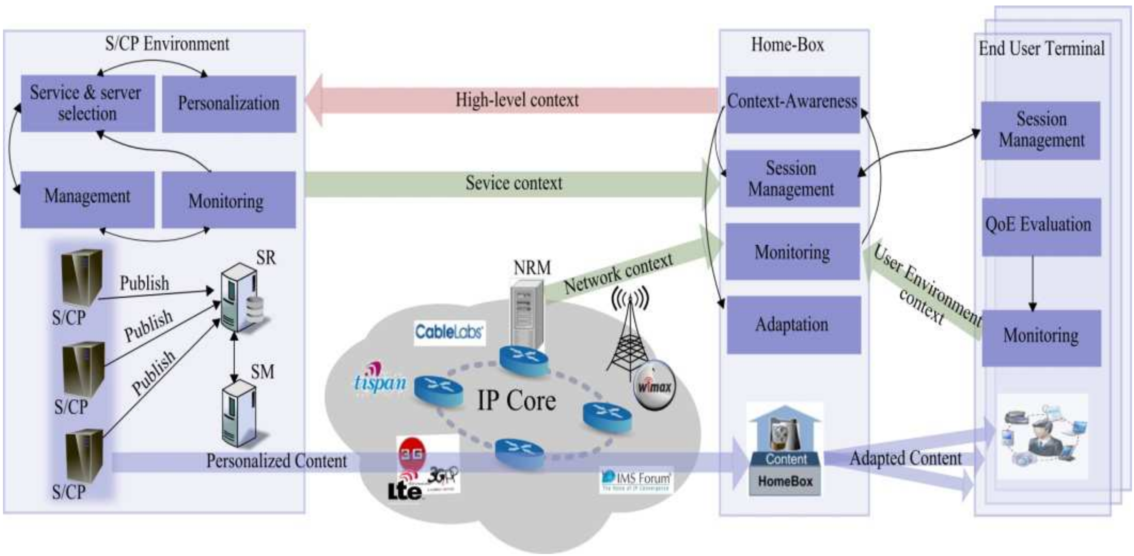

Figure 4.1. *ALICANTE Functional architecture regarding Context-Awareness.*

The HB is enhanced with the following functionalities (depicted in Figure 4.1):

Context management: the HB layer is responsible for the context-awareness feature. Context information acquired from different environments (user, service and network environments) is then abstracted to be integrated in a formal (markup or ontology-based) model permitting it to be shared between different users, devices and services. As it will be detailed in the next sections, situational context is also deduced or inferred depending on the used model to dynamically supply services with accurate high-level context information, when needed. The distribution of context management among HBs has several advantages:

- First, it relieves the resource-constrained devices for the task of context reasoning;
- Scalability can be achieved by distributing the context management features among Home-Boxes. Especially in the second contribution in which the context ontology ABox (the context information instances) and the reasoning task are distributed among the HBs (the number of users that can be managed by a HB is set limited);
- The Home-Box is the central element in Home Network, so besides managing context for Internet services, it can also manage the home appliances and services;
- Being an intermediate equipment that can be easily reached from any of the User, Service and Network Environments (identified in the architecture) and implementing itself the major context consumers processes (content personalization, adaptation and service and session management), the HB constitutes an ideal place to which the monitoring information can converge while minimizing communication overhead.

Service personalization and content adaptation*:* to overcome the high context heterogeneity, adaptation seems to be an effective solution. Multimedia contents should be adapted to always meet the computing device capabilities, the access network conditions or the supported format by the displaying applications ensuring thus a satisfying service quality. Some adaptation actions, such as the selection of the right content format when it is available, can be achieved at the Service/Content Provider S/CP side. However, adaptation actions such as resizing and transcoding can't be performed neither in a centralized way (at the CP side) nor at the end-user terminal side due to the huge processing capabilities that they require. Such actions are performed at the HB equipment that will act as an intermediate node for its associated User Environment. Another trend goes in the direction of multimedia service personalization towards providing end-users with appropriate and enriched contents that fit their preferences and interests. In the proposed framework, this feature is achieved in collaboration between the HB that sets the parameters of the discovery request based on the user profile and the Service registry SR that performs the matching of the request with the service' descriptions previously published by Service Providers (SPs). The decisions taken within these processes could be either invoked (Markup schemes) or directly derived from context by means of reasoning or a result of some algorithms whose execution will be triggered by events sent by the context-aware feature (Ontology-based schemes).

Session Management*:* Since the objective is to abstract as much as possible the dynamicity of the context-based adaptive behaviour of the system to the Service Environment for more scalability, the HB is acting as a proxy for its related devices for all the signalling process. The User Environment context-driven adaptive management actions are hence performed locally. An example of such actions can be the achievement of multiple device deployment by enabling a seamless handover of the session from one device to another when the user moves.

To ensure high service quality and keep the system under control, ***monitoring*** software runs in all the system nodes to collect computational and connectivity context. The reporting can be periodic or on-demand according to the system needs. The monitoring software is also coupled with a subjective quality evaluator tool [142] at the end-user device side that dynamically evaluates the quality perceived by the user.

The following sub-sections present the two proposed frameworks to achieve context awareness. We first present the markup scheme-based framework and, second, the one using ontology and rules.

4.3. Markup scheme-based context framework

This contribution is built on the Markup scheme-based languages and tools [143]- [124]. The aim is to propose a context-aware framework with high responsiveness and performance in order to ensure the real-time requirements of multimedia services. In the following, we present the proposed context model (User Profile model), its management and the context-awareness middleware implemented and deployed in the Home-Box equipments.

4.3.1. Markup scheme-based context model – The User Profile

The first step of our work consisted in identifying the pertinent context information that will constitute the overall Markup scheme-based model, named User Profile. For this, we have considered the wide adopted definition of a context given by Dey [16] that defines context as *any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves*. We thus started by identifying the entities considered to form the necessary parts of the context. Four entities have been identified as relevant to the user profile design, namely the *User, Device, Service* and *Context* entities. Each entity is represented as a sub-profile. The first three sub-profiles represent static data while the last one considers dynamic information.

Since we want context to be shared in the User Environment, each Home-Box will only maintain one instance of User Profile, which includes all users, devices, services and dynamic information within the Home Network.

Since much work have been done in XML-based context modeling, we have reused some parts of profiles that we consider relevant to our modeling. Table 4.1 enumerate the used standard with the associated prefix/namespaces that we have included in our profile.

Prefix	Namespace
xmlns:mpeg7	urn:mpeg:mpeg7:schema:2004
xmlns:dia	urn:mpeg:mpeg21:2003:01-DIA-NS
xmlns:upnpd	urn:schemas-upnp-org:device-1-0
xmlns:**ALICANTE**	http://www.ict-**ALICANTE**.eu/schema/2011/userprofile

Table 4.1. *Standardized profiles Prefix/Namespaces included in the User Profile.*

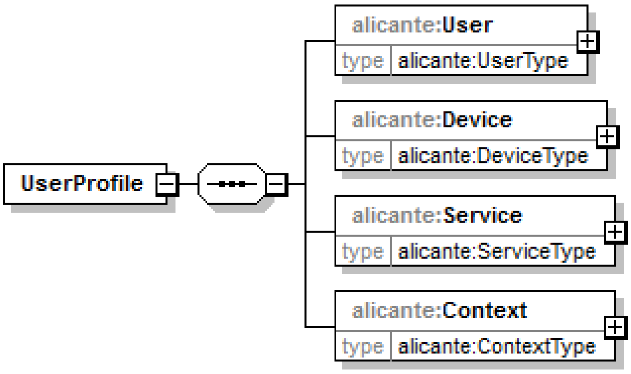

Figure 4.2. *High level User Profile structure.*

The schema of each sub-profile is detailed in the following sub-sections. For clarity, only the graphic representation (generated with XMLSpy) of the User Profile Schema is shown.

4.3.1.1. User Sub-Profile

This sub-profile stores the basic identity information of home users (name, age, gender, address, phone number, profession, etc.) and the user preferences. The User Sub-Profile is composed of one or more "User" elements, whose structure is detailed in Table 4.2. In summary, each "User" element contains the unique ID of the user, his general information as described in a mpeg7:PersonType element, his usage activity and his static configuration, such as preferences and favorite items.

The definition of the User element relies on some standardized element types (e.g. PersonType, SemanticBasicType) depicted in Table 4.2.

Element Name	Cardinality	Element Type	Comments
UserID *	1	ID	Unique Identifier for the user (*: Attribute)
General	1	mpeg7:PersonType	Including User's name, affiliation, citizenship,
			Address, Electronic Address
Activity	0…∞	mpeg7:SemanticBasicType	Usage history
Preference	0…∞	mpeg7:UserPreferenceType	User preferences
Favorite	0…∞	mpeg7:SemanticBasicType	List of favourite items: books, movies, web pages, etc.
GUIPreference	0…1	GUIPreferenceType	List of favourite graphical appearance details: font, colour, background, etc.

Table 4.2. *UserType Schema for User Sub-Profile.*

The device sub-profile (Table 4.3) will be based on MPEG-21 DIA UED and UPnP Device Descriptions. UPnP device schema is used to describe some hardware information such as serial number and model description of the terminal devices. MPEG-21 DIA provides tools to describe terminal capability, including encoding and decoding capabilities, device characteristics such as power, storage, data I/O, display and audio output capabilities.

Element Name	Card	Element Type	Comments
DeviceID*	1	ID	Unique ID of the terminal device (*: Attribute)
DeviceDescription	1	upnpd: deviceType	Device Type defined by UPnP, including manufacturer, model and associated services.
NetworkInterface	0…∞	NetworkInterfaceType	This type is extended from dia:NetworkType, it describes the network interface(s) of the device: MAC/IP address, interface class (wired/wireless), max/min capacity, error correction capability.
Terminal	0…1	dia: TerminalType	The elements contains capability information on the terminal device: different capability definitions in dia or mpeg7can be used
Terminal Capability Definitions that can be contained by Terminal element (dia:TerminalType)			
SupportedFormat	0…∞	mpeg7:MediaFormatType	The media formats supported by the device, including information such as file format, coding format, file size, bit rate, etc.
DisplayCapability	0…1	dia:DisplayCapabilityType	mode, screen size, rendering format, etc.
AudioOutputCapability	0…1	dia:AudioOutputCapabilityType	Frequency, SNR, power, number of channels, mode, etc.
PowerCharacteristics	0…1	dia:PowerCharacteristicsType	Average power consumption, battery mode, battery remaining capacity and time.
StorageCharacteristics	0…1	dia:StorageCharacteristicsType	I/O transfer rate, size, writable or not.

Table 4.3. *DeviceType for Device Sub-Profile.*

Here is a concrete example of a Terminal element with different capability definition:

```
<ALICANTE:UserProfile xmlns:dia="urn:mpeg:mpeg21:2003:01-DIA-NS"
xsi:schemaLocation="http://www.ict-ALICANTE.eu/schema/2011/userprofile
http://www.ict-ALICANTE.eu/schema/2011/userprofile"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:ALICANTE="http://www.ict-ALICANTE.eu/schema/2011/userprofile"
xmlns:mpeg7="urn:mpeg:mpeg7:schema:2001">
   <ALICANTE:Device>
     <ALICANTE:Terminal>
       <dia:TerminalCapability xsi:type="dia:DisplaysType">
         <dia:Display>
           <dia:DisplayCapability xsi:type="dia:DisplayCapabilityType">
             <dia:Mode>
               <dia:Resolution horizontal="1920" vertical="1080"/>
             </dia:Mode>
             <dia:Mode>
               <dia:Resolution horizontal="800" vertical="600"/>
             </dia:Mode>
             <dia:ScreenSize horizontal="1920" vertical="1080"/>
```

```
            </dia:DisplayCapability>
          </dia:Display>
        </dia:TerminalCapability>
        <dia:TerminalCapability xsi:type="dia:CodecCapabilitiesType">
          <dia:Decoding xsi:type="dia:VideoCapabilitiesType">
            <dia:Format
href="urn:mpeg:mpeg7:cs:VisualCodingFormatCS:2001:3.1.2">
              <mpeg7:Name xml:lang="en">
                                          MPEG-4 Visual Simple Profile @ Level 1
                              </mpeg7:Name>
            </dia:Format>
          </dia:Decoding>
        </dia:TerminalCapability>
      </ALICANTE:Terminal>
    </ALICANTE:Device>
</ALICANTE:UserProfile>
```

4.3.1.2. Service Sub-Profile

The service sub-profile (Table 4.4) is used to record the information about services which the EU subscribed to (name, version, related protocols and ports, etc.) and the contents that they manipulate (multimedia contents, databases, files, etc.). It also contains information that can serve in the publishing, discovering, presenting and billing phases.

We import MPEG-7 schema and MPEG-21 DIA UED to describe service-sub profile. *mpeg7*:*ControlledTermUseType* and the corresponding Classification Scheme are used to define *ServiceType* and *ServiceLevel*. They provide a mechanism for defining vocabularies for specific domains in an interoperable and extensible way. A Classification Scheme defines a set of terms, which are referenced in the controlled terms. Controlled terms are represented as elements of type *mpeg7*:*ControlledTermUseType*, a Classification Scheme is represented as *mpeg7*:*ClassificationScheme*. The *mpeg7*:*ClassificationScheme* specifies a domain, to which it is applicable, and it associates a URI with each term it defines. The URI is then used by the *mpeg7:ControlledTermUseType* to reference that term. *mpeg7:CreationInformationType* is used to define *ServiceDescription* and *ContentMetaData*. dia:*LimitConstraintType* is used to define the child elements of *ServiceRequirements*.

Element Name	Cardinality	Element Type	Comments
ServiceId	1	anyURI	Unique identifier for the service
ServiceInformation	0..∞	ServiceInformationType	Complex type including information about the service (name, type, level discovery location, version, period of validity and description)
ContentMetadata	0..1	mpeg7:CreationInformationType	Complex type for describing the service and its associated content
ServiceRequirements	0..1	ServiceRequirementsType	Complex type including information about minimum network and EU display requirement for the service. Other requirements specific for the service can be added.

Table 4.4. *Service Sub-profile schema.*

Figure 4.3 presents the *ServiceType* elements and Figure 4.4 details the structure of the *ServiceInformationType*.

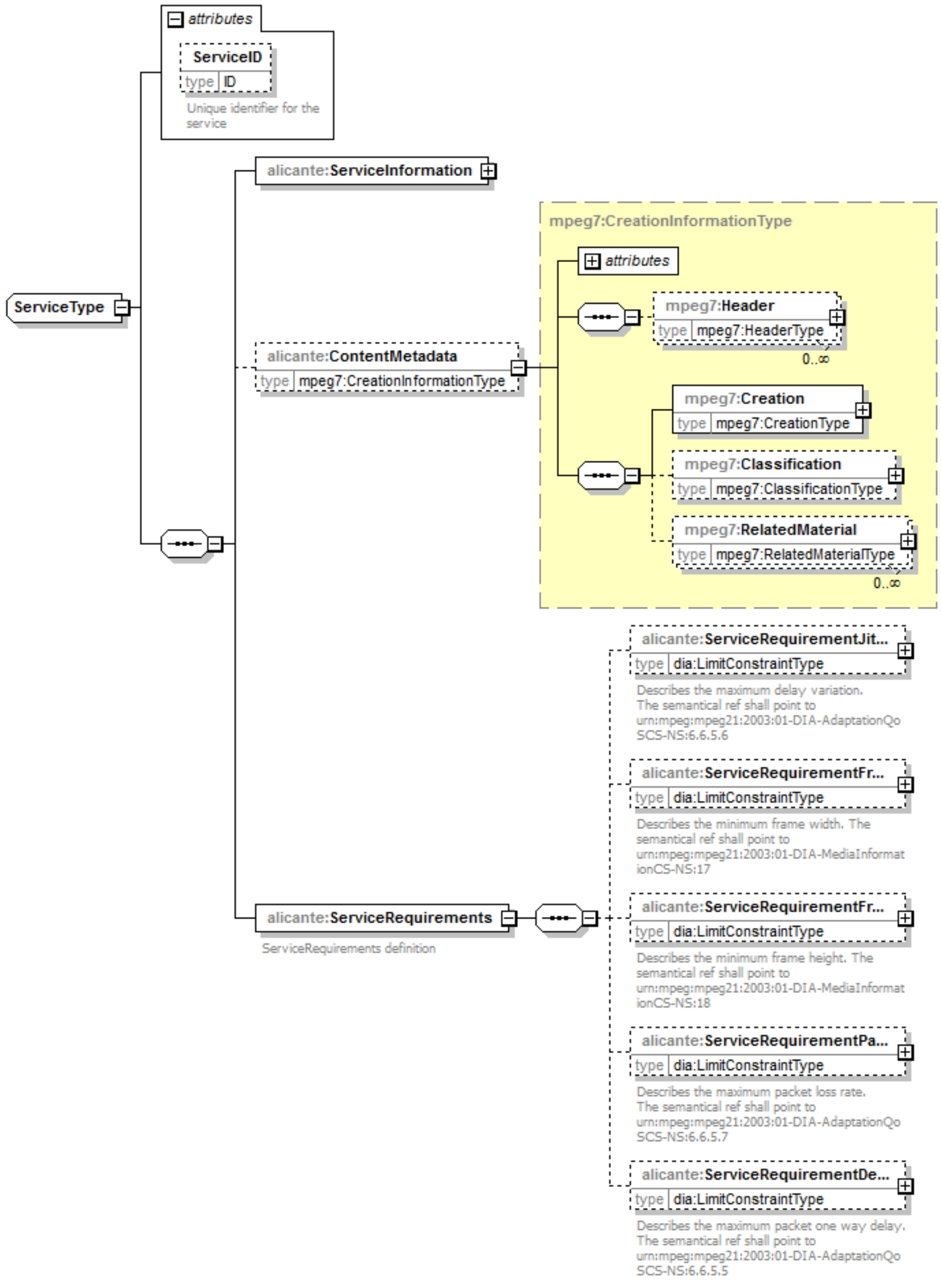

Figure 4.3. *serviceType structure.*

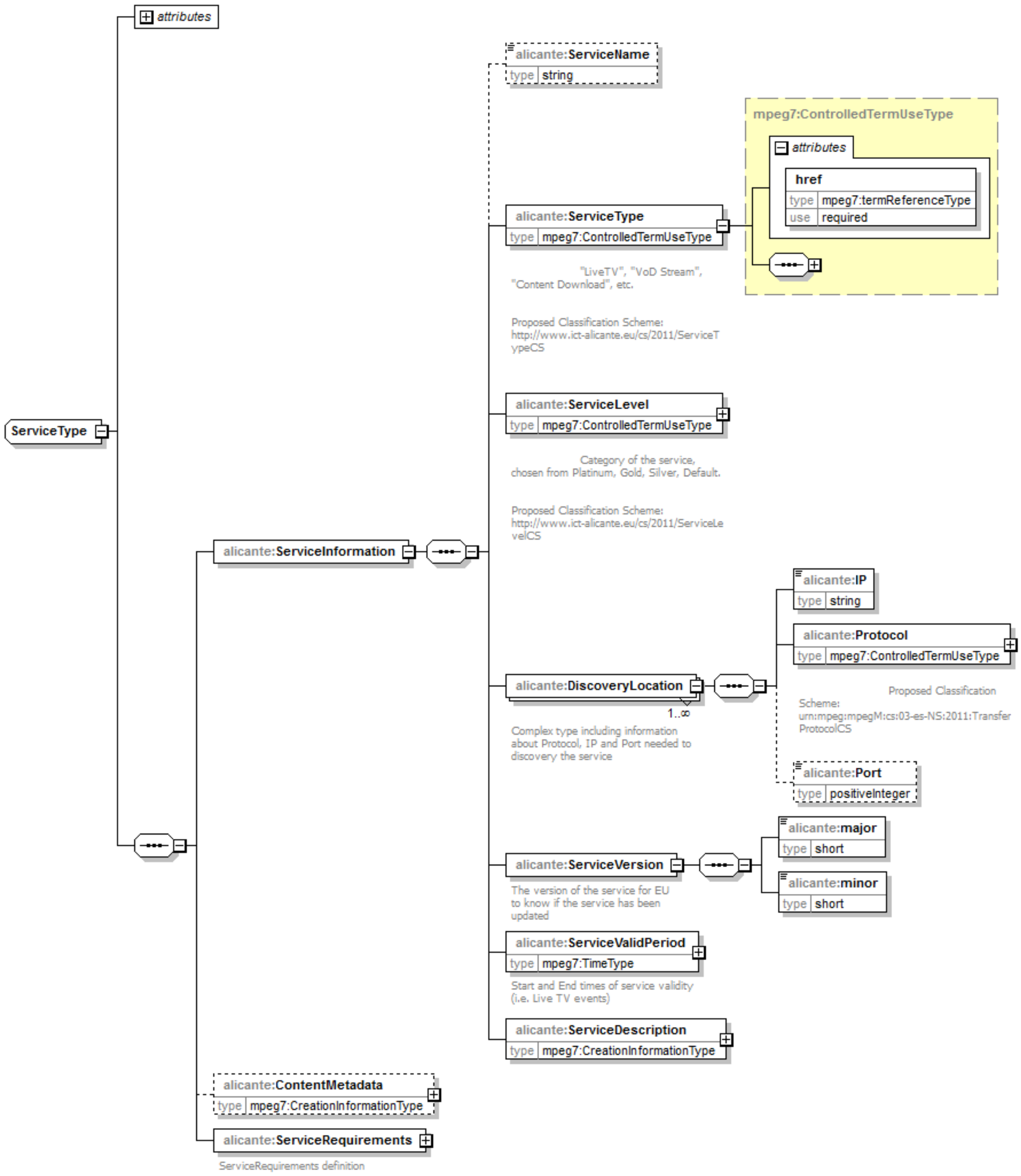

Figure 4.4. *ServiceInformationType structure.*

4.3.1.3. Context Sub-Profile

The context sub-profile is the dynamic part of the User Profile, which is mainly composed of volatile data that characterize the current User Environment (time, date and indoor/ outdoor location, running applications, etc.), the serving devices (CPU charging, battery level, etc.) and the actual serving networks conditions (available bandwidth, loss and error rate, etc.). The data contained in this part of the profile are collected by monitoring functions embedded at the user terminals.

As presented in Figure 4.5, the context sub-profile is modeled in element Context within a complex type ContextType which is composed of two elements:

- The Session element that references four elements by IDREF attributes: the *UserRef*, the *DeviceRef*, the *ServiceRef* and the *NetworkRef*. These referenced elements are the entities that the session implies (the user that initiates the session, the requested service, the devices and network trough which the service is accessed) ;
- The generic element *ContextParameter* within a complex type *ContextParameterType* is characterized by Name, Value and Category. A *ContextParameter* is associated to a given session directly by referencing it by the IDREF attribute *SessionRef* or indirectly by referencing an element already referenced by the session. Indeed, context parameters can be common to more than one session. For example, several sessions can share the same network, so the parameter *NetworkInterfaceUtilisation* is associated with a given network but it is common to all the sessions that share this network.

All the monitored QoS/QoE parameters are modeled as *ContextParameter* elements. For example, session-related parameters will be modeled as session data rate; QoS parameters as packet loss; Terminal context as CPU load, battery level, etc. Since the Context sub-profile is composed of monitored data, each *ContextParameter* is valid for a period of time and its expiration is specified in the attribute *expiresAt*.

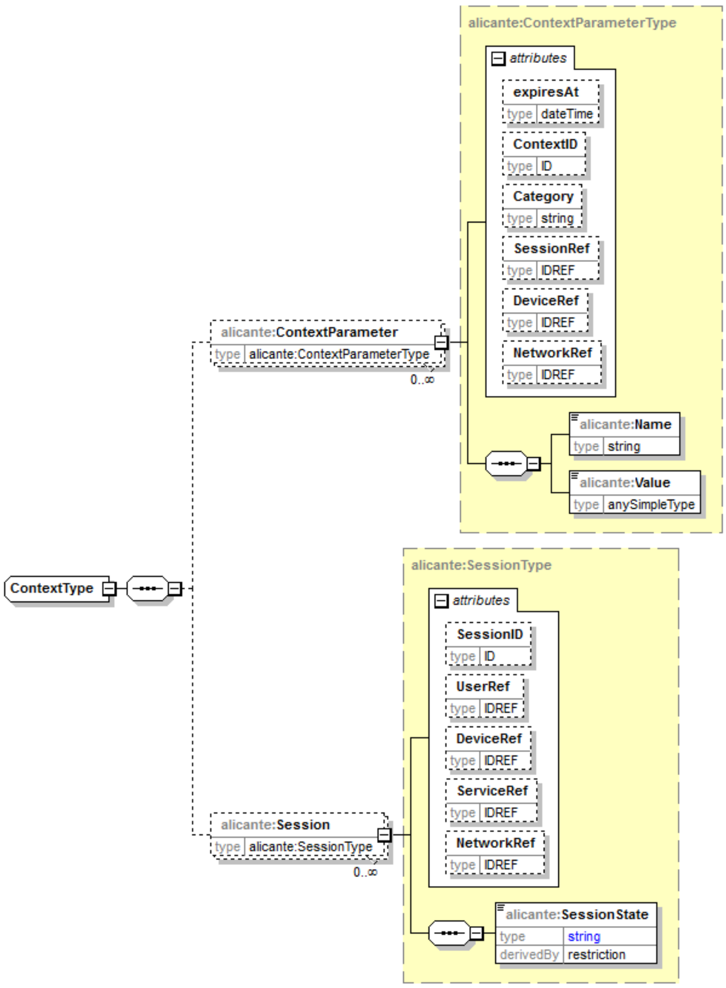

Figure 4.5. *ContextType structure.*

4.3.2. Context-Awareness middleware overview

Based on the context model and in collaboration with the other service management and adaptation components, the context-awareness middleware ensures a ubiquitous access to services for the related home users. As illustrated in Figure 4.6, the middleware consists of an intermediate entity between the context providers and the context consumers. Indeed, raw context data is first acquired from multiple sources located at the different User, Service and Network Environments, then integrated to the context model on which new high level context is deduced thanks to intelligent queries and finally provided in the right format to the Service Environment that will take decisions on it and perform some actions to adapt services and contents to the user context. The functional architecture of the middleware is described in the following.

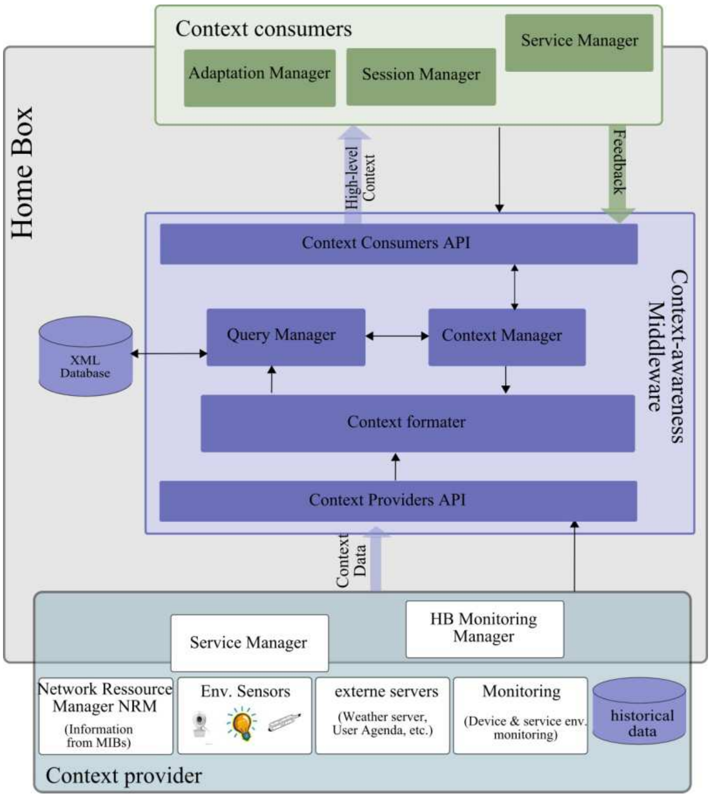

Figure 4.6. *The context management middleware architecture.*

The middleware presents two types of ***APIs*** for both Context Providers and Context Consumers. A new Context Provider can hence easily subscribe to the middleware to provide it with the context information. Context Consumers can also subscribe to be notified when some context information change.

The ***Context Formatter*** is in charge of the translation of context information from the format given by the context provider to the format defined in the profile schema.

Query Management. In order to enable context awareness, each service should be able to easily access the profile data. This access is managed by the Query Management component. The Query Management maintains XQuery models that express the context information to which context consumers have subscribed. It is also responsible for the update of the profile stored in the XML database.

The ***XML Database*** is responsible for the storage of the profile information. Moreover, it offers to other modules the possibility to track important profile parameters using XQuery triggers and to propagate the important events to other components. Both relational database and native XML database can be used. However, since the native XML databases provide better XML support and flexibility [145], we have opted for this choice.

The ***Context Manager*** is the central component of the middleware. It is in charge of maintaining all the information concerning the Context Providers and Context Consumers. For Context Consumers, it maintains the context information for which they have subscribed, the format (data structure and type) and the mode (periodic/on-demand) in which the latter should be sent and the frequency of context provisioning, if this process is done periodically. In parallel, it maintains, for the context providers, the set of information that they provide, the format (if it does not correspond to the profile schema), the frequency and the related access information.

4.3.3. Middleware Implementation

The context management middleware is implemented in Python. The profile is stored in the native XML database Sedna XML [146] and is accessed through XQuery queries [147]. Sedna XML database is an open source implementation providing a full range of database services for XML native data. Basic operations such as XQuery-based request, update and triggers configuration are supported by the Sedna API.

Concerning triggers configuration, the database trigger is a procedural code that is automatically executed in response to some predefined conditions. It is mostly used to log events, prevent changes and keep the database integrity. Sedna supports XQuery trigger, with the following syntax:

```
CREATE TRIGGER trigger-name
( BEFORE | AFTER ) (INSERT | DELETE | REPLACE)
ON path
( FOR EACH NODE | FOR EACH STATEMENT )
DO {
  Update-statement ($NEW, $OLD, $WHERE);
            . . .
  Update-statement ($NEW, $OLD, $WHERE);
  XQuery-statement ($NEW, $OLD, $WHERE);
}
```

The XQuery trigger can be used to monitor values of QoE/QoS parameters. The following trigger example raises an error when the MOSScore value is lower than 3:

```
CREATE TRIGGER "mos-alarm"
AFTER REPLACE
ON doc("userprofile.xml")/UserProfile/Context/ContextParameter/
FOR EACH NODE
DO {
 if(($NEW/name = "MOSScore") and ($NEW/value < 3))
 then error("Low MOS value detected.");
}
```

The communication protocol between the middleware and the other component is SOAP/XML. The soaplib v is used. Python 2.7 web services interfaces are integrated with the sedna dababase, and soaplib 0.8.1 is used as SOAP server, and suds 0.4, as soap client.

The following SOAP message illustrates an example of message sent by QoE Monitor located at the user terminal to the middleware:

```
<?xml version="1.0"?>
<soap:Envelope
xmlns:soap="http://www.w3.org/2003/05/soap-envelope">

<soap:Body xmlns:m="http://www.ict-ALICANTE.eu/schema/2011/UPMngr">
     <m:UpdateUserProfile>
             <SessionID>uuid</SessionID>
             <UserProfileParameters>
                     <!-- Parameters to be updated -->
             </UserProfileParameters>
     </m:UpdateUserProfile>
</soap:Body>
</soap:Envelope>
```

The SOAP message is received by the middleware that updates the profile in the database through XQuery Update query. The following XQuery Update query updates the context parameters for a given session.

```
UPDATE
replace
doc("userprofile.xml")/Context/ContextParameter[SessionID=uuid]
with NEW_CONTEXT_PARAMETERS
```

This contribution provides then a light weight XML-based context-aware framework for multimedia services. However, the use of more formal models such as ontology and rules can enrich significantly the context expressiveness and the framework flexibility, extensibility and interoperability. In the following contribution a context-aware framework using ontology is presented. With this contribution we have proven that onlologies could be applied in the field of multimedia services but with efficient distribution.

4.4. Ontology-based context framework

In this contribution [199], we propose (1) an ontology-based model characterizing a variety of context information with explicit semantic representation and (2) a large-scale framework that enables context-aware Internet multimedia services, in which the context management and adaptation features are distributed among Home-Boxes. For this, we introduce a reasoning-based middleware that is designed to support different tasks involved in the context management feature. The middleware acquires context from different sources, integrates it to the context knowledge database, reasons on it to derive situational context needed to manage services and, finally, supplies the Service Environment with accurate context information needed in the discovery, invocation and adaptation features.

4.4.1. Ontology-based Context Model

As a formal representation of context is a condition to process it by software agents, a well designed context model is the fundament of any context-aware system. In the following, we first introduce the languages that we have used in context modeling and then present the proposed context model for Internet multimedia services.

4.4.1.1. Motivations for the use of ontology in context modelling

Developed first in Artificial Intelligence to facilitate knowledge sharing and reuse, ontology, as mentioned in the state of the art (cf. Section 2.2.2.6), may play a major role in context-awareness systems. An often cited definition of ontology by Tom Gruber [46] defines it as *a formal explicit specification of a shared conceptualization of a domain of interest*. Ontology provides means that allow a formal description of the semantic of context information in terms of concepts and roles. Context could then be encoded in such a way that software agents should not only process but also understand and draw new conclusions on the user situation through reasoning, as humans could do. For all these reasons, we have studied the possibility to base our context model on ontology.

Prominent ontology languages are the standards RDF/RDF Schema [33], OWL (Ontology Web Language) [49]. Since OWL is much more expressive than RDF or RDFS, is designed as a standard and has the support of a well known and regarded standard organization (W3C), we decided to rely our context model on it. For more interoperability, the service-related context is also modeled using OWL-S [148] and the rules are modeled in DL-safe SWRL [149]. These languages are introduced in the following sub-sections.

A. Ontology Web Language

Ontology Web Language (OWL) [49] is a W3C recommendation ontology language for the Semantic Web. An OWL ontology may include description of classes, instances of classes and relationships between these instances. The basic concept in a domain is a class that is represented by *owl*:*Class*. It represents a group of individuals that belong together because they share a number of properties. Classes can be organized in a specialization hierarchy using *subClassOf*.

Properties are binary relations that are used to asset facts about either a member of classes or a specific individual. Properties are classified into two types: *dataTypeProperties* that are used to describe members of a class and *objectProporties* for describing relations between members of two classes. To enable reasoning, it is possible to set characteristics on properties. A property can be:

- Transitive: P(x, y) and P(y, z) implies P(x, z) ;
- Symetric: P(x, y) if P(y, x) ;
- Functional: P(x, y) and P(x, z) implies y = z ;
- Inverse: P1(x, y) if P2(y, x) ;
- Inverse functional: P(y, x) and P(z, x) implies y = z ;

Restrictions can also be set on how the properties are used or on cardinalities.

OWL provides also means for both constructing new classes on existing classes such as *intersectionOf*, *unionOf*, *complementOf*, *oneOf* and *disjointWith* and for expressing either equivalence on classes *equivalentClass*, properties *equivalentProperty* and individuals *sameAs* or differences between individuals *differentFrom, AllDifferent*.

OWL comes in several variants that are OWL-Full, OWL-DL, and OWL-Lite. Each variant corresponds to a DL of different expressivity and complexity.

There are many implemented OWL reasoners available for both OWL-DL and OWL-Lite that cover all major DL inferencing tasks. Each of these reasoners provides access to their functionality either through proprietary APIs that conform to their implementation language, or support the standard DIG interface [150] for handling DL elements in an XML format. OWL reasoners include FaCT++ [151] and Pellet for OWL-DL [152] as part of the OWL-API, Racer/RacerPro [RACER] for OWL-Lite and OWL-DL with approximations for nominals. Another prominent framework for reasoning with OWL are KAON1 and KAON2 [KAON] with extension to the decidable DL-safe fragment of SWRL.

B. OWL-S

OWL-S is an upper ontology used to describe the semantics of services based on OWL and is grounded in WSDL. OWL-S is expected to enable Automatic Web service discovery, invocation, composition and interoperation. OWL-S is built on top of OWL and consists of three main upper ontologies: the Profile, Process Model, and Grounding, as illustrated in Figure 4.7. These ontologies are presented briefly in the following:

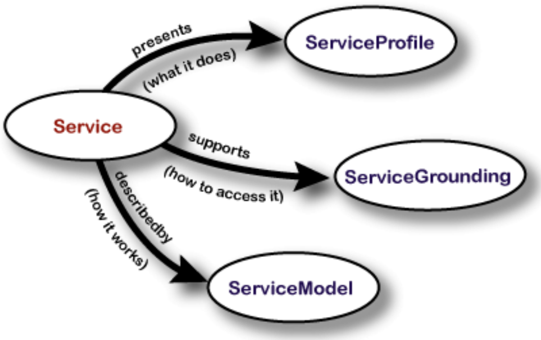

Figure 4.7. *The top level OWL-S Ontology*

The ***Service Profile*** provides a concise description of the services that may be used for service discovery purpose. An OWL-S service profile encompasses functional parameters, i.e. Input and Output, and precondition and effect (IOPEs), as well as non-functional parameters such as *serviceName*, serviceCategory, *textDescription*, and metadata about the different actors (cf. Figure 4.8).

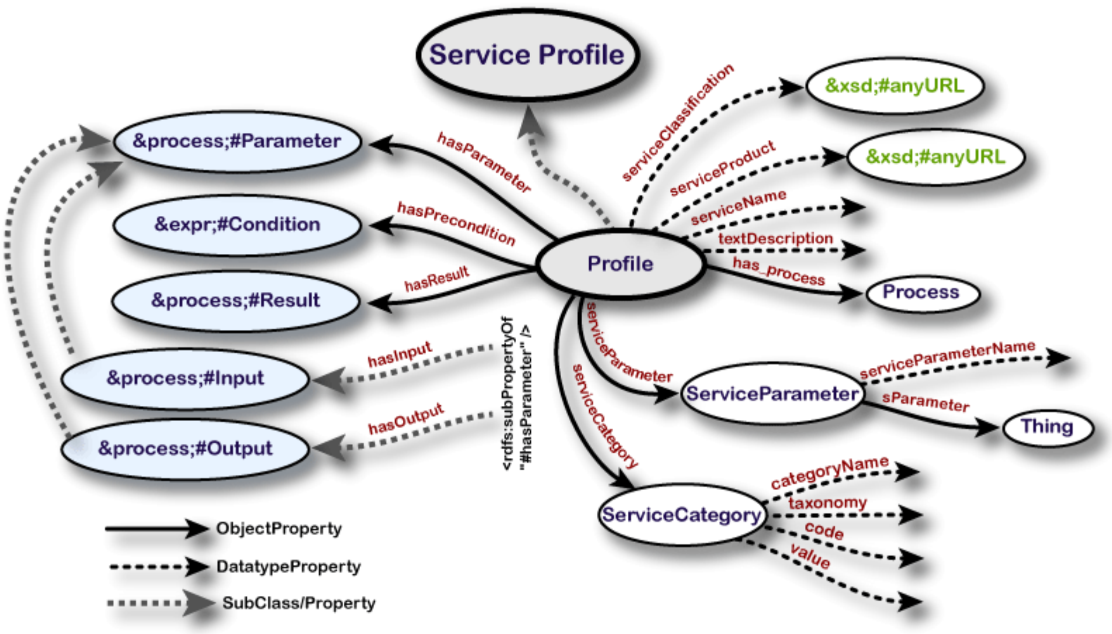

Figure 4.8. *OWL-S Service Profile structure.*

The ***Process Model*** describes the composition of one or more services. It provides then the enactment of constituent processes with respective communication pattern. In OWL-S, this is captured by a common subset of workflow features like split, join, sequence, and choice (cf. Figure 4.9).

As illustrated in Figure 4.9, a process in OWL-S can be atomic, simple, or composite. An atomic process is a single black-box process description with exposed IOPEs. Simple processes provide means for describing service or process abstractions which have no specific binding to a physical service, thus have to be realized by an atomic process, e.g. through service discovery and dynamic binding at runtime, or expanded into a composite process. An OWL-S process model of a composite service can also specify that its output is equal to some output of one of its sub-processes, whenever the composite process gets instantiated. Moreover, for a composite process with a Sequence control construct/format, the output of one sub-process can be defined to be an input to another sub-process. Finally, OWL-S allows the specifications of conditional outputs (inCondition).

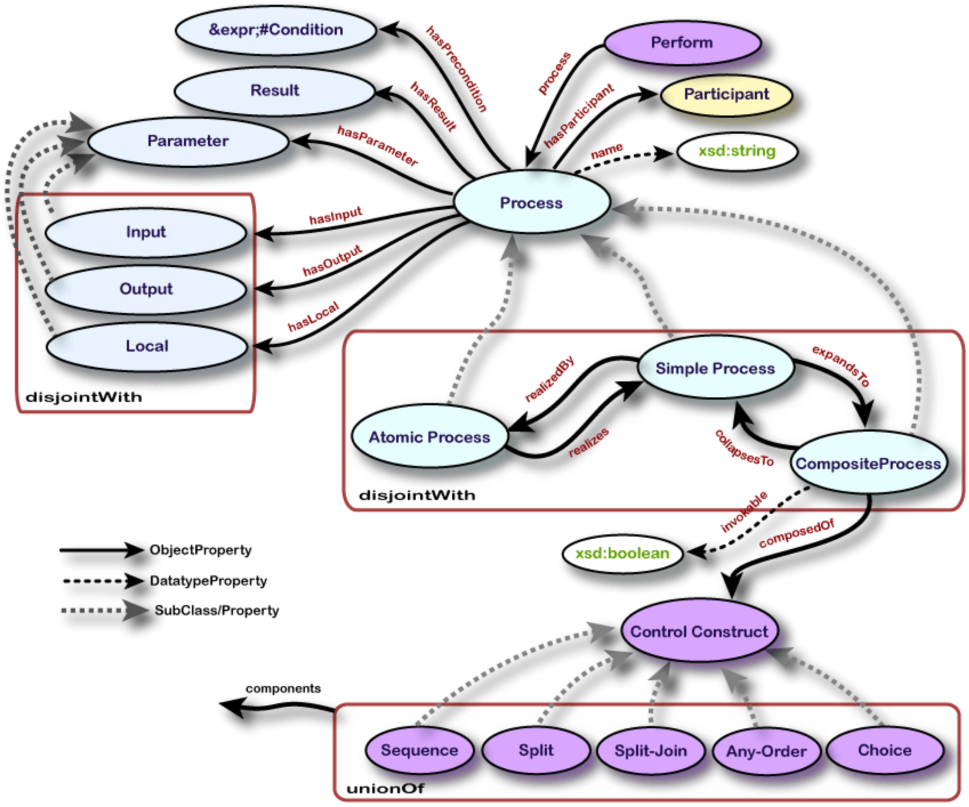

Figure 4.9. *The OWL-S Process Model*.

The ***Service Grounding*** of a given OWL-S service description provides a pragmatic binding between the logic-based and XMLSchema-based service definitions for the purpose of facilitating service execution. Since the WSDL is a widely used existing Web service, the grounding is exemplified for WSDL. In particular, the OWL-S process model of a service is mapped to a WSDL description in which each atomic process is mapped to a WSDL operation, and the OWL-S properties used to represent inputs and outputs are grounded in terms of respectively named XML data types of corresponding input and output messages. Unlike OWL-S, WSDL cannot be used to express pre-conditions or effects of executing services.

Examples of available software support of developing, searching, and composing OWL-S services are:

- OWL-S IDE integrated development environment, the OWL-S 1.1 API with the OWL-DL reasoner Pellet and OWL-S editors for development [152];
- OWL-S service matchmakers OWLS-UDDI, OWLSM and OWLS-MX with test collection OWLS-TC2 for discovery [153];
- OWL-S service composition planners OWLS-XPlan [154] for composition.

C. Semantic Web Rule Language (SWRL)

The Semantic Web Rule Language (SWRL) [149] is a combination of the OWL sub-languages (OWL-DL and OWL Lite) with the Unary/Binary Datalog RuleML sublanguages of the Rule Markup Language. It extends the set of OWL axioms to include Horn-like rules. It thus enables Horn-like rules to be combined with an OWL knowledge base.

The proposed rules are of the form of an implication between an antecedent (body) and consequent (head). Both the antecedent (body) and consequent (head) consist of a conjunction of atoms.

An abstract EBNF syntax, consistent with the OWL specification, is useful for defining XML and RDF serializations. However, the relatively informal "human readable" is often used. In this syntax, a rule has the form: *antecedent ⇒ consequent* where both *antecedent(body)* and *consequent(head)* are conjunctions of atoms. Variables are indicated using the standard convention of prefixing them with a question mark (e.g., ?x). Using this syntax, a rule asserting that the person that uses a device located at room1 is also located at room1 is written:

Person(?x) ∧ Device(?dev) uses(?x,?dev) ∧ locatedIn(?dev,office) ⇒ locatedIn(?x,office)

A problem that occurs when combining first-order ontologies with rules from normal logic programming is that the combination of decidable fragments of both worlds can be undecidable. Similarly, the combination of SHOIN(D) with function-free Horn in the SemanticWeb rule language SWRL is undecidable [155].

The syntactic DL-safety condition for rules avoids this problem [156]. It requires that each variable occurring in a rule must occur in a positive non-DL atom in the rule body, and may therefore be bound only to constants, that are named (not anonymous) individuals in the ABox of the DL part of the combined knowledge base.

Prominent examples of implemented frameworks that support SWRL are KAON2 reasoner, the FUB SWRL engine and Hoolet and the Jess rule engine that can be bridged to Protégé SWRLTab.

4.4.1.2. The OWL-based context model

In this contribution, we have kept the same context information that we have identified as relevant for multimedia services in the previous contribution (cf. section 4.3.1). However, by using ontology, beyond representing the structure of context information, the model describes semantic of context information and the relations between them.

In our model, context data is also classified as static and dynamic context. Dynamic context is the one related to the session context and experiences frequent changes during the session duration. Note that classifying context data in the static context does not mean that it does not change but only that it usually does not happen to change during the service usage. The static context is used in the service discovery and composition phases for more personalized services whereas dynamic context is used in the adaptation phase to keep the service quality at a satisfactory threshold.

As illustrated in Figure 4.10, we have considered six generic interrelated entities that we have identified common to any domain, namely the user, device, network, service, environment and session entities. As in the first contribution, the context is classified as either static context or dynamic context. The five first entities are then classified as static context entities and the last one is the entity to which the dynamic context information will be associated. Since in ontology, not only hierarchical relations can be modeled the device entity is just related to the network entity by the relation *connectedTo* and does not incorporate it as in the XML-based model. Each of the latter entities is then described and extended with more additional concepts and relations that are needed to provide personalized and dynamically adaptive Internet multimedia services.

The *User* entity is described in different profiles:

- The *GeneralProfile* that contains general information about the user such as name, age, etc.;
- The *SubscriptionProfile* that contains information on the different services for which the user have subscribed and the services that he may access;
- The *ContactProfile* that contains the contact information of the user such as his address, phone number, SIP URI, etc;
- The *Affiliationprofile* that contains information about the different organization to which the user is affiliated;
- The *AuthenticationProfile* that contains information that allows the user to be authenticated;
- The *PreferenceProfile* that contains the user-defined preferences or the deduced preferences from usage. The user preferences could be generic and applied to any service or situation or they could target a specific service or context entity and thus be applied only when the latter is involved.

The *Device* entity is described in terms of (1) *HardwarePlatform* that can be modeled in hierarchical way since the *components* can be *atomic* or *composite* and (2) *Softwareplatform* by presenting the User and System softwares that the device runs. As a specialization of application, the multimedia applications could be associated with other software such as the Codecs.

The description of the *Network* entity comprises information such as the name of the network and the theoretical parameters that characterize it. Dynamic parameters are described further in the Session entity.

The *Environment* entity is the union of different parameters. For each of them, we define its name and its value as well as its properties.

The *Service* entity represents the different services that the user can access. The Internet services provided by the Service Provider as well as the “home” services are represented within this entity. The service entity is modeled in an OWL-S Service Profile that models its IOPEs parameters. The service may subscribe for events that will trigger some actions integrated in the applications that implement the service.

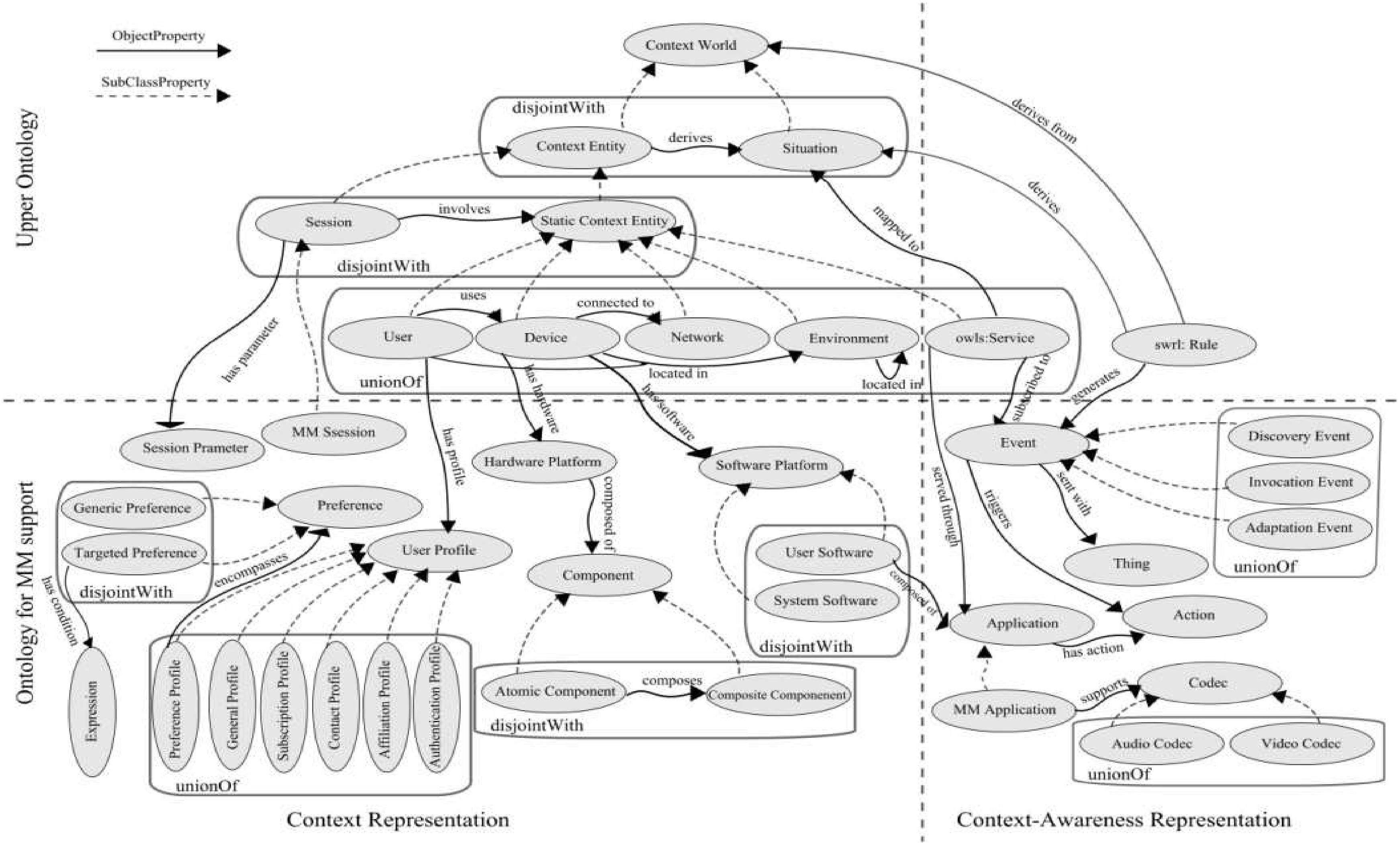

Figure 4.10.*Ontology based context model.*

The *Session* entity is related to different session parameters that model the dynamic context, related to the different entities involved in the session. Such parameters could be a user location, a loss or error rate experienced by a multimedia session within a related network and reported by monitoring modules, a dynamically evaluated user experience, etc.

In addition to the OWL property characteristics that enable reasoning, the model also include SWRL rules that add more expressivity to the model and allow some content adaptation and service personalization decisions triggering. More details and examples are given in the next section.

The personalization and adaptation processes usually need more context abstraction. These processes commonly base the behavior changes decision on high-level context (called also situational context). The latter is derived from the raw-level using DL- and rule-based reasoning context with the objective of triggering context-aware decisions.

As illustrated in Figure 4.10, our context model is then composed on two ontologies: (1) the upper layer ontology that contains these generic entities and their classification, and (2) the multimedia-specific ontology that extends these entities with concepts, relations and rules that allow a better support of multimedia services. It should also be mentioned that the model represents the context information as well as the way in which it will be exploited.

4.4.2. Context-based service management and adaptation

The Service Environment advertises each category of services to Home-Boxes using OWL-S service profile. The service parameters that will be given in the discovery and invocation requests are inferred from the user context and situation using SWRL rules. Discovery and composition tools such as OWLS-MX [153] and OWLS-XPlan [154] that support OWL-S could then be used to retrieve the appropriate service and compose it to fit the user context. In the cases where the service discovery process does not support the web semantic tools, the service description can then be grounded to WSDL [157].

The following rule selects the category and the size of video parameters according to the user preferences and location. The rule states that when Bob is at his office premises, he prefers to get the match highlights of recently played football matches when he requests for a VoD service.

```
User(Bob) ^ Service(VoD) ^ Environment( ?env) ^
hasPreferenceProfile(Bob, ?pp) ^
encompasses(?pp, ?pref) ^
hasPreferenceName(?pref, Video Category) ^
hasPreferenceValue(?pref, Football) ^
locatedIn(Bob,?env) ^ location(?env, Office) ^
presents(VoD,?service_profile) ^
hasinput(?service_profile, ? param) ^
ServiceParameterName(?param, Video Category) →
sParameter(?param, Football highlights)
```

Rules can also serve to implement some adaptation strategies and/or generate events that will trigger some adaptation processes. The delivered multimedia content can then be dynamically adapted to the context changes. The following rule example triggers a bitrate adaptation of the streamed video content when the network congestion results on a degradation of the service quality. The listener configured on the top of the relation *relatedTo* will send the event to the session manager within the required context parameters in the adaptation process.

```
Service(VoD) ^ Session(?s) ^ Network(?net)
involves(?s, VoD) ^ involves(?s,?net)
hasQoSLossParamThreshold(?serv, ?loss_threshold) ^
hasSessionParam(?s, ?sparam) ^
SessionParameterName(?sparam, registeredLoss) ^
SessionParameterValue(?sparam, ?loss) ^
swrlb:greaterThan(?loss, ?loss_threshold) ^
congested(?net, true) →
 relatedTo (?s, AdaptationBrEvent )
```

The rules are maintained by both the Service Provider to express management and adaptation policies and the user to set some preferences that are related to consumption of the subscribed services.

In addition to the management of the Internet services, the rules can also be used to manage the home services such as the control of the home appliances for more user comfort. The system can then for example dim light when the user is consuming video services via their TV or acts on the air-conditioner to prepare the home to the users' arrival according to their schedules.

4.4.3. Context-awareness middleware

Based on the context model and in collaboration with the other service management and adaptation components, the context-awareness middleware ensures a ubiquitous access to services for the related home users. As illustrated in Figure 4.11, the middleware consists in an intermediate entity/node/agent between the context providers and the context consumers. Indeed, raw context data is first acquired from multiple sources located at the different User, Service and Network Environments, then integrated to the context model on which new high level context is inferred and finally provided in the right format to the Service Environment that will take decisions on it and will perform some actions to adapt services and contents to the user context. The Context awareness middleware comprises the following functional components:

Similar to the first contribution, the ***Context Formatter*** is in charge of formatting the raw context data, acquired from the different context providers, in order to integrate it in the context model. Thanks to this, it permits to abstract the heterogeneity of context data format to the rest of the middleware components. In addition to the validation of context information, the context formatter is also in charge of the translation of the user and service policies to the correct DL-Safe SWRL format to be integrated in the rule set of the system.

The ***Context Knowledge*** encompasses the generic upper layer ontology and the specific one that allows the support of Internet multimedia services. It also integrates the management rules that will infer situational context.

The ***Context Manager*** is responsible for maintaining the context knowledge. It is also in charge of the subscription/notification process that supplies the Context Consumers with needed context information. It thus maintains a set of events that must be generated when the related situations are satisfied and sends them to the subscribed services along with accurate context information needed in the processes triggered by these events. Following the same behaviour as in the abovementioned context framework, the context manager is also responsible for maintaining the set of context providers that supply the context knowledge through sensing information. For each of them, it maintains its access information, the set of provided context information and its structure, the mode of context provisioning (periodic/on-demand) and its period, if the periodic mode is applied.

The ***Inference engine*** reasons on the context knowledge to derive (1) the high level or situational context that will trigger the service and content adaptation actions and (2) the context-aware service profile that will be incorporated to the services request to enable personalized service discovery and composition. The inference engine is based on DL- and SWRL reasoning.

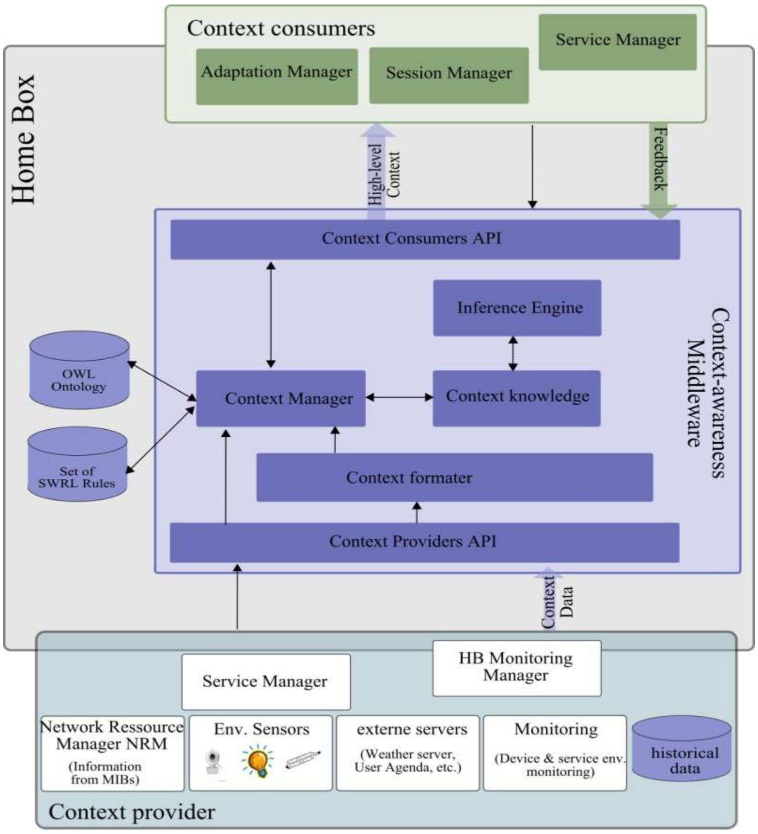

Figure 4.11. *The context awareness middleware.*

4.4.4. Middleware Implementation overview

We have implemented a prototype of the proposed context-awareness middleware in java 1.6. We have used Protégé-OWL API and the SWRLTab API to manipulate the context ontology and system rules as well to generate events by configuring listeners and associating to them the context information to be provided with the event. The rules execution is done by the Jess rule engine that is bridged to SWRLTab. The context-aware middleware prototype is deployed in the Home Box component with the following configuration: Intel (R) Core(TM)2 Duo processor with a frequency of 2.1 GHz and 4GB of memory.

The context ontology is stored as an OWL file at the Home Box. The ontology comprises the context of 5 users, 7devices and the description of 3 multimedia services (Video on

Demand, IPTV and telephony service). The OWL file is composed of 40 OWL classes, 91 properties, 327 individuals and 17 management rules. We have estimated the loading time of the context model to 903 ms and the inference time to approximately 350 ms. The results show that the context reasoning runtime is acceptable for multimedia services since the monitoring and adaptation frameworks have commonly longer periods. The loading time is somewhat long but the model is loaded once at the initiation phase.

Concerning the material resources capabilities needed by the middleware, we have derived the following conclusions. The inference process is light in memory since it only consumes about 75MB of the system memory which is available in all current material configurations. However, it takes around 50% of CPU processing which implies that the current Home gateway has to be enhanced with greater CPU capabilities to be able to handle reasoning-based context-awareness middleware and supply efficiently very context-sensitive services such as the multimedia services.

4.5. Conclusion

This chapter have presented our solutions for building a large-scale context-aware framework for Internet multimedia services that aim to provide end-users with ubiquitous multimedia services that always meet their context and situation.

Two solutions have been presented.

The first one is based on the markup-scheme languages that are often used in such a framework for their efficiency. Since XML only supports syntactic modelling, low-level semantic was implemented in the XQuery queries.

The second solution is based on ontology and related languages and tools. For this, we have proposed a context model using ontology and a set of rules that bind the situational context to services ensuring thus a context-based automated service retrieving, management and adaptation. To demonstrate the feasibility of both context-awareness frameworks, we have implemented prototypes of the context-awareness middleware which evaluation has shown that it presents acceptable runtimes to ensure good service quality. However it needs to be deployed in rich-resource equipment.

The Markup scheme-based solution is a lightweight solution that is suitable and more performance-effective for multimedia systems. In addition, this solution benefits from the generalization use of XML to reuse some standardized models and related tools. However, it only covers the syntactic level. Although some Intelligence and semantic are implemented in the XQuery requests and the context management, it is far to reach the semantic power of ontology.

The ontology-based solution is richer and allows more expressiveness, flexibility and extensibility. On the other side, it suffers from the complexity of ontology-based reasoning. Such a solution is more resource-hungry. The design decision of distributing the context-awareness feature among the HBs, makes such a solution feasible although its performance still highly dependent on the HB resources. The investments made by ISPs, in the deployment of more powerful home gateway encourages the building of such middleware on top of it.

Chapter 5

A VoD Content delivery solution over IP-based media distribution systems

5.1. Introduction

The wide adoption of Internet in the everyday life along with the growing popularity of multimedia services, that are known to be both resource-consuming and performance sensitive, places stringent requirement on current Internet infrastructure in term of scalability and performance. In addition, as the broadband connectivity becomes more and more pervasive, consumers' expectations for personalization and better quality experience in a disparate context increase.

Caching and replication are widely adopted techniques to improve Internet services' performance, scalability and reliability. Their aim is to store copies of contents at strategically chosen locations at the edges of the Internet, closer to end-users in anticipation of future requests.

This chapter presents our contribution on the Video-on-Demand (VoD) services distribution. Our proposal takes advantage of the new HB-layer caching capabilities to improve the services scalability, availability, reliability and access time, towards a better user experience. For this, the solution includes a context-aware request redirection algorithm and mechanism that allows end-users to always access the VoD service from the *best* server that meets the requirements from the End-Users, the Service Providers and the Network Providers as well as an online popularity-based video placement and replacement in HBs' caches that efficiently spread the video contents over the HB overlay.

5.2. Caching and replication in today's media distribution systems

Caching and replication are two different techniques commonly used in Internet best effort service distribution (cf. Section 2.3.2). However, they share many similar concepts and technologies. Caching copies of popular contents in strategic edge locations closer to end-users allows faster service access while saving network and server bandwidth. On the other hand, replication maintains the distributed contents under the control of content/service providers.

Caching can be classified in three categories according to the location of caches in the network [158], namely: the browser cache, proxy cache and surrogate cache. Browser cache is located in the client host as part of the browser to exploit the temporal locality of the user's requests. This type of caching has the least/smallest benefit since the cache is usually quite small and there is no sharing between end-users. Surrogate caches are located at the web server side and are typically owned and operated by the content provider. The aim is to accelerate the server's performance. Concerning proxy caches, they are located in nodes between the server and the end-user, typically closer to end-users than to the content provider servers. These nodes are owned by network providers or by companies operating caching nodes connected to the Internet. This type of caching reduces service access time and permits to save bandwidth by bringing contents closer to end-users. The distinction between these three types of caching techniques is useful from the deployment point of view. However, from research perspective, these caches types share many research challenges. Thus, solutions developed for one type can be directly applied to the others as it will be seen further in this chapter.

The benefits of caching and replication are numerous [158]-[159]-[160]:

- From the perspective of network infrastructures, these techniques decrease network traffic and thus minimize network congestions and improve performance. In addition, for Content Providers that pay ISPs to transport their traffic, reduced traffic means lower costs;
- From the content providers point of view, caching and replication reduce workload of their servers and service availability, reliability and responsiveness;
- From the client point of view, caching reduces significantly the service access latency for both popular contents since they get them from nearby servers and unpopular contents since the contents are faster retrieved due to reduction of network congestion.

However, there are a number of potential problems related to web caching and replication. For example, cache misses (when the content is not present in cache and has then to be retrieved from the origin server) decrease the service access time due to the cache processing. Users might also consume stale/out-dated contents if caches are not properly updated.

Maximizing the benefits of caching and replication solutions requires a careful and intelligent design. Some issues such as cache organization and cooperation, cache placement, decisions on the cachable contents, on when and where to place or replace contents and which cache will provide a requested content from a certain client, have then to be solved [58].

In this context, the following sections present our proposals to request redirection and content placement and replacement in the cache issues.

5.3. A context-aware request redirection scheme based on an anycast model

In this section, we present our proposal for an efficient context-aware application-layer anycast request redirection scheme. The innovation in the proposed approach consists of an adaptive request redirection algorithm that filters the multimedia services and contents for selecting the *best* server to handle the request. This selection considers both user and servers context, as well as the underlying network conditions. Indeed, the proposed server selection strategy permits a preliminary filtering based on the clients' contexts and provides accurate network distance information, using not only the end-to-end delay metric but also the servers' load one. This approach takes in consideration the path capacity and permits to obtain the best paths for offering End-Users the best QoE possible. This may or may not correspond to the least path in term of number of hops or delay. A transparent SIP-redirection mechanism is proposed to orchestrate the communication flow between the involved entities. The following subsections will present the features associated to the proposed solution, in a surrogate replication scheme as well as in the HB-based replication scheme. An evaluation of the solution is given for the two schemes.

5.3.1. The Anycast approach

Anycast is one of the schemes that support the request redirection (see 2.3.2.1-D). It was originally defined at the IP level [161] as a service that allows a node to connect to one, and maybe the "best" (according to the routing protocol metric), member of a group that serves the anycast address. The group is formed of resources that offer interchangeable services. It can then consist of the surrogates, either servers or proxy caches, which can deliver the requested content.

Although IP-level anycasting was described since almost two decades ago, very few practical networks have been built considering it. This is probably because of the technical challenges inherent in efficiently locating the best server. DNS [162] deployment and "6to4" [163] router constitute the most widely publicized uses of unicast addressing scheme. Other research works were made on network-layer anycasting. [164]-[165] focus on the scalability issue, others such as [166]-[167] focus on the design of routing algorithms based on active routers and papers such as [168] and [169] proposed proxy-based infrastructures to address network-layer anycast issues, like scalability or session-based services support.

However, this network-layer anycasting approach presents some limitations such as (1) the necessity for routers to support anycast and (2) the allocation of IP address space for anycast address, making difficult its integration on the existing infrastructure. Furthermore, this approach does not consider any user context options, neither the stateless nature of IP nor a set of metrics for choosing the most suitable server.

The network-layer anycasting limitations led the researchers to define the anycast paradigm at the application layer. In [101]-[108] and [102], the authors define the anycasting paradigm at the application layer as a service that maps anycast domain names into one or more IP addresses using anycast resolvers. The resolver decides which server among the replicated ones is the best based on different metrics at network and server side. For this purpose, the resolver maintains the servers' performance information. Application-layer anycasting consists of a good solution for distributed Internet services provisioning, especially when it requires no modification in the existing infrastructure. Another motivation to use application-layer anycasting is its ability to manage QoS and to define service requirements on a per-service basis.

In the field of multimedia services distribution, authors in [193] present an algorithm theoretically related to an economical model with a queuing theory based on the available free buffer, the available bandwidth, the average arrival rate of requests and the call blocking probability. In [170], three anycast-based multimedia distribution architectures are proposed, namely the identical, the heterogeneous and the semi heterogeneous architectures, to identify the best media server selection for different application domains. From our best knowledge, except in [170], all the above works have based their selection strategies on the servers' performance and have considered neither the client context nor the network conditions, which strongly affects the quality of multimedia services delivery. Additionally, the solution presented in [170] is only designed and evaluated in small-scale environment.

5.3.2. The Context-Aware Anycast Model and System

The proposed application-layer anycast architecture aims to provide clients with an efficient and transparent multimedia provisioning service. As described in chapter 3, we assume that the Service Environment (SE) hosts a Service Registry (SR) node which will receive the requests of its related clients. The SR has two main roles: first, it plays the role of an anycast resolver that is in charge of performing the mapping of the anycast address of the client request to the unicast address of the most appropriate server. Second, it will retrieve and maintain the servers' contexts and their contents descriptions. We also assume that the set of SRs can also collaboratively perform the server selection feature when the latter fail at one of them.

At a first step of our proposal for request redirection, we consider video content replication at the SE side [194]-[195]. Video contents are then replicated among surrogate servers distributed over the Internet. The infrastructure, as illustrated in Figure 5.1, is based on three types of nodes: the *client's nodes*, in the User Environment, requesting the service, the *server's nodes* providing the service and the *SRs nodes* serving the anycast address, handling the client's requests and performing the server selection strategy. Figure 5.1 also depicts the engaged dialogue between the different agents that run in the system nodes in order to establish the media session. This process is more detailed in the following sub-sections.

The agents that run in the different system nodes are communicating using SIP (Session Initiation Protocol) [171], signalling protocol for handling multimedia session and its extension for event state publication [172]. SIP is foreseen to become the key signalling protocol for Next Generation Networks (NGN) platforms. Even though coming from the Internet world, it has already taken over the Telecommunication world (e.g. Tispan, IMS, 3GPP [76]) and operators place high expectations on it. Therefore, it appears as the predominant candidate for Future Internet signalling. Following this conjecture, the designed communication protocol has been based on SIP as illustrated in the sequence diagram of Figure 5.2.

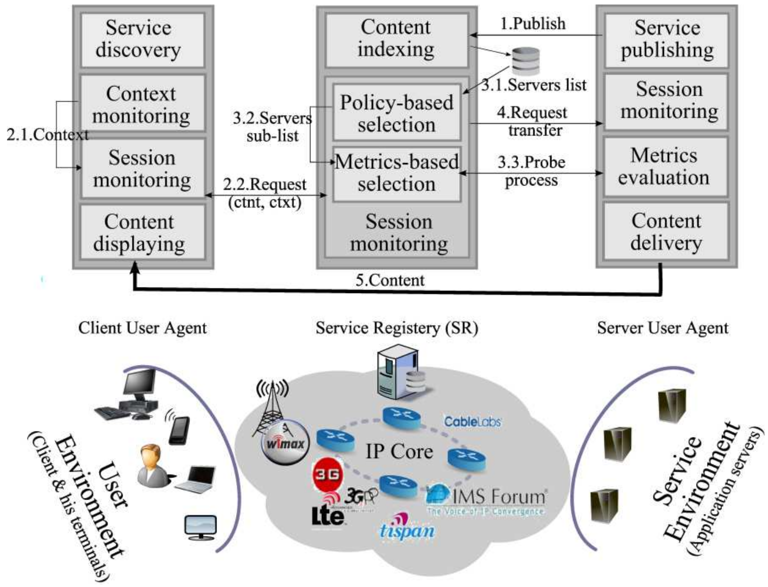

Figure 5.1. *Anycast functional architecture.*

The different processes involved in the request redirection mechanism and the SIP messages used to support the system nodes communication are detailed in the following sections.

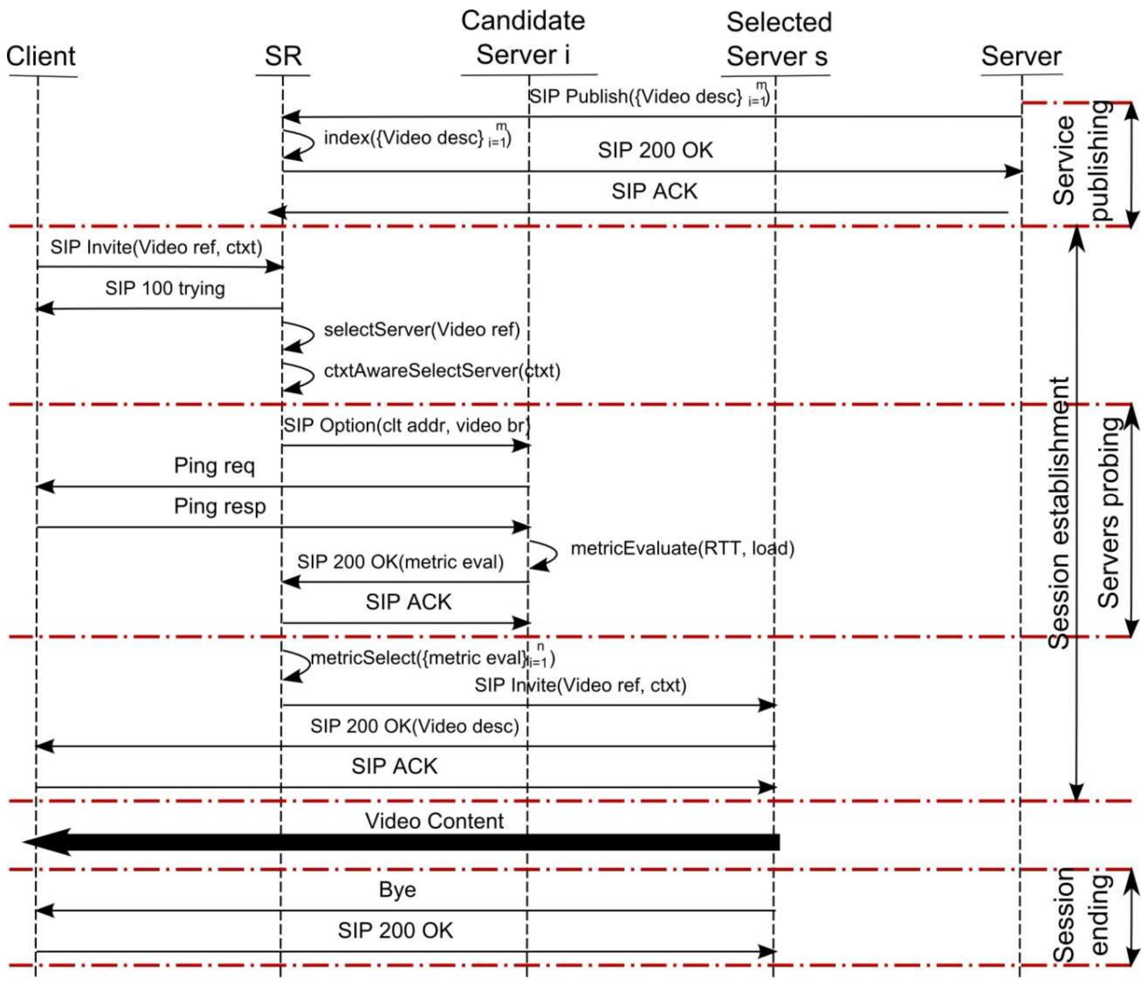

Figure 5.2. *SIP-based interaction flow.*

5.3.2.1. Service publishing

We consider in this work that the servers have heterogeneous contexts (output link bandwidth, location, etc.) and provide clients with different contents. Each node that wants to act as a server should send a *SIP Publish request* message within the anycast service address, as illustrated in Figure 5.2. The request contains its context and the list of the contents that it will deliver with their metadata. In the case of Video Streaming service, the metadata could correspond to the video content reference, the available coding formats, the resolution, the bitrate, the language of the video, etc. The request will then be directed to the nearest SR by the underlying routing protocol. This SR caches the received information and replies to the server with a final 200 ok response.

5.3.2.2. Servers probing

To maintain the dynamic server's information database, the SR must probe the servers. It can do it in either a proactive or a reactive manner. However, a proactive or periodical probing cannot ensure the accuracy of the retrieved information. Improving accuracy means decreasing the probing period which in fact, leads to increasing the network and servers load. In addition, one of the metrics used in our server selection strategy is the server to client delay, for which we need to transmit the client's address to the server. For these reasons, we opted for the proactive probing. When the SR receives a *SIP Invite request* from a client, it selects from its database a list of candidate servers and sends to each of them a *SIP Option request* containing the client address and required bandwidth.

The server calculates its decision based on its load and the path delay between it and the client. Then, the server sends the result to the interrogating SR in a *SIP 200 ok response*. Thanks to this probing, the SR obtains the accurate information about the servers and network performance and can thus select the most suitable server according to the retrieved information. The comprehensive detail of the server selection strategy will be given in the next section.

5.3.2.3. SRs collaboration

The SRs are interconnected in a multicast scheme. When a SR receives a client request and concludes, after processing it, that the content cannot be delivered by its registered servers (or that the required QoS cannot be ensured due to, e.g. an overloaded situation), it multicasts the request to its neighbour SRs. Each SR processes the received request and selects, if possible, a suitable server among its registered servers. Then, if the selection has led to a result, it responds to the original SR of the request with a *200 OK response* containing the IP address of the selected server and its evaluation of the selection function. Otherwise, it responds with a *404 not found* response. The original SR then selects the most suitable server (for example according to network proximity) among the received responses and forwards the client request to it. In case the SR receives only *Not Found* responses, the request fails and the client is notified.

5.3.2.4. Session establishment

The video session establishment process is performed through the previous processes as illustrated in the sequence diagram of Figure 5.2 and explained in the following:

- We assume that the servers have published their contents as explained in the session publishing sub-section (cf. section 5.3.2.1);
- The client trying to access the anycast service initiates a session with a simple *anycast Invite request*, into which the requested content is specified, as well as the user context provided by the context-aware feature (as described in chapter 4);
- The request is then routed to the nearest SR which will (1) send to the client a temporary *SIP 183 response*, (2) retrieve the client context and requirements and (3) try to select, alone or in collaboration with the others SRs, the most suitable server among a set of candidate servers. The result of the selection is based on one hand on the user context and on the other hand on the information retrieved in the probing stage, as it will be explained in the next section (cf. section 5.3.3);
- If the SR succeeds to select a server, it positively responds to the client request. Otherwise, it sends to the client an error response to inform it that its request has failed;
- In the case where the selection succeeded, the selected server receives the client *Invite request*, processes it, confirms the establishment of the session with the SIP 200 response and finally starts the streaming of the requested content to the client.

5.3.3. Context-Aware Server Selection Algorithm

As explained previously, our proposed approach for video distribution is mainly based on the selection of the *best server* for each client request among an anycast group of servers. By *best server* here, we mean the non-overloaded server that best suits both the client environment (connectivity and user & terminal characteristics) and the requirements of the underlying network conditions – from the server to the client – for finally improving the perceived QoE at the client side. To this end, a two-level filtering technique has been conceived in order to ensure the accuracy of the selection result: the first is based on “policy-based filtering”, the second on “metrics-based filtering”. The server selection algorithm relies on this 2-step filtering process as follows.

5.3.3.1. Policy-based filtering

Each SR maintains at its side all the list of video contents published by the attached video streaming servers. For each video, it maintains both a set of servers that deliver it and a set of metadatas that describe it.

When requesting a service, the client specifies in addition to the requested video reference, its context (available bandwidth, terminal resolution, etc.) and preferences like the video language by including them in the *SIP Invite request* that initiates the service. The user context is acquired from the Context-Aware function. The registered services at the SR side are then filtered by a set of predefined policies (cf. section 5.3.3.1) in order to only keep the services that deliver contents matching the user context. The policies define the mapping of, on one side, the maintained servers' contexts and related contents information and on another side, the client context and requirements. The mapping is highly dependent on the model in which the user context and the service and content metadata are represented.

At the end of this step, the SR selects a list of servers which provide contents that meet the user context, in order to take advantage as much as possible of the context-aware feature.

5.3.3.2. Metrics-based filtering

The objective of this second phase is to select one server (the best) from the list constructed in the previous step. This selection will be performed based on a distance evaluated during the selection process. The metrics to use and their exploitation strategy directly depend on the application and will therefore be selected accordingly.

Since we address the video streaming service, which is known to be very sensitive to the packet loss and delay metrics, the main requirement that we have considered when designing our server selection strategy is to avoid congestion and this at different levels. At the policy-based step, we have considered the congestion at the client level by taking into account the client context and thus the client connectivity.

At this step (metrics-based), we consider the congestion at both the server and network levels. Thus, the defined filter for this step is a weighted function that involves two metrics: the server load and the server-to-client delay. The combination of these two metrics permits to avoid congestion (1) at the server side by avoiding server overloaded situation, and (2) at the network side by considering the current client-to-server delay. The evaluation of this function is processed as follows:

$F(A_c, Rbr_c)\{$

$\quad d_{sc} = delay(A_c);$

$\quad \underline{if} \left(\left(\sum_{i=1}^{n} Rbr_i + Rbr_c \right) \Big/ br < 1 \right) \underline{then}\ \{$

$\quad\quad \underline{return}\ d_{sc};$

$\quad \}\underline{else}\ \{$

$\quad\quad \underline{return} \left(\alpha * d_{sc} * \left(\left(\sum_{i=1}^{n} Rbr_i + Rbr_c \right) \Big/ br \right) \right);$

$\quad \}$

$\}$

Figure 5.3. *The server selection function algorithm.*

The SR probes all the servers that constitute the retrieved sub-list, from the policy-based filtering, in order to evaluate for each of them the server selection function *F* described in . Because *F* combines the server-to-client delay and the server load, the Figure 5.2the probing *SIP Option request* must contain the client address A_c and the required video bitrate Rbr_c. On the other side, the server must also maintain its load. Indeed, whenever the server accepts the establishment of a multimedia session or ends one of its current sessions, it must update its load. The current load is calculated as follows:

$$load = \sum_{i=1}^{n} Rbr_i \Big/ br \qquad (1)$$

Where n is the number of currently active video sessions at the server side, Rbr_i is the already required and allocated bitrate for the video session i and br is the bitrate of the output link of the server.

The probed servers then evaluate the function F. As illustrated in Figure 5.3, F is based on the server-to-client delay d_{sc} and the server load. While the server is not overloaded, it only takes into account the client-to-server delay. But as soon as the server gets overloaded, the function involves both the two metrics and the priority is inversed. The α parameter should be fixed according to the network topology in order to give the server load metric the top priority.

After receiving the servers' evaluations of the selection function F, the SR selects the best server *s* such as:

$$F_s(A_c, Rbr_c) = \min_{1<i<m}\left(F_i(A_C, Rbr_c)\right) \qquad (2)$$

Where m is the number of the received responses. It should be noticed here that m is not necessarily equal to the number of probed servers. For each client request, the SR sets a timer and when this timer expires, if the SR had not yet received all the responses from the probed servers, it selects the best server based on the received responses.

5.3.4. Performance Evaluation of the context-aware anycast-based server selection

5.3.4.1. Simulation environment

This section highlights the evaluation of the anycast video distribution approach explained in the two previous sections. The simulation was done using the Network Simulator NS2. The network topology consists of 1000 clients, 500 servers, 5 servers registry and 100 routers placed in 5 Autonomous Systems (AS). The connectivity between routers is constant at 2 Gb/s. Servers also have a constant connectivity of 20Mb/s. Client connectivity, however, varies between 512Kb/s and 100Mb/s.

The simulated video streaming service is providing clients with 10 different video contents. All the videos are present at all servers sides but in different resolutions and bitrates. Each video can be provided in three resolutions: 352x288, 720x576 and 1408x1152 and for each resolution in 3 different bitrates. The probability of the availability of a video at a given server with the first resolution is 1/5 and 2/5 for the two others. Each client requests a service one time during the simulation time. The client requests are generated in a Poisson model during 250s and the requests are uniformly distributed on the five ASs. We assume that all videos have a minimum duration of 250s for keeping active all the 1000 video sessions simultaneously.

5.3.4.2. Simulation metrics and results

For evaluating the effectiveness of our video distribution protocol, we compare it to the random server selection scenario based on the uniform distribution that we simulated in the same environment. In this paper, the comparison is done according to different metric parameters: the average servers' overloads, the server load variance, the average packet loss, the

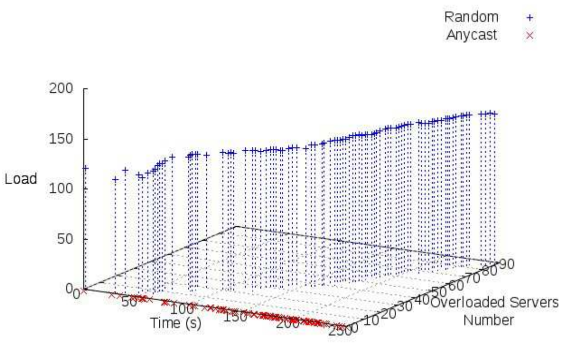

Figure 5.4. *Sim 1 - Overloaded servers' number and average load*

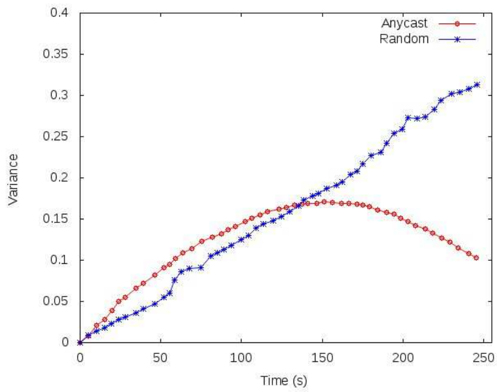

Figure 5.5. *Sim 1 - The servers' load variance*

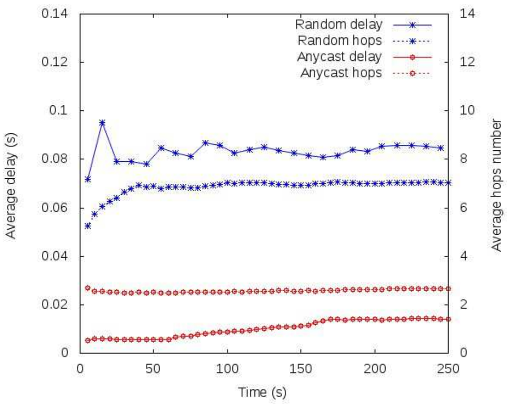

Figure 5.6. *Sim 1 - Average path distance (delay & hops number)*

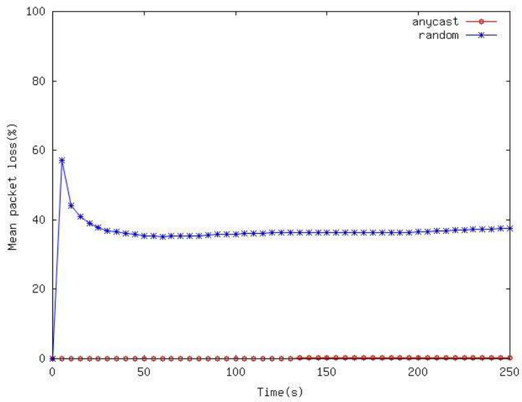

Figure 5.7. *Sim 1 - Mean packet loss at all active session*

Metric	Anycast	Random
Context matching	100%	33.7%
SIP overhead	74.59 KB	1.72 KB
Total service throughput	219.8 MB	203.14 MB
Inter AS connections	0%	78.5%
Average service response time	1.05 s	0.02 s

Table 5.1. *Additional simulation results*

average path distance (delay and hop number) and some additional results such as the selection overhead, the service throughput, etc.

A. Overload

Figure 5.4 depicts the evolution of both the number of overloaded servers and their mean charge. As illustrated in the figure, no server was experiencing overload during all the simulation time under our approach. By contrast, the overloaded servers' percentage reaches 17.4% after initiating all the sessions under the random approach. The mean overload of these servers varies between 120% and 150%, which induces congestion at the servers' side.

B. Servers' load variance

The server load variance metric reflects the achievement of load balancing. Figure 5.5 depicts this metric parameter versus time. In the first stage of the simulation [0s ,137s], the random selection approach is experiencing a less servers' load variance than our approach, reflecting the fact that, in our approach, since the servers are not overloaded the top priority is given to the path delay when calculating the network distance. However, in the second stage [137s, 250s], the variance in the random selection continues to increase while it decreases in our approach. This is the consequence of giving the priority to the servers' load when calculating the network distance in this stage. The curves illustrate a more efficient spread of the clients' requests in our approach.

C. Packet loss

Figure 5.7 depicts the average packet loss of all the initiated sessions versus time. We can note that the packet loss under our approach is almost none. It approaches 0.22% with 1000 active sessions. On the contrary, the selection scenario reaches a loss percentage greater than 35% during all the simulation time, and this even with less traffic to manage (cf. Table 5.1). The effectiveness of our approach is the result of better congestion avoidance at all levels, especially in the User Environment, where packet loss is known to be an essential aspect. Indeed, the *last-mile* is known to be the most common causes of packet loss and consequently, the video service degradation over the end-to-end path. The proposed server selection corrects/alleviates this problem in its first stage, by selecting only the services that cope with the user context. Moreover, the packet loss is avoided at the server level by considering the servers' load and at the network level by getting context from nearby servers.

D. Average path distance

Figure 5.6 represents the average path distance, represented in terms of path delay and path hops number. We can clearly note that the mean path distance is reduced with our solution for both metrics. Indeed, under our approach, the mean path delay varies in the interval [5ms,14ms] and the mean hops number is almost 2.5 while, under the random scenario, the mean path delay varies in the interval [71ms, 96ms] and the mean hops number varies within [5.2, 7.06]. These results reflect the consideration of the server-to-client delay metric in our selection strategy. Indeed, the SR always selects the nearest server among the non-overloaded candidate servers.

Thus, we can conclude that our approach ensures a better service delay but also a better distribution of clients' requests on servers and, consequently, a better congestion avoidance at network level, as seen in Figure 5.7.

E. Some additional results

Table 5.1 summarizes some additional results such as the delivered content matching the client context, the percentage of inter AS sessions, the average SIP overhead, the service throughput and the service response time. We can note that, thanks to the context-aware selection, the delivered video content always meets the client context contrarily to the random selection scenario where the percentage is 33.7%. We can also note that although the SIP traffic overhead (due to the exchanged messages in order to perform the server selection) is greater under our approach, it is not significant comparing to the total service throughput. The additional processes necessary to perform the selection strategy also induce a longer service response time (the duration between the requesting of the service by the client and the establishment of a media session between the client and the selected server), as seen in Table 5.1. Even though the service average response time under our approach lengths 1.05 s, it is still far from reaching the 32 s fixed in [171] to conclude that the SIP session has expired.

5.3.5. Anycast-based server selection applied to the HB-based video delivery infrastructure

In this section, we apply the context-aware anycast-based server selection explained in the previous section to the proposed architecture [196]. At this step of our work, we consider replication at the HB nodes. As explained in chapter 3, the HB is enhanced with caching capabilities. We assume that contents are pushed from content provider servers to the HB overlay. The way the contents are replicated over the HBs will be explained in the next section.

5.3.5.1. Video placement in HB caches

The goal of placing video contents in the Home Boxes' overlay is to minimize the clients' costs to access them. Indeed, the Home Box is located at the User Environment which makes it the closest equipment to end users that can be managed by SPs. So, considering caching solution on this equipment will significantly decrease latency, save the server and network bandwidth and accordingly, decrease packet loss.

The video contents placement in the HBs caches is performed as follows:

Considering N videos, classified in the order of their popularity from 1 to N, only the set of most popular videos Vp are cashed in the HBs. Previous works such as in [173, 174] demonstrated that video popularity fits the zipf-like distribution with highly variable skew factor. Relying on the zipf-like distribution, the probability of requesting the video content of rank k is:

$$P_N(k) = \frac{1/_{k^\alpha}}{\sum_{i=1}^{N} 1/_{i^\alpha}} \quad (2)$$

Where, α is the skew of the distribution.

In our Model, the video popularity is also reflected in the frequency of the video at the HBs overlay. The frequency of presence of the most popular videos is also evaluated based on the video popularity with the same skew value. Accordingly, the more the video is popular, the more its frequency of presence in caches is bigger. Then, the probability of presence of a video of rank k in the HB caches is:

$$P_{|HB|}(k) = \frac{P_N(k)}{\sum_{j=1}^{M_p} P_N(j)} \quad (3)$$

The videos are placed in HB caches in the order of their popularity. The most popular are placed first. For simplicity, we assume that the video popularity is user location independent. Thus, for each video, the set of $\left(P_{|HB|}(k) * |HB|\right)$ HBs that will cache it are selected uniformly from the $\left(\left(\sum_{i=1}^{k-1} P_{|HB|}(k)\right)| * HB|\right)$ available HBs, with k being the video popularity rank. Consequently, the most popular videos are then best spread.

5.3.5.2. Server selection strategy

Our selection is based on the context-aware anycast server selection strategy explained previously and relies on a two level filtering process. Nevertheless, the selection was adapted to the new architecture that includes the HB virtual layer. Since the content can be accessed from both HBs and CP servers in our model, the term “server” designates both of them in the following sections.

As detailed in the pseudocode bellow , the Service Manager selects the closest server, from the resulting list of the policy-based filtering retrieved from the SR, based on accurate metrics retrieved thanks to an on-demand probing process. Each probed server evaluates its load and returns it to the Service Manager. Based on the load information retrieved from the servers themselves and the server-to-requesting HB delay retrieved from the Network Ressource Manager (NRM), The Service manager evaluates the distance (as explained in section 5.3.3.2) for each server and proceeds to the selection process. The result will be either the closest HB if the responding HBs are not overloaded or otherwise the closest CP server.

```
/****************************************************************
     SMngr side pseudo code
****************************************************************/

inputs :
  L ;         //list of servers provided by the policy-based step
  A ;         //address of the local HB
  br ;     //required bitrate for the media session

//probe servers to get accurate metrics values
function send_probe_req( L, A, br) {
  foreach s ∈ L do {
    send_req(br);
    retrieve_delay(s, A)  /*retrieve from cache if it exists
                            otherwise from NRM the delay from s
                            to the client  */
    arm timer t;
  }
}

// call the select function if the timer expires
function timet_expire(t) {
  //check if the selection hasn't been done
  if (!selected) {
    selected = true ;
    select();
  }
}

/* update the servers context and call the select
   function if all the servers have responsed      */
function receive_probe_resp(resp, s) {
  if (!selected) {
    L[s].load = resp.load;
    nb_resp++;
    if (nb_resp == L.size()) then{
      selected = true ;
      select();
    }
  } else
    Ignore response;
  }
}
```

```
/*select the HB or the Content Server
  that will respond the client        */
function select() {
  HBdelay = ∞;
  Sdelay = ∞;
  selecteHB = NULL;
  selecteS = NULL;
  foreach s ∈ L do {
    if L[s].type == HB) {
      if (L[s].load > 1) then{
        L.erase(s);
      }else if (L[s].delay < HBdelay) {
        HBdelay = L[s].delay;
        selecteHB = s;
      }
    } else {
      if (L[s].load > 1) then {
        L[s].delay = α * L[s].delay * L[s].load ;
      }
      if (L[s].delay < Sdelay) then {
        Sdelay = L[s].delay;
        selectedS = s;
      }
    }
  }
  if (HBdelay < ∞) then {
    return selecteHB;
  } else {
    return selecteHB;
  }
}

/****************************************************************
     server side pseudo code
****************************************************************/
inputs :
  sum_br ;  //sum of engaged media sessions bitrates
  br ;              //the bitrate of the server uplink

//process the probe request and reurns accurate metric values
function receive_probe_req(req) {
  load = (sum_br + req.br) / br;
  send_probe_resp(load);   /* response SMngr whith the
                              requested metrics values*/
}
```

5.3.5.3. Video delivery cost

An important parameter to measure the video distribution efficiency is the distribution cost. Since getting the video from neighbour HB is almost always less than getting it from a content provider server, the aim is then to maximize the HB overlay hit ratio (the ratio of request satisfied by the HB overlay from the total number of requests). Then before deriving the system cost, we will first derive the HB overlay hit ratio $h(R)$.

Assume C the number of clusters formed by the HBs and R the request rate that represents the average arriving requests per T, which is also the time that takes the video to be streamed. R is then, the number of concurrent streams and also the duration for which we will derive the system cost. But first, let us consider the hit ratio h_R of HBs caches for the R concurrent streams. h_R is defined as the ratio of video requests satisfied by the HB caching solution.

As explained in the last section, the requested video i is accessed from an HB cache only if the video is present at one or more than one HB of the same cluster and the session can be accepted without exceeding the upload capacity of this cluster. For this, we assume n the average number of videos that a HB can stream in parallel. The average number of supported streams S_i in a cluster for the video i is:

$$S_i = \frac{1}{C} * P_{|HB|}(i) * |H| * n \qquad (4)$$

The number of the preceding requests for the video i shouldn't then exceed $S_i - 1$. We can then derive the hit ratio h_j at the *j*-th request, as follows:

$$h_j = \begin{cases} \sum_{i=1}^{|V_P|} \frac{P_N(i) * P_{|HB|}(i)}{C} & if \;\; j < S_i \\ \sum_{i=1}^{|V_P|} \left[\frac{P_N(i) * P_{|HB|}(i)}{C} * \sum_{l=0}^{S_i - 1} P_N(i)^l * (1 - P_N(i))^{j-l-1} \right] & otherwise \end{cases} \qquad (5)$$

The hit ratio over the R requests is:

$$h(R) = \frac{1}{R} \sum_{j=1}^{R} h_j \qquad (6)$$

The final expression of hit ratio is then:

$$\mathrm{h(R)} = \frac{1}{\mathrm{R}} \sum_{\mathrm{j}=1}^{\mathrm{R}} \sum_{\mathrm{i}=1}^{|\mathrm{V_P}|} \left[\frac{\mathrm{P_N(i)} * \mathrm{P_{|HB|}(i)}}{\mathrm{C}} * \sum_{\mathrm{l}=0}^{\mathrm{S_i}-1} \mathrm{P_N(i)^l} * (1 - \mathrm{P_N(i)})^{\mathrm{j-l-1}} \right] \qquad (7)$$

Consider c_{hb} the average cost to access the video from a HB and the c_s the average cost to access the video from the closest CP. The total cost of our video delivery model is then:

$$cost = R * \left(h(R) * c_{hb} + \left(1 - h(R)\right) * c_s\right) \qquad (8)$$

Note that, this cost does not comprise the HBs' caches maintenance. This process is made offline, in the time in which the traffic decreases. Indeed, many works [173] have been carried to study the user behavior in large scale VoD systems and concluded that the user access follows a clear daily pattern. Another important conclusion of this work is that the users' arrival rate varies significantly over the day time. Generally, the access pattern is subject to important peaks during breaks, after work and during the week-end. It is why the maintenance of the HBs caches is scheduled when the video traffic decreases, i.e. in the early morning, according to the study.

5.3.5.4. Performance evaluation of the anycast server selection combined with HB caching

A. Simulation environment

This section presents the evaluation by simulation of the video streaming delivery presented in the previous section. The simulation has been done under the Network Simulator NS2. The network topology is derived from the Survivable Network Design Library (SNDlib) [175] that count 22 IP backbone networks. We have selected the FRANCE instance. We kept the node placement and the link assignment. However, we made our own assumptions on the service demand and the links capacities have been fixed to 2.5 Gbps. To each node, we have attached 20 HBs with 70 Mbps downlink capacity and 30Mbps uplink capacity. To the HBs, we have attached 1 to 4 clients with an average of 2 clients per HB. The clients' connectivity varies from 0.5 to 100 Mbps. We have also attached four contents servers, SR and Service Manager nodes to the highest degree routers with 2.5 Gbps connectivity. Figure 5.8 illustrates the simulation topology.

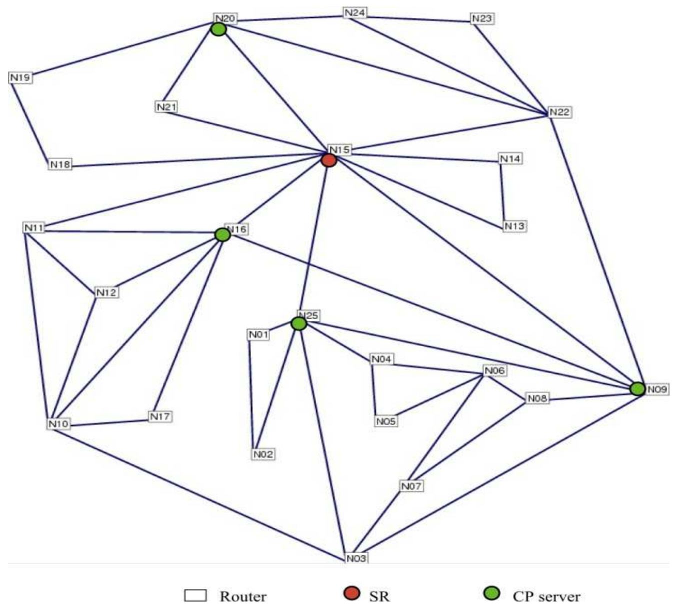

Figure 5.8. *Simulation topology.*

Concerning the served video contents, we have considered 500 videos. All videos are present in each server. A video can be streamed in three resolutions 352x288, 720x576 and 1408x1152 and in three different bitrates for each resolution. The 100 most popular videos are cached within the HB layer. Each HB can hold two videos. The video frequency at the HB layer fits the zipf-like distribution. All the 100 videos are uniformly distributed among the HBs of the overall topology in the order of their popularity.

The video popularity is also zipf-like distribution. Each client requests one video during the entire simulation time. The clients' requests are generated in a poisson model during 250 s which is also the video duration. All the videos have the same duration. For each request, the probability to request the video in the three resolutions presented above is respectively 1/5, 2/5, 2/5. The video bitrate is selected according to the client connectivity.

The proposed delivery approach is evaluated in two scenarios: the first one with the support of the 50% of HBs and the second with the support of all the HBs (100% of HBs). To evaluate the approach, we have also compared it to the server selection without any HB support and the widely studied Server Load based selection. All the scenarios are simulated in the same environment. However, since the server load selection does not consider any context-aware features, at most two videos can be served in the maximum resolution to the same HB in order to avoid rapidly overloaded situations. We have compared the four scenarios according to different metrics, namely: the average server load, the RTP path distance in terms of delay and hops number, and finally the RTP packet loss.

Server load: Figure 5.9 illustrates the evolution of the average server load for the four scenarios through the simulation time. We can easily notice that the more HBs support the video delivery, the more the server load is saved. The average server charge is just 50% with 100% HBs support and 60% with 50% HBs support while it reaches the 90% without HBs support and 85% with the server load selection. The system cost is significantly reduced with the HB support.

Packet loss: Figure 5.10 clearly shows that our approach avoids losses. It also shows that the higher is the distribution, the less are the probabilities for packet loss to occur. While the packet loss reaches the 10% when having all the sessions actives under the server load based selection, it is reduced to 5%, 0.2% and 0.15% under our server selection and this with respectively 0%, 50% and 100% HBs support. The Packet loss decreasing is the result of avoiding congestion (1) at the user level by considering the user context, (2) at server level by considering the server load and (3) at network level by bringing the contents closer to the clients and taking into account the delay metric to select from which server the client will access the content.

Path distance: Figure 5.11 and Figure 5.12 illustrate the path distance in terms of delay and number of hops metrics. We can notice that the path distance is significantly reduced for both metrics under our selection approach. The delay, close to 110 ms under the server load based approach, decreases to around 30, 20 and 15 ms under our approach with respectively 0%, 50%, 100% HBs support. This result is due to considering the delay metric in the metric-based selection phase and to the HBs support for video delivery. The contents are then closer to clients and the delays are accordingly shorter. We can also observe that the evolution of the path delay is slow under our approach while it increases rapidly under the server load based selection strategy.

B. Simulation results

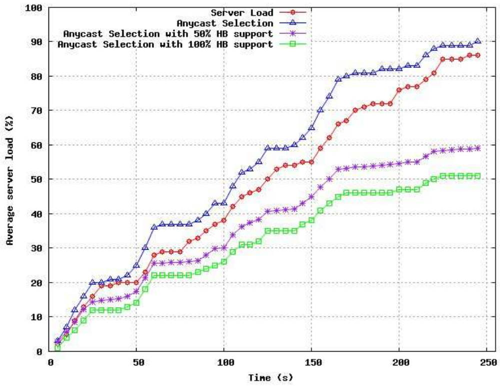

Figure 5.9.*Sim 2 - Average server load vs simulation time.*

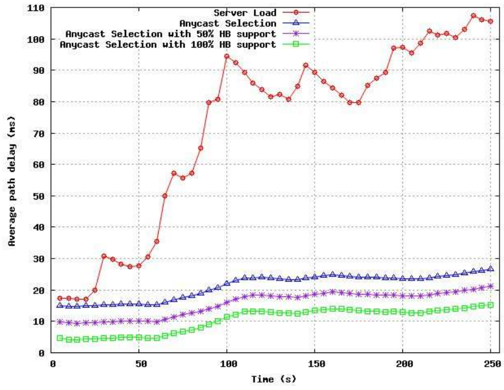

Figure 5.11. *Sim 2 - Average path delay vs simulation time.*

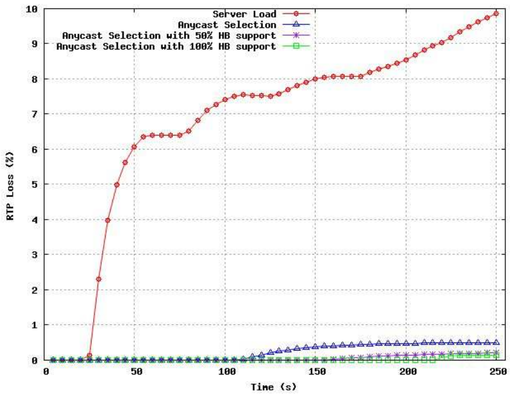

Figure 5.10. *Sim 2 - Average RTP packet loss vs simulation time.*

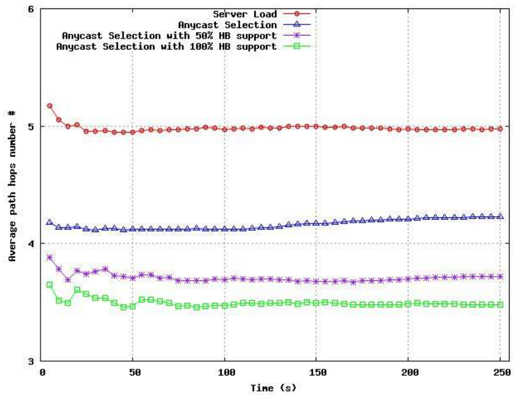

Figure 5.12. *Sim 2 - Average hop number path vs simulation time.*

5.4. A Popularity-based Video Caching Strategy

This solution [197]-[198] extends the latter one in two ways: i) by considering a peering CDN that extend the CDN capacity by introducing the Home-Box (HB) equipment in the service delivery chain and ii) the design of an adaptive popularity-based video contents caching strategy. The HBs form then a managed P2P overlay able to support CDN servers in the delivery process. Video contents are dynamically cached in HBs following users' demands, allowing their delivery to End-Users from optimal places. The idea is to leverage the participating and already deployed Home-Boxes disk caching and uploading capabilities to achieve service performance, scalability and reliability, especially in current context where the broadband providers are heavily investing to build out their high speed last mile networks.

Recently, some studies on peer-assisted CDNs have been proposed. The great advantage of such solutions have been demonstrated analytically [176], by simulation [177], or with real implementation and deployment at a large scale [178]. Our proposal differs from these works by relying on the User Environment's storage and connectivity without directly involving clients' nodes participation. The peer node, in our model, is the HB that, despite the fact that it is located in the User's Environment, it still acts as the always-connected "hub" between the User, Network and Service Environments, respectively being capable of shared ownership between the three actors (End-User, Service Provider, Network Provider). Therefore, the HB can be easily managed and monitored by the respective actors, feature that cannot be possible with End-User terminals. Our model is then a fully managed model and consequently overcomes the issues induced by the P2P part in this kind of hybrid system (e.g. free riding, reliability, peer churn issues, etc.).

Enabling a distributed edge content hosting is considered as the next step in content distribution paradigm [179]. Recent works have proposed architectures that rely on boxes deployed at the edges of the network, close to Users' terminals, for live video streaming services [180]-[181] or VoD services [182]-[183]. However in the latter, video contents are placed offline, which involves an additional delivery cost. In our proposed solution, the videos are placed during their consumptions by End-Users. In addition, an efficient spread of video content copies among the HBs is also considered. The HB clustering and cooperation that we consider in our proposal will also reduce the frequency of the video contents replacement in a single HB cache.

5.4.1.1. Video frequency and spread

The designed caching strategy aims to make available the video contents to the as possible users' demand while reducing delays and network bandwidth consumption. Towards this, meeting the users' demand is one of the main requirements of our replication strategy; therefore, the video popularity is one of the most important parameter considered in our design. The more popular is the video, the bigger is its frequency in the HB overlay. As explained previously, the video popularity is periodically tracked by the Service Manager. For our analysis, the video popularity fits zipf-like distribution. Considering M videos, ranked in the order of their popularity from 1 to M, video 1 being the most popular, the probability V_i to request the video of rank i is:

$$P_M(i) = \frac{1/_{i^\alpha}}{\sum_{k=1}^{M} 1/_{k^\alpha}} \quad (1)$$

where α represents the skew of the distribution. The number of HBs that will hold each video in each cluster is then directly related to the requesting probability of this video. Assuming N the number of participating HBs to the overlay, M_p the number of the most popular videos considered in the caching strategy, the number N_i of HBs that will cache the video V_i is:

$$N_i = ceil\left(\frac{P_M(i)}{\sum_{j=1}^{M_p} P_M(j)} * N * C\right) \quad (2)$$

where M_p is assumed large enough so that N_i never exceeds N.

It is important to replicate videos in the overlay according to their popularity; however, another important issue is how to better distribute them among the HBs. Towards an efficient distribution, we have used the K-means algorithm to determine the distance to respect between any two HBs that will cache the same video content. Given a set of points P, the k-means clustering problem seeks to find a set K of k centers, so that $\sum_{p\in P} d(p,K)^2$ is minimized. The k-means problem is NP-hard. However, there exist approximation algorithms that can be used.

In our case, HBs sub-clusters are calculated for each video following the order of their popularity. If we consider the video V_i, the parameters P and $K$of the K-means algorithm are represented respectively by the set of HBs that have enough place in their caches and by the N_i HBs that should hold it. Once the N_i sub-clusters for V_i are formed, the maximum distance between the couples of nodes under each sub-cluster c_i is calculated and the average of these distances d_i is assigned to V_i. d_i represents the distance that will separate any couple of HBs that will cache the video V_i. At the end of this process, a set of distances $\{d_1, d_2, \ldots, d_{M_p}\}$ are then assigned to the set of the M_p most popular videos $\{V_1, V_2, \ldots, V_{M_p}\}$. Since the HBs are considered stable components, this process should not happen frequently. It can be performed offline when the percentage of HBs that have joined or left the network exceeds a certain threshold, identified as a potential threat for system performance.

5.4.1.2. Video placement and replacement

In addition to the video caching frequency, another major design decision is to determine where and when the video has to be either cached or removed in order to be replaced by another one. Some works [184]-[185] considered that the cache replacement issue was less important and the main presented argument is that the caches are large enough to host most of the requested contents. However, with the explosive growth experienced by multimedia contents, that are known to require large storage and bandwidth resources, this argument cannot remain valid and the cache size becomes again a limit to build lucrative video streaming services. This limit is all the more important with the advent of User Generated Contents (UGCs) that bring an unlimited choice of videos to the Service Providers libraries, since enabling millions of users to be not only content consumers but also producers.

Before explaining our proposal to video placement and replacement issue, we would first present briefly an overview of cache replacement techniques that complete the caching techniques presented in section 2.3.2.1-C.

A. Representative cache replacement strategies

Taxonomy on web cache replacement strategies is given in [186]. The latter presents the following classes:

Recency-based strategies replace from the cache the objects that have not been requested for the longest time. Most of them are extensions of the well-known Least Recently Used (LRU) strategy such as LRU-Min [187], PSS [188], LRU-LSC [189], etc. The latter exploits the temporal locality of users' requests as a critical factor for objects purge from the cache.

Randomized strategies are the simplest approach to implement. The aim is to reduce the complexity of replacement strategies without sacrificing quality too much.

Frequency based strategies are based on the frequency replacement factor. These strategies are direct extensions of the Least Frequently Used (LFU) strategy. They exploit the fact that objects have different popularities that are reflected on the frequency of users' requests for these objects. The objects popularity can be maintained for either all objects or only the objects that are present in the cache.

Recency/Frequecy-based strategies are more complex strategies that mix recency and frequency to designate the object that will be removed from the cache. Example of such strategies are SLRU [190], LRU* [191], LRU-Hot [192], etc.

Function-based strategies evaluate a specific function that combines different factors such as recency, frequency, object size, cost of fetching the object, etc. Most of these factors are weighted. The weights are commonly deduced from usage traces.

B. Popularity-based video placement and replacement strategy

In this proposal, we aim to design an adaptive and cooperative popularity-based video placement and replacement strategy. The decision to place or replace the video in the cache of the requesting HB depends not only on the videos which are present in the cache of that HB but also on the contents cached inside its neighbor HBs. This set of neighbor HBs is determined according to the distances calculated in the previous section.

In order to make the model tractable, we have made the following assumptions:

- All HBs have the same storage capacity and upload connectivity, respectively represented by C and n (n is expressed in term of average simultaneous supported streams);
- We assume stable the HBs connectivity during all the requests for which we formulate the HB overlay hit ratio (percentage of requests satisfied by HBs);
- The users' requests are independent and fit a poisson model with a mean rate of λ requests per time unit;
- The videos length is represented by the mean duration of T time units;
- We consider that the HB overlay is already organized in L geographical clusters in which the proposed video replication solution is considered separately.

Considering then the j-th request for service/content consumption, the HB caches a video V_i if the following conditions are satisfied:

The HB is requesting the video V_i, which is estimated by the probability $P_M(i)$;

The closest HB that caches V_i is located at a distance greater than d_i from it, which means that, at request j, V_i has not yet been cached in the sub-cluster c_i of the requesting HB at a previous request to j.

The HB has either sufficient available cache capacity to store V_i or at least one of the C cached videos at its side is less popular than V_i. In this case, the latter will be replaced by V_i.

More formally, assuming $Pb_c(i,j)$ is the mean probability that V_i had been cached at a previous request m to j and had not been replaced till j, $Pb_c(i,j)$ can then be derived as follows:

- In both the initial phase where the HB cache is not full ($j \leq C$) and the case where V_i is one of the C most popular videos ($i \leq C$), V_i is cached if the following conditions are satisfied:
- V_i is requested at m (which is estimated by $P_M(i)$) ;
- and V_i has not been already cached in the cluster c_i of the requesting HB at a previous request to m (which is estimated by $\frac{1}{N_i} * \left(1 - P_c(i,m)\right)$);
- Otherwise, we derive first the probability that either V_i hasn't been cached at any previous request to j or V_i has been cached at a previous request m to j but has been replaced before j. In the first case, V_i either has not been requested $\left(1 - P_N(i)\right)$ or it has been requested but has already been cached in one of the HBs that belongs to the cluster c_i of the requesting HB $\left(\frac{P_c(i,m)}{N_i}\right)$. As well, the cache of the HB might also be full, meaning that the cache holds C more popular videos than V_i before the request m. This is estimated by: $\left(\frac{1}{N} * \frac{1}{i-1} \sum_{l=1}^{i-1} P_c(l,m)\right)^C$. In the second case, where V_i is cached but replaced before j, the HB at request j should have in its cache C more popular videos than V_i, which is estimated by $\left(\frac{1}{N} * \frac{1}{i-1} \sum_{l=1}^{i-1} P_c(l,j)\right)^C$.

To summarize, $Pb_c(i,j)$ is then:

$$P_c(i,j) = \begin{cases} \dfrac{1}{j-1} \displaystyle\sum_{m=1}^{j-1} P_M(i) * \frac{1}{N_i}\left(1 - P_c(i,m)\right) & if\left((i \leq C) || (j \leq C)\right) \\ \dfrac{1}{j-1} \displaystyle\sum_{m=1}^{j-1} \left(1 - \left(\left(1 - P_M(i)\right) + P_M(i) * \left(\left(\frac{P_c(i,m)}{N_i} + \left(\frac{1}{N} * \frac{1}{i-1} \sum_{l=1}^{i-1} P_c(l,m)\right)^C \right) + \left(\left(1 - \left(\frac{P_c(i,m)}{N_i} + \left(\frac{1}{N} * \frac{1}{i-1} \sum_{l=1}^{i-1} P_c(l,m)\right)^C\right)\right) * \left(\frac{1}{N} * \frac{1}{i-1} \sum_{l=1}^{i-1} P_c(l,j)\right)^C\right)\right)\right)\right) & otherwise \end{cases} \quad (3)$$

The probability that the video V_i is present in k HBs at request j is then:

$$P_c(i,j,k) = \frac{C_{j-1}^{k} * P_c(i,j)^k * \left(1 - P_c(i,j)\right)^{j-k-1}}{\sum_{l=0}^{N_i} C_{j-1}^{l} * P_c(i,j)^l * \left(1 - P_c(i,j)\right)^{j-l-1}} \quad (4)$$

inasmuch as $N_i * L$ is the maximum number of HBs that can hold V_i in the whole HB overlay.

5.4.2. Cost optimization

The efficiency of a replication strategy is a trade-off between the service scalability and quality from one side and the operating cost of the system from the other side. To be efficient, our strategy thus needs to satisfy the maximum simultaneous users' requests by the HB overlay while keeping quality at satisfying threshold for users. Indeed, making the videos reachable from the HB overlay reduces considerably the path delay, saves network and server bandwidth and consequently reduces packet loss. The goal is then to maximize the HB overlay hit ratio. Toward this goal, and considering on one hand the high number of video contents that constitute the VoD catalogues (especially with the advent of User Generated Contents UGC), and on the other hand, their extremely variant popularities, it is important to model the relationship between the number of the most popular videos that will be cached in the HB overlay and the system cost which is directly related to the HB overlay hit ratio. The way to formulate such equation thus constitutes a key modeling issue.

To this end, we have derived the HB overlay hit ratio for the case of finite HB overlay capacity N and finite number R of requests. Considering the hit ratio $h(HB,j)$ of the j-th request (where $1 \leq j \leq R$), the request is satisfied by the HB overlay if and only if: i) the requested video V_i is present at the HB overlay and ii) the streaming capacity of the cluster for V_i has not been exceeded. Assume $S_i = N_i * n$ the maximum parallel streams supported by the HB overlay for V_i, λ_i the average number of requests for V_i per T time units in the cluster with $\lambda_i = P_N(i) * \lambda * T/L$ and M_p the number of the most popular videos that will be cached in the HB overlay, the hit ratio $h(HB,j)$ of j -th request can be derived as follows:

Assume m the number of requests for V_i in the last T time units, l the number of HBs that hold V_i , the HB overlay is able to serve another request for V_i if the following conditions are satisfied:

l is large enough to serve the m previous request for V_i which means that $l \geq ceil\left(\frac{m+1}{n}\right)$;

and because a HB can cache multiple videos, these l HBs must have served at most $(l * n - 1)$ videos V_j that they have already cached (with $\left(j \leq M_p\right)$) .

Note that the two previous conditions are for the case that m does not exceed the maximum number of streams that can be supported by the HB overlay. For the case where m exceeds this value, the request cannot be satisfied by the HB overlay since it would be saturated for V_i.

Then, in the particular case where j is large enough that $\forall i : S_i < j$ (note that if $S_i < j$ then $N_i < j$), the hit ratio $h(HB,j)$ is derived as follows:

$$\mathrm{h(HB,j)} = \sum_{\mathrm{V_i}} \left(\left(\left(\mathrm{P_M(i)} * 1 - \sum_{\mathrm{m=S_i}}^{j-1} \mathrm{e}^{\lambda_i} \frac{\lambda_i{}^m}{\mathrm{m!}} + \sum_{\mathrm{m=0}}^{\mathrm{S_i-1}} \mathrm{e}^{\lambda_i} \frac{\lambda_i{}^m}{\mathrm{m!}} * \left(\sum_{\mathrm{m=0}}^{\mathrm{ceil}\left(\frac{\mathrm{m+1}}{\mathrm{n}}\right)-1} \mathrm{P(i,j,l)} + \left(\sum_{\mathrm{l=ceil}\left(\frac{\mathrm{m+1}}{\mathrm{n}}\right)}^{\mathrm{N_i*L}} \mathrm{P(i,j,l)} * \left(\sum_{\mathrm{s=l*n}}^{\mathrm{j-1}} \left(\mathrm{e}^{\lambda} \frac{\lambda^s}{\mathrm{s!}} * \left(\frac{\mathrm{l}}{\mathrm{N*s}} \sum_{\mathrm{t=j-s}}^{\mathrm{j-1}} \mathrm{h(HB,t)} \right)^{\mathrm{l*n}} \right)\right)\right)\right)\right)\right)\right) \quad (5)$$

More generally $\mathrm{h(HB, j)}$ can be derived for the three cases where V_{i} can respectively be part of the three complementary following sets of videos at request j:

- MP1: if $j < N_i'$: the number of requests received by the system is less than the number N_i' of maximum HBs that can hold V_{i} in the system (number of caches) then the maximum number of caches are $j - 1$;
- MP2: if $N_i' < j < S_i'$: the number of requests received by the system is greater than the number V_{i} of HBs that can hold V_{i} but less than the number S_i' of the parallel streams supported by the HB overlay for V_{i}, then the maximum parallel active streams supported by the HB overlay for V_{i} is $j - 1$;
- MP3: if $S_i' > j$: the number of requests received by the system is greater than the number $S_i'$of the parallel streams supported by the system for V_{i} which is the particular case described previously.

$\mathrm{h(HB, j)}$ is then the sum of the three previous cased since the request are independents and the cases are complementary :

$$
\begin{aligned}
\mathrm{h(HB,j)} = & \sum_{\mathrm{V_i \in MP1}} \left(\left(\sum_{\mathrm{m=0}}^{\mathrm{j-1}} \mathrm{e}^{\lambda_i} \frac{\lambda_i{}^m}{\mathrm{m!}} * \left(\mathrm{P_M(i)} * 1 - \sum_{\mathrm{m=0}}^{\mathrm{ceil}\left(\frac{\mathrm{m+1}}{\mathrm{n}}\right)-1} \mathrm{P(i,j,l)} + \left(\sum_{\mathrm{l=ceil}\left(\frac{\mathrm{m+1}}{\mathrm{n}}\right)}^{\mathrm{j-1}} \mathrm{P(i,j,l)} * \left(\sum_{\mathrm{s=l*n}}^{\mathrm{j-1}} \left(\mathrm{e}^{\lambda} \frac{\lambda^s}{\mathrm{s!}} * \left(\frac{\mathrm{l}}{\mathrm{N*s}} \sum_{\mathrm{t=j-s}}^{\mathrm{j-1}} \mathrm{h(HB,t)} \right)^{\mathrm{l*n}} \right) \right) \right) \right) \right) \right) \\
& + \sum_{\mathrm{V_i \in MP2}} \left(\left(\left(\sum_{\mathrm{m=0}}^{\mathrm{j-1}} \mathrm{e}^{\lambda_i} \frac{\lambda_i{}^m}{\mathrm{m!}} * \left(\mathrm{P_M(i)} * 1 - \sum_{\mathrm{m=S_i}}^{j-1} \mathrm{e}^{\lambda_i} \frac{\lambda_i{}^m}{\mathrm{m!}} + \sum_{\mathrm{m=0}}^{\mathrm{ceil}\left(\frac{\mathrm{m+1}}{\mathrm{n}}\right)-1} \mathrm{P(i,j,l)} + \left(\sum_{\mathrm{l=ceil}\left(\frac{\mathrm{m+1}}{\mathrm{n}}\right)}^{\mathrm{N_i}} \mathrm{P(i,j,l)} * \left(\sum_{\mathrm{s=l*n}}^{\mathrm{j-1}} \left(\mathrm{e}^{\lambda} \frac{\lambda^s}{\mathrm{s!}} * \left(\frac{\mathrm{l}}{\mathrm{N*s}} \sum_{\mathrm{t=j-s}}^{\mathrm{j-1}} \mathrm{h(HB,t)} \right)^{\mathrm{l*n}} \right) \right) \right) \right) \right) \right) \right) \\
& + \sum_{\mathrm{V_i \in MP3}} \left(\left(\left(\sum_{\mathrm{m=0}}^{\mathrm{S_i-1}} \mathrm{e}^{\lambda_i} \frac{\lambda_i{}^m}{\mathrm{m!}} * \left(\mathrm{P_M(i)} * 1 - \sum_{\mathrm{m=S_i}}^{j-1} \mathrm{e}^{\lambda_i} \frac{\lambda_i{}^m}{\mathrm{m!}} + \sum_{\mathrm{m=0}}^{\mathrm{ceil}\left(\frac{\mathrm{m+1}}{\mathrm{n}}\right)-1} \mathrm{P(i,j,l)} + \left(\sum_{\mathrm{l=ceil}\left(\frac{\mathrm{m+1}}{\mathrm{n}}\right)}^{\mathrm{N_i}} \mathrm{P(i,j,l)} * \left(\sum_{\mathrm{s=l*n}}^{\mathrm{j-1}} \left(\mathrm{e}^{\lambda} \frac{\lambda^s}{\mathrm{s!}} * \left(\frac{\mathrm{l}}{\mathrm{N*s}} \sum_{\mathrm{t=j-s}}^{\mathrm{j-1}} \mathrm{h(HB,t)} \right)^{\mathrm{l*n}} \right) \right) \right) \right) \right) \right) \right)
\end{aligned}
\tag{6}
$$

The average hit ratio over the R requests is:

$$h(HB,R) = \frac{1}{R}\sum_{j=1}^{R} h(HB,j) \quad (7)$$

Now that the hit ratio for the HB overlay is completely derived, let us consider the hit ratio of the CDN server. Since video content are large, we assume the CDN server has a limited capacity of C_S videos with C_S considerably larger than C. When the video cannot be streamed by the HB overlay, it is accessed from a CDN server if it is part of the C_S most popular videos and it was accessed at least once in the previous requests. The hit ratio of a CDN server $h(S,j)$ is then:

$$h(S,j) = \sum_{i=1}^{C_S} P_M(i) * \left(1 - \left(1 - P_M(i)\right)^{j-1}\right) \quad (8)$$

We have defined the hits ratio of both the HB overlay and the CDN server. Now let's consider the total cost of the system. Assuming the mean cost to access the video from a HB, a CDN server and a CP being respectively ct_{HB} ct_{CDN} and ct_{CP}, the total system cost of the system for the R requests is then:

$$TCost = R * \left(h(HB,R) * ct_{HB} + \left(1 - h(HB,R)\right) * \left(h(S,R) * ct_{CDN} + \left(1\_h(S,R)\right)ct_{CP}\right)\right) \quad (9)$$

ct_{HB} is the lowest of the three costs. Therefore, the total system cost is maximized when the HB overlay hit ratio is maximized. Since we are particularly interesting on the relationship between the system cost and the number of the most popular videos $M_P$that will be cached in the HB overlay, we gradually increase M_P starting at the initial value of $N/_C$. For each value, we calculate the total system cost $TCost$. The value M_P that optimizes the total system cost for the R requests is obtained when $TCost(M_p) < TCost(M_p + 1)$.

5.4.3. Performance Evaluation

This section provides the performance evaluation of the proposed on-line popularity-based video content replication combined with the two-level selection strategy explained and simulated in the previous sections, within the hybrid CDN-P2P content delivery platform.

5.4.3.1. Simulation environment

The simulation has been done under NS2. The hierarchical network topology in which we have simulated our cache strategy was generated with Brite in the Waxman model. The generated topology is fully conforming to the architecture presented. It consists of 200 routers with constant connectivity of 1Gb/s. The routers are placed in 5 ASs. In average, 5 HBs are attached to each router with a connectivity of 100 Mb/s. To simulate a real home network system, 1 to 4 clients are attached to each HB with a connectivity that varies from 0.5 to 100 Mb/s. We have placed, in each AS, a CDN server with a 2Gb/s connectivity and attached it to the router with the highest degree (highest number of routers linked to it). The topology also comprises Service Registry and Service Manager nodes that are also linked to the routers with the highest degree.

The video catalogue is composed of 10000 videos provided in three different formats:

- CIF for mobile terminals (352 x 288);
- 4CIF to provide TV-like quality (720 x 576);
- and 16CIF to provide High Definition video contents (1408 x 1152).

The probability to request the video in each format is respectively 1/5, 2/5 and 2/5. The video popularity fits the zipf distribution with a skew of 7.33.

Only the 1000 most popular videos are cached in the HB overlay. The others are only provided by CDN servers.

10000 requests are generated in a poison model during 30 min with an average of 400 requests per min, which implies situations of 2000 parallel streams. The video duration is set to 5 min. The clients linked to each HB request the VoD system 4 to 16 times with an average of 10 requests per HB.

All the HBs participate to the overlay and each of them can cache 5 videos and can provide in average 5 parallel streams. At the beginning of the simulation, all the caches of the HBs are empty.

5.4.3.2. Simulation results

In order to evaluate the efficiency of the proposed solution, it is also compared to a pure CDNs platform composed only with CDN server. The two scenarios are simulated in the same conditions with the same traffic model. The two approaches are compared according to the following metrics: The average server load and the data path delay.

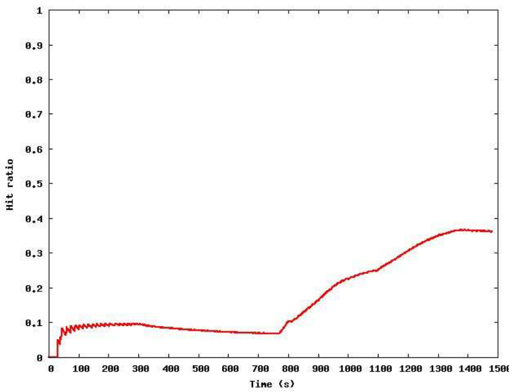

Figure 5.13. *HB overlay hit ratio.*

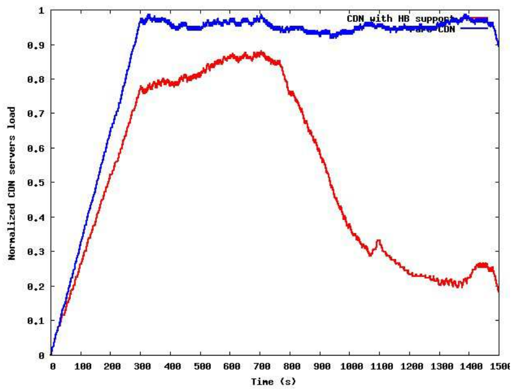

Figure 5.14. *Sim3 - Average CDN servers consumed bandwidth.*

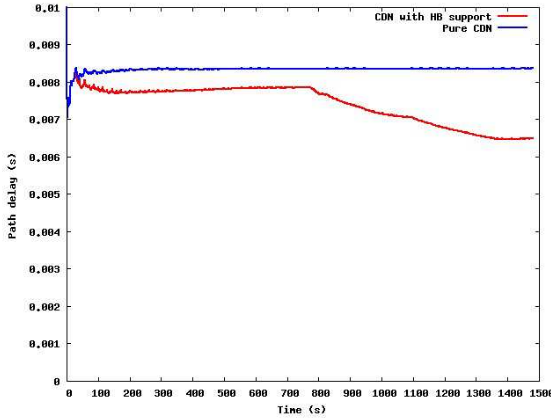

Figure 5.15. Sim3 - *Average data path delay*.

A. Overlay hit ratio

The overlay hit ratio, presented in Figure 5.13, represents the ratio of requests that are satisfied by HBs. As illustrated by the figure, in the initial phase of the simulation, the hit ratio is low. The reason is that the videos were not yet placed in the HBs caches. In our cache strategy, the videos are progressively placed in the HBs' caches while they are requested and consumed by End-Users. After almost three video durations (~900s), we can note that the hit ratio increases continuously until the end of the simulation. Thus, the more the HBs' caches get filled with videos, the highest gets the HB overlay hit ratio.

B. Server load

Figure 5.14 represents the average loads of CDN servers. We can note that our approach saves approximately 70% of the servers' load at the end of the simulation. In the initial phase of the simulation, the two loads are almost the same while as the HB caches get filled with videos, the difference between them increases. This result is in line with the increase of the HB overlay hit ratio and permits to show the benefit of our solution towards CDN servers' load, high impacting metric in investment strategies for SPs.

C. Path delay

Another system cost metric considered in the evaluation of our strategy is the path delay. This metric represents the average delay of RTP data packets of all the engaged video sessions in the system at a given moment. As expected and illustrated in Figure 5.15, the delay is always lower under our approach. However, the difference becomes even more significant as the simulation time increases and the videos are more and more distributed from the HBs' caches. This result is in line with the increase of the HB overlay hit ratio and permits to show the benefit of our solution towards the expected quality of the video service, high impacting metric in service adoption by End-Users.

We can also mention that our approach does not need an additional bandwidth for caching the served videos in the HBs' caches, since the caching process is performed while consuming the videos. Contrary to offline video placement approaches that require additional bandwidth for the placement of video contents in caches, the total bandwidth cost of our solution is exactly the necessary bandwidth for the service/content consumption.

5.5. Conclusion

In this chapter, we have proposed a context-aware video delivery solution to be used within the proposed Future Media Internet architecture. Major innovations consist on the design of (1) an enhanced service oriented architecture based on a virtual Home-Box overlay able to assist the video delivery infrastructure. The idea is to leverage the User Environment capabilities, as well as the Service/Content Provider features, in order to reach high-efficiency in delivering large-scale and high quality VoD services. The foreseen solution benefits from the self-scaling property of the P2P systems, while keeping the control and quality guarantees of managed infrastructures; (2) a popularity-based video replication strategy over the HB overlay that allows an efficient distribution of the video contents among the HBs cache as well as an adaptive and cooperative video content replacement strategy; (3) an efficient server selection strategy that copes with the video services requirements and the end user experience. Indeed, the proposed strategy combines multiple filters based on both context-aware policies and accurately measured metrics for selecting the most suitable server for each client request.

The evaluation of our proposal through simulation shows the gain of bandwidth, storage and service quality induced by such solution. As well, the obtained results clearly highlight the promising aspects of considering the Home-Box node in the video delivery chain.

Another advantage of the proposed video delivery scheme is its ability to be easily integrated in the next generation networks platforms since the HB is a component stemming from already deployed home devices (home gateways, set-top boxes).

One issue in the deployment of the proposed infrastructure and associated solution relies in its strong dependency to the last mile network resources. However, the progress deployment of Fiber-To-The-Home (FTTH) and the investments made by ISPs in the deployment of more powerful home gateways and devices encourage such a solution

Chapter 6

Conclusion

A large-scale context-aware and efficient distribution of multimedia services over Internet constitutes a hot research topic. Indeed, the exponential growth experienced by these services along with the diversification of the computing context of End-Users and their higher expectations for better experience, give rise to many demanding challenges.

The solutions presented in this thesis can be classified in two main proposals: (1) providing large-scale context-aware framework that enables personalized and context-adaptive multimedia services and (2) enhancing the current multimedia distribution platforms to achieve scalability while keeping a quality threshold at satisfying level for end users.

In more details, we have made the following contributions:

- Introduction of a novel architecture for Future Media Internet (within the ALICANTE - MediA Ecosystem Deployment through Ubiquitous Content-Aware Network Environments – European project) that aims to facilitate the deployment of an integrated Media Ecosystem within Future Internet, where all involved actors (Service/Content Creators and Providers, Network Operators/Providers and End-Users) collaborate for the efficient distribution of rich Media Services, and addressing the media distribution related challenges. For the research work presented in this thesis, the focus is put on the User and Service Environments to enable (1) high quality services and personalization towards a better experience from the End Users point of view and, (2) advanced media delivery features (including scalability, reliability, context-awareness, low cost, etc.) from the Service/Content Providers (S/CPs) perspectives.
- Based on the proposed architecture, we have built around the novel Home-Box virtual layer a context-aware framework that supports a wide range of multimedia services while achieving scalability. The two main basis of context-awareness: context modelling and context management were covered in two contributions based on the two mainly used models. Based then on markup schemes models, the first contribution aimed to provide a light weight context-aware multimedia framework with high responsiveness. In the second contribution, the context is modeled using ontology and rules for more expressiveness and formality. The aim was to prove the feasibility of such models in the field of large-scale multimedia services. Since the context model support reasoning, this function is incorporated in the context management to ensure context consistency

checking, high-level context inference and context-aware decisions triggering. In the two contributions a prototype of the context-awareness middleware has been implemented.

- Concerning the multimedia delivery, our work has been focused on the Video-on-Demand services. The proposed solutions aimed to enable an efficient, scalable and reliable context-aware VoD delivery. For this the widely used Content Delivery Network has been enhanced with the caching and connectivity capabilities of the HB virtual layers. The HB are the organized in a P2P overlay, ensuring thus better service scalability, availability, reliability and quality insurance. To optimize the benefit of the proposed platform, efficient anycast-based request redirection algorithm and adaptive online video caching and replacement based on the video popularity have been proposed. The promising obtained simulation results confirm the effectiveness of the proposed solutions.

The work achieved in this thesis constitutes only a part of a global solution for an efficient Internets multimedia services provisioning over Internet. Several perspectives have then been identified for ongoing and further work. For enhancing the context-awareness framework, we have identified the following directions:

- The proposed context model proposed in this thesis is dedicated to multimedia services and integrates a wide range of context information relevant to the latter services as well as the relations that can exist between them. However, a missing feature is our model is the representation of the confidence of the context information sources. This feature can be very important when determining the quality of context information. Conflicts could also be resolved using some probabilistic models based on this information.
- The realized context-awareness prototype developed in this thesis using ontology had only the aim to prove the feasibility of such approach within the field of multimedia services. Ongoing works will then be dedicated to fully implement our solution and to deploy it in the HB equipment in order to test it within a useful and relevant scenario.

Concerning the multimedia services provisioning the following works have been planed:

- In the video caching strategy proposed in this thesis, the granule is the video. The granularity can be affined, by exploiting the Scalable Video Coding (SVC) technologies, to a video layer. This will permit on one hand to ensure more availability for basic layer that are the most consumed and on the other hand to exploit the multi-sourcing techniques that will achieve more efficiency in the network level by distributing the load on different network links. However, the request redirection algorithm and mechanism have to be enhanced to support the latter distribution scheme.
- Since the proposed solutions have only been evaluated by simulation, the next step consists on implementing the proposed algorithms and integrating them in the HB equipment in order to achieve their evaluation in the large-scale testbed platform of the ALICANTE project. The latter is composed of a distributed multi-domain networking environment, divided in four autonomous (but cooperative) pilot islands, and based in different Partner locations. Namely, the **ALICANTE** system pilot will be geographically located at the PT Inovação headquarters in Aveiro – Portugal; at the FTB headquarters in

Beijing – China; at the UPB Campus in Bucharest – Romania; and at CNRS-LaBRI Campus in Bordeaux – France [124].

The proposed solutions focused on VoD services. A future contribution is dedicated to extend the proposed solution to other multimedia services and particularly the IPTV service (live streaming).An SVC-based IPTV provisioning can also benefit from the connectivity capabilities of the HB layer and the adaptation capabilities of the CAN virtual layer that is not addressed in this thesis as well as from the context-awareness feature designed in this thesis to achieve an adaptive IPTV provisioning that aims to enhance the user experience.

Appendix A

Résumé en français de la thèse

A.1. Introduction

Internet fait partie intégrante de notre vie quotidienne et les usages que nous en faisons sont multiples (recherche et échange d'informations, commerce, apprentissage, réseau sociaux, etc.). Un Cependant, cet usage et particulièrement marqué par l'enthousiasme des utilisateurs pour les services média (ex. Vidéo à la demande, IPTV, téléphonie sur IP, etc.). Ces services ont connu une véritable percée ces dernières années. D'après Cisco VNI [1], le trafic vidéo constituera 91% du trafic IP global d'ici 2014. De plus, les consommateurs sont de plus en plus exigeant sur la qualité des services demandés. Toujours d'après Cisco, 46% du trafic vidéo sur Internet va être en qualité HD/3D d'ici 2014.

De plus, avec la prolifération de terminaux mobiles (smartphones, tablettes, etc.) et les progrès rapides et importants que connaissent les réseaux sans fils et mobiles, l'attente des utilisateurs pour des services ambiants et de haute qualité est de plus en plus grande. En effet, d'un point de vue utilisateur seule la valeur du service importe, le terminal et le réseau utilisé pour y accéder ne sont que des moyens pour y accéder.

Si cette évolution dans l'usage de l'Internet semble très attrayante, elle impose également de lourdes contraintes sur son infrastructure en termes de montée à l'échelle, de fiabilité et de performance. Il parait donc aujourd'hui évident que l'Internet doit évoluer dans sa taille, ses fonctionnalités et sa capacité afin de permettre une création et une distribution plus efficace de services média riches et avancés.

Dans ce cadre, nous proposons dans cette thèse une nouvelle architecture, offrant un environnement collaboratif pour le partage et la consommation de services média. Cette architecture repose sur deux piliers : (1) un Environnement Service innovant incluant un nouvel overlay formé par des passerelles résidentielles dotées de nouvelles fonctionnalités ; et (2) un Environnement Réseaux amélioré avec notamment une sensibilité au contenu au moyen d'overlay virtuel dédiés au transport des flux média. Le design de cette architecture inclut également un Environnement Utilisateur, collaborant avec la Home-Box (HB), pour permettre un accès ubiquitaire aux services. Dans le cadre de cette thèse, nous nous somme intéressés aux Environnements Service et Utilisateur et plus particulièrement à la couche virtuelle, formée par les HB, partagée entre ces environnements. Nos propositions autour de cette couche virtuelle portent sur : (1) un Framework sensible au contexte permettant de services personnalisés et adaptatifs à grande échelle; et (2) une plateforme permettant une distribution efficace et à grande échelle des services média.

Dans le cadre de la première contribution portant sur la sensibilité au contexte, deux approches ont été proposées pour la modélisation et la gestion des informations de contexte. La première approche est basée sur les langages de balisage afin d'assurer une bonne responsivité des services. L'autre approche consiste à modéliser le contexte avec les ontologies afin d'assurer plus d'expressivité et de formalité. Notre but en traitant des ontologies est de prouver la faisabilité de cette approche dans le cas des services médias.

En ce qui concerne la distribution des services medias, la solution que nous proposons introduit les HB dans la chaine de distribution. L'idée est de tirer bénéfices des ressources (connectivité et espace de stockage) déployées dans les HB pour assister les serveurs dans leur tâche de distribution de contenus et améliorer ainsi la montée à l'échelle, la disponibilité et la fiabilité des services. Cette solution est accompagnée d'une stratégie de mise en cache et de remplacement des contenus vidéo et d'une stratégie de redirection de requête pour choisir pour chaque requête la meilleure HB ou le meilleur serveur pour la traiter.

Le reste de ce résumé est organisé comme suit. Nous commençons par présenter notre proposition d'architecture pour faciliter le déploiement de services média dans l'Internet du futur dans la section A.2. Puis nous présenterons la solution de sensibilité au contexte et particulièrement notre approche basée sur les ontologies dans la section A.3. Dans la section A.4, nous présentons nos propositions pour la distribution de contenus média sur Internet. Nous présentons ainsi une stratégie de mise en cache online et adaptative puis une stratégie de sélection du meilleur serveur sensible au contexte et basée sur le modèle anycast. Nous finirons ce résumé par une conclusion.

A.2. Architecture pour l'Internet du futur

Comme mentionné dans la section précédente, avec l'évolution qu'à connu l'usage d'Internet ces dernières années, met des contraintes très strictes sur l'Internet d'aujourd'hui. Il paraît donc évident que ce dernier doit évoluer à la fois dans ses fonctionnalités, sa capacité et sa taille afin de répondre aux attentes des utilisateurs finaux. En effet, différentes limites, liées au passage à l'échelle, à l'utilisation efficace du contexte des utilisateurs finaux, au support de la qualité de service, à la sécurité et à bien d'autres propriétés et fonctionnalités, ont été identifiées dans l'Internet d'aujourd'hui [3].

C'est dans ce cadre que nous proposons une nouvelle architecture pour l'internet média du futur. Cette architecture a pour but de faciliter le déploiement de services médias riches et de qualité et de rendre leur distribution plus efficace et optimale à bien des niveaux. Une description de l'architecture, de ses composants et de ses fonctionnalités, est présentée dans les sections à suivre.

A.2.1. Présentation générale de l'architecture ALICANTE

Dans cette section nous présentons une vue générale sur l'architecture que nous proposons pour l'Internet média du futur. Cette architecture, nommée ALICANTE, comprend plusieurs

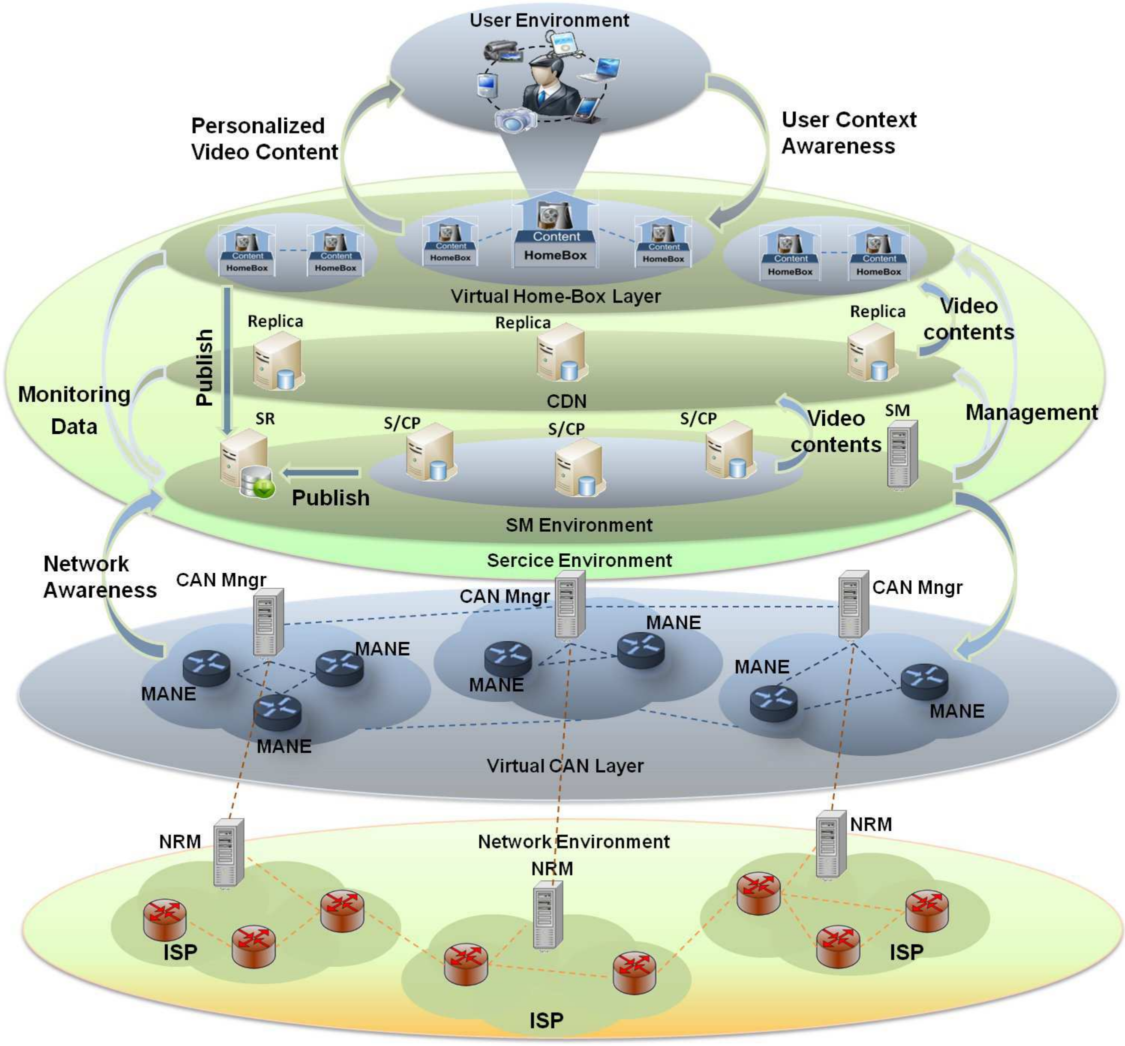

Figure A.1. *Vue générale sur l'architecture ALICANTE.*

environnements et couches comme illustré dans la figure A.1.

L'environnement Utilisateur offre aux utilisateurs finaux la possibilité de consommer des contenus et services médias personnalisés et sensibles au contexte mais aussi, de créer de nouveaux contenus et services et de les distribuer efficacement à d'autres utilisateurs. Ces nouvelles fonctionnalités s'appuient sur un profil utilisateur composé de paramètres statiques et dynamiques caractérisant l'utilisateur et son environnement. La partie dynamique de ce profile repose sur une évaluation continue de la QoE et des paramètres de QoS à différents niveaux. Une interface graphique conviviale et multiplateforme est également prévue afin de faciliter l'exploitation de ces nouvelles fonctionnalités par les utilisateurs finaux. La plupart des fonctionnalités de cet environnement sont déployées dans les terminaux des utilisateurs qui sont connectés à la passerelle résidentielle (Home-Box).

L'**environnement Service** est l'entité architecturale du système permettant de pallier aux différentes limitations rencontrées par les fournisseurs de services/contenus. Il permet à ces derniers d'offrir des services et contenus riches et personnalisés à travers :

- un cluster de serveurs médias (SP/CP servers);

- un registre (Service Registry) pour le maintien des métas-datas sur tous les services disponibles;
- des fonctionnalités de composition, de gestion, de monitoring et d'adaptation afin de supporter toutes les étapes du cycle de vie des services.

La couche virtuelle formée par l'interconnexion des HB a la caractéristique d'être commune aux deux environnements, Utilisateur et Service, et facilite ainsi la collaboration entre ces derniers. La HB peut être vue comme une évolution orientée média de la passerelle résidentielle. Elle intègre de nouvelles fonctionnalités telles que le caching, la gestion du contexte, la gestion, l'adaptation et la distribution des services est contenus et la gestion de la mobilité et de la sécurité. Les HB sont organisées en overlay offrant ainsi une infrastructure flexible qui peut être configurée pour une distribution des contenus médias en mode hiérarchique unicast ou multicast ou en mode pair à pair.

L'**environnement Réseau** fournit au couches supérieures, un réseau virtuel adaptable aux besoins des fournisseurs de contenus médias. Cet environnement inclut :

- La nouvelle couche virtuelle offrant, au-dessus de l'infrastructure IP, des réseaux virtuels sensibles aux contenus transportés (VCAN) sur un ou plusieurs domaines. Ces VCAN permettent, par des moyens de classification et d'autres traitements appropriés (monitoring, filtrage, routage, évaluation de la QoS, adaptation dynamique, sécurité, etc.), d'améliorer la distribution des contenus média. Cette couche virtuelle est gérée par des modules dédiés (CANMngr). Afin de faciliter le déploiement aisé et progressif de l'architecture, un CANMngr est déployé dans chaque domaine.
- L'infrastructure IP, sur laquelle sont instanciés les CAN via son gestionnaire de ressources réseaux IntraNRM à chaque requête du CANMngr. Les fonctionnalités avancées de l'architecture sont supportées au niveau réseau par Les MANE, des routeurs améliorés avec de nouvelles capacités matérielles et fonctionnelles.

A.2.2. Conformité de l'architecture ALICANTE avec l'effort de standardisation sur l'Internet du futur

L'internet du futur est un domaine emergent et plusieurs travaux et études sont en cours. Afin d'homogénéiser la recherche dans ce domaine, le FMIA-TT (Future Media Internet Architecture – Think Tank group) a définit comme modèle de référence une architecture Internet de haut niveau orientée média [4]. Cette architecture couvre la consommation, l'adaptation/enrichissement et la distribution des services et contenus médias dans l'internet du futur. L'architecture est constituée de plusieurs couches :

- *Un overlay d'application :* comprenant les applications qui peuvent utiliser plusieurs services et les contenus ;
- *Overlay d'informations (IO):* comprenant des nœuds intelligents et des serveurs dotés d'une connaissance sur les contenus et services distribués, la stratégie de distribution (caching, location, etc.) et sur la manière et de les conditions d'instanciation des réseaux de distribution ;

- *Overlay sensible aux contenus et services distribués (DCSAO)* : contenant des nœuds capables de plusieurs traitements sur les contenus et services ;
- *Infrastructure des fournisseurs de services et réseaux (SNI)*: représentant l'infrastructure offerte par les ISP.

Une comparaison de l'architecture ALICANTE et du modèle de référence du FMIA-TT est présentée dans le tableau qui suit.

FMIA-TT	***ALICANTE***
SNI	Environnement Réseau+Environnement Service
DSCAO	Couche virtuelle CAN
IO	Couche virtuelle composée par les HB
AL	Environnement Utilisateur + Environnement Service

Tableau A.1. *La corrélation entre FMIA et ALICANTE.*

En conclusion, l'architecture ALICANTE a été conçue en parfait accord avec le modèle de référence le plus récemment proposé pour les réseaux médias du futur. Pour plus de détail sur ce modèle et sur d'autres propositions d'architectures ainsi que leurs comparaison à l'architecture ALICANTE, consultez la section 3.2 de ce manuscrit.

A.3. Framework sensibilité au contexte dédié au déploiement service média sur Internet

Un aspect clé de l'informatique ubiquitaire est la sensibilité au contexte. L'idée et de récupérer les informations de contexte grâce à des capteurs, de les analyser, de les comprendre et d'adapter en conséquence le comportement des services afin de satisfaire au mieux les attentes des utilisateurs finaux. L'intérêt porté à la sensibilité au contexte est d'autant plus important avec la popularité que connaissent les services médias sur Internet. Cependant, cette popularité et les ressources importantes nécessaires à leur adaptation induisent également des contraintes de passage à l'échelle dont il faut tenir compte lors du design d'un Framework dédié à ce type de services.

A.3.1. Etat de l'art

A.3.1.1. Définition et modélisation du contexte

Le contexte est un terme générique qui jusqu'à aujourd'hui, n'a pas encore de définition standard et claire. Cela dit, dans le cadre de l'informatique ubiquitaire, plusieurs définitions ont été données au contexte dans la littérature. Une approche très utilisée dans la définition du contexte, est celle qui consiste à énumérer les informations qui peuvent y être incluses [5]-[10]. Une autre approche consiste à définir le contexte en se servant de synonymes tels qu'environnement ou situation tel que dans [7] et [11]. Dans d'autres travaux le contexte est défini de manière plus formelle [12]-[14]. Mais la définition la plus générique, largement adoptée par la communauté de recherche, est celle de Dey [15], [16] qui définit le contexte comme toute information pouvant servir à caractériser la situation d'une entité. Une entité pouvant être une personne, un lieu ou un objet considéré comme pertinent pour l'interaction entre un utilisateur et une application, y compris l'utilisateur et les applications.

"Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves". [15] [16]

Dans [16], Dey dérive aussi la notion de situation et la définit comme étant la description des états d'entités pertinentes. Cette notion de situation a donc été reprise dans la plupart des systèmes sensibles au contexte.

Un des fondements de la sensibilité au contexte est la modélisation du contexte. En effet, afin de les traiter automatiquement, les informations de contexte doivent être décrites de manière formelle dans un modèle de données.

Différents modèles on été utilisés dans la littérature, à savoir :

- les modèles attribut-valeur décrit dans [21-22] ;
- les modèles basés sur les langages de balisage tels que GUP [23-25], MPEG-7 [29] et MPEG-21 [30], CC/PP [31] et bien d'autres encore ;
- les modèles graphiques utilisés dans [37-39];
- les modèles orientés objet tel que l'UDC (User Data Convergence) [40-43];
- les modèles basés sur les règles utilisés dans [44-45] ;
- et enfin les modèles basés sur les ontologies souvent basés sur OWL (Ontology Web Language) [49]. On peut citer comme exemples : CONON [50], COBRA-ONT [51] et SOUPA [52].

Des études comparatives entre les différents modèles montrent que les modèles basés sur les ontologies sont ceux qui satisfont le mieux les exigences liées à la modélisation du contexte. Ce modèle est aussi intéressant de par son lien avec le web sémantiques.

A.3.1.2. Systèmes et Framework sensibles au contexte

La recherche sur sensibilité au contexte date du début des années 90 avec l'émergence l'informatique mobile. Deux travaux pionniers dans ce domaine sont le système de badges actifs [60] conçu à Olivetti Research Lab et la carte active [61] de Xerox PARC. Depuis, une multitude de systèmes sensibles au contexte ont été proposés. Ces systèmes sont souvent basés sur une information de contexte importante qu'est la localisation tel que dans [62],[63]. Cependant, comme le contexte et beaucoup plus que l'information de localisation, certain travaux, tels que dans [17] et [64], l'ont combiné avec d'autres informations telles que la date et l'heure, l'activité et les centres d'intérêt de l'utilisateur afin de concevoir des systèmes plus adaptatifs. Ces systèmes sont souvent propriétaires et dépendent des applications pour lesquelles ils sont conçus.

Pour plus de flexibilité et d'évolutivité, d'autres travaux, tels que GUP [23], UDC [40], Context Toolkit [69,15], SOCAM [70] and COBRA [51],C-CAST [71] se sont focalisés sur la conception de Framework permettant un prototypage plus rapide et plus facile d'applications sensibles au contexte.

Ces Framework offrent souvent les fonctionnalités suivantes :

- la capture des informations de contexte par des capteurs physiques, logiques ou virtuels ;
- une API permettant de récupérer les informations de contexte de manière transparente ;
- la modélisation et le stockage des informations de contexte ;
- le traitement des informations de contexte qui consiste en l'interprétation des informations bas niveau et dans certains systèmes l'inférence de nouvelles informations de plus haut niveau ;
- et enfin, la distributions de ces informations de contexte aux différentes applications afin qu'elle adaptent leur comportement en conséquence.

A.3.2. La sensibité au contexte dans l'architecture ALICANTE

Comme mentionné dans la section précédente, une des fonctionnalités essentielles de l'architecture ALICANTE est la sensibilité au contexte qui est à la base de la personnalisation et de l'adaptation des services média. Cette fonctionnalité est principalement implémentée par l'overlay des HB afin d'assurer la propriété de passage à l'échelle.

La HB est donc améliorée avec les fonctionnalités suivantes :

- le traitement et la gestion du contexte provenant des différents environnements de l'architecture ;
- la personnalisation et l'adaptation des services et contenus média afin de pallier à la grande hétérogénéité du contexte ;
- la gestion des sessions des utilisateurs et abstraire ainsi certaine tache de signalisation aux serveurs du SE telle qu'assurer une mobilité de session transparente entre les différents terminaux de l'utilisateur ;
- et enfin, pour garder le système sous contrôle, des programmes de monitoring sont installés dans tous les nœuds du système.

Le déploiement de ces fonctionnalités de manière distribuée dans les HB a différents avantages :

- relever les terminaux légers de certaines taches contraignantes en ressources comme l'inférence ;
- assister les serveurs de l'Environnement Service pour des fins de passage à l'échelle ;
- intégrer les services média sur Internet aux services domotiques déployés dans le réseau domestique ;

- la HB est un élément partagé et facilement atteignable par les différents environnements de l'architecture. Elle consiste donc un élément idéal vers lequel les informations de contexte peuvent converger.

A.3.3. Framework basé sur les ontologies

Dans cette contribution, nous proposons (1) une ontologie regroupant une grande variété d'informations de contexte et (2) un Framework permettant aux services média sur Internet de bénéficier des fonctionnalités de sensibilité au contexte et d'adaptation grâce à un middleware déployé dans les HB. Basé sur l'inférence, ce middleware permet d'acquérir les informations de contexte de différentes sources, de les intégrer à la base de connaissance hébergé dans la HB, de raisonner sur cette base pour inférer des situations et enfin de fournir l'Environnement Service avec les informations de contexte pertinentes qui vont être utilisées dans le phase de découverte, d'invocation et d'adaptation des services

A.3.3.1. Modèle de contexte basé sur les ontologies

La représentation du contexte dans un modèle de données est nécessaire à son traitement par des agents logiciels. Parmi les modèles de représentation de contexte, les ontologies devraient jouer un rôle majeur dans le domaine de la sensibilité au contexte. La définition la plus citée des ontologies, donnée par Tom Gruber [46], définie les ontologies comme une spécification formelle et explicite d'une conceptualisation partagée d'un domaine d'intérêt. Les ontologies offrent donc un moyen de représenter formellement la sémantique des informations de contexte en termes de concepts et de rôles. Modélisées avec les ontologies, les agents logiciels ne font pas qu'interpréter ces informations mais peuvent également les comprendre et en tirer des conclusions comme le ferait un humain.

Dans le modèle que nous proposons, les informations de contexte sont classées comme statiques ou dynamiques. Le contexte dynamique regroupe les informations qui caractérisent la session. Ces informations sont amenées à changer fréquemment tout au long de la session. Les informations de contexte statiques sont également amenées à changer mais beaucoup mois fréquemment.

Pour notre modèle, nous avons identifié six entités génériques que nous considérons communes à tous les domaines. A savoir, l'utilisateur, le terminal, le réseau, le service, l'environnement et la session. Les cinq premiers concepts sont classés comme statique et le dernier comme dynamique. Chacune de ces entités est ensuite décrite et étendue avec des concepts et relations additionnels liés au domaine des services média. L'entité utilisateur est alors décrite avec plusieurs profiles décrivant des informations d'ordre général sur l'utilisateur, sa souscription au service, ses contacts, son affiliation, ses informations d'authentification, et enfin ses préférences ; le terminal est décrit avec des informations caractérisant le matériel qui le compose et les logiciels qui y sont installés et ainsi de suite.

L'ontologie est donc à deux niveaux : une ontologie, de haut niveau, composée des concepts générique et des relations entre eux et une ontologie dédiée aux services média qui étend la première pour un meilleur support de ce genre de services par le Framework.

Pour plus d'expressivité, le modèle inclut aussi des règles. Ces dernières permettent une inférence qui débouche sur le déclenchement de décisions de personnalisation ou d'adaptation des services. En effet, ces décisions reposent sur les informations de contexte de haut niveau, qui sont obtenues par un raisonnement basé sur la logique descriptive exprimée dans l'ontologie et les règles.

Concernant le langage, notre proposition d'ontologie est basée sur la recommandation du W3C pour le web sémantique, à savoir OWL (Ontology Web Language)[49]. Pour des raisons d'interopérabilité, nous nous sommes basés sur OWL-S [148] pour la modélisation de l'entité service et de DL-safe SWRL [149] pour la représentation des règles.

A.3.3.2. Middleware pour la sensibilité au contexte

En se basant sur le modèle de contexte et en collaboration avec les autres composants en charge de la gestion et de l'adaptation des services, le middleware en charge de la sensibilité au contexte permet un accès ubiquitaire aux services pour les utilisateurs du réseau domestique. Il constitue donc une entité intermédiaire ente les producteurs du contexte pouvant être situés dans les différents environnements de l'architecture et les consommateurs du contexte. Le middleware est composé de composants fonctionnels suivants :

- le *Context Formatter* en charge de la validation et de l'intégration des informations de contexte dans l'ontologie, abstrayant ainsi l'hétérogénéité de leurs sources ;
- la base des connaissances comprenant l'ontologie décrivant le contexte et les règles ;
- le *Context Manager* en charge de maintenance de la base de connaissance et du processus de souscription/notification à base d'événement qui permet de fournir les consommateur du contexte avec les information voulue.
- et le moteur d'inférence qui raisonne sur la base de connaissance pour inférer de nouvelles informations de contexte de haut niveau qui vont déclencher des décision sur la personnalisation et l'adaptation des services

A.3.4. Implémentation et evaluation du middleware

Un prototype du middleware précédent a été implémenté en Java 1.6. Nous avons utilisé les API Protégé-OWL et SWRLTab pour l'interfaçage et la manipulation de l'ontologie modélisant le contexte ainsi que pour générer des événements en configurant des *listeners*. L'inférence est réalisée par le moteur d'inférence Jess qu'on relié à SWRL Tab. Le middleware a été ensuite déployé dans une HB avec un processeur Intel (R) Core(TM)2 Duo et 4GB de RAM.

Avec un ficher OWL, décrivant l'ontologie, composé de 40 classes, 91 propriétés, 327 instances et 17 règles. Nous avons estimé le temps de chargement de l'ontologie à environs 903ms et le temps nécessaire à l'inférence à 350ms.

Les résultats montrent que le temps nécessaire à l'inférence reste acceptable dans le cas de services média vue que les processus de monitoring et d'adaptation de ces services ont habituellement des périodes plus longues. Le temps de chargement quant est plutôt long, mais comme ce processus est réalisé à la phase d'initialisation du système, il n'entrave pas la réactivité de ces services.

A.4. Solutions pour la distribution des contenus medias

Les plateformes de distribution de contenus média sur Internet peuvent être classées en deux catégories : (1) les plateformes gérée intégrant des fonctionnalités permettant d'assurer la QoS et (2) les plateformes non gérées dites best effort avec aucun contrôle sur l'approvisionnement en ressource et par conséquent sur la QoS/QoE.

Dans la première catégorie, les solutions sont souvent propriétaires malgré l'effort considérable de plusieurs groupes de standardisation tels que l'ITU-T, l'ETSI TISPAN IPTV, le 3GPP MBMS, etc. En effet, la plupart des opérateurs sont aujourd'hui capables de fournir des services triple-play (téléphonie, accès Internet et IPTV), mais en s'appuyant sur des solutions propriétaires fermées qui ne sont accessible que depuis leurs propres réseaux. Un exemple de ce type de plateformes est Microsoft Mediaroom IPTV Platform [75], utilisé par plus de 40 opérateurs leadeur de par le monde. Cependant, afin de permettre un déploiement plus facile et une meilleure interopérabilité, plusieurs standards pour l'IPTV, tels que TISPAN IPTV [79]-[80], ITU-T FG IPTV [81]-[82] ou la spécification de Open IPTV Forum [83] ont émergé. Le but étant une intégration aisée de l'IPTV dans des architectures de types NGN.

La deuxième catégorie quant à elle inclut essentiellement les CDN et les réseaux pair-à-pair. Ces deux plateformes ont l'objectif commun de passage à l'échelle. Cependant leurs designs différent significativement ; et cette différence se reflète directement sur la performance, la charge, et la stratégie de réplication à adopter.

Le caching et la réplication sont deux techniques communément utilisées dans les plateformes de distribution de contenus de type *best effort*. La mise en cache des contenus les plus populaires à des nœuds stratégiques sur la bordure du réseau, près des utilisateurs finaux, permet un accès plus rapide tout en optimisant la consommation de bande passante au niveau réseau et serveurs. D'un autre coté, la réplication permet de maintenir les contenus distribués sur ces différents nœuds sous le contrôle des fournisseurs de services et contenus.

Les avantages de ces techniques sont donc multiples :

- D'un point de vue réseau, ces techniques permettent de réduire le trafic et par conséquent, d'éviter des situations de congestion et d'améliorer ainsi la performance des réseaux de transport ;
- D'un point de vue fournisseurs de services et contenus, ces techniques réduisent la charge sur leurs serveurs et améliorent la disponibilité, la fiabilité et la réactivité de leurs services. Elles permettent également de réduire les coûts de distribution ;
- Et enfin, d'un point de vue utilisateur, elles permettent un accès plus rapides aux services et ainsi une meilleure expérience.

Cependant, l'optimisation des avantages de ces techniques entraine plusieurs problématiques. À savoir, le placement des caches, la coopération entre caches, quels contenus doivent être mis en cache, quand et dans quels caches les mettre, quand les remplacer, quel cache doit répondre à une requête d'un utilisateur, etc. Dans cette thèse, nous nous sommes particulièrement intéressés à deux problématiques : la redirection des requêtes des utilisateurs et l'élaboration d'une stratégie de mise en cache.

A.4.1. Modèle de redirection de requêtes sensible au contexte basée sur l'anycast

Dans cette section nous présentons notre modèle de redirection de requêtes. L'innovation de notre approche consiste en un algorithme adaptatif de redirection de requête à deux niveau considérant à la fois le contexte coté utilisateur, serveurs et réseaux de transport dans la sélection du meilleur serveur pour traiter une requête utilisateur. En effet, notre approche repose sur un premier filtrage basé sur le contexte utilisateur puis un second basé sur distance réseau qui combine deux métriques : le délai de bout en bout et la charge des serveurs. Le serveur sélectionné n'est pas forcément le serveur le plus proche en termes de nombre de sauts ou de délais. Par contre, c'est le serveur qui offrira la meilleure QoE possible.

A.4.1.1. Anycast – état de l'art

L'anycast a été défini à l'origine au niveau réseau [161] comme un service qui permet à un hôte de se connecter à un membre, probablement le meilleur, d'un groupe d'hôtes servant la même adresse anycast. Le groupe est formé de ressources offrant un même service.

Des challenges techniques inhérents à la sélection du meilleur serveur ont fait que l'anycast au niveau réseau ne fut jamais déployé. En effet, ce modèle présentent plusieurs limitations, telles que l'allocation d'une plage d'adresses pour l'anycast, la nécéssité de routeurs supportant l'anycast, la non possibilité de prise en compte d'informations de contexte et la nature sans états d'IP, rendent son intégration, dans des infrastructures existantes, difficile. Cependant, quelques recherches ont été basées sur ce modèle. [166] et [167] se sont concentrés sur le design d'algorithmes de routage se basant sur des routeurs actifs. [168] et [169] proposent des infrastructures basées sur des proxys.

Les limitations de l'anycast tel qu'il est défini au niveau réseau ont poussé à définir l'anycast au niveau applicatif. Dans [101]-[108] et [102], les auteurs définissent l' anycast au niveau applicatif comme un service permettant de mapper un nom de domaine anycast à une ou plusieurs adresses IP en se servant d'un nœud intermédiaire. Ce genre d'approches supporte bien la délivrance de services internet distribués et ne requière pas de changement dans les infrastructures actuelles. Quelques papiers tels que [193] et [170] explorent ce sujet dans le cadre de la distribution de services média.

A.4.1.2. Mécanisme de redirection de requêtes

Notre mécanisme de redirection de requêtes a pour but d'offrir aux utilisateurs finaux un service de distribution de contenus vidéos efficace et transparent. Ce mécanisme est basé sur le modèle anycast. Le Service Manager SM et le Service Registry SR hébergés par l'Environnement Service collaborent pour mapper l'adresse anycast affectée au service à l'adresse du meilleur serveur pouvant répondre à une requête utilisateur.

La communication entre les différents acteurs (clients, serveur, SM et SR) se fait via SIP (Session Initiation Protocol) [171], un protocole de signalisation dédié au contrôle de sessions média. Le dialogue passe par différente phases :

- **Publication des services** : nous considérons dans notre travail que les serveurs peuvent avoir des contextes hétérogènes et fournir des contenus avec des caractéristiques différentes. Chaque serveur doit donc publier son contexte et la liste des contenus qu'il peut fournir avec leurs caractéristiques via une requête *SIP Publish.* Ces informations sont maintenues au niveau du SR.
- **Sondage des serveurs** : pour maintenir les informations de contexte dynamiques des serveurs comme leurs charges, ces derniers peuvent être sondés de façon proactive ou réactive. Cependant, le sondage proactif n'assure pas la validité de l'information au moment de son utilisation à moins de minimiser la période du sondage. Ce qui résulterait en une charge plus importante sur le réseau et les serveurs. De plus, une des métriques utilisées dans notre cas est le délai entre le serveur et le client. On a donc besoin de transmettre au serveur l'adresse du client pour obtenir cette métrique pour une requête donnée. Le Sondage des serveurs se fait via la requête *SIP Option*.

L'établissement d'une session vidéo se fait comme suit :

- Nous supposons que les serveurs auront préalablement publié leurs contenus selon le processus de publication décrit ci-dessus auprès du SR ;
- Le client essayant de d'accéder au service vidéo initie une session avec une requête SIP Invite à l'adresse anycast en spécifiant la vidéo demandé mais aussi ses informations de contexte.
- La requête du client est alors routé au SM qui lui doit envoyer une réponse *SIP temporaire 183 ;*
- Le SM entre alors dans de processus de sélection du meilleur serveur en faisant un premier filtra
- Ge en collaboration avec le SR qui sera basé sur le contexte utilisateur puis le résultat est filtré dans une seconde étape, en se basant sur un métrique réseau obtenue grâce au sondage des serveurs, afin de sélectionner le meilleur serveur ;
- La requête de l'utilisateur est ensuite transférée à ce serveur qui répond au client avec une réponse *SIP 200 OK* afin de lui confirmer l'établissement de la session et commence le streaming du contenu demandé.

A.4.1.3. Algorithme de sélection du meilleur serveur

Comme expliqué auparavant notre approche de distribution de contenus média repose sur la sélection du *meilleur serveur* parmi un groupe anycast. Par *meilleur serveur*, nous sous entendons le serveur non surchargé qui répond au mieux aux contraintes liées au contexte du client ainsi qu'au réseau de transport, toujours dans un but d'offrir aux clients la meilleure expérience possible. Afin de permettre ceci, la sélection du meilleur serveur passe par deux phases : une première phase où les serveurs sont filtrés en se basant sur des règles, puis une seconde phase où le meilleur serveur est sélectionné sur la base d'une métrique réseau. Ce processus de sélection à deux niveaux est décrit dans ce qui suit.

A.4.1.4. Filtrage base dur les règles

Le SR maintient la liste de tous les contenus vidéo offerts par le service de VoD. Pour chaque contenu, il maintient la liste des réplicas qui l'hébergent et les métas donnés le décrivant.

Au moment où le client demande un contenu, il envoi dans sa requête ses informations de contexte (la bande passante dont il dispose, les caractéristiques de sa machine, etc.) ainsi que ses préférences. Ces informations sont encapsulées dans le corps de la requête SIP Invite qui initie le service. A la réception de la requête par le SM, ces informations sont envoyées au SR qui va filtrer les contenus qu'il maintient, à base de règle prédéfinies, pour n'en garder que ceux qui répondent bien au contexte de l'utilisateur. Des exemples de couple d'informations pouvant être mappées sont : la résolution de l'écran du terminal utilisateur et la résolution de la vidéo, la bande passante dont dispose l'utilisateur et le bitrate de la vidéo, etc. La liste des serveurs hébergeant ces contenus est alors retournée au SM.

A la fin de cette étape la liste des serveurs hébergeant des contenus qui correspondent bien au contexte de l'utilisateur est alors sélectionnée.

A.4.1.5. Filtrage basé sur des métriques réseaux

L'objectif de cette étape est de sélectionner parmi la liste des serveurs construite à l'étape précédente le serveur (le meilleur) qui traitera la requête de l'utilisateur. Cette sélection est basée sur l'évaluation d'une distance combinant des métriques réseaux. Le choix de ces métriques est dépendant du service traité et l'efficacité de ce choix à des conséquences fortes sur la qualité ressentie au niveau utilisateur.

Comme le service traité dans ce travail est le service de VoD, connu pour être très sensible aux pertes et au délai, l'objectif premier de notre processus de sélection est d'éviter la congestion et ce à tous les niveaux. Dans la phase précédente nous avons traité le problème de congestion au niveau du client en considérant ses informations de contexte. Dans cette phase, nous traiterons de la congestion au niveau serveur et au niveau réseau. Par conséquent, le filtre définie dans cette phase est basé sur une fonction pondérée impliquant deux paramètre : la charge au niveau du serveur et le délai entre de serveur et le client. La combinaison de ces deux métrique permet d'éviter la congestion (1) au niveau serveur en évitant la surcharge des serveurs, et (2) au niveau réseau en essayant de minimiser le délai courant entre le serveur et le client. L'évaluation de cette fonction se fait comme suit au niveau de chaque serveur :

$F(A_c, Rbr_c)$ {

  $d_{sc} = delay(A_c)$;

  <u>if</u> $\left(\left(\sum_{i=1}^{n} Rbr_i + Rbr_c\right) \big/ br < 1\right)$ <u>then</u> {

    <u>return</u> d_{sc} ;

  }<u>else</u> {

```
        return(α * d_sc * (( Σ_{i=1}^{n} Rbr_i + Rbr_c ) / br ));
    }
}
```

$$\text{return}\left(\alpha * d_{sc} * \left(\left(\sum_{i=1}^{n} Rbr_i + Rbr_c\right)\Big/ br\right)\right);$$

Les serveur de la liste résultant de l'étape précédente sont alors sondés afin de récupérer pour chacun d'eux l'évaluation de la fonction F. Afin de pouvoir effectuer cette évaluation les paramètres de la fonction *F*, A_c représentant l'adresse du client et Rbr_c représentant le bitrate demandé par ce dernier, sont fournis aux serveurs sondés dans la requête *SIP Option*. D'un autre coté chaque serveur doit également réévaluer sa charge à chaque initiation ou terminaison de session. La charge courante du serveur est évaluée comme suit :

$$load = \sum_{i=1}^{n} Rbr_i \Big/ br \quad (1)$$

Où n est le nombre de sessions actives auxquelles il participe, Rbr_i est le bitrate de la session i and br est le bitrate du lien montant du serveur.

Nous remarquerons que tant que le serveur n'est pas surchargé, seule la métrique délai d_{sc} est considérée dans l'évaluation de la fonction F afin de toujours sélectionner le serveur le plus proche, mais dès que le serveur est en situation de surcharge, la priorité données aux deux métrique est alors inversée afin d'éviter les serveurs surchargés.

Après réception des distances évaluées par les serveurs sondés, le meilleur serveur est sélectionné comme suit :

$$F_s(A_c, Rbr_c) = \min_{1<i<m}\left(F_i(A_C, Rbr_c)\right) \quad (2)$$

Où m est le nombre des réponses reçues. On notera que m n'est pas nécessairement égal au nombre de serveurs sondés. Si le timer limitant le temps du processus de sondage expire, le meilleur serveur est sélectionné parmi ceux qui ont répondu.

A.4.1.6. Evaluation de la stratégie de sélection

Dans le but d'évaluer l'efficacité de notre stratégie de mise en cache nous l'avons comparée à la sélection aléatoire. Les deux scénarios ont été simulés dans le même environnement sous NS2 avec la même topologie que nous avons généré avec le générateur de topologie Brite. A cette topologie nous avons attaché 500 serveurs. Nous avons également utilisé le même modèle de trafic et le même catalogue constitué de 10 vidéos de 5min chacune. Les vidéos peuvent avoir des caractéristiques différentes d'un serveur à un autre. Nous les avons particulièrement différenciées selon trois critères : la résolution, le bitrate et la langue.

Les deux scénarios ont ensuite été comparés selon différente métriques comme indiqué dans ce qui suit.

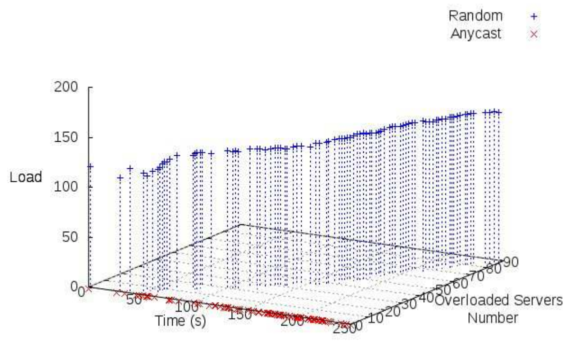

Figure A.1. *Nombre et surcharge moyenne des serveurs en situation de surcharge.*

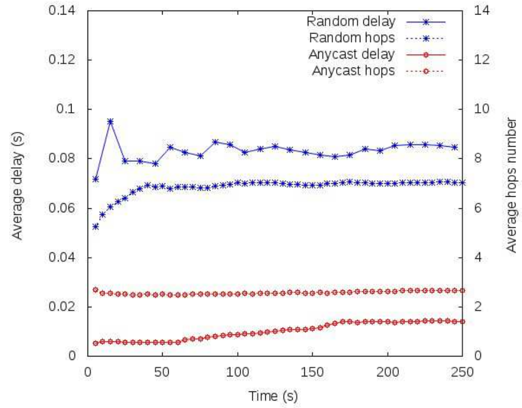

Figure A.4. *Distance moyenne des chemins empruntés (délai et nombre de sauts).*

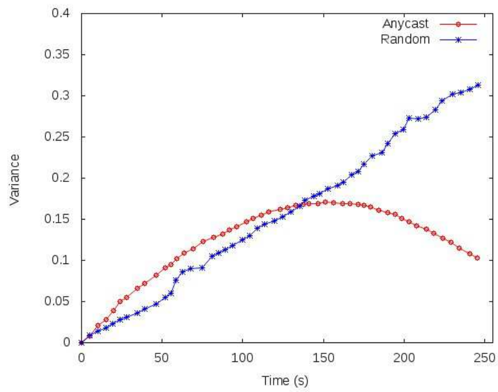

Figure A. 3. *Variance de la charge des serveurs.*

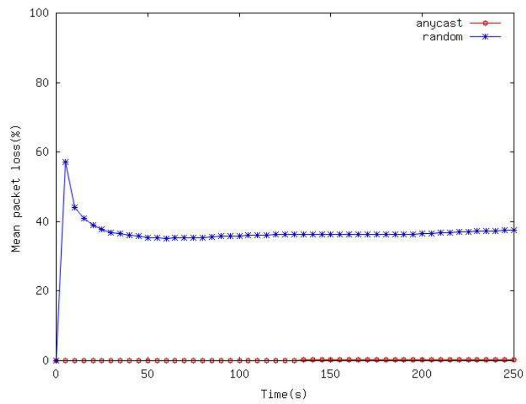

Figure A.2. *Perte moyenne sur toutes les sessions actives.*

Métriques	**Anycast**	**Aléatoire**
Correspondance au contexte utilisateur	100%	33.7%
Charge supplémentaire due à la signalisation	74.59 KB	1.72 KB
Rendement du service	219.8 MB	203.14 MB
Connexion inter-AS	0%	78.5%
Temps de réponse moyen	1.05 s	0.02 s

Tableau A.2. *Quelques résultats additionnels.*

Comme illustré dans la figure A.2, aucune situation de surcharge n'est enregistrée avec notre stratégie de sélection contrairement à l'autre scénario où 17% de surcharge en moyenne sont enregistrés au moment où toutes les sessions sont actives.

C'est important d'éviter des situations de surcharge mais c'est tout aussi important de bien répartir la charge sur les différents serveurs. La figure A.3 reflète bien la manière dont la répartition est effectuée dans notre processus de sélection. On remarque bien qu'il y a deux phases dans la simulation. Dans un premier temps, dans un intervalle [0s ,137s], la sélection aléatoire connait une variance moins importante que dans notre sélection. Cela s'explique par le fait que dans notre sélection nous donnons la priorité à la métrique délai tant que les serveurs ne sont pas proches d'une situation de surcharge. Puis dans un deuxième temps [137s, 250s], la variance continue à augmenter avec la sélection aléatoire alors qu'elle diminue sous notre sélection. Ceci est le résultat de l'inversement de priorité prévue dans notre sélection à l'approche d'une situation de surcharge.

Une métrique, très importante également dans le cas de services VoD, est le niveau de perte. Nous remarquons (cf. Figure A.5) que les pertes sont presque nulles sous notre sélection Durant toute la durée de simulation. Contrairement à la sélection aléatoire qui approche les 35% durant presque toute la durée de simulation. L'efficacité de notre approche est le résultat de la prise en conte de la congestion à tous les niveaux. Au niveau du client en considérant son contexte dans la première phase de sélection, au niveau du serveur, en prenant comme une des métriques de sélection la charge des serveurs et enfin au niveau réseau, en sélectionnant toujours le serveur le plus proche tant que celui-ci n'est pas surchargé. On peut d'ailleurs remarquer (cf. Figure A.4) que les serveurs choisis sont ceux qui correspondent aux chemins les plus court. En effet, on peut remarque une différence significative à la fois en terme de délai qu'en nombre de sauts entre les deux scénarios.

Les résultats de simulation montrent bien l'efficacité de notre approche. Nous avons également évalué la stratégie de sélection du meilleur serveur que nous proposons dans un cadre d'infrastructure hybride où les serveurs et les HB collaborent pour distribuer des contenus vidéo et en la comparant à une stratégie se basant sur la charge des serveurs (la stratégie la plus utilisée dans les papier de recherche) et les résultats on confirmés ceux décrit dans cette section (cf. Section 5.3.5).

A.4.2. Stratégie de mise en cache basée sur la popularité

Dans ce travail, nous nous sommes placés, comme cité précédemment, sur une plateforme hybride. Un overlay formé de HB en mode pair à pair vient assister les serveurs d'un CDN traditionnel dans la fonction de distribution de contenus vidéo à la demande. Ce type de plateformes est considéré comme la prochaine étape dans le paradigme de la distribution de contenus [179]. Quelques travaux récents ont également proposé des solutions reposant sur des passerelles déployées sur la bordure du réseau pour le live streaming [180]-[181] ou la VoD [182]-[183]. Cependant ces travaux proposent des solutions de mise en cache offline nécessitant un coût supplémentaire pour la distribution des vidéos sur les caches. Dans l'approche que nous présentons ici, la stratégie de mise en cache est voulue évolutive et réactive. Les contenus vidéo sont donc mis de manière dynamique dans les caches sur la base de leur popularité et seuls les contenus les plus populaires sont sujets à la mise en cache dans l'overlay. Le détail de la stratégie est donné dans les sections qui suivent.

A.4.2.1. La fréquence des vidéos dans les caches.

Le but premier de cette stratégie est d'assurer la montée à l'échelle du service VoD. La popularité des contenus vidéo est donc un paramètre important dans le design de notre stratégie.

Plus la popularité d'un contenu vidéo est grande, plus sa fréquence dans l'overlay des HB est grande. La popularité, dans notre cas, est évaluée à un niveau global par le *Service Mngr*.

Il a été prouvé dans plusieurs travaux de recherche que la popularité des vidéos suit une distribution zipf. Classées dans l'ordre de leur popularité de 1 à M, la vidéo 1 étant la plus populaire, la probabilité de demande de la vidéo V_i de rang i est évaluée à :

$$P_M(i) = \frac{1/_{i^\alpha}}{\sum_{k=1}^{M} 1/_{k^\alpha}} \quad (1)$$

Avec α une constante évaluée 0.733. Le nombre de HB qui vont héberger V_i est donc directement proportionnel à cette probabilité. En supposant que le nombre de HB formant l'overlay est N, le nombre de vidéos les plus populaires, considérées par la stratégie de cache, est M_p, la capacité d'un cache est C et le nombre N_i de HB qui hébergeront V_i est évalué comme suit :

$$N_i = ceil\left(\frac{P_M(i)}{\sum_{j=1}^{M_p} P_M(j)} * N * C\right) \quad (2)$$

Avec M_p supposé assez grand pour que N_i ne dépasse jamais N.

A.4.2.2. La répartition des vidéos sur les HB de l'overlay

La prise en compte de la popularité des vidéos dans le calcul de leurs fréquences est un facteur déterminant pour l'efficacité de la stratégie de mise en cache. Cependant, si toutes les copies d'une vidéo se retrouvent sur des HB voisines, la stratégie perd de son efficacité. Une distribution efficace sur l'ensemble des HB de l'overlay est donc tout aussi importante.

Dans ce travail, nous utilisons l'algorithme k-means pour déterminer la distance à respecter entre chaque couple de HB hébergeant des copies d'une même vidéo. Pour chaque vidéo V_i, l'overlay des HB est donc partitionné en N_i clusters. Pour chaque cluster ainsi formé, on calcule la distance maximale entre tous les nœuds deux à deux. La moyenne de ces distances, qu'on nommera d_i, est assignée à la vidéo V_i. Toutes les HB qui hébergeront V_i seront donc à une distance supérieure ou égale à d_i. Ce processus est réitéré pour les M_p vidéos les plus populaires, dans l'ordre de leurs popularités.

A.4.2.3. Placement et remplacement des vidéos

En plus du calcul de la fréquence et de la répartition des vidéos, un autre aspect majeur dans le design d'une stratégie de caching est la prise de décision sur le placement d'une vidéo dans le cache d'une HB ou de sa suppression pour la remplacer par une autre vidéo. Dans ce travail, le placement des vidéos se fait comme suit :

En considérant la $j^{ème}$ requête, la HB générant cette requête mets la vidéo V_i dans son cache si les conditions suivantes sont satisfaites :

- La vidéo demandée dans la requête est bien V_i, ce qui estimé à $P_M(i)$ (la stratégie de mise en cache est une stratégie on-line) ;
- La HB la plus proche hébergeant V_i est à une distance supérieure ou égale à d_i ;

- La HB a suffisamment de place dans son cache pour y placer V_i ou au moins une des vidéo de son cache est moins populaire que V_i. Dans ce dernier cas cette vidéo est remplacée par V_i.

Plus formellement, nous avons estimé la probabilité de présence de la vidéo V_i dans le cache $Pb_c\ (i,j)$. Ce qui impliquerait que V_i ait été mise en cache à une demande antérieure à la requête j et qu'elle n'ait pas été remplacée entre temps par une autre vidéo.

Et grâce à cette probabilité nous avons pu estimer le hit ratio de l'overlay des Hb(cf. Section 5.4.).

A.4.2.4. Optimisation du coût

L'efficacité d'une stratégie de réplication est un compromis entre d'une part, sa montée à l'échelle et la qualité des contenus distribués et d'autre part, le coût de cette distribution. Pour être efficace, la stratégie de réplication, présentée ici, doit donc satisfaire le maximum de requêtes simultanées par l'overlay des HB tout en maintenant la qualité des contenus distribués à un seuil satisfaisant pour les utilisateurs. En considérant donc, d'une part, le nombre de plus en plus grand des vidéos dans les catalogue, tout particulièrement avec l'avènement des contenus UGC, et d'autre part, la capacité finie du système en connectivité et en stockage, il est important de modéliser la relation entre le nombre de vidéos les plus populaire à stocker dans l'overlay et le coût de distribution du système qui est directement lié au hit ratio de l'overlay des HB. La valeur optimale de M_p est donc obtenu en l'augmentant graduellement de la valeur initiale de $N/_C$ et de calculer ainsi pour chaque valeur le coût de distribution du système jusqu'à l'obtention de la valeur minimale de ce dernier.

A.4.3. Evaluation de la stratégie de mise en cache

Dans le but d'évaluer l'efficacité de notre stratégie de mise en cache, nous l'avons comparée à un modèle de CDN traditionnel. Les deux scénarios ont été simulés dans le même environnement sous NS2 avec la même topologie, que nous avons généré avec le générateur de topologie Brite, le même modèle de trafic et le même catalogue constitué de 10000 vidéos de 5min chacune. Nous les avons ensuite comparé selon différente métriques comme indiqué dans ce qui suit. Les figures A.6, A.7 et A.8 présentent les résultats des simulations.

Comme illustré dans la figure A.6, le hit ratio de l'overlay des HB est bas dans la phase initiale de la simulation. Ce qui est dû au fait que les vidéos ne sont pas encore placées dans les caches des HB, vue que dans notre stratégie de mise en cache les vidéos sont mise progressivement dans les cache au moment où elles sont demandées. Après approximativement trois fois la durée d'une vidéo (~900s), le hit ratio augmente de manière continue jusqu'à la fin de la simulation. On peut donc conclure que plus les HB hébergent des vidéos plus le hit ratio de l'overlay augmente.

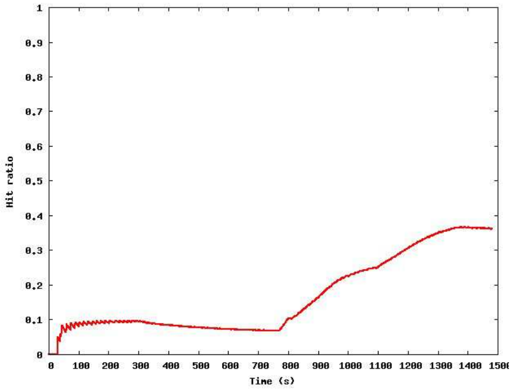

Figure A.3. *Le hit ratio de l'overlay des HB.*

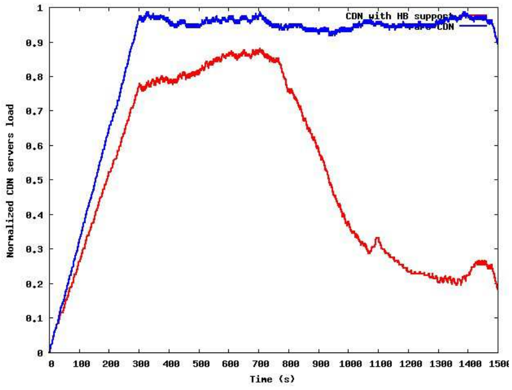

Figure A.4. *La consommation moyenne de bande passante des serveurs du CDN.*

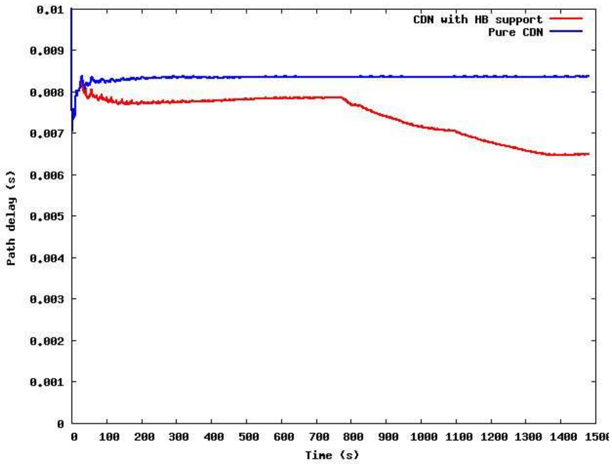

Figure A.5. *Délai moyen des chemins empruntés par les vidéos.*

Figure A.7 représente la charge moyenne des serveurs du CDN en connectivité. Nous pouvons remarquer que notre stratégie permet de sauvegarder jusqu'à 70% de la bande passante des serveurs en fin de simulation. Les deux charges sont presque équivalentes dans la phase initiale de la simulation mais plus les caches des HB se remplissent avec les vidéos dans notre stratégies, la différence entre les deux charges devient plus significative. Ce résultat est cohérent avec le résultat précédent concernant le hit ratio de l'overlay.

Une autre métrique considérée est le délai moyen des chemins empruntés par les vidéos. Cette métrique correspond au délai moyen des paquets RTP des sessions actives à un moment donné de la simulation. Comme illustré dans la figure A.8, le délai est toujours plus bas dans notre stratégie. Cependant, la différence entre les deux scénarios devient de plus en plus significative en avançant dans la simulation.

Nous noterons également que notre approche ne requière pas de bande passante supplémentaire pour la mise en cache des vidéos au niveau des HB vue qu'elles sont placées au moment où elles sont demandées par les utilisateurs finaux.

L'évaluation de notre approche montre donc clairement les aspects prometteurs résultant de la considération de la HB dans la chaine de distribution des vidéos.

Un autre avantage considérable de notre modèle de distribution est sa facilité d'intégration dans les plateformes de nouvelle génération, vue que les HB sont des composants déjà déployés (passerelle résidentielle, set-top box).

Nous pouvons tout de même relever un problème dans le déploiement de notre infrastructure et des solutions associées. C'est sa dépendance aux performances des réseaux d'accès. Cependant, avec le progrès que connais le déploiement de la fibre optique et les investissements des ISP pour rendre les passerelles résidentielles de plus en plus performantes nous encouragent dans nos choix.

A.5. Conclusion

La diversification du contexte des utilisateurs finaux et la grande popularité qu'ont connue les services médias, ces dernières années, ont donné naissance à de nombreuses problématiques de recherche. Dans cette thèse nous nous sommes particulièrement intéressés à deux de ces problématiques. À savoir, la sensibilité au contexte de consommation des services médias ; et le passage à l'échelle de leur distribution sur le réseau Internet tout en garantissant une qualité d'expérience satisfaisante aux utilisateurs finaux.

La première contribution de cette thèse a été de proposer une nouvelle architecture pour les réseaux média du futur. Cette dernière a pour but de faciliter le déploiement d'un écosystème, dans lequel les différents acteurs (fournisseurs de services et/ou de contenus, opérateurs réseaux et utilisateurs finaux) collaborent pour une distribution efficace de services médias riches, personnalisés et adaptatifs sur l'Internet du futur. Les travaux de recherche menés dans cette thèse se fondent sur cette architecture et plus particulièrement, sur les couches hautes de cette dernière. Cependant, conçues de manière générique, les solutions proposées ne sont pas dépendantes du déploiement complet de cette architecture proposée et peuvent tout aussi bien être déployées sur d'autres architectures NGN.

Afin de prendre en compte la sensibilité au contexte dans la distribution des services média, nous avons proposé un Framework autour de la couche virtuelle formée par les HB. Les deux fondements de cette propriété, à savoir la modélisation du contexte et sa gestion, ont été traités dans deux contributions. Dans un premier temps, nous nous sommes basés sur les langages de balisage afin de proposer un Framework léger offrant une réactivité indispensable dans le cas des services média. Puis, nous nous sommes intéressés aux modèles basés sur les ontologies et les systèmes de règles pour plus d'expressivité et de formalité. Dans les deux contributions, un modèle de contexte a été proposé et un prototype du middleware supportant la gestion du contexte a été développé.

Les réseaux de distribution des services médias sur Internet ont également été améliorés pour permettre une distribution efficace et qui passe à l'échelle. Ces réseaux ont donc été augmentés avec les capacités de cache et de connectivité offerte par la couche virtuelle formée par l'interconnexion des HB. Afin d'optimiser l'utilisation de ces nouvelles ressources, nous avons proposé, d'une part, un schéma efficace pour la redirection des requêtes des utilisateurs finaux, qu'on a basé sur le modèle le l'anycast applicatif ; et d'autre part, une stratégie réactive de mise en cache et de remplacement dans les caches qu'on a basé sur la popularité des contenus média demandés.

Appendix B

List of acronyms

ADTF	Adaptation Decision Taking Framework
API	Application Programming Interface
ATM	Asynchronous Transfer Mode
CAN	Content-Aware Network
CBIM	Common Baseline Information Model
CC/PP	Composite Capabilities/Preference Profiles
C-CAST	Context CASTing
CCN	Content-Centric Networking
CDN	Content Delivery Networks
CoBrA	Context Broker Architecture
CONON	CONtext Ontology
DDL	Description Definition Language
DIA	Item Adaptation
ETSI	European Telecommunications Standards Institute
EU	End User
FI	Future Internet
FMIA	Future Media Internet Architecture
FMIA-TT	Future Media Internet Architecture – Think Tank group
FTTH	Fiber-To-The-Home
GPS	Global Positioning System
GSLB	Global Server Load Balancing
GSM	Global System for Mobile Communications
GUP	Generic User Profile
GUP	Generic User Profile
HB	Home-Boxe
HLR	Home Location Register
HSS	Home Subscriber Server

IGMP	Internet Group Management Protocol
IMS	IP Multimedia Subsystem
IPTV	Internet Protocol based Television
KB	Knowledge Base
LFU	Least Frequently Used
LRU	Least Recently Used
MANE	Media-Aware Network Element
MPEG	Moving Picture Experts Group
MPLS	Multiprotocol Label Switching
NASS	Network Attachment Subsystem
NE	Network Environment
NGN	Next-Generation Networks
NRM	Network Ressource Manager
ORM	Object Role Modeling
OWL	Ontology Web Language
OWL-DL	Ontology Web Language Description Logic
OWL-S	Semantic Markup for Web Services
P2P	Peer-to-Peer
QoE	Quality-of-Experience
QoS	Quality of Service
RACS	Resource Admission and Control Subsystem
RDFS	Resource Description Framework Schema
RTP	Real-time Transfert Protocol
S/CP	Service/Content Provider
SE	Service Environment
SIP	Session Initiation Protocol
SLA	Service Level Agreements
SNDlib	Survivable Network Design Library ()
SOCAM	Service-oriented Context Aware Middleware
SOUPA	Standard Ontology for Ubiquitous and Pervasive Application
SVC	Scalable Video Coding
SWRL	Semantic Web Rule Language
TISPAN	Telecoms Internet converged Services Protocols for Advanced Networks
UAProf	User Agent Profile
UDC	User Data Convergence
UDR	User Data Repository
UE	User Environment
UGC	User Generated Contents

UML	Unified Modeling Language
URL	Uniform Resource Locator
VCAN	Virtual Content-Aware Network
VoD	Video-on-Demand
W3C	World Wide Web Consortium
XML	Extensible Markup Language

Appendix C

Publications

C.1. Published papers in international conferences and workshops

S. Ait Chellouche, D. Négru, E. Borcoci, E. LeBARS, "*Context-Aware Distributed Multimedia Provisioning based on Anycast Model towards Future Media Internet*", in Proc. of 8th IEEE Consumer Communications and Networking Conference (CCNC), Las Vegas, NV, 2011.

S. Ait Chellouche, D. Négru, Y. Chen, "*Home-Boxes Support for an Efficient Video-on-Demand Distribution*", Proc. the IEEE International Conference on Multimedia and Expo (ICME'11), Barcelona, Spain, 2011.

S. Ait Chellouche, D. Négru, E. Borcoci, E. LeBARS, "*Anycast-based Context-Aware Server Selection Strategy for VoD Services*", in Proc. of IEEE GLOBECOM Workshops (GC Wkshps), Miami, Florida, 2010.

S. Ait Chellouche, J. Arnaud, D. Négru, "*Flexible User Profile Management for Context-Aware Ubiquitous Environments*", in Proc. of 7th IEEE Consumer Communications and Networking Conference (CCNC), Las Vegas, NV, 2010.

C.2. Papers under submission in International Conferences

S. Ait Chellouche, D. Négru, Y. Chen, "*Home-Box-assisted Content Distribution Network for Internet Video-on-Demand Services*", submitted to IEEE Symposium on Computers and Communications (ISCC2012), Cappadocia, Turkey, 2012.

S. Ait Chellouche, D. Négru, "*Context-aware multimedia services provisioning in future Internet using ontology and rules*", submitted to the Tenth International Conference on Pervasive Computing (PERVASIVE 2012), Newcastle, UK, 2012.

C.3. Reports

Y. Chen, **S. Ait Chellouche** et al.,"*D 3.1.1: ALICANTE User Profile – Intermediate*", ALICANTE deliverable, 2011.

S. Ait Chellouche, D. Négru, Y. Chen, "*Home-Box-assisted Content Distribution Network for Internet Video-on-Demand Services*", Technical report, 2012.

References

[1] Cisco Visual Networking Index, 2009. Available : www.cisco.com

[2] Cisco Visual Networking Index, 2011. Available : www.cisco.com

[3] “The future internet”, John Domingue, Alex Galis, Anastasius Gavras, Theodore Zahariadis, and Dave Lambert (Eds.). Springer-Verlag, Berlin, Heidelberg, 2009.

[4] Future Media Internet Architecture Think Tank (FMIA-TT), “Future Media Internet Architecture Reference Model”, 1. March 2011, http://www.future-internet.eu/uploads/media/FMIA_Reference_Architecture.pdf.

[5] B.N. Schilit, M. Theimer, “Disseminating Active Map Information to Mobile Hosts”, IEEE Network, Vol.8, N°5, pp. 22-32, September/October 1994.

[6] B.N. Schilit, N.I. Adams and R. Want. Context-Aware Computing Applications. In Proceedings of the IEEE Workshop on Mobile Computing Systems and Applications (WMCSA94), Santa Cruz CA, December 1994.

[7] P.J. Brown, “The Stick-e Document: a Framework for Creating Context-Aware Applications”, Electronic Publishing ’96, pages 259-272, 1996.

[8] A. Held, S. Buchholz, A. Schill, “Modeling of context information for pervasive computing applications”, In Proceedings of SCI 2002/ISAS 2002, 2002.

[9] N. S. Ryan, J. Pascoe,and D.R. Morse “Enhanced reality fieldwork : the context-aware archaeological assistant”, In V.Gaffney, M. van Leusen, and S. Exxon, editors, Computer Applications in Archaeology 1997. Oxford : Tempus Reparatum, , Oct. 1998.

[10] A.K. Dey, “Context-aware computing: the CyberDesk project”, Proceedings of the AAAI, Spring symposium on Intelligent Environments, Menlo Park, CA, pp.51–54, 1998.

[11] A. Ward, A., Jones, A. Hopper, “A New Location Technique for the Active Office”, IEEE Personal Communications 4(5), pages 42-47, 1997.

[12] A. Schmidt, K.A. Aidoo, A. Takaluoma, U. Tuomela, K. Van Laerhoven, and W. Van de Velde, “Advanced interaction in context”, In Proceedings of First International Symposium on Handheld and Ubiquitous Computing, HUC'99, pages 89-101, Karlsruhe, Germany, September 1999.

[13] GREGORY D. ABOWD and ELIZABETH D. MYNATT, “Charting Past, Present, and Future Research in Ubiquitous Computing”, ACM Transactions on Computer-Human Interaction, Vol. 7, No. 1, Pages 29–58, March 2000.

[14] G. Chen, D. Kotz, “A Survey of Context-Aware Mobile Computing Research”, Dartmouth College, Hanover, NH, 2000.

[15] A.K. Dey, G.D. Abowd, D. Salber, “A conceptual framework and a toolkit for supporting rapid prototyping of context-aware applications”, Human-Computer Interactions (HCI) Journal, Vol. 16, Nos. 2–4, pp.7–166, 2001.

[16] A.K. Dey and Gregory D. Abowd, “Towards a Better Understanding of context and context-awareness”, Proceedings of the workshop on the what, who, where, when and how of context-awareness, ACM Press, New York, 2000.

[17] A. K. Dey, M. Futakawa, D. Salber, and G. D. Abowd, “The Conference Assistant: Combining Context-Awareness with Wearable Computing”, Proc. the 3rd International Symposium on Wearable Computers (ISWC '99), San Francisco, CA, 1999.

[18] J.Gwizdka, “What’s in the Context”, CHI2000 Workshop, GVU Technical Report;GIT-GVU-00-18, Georgia Institute of Technology, 2000.

[19] M.A. Razzaque, S. Dobson, P. Nixon, “Categorization and Modelling of Quality in Context Information”, of the IJCAI 2005 Workshop on AI and autonomic communications, Edinburgh, Scotland, 2005.

[20] D. Petrelli, E. Not, C. Strapparava, O. Stock, M. Zancanaro, “Modeling Context is Like Taking Pictures”, CHI2000 Workshop, GVU Technical Report GIT-GVU-00-18, Georgia Institute of Technology, 2000.

[21] B. N. Schilit, N. L. Adams, R. Want, Context-aware computing applications. In IEEE Workshop on Mobile Computing Systems and Applications, Santa Cruz, CA, US, 1994.

[22] G. Klyne, “A Syntax for Describing Media Feature Sets”, RFC 2533, Mar 1999. Available at: ttp://www.faqs.org/rfcs/rfc2533.html)

[23] ETSI TS 1222 240 V10.0.0: “Service requirements for 3GPP Generic User Profile (GUP), Stage 1", 05-2011.

[24] ETSI TS 123 240 V10.0.0: “3GPP Generic User Profile (GUP) requirements , Architecture (Stage 2)”, 03-2011.

[25] ETSI TS 129 240 V10.0.0: “3GPP Generic User Profile (GUP), Network (Stage 3)”, 05-2011.

[26] H.S. Thompson, D. Beech, M. Maloney, N. Mendleshon, "XML Schema Part 1: Structures," Recommendation, World Wide Web Consortium, May 2002. http://www.w3.org/TR/xmlschema-1/.

[27] P.V. Biron, A. Malhotra, "XML Schema Part 2: Datatypes," Recommendation, World Wide Web Consortium, May 2002. http://www.w3.org/TR/xmlschema-2/.

[28] P.V. Biron, A. Malhotra, "XML Schema Part 2: Datatypes," Recommendation, World Wide Web Consortium, May 2002. http://www.w3.org/TR/xmlschema-2/.

[29] ISO/IEC 15938-5 (2003), Information technology - Multimedia content description interface (MPEG-7) - Part 5 : Multimedia Description Schemes

[30] ISO/IEC 21000-7 (03-2004), Information Technology - Multimedia Framework (MPEG-21) - Part 7: Digital Item Adaptation.

[31] W3C, "Composite Capabilities / Preferences Profile (CC/PP)". http://www.w3.org/Mobile/CCPP.

[32] O. Lassila, R. Swick, "Resource Description Framework (RDF) Model and Syntax Specification", W3C Recommendation, Feb 22, 1999. http://www.w3.org/TR/1999/REC-rdf-syntax-19990222/

[33] "RDF Vocabulary Description Language 1.0: RDF Schema", W3C Recommendation, 10 February 2004. http://www.w3.org/TR/rdf-schema/

[34] Wireless Application Protocol Forum Ltd, "WAG UAProf, Proposed Version 30-May-2001". http://www1.wapforum.org/tech/documents/

[35] J. Indulska, R. Robinson, A. Rakotonirainy, K. Henricksen, "Experiences in Using CC/PP in Context-Aware Systems", Proceeding of the 4th International Conference on Mobile Data Management, Springer-Verlag, Melbourne Australia, January, 2003, pp. 247-261.

[36] Mark H. Butler, "CC/PP and UAProf: Issues, Improvements and Future Directions", W3C Workshop on Delivery Context, Hewlett-Packard, Sophia-Antipolis, France, March, 2002.

[37] J. Bauer, "Identification and Modeling of Contexts for Different Information Scenarios in Air Traffic', Diplomarbeit, 2003.

[38] K. Henricksen, J. Induiska, A.Rakotonirainy, "Generating Context Management Infrastructure from High-Level Context Models", In Industrial Track Proceedings of the 4th International Conference on Mobile Data Management (MDM2003), Melbourne, Australia, 2003.

[39] T. A. Halpin, "Information Modeling and Relational Databases: From Conceptual Analysis to Logical Design", Morgan Kaufman Publishers, San Francisco, 2001.

[40] 3GPP TR 22.985 V10.0.0: "User Data Convergence (UDC)", 03-2011.

[41] 3GPP TR 32.182 V10.0.0: "User Data Convergence (UDC); Common Baseline Information Model (CBIM)", 03-2011.

[42] 3GPP TS 32.182 V10.0.0: "User Data Convergence (UDC); Framework for Model Handling and Management3GPP", 03-2011.

[43] T. Hofer, W. Schwinger, M. Pichler, G. Leonhartsberger, J. Altmann, "Context-awareness on mobile devices – the hydrogen approach', Proceedings of the 36th Annual Hawaii International Conference on System Sciences, pp.292–302, 2002.

[44] J. MCCARTHY, "Notes on formalizing contexts", In Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence, San Mateo, California, 1993

[45] J. McCarthy, BUVAˇC, "Formalizing context (expanded notes)", in Working Papers of the AAAI Fall Symposium on Context in Knowledge Representation and Natural

Language, Menlo Park, California,1997.

[46] T. R. Gruber, "The Role of Common Ontology in Achieving Sharable, Reusable Knowledge Bases", In J. A. Allen, R. Fikes, & E. Sandewall (Eds.), Principles of Knowledge Representation and Reasoning: Proceedings of the Second International Conference, Cambridge, MA, pages 601-602, Morgan Kaufmann, 1991.

[47] M. Gruninger, J. Lee, "Ontology – Applications and Design", in communications of the ACM, 2002.

[48] T. R. Gruber, "A Translation Approach to Portable Ontology Specifications ", Knowledge Acquisition, 5(2):199-220, 1993.

[49] OWL Web Ontology Language Overview, W3C Recommendation 10 February 2004. ttp://www.w3.org/TR/owl-features/

[50] X. H. Wang, T. Gu, D. Q. Zhang, and H. K. Pung: "Ontology Based Context Modeling and Reasoning using OWL", Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications Workshops, Orlondo, Florida, USA, 2004.

[51] H. Chen, T. Finin, A. Joshi, "An ontology for context-aware pervasive computing environments", The Knowledge Engineering Review, Cambridge University Press, Vol. 18, pp.197–207, 2003.

[52] H. Chen, F. Perich, T. Finin, A. Joshi, "SOUPA: Standard Ontology for Ubiquitous and Pervasive Applications", The First Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services, (MOBIQUITOUS 2004), Boston, Massachusetts, USA 2004.

[53] D. Brickley, L. Miller, "FOAF vocabulary specification, In RDFWeb Namespace Document", RDFWeb, xmlns.com, 2003.

[54] F. Perich, A. Joshi, Y. Yesha, T. Finin, "Neighborhood-Consistent Transaction Management for Pervasive Computing Environments", In 14th International Conference on Database and Expert Systems Applications (DEXA 2003), Prague, Czech Republic, September 2003.

[55] F. Pan, J. R. Hobbs, "Time in OWL-S", In Proceedings of AAAI-04 Spring Symposium on SemanticWeb Services, Stanford University, California, 2004.

[56] H. Chen, T. Finin, A. Joshi, "An ontology for contextaware pervasive computing environments", Special Issue on Ontologies for Distributed Systems, Knowledge Engineering Review, 2003.

[57] F. Perich, "MoGATU BDI Ontology", 2004.

[58] L. Kagal, T. Finin, A. Joshi, "A Policy Based Approach to Security for the Semantic Web", In 2nd International Semantic Web Conference (ISWC2003), September 2003.

[59] T. Strang, C. Linnhoff-Popien, "A Context Modeling Survey", Proc. of Workshop on Advanced Context Modelling, Reasoning and Management, (UbiComp 2004) - The Sixth International Conference on Ubiquitous Computing. 2004.

[60] R. Want, A. Hopper, V. Falcão, J. Gibbons, “The Active Badge location system”. ACM Transactions on Information Systems, 10(1):91-102, 1992.

[61] R. Want, B.N. Schilit, N.I. Adams, R. Gold, K. Petersen, D. Goldberg, J. R. Ellis, and M. Weiser. “An overview of the PARCTAB ubiquitous computing experiment”. IEEE Personal Communications, 2(6):28-43, December 1995.

[62] S. Long, R. Kooper, G. D. Abowd, and Christopher G. Atkeson, “Rapid prototyping of mobile context-aware applications: the Cyberguide case study”. In Proceedings of the Second Annual International Conference on Mobile Computing and Networking, pages 97-107, White Plains, NY, November 1996.

[63] A. Asthana, M. Cravatts, P. Rzyzanowski. “An indoor wireless system for personalized shopping assistance”. In Proceedings of IEEE Workshop on Mobile Computing Systems and Applications, pages 69-74, Santa Cruz, California, 1994.

[64] N. Davies, K. Cheverst, K. Mitchell, A. Friday. “Caches in the air: Disseminating tourist information in the GUIDE system”. Proc. of Second IEEE Workshop on Mobile Computing Systems and Applications, New Orleans, Louisiana, 1999.

[65] J. Indulska, P. Sutton, “Location management in pervasive systems”, Proceedings of the Australasian Information Security Workshop (CRPITS’03), pp.143–151, 2003.

[66] A. Schmidt, K. van Laerhoven, “How to build smart applications?”, IEEE Personal Communications, Vol. 8, No. 4, pp.66–71, 2001.

[67] M. Baldauf, “A survey on context-aware systems”, International Journal of Ad Hoc and Ubiquitous Computing, Vol. 2, No. 4, 2007.

[68] 3GPP TR 32.182 V10.0.0: “User Data Convergence (UDC); Technical realization and information flows”, 03-2011.

[69] A.K. Dey, “The Context Toolkit – A Toolkit for Context-aware Applications”, 2001, http://www.cs.berkeley.edu/~dey/context.html.

[70] T. Gu, H.K. Pung, D.Q. Zhang, “A middleware for building context-aware mobile services”, Proc. IEEE Vehicular Technology Conference (VTC 2004), Milan, Italy, 2004.

[71] ICT/FP7 C-CAST, http://www.ict-ccast.eu

[72] B. Cain, S. Deering, I. Kouvelas, B. Fenner, A. Thyagarajan, “Internet Group Management Protocol”, Version 3, RFC 3376, October 2002.

[73] H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson, "RFC 3550 : RTP : A Transport Protocol for Real-Time Applications", Request for Comments, IETF, July. 2003.

[74] H. Schulzrinne, A. Rao, R. Lanphier, "RFC 2326: Real Time Streaming Protocol (RTSP)", Request for Comments, IETF, April 1998.

[75] Microsoft Mediaroom IPTV Service. http://www.microsoft.com/mediaroom.

[76] 3GPP Forum, “IP Multimedia Subsystem (IMS)”, Stage 2, 3GPP TS 22 228 v8.11, 2009-12-14, 2009.

[77] B. Chatras, M. Said, “Delivering Quadruple Play with IPTV over IMS”, International Conference on Intelligence in Networks (ICIN2007), Bordeaux, France, 2007.

[78] E. Mikoczy, D. Sivchenko, Xu Bangnan, J.I. Moreno, “IPTV Systems, Standards and Architectures: Part II - IPTV Services over IMS: Architecture and Standardization”, IEEE Communications Magazine, 2008.

[79] ETSI TS 183 064 V3.4.1, tech. spec. draft, “Telecommunications and Internet converged Services and Protocols for Advanced Networking (TISPAN); NGN integrated IPTV subsystem stage 3 specification”, February 2011.

[80] ETSI TS 182 027 V3.5.1, “Telecommunications and Internet converged Services and Protocols for Advanced Networking (TISPAN), IPTV Architecture, IPTV functions supported by the IMS subsystem”, March 2011.

[81] ITU-T FOCUS GROUP ON IPTV, wkg. doc. IPTV-DOC-0084, “IPTV Architecture,” 4th FG IPTV mtg., Bled, Slovenia, 2007.

[82] ITU-T Rec. Y.2012, “Functional Requirements and Architecture of the NGN,” Sept. 2006.

[83] Open IPTV Forum, “Service and Platform Requirements”, “Services and Functions”, “Functional Architecture, Solution Specification” – volumes 1-7, http://www.openiptvforum.org/.

[84] A. Vakali, and G. Pallis, “Content Delivery Networks: Status and Trends,” IEEE Internet Computing, IEEE Computer Society, pp. 68-74, 2003.

[85] M. Pathan, R. Buyya,"A Taxonomy and Survey of Content Delivery Networks", Technical Report, GRIDS-TR-2007-4, Grid Computing and Distributed Systems Laboratory, the University of Melbourne, Australia. 12 February, 2007.

[86] Y. Bartal, “Probabilistic approximation of metric space and its algorithmic applications,” in 37th Annual IEEE Symposium on Foundations of Computer Science, Oct. 1996.

[87] S. Jamin, C. Jin, Y. Jin, D. Raz, Y. Shavitt, and L. Zhang, “On the placement of Internet instrumentation,” In Proceedings of IEEE INFOCOM, pp. 295-304, Tel-Aviv, Israel, March 2000.

[88] B. Li, M. J. Golin, G. F. Ialiano, X. Deng, “On the Optimal Placement of Web Proxies in the Internet”, In Proc. of INFOCOM’99, March 1999.

[89] L. Qiu, V. N. Padmanabhan, and G. M. Voelker, “On the Placement of Web Server Replicas,” In Proceedings of IEEE INFOCOM, Anchorage, Alaska, USA, pp. 1587-1596, April 2001.

[90] S. Jamin, C. Jin, A. R. Kure, D. Raz, Y. Shavitt, “Constrained Mirror Placement on the Internet”, In Proceedings of IEEE INFOCOM, Anchorage, Alaska, USA, April 2001.

[91] M. Yang, Z. Fei, “A model for replica placement in content distribution networks for multimedia applications”, IEEE International Conference on Communications ICC, Anchorage, Alaska, USA, 2003.

[92] D. Wessels, K. Claffy, “Internet Cache Protocol (ICP) version 2”, Internet Engineering Task Force RFC 2186, September 1997.

[93] A. Rousskov, D. Wessels, “Cache Digests”, Computer Networks and ISDN Systems, Vol. 30, No. 22-3, pp. 2155-2168, November 1998.

[94] S. Gadde, M. Rabinovich, J. Chase, “Reduce, Reuse, Recycle: An Approach to Building Large Internet Caches”, In Proceedings of 6th Workshop on Hot Topics in Operating Systems, pp.93-98, April 1997.

[95] V. Valloppillil, K. W. Ross, “Cache Array Routing Protocol v1.0”, Internet Draft, February 1998.

[96] D. Karger, A. Sherman, A. Berkheimer, B. Bogstad, R. Dhanidina, K. Iwamoto, B. Kim, L. Matkins, Y. Yerushalmi, “Web Caching with Consistent Hashing”, Computer Networks, Vol. 31, No. 11-16, pp. 1203-1213, 1999.

[97] S. Sivasubramanian, M. Szymaniak, G. Pierre, and M. Van Steen, “Replication of Web Hosting Systems,” ACM Computing Surveys, Vol. 36, No. 3, ACM Press, NY, USA, 2004.

[98] M. Szymaniak, G. Pierre, and M. van Steen, “Netairt: A DNS-based Redirection System for Apache”, in Proceedings of International Conference WWW/Internet, Algrave, Portugal, 2003.

[99] D. Karger, A. Sherman, A. Berkheimer, B. Bogstad, R. Dhanidina, K. Iwamoto, B. Kim, L. Matkins, and Y. Yerushalmi, “Web Caching with Consistent Hashing,” Computer Networks, Vol. 31, No. 11-16, pp. 1203-1213, 1999.

[100] N. Bartolini, E. Casalicchio, and S. Tucci, “A Walk Through Content Delivery Networks,” In Proceedings of MASCOTS 2003, LNCS Vol. 2965/2004, pp. 1-25, April 2004.

[101] S. Bhattacharjee, M.H. Ammar, E.W. Zegura, V. Shah, and Z. Fei, “Application-Layer Anycasting,” IEEE INFOCOM’97, Kebe, Japan, April 1997.

[102] E. Zegura, M. Ammar, Z. Fei, and S. Bhattacharjee. “Application-layer anycasting: A server selection architecture and use in a replicated web service”. ACM/IEEE Transactions on Networking, 8(4):455-466, 2000.

[103] M. Hofmann, and L. R. Beaumont, “Content Networking: Architecture, Protocols, and Practice”, Morgan Kaufmann Publishers, San Francisco, CA, USA, pp. 129-134, 2005.

[104] Nortel Networks: “Web OS Switch Software 10.0 Application Guide”, Nortel Networks, Part Number 212777, pp. 207–219, February 2002.

[105] Foundry Networks: “Global Server Load Balancing with Serveriron,” Application Note.

[106] A. Barbir, B. Cain, R. Nair, O. Spatscheck, “RFC 3568: Known Content Network (CN) Request-Routing Mechanisms”, Request for Comments, July 2003.

[107] C. Partridge, T. Mendez and W. Milliken, “RFC1546: Host Anycasting Service”, Rrequest For Comment, 1993.

[108] Z. Fei, S. Bhattacharjee, E.W. Zegura, M. Ammar, “A Novel Server Selection Technique for Improving the Response Time of a Replicated Server”, INFOCOM'98, 1998.

[109] Akamai. http://www.akamai.com;

[110] Bruce Maggs, “How Akamai handles large events”, Network Mapping and Measurement Conference (NMMC2009), 2009.

[111] J. Dilley, B. Maggs, J. Parikh, H. Prokop, R. Sitaraman, B. Weihl, “Globally distributed content delivery”, Internet Computing, IEEE, Volume 6 Issue 5, pages 50-58, 2002.

[112] D. S. Milojicic, V. Kalogeraki, R. Lukose, K. Nagaraja, J. Pruyne, B. Richard, S. Rollins, Z. Xu, “Peer-to-Peer Computing”, Technical Report, HP Laboratories, Palo Alto, CA, HPL-2002-57, March 2002.

[113] Bittorrent. Available at http://www.bittorrent.

[114] Napster. Available at http://www.napster.com/

[115] L. Eng Keong, J. Crowcroft, M. Pias, R. Sharma, S. Lim, “A survey and comparison of peer-to-peer overlay network schemes”, IEEE Communications Surveys &Tutorials, Volume 7, Issue 2, pp. 72 – 93, Mars 2006.

[116] S. German, W. Markus, U. Herwig, “Search Methods in P2P Networks: A Survey”, Lecture Notes in Computer Science, Volume 3473/2006, pp. 59-68, 2006.

[117] T. Klingberg, R. Manfredi, “Gnutella 0.6”, Technical report, June 2002. Available at: http://rfc-gnutella.sourceforge.net/src/rfc-0_6-draft.html.

[118] I. Clarke, O. Sandberg, B. Wiley, T. W. Hong, “Freenet: A distributed anonymous information storage and retrieval system”, Technical report, July 2000.

[119] S. Ratnasamy, P. Francis, M. Handley, R. Karp, S. Shenker, “A scalable content-addressable network”, Technical report, ACM SIGCOMM, 2001.

[120] I. Stoica, R. Morris, D. Liben-Nowell, D.R. Karger, M.F. Kaashoek, F. Dabek, H. Balakrishnan, “Chord: A Scalable Peer-to-Peer Lookup Protocol for Internet Applications”, IEEE/ACM Trans. Net., vol. 11, no. 1, pp. 17–32, 2003.

[121] B. Y. Zhao, L. Huang, J. Stribling, S. C. Rhea, A. D. Joseph, J. D. Kubiatowicz, “Tapestry: A resilient global-scale overlay for service deployment”, Technical report, IEEE Journal, January 2004.

[122] B. Y. Zhao, J. D. Kubiatowicz, A. D. Joseph, “Tapestry: An infrastructure for fault-tolerant wide-area location and routing”, Technical report, April 2001.

[123] J. Choi, J. Han, E. Cho, T. Kwon, Y. Choi, "A Survey on content-oriented networking for efficient content delivery," Communications Magazine, IEEE , vol.49, no.3, pp.121-127, March 2011.

[124] ALICANTE “MediA Ecosystem Deployment through Ubiquitous Content-Aware Network Environments” deliverables, 2011.

http://www.ict-alicante.eu/index.php?page=pages&id=5.

[125] “Future Media Internet Architecture reference model outline”, www.fi-nextmedia.eu/.

[126] Howarth, M.P. et al., “Provisioning for Interdomain Quality of Service: the MESCAL Approach”, IEEE Communications Magazine, June 2005, pp. 129-137.

[127] MESCAL D1.2: “Initial Specification of Protocols and Algorithms for Inter-domain SLS Management and Traffic Engineering for QoS-based IP Service Delivery and their Test Requirements”, January 2004, www.mescal.org.

[128] T.Ahmed, A. Asgari, A.Mehaoua, E. Borcoci, L.B Équille, K. Georgios “End-to-end quality of service provisioning through an integrated management system for multimedia content delivery” Computer Communications, Special Issue: Emerging Middleware for Next Generation Networks, Volume 30, Issue 3, Pages 638-651, February 2007.

[129] M. Boucadair, P. Lévi, D. Griffin, N. Wang, M. Howarth, and G. Pavlou, E. Mykoniati and P. Georgatsos, B. Quoitin, J. Rodríguez Sánchez, M. L. García-Osma, “A Framework for End-to-End Service Differentiation: Network Planes and Parallel Internets”, IEEE Communication Magazine, pp. 134-143, Sept. 2007.

[130] 4WARD, “A clean-slate approach for Future Internet”, http://www.4ward-project.eu/.

[131] Publish-Subscribe Internet Routing Paradigm, http://www.psirp.org/.

[132] M. Ain ed. et al., Public Delivrable D2.2, “Conceptual Architecture of PSIRP” http://www.psirp.org/.

[133] N. Fotiou, et. al., “Developing Information Networking Further: From PSIRP to PURSUIT”.

[134] Open ContEnt Aware Networks (OCEAN), http://www.ict-ocean.eu/.

[135] “P2P-Next, Next Generation Peer-to-Peer Content Delivery Platform”, http://www.p2p-next.org.

[136] F.J. Ramón Salguero, “COntent Mediator architecture for content-aware nETworks”, 2010, http://www.comet-project.org/.

[137] G. García de Blas, (Ed) et al., Public Deliverable D2.1, “Business Models and System Requirements for the COMET System”, 2011, http://www.comet-project.org/.

[138] W. K.Chai, N. Wang, I. Psaras G. Pavlou, C. Wang, G. García de Blas, F.J. Ramon Salguero, Lei Liang, S.Spirou, Andrzej Beben, E. Hadjioannou, “CURLING: Content-Ubiquitous Resolution and Delivery Infrastructure for Next-Generation Services” , IEEE Communications Magazine, March 2011, pp.112-120.

[139] M. Rockl, K. Frank, P.G. Hermann, M. Vera, “Knowledge Representation and Inference in Context-Aware Computing Environments”, Proc. Second International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies, Valencia, Spain, 2008.

[140] Jaeyoung Choi, Jinyoung Han, Eunsang Cho, Kwon T.,Yanghee Choi, “A Survey on content-oriented networking for efficient content delivery”, IEEE Communications

Magazine, Volume: 49 Issue:3, pp. 121 – 127, mars 2011.

[141] M. Weiser, "The computer for the twenty-first century," Scientific American, pp. 92-99, 1991.

[142] W. Cherif, A. Ksentini, D. Négru, M. Sidibé, “A_PSQA: Efficient Real-Time Video Streaming QOE Tool in a Future Media Internet Context”, proc. IEEE International Conference on Multimedia and Expo (ICME 2011), Barcelona, Spain, 2011.

[143] S. Ait Chellouche, J. Arnaud, D. Négru, “Flexible User Profile Management for Context-Aware Ubiquitous Environments”, in Proc. of 7th IEEE Consumer Communications and Networking Conference (CCNC), Las Vegas, NV, 2010.

[144] UPnP™ Forum, "UPnP™ Device Architecture 1.1," 2008.

[145] G. Pavlovic-Lazetic, “Native XML databases vs. relational databases in dealing with XML documents”, Kragujevac Journal of Mathematics, Citeseer, 2007.

[146] Native XML Database System : http://modis.ispras.ru/sedna/.

[147] “XQuery 1.0: An XML Query Language (Second Edition)”, W3C Recommendation 14 December 2010. http://www.w3.org/TR/xquery/

[148] OWL-S. Semantic markup for web services; w3c member submission, 22 November 2004. http://www.w3.org/Submission/OWL-S/

[149] I. Horrocks, P.F. Patel-Schneider, H. Boley, S. Tabet, B. Grosof, M. Dean,: SWRL: A Semantic Web Rule Language Combining OWL and RuleML, 2004.

[150] S. Bechhofer et al., “DIG 2.0 Towards a flexible interface for Description Logic reasoners”, Proceedings of International Workshop on OWL Experiences and Directions (OWLED), USA, 2006.

[151] FaCT++Available at: owl.man.ac.uk/factplusplus/

[152] I. Horrocks, U. Sattler, “A Tableaux Decision Procedure for SHOIQ”, Proceedings of 19th International Joint Conference on Artificial Intelligence (IJCAI), Scotland, UK, 2005.

[153] M. Klusch, B. Fries, K. Sycara, “Automated Semantic Web Service Discovery with OWLS-MX”, Proceedings of the 5th International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS), Hakodate, Japan, 2006.

[154] M. Klusch, K-U. Renner, “Dynamic Re-Planning of Composite OWLS Services”, Proceedings of 1st IEEE Workshop on Semantic Web Service Composition, Hongkong, China, IEEE CS Press, 2006.

[155] I. Horrocks, P. Patel-Schneider, “A proposal for an OWL rules language”, Proceedingsof 13th International World Wide Web Conference (WWW), 2004.

[156] M.J. O'Connor, H. Knublauch, S.W. Tu, M.A. Musen,: “Writing Rules for the Semantic Web Using SWRL and Jess”. Protege with Rules Workshop 2005, Madrid, Spain, 2005.

[157] Web Services Description Language (WSDL) 1.1, W3C Note 15 March 2001. http://www.w3.org/TR/wsdl

[158] J. Wang, “A survey of web caching schemes for the Internet”, ACM SIGCOMM Computer Communication Review Volume 29 Issue 5, October 1999.

[159] B. D. Davison, “A Web Caching Primer”, IEEE Internet Computing Journal, Volume 5 Issue 4, July 2001.

[160] D. Zeng, F. Y. Wang, M. Liu, “Efficient web content delivery using proxy caching techniques”, IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, volume 34 issue 3, pp. 270-280, 2004.

[161] C. Partridge, T. Mendez and W. Milliken, “Host Anycasting Service”, RFC1546, 1993.

[162] T. Hardie, “Distributing Authoritative Name Servers via Shared Unicast Addresses”, RFC 3258, April 2002.

[163] C. Huitema, “An Anycast Prefix for 6to4 Relay Routers”, RFC3068, June 2001

[164] D. Katabi, J. Wroclawski, “A Framework for Scalable Global IP-Anycast (GIA)”, SIGCOMM’00, Stockholm, Sweden, 2000.

[165] W. Xiaonan, Q. Huanyan, “An Anycast Communication Model on Solving Scalability in IPv6”, Control, Automation, Robotics and Vision, 2006.

[166] H. Miura, M. Yamamoto, “Server Selection Policy in Active Anycast”, IEICE Trans. Commun., Vol. E84.B, No. 10 October 2001.

[167] H.B. Hashim, J.A. Manan, “An active anycast rtt-based server selection technique”, Proceedings of 13th IEEE International Conference on Networks 1 145–149, 2005.

[168] H. Ballani, P. Francis, “Towards a global IP Anycast service”, Proceedings of the 2005 conference on Applications, technologies, architectures, and protocols for computer communications 301–312, 2005.

[169] T. Stevens, J. Vermeir, M. De Leenheer, C. Develder, F. De Turck, B. Dhoedt, P. Demeester, "Distributed Service Provisioning Using Stateful Anycast Communications", lcn, pp.165-174, 32nd IEEE Conference on Local Computer Networks, 2007.

[170] H.-Y. Kung, C.-M. Huang, H.-H. Ku, C.-Y. Lin, “Efficient multimedia distribution architecture using anycast”, Computer Communications, v.30 n.2, p.288-301, January, 2007.

[171] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, R. Sparks,M. Handley, E. Schooler, “SIP: Session Initiation Protocol”, RFC 3261, June 2002.

[172] A. Niemi, “RFC 3903: Session Initiation Protocol (SIP) Extension for Event State Publication”, Request For Comment, October 2004.

[173] H. Yu, D. Zheng, B. Zhao, W. Zheng, “Understanding User Behavior in Large-Scale Video-on-Demand Systems,” Proc. EuroSys Conf., 2006.

[174] C. Griwodz, M. Bar, L.C. Wolf, "Long-term Movie Popularity Models in Video-on-Demand Systems or, the Life of an On-Demand Movie," Proc. Multimedia 97, ACM Press, New York, pp. 349-357, 1997.

[175] S. Orlowski, M. Pióro, A. Tomaszewski, R. Wessäly, "SNDlib 1.0-Survivable Network Design Library", Networks 55:3, pp. 276-286, 2010.

[176] D. Xu, S. S. Kulkarni, C. Rosenberg, H.-K. Chai, "Analysis of a CDN-P2P Hybrid Architecture for Cost-Effective Streaming Media Distribution", Multimedia Systems, vol. 11, no. 4, pp. 383-399, 2006.

[177] M. Tran, W. Tavanapong, "Peers-assisted Dynamic Content Distribution Networks", IEEE Conference on Local Computer Networks , Sydney, Australia, 2005.

[178] H. Yin, X. Liu, T. Zhan, V. Sekar, F. Qiu, C. Lin, Hui Zhang, B. Li, "LiveSky: Enhancing CDN with P2P", ACM Transactions on Multimedia Computing, Communications, and Applications (TOMCCAP), vol. 6, Issue 3, August 2010.

[179] N. Laoutaris, P. Rodriguez, L. Massoulié, "Echos: edge capacity hosting overlays of nano data centers", SIGCOMM computer communication Review 38 (1),(2008).

[180] J. He, A. Chaintreau, C. Diot, "A performance evaluation of scalable live video streaming with nano data centers". Computer Networks 53(2): 153-167, 2009.

[181] M. Cha, P. Rodriguez, S. Moon, J. Crowcroft, "On Next-Generation Telco-Managed P2P TV Architectures", Proc. the 7th International Workshop on Peer-to-Peer Systems (IPTPS '08), Tampa Bay, Florida, USA, 2008.

[182] V. Valancius, N. Laoutaris, L. Massoulié, C. Diot, P. Rodriguez, "Greening the internet with nano data centers", Proceedings of the 5th international conference on Emerging networking experiments and technologies (CoNEXT '09), Rome, Italy, 2009.

[183] A. Nafaa, S. Murphy, L. Murphy, "Analysis of a Large-Scale VOD Architecture for Broadband Operators: A P2P-Based Solution", IEEE Communications Magazine, vol. 46, issue 12, 2008.

[184] B. Krishnamurthy, J. Rexford, "Web Protocols and Practice: HTTP/1.1, Networking Protocols, Caching, and Traffic Measurement", Addison-Wesley, Reading, MA, USA 2001.

[185] M. Rabinovich, O. Spatschek, "Web caching and replication", Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA, 2002.

[186] S. Podlipnig, L. Böszörmenyi, "A survey of Web cache replacement strategies", ACM Computing Surveys (CSUR), Volume 35 Issue 4, December 2003.

[187] M., Abrams, C. R. Standridge, G. Abdulla, S. Williams, E. Fox, 1995, "Caching proxies: Limitations and potentials", In Proceedings of the 4th International World Wide Web Conference, 1995.

[188] C. C. Aggarwal, J. L. Wolf, P. S. Yu, "Caching on the World Wide Web", IEEE Trans. Knowl. Data Eng. 11, 1 (Jan.), 94–107, 1999.

[189] S. Hossini-Khayat, "Investigation of generalized, caching". Ph.D. dissertation, Washington, University, St. Louis, MO, 1997.

[190] M. F. Arlitt, R. J. Friedrich, T. Y. JIN, "Performance evaluation of Web proxy cache

replacement policies", Tech. rep. HPL-98-97(R.1), Hewlett-Packard Company, Palo Alto, CA, 1999.

[191] C.-Y. Chang, T. McGregor, G. Holmes, "The LRU* WWW proxy cache document replacement algorithm", In Proceedings of the Asia Pacific Web Conference, 1999.

[192] J.-M. Menaud, V. Issarny, M. Banatre, "Improving effectiveness of Web caching", in recent Advances in Distributed Systems. Lecture Notes in Computer Science, vol. 1752. Springer-Verlag, Berlin, Germany, 375–401, 2000.

[193] Zheng da Wu, Chris Noble, and D. Huang. "Optimal Video Distribution Using Anycasting Service", Information Technology papers, 1999.

[194] S. Ait Chellouche, D. Négru, E. Borcoci, E. LeBARS, "Anycast-based Context-Aware Server Selection Strategy for VoD Services", in Proc. of IEEE GLOBECOM Workshops (GC Wkshps), Miami, Florida, 2010.

[195] S. Ait Chellouche, D. Négru, E. Borcoci, E. LeBARS, "Context-Aware Distributed Multimedia Provisioning based on Anycast Model towards Future Media Internet", in Proc. of 8th IEEE Consumer Communications and Networking Conference (CCNC), Las Vegas, NV, 2011.

[196] S. Ait Chellouche, D. Négru, Y. Chen, "Home-Boxes Support for an Efficient Video-on-Demand Distribution", Proc. the IEEE International Conference on Multimedia and Expo (ICME'11), Barcelona, Spain, 2011.

[197] S. Ait Chellouche, D. Négru, Y. Chen, "Home-Box-assisted Content Distribution Network for Internet Video-on-Demand Services", submitted to IEEE International Conference on Communications (ICC2012), Ottawa, Canada, 2012.

[198] S. Ait Chellouche, D. Négru, Y. Chen, "Home-Box-assisted Content Distribution Network for Internet Video-on-Demand Services", Technical report, 2012.

[199] S. Ait Chellouche, D. Négru, "Context-aware multimedia services provisioning in future Internet using ontology and rules", submitted to the Tenth International Conference on Pervasive Computing (PERVASIVE 2012), Newcastle, UK, 2012.