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 3.5

▪

Revised state of the art on Content Distribution

Auteurs :

Pierrick PHILIPPE (Orange Labs)

Jean KYPREOS (Envivio)

Gérard MADEC, Annie GRAVEY (Télécom Bretagne)

March 2013

▪

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

Table of Contents

Summary

This document proposes a brief revision of the state of the art on content delivery and adaptation techniques, and justifies the technical choices made in the course of the VIPEER project.

Keywords: Dynamic streaming, DASH, adaptation engine, segments, manifests file, playlist, cache, and redirector.

1. Introduction

The VIPEER project is about delivering multimedia content on CDN and distributed CDN. To achieve that, caching and content adaptation are both crucial in order to serve the customer in a satisfactory fashion.

In this project, a hybrid technology has been proposed. The content is adaptively delivered depending on the client throughput, in order to satisfy as much as possible the client quality of experience (QoE), while complying with the network / CDN constraints in terms of policy and infrastructure.

As proposed and defined by the WP1 of this project, the dTracker plays a central role in the VIPEER architecture. It is the interface between a media consumer and the dCDN. The dTracket also drives content management in the sense that a media client is given a content based on the dTracker decision.

The dTracker is in charge of the content addressing and maps the content requested by the consumer to a location in the delivery network implemented by the operator (i.e. the dCDN). This is because it selects both the content and its source dynamically according to numerous factors such as the network topology, the local content availability and the end-user device or network capability.

The selection is dynamic in the sense that the client will be served depending on the current state of the resources available in the network.

Moreover, once the user retrieves the content location, individual pieces of the media are served depending on the client throughput and other metrics, typically defined by the WP2 of this project.

This second source of adaptation is permitted through the use of the Dynamic Adaptive Streaming (DASH) technology, selected in this project as the media delivery technology. This technology is reviewed in the present report and its benefits to the proposed VIPEER architecture are highlighted.

2. Adaptive streaming

The media content is first segmented in into small time periods, the so-called “segments”. Each time period can be coded at different bit rates or resolutions.

As each segment is independent of the others, the interest of adaptive streaming lies in the possibility of the selection successive segments according to a specific strategy with a very high flexibility. The player, according to its estimation of its local media capabilities, performs this selection.

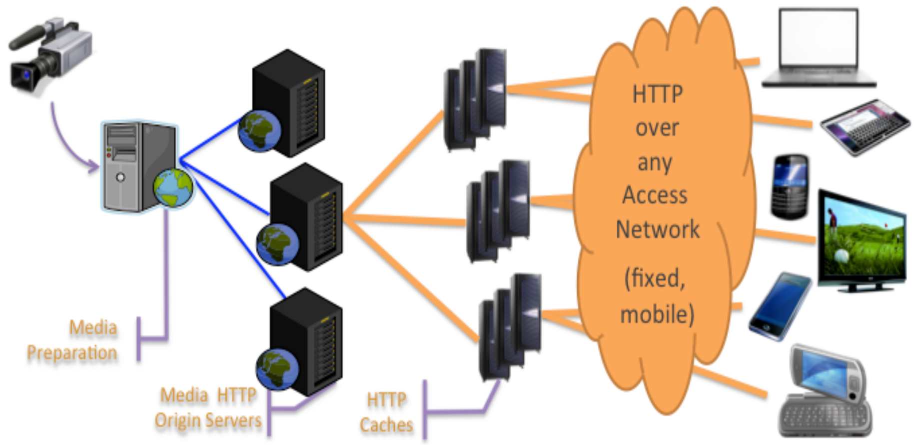

Figure 1 HTTP Streaming Scheme

Figure 1 shows a Dynamic Adaptive Streaming over HTTP application and relevant function covered by the technology.

2.1 Adaptive mechanisms

As previously mentioned, flexibility of the segments selection may be controlled by numerous factors such as the availability of the segments in a given server, the client bandwidth, the terminal features, the QoE at the client side, and so on. However, both the client's side and the system's side may control segment selection.

2.1.1 Adaptation at the client (terminal) side

When as session of adaptive streaming launches the client initiates a chunk request with a specific resolution / bit rate ("profile"). As the client receives segments, it decides to change or not the segment profile according to a local bandwidth measure made by the client itself. In this mechanism the client therefore adapts itself to the network's state: it can request a segment with lower resolution if it assesses the network as "congested", but it can also request for a higher resolution if it assesses the network as "non congested". This is done independently from the server, and from the network itself. However, it implicitly assumes that the path between the server and the client is (semi) static, i.e. that it does not change from segment to segment unless in case of a network failure affecting the path.

2.1.2 Adaptation at the system's side

In the VIPEER architecture, content is distributed over the network and may be delivered by the system to the client from different servers. This implies that network's performance assessment by the client may be inconsistent. Moreover, the (selfish) adaptation by each client of its own delivery rate may negatively impact a global system's optimization policy that tries to balance the load between various servers and various portions of the network.

VIPEER therefore proposes that rate adaptation be performed or at least controlled by the system. Segments are stored at different locations in the network through CDN or distributed CDN, and the dTracker provides the appropriate segment address to the client according to numerous parameters from the network (client distance, its bandwidth, etc …).

Such a global system's control can work in parallel with the original DASH adaptation policy since it consists in either accepting or overriding clients requests.

2.1.3 Communication using HTTP

In order to transfer simple messages within the network, adaptive streaming mechanisms use HTTP requests.

The main advantage of HTTP is its ability to go through most firewalls and NAT implementations; moreover, additional processing of the segments by HTTP is negligible in terms of complexity and is very flexible.

2.1.4 Addressing different types of terminals

By construction, multiple representations (bit rate & resolution) of the same media content are available in the network. It is therefore very easy to feed a terminal with its appropriate representation. Multiple types of terminals may be then managed with a high flexibility.

2.2 Dynamic Adaptive Streaming over HTTP

Dynamic Adaptive Streaming over HTTP (DASH) is efficient and easy to use on existing CDNs, proxies, caches, NATs and firewalls. It can easily be deployed on top of HTTP-CDNs (Web Infrastructures, caching).

2.3 DASH: the standard for HTTP adaptive streaming

Akamai's principal architect for media engineering, commented in [1]: *"We've spent the past five years delivering a variety of adaptive video formats—SmoothHD, HDNI*, HLS and HDS—all of which are 80 percent the same but 100 percent incompatible."*

DASH is the ISO standard for Dynamic Adaptive Streaming over HTTP: its official name is MPEG-DASH ISO/IEC 23009-1. It specifies formats enabling delivery of media content from standard HTTP servers to HTTP clients and enabling caching of content by standard HTTP caches.

The rationale for the establishment of this new standard is to encompass all the existing practices in terms of HTTP streaming and to propose a common interface for distributing media content over HTTP. As a consequence MPEG DASH has proposed different alternatives ("profiles" in MPEG terminology) to mimic the existing adaptive streaming techniques. Particularly, a DASH compliant player can support both Apple HLS format and Microsoft Smooth Streaming formats, with subtle changes. These actors were also part of the standardization process through the participation of some of their experts.

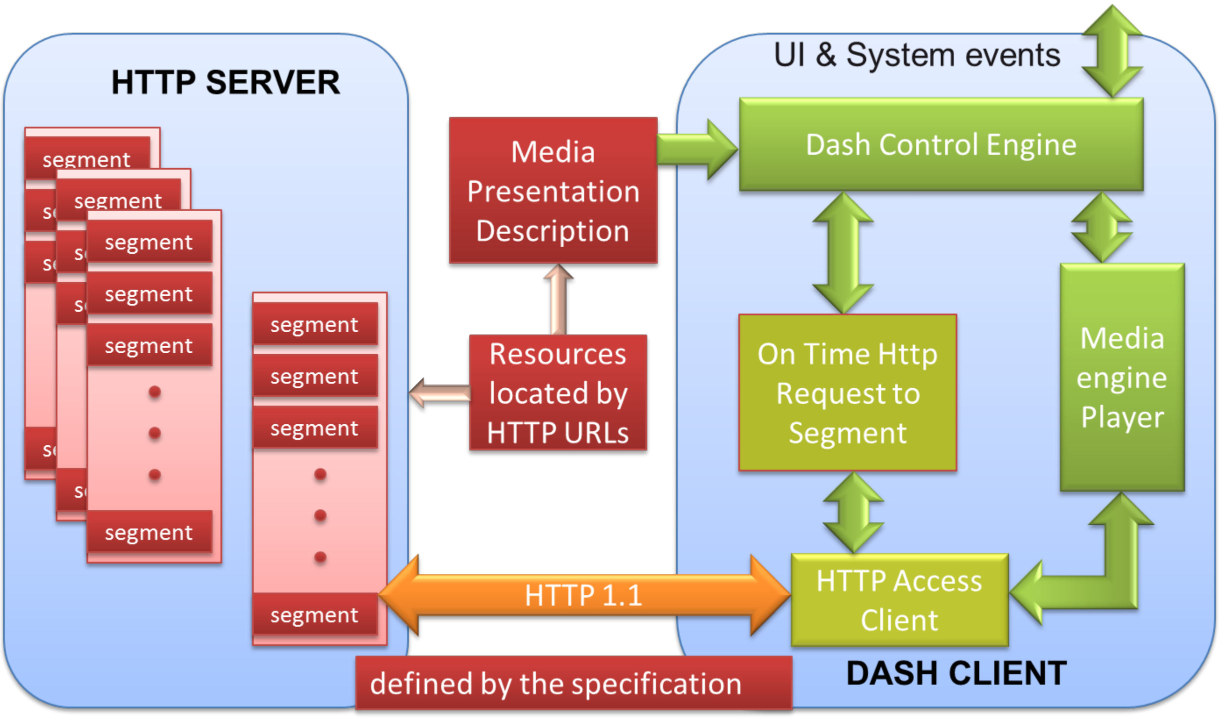

Figure 2 Scope of the MPEG-DASH Standard

The scope of the standard is presented in **Figure** 2. Attention should be paid to the fact that DASH specifies the description of the available media/ segments (the so-called manifest file, that consists in a .xml file named "**MPD**: **Media Presentation description**") and the actual format for those media called "**chunk**" or "**segment**" located on a HTTP server (which can adhere to the currently dominant formats that are MPEG-2 Transport Stream and MPEG-4 File Format).

The client is out of the DASH standard: a basic client should be able to read the MPD, get the segments and play the audiovisual elementary stream formed with the concatenation of the received segments. A basic scenario is represented in Figure 2:

1. The DASH control engine includes User Interface (UI) and System Events Receiver. It reads the MPD associated to the available A/V services, and the events that are pushed to the client (e.g. an MPD update). The DASH control engine can adapt the bitrate by analyzing the time to get segments (regulation has to be smoothed!).
2. On Time HTTP Requests to segment are send the HTTP GET function
3. HTTP access get the segments and forwards them to the media engine player
4. The Media Engine Player compliant with the type of segment MP4 or TS buffered and synchronized the different elementary streams, decodes the elementary streams and plays them via adapted players.

This international standard has a large support from industry**:** a group was constituted with the objective of promoting DASH through interoperability (**http://dashpromotersgroup.com/).** MPEG DASH is also a superset for system different specifications which target a particular application domain:

- 3GPP Release-9 AHS in TS 26.234 and 3GPP Release-10 DASH TS26.247 for mobile media services
- Open IPTV Forum HTTP Adaptive Streaming (HAS), for IPTV
- HbbTV, for interactive TV services

Hence DASH is an open standard, since it is covered by an open documentation and allows for numerous implementations. A consequence of that, a DASH client has recently be made available as a plug-in for the popular, multiplatform VLC multimedia player and as a Jsavascript API on HTML5 [5].

This openness has made DASH an obvious choice for the VIPEER project:

- adaptive streaming is a technology allowing flexibility in the CDN through classical url mapping redirections;
- HTTP enables redirection to alternate servers as a content becomes available on a server;
- Quality of experience can be accommodated through the local adaptation, at the client side, but also at system's side by accepting or overriding clients' requests, and selecting an optimal server.
- Implementations are easily available and can be even found as open source projects.

3 How DASH fits VIPEER's requirements

In the VIPEER project, the dTracker network element generates manifest files that describe the position of the segments on the CDN nodes: the dTracker prepares the manifest files that contains the URLs addressing the parts constituting the media with different representations. The generation process can take account of the user's characteristics such as its location and terminal type. Consequently the segments are made available on the CDN nodes, possibly located close to the end user leaf, and the end-user can access to the requested content.

Thanks to DASH, and due to the manifest element, the URL addressing the media segments can easily be changed, hence caching can easily be managed by a simple change in target addresses. The manifest element allows for efficient dynamic caching and is therefore the main adaptation tool used by VIPEER .

Since DASH relies on HTTP, standard equipment compatible with this protocol can be re-used in a straightforward fashion. This includes servers (e.g. Apache servers can be re-used) and CDNs' infrastructure that do not need a profound refactoring to fit this technology.

The availability of DASH players as open source distributions makes also possible the integration of specific algorithms such like the QoE estimation at the end-user side. QoE metrics (derived by the WP2 of this project)

The openness of this technologies and the massive support by the industry enables DASH to be a suitable solution as the media delivery technology for this project.

4 Perspectives and conclusion

The present section positions the technical choices made by VIPEER relatively to potential alternatives.

4.1 Scalable Video Coding (SVC) versus DASH

When VIPEER was launched, the consortium had not selected a video distribution method. As stated previously, DASH was selected by VIPEER, although it is less mature than SVC, and is not yet standardized, contrarily to SVC.

Indeed, SVC is a video codec that has been standardized by MPEG and ITU in July 2007. SVC was designed to generate a single high quality stream video that contains multiple sub streams that could be decoded separately. A SVC stream is therefore made of multiple layers that are encapsulated together. It is a superset of the AVC/H.264 specification.

SVC enables multiple scalability schemes:

- Temporal scalability: the same media content is available at different frame rates
- Spatial scalability: the same media content is available at different spatial resolutions
- SNR scalability: a given spatial resolution of the media content is available with different qualities (equivalent to different bit rates)

SVC's major interest lies in the fact that its base layer is common to all the desired quality that might be requested by the user. It is therefore possible to cache a significant amount of additional videos, albeit the video cached is the lowest representation in terms of video quality. Caching of SVC content in a CDN has been studied in the course of the IST Ocean project [6].

Compared to HTTP adaptive streaming described previously, SVC appears to be a serious competitor but nevertheless suffers from several drawbacks:

- Processing in the network: extraction of the desired layer needs parsing of the original stream and requires processing. These in-network processing features are currently not available in legacy servers or CDNs.
- Infrastructure:
  - the entire bit stream has to be transmitted in the network even if only a single representation (a layer) is to be used
  - as a desired representation of a media has to be extracted, management of the overall representations is much less flexible as opposed to HTTP requests
- Decoding: Terminals needs to embed a specific SVC decoder to play the final representation: albeit SVC is standardized since 2007, the implementations are rare, as even the "Swiss Knife" of the media player, VLC, does not support it.
- Codec technology dependency: management of media content is intrinsically tied to SVC mechanisms, and future desired evolutions would be impossible without major changes.

4.2 Recent video compression standards

Many video compression standards have been standardized (or are under normalization process) by MPEG:

- MVC (multiview video coding) for 3D video coding, and currently HEVC (High Efficiency Video Coding) is the next standard after AVC/H264 to be finally standardized in January 2013.
- A new standard SHVC (scalable high video coding) is under development and based on extension of HEVC used as a base layer with the scalability feature. It is expected to be finalized in mid 2014, the scalability mechanism is similar to the ones developed for SVC.
- Proprietary alternative video coding techniques have also been developed recently such as the VP8/WebM technology proposed by Google.

These alternatives are quite recent, and none is as flexible as DASH.

4.3 Conclusion

As mentioned previously, HTTP streaming is a very flexible technology, as long the content is segmented into segments.

Its major strength is that it is independent from the video compression standard type. Emerging and future video standards can thus take advantage of a deployed DASH architecture, as this architecture indeed supports any codec type using only a simple description such as the codec type that is transmitted in the manifest file describing the media content streaming.

Hence, newly developed codecs can be integrated in a transparent fashion, as far as the delivery process is concerned. In that sense the architectural approach for the adaptive content delivery developed by VIPEER is generic.

5 References

[1] *"Apple HTTP Live Streaming description*", available on IETF website: http://tools.ietf.org/html/draft-pantos-http-live-streaming-06

[2] *"Microsoft Smooth Streaming presentation"*, available at http://www.iis.net/download/SmoothStreaming

[3] *"MPEG Overview of Dynamic Adaptive Streaming over HTTP*", document ISO/IEC JTC1/SC29/WG11/N11964, 2011, available at http://mpeg.chiariglione.org/technologies/mpeg-b/mpb-dash/index.htm

[4] "*Content adaptation use cases and requirements – Architectural specifications of the content adaptation engine*", ViPEER deliverable D3.1, July 2010

[5] http://www-itec.uni-klu.ac.at/dash/

[6] Ocean project web page: http://www.ict-ocean.eu/

[7] "MPEG DASH: the file format of the future. http://www.streamingmedia.com/Articles/Editorial/Featured-Articles/MPEG-DASH-The-File-Format-of-the-Future-78835.aspx