Programme ANR VERSO

Projet VIPEER

Ingénierie du trafic vidéo en intradomaine basée sur les paradigmes du Pair à Pair

Décision n° 2009 VERSO 014 01 à 06

du 22 décembre 2009

T0 administratif = 15 Novembre 2009

T0 technique = 1er Janvier 2010

Livrable 1.1

Analysis of the value chain and business models

Auteurs :

G. Simon, Z. Li, A. Gravey (Telecom Bretagne), F. Guillemin, S. Moteau (France Telecom), D. Alliez (NDS), F. Albanese (Eurecom)

Compilé par :

F. Guillemin (France Telecom)

Décembre 2010

Telecom Bretagne ; Eurecom ; INRIA ; France Telecom ; NDS ; ENVIVIO

Résumé :

Ce livrable identifie le sujet d'étude du projet VIPEER, soit la distribution de contenus vidéo dans le canal Internet. Les acteurs de ce service sont identifiés, ainsi que la chaine de valeur associée à la distribution de contenus dans le canal Internet. Le rôle central des réseaux de distributions de contenus est mis en évidence, et les raisons pour lesquelles il est nécessaire de revisiter les options d'architecture actuelles sont identifiées. La notion de réseau de distribution de contenu distribué (dCDN) est introduite dans ce contexte. Enfin, les avancées technologiques et les principaux projets de recherche en relation avec cette problématique sont identifiés.

Mots-clés : Distribution de contenus, Réseau de Distribution de Contenus (CDN), Chaîne de valeur, ISP

Table des matières

Table des figures

1. Introduction

Content delivery systems are becoming a key challenge for network operators and service providers. As a matter of fact, the ever-growing consumption of voluminous content by end-users, notably video on demand (YouTube, Dailymotion, etc.) and TV via the Internet, dramatically increases the need for high bandwidth in networks. This need is especially critical on peering links between Autonomous Systems. These links are indeed natural bottlenecks supporting all types of traffic and can significantly degrade the quality perceived by end-users. In an Autonomous System, the bandwidth available to carry traffic is in general sufficient (assessed by using simple over-dimensioning rules) except at the edge of the network (typically the ADSL access line and the radio access channel). The "last mile" is in general the second congestion point, after peering links, along the path of a stream of packets traversing a sequence of Autonomous Systems.

When considering services and especially video, we have to distinguish between two cases:

- "Walled garden" services (classical IPTV services and Video on Demand): a network operator, which fully controls the delivery of content by allocating specific resources, manages these services. In general, service platforms are connected to the network of the operator, which carries the corresponding volume of traffic by allocating a sufficient amount of resources. In order to cope with bandwidth demand for TV, multicast is in general used so as to decrease the bandwidth needed in the core of the network. Multicast can be used down to the DSLAM in current architectures so that the link between the end-user gateway (e.g., the Livebox) and the DSLAM remains the only critical point in the architecture. No caching systems are used for these services in general but with new usages (Catch up TV, delinearised TV), CDS systems may rapidly be introduced.

- Services in the "best effort channel": these services are all multiplexed in a unique transmission channel and compete for bandwidth. The quality offered to these services is much more difficult to monitor. Specific priority mechanisms can be implemented to differentially improve the quality of some services against some others. Since no specific resources are allocated to this type of service, Content Delivery Systems are highly desirable in order to reduce potential congestion on peering links.

In the remainder of this document, we shall consider services in the "best effort" channel. Because of the current lack of inter-domain QoS due to excessive competition between network operators and the saturation of peering links, Content Delivery Systems (including Content Distribution Networks) are hot topics for both network operators and service providers. The aim of Content Delivery Systems is to bring content closer to the user in order

- Alleviate the load of peering links.

- Reduce the access time to the content.

- Improve the Quality of Service (reduction of packet loss and delay because information do not cross anymore saturated peering links).

In reality, content providers are pushing operators to directly connect their servers to national networks so that they can offer their services with good quality by avoiding peering links and poor transit networks. Perceived quality is one important parameter for network operators to do so. Google is well known for applying this policy. The servers of content providers are then updated by using overlay networks.

Several **architectures** can be envisaged to implement Content Delivery Systems:

- **Overlay:** A Content Delivery System (CDS) can be implemented as an overlay network interconnecting servers installed in various Autonomous Systems. The content provider thus operates the CDS. The servers of the CDS are updated according to some policies implemented by the content provider. The network of the content provider carries content. The servers of the CDS are either national (e.g., one server per country, possibly connected to several ISP of the same country) or regional (e.g., on the PoP of a network operator). While the backbone links of the CDS are under the responsibility of the content provider (possibly via leased lines or Virtual Private Networks spanning over several networks), the "last mile"(including the access network and the collect network) is under control of the local operator. This is the "weak" point of this architecture and some content providers are willing to operate the links to the end-users (e.g., Google deploying FTTH). Another weak point of this architecture can appear if various networks support the network of the CDS since there is no inter-domain QoS in the current Internet. While this situation may not be critical for delay tolerant content, this may raise some issues for delay sensitive content. The update of the servers is an incompressible overhead of the architecture. An advantage of this architecture is that content is delivered by a single entity (the content provider) and thus via adequate agreement, there is no copyright issues.

- **In-network CDN:** A CDS can be local to a network operator. In that case, the network operator controls the CDS system, which is updated according to the requests, observed on the fly by the network operator or via an indirection procedure between the content provider and the network operator. When requests are observed on the fly and content cached according to the frequency of requests, cached content must be not copyrighted. When there is an agreement between the content provider and the network operator, then there is no copyright issue. The update of the CDS is under the responsibility of the network operator. In that case, it is preferable that the network operator owns a transit network to be able to carry traffic from the source to the servers of the CDS in case of delay sensitive content.

- **In-network storage capacities:** The third model is when the network operator offers storage capacities, which can be the basic building blocks of a CDS, which is operated by a content provider. Such a possibility is now feasible with the emergence of new technologies such a virtualization. A same machine (e.g., a server) can be sliced in such a way the different slices are completely separated and appear as dedicated resources available for a given task.

- **Peer to peer:** The storage capacities of end-users are used to cache and disseminate content. This model can be considered for copyright free material only or under some special agreement with the content provider. Cooperation between peers and the network could be envisaged either via information provided by the network (e.g., ALTO, p4p) or via the participation of the network in the peer-to-peer network (e.g., an indexation server or a tracker installed in the network). See Annex B for details.

The above architectures are in line with the current scenarios envisaged for content delivery. Servers of the CDS are placed relatively "high" in the network, namely in a PoP at the edge of collect IP network (e.g., the RBCI of France Télécom) or directly connected to an interconnection router. In that case, the path between the servers of the CDS and the end-user is still a critical point where QoS can significantly be degraded.

The objective of this document is to study how distributed CDN (dCDN) can help improve the global quality in the "last segment" between the customer premises and the edge of the network.

For dCDN, the same architectural principles could be used but a dCDN should be imbedded deeper in the network. Contrary to the classical CDN, which rely on servers installed relatively high in the network, dCDN are intended to be disseminated close to the end-user, for instance in DSLAM, routers/switches of the collect network and maybe in boxes hosted by the end-users (e.g., home gateways or set top boxes). **A dCDN should be seen as an imbedded content delivery system**.

To conclude this section, let us mention that a new paradigm is emerging for the dissemination of information: Content Centric Networking. In that case, content is divided into pieces stored in routers (from chunks of several megabytes to single packet of a few hundreds of bytes). Information is completely disseminated in the network. This can be viewed as an extreme case of distributed content distribution. In the following, we consider that content is disseminated in the last miles and not throughout the whole of the network.

2. Actors involved in the content distribution chain

2.1 Internet driven description of actors

To analyze the value chain in the delivery of content, we first have to identify the different actors playing a role.

- **The content producer:** the entity creating the content (e.g., a single individual archiving a movie on a YouTube server, a studio producing movies, etc.). A piece of content can be copyrighted. This may raise intellectual property issues.

- **The Content provider:** the entity in charge of making available a piece of content to end-users.
- **Service provider:** the entity offering services on the top of the Internet.
- **The Content Delivery Network** CDN**:** the entity in charge of disseminating the pieces of content (e.g., replicating the pieces of content).
- **The Internet Service Provider (ISP):** the operator providing Internet services to the end-user, in particular connectivity to the Internet.
- **Internet Transit Provider (ITP):** the actor interconnecting the networks of various ISPs.
- **Internet Access Provider, (IAP):** the network operator in charge of connecting the end-user to the ISPs.
- **The end-user:** the entity purchasing services and accessing content.

The roles identified above are often mixed in the current Internet:

- CDN can realize some functions of Tiers 1 and Tiers 2 operators (ITPs).
- Many ISP are also IAP. This is the case of the major network operators in France, including Orange.

The current debate about Network Neutrality aims at separating the roles of **network operators** and **service providers** so that the competition between the services offered by the two types of actors is fair and unbiased. But this debate is far to be closed since, amongst other reasons, network operators are at the best place to offer services with controlled quality. If quality is an incentive for users to purchase a service, then network neutrality cannot be naïvely applied. The only requirement to put on a network operator is to offer interfaces so that OTT service providers can also offer services with quality.

Beyond the debate about network neutrality and the mixed role of ISP and network operator, the diffusion of contents such as TV channels through the Internet introduces new roles because the diffusion of contents is technically more complex via the Internet than through the air. In the classical TV model, the actors are:

- the content producer,
- the content provider (the TV channel),
- the TV broadcaster (e.g., TDF in France).

When TV and audiovisual contents are distributed on the Internet, ISP and IAP play the role of "TV broadcaster". This is already the case for walled garden services (e.g., Orange TV). For services in the best effort channel, the situation is more complex. Because audiovisual services put much more pressure on infrastructures through the increase of bit rates, Content

Delivery Operators (e.g., Akamai) are progressively led to play an increasingly important role.

2.2 TV Driven description of actors

For the sake of completeness, we recall in this section the various actors in the classical TV distribution. A possible definition of the actors is as follows:

- **Content owner**: the entity that owns the media asset (has copyright)- producers, authors.
- **Content distributor**: the entity who sales the assets (has a mandate to sell) - movie studio, local representatives.
- **Publisher**: the entity in charge of aggregating various contents, typically a channel – TV channels
- **Service provider:** a network / channel lineup (with package & exclusivity deals) – e.g., Universal / discovery / mtv networks.
- **Service distributor:** the company that charges final users for subscription / collecting transactions (can be a retailer in some cases) – e.g., Canalsat.
- **Access network provider**: the company that own/manages the digital pipe delivering content to final users (DTT or DTH operation, but could also typically be an ISP) > eutelsat
- **Customer Electronics vendor.**

Obviously, some players are more or less "vertically integrated", and with time fewer and fewer market players are "pure players":

- Some ISPs are playing all roles (e.g., Orange), but mostly focused on service provider, service distribution and access network.
- Some other like Canal+/Canalsat are service and distribution centric, relying on others to provide the access network

3. CDN Business Model: a perspective

CDN business was created to deal with the growth of traffic related to the development of Internet. Reliance issues prominently drove the first steps toward distributed hosting of web services. Distributed hosting was further driven by the growth of Internet itself, for data volumes handling purposes. As more users were connected, websites had to manage a huge increase of traffic. As the amount of traffic over the Internet exploded, the needs of two families of players in the value chain converged:

- On the one hand, the demand for distributed hosting by services providers was increasing: Internet services had to be distributed for redundancy and scalability reasons.

- On the other hand, ISPs had to find ways of minimizing traffic from their network to foreign networks and vice versa while still fulfilling the requests of their own users. The main focus was to reduce "transit" traffic, in order to keep infrastructure costs under control.

From this pressure, a new category of market players emerged: CDN (Content Delivery Network). Taking into account the position of content providers and ISP on the market, the "magic trick" of CDN has been to build a dual-sided business:

- On the one hand, CDN would provide ISP and hosting capacity providers with caching capacities "as a service". Typically, CDN would invest and run caching capacities, strong of a specific vertical knowledge, and service them "inside" ISPs infrastructure or at the transit edge of it. To reward CDN for this investment, ISPs would allow CDN to use "a share" of serviced capacity to build "their own business" ;

- On the other hand, CDN would provide content providers with a "distributed hosting capacity" offering hard to match characteristics. They would provide pretty exclusive scalability capacities and extreme flexibility to web services owners. CDN would deal with distribution complexities, managing multiple ISP referencing, and dynamically manage content propagation as unique selling points to the market.

As such, matching a clear market need, CDN business grew fast. Key to this growth was the "selling mode" of CDN services: instead of renting shares of serviced infrastructures, CDN choose to match hosting principles, pricing their services according to the volume of traffic served – the main metric to measuring this being the total in Gigabytes of data transferred. This charging logic made the success of CDN business. Matching services provider's acceptance of such a variable pricing, CDN gave rise to a (very) virtuous business: key success factors for market players of this industry were their capacity to invest and deliver fast. Return of investments and rise in traffic provided significant growth.

As CDN business flourished, this market saw significant concentration over the past few years, to reach a high level of investment but also to allow market players to keep a critical mass as the all Internet market evolved and major services providers grew in size.

Four major market evolutions have changed this situation:

- First, as the ISP market became mature and reached denser and more complex levels of dependencies in peering with other networks and as transit costs drastically decayed (with the development of direct peering between ISP), CDN "market need" stabilized;

- Second, as the amount of traffic from CDN cache servers increased, bottlenecks appeared inside ISP infrastructures, at core or backhaul network level. To overcome this problem, ISP started to put restrictions on CDN generated traffic, raising variable overlooked costs, such as:

    - Bandwidth used (output from CDN infrastructure to ISP network), often charged and thus implying a variable cost structure for CDN,

    - Diminishing the share of cache storage allocated to CDN business impacting investment level required to run the service (need to provide more storage to ISP for a given CDN capacity allocation);

- Third, Internet Transit Providers (ITPs) jumped into the CDN business, aggressively pressuring pricing of incumbents (e.g. Teliasonera in Northern Europe). ITP, benefiting from very different costs and revenues structures, had to react to the development of direct peeing between ISPs and the development of caching capacities impacting their core revenues (transit). They did so by lowering drastically CDN pricing to gain market shares, lowering at the same time global market revenues potentials of CDN market at large;

- Fourth, the nature of traffic evolved: from web and asynchronous applications, Internet evolved toward video, low latency, real time services. Some of those applications are audiovisual services such as catch-up services, video on demand, web TV, or User generated content. This significantly changed the shape of global Internet traffic in terms of type and global weight evolutions over time, as well as in terms of global quality of service requirements. This impacted CDN business by requiring specializations, raising corresponding costs with it (license fees). In particular, operational algorithms that were optimized for serving html pages are not optimized for video.

In order to cope with huge cut in revenues due to competition and significant change in costs structures, CDN businesses are today challenged to reinvent themselves. The original need of a third party between ISP and Services Providers is also challenged: as bottlenecks moved deeper into ISP infrastructures, CDN are now considering servicing their technological expertise rather than running services themselves, in the context of so called “dCDN” architectures. The future of CDN businesses is likely to live deeper into ISP networks, more integrated into and interleaved with ISP infrastructures.

That deeper integration raises a key investment issue: provided that complexity of ISP network is increasing, going deeper toward the final user, solutions are requiring major evolutions in the “build” and “run” phases of such systems:

- In terms of amounts/scalability of key components (investment and running costs, durability).

- In terms of flexibility with respect to traffic evolution (QoS at large).
- In terms of serviceability of components (localization, maintenance, support).

Key challenges are to better characterize traffic evolution to better match market needs. Another key will be to question current infrastructures components and their potential evolutions on network operator, content delivery operator, ISP or final user's side.

Besides architectural aspects, some further constraints appear when examining the value chain: most prominent are the conditions of operation of such resources – and thereby the players potentially involved in the funding of such dCDN services. Current debate about net "neutrality/quasi-neutrality" will interfere with this business goal, potentially affecting global value chain by applying legal constraints. In the absence of regulation, the positions of players in the value chain are pretty clear at this stage, and must be considered in building the targeted architectures:

- Services providers:
    - Want to be able to address any customers without having to engage under specific deals with network operators (out of exclusivities or potential syndication deal to be also considered as particular use cases).
    - Want to drastically lower their cost structure by any means:
        - Positively,
            - by assuming a share of the required investment if this leads to build non-variable costs structures,
            - or by respecting distribution formats to match network operator handling preferred mode of content delivery (as the current development of HTTP adaptive streaming attests).
        - Negatively by pressuring ISP to deliver revenues back in relation with provided services popularity.
    - Need to be able to build meaningful offering at "national level", especially in Western Europe, to face global players competition:
        - Being mostly financed by incumbent, content industry must find some balance to build profitability in constrained market.
        - National players must find a way to exist over Internet to balance international players, especially when cultural

diversity is at stake (both in application and content services).

  - Want to be able to deliver highest quality of service to match their branding/service efforts.

- CDN providers:
  - Want to remain in the game by returning to more healthy level of profitability:
    - Getting rid of variable costs,
    - Find further added value in content handling to collect more revenues.
  - Want to participate to global infrastructure investments, to better balance and value their core assets (technologies, knowledge and current infrastructures);

- ISP:
  - Want to maintain costs structures under control to remain competitive on Internet access market (i.e. maintaining low pricing/level of margin).
  - Want to be able to control global QoS of services and maintain a certain level of differentiation.

- End-users
  - Want "freedom" to be able to access any service from anywhere in the world, at any time.
  - Want to get the best "value for money" (considering the current balance is highly in their favor).

This historical perspective and rough business needs are making clear that many market player objectives converge in spite of disparities. Some divergences obviously still exist but the global interest of the value chain player is converging toward better collaboration to match Internet evolution at large and its impact on a wider scope of industries such as the content industry. This project explores path of potential collaborations to enable next generation of distributed services. Path to this collaboration will be to negotiate a consensus between value chain players.

4. Analysis of the value chain

4.1 Classical data delivery chain

The classical delivery chain of data through the Internet is illustrated in Figure 1. Data are produced by the Content Producer and are then handled by the Content Provider. This latter is client of the Service Provider, who packages the data to offer services subscribed by the End-user. The data are then delivered to the Internet Service Provider and are transferred through the network (transit, ISP, IAP) up to the End-user. This chain describes the role of the various actors.

When a CDN is used to deliver content, the Service Provider transfers the data to the CDN for delivery to the End-user through the ISP and maybe directly through the Internet Transit Provider.

It is also worth noting that the current trend is to merge the two actors Service Provider and Content Provider in order to offer rich services. For instant, service providers as YouTube or Dailymotion are currently both Service Provider (movie delivery) and Content Provider (hosting the movies).

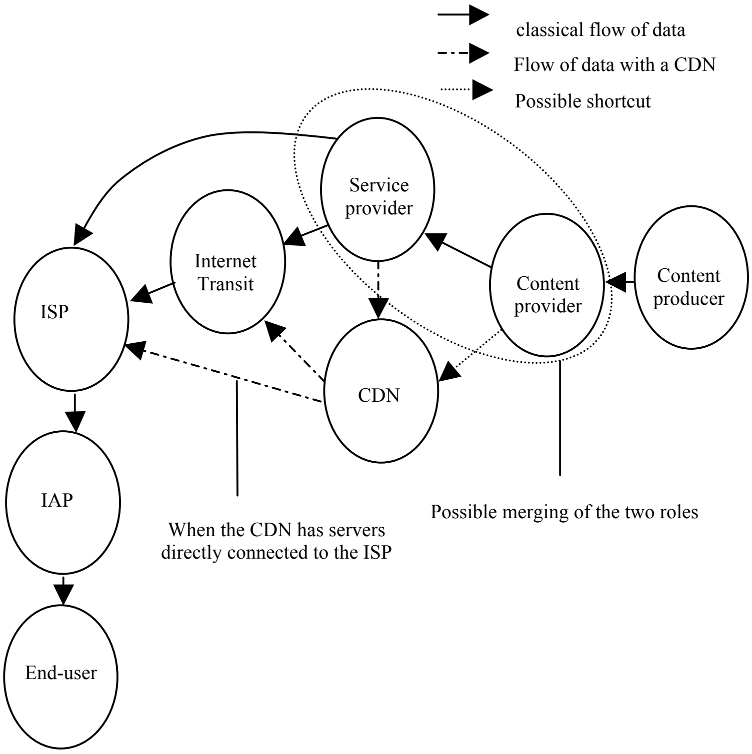

Figure 1 : Data flows in the classical delivery case with or without CDN

4.2 Business relationships in the case of a CDN

The business relationships between the various actors are illustrated in Figure 2 when there is no CDN. The End-user pays the ISP for accessing Internet services. When the ISP is not the IAP, then the former pays the latter for using the access infrastructure. The ISP also pays the Internet Transit Provider for global connectivity. It is worth noting that an IAP is often an ISP and that the flow of money between the ISP and the IAP is only virtual. The same remark hold when the ISP is also an ITP This is notably the case of Orange, which is an IAP, an ITP and an ISP.

Once connected, the End-user can access services offered by the Service Provider and can afford services. However, the End-user does not always pay the Service Provider and the Content Provider. These two actors very often earn money via advertising. Business related to advertising is completely orthogonal to the considered content delivery chain. This is by the way a major issue for Service Providers and Content Providers to find a business model (e.g., for YouTube). In addition, in the case of User Generated Content (typically for YouTube), the Content Provider does not always pay the Content Producer.

To conclude the description of the classical delivery chain, note that the business relationships between ISP and ITP are more complex because of peering agreements and depend on their rank in the Internet hierarchy (Tiers 1, Tiers 2, etc.).

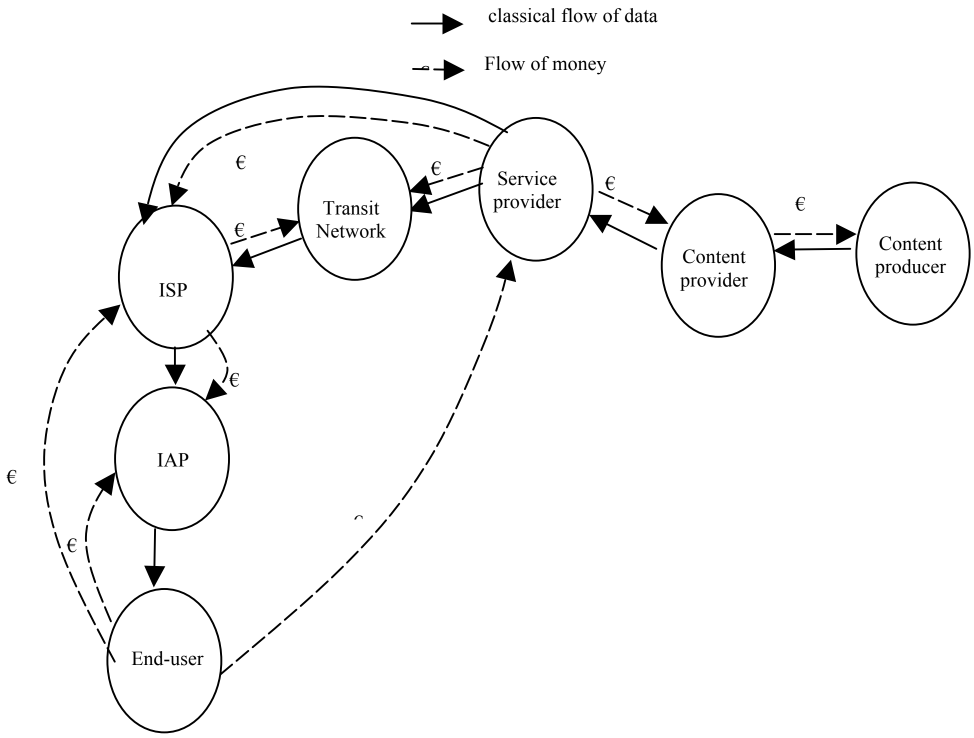

Figure 2 : Business Relationships between the various actors (no CDN)

Now, when we consider the presence of a CDN, the above business relationships are slightly modified, see Figure 3. In the classical scenario, the Service Provider pays the CDN for delivery of the data generated by services. A possible shortcut consists for the Content Provider to directly transfer Content through a CDN and hence to eliminate the Service Provider from the delivery and business chain. This would lead to raw Content delivery not packaged into services.

The flow of money between the ISP and the CDN takes place only if the CDN has servers connected to the ISP infrastructure. Some CDN may have servers connected to transit networks only. In that case, there is a flow of money between the two parties.

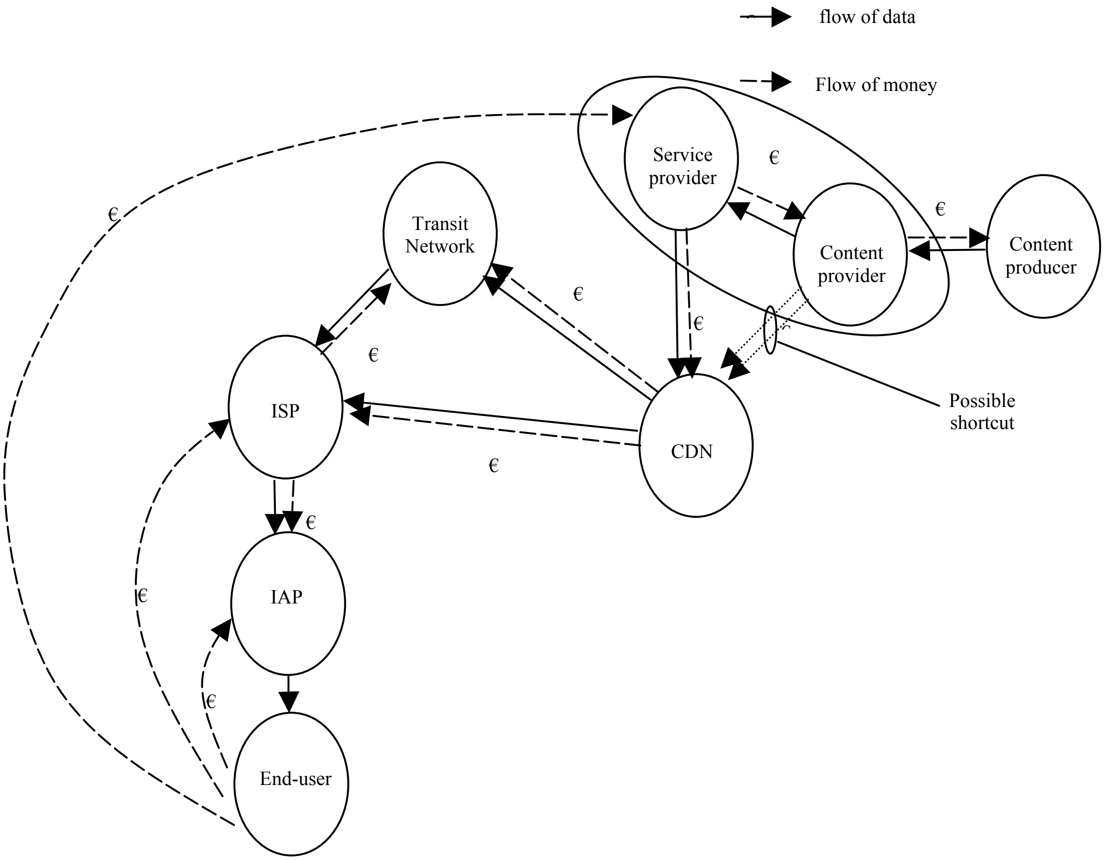

Figure 3 : Business relationships between the various actors in case of CDN

The classical chain described in Figure 3 may slightly change in the near future. As a matter of fact, CDN operators argue that ISPs save money by reducing the amount of traffic received via peering links and what they pay to ITP (for instance Tiers 1 operators). This is an opportunity for CDN to be paid back because of these money savings. The fact that there is no inter-domain QoS, CDN can use routing to degrade the QoS of the End-user of an ISP and thus force an ISP to either install a CDN server in its own infrastructure or to pay back the CDN. This raises the issue of Content Neutrality.

4.3 The case of distributed CDN (dCDN)

In the current architectures, CDN are based on servers disseminated in several networks. Most large CDNs have servers connected to local operators in many countries. This allows a significant improvement of the quality of service since flows do not transit throught a series of networks from the server hosting the content and the end-user. This also leads to a reduction of the load of peering links and has the benefit of improving the quality of other services. To solve possible QoS issues in the last miles, dCDN are foreseen as a solution to put the content closer to the end-user. In that case, instead of having a national server, there are a number of storage capacities disseminated in the infrastructures of the ISP and the IAP.

For business relationships, two possibilities can be envisaged:

- The ISP and the IAP (often the same administrative entity) operate the dCDN. This solution presents the advantage to offer efficiency in the delivery of content since the ISP knows the topology of its own network (including link capacities). A CDN thus pushes contents to the dCDN and does not have a precise knowledge of how it is implemented. This is close to a cloud computing approach (see next section).

- The dCDN is divided into various computing and storage capacities and are then used by the CDN operator to distribute content as close as possible to the end-user. This should be made possible in the near future thanks to virtualization techniques (see next section). This is a means for the ISP and IAP to monetize its infrastructure.

Depending on the chosen option, the interfaces offered by the dCDN to the CDN are not the same but the precise management of the content raises the same issues: level of granularity, knowledge of the available bandwidth, popularity of the content, etc.

In the first case, the CDN pays the dCDN for managing the content down to the user. In the second case, the dCDN is operated by the CDN, which has to afford storage and computing capacities from the IAP, the ISP, and maybe the ITP.

5. Technical changes

In this section we describe some recent technological advances, which open the doors to the implementation of new services and network capabilities.

5.1 Open platform: the case of Juniper

Juniper through the Junos SDK product offers an opportunity of designing, developing, and deploying specialized applications that partners or operators can then publicly monetize. This flexibility enables multiple Junos SDK applications to be uploaded; they can interact with each other, with other Juniper products, and with network servers in real time. In addition, the Junos SDK allows customers and partners to incorporate their own unique application that runs on the Juniper platform.

Follows some extracts about this topic from Juniper's web site [4].

- "Operator networks are in the midst of a transformation to IP that is leading to the convergence of service platforms within fixed and mobile networks. This transformation introduces technical challenges for the network and is clearly challenging the service and operational domains as well. We are just beginning to see how operators can deliver services that may be shared, personalized, and monetized, resulting in unexpected innovation and application creation from system vendors, network operators, content providers, and end-users. New tools, however, are required to deliver this innovation, and new vehicles are also needed to accelerate the time to market of these applications. Juniper Networks has been pushing the boundaries of innovative design ever since it introduced the Junos SDK."

- "The Junos SDK enables developers to innovate on top of Junos and Juniper Networks platforms, so you can create, deploy, and validate innovative applications tailored to your needs."

“The types of applications written by our development community include:

- Intelligent agents that monitor network traffic and routing performance for real-time deterministic traffic QoS.
- Signaling protocols development, fostering exploration of enhancements to network performance by improving upon existing industry-standard signaling protocols.
- Applications that ensure service-level agreements are met by monitoring and managing specific types of traffic on the network, delivering the best possible QoS for applications like voice and video.
- Deep packet inspection applications to monitor network traffic, looking for security threats or other issues and taking corrective action as needed.
- Encryption and tunneling applications to satisfy specific needs of subscribers or local regulations.”

“Juniper's approach is to invite applications that are already developed to run within the routing code, enabling faster time to market for those applications as well as a solid platform on which to operate. The top operator networks that deliver video and voice services today are looking for these types of tools to assist them.”

5.2 Content Centric Network (CCN)

Today, the vast majority of the Internet usage is about content and services discovery& retrieval, content delivery and streaming and Web services access. The user cares only about the content or service itself and proper delivery while oblivious to their location. That is, the user knows that he/she wants news from the BBC, videos from YouTube or weather information, concrete and delivered in suitable quality and format, but does not know or care on which machine the desired data or service resides, as soon as reliability, security and privacy are guaranteed.

The content-centric view refers to the central role that rich media content is playing in attracting users to Internet services, as content consumers are increasingly also as content producers, and how the transfer of media content can impact the network operation.

The concept of CCN is that a communication network should allow a user to focus on the data he or she needs, rather than having to reference a specific, physical location where that data is to be retrieved from. CCN makes all network functions centered on content transport. For example, a request is based on the name of a movie, the network sends a list of such corresponding movies and then one of them is chosen. The network will then give the content without any request about a server. A large literature exists on this topic. For more details see [5], [6], [7] and [8].

5.3 Cloud Computing

The Cloud Computing principle is about using computing means installed everywhere in the world, by a web portal. Instead of having servers, software, networks, the user consume the needed resources via the network. The computer science become like TV or electricity: to have it at home, you do not need a transmitter in your living room or a power station in your cellar.

In this approach, the user or company do not need to have a computer room in its place. This is not new: Software as a Service (SaaS) is that for software. E-mail services like Google Mail, Yahoo or laposte.net are based on this way. The new thing is the possibility to access, with the same mean, to other applications different from e-mail services. A company rent time on servers and in-line storage (and so doesn't buy a server). It is the same with software: a company do not buy license and do not manage the update.

After software, data center infrastructure is concerned (including servers, networks, storage) and is named Infrastructure as a Service, (IaaS).

Between IaaS and SaaS another model, named Platform as a Service (PaaS), exists. It is for application developers where the user buys computer resources (of type IaaS) but too some additional tools to help him creating application and get it to the final user.

The set of these in-line resources, IaaS, PaaS and SaaS, are what we call Cloud Computing. For more details see [9] and [10].

Annexes

A. Content Delivery Networks

A.1 Introduction

The proliferation of the Internet has triggered the design of massive scale applications involving large numbers of users in the order of thousands or millions. Such scenario may cause unmanageable levels of traffic for content providers that employ the classical client-server model, where a centralized web server exclusively serves its content to clients. Possibly, the web server gets overwhelmed with traffic due to a sudden spike in its content popularity, resulting in a temporary content unavailability or degraded delivery performance.

In order to improve Internet service quality, Content Delivery Networks (CDNs) have emerged to efficiently deliver content to a large audience. A CDN is a collection of network elements arranged for more effective delivery of content to end-users: they offer fast and reliable applications and services by distributing content to proprietary servers located close to users [35]. Indeed, a CDN provides better performance through caching (or replicating) content over some servers strategically placed at various locations in order to prevent possible flash crowds. Moreover, the users are redirected to the surrogate server nearest to them, enhancing the response time to users requests.

The three keys actors in a content distribution context are:

- content provider
- CDN provider
- end-users

A *content provider* or *customer* is one who delegates the URI name space of the Web objects to be distributed. The *origin server* of the content provider holds these objects. A *CDN provider* is a proprietary organization that provides infrastructure facilities to content providers in order to deliver content in a reliable and timely manner. *End-users* or *clients* are the entities who access content from the content provider's website.

CDN providers replicate the content using *caching* and/or *replica* servers located in different geographical locations: CDN servers are commonly called *surrogates* or *edge servers*. Client requests are routed to the nearby surrogate, and a selected surrogate server delivers requested content transparently to the end-user.

A CDN includes typically the following functionalities:

- *Content outsourcing and distribution* to cache and/or replicate content to surrogates on behalf of the origin server
- *Request redirection and content delivery* to direct a request to the closest suitable surrogate

- *Management services* to manage and monitor the network components, to handle accounting, and to keep track of content usage

A.2 Key issues in CDNs

In this section we discuss the key issues to take into account while dealing with the design of a CDN system.

A.2.1 Surrogate placement

Since location of surrogate servers is closely related to the content delivery process, the issue of choosing the best location for each surrogate is critical: indeed, the goal of optimal surrogate placement is to reduce user perceived latency for accessing content and to minimize the overall network bandwidth consumption for transferring replicated content from servers to clients. In this context, some theoretical approaches such as *minimum k-center problem* [25], *k-hierarchically well-separated trees* [15] have been proposed. Due to computational complexity of these algorithms, some heuristics such as *Greedy replica placement [30]* and *Topology-informed placement strategy* [19] have been developed: the first requires the knowledge of the clients locations in the network and all pair wise inter-node distances, while the latter takes into account the outgoing degrees of each node - i.e. the number of other nodes connected to a node.

Other approaches include algorithms like *Hot Spot* [38] and *Tree-based replica placement* [32]; the hotspot algorithm places replicas near the clients generating greatest load, whereas the tree-based replica placement algorithm is based on the assumption that the underlying topologies are trees. Another example is *Scan* [18], which is a scalable replica management framework that generates replicas on demand and organizes them into an application-level multicast tree.

For surrogate server placement, the CDN providers also determine the optimal number of surrogate servers using *single-ISP* and *multi-ISP* approach [43]. In the *Single-ISP* approach, a CDN provider typically deploys at least 40 surrogate servers around the network edge to support content delivery [21]. The policy in a single-ISP approach is to put one or two surrogates in each major city within the ISP coverage. In *Multi-ISP* approach, the CDN provider places numerous surrogate servers at as many global ISP Points of Presence (POPs) as possible; some large CDN providers such as Akamai has more than 20000 servers deployed in various ISPs on a global scale [24]. Estimation of performance of these two approaches shows that single-ISP approach works better for sites with low-to-medium traffic volumes, while the multi-ISP approach is better for high-traffic sites [43].

A.2.2 Content management

Content management is essential for CDN performance in order to minimize the perceived latency on the user side - enhancing the hit ratio - while minimizing the access to the origin server. Both replication and caching can be applied to manage the content in a CDN.

According to [41], replication involves creating and permanently maintaining duplicate copies of content on different nodes. Replication in a CDN is typically initiated when the origin server pushes content to any surrogate server. The surrogate servers then manage the replication of the content among each-other, either on-demand or beforehand.

In on-demand replication, the surrogate server that has received a query and experienced a cache miss pulls the requested content from the origin web server or other surrogate servers. Beforehand replication implies different strategies that replicate objects a priori and dynamically adjust their placement in a way that brings them closer to the clients and balances the load among surrogate servers [36]. Moreover, under this approach a greedy-global heuristic algorithm is suitable for making replication decision among cooperating surrogate servers [26]. However, due to replication requirements in terms of cost and time, any replica placement should be static for a large amount of time.

Given that popularity of objects may fluctuate with time, some replicas may become redundant and unnecessary. This leads to unoptimized storage management at surrogate servers. That is why caching can be seen as an interesting alternative to replication, especially in cases where unpredictable numerous users have suddenly interest in the same content. In this case, content objects are dynamically cached and evicted from the cache according to cache replacement and update policies. Examples of cache update policies are given in [45]:

- periodic update,
- on-demand update,
- invalidation,
- update propagation.

Since the cache has a limited storage size, the server might need to evict cached objects via one of the cache replacement policies treated in [45]. The most famous of these policies is LRU, which replaces the rarely requested objects stored in the local cache with the new incoming objects.

An evaluation of caching and replication as separate approaches in CDNs is covered in [16]. If replication and caching cooperate they may greatly fortify the CDN since both of them deal with the same problem but from a different approach. Indeed, [41] has proved that potential performance improvement is possible in terms of response time and hit ratio if both techniques are used together in a CDN. CDNs may take advantage of the dynamic nature of cache replacement policies while maintaining static replicas for availability and reliability.

A.2.3 Requests routing

A request-routing system is vital to route the client requests to an appropriate surrogate server for the delivery of content. Indeed, request routing is needed to choose the closest surrogate that can best serve the request for each client request. Thus, a request-routing system relies on a set of metrics such as network proximity, client perceived latency, distance, and edge server load in an attempt to direct users to the closest surrogate that can best serve the request.

The request-routing system in a CDN has two parts: deployment of a request-routing algorithm and use of a request-routing mechanism. A request-routing algorithm is invoked on receiving a client request. It specifies how to select an edge server in response to the given client request. On the other hand, a request-routing mechanism is a way to inform the client about the selection. Such a mechanism at first invokes a request-routing algorithm and then informs the client about the selection result it obtains.

A.2.3.1 Routing algorithms

The algorithms invoked by the request-routing mechanisms can be *adaptive* or *non-adaptive*. Adaptive algorithms consider the current system condition to select a cache server for content delivery. Current condition of the system is obtained by estimating some metrics like load on the replica servers or the congestion of selected network links. Non-adaptive request-routing algorithms use some heuristics for selecting a cache server rather than considering the current system condition. The latter is easy to implement, while the former is more complex.

A non-adaptive algorithm works efficiently when the assumptions made by the heuristics are met. On the other hand, an adaptive algorithm demonstrates high system robustness [45] in the face of events like flash crowds.

Non-adaptive algorithms The most common and simple non-adaptive algorithm is round-robin, which distributes all requests to the CDN cache servers and attempts to balance load among them. Such algorithm is efficient for clusters, where all the surrogates are located at the same place [34].

Sivasubramanian et al. [40] propose another non-adaptive request-routing algorithm, in which all replica servers are ranked according to the predicted load on them: the prediction is done based on the number of requests each of the servers has served so far. This algorithm takes client-server distance into account and client requests are directed to the replica servers in such a way that load is balanced among them.

Karger et al. [27] have proposed a non-adaptive algorithm which calculates a hashing function *h* from a large space of identifiers, based on the URL of the content. This hashing function is used to efficiently route client requests to a logical ring consisting of cache servers with IDs from the same space. Variations of this algorithm have been used in context of intra-cluster caching [33] and P2P file sharing systems [14].

Adaptive algorithms Globule [37] uses an adaptive request-routing algorithm that selects the replica server closest to the clients in terms of network proximity. The metric estimation in Globule is based on path length which is updated periodically. The metric estimation service used in Globule is passive, which does not introduce any additional traffic to the network.

Andrews et al. [17] and Ardiaz et al. [13] have proposed adaptive request-routing algorithms based on client-server latency. In this approach, either client access logs or passive server-side latency measurements are taken into account, and the algorithms decide to which replica server the client requests are to be sent. It is important to notice that these algorithms require the maintenance of central database of measurements, which limits the scalability of systems on which these algorithms are deployed.

Akamai uses a proprietary adaptive request-routing algorithm [20]. It takes into consideration a number of metrics such as surrogate server load, the reliability of loads between the client and each of the surrogates, and the bandwidth that is currently available to a replica server.

A.2.3.2 Routing mechanisms

Request-routing mechanisms inform the client about the selection of surrogate server, generated by the request-routing algorithms. We list the most important mechanisms currently in use.

- **DNS-based routing.** In this approach [16] [43], the content distribution services rely on the modified DNS servers to perform the mapping between a surrogate server symbolic name and its numerical IP address. The performance and effectiveness of DNS-based request-routing has been examined in [39].
- **URL rewriting.** In this approach, the origin server redirects the clients to different surrogate servers by rewriting the dynamically generated pages URL links [21].
- **HTTP redirection** The approach propagates information about replica server sets in HTTP headers. HTTP protocols allow a Web server to respond to a client request with a special message that tells the client to re-submit its request to another server.
- **CDN peering** In this case, peer-to-peer content networks are formed by symmetrical connections between host computers. Content location and retrieval in CDN peering is highly dependent on the indexing abstraction of the peer-to-peer architecture: for example, a system like Chord [42] employs the concept of DHT to also perform request-routing.

A.3 Performance measurement

Performance measurement of a CDN is necessary to measure its ability to serve the customers with the desired content. [21] [29] give a list of typical considered metrics while dealing with content distribution:

- **latency** gives the user perceived response time,
- **cache hit ratio** measures the ratio of the number of already cached documents over the total number of documents requested, describing the effectiveness of the adopted caching policies,
- **surrogates utilization** refers to the load and resource consumption of each surrogate,

- **packet loss** measurements are used to determine the quality and the relative congestion of surrogates links,
- **reserved bandwidth** is the measure of the bandwidth used by the origin server

For performance measurement different network statistics acquisition techniques exist. Such techniques may involve network probing or traffic monitoring:

- **Network probing.** Network probing is a measurement technique where the possible requesting entities are probed in order to determine one or more metrics from each surrogate or a set of surrogates. An example of probing is given by [24], where the authors retrieve an enormous set of open DNS servers in order to run a King-based latency estimation of surrogate servers in two distinct commercial CDNs.
- **Traffic monitoring.** Traffic monitoring is a measurement technique where the traffic between the client and the surrogate is monitored to know the actual performance metrics. In [31] a latency estimation technique is described, which is based on the monitoring of TCP handshakes between clients and Google CDN surrogates.

B. P4P and ALTO

The method of exploiting information on the physical network to optimize P2P applications is referred to as Application Layer Traffic Optimization (ALTO). The IETF Working Group (WG) on this topic has been started in early 2008. The goal of the WG is to define a normalized service, which gives resource selection guidance to service clients, in order to improve both application performance and resource utilization of the underlying network. The detailed formulation of ALTO problem and its goal are available in RFC5693 [11].

This WG has defined a generic and open service, called ALTO service. This service is generic because it covers all applications involving resource selection, including but not limited to P2P applications. Typically, the ALTO service can also be used for cache/mirror selection. The ALTO service is also open: not only the network service provider, but also third parties or any user community can provide the information on the underlying network.

The ALTO service can be considered as an extension of an earlier work called P4P [12]. P4P has proposed a simple and flexible framework to enable cooperation between P2P applications and network providers. P4P relies on an entity called iTracker, which is controlled by the network providers. It is linked with both peers and application tracker to provide accurate information for the P2P service, such as network capability, distance among peers and operator policies.

The concept of cooperation between P2P and network providers is proposed by P4P and adopted by IETF ALTO WG. While P4P has given the proof-of-concept, ALTO WG aims at defining a protocol standard with a detailed interface. So far, two internet-drafts of system requirements and ALTO protocol have been issued (http://tools.ietf.org/wg/alto/).

The ALTO architecture is as follows. Similar to the iTracker in P4P, the ALTO server has access to network information, including routing protocols and operator policies. It can also receive information from third parties and content providers. Then, this information is sent to the ALTO client through the ALTO protocol. The communication protocol between ALTO server and ALTO client is the focus of the WG. A standard protocol is being defined: ALTO protocol uses HTTP with a RESTful interface, in order to leverage current HTTP implementations and infrastructure. ALTO protocol allows ALTO server to provide different services to ALTO clients. For example, providing a ALTO client with network map or properties of a given endpoint. ALTO also defines the message format, to guarantee both the interpretability for ALTO information user, and the anonymity of this information.

The ALTO service can be easily integrated into current P2P applications, in particular BitTorrent. The BitTorrent tracker can host a ALTO client. The tracker can thus download ALTO information from the ALTO server, and provide peer selection guidance to all registered peers. The ALTO client can be hosted directly by a peer. In that case, the peer connects to both tracker and ALTO server. In summary, ALTO provides an open and generic service, allowing information sharing between network providers and P2P applications.

The ALTO protocol is still in progress; some important issues need to be clarified:

- Cross-Domain links: ALTO protocol assumes that the ALTO server conveys the network information from the perspective of a network region. For this, it is possible for a ALTO server to receive information from several network providers. Every network provider is able to provide accurate information on its own network. The ALTO server needs to aggregate information from different sources, in order to obtain a view of the complete region instead of a set of independent networks. Especially, the ALTO server should be aware of cross-domain links, which is in general difficult to obtain.
- Scalability: ALTO adopts a client/server architecture. So the scalability issue on the ALTO server needs to be resolved: how many clients the ALTO server can support? How much network information it can store and keep updated?
- Security: Both the security of the architecture, e.g. preventing DoS attack against the ALTO server, and the security of the information, which guaranteeing information accuracy, need to be considered.

C. Related projects

C.1 NADA

NADA (http://www.nanodatacenters.eu/), which stands for nano-data-centers, deals with building a distributed infrastructure made of set-top-boxes (STBs) located at residential access sites. The underlying idea of the NADA architecture is to virtualize (high-end) STBs and instantiate a number of "slices" on top of them. Slices are subdivided into core-slices and application-slices. Core-slices take care of managing the infrastructure, measuring the status of the system, the network that connects the STBs, and the application-slices. Application-slices are instantiated with all sorts of applications: their deployment, elasticity, migration is open to third-parties.

In this solution, ISPs are now able to measure, assess the performance, and trouble-shoot their networks, and additionally to sell STBs resources to third parties, without paying additional costs of deployment. Such an architecture, which is the equivalent of a distributed version of cloud services such as EC2 (Amazon) or AppEngine (Google), has also the advantage of being less expensive in terms of energy consumption of equivalent systems running on monolithic data-centers.

In NADA, the focus is on building the architecture, and showcase two applications: multi-player on-line gaming and user data backup applications.

Further reference on NADA can be found in [1], [2], and [3].

C.2 OCEAN

OCEAN (http://www.ict-ocean.eu/) aims at finding solutions to the imminent problem of multimedia content traffic clogging up future aggregation networks, when the offer of online video of high quality over the Open Internet continues to increase.

- OCEAN will design a new open content delivery framework that optimizes the overall quality of experience of end-users by caching content closer to the user than traditional CDNs and by deploying network-controlled, scalable and adaptive content delivery techniques.
- The OCEAN architecture will clearly define light-weight signaling protocols and public interfaces between its major building blocks in order to foster multi-vendor solutions and contribute to cut down content delivery costs.
- OCEAN will elaborate business strategies providing better investment incentives to the different types of players in the value chain (content providers, Internet service providers, CDN service providers and industrials).
- OCEAN will build 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.
- New media-aware congestion control mechanisms based on slight, but controlled quality degradation will provide a better alternative than mere blocking of user requests.
- The validity and performance of these algorithms and mechanisms will be assessed through simulations, large-scale emulations and a trial in a real ISP network.

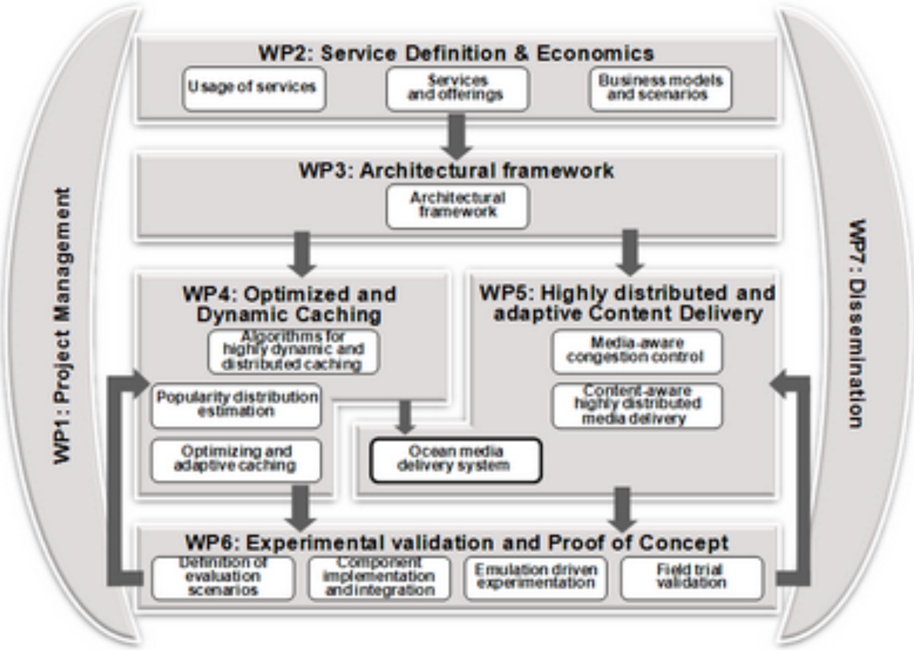

Figure 4 : OCEAN work package structure (http://www.ict-ocean.eu/workplan)

C.3 ENVISION

Future networked media applications will be multi-sourced, highly interactive distributed mixes of HD and 3D multi-sensory channels. While such high-quality user-centric applications offer tremendous advantages to their users and to society at large, they present a major challenge for ISPs. They require unprecedented quantities of network resources in unpredictable locations. Pre-provisioning enough resources everywhere is not economically relevant given the huge capacities required. If these services become a reality then a fundamentally different approach to dimensioning networks is necessary. ENVISION (http://www.envision-project.org/overview/index.html) proposes a three-pronged, cross-layer solution through cooperation between service providers, ISPs, users and their applications, where:

i) intelligent overlay applications are optimized for true end-to-end performance according to the actual capabilities of the underlying ISPs;

ii) network resources are dynamically used where they are most needed; and

iii) content, the way it is generated, accessed and distributed, is adapted on-the-fly to what the network is able to deliver and according to the terminal capabilities.

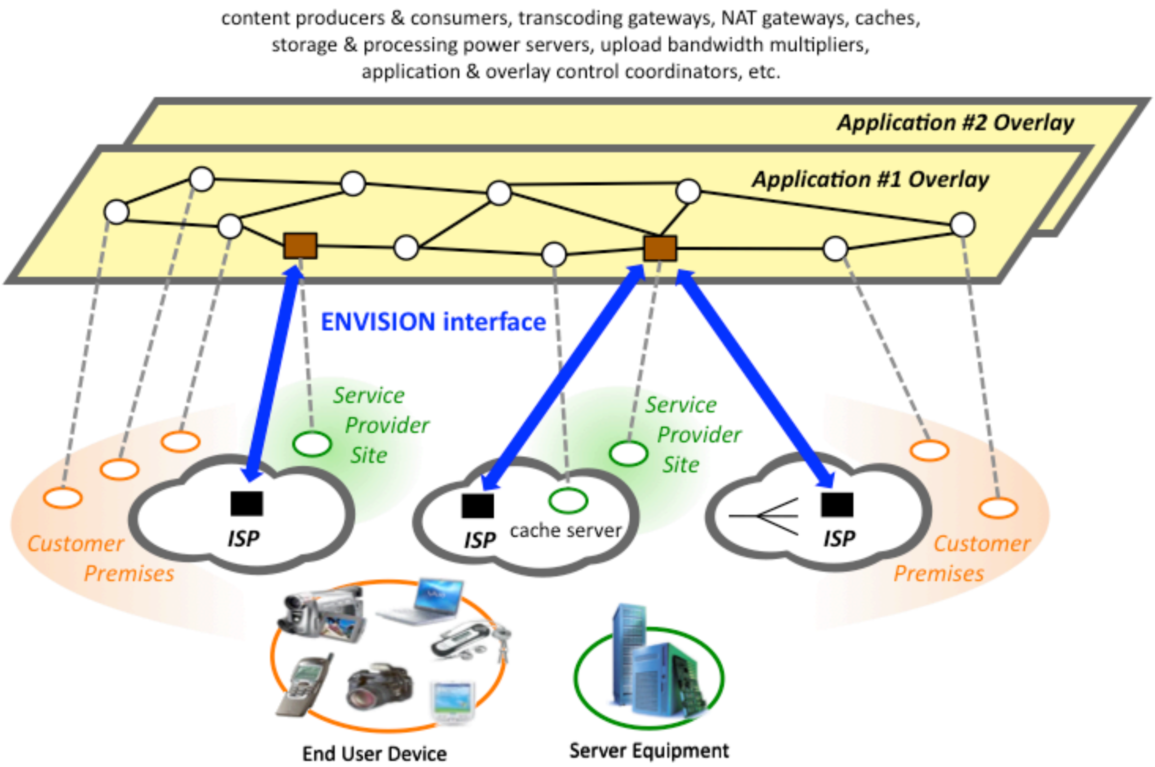

Figure 5 : ENVISION overview

(5th FP7 Networked Media Concertation Meeting, 3-4 February 2010, Brussels)

References

[1] http://www.nanodatacenters.eu/

[2] V. Valancius, N. Laoutaris, L. Massoulie, C. Diot, P. Rodriguez,"Greening the Internet with Nano Data Centers," in Proc. of ACMCoNEXT'09.

[3] N. Laoutaris, P. Rodriguez, L. Massoulie, "ECHOS: edge capacity hosting overlays of nano data centers," ACM SIGCOMM Computer Communication Review, vol. 38, no. 1, pp.51-54, Jan. 2008.

[4] http://www.juniper.net/

[5] http://www.future-internet.eu/fileadmin/documents/prague_documents/FIA-FCN_Internet_Architecture_20090507.pdf

[6] http://www.iti.gr/iti/files/document/FIA_Valencia_Book_2010.pdf

[7] http://www.xerox.com/innovation/news-stories/networking/enus.html

[8] http://www.parc.com/content/newsroom/CCN_backgrounder.pdf

[9] https://www-304.ibm.com/easyaccess/publicfrance/contenttemplate/!!/xmlid=174162

[10] http://cloudcomputing.sys-con.com/node/934716

[11] Jan Seedordf and Eric W. Burger. Application-Layer Traffic Optimization (ALTO) Problem Statement. Request for Comments 5693, October 2009.

[12] Haiyong Xie, Yang Richard Yang, Arvind Krishnamurthy, Yanbin Liu and Avi Silberschatz. P4P: Provider Portal for Applications. In ACM SIGCOMM, 2008.

[13] Oscar Ardaiz, Felix Freitag, and Leandro Navarro. Improving the service time of web clients using server redirection. *SIGMETRICS Perform. Eval. Rev.*, 29(2):39–44, 2001.

[14] Hari Balakrishnan, M. Frans Kaashoek, David Karger, Robert Morris, and Ion Stoica. Looking up data in p2p systems. *Commun. ACM*, 46(2):43–48, 2003.

[15] Y. Bartal. Probabilistic approximation of metric spaces and its algorithmic applications. In *FOCS '96: Proceedings of the 37th Annual Symposium on Foundations of Computer Science*, page 184, Washington, DC, USA, 1996. IEEE Computer Society.

[16] Novella Bartolini, Emiliano Casalicchio, and Salvatore Tucci. A walk through content delivery networks. In *MASCOTS Tutorials*, pages 1–25, 2003.

[17] Matthew Andrews Bruce, Bruce Shepherd, Aravind Srinivasan, Peter Winkler, and Francis Zane. Clustering and server selection using passive monitoring. In *In Proc. IEEE INFOCOM 2002*, pages 1717–1725. IEEE Computer Society Press, 2002.

[18] Yan Chen, Randy H. Katz, and John Kubiatowicz. Dynamic replica placement for scalable content delivery. In *IPTPS '01: Revised Papers from the First International Workshop on Peer-to-Peer Systems*, pages 306–318, London, UK, 2002. Springer-Verlag.

[19] Eric Cronin, Sugih Jamin, Cheng Jin, Anthony R. Kurc, Danny Raz, Yuval Shavitt, and Senior Member. Constrained mirror placement on the internet. In *IEEE Journal on Selected Areas in Communications*, pages 31–40, 2002.

[20] John Dilley, Bruce Maggs, Jay Parikh, Harald Prokop, and Bill Weihl. Globally distributed content delivery. *IEEE Internet Computing*, 6:50–58, 2002.

[21] Fred Douglis and M. Frans Kaashoek. Guest editors' introduction: Scalable internet services. *IEEE Internet Computing*, 5(4):36–37, 2001.

[22] Michael Freedman, Eric Freudenthal, David Mazires, and David Mazi Eres. Democratizing content publication with coral. In *In NSDI*, pages 239–252, 2004.

[23] Michael J. Freedman. Appeared in 7th usenix symposium on network design and implementation (nsdi 10) experiences with coralcdn: A five-year operational view.

[24] Cheng Huang, Angela Wang, Jin Li, and Keith W. Ross. Measuring and evaluating large-scale cdns paper withdrawn at microsoft's request. In *IMC '08: Proceedings of the 8th ACM SIGCOMM conference on Internet measurement*, pages 15–29, New York, NY, USA, 2008. ACM.

[25] Jamin Jin Jin. On the placement of internet instrumentation, 2000.

[26] Jussi Kangasharju, James Roberts, France Tlcom R, Keith W. Ross, Sophia Antipolis, and Sophia Antipolis. Object replication strategies in content distribution networks. In *Computer Communications*, pages 367–383, 2001.

[27] David Karger, Alex Sherman, Andy Berkheimer, Bill Bogstad, Rizwan Dhanidina, Ken Iwamoto, Brian Kim, Luke Matkins, and Yoav Yerushalmi. Web caching with consistent hashing. *Comput. Netw.*, 31(11-16):1203–1213, 1999.

[28] Magnus Karlsson and Mallik Mahalingam. Do we need replica placement algorithms in content delivery networks. In *In Proceedings of the International Workshop on Web Content Caching and Distribution (WCW*, pages 117–128, 2002.

[29] Balachander Krishnamurthy, Craig Wills, and Yin Zhang. On the use and performance of content distribution networks. In *IMW '01: Proceedings of the 1st ACM SIGCOMM Workshop on Internet Measurement*, pages 169–182, New York, NY, USA, 2001. ACM.

[30] P. Krishnan, Danny Raz, and Yuval Shavitt. The cache location problem. *IEEE/ACM Trans. Netw.*, 8(5):568–582, 2000.

[31] Rupa Krishnan, Harsha V. Madhyastha, Sridhar Srinivasan, Sushant Jain, Arvind Krishnamurthy, Thomas Anderson, and Jie Gao. Moving beyond end-to-end path information to optimize cdn performance. In *IMC '09: Proceedings of the 9th ACM SIGCOMM conference on Internet measurement conference*, pages 190–201, New York, NY, USA, 2009. ACM.

[32] Bo Li, Xin Deng, Mordecai J. Golin, and Kazem Sohraby. On the optimal placement of web proxies in the internet: The linear topology. In *HPN '98: Proceedings of the IFIP TC-6 Eigth International Conference on High Performance Networking*, pages 485–495, Deventer, The Netherlands, The Netherlands, 1998. Kluwer, B.V.

[33] Jian Ni and D.H.K. Tsang. Large-scale cooperative caching and application-level multicast in multimedia content delivery networks. *Communications Magazine, IEEE*, 43(5):98 – 105, may 2005.

[34] Vivek S. Pai, Mohit Aron, Gaurov Banga, Michael Svendsen, Peter Druschel, Willy Zwaenepoel, and Erich Nahum. Locality-aware request distribution in cluster-based network servers. *SIGPLAN Not.*, 33(11):205–216, 1998.

[35] George Pallis and Athena Vakali. Insight and perspectives for content delivery networks. *Commun. ACM*, 49(1):101–106, 2006.

[36] Gang Peng. Cdn: Content distribution network. Technical report, 2003.

[37] Guillaume Pierre and Maarten Steen. Globule: A collaborative content delivery network. *IEEE Communications Magazine*, 44:127–133, 2006.

[38] Lili Qiu, Venkata N. Padmanabhan, and Geoffrey M. Voelker. On the placement of web server replicas. In *In Proceedings of IEEE INFOCOM*, pages 1587–1596, 2001.

[39] Anees Shaikh, Renu Tewari, and Mukesh Agrawal. On the effectiveness of dns-based server selection. In *In Proceedings of IEEE Infocom*, 2001.

[40] Swaminathan Sivasubramanian, Michal Szymaniak, Guillaume Pierre, and Maarten van Steen. Replication for web hosting systems. *ACM Comput. Surv.*, 36(3):291–334, 2004.

[41] Konstantinos Stamos, George Pallis, and Athena Vakali. Integrating caching techniques on a content distribution network. In *In Proceedings of 10th East-European Conference on Advances in Databases and Information Systems (ADBIS 2006*. Springer-Verlag, 2006.

[42] Ion Stoica, Robert Morris, David Karger, M. Frans Kaashoek, and Hari Balakrishnan. Chord: A scalable peer-to-peer lookup service for internet applications. pages 149–160, 2001.

[43] Athena Vakali and George Pallis. Content delivery networks: Status and trends. *IEEE Internet Computing*, 7(6):68–74, 2003.

[44] Jia Wang. A survey of web caching schemes for the internet. *SIGCOMM Comput. Commun. Rev.*, 29(5):36–46, 1999.

[45] Limin Wang, Vivek Pai, and Larry Peterson. The effectiveness of request redirection on cdn robustness. In *OSDI '02: Proceedings of the 5th symposium on Operating systems design and implementation*, pages 345–360, New York, NY, USA, 2002. ACM.