Control Plane Design and Implementation for a Media Streaming System working in Over the Top style

Eugen Borcoci, Serban Georgica Obreja, Cristian Cernat, Radu Iorga
University POLITEHNICA of Bucharest
Bucharest, Romania
emails: eugen.borcoci@elcom.pub.ro, serban@radio.pub.ro, cristian.cernat@elcom.pub.ro, radu.iorga@elcom.pub.ro

Jordi Mongay Batalla
National Institute of Telecommunications
Warsaw, Poland
email:jordim@interfree.it
Daniel Negru
LaBRI Lab, University of Bordeaux
Bordeaux, France
email:daniel.negru@labri.fr

***Abstract* — Content/media streaming services are frequently used in the current Internet and this trend is estimated to continue in the next years. Complex architectures, like Content Delivery Networks, have been developed and are currently exploited on a large scale, or novel architectures like Content Oriented Networks are proposed. In parallel, cheaper and light architecture, working in over-the-top (OTT) style above the current IP networks multi-domain infrastructures have been recently considered, as an attractive solution for media distribution. A content streaming OTT-like system is considered here, working on top of a multi-domain IP network. The system integrates functionalities such as content server initial selection based on multi-criteria algorithm and then performs dynamic media adaptation and/or server switching - during the media session. This paper continues and extends a preliminary work; it presents more deeply the main design concepts, then develops the implementation solutions and presents the functional conformance testing for the Control Plane of the above system. The functional testing is described, performed on a real-life multi-domain testbed as a part of the overall validation campaign.**

Keywords — Content delivery, Dynamic Adaptive Streaming over HTTP, Monitoring, Server and Path selection.

I. INTRODUCTION

In the current Internet, content/media streaming services are frequently used and this trend is estimated to continue in the next years. Several approaches are encountered in practice: light and cheap over-the-top (OTT) solutions like that described in [1] or complex ones, the latter being usually based on powerful and costly management system. Complex architectures like Content Delivery Networks [2], have been developed and are currently exploited on a large scale, or novel architectures like Content Oriented Networks (CON) [3][4], are proposed.

Some cheaper and light over-the-top (OTT) solutions are recently proposed and developed for media/content delivery, where the services are delivered over the current Internet by an entity called Service Provider (SP). The SP is not directly responsible for the quality of the flow transmission to the end-user. The users have access to the services via the "public Internet". An OTT SP could exist as a separate entity from traditional Internet Service Provider (ISP), or might be embedded in ISP. Also, combined solutions exist, with OTT Service Providers using the Content Delivery Network (CDN) Providers' infrastructure to improve the quality of delivery.

A *light (OTT-like) novel architecture* for content streaming systems over the current Internet is proposed by the European *DISEDAN* Chist-Era project [1][5], (*service and user-based* ***DI****stributed* ***SE****lection of content streaming source and* ***D****ual* ***A****daptatio****N*****,** 2014-2015).

This paper is an extension work of a previous one [1] presented at AICT 2015 Conference, dedicated to develop the design of the Control Pane system of the DISEDAN project. The present work goes further, towards full implementation and system functional testing. The additional material is contained mainly in the Sections V and VI.

The business actors involved are: *Service Provider (SP)* - an entity/actor which delivers the content services to the users and possibly owns and manages the transportation network); *End Users (EU)* consumes the content; a *Content Provider (CP)* could exist, owning some *Content Servers (CS)*.

The DISEDAN project proposed an evolutionary solution to enhance the content delivery via Internet. It has been focused on research in the area of multi-criteria content source (server) selection and handover, considering user and server contexts. Two main concepts have been studied and implemented: *two-step server initial selection mechanism* and *dual adaptation in-session mechanism.*

The "Two-step server initial selection mechanism", is run before the media session start. It allows for cooperation between Service Provider (an entity offering the content distribution service, owning or not a network infrastructure) and End User, making use of innovative server selection algorithms that consider context- and content-awareness. A solution is proposed for the (multi-criteria-hard) problem of best content source (server) selection, considering user context, servers' availability and requested content.

The "Dual adaptation mechanism" is activated during the media session. It is based on combining in a single solution the advantages of Media (flow rate) adaptation methods and/or content server handover.

At client side the proposed streaming system is able to function as a standalone client application, without any modifications applied to the Service Provider (SP). However in DISEDAN the SP is supposed to be able to provide to the client a list of available and appropriate servers. This additional information is helpful for the initial client decision, thus optimizing the initial server selection. Consequently, a set of optional Provider side modifications have been identified and implemented (w.r.t. useful information and metrics provided by SP to the client) (see [14]) that can further optimize server selection. The design of the system took the backwards-compatibility into consideration, ensuring that each of the client-side and Service Provider - side modifications works well with the other side, while the dialogue partner (SP/client) is using existing content distribution solutions.

In [14] clear rules for deciding which adaptation action to perform have been defined, based on the evaluated current delivery conditions. Possibilities of inferring the optimum adaptation decision by estimating network state from various client measurements have been studied.

The research results have been verified by performing tests of the prototype implementation of the solution. These are partially reported in this paper. The system implementation can be released as open source client library which will allow other institutions to reproduce the results and continue future investigations on these issues. Simulation models have been elaborated [5], during the project and the associated experiments have been run to complement the implementation - based tests, in order to assess the scalability of the proposed solutions in the context of large networks and users communities.

Note that DISEDAN system management does not deal with contractual CP-SP relationships; therefore one may assume that CSs are also owned by the SP. Our solution consisting in: (1) *two-step server selection mechanism* (at SP and at EU) using and (2) *dual adaptation mechanism* consisting of dynamic *media adaptation* and/or *content source switching* (by *streaming server switching*) could be rapidly deployed in the market since it does not require complex architecture like Content Delivery Networks [2], or Content Oriented Networking [3][4].

The *Dynamic Adaptive Streaming over Hypertext Transfer Protocol- HTTP (DASH)* technology has been selected for in-session media adaptation. The DASH was recently adopted as multimedia streaming standard, to deliver high quality multimedia content over the Internet, by using conventional HTTP Web servers [6] - [10]. It uses the HTTP protocol, minimizes server processing power and is video codec agnostic. Its basic concept is to enable automatic switching of quality levels according to network conditions, user requirements, and expectations.

A DASH client continuously selects the highest possible video representation quality that ensures smooth play-out, in the current downloading conditions. This selection is performed on-the-fly, during video play-out, from a pre-defined discrete set of available video rates and with a pre-defined granularity (according to video segmentation).

The DASH offers important advantages (over traditional push-based streaming), like: significant market adoption of HTTP and TCP/IP protocols to support the majority of the today Internet services; HTTP-based delivery avoids NAT and firewall- related issues; the HTTP-based (non-adaptive) deployment of progressive download existing today, can be conveniently be upgraded to support DASH; the ability to use standard/existing HTTP servers and caches instead of specialized streaming servers allows reuse of the existing infrastructure.

The work of this paper is dedicated to continue the design, following the decisions presented in [1] (for details of the major design decisions one can see [14]), for a light Control Plane (CPl) of the OTT streaming system and develop the implementation. Functional testing framework and results are outlined in a dedicated section.

Note that implementation of the DASH adaptation details are not in the scope of this paper.

A special attention has been paid for the Monitoring subsystem (MON), whose components are developed at SP, CS and optionally at EU Terminal (EUT). The MON is an essential functional component, contributing to the evaluation of the QoS and QoE and charged to trigger appropriate actions if it is the case. The information delivered by MON is used to support both the initial server selection and then in-session actions.

Note that our main purpose in this work has not been to essentially innovate in monitoring tools (a lot of implementations are available), but to integrate different components, aiming to develop a complete monitoring subsystem appropriate for DISEDAN light architecture.

The paper structure is described here. Section II is a short overview of related work. Section III outlines the overall architecture and problem description. Section IV is focused on defining CPl design decisions and implementation-related implications. Section V presents the main implementation characteristics. Section VI defines a testbed to support the system and presents samples of functional tests results. Section VII contains conclusions and future work outline.

II. RELATED WORK

The real-time adaptation in content streaming is a powerful and dynamic technique, adopted to solve the fluctuations in QoE/QoS. One can classify adaptation as acting on Media (flow) and/or on Content Server. The *Media adaptation* is a significant technique and constitutes a main research and innovation area in media streaming applications [7][8][15]. *CS adaptation* means a new content server selection (during the media session) and switching (handover), depending on the consumer device capabilities, consumer location, content servers state and/or network state [11][12].

A so-called "dual adaptation" is a process that integrates the above adaptation methods. The DISEDAN novel architecture [5] combines the initial server selection (result of cooperation between SP and EU) with session-time dual adaptation, in a single solution.

The initial server selection is based on optimization algorithms like *Multi-Criteria Decision Algorithms (MCDA)* [11][12], or *Evolutionary Multi-objective Optimization algorithm* (EMO) [13], modified to be applied to DISEDAN context. In these works several scenarios are proposed, analyzed and evaluated. In particular, the availability of different static and/or dynamic input parameters for optimization algorithms is considered. The result of this variability is that several CPl designs are possible, different in terms of performance and complexity. The dynamic capabilities for the initial CS selection and then for adaptation decisions depends essentially on the power of the DISEDAN monitoring system. Different design variants, offering actually a family, have been in detail analyzed in [14].

The challenge in DISEDAN is to combine the DASH-related functionalities with additional monitoring in order to finally realize the dual adaptation.

The standard ISO/IEC 23009-1, "Information technology -- Dynamic adaptive streaming over HTTP (DASH)" [8], defines the DASH-Metrics client reference model, composed of *DASH access client (DAC)*, followed by the *DASH-enabled application (DAE)* and *Media Output* (MO) module. The DAC issues HTTP requests (for DASH data structures), and receives HTTP request responses. Consequently three observation points (interfaces – I/F) can be identified (see Figure 1) :

- O1 *at network-DAC I/F*: a set of TCP connections, each defined by its destination IP address, initiation, connect and close times; a sequence of transmitted HTTP requests, each defined by its transmission time, contents, and the TCP connection on which it is sent; and for each HTTP response, the reception time and contents of the response header and the reception time of each byte of the response body.

- *O2 at DAC-DAE I/F*: consists of encoded media samples. Each encoded media sample is defined as: media type; decoding time; presentation time; the @*id* of the Representation from which the sample is taken; the delivery time.

- *O3 at DAE-MO I/F*: consists of decoded media samples. Each decoded media sample is defined as: the media type; the presentation timestamp of the sample (media time); the actual presentation time of the sample (real time); the @id of the Representation from which the sample is taken (the highest dependency level if the sample was constructed from multiple Representations).

A summary of the metrics semantic defined in ISO/IEC 23009-1 [8], is: *Transmission Control Protocol (TCP) connections, HTTP request/response transactions, Representation switch events, Buffer level, Play list.* A similar list of QoE metrics standardized by 3GPP defined in 3GPP in 26.247, applicable for DASH, [10][15], contains: *HTTP request/ response transactions; Representation switch events; Average throughput; Initial play-out delay; Buffer level; Play list; MD information*.

For completeness of the text, the full architecture of the DISEDAN system is presented in the following sections; however the focus here is on server selection functionality.

III. DISEDAN System Architecture

This section shortly presents the system architecture, starting with the general framework and assumptions and then outlining the general functional architecture.

A. *General framework and assumptions*

The definition and details of the system architecture are already given in [5][11][12][14]. In this section, a summary only will be presented, to facilitate the understanding of the Control Plane design decisions.

The main business entities/ actors are those already mentioned in Section I: Service Provider (SP), End User (EU) and Conent Server (CS). In practice there can exist another distinct entity, Content Provider (CP), which is the normal owner the Content Servers. However, considering the OTT- style and simplified design of our system, the SP and Content Provider (CP) entities are not seen as distinct. Also, a full CS management is out of scope of this system. The connectivity between CSs and EU Terminals (EUT) is assured by traditional *Internet Services Providers (ISP) / Network Providers (NP*) - operators. The ISP/NPs whose networks are crossed by the media flows might or not to have business relationships but these are not visible in the management DISEDAN architecture.

However, the DISEDAN solution can be also emedded in more complex business models, e.g., involving Cloud Providers, CDN providers, etc. The relationships between SP and such entities could exist, but their realization is out of scope of this study. While Service Level Agreements (SLAs) might be agreed between SP and ISPs/NPs, related to connectivity services offered by the latter to SP, such SLAs are not directly visible at DISEDAN system level.

The system can work over the traditional TCP/IP mono and/or multi-domain network environment. The EUTs are not supposed to have explicit knowledge about the managed/non-managed characteristics of the connectivity services. No reservation for connectivity resources, neither connectivity services differentiation at network level are explicitly supposed (they might exist or not). This architectural choice proves the system flexibility: it can work in OTT style, or over a managed connectivity service offered by the network. Therefore, the SP does not commit to offer strong QoS guarantees for the streaming services provided to EUs. Consequently, DISEDAN does not suppose, but does not exclude, establishment of a SLA relationships between EUs and SPs management entities. However, it is assumed that a Media Description Server exists, managed by SP, to which EUT will directly interact.

The media streaming actions are independent on the transport networking technology. The EUT part (client side) works as a standalone client application, without any mandatory modifications applied to the SP; however, SP should provide some basic information to EUT, to help it in making initial server selection (and optionally to help in-session CS switching). The in-session decision about dual adaptation (media flow adaptation and/or CS switching) will be taken mainly locally at EUT, thus assuring complete User independency and avoiding complex signaling in the Control Plane, between EUT and SP during the session.

An assumption is made in our system: several CSs might exist, known by SP (geographical location, server availability level, access conditions for users). Among them, the SP and/or EUs can operate server selection (initial phase) and/or switching (in-session action).

The DISEDAN system research study has been limited to innovative parts. The proposed architecture does not treat how to solve failures inside the networks, except attempts to perform media flow DASH adaptation or CS switching when the service quality observed at EU is too low. The proposed system does not explicitly treat or innovate in the domain of content protection, Digital Rights Management (DRM), etc., but might use currently available solutions. Billing, financial aspects and other business related management of the DISEDAN high level services are out of the project scope.

The work [14], elaborated also in DISEDAN framework, has defined all requirements coming for EU, SP and out of them derived the general and specific System requirements, together with some assumptions and constrains imposed to such a system. The resulting high level architecture has been determined by such requirements. This work is based on the assumption of fulfillment of those requirements.

B. General Architecture

Figure 1 shows a simplified high level view of the general architecture.

The Control Plane at SP includes the following functional modules:

- *MPD File generator* – dynamically generates Media Presentation Description (MPD) XML file, containing media segments information (video resolution, bit rates, etc.), ranked list of recommended CSs and, optionally - current CSs state information and network state (if applicable).
- *Selection algorithm* –runs Step 1 of server selection process. It exploits *MCDA* [11][12], modified to be applied to DISEDAN context, or *EMO* [13], etc., to rank the set of recommended CSs and media representations, while aiming to optimize servers' load as well as to maximize system utilization.
- *Monitoring module* – collects monitoring information from CSs and performs the processing required to estimate the current state of each CS. Note that if some EU information should go to SP, then this information is transited (and aggregated) from EUT via CS towards SP.

The End User Terminal (EUT) entity includes the modules:

- *Data Plane: DASH (access and application)* – parses the MD file received from SP and handles the download of media segments from CS; *Media Player* – playbacks the downloaded media segments.
- *Control Plane: Content Source Selection and Adaptation engine* –implements the dual adaptation mechanism; *Selection algorithm* – performs the Step 2 of server selection process. It can also exploit MCDA, EMO, or other algorithms to select the best CS from the set of candidates recommended by SP; *Monitoring module* – monitors the changes in the local network and server conditions.

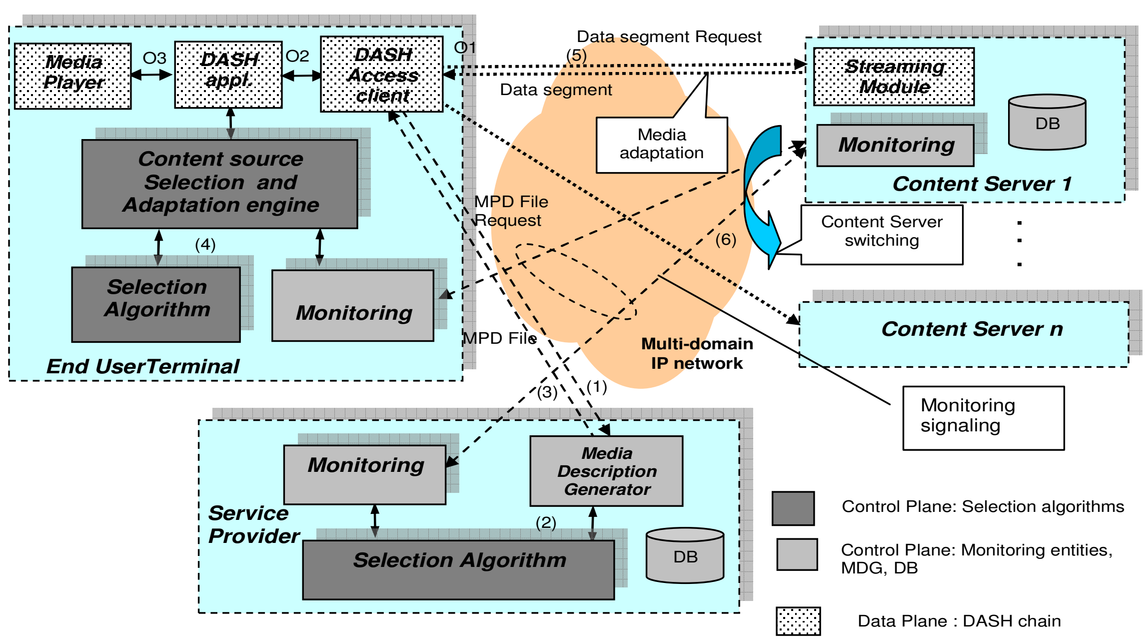


Figure 1. DISEDAN general architecture; DASH - Dynamic Adaptive Streaming over HTTP; MD – Media Description; DB – Data Base ; O1, O2, O3 – DASH Observation Points [ISO/IEC 23009-1]

The Content Server (CS) entity includes the modules:

- *Data Plane: Streaming module* – sends media segments requested by End Users.
- *Control Plane*: *Monitoring module* – monitors some CS performance metrics (CPU utilization, network interfaces utilization, etc.). In a complex implementation of the CS, the monitoring can evolve from a simple metering probe to an advanced monitoring module, capable to supervise not only the active sessions but also some connectivity characteristics from this CS to different groups of users.

Figure 1 also shows the following operational steps (simplified description):

(1) EUT issues to SP a media file request.

(2) SP analyzes the status of the CSs and runs the MCDA/EMO selection algorithm (optionally the SP could make first, a current probing of the CSs). For each user request the SP could consider also the user profile, the policies of the SP for this user's class and other information at the SP side (e.g., state of the servers and possibly network-related information).

(3) SP returns to EUT a ordered list of candidates CS (this is the SP proposal embedded in a MD- xml file).

(4) The EUT performs the final CS selection, by running its own selection algorithm. This can have as inut local iformations available at EUT.

(5) The EUT starts asking video segments from the selected CS.

During media session the EUT makes quality and context measurements. Continuous media flow adaptation is applied using DASH technology if necessary, or (6) CS switching is decided.

From the EU point of view, the steps 1-2-3 composed the so-called Phase1 and steps 4-5-6 the Phase 2.

During the receipt of consecutive chunks, the user's application can automatically change the rate of the content stream (internal DASH actions - which are out of scope in this paper) and/or also can switch to another CS. When EUT receives requested segments, it performs measurements to monitor parameters of download process. Note that the system is flexible in terms of monitoring procedures to follow. For instance, if EUT detects deterioration of the downloading rate, it can use SP information about alternate CSs and/or it can start probing other CSs. When the probing process is finished, EUT starts the dual adaptation process to decide: either DASH-based media adaptation, or server adaptaion (i.e. server switching). If the first is selected, then EU downloads (via DASH) next segments with reduced rate, otherwise it switches to another CS (CS probing by te EUT might be involved in such an action).

IV. MONITORING SUBSYSTEM

The architecture of the DISEDAN CPl is flexible. Several variants/versions of designs can be considered, i.e., a basic one or more complex, essentially depending on the roles of the business entities and their capabilities, interactions and also on SP and EU policies. The selection algorithms MCDA/EMO might work with different sets of static and/or dynamic input parameters. An important component of the CPl is the Monitoring subsystem *(MON@DISEDAN*).

A. *Monitoring Architecture*

Three MON modules have been identified in Figure 1: MON@SP, MON@CS, MON@EUT. However, not all these entities must participate to all operational phases. The variety of solutions will determine the system overall performance, but with additional cost for the more complex solutions. The monitored data are used to accomplish the following macro objectives:

- guide the initial server selection at SP and (optionally) at EU,
- guide the media adaptation and/or CS switching.

From the EUT point of view, two phases are distinguished: *Phase1* in which the EUT is not connected to any CS, but it just tries to do this, by contacting the SP; *Phase2* in which the EUT is currently served by a CS (media session time). The monitored data at EU level are different in Phase 1 w.r.t. Phase 2.

Note also that during media session, the DASH subsystem performs its own evaluation of the QoE and, based on this, decides upon requested rate of the next video segment. The implementation of this type of monitoring is out of MON scope. However, the data collected from such on-line monitoring can be combined with other values delivered by MON@EU and delivered to other entities in the hierarchy (CS, SP). Actually, we adopted the approach described in [15] where it is recalled that in the 3GPP DASH specification TS 26.247 [7-8], QoE measurement and reporting capability is defined as an optional feature for client devices. If the EUT supports the QoE reporting feature, the DASH standard also mandates the reporting of all of the requested metrics at any given time; that is, the client should be capable of measuring and reporting all of the QoE metrics specified in the standard.

The standard TS 26.247 also specifies two options for the activation or triggering of QoE reporting:

a. via the Quality Metrics element in the MPD;

b. via the OMA Device Management (DM) QoE Management Object.

In both cases a and b, the trigger message from the CS would include reporting configuration information such as the set of QoE metrics to be reported, the URIs for the server(s) to which the QoE reports should be sent, the format of the QoE reports, information on QoE reporting frequency and measurement interval, percentage of sessions for which QoE metrics will be reported, and access point names to be used for establishing the packet data protocol (PDP) context to be used for sending the QoE reports.

The selection algorithms MCDA/EMO might work with different sets of static and/or dynamic input parameters.

To achieve scalability of the monitoring system an important design decision is *to avoid direct signaling between EUT and SP*, except the initial request issued by EUT towards SP, in order to get the MPD xml file. Apart this phase, any monitored information obtained in EUT premises will be sent to the current CS serving that EUT.

Three control bi-directional channels are defined (see Figure 1) :

EUT-SP to generate the EU request to SP and to get the MPD file from SP. This is performed in Phase 1 of the DISEDAN functional cycle, i.e., at CS selection time.

EUT–CS triggered by the serving CS, to report, the monitored data about current EU status and media session data. This signaling is performed during Phase 2 time life for this EUT (i.e., media session).

CS-SP- to report: CS status data (capacity occupied, number of connections currently served, etc.); status data received from EUT (such data can be related to some individual users or aggregated at the CS level. The communication on this channel is triggered by the SP.

B. Typical Scenarios

Figure 2 presents a simplified *Message Sequence Chart (MSC)* illustrating the activities, communication in Data Plane (DASH) and the associated signaling executed in the Control Plane. One can see the Phase 1 and Phase 2 sets of actions, performed by EUT1.

Several types of monitoring activities are performed, described below.

Proactive monitoring: executed in some continuous mode (at SP level and possibly at EUT level- see the “loop” notations in Figure 2); such information is input for the CS selection algorithm (Phase 1), when some new content requests arrive from a given EU to SP. At SP, this means supervision of different servers, maybe networks, and user communities, depending on its policies. SP/CS cooperation on this purpose is envisaged. Such data can be also used to construct a history and updated status of the environment envisaged by the SP. The CSs could be involved in proactive monitoring, provided they are capable to probe the connectivity characteristics towards different groups of users (indicated by the SP).

At EU side, proactive monitoring might be performed, depending on capabilities of the EUT and its SW. In some more complex scenarios the EU can construct history, dedicated to its usual content connections (if they are estimated to be repeated in the future). The terminal context can be evaluated by such measurements, including its access network status.

In-session monitoring: it is performed on a single media flow and measured results are collected in real time, to assess the level of QoS/QoE observed at EU side.

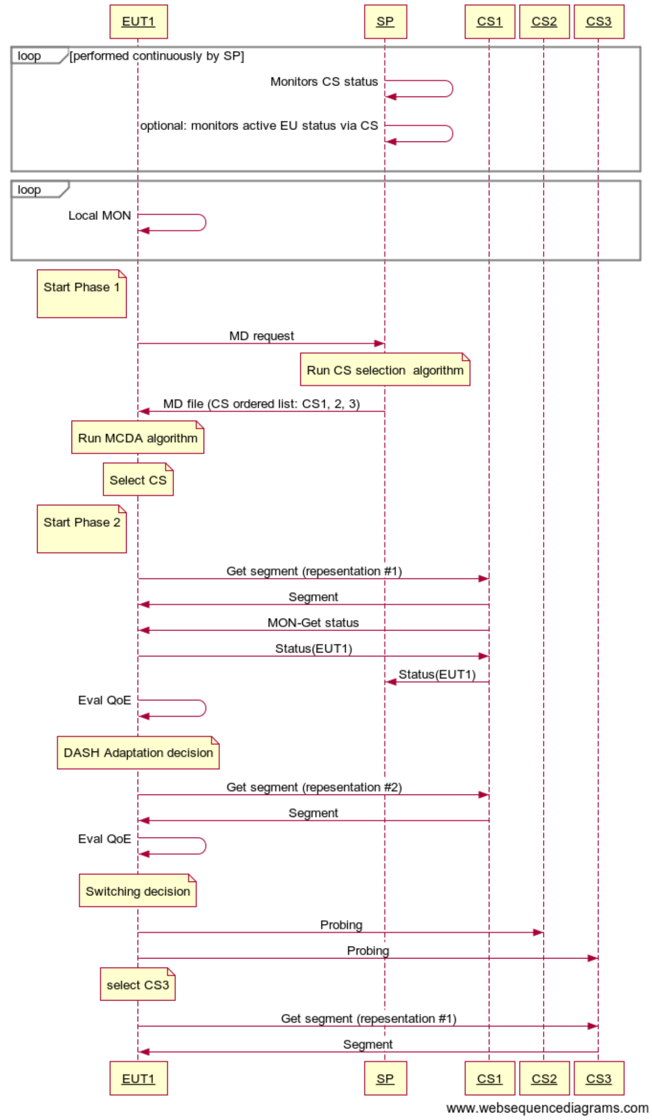


Figure 2. Typical activity and signaling diagram

Note that two kind of information are produced:

- collected by the DASH mechanisms, to serve internally as real time inputs to adaptation decision engine at EU;
- collected by the MON@EUT, which can be consolidated with those produced by the DASH, thus offering a more complete view not only about the reception of he media flow but also on general status and environment of the EUT.

In more complex DISEDAN variants, the SP and/or CS can be involved in such monitoring, at least in being aware of results (note that no SLA concerning mutual obligations of SP/EUs, related to QoE are established in DISEDAN system): for all active users or subsets; for all monitored data or summaries; full or summary monitored values.

Opportunity related monitoring: measurements essentially performed by the EUT to test the opportunity of switching the CS that delivers the content to EU. An example of such category is the probing action of some CS candidates if a CS switching action is prepared.

C. Metrics and MON versions

Apart from DASH defined metrics (in-session observed), the MON subsystem may collect information on:

MON@EUT: CS accessibility (probing); EUT local dynamic context; historical and prediction data on servers and paths utilization.

MON@SP: CS status (collected from CS); active Users status; current load on some paths (here the network monitoring of the NP should cooperate); other dynamic, characteristics of some paths (e.g., loss, jitter); historical and prediction data on servers and paths utilization.

MON@CS: CS status (load); CS environment data (network paths, connectivity paths dynamic characteristics - evaluated at overlay level – where paths are considered from CS to different groups of users; EUTs data, active user groups data.

Therefore the overall MON system design is flexible, since it can combine different features of the above components.

V. IMPLEMENTATION

SP and CS will have an internal database that will contain monitored and/or post-processed data. Also these two entities will be capable to send and receive JSON messages embedded in simple HTTP calls. EU might not have any internal database; it will just have the basic capability to send only simple HTTP calls to either SP or CS.

For Database it has decided to to use *PostgreSQL*, technology [18]. The *PostgreSQL* is a powerful, open source object-relational database system. It runs on all major operating systems, including Linux, UNIX, and Windows.

SP and CS must be able to receive and send simple HTTP messages to each other. For this reason it is needed a web server and a programming language to implement these features. Web server of choice is Node.js [19].

The *Node.js* [19] is an open source, cross-platform runtime environment for server-side and networking applications. The *Node.js* applications are written in JavaScript, and can be run within the *Node.js* runtime on OS X, Microsoft Windows, Linux and FreeBSD.

VI. FUNCTIONAL TESTING

This section presents the functional testing approach and samples of results. To illustrate the real implementation proof of working, some details of the messages collected during test scenarios are inserted. For readers interested in testing philosophy only,   such details could be skipped.

A. Experimental testbed

An experimental laboratory testbed has been built as hardware support for DISEDAN functionalities validation, as presented in Figure 3.

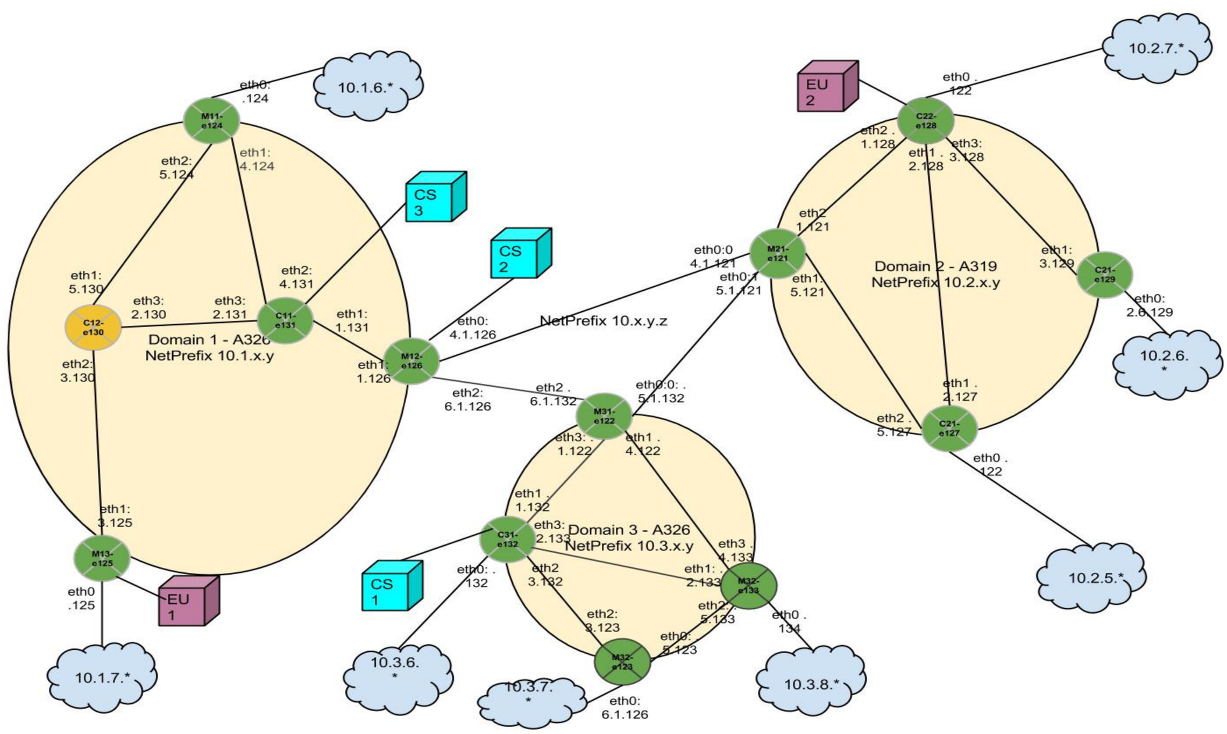


Figure 3. Experimental test-bed structure (multi-domain network)

The system comprises three independent IP network domains, equivalent to some core networks, each of them equipped with several core and edge/border routers (Linux based). No QoS technologies at network layer are active in these networks. Several DISEDAN entities are connected to this network: SP, EU, CS through some access networks. Note that the presence of the access networks in the overall system is not essential, given the OTT-style of work of the system.

B. Functional conformance tests(for initial phase server selection)

The first set of tests has been defined and executed in order to check the correctness of Control Plane. Samples are given below for illustrative purposes. More complete results can be found in [20].

DISEDAN_CT_1 (Unit-level functional testing)

Objective: to assess the functional correctness of each (individual) module. The information messages sent and received by modules are captured and allow to understand the correctness of the steps that each module takes in order to perform its DISEDAN task.

Conditions: the individual component runs and information messages are collected. The software entities irrelevant for the given module under test might not be active (e.g., EUT can run even if monitoring on CS is down; SP should just returns a list of servers).

DISEDAN__CT_2 (System level functional testing)

Objective: to assess the functional correctness of modules cooperation (i.e. the entire DISEDAN signaling chain). Example (simplified) : EUT send a request; SP asks monitoring; Monitoring responds; SP runs MCDA aselection algorithm; SP answers to EUT; The EUT runs its MCDA again; EUT gets content.

Conditions: all the components run in a complete DISEDAN assembly. Info messages are also collected. The difference from the first TC is that now all the software entities should be up and running.

- ***CT_1 Results:***

The following sample sequence of messages shows the results of the test CT1:

- **EUT Functional testing results**

requestServerList*: url = http://141.85.43.130:5000/getserver/1001*
parseServerList*:*
CS address: 141.85.43.132
CS address: 141.85.43.131
CS address: 141.85.43.129
getServersInfo*:*
Open Thread for CS 141.85.43.132
Open Thread for CS 141.85.43.131
Open Thread for CS 141.85.43.129
getNetworkInfo*: Get network information for CS 141.85.43.132*
getNetworkInfo*: Get network information for CS 141.85.43.131*
getNetworkInfo*: Get network information for CS 141.85.43.129*
RTT for 141.85.43.131 is 2.14
RTT for 141.85.43.132 is 2.17
Hop Count for 141.85.43.132 is 16
Hop Count for 141.85.43.131 is 16
141.85.43.129 is down
getBestServer*:*
MCDA Matrix created:
1 0.5 0
0.988182 0.988485 0
1.11111 1.11111 0
MCDA*:*
best server is 141.85.43.132
accessContent*:*
vlc http://141.85.43.132/bunny_1s/BigBuckBunny_1s_isoffmain_DIS_23009_1_v_2_1c2_2011_08_30.mpd

The above sequence proves the correct Control Plane signaling at EUT level.

CT_2 Test:

The basic signalling and message exchange have been tested. The sequence of steps is illustrated in Figure 4:

1. The EUT requests a streaming service from SP. The request contains the ID of the media service requested.
2. The SP gets from its local database the identity of the servers hosting the requested content. Using its monitoring module, the SP interrogates the monitoring agents located on the CSs about the CS state. The main monitoring parameters collected by SP are: CS processor load, CS free memory (normalized at the total memory), number of streaming processes on CS, total bandwidth on the network interface.
3. After receiving the monitoring parameters, the SP selects, using the MCDA, the best servers from the list of servers hosting the requested resource.
4. An ordered list of selected servers is returned to EUT. The best server from SP's point of view is placed on the first position in the list.

5-6. The EUT performs a second selection step. The *Round Trip Time (RTT)* and the distance in terms of *hop count* is measured from EUT to the CSs in the list recommended by SP. The communication on step 5 is performed only with the servers selected by SP. Based on this metric, the best server is finally selected by EUT.

Final step: the EUT requests the service from the selected CS, which start to stream packets towards the EUT.

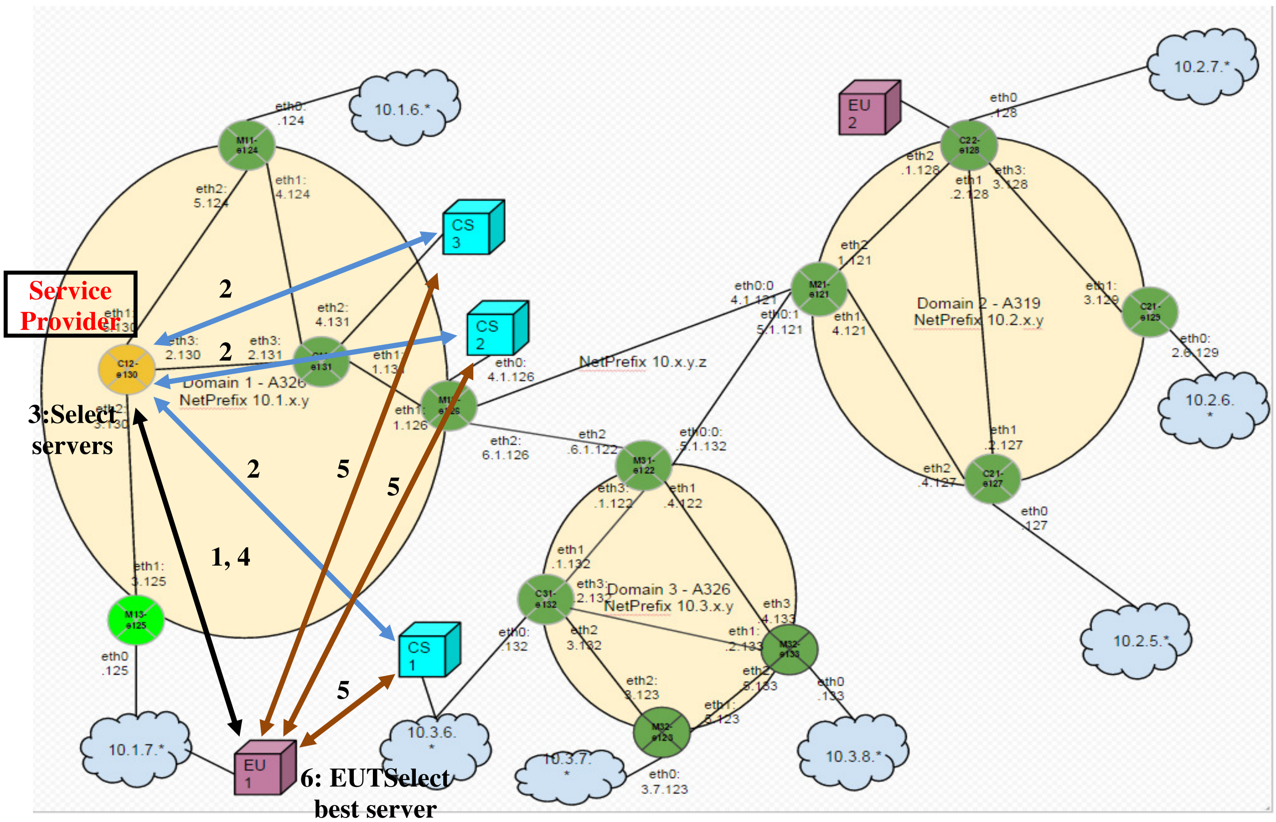


Figure 4. The steps of the functional validation for server selection in DISEDAN testbed

All the functional steps described above have been validated by capturing and analyzing the messages exchanged between EUT, CS, SP.

The following sample sequence of messages shows the results of the test CT_2:

- **EUT functional testing results**

Messages captured on the EUT terminal:

al@e125:~/disedan/project_disedan/DisedanEndUser/bin$ ./DisedanEndUser

STEP1:
requestServerList: *url = http://141.85.43.130:5000/getserver/100*
STEP 4:
parseServerList:
CS address: 10.3.6.132
CS address: 10.1.1.126
STEP5:
getServersInfo:
Open Thread for CS 10.3.6.132
Open Thread for CS 10.1.1.126
getNetworkInfo: *Get network information for CS 10.1.1.126*
getNetworkInfo: *Get network information for CS 10.3.6.132*
RTT for 10.1.1.126 is 0.593
RTT for 10.3.6.132 is 0.999
Hop Count for 10.1.1.126 is 5
Hop Count for 10.3.6.132 is 7
STEP 6:
getBestServer:
Matrix created:
1.5 1
1.00001 1.00411
0.333333 0.555556

MCDA:
best server is 10.1.1.126

STEP 7: accessContent:
vlc
http://10.1.1.126/bunny_1s/BigBuckBunny_1s_isoffmain_DIS_23009_1_v_2_1c2_2011_08_30.mpd
VLC media player 2.2.0-rc2 Weatherwax (revision 2.2.0-rc1-118-g22fda39)
[000000000992118] core libvlc: Running vlc with the default interface. Use 'cvlc' to use vlc without interface.

- **Messages captured at the SP:**

(flask)root@e130:~/serban/microblog# python sp_server_monitoring.py

INFO:werkzeug: * Running on http://0.0.0.0:5000/ (Press CTRL+C to quit)

STEP 1:
Got request from 10.1.7.125 for resource with ID = 100
[Service Provider]: List of servers hosting resource id = 100 :
`['10.1.4.131', '10.3.6.132', '10.1.1.126']`

STEP 2:
[Service Provider]: Network distance list between user and CSes:
[4, 6, 3]

INFO:requests.packages.urllib3.connectionpool:Starting new HTTP connection (1): ***10.1.4.131***
*DEBUG:requests.packages.urllib3.connectionpool:"****GET /loadavg*** *HTTP/1.1" 200 None*
INFO:requests.packages.urllib3.connectionpool:Starting new HTTP connection (1): ***10.3.6.132***
*DEBUG:requests.packages.urllib3.connectionpool:"****GET /loadavg*** *HTTP/1.1" 200 None*
INFO:requests.packages.urllib3.connectionpool:Starting new HTTP connection (1): ***10.1.1.126***
*DEBUG:requests.packages.urllib3.connectionpool:"****GET /loadavg*** *HTTP/1.1" 200 None*
[Service Provider]: CS servers' LOAD list:
[0.0146484375, 0.0146484375, 0.0146484375]

INFO:requests.packages.urllib3.connectionpool:Starting new HTTP connection (1): ***10.1.4.131***
*DEBUG:requests.packages.urllib3.connectionpool:"****GET /system_status?proc=apache*** *HTTP/1.1" 200 None*
INFO:requests.packages.urllib3.connectionpool:Starting new HTTP connection (1): ***10.3.6.132***
*DEBUG:requests.packages.urllib3.connectionpool:"****GET /system_status?proc=apache*** *HTTP/1.1" 200 None*
INFO:requests.packages.urllib3.connectionpool:Starting new HTTP connection (1): 10.1.1.126
*DEBUG:requests.packages.urllib3.connectionpool:"****GET /system_status?proc=apache*** *HTTP/1.1" 200 None*
[Service Provider]: CS servers' number of Apache processes list:
[4, 4, 4]
……………………
Monitoring system – samples of results

The monitoring system has been implemented with two components: a static component and a dynamic one. The static component deals with some static information about the network, content servers and terminals. These data are related to the configuration of the system and are stored in the system's database. They are statically inserted in the database in the initialization phase. The second component is the dynamic one and consists in a monitoring module running at SP and monitoring agents running on each content server. Whenever a new request is received by SO, the monitoring module connects to the monitoring agents requesting monitoring data about the content servers.

Some examples of monitoring data are given below. Figure 5 shows the list of network distances between the access networks known by SP. Figure 6 shows the list of media object resources available at different CSs as they are known at SP level.

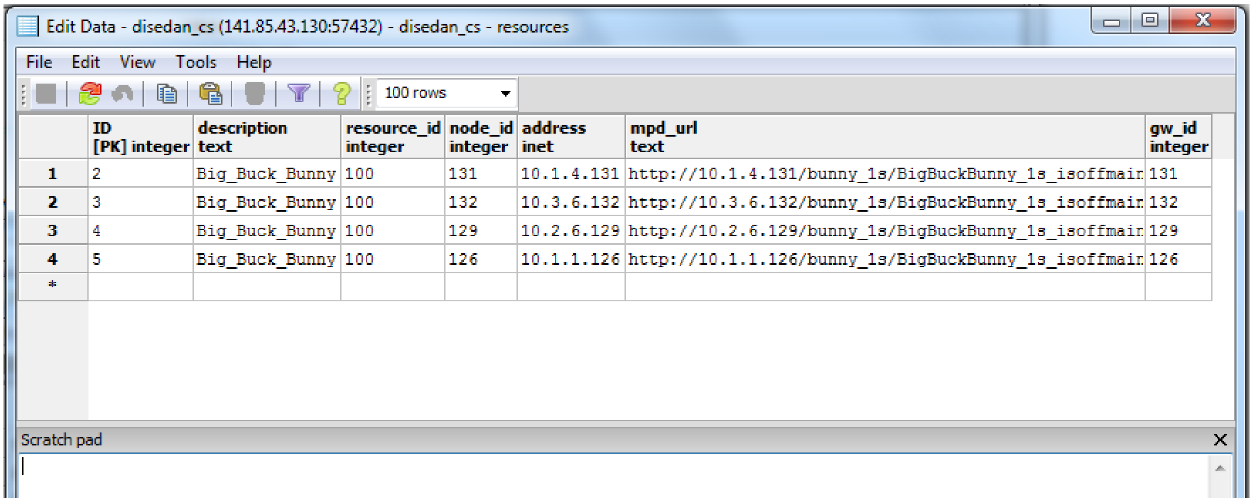
Edit Data - disedan_cs (141.85.43.130:57432) - disedan_cs - peer_nodes

File Edit View Tools Help

100 rows

	ID [PK] integer	node_id_source integer	distance integer	node_id_destination integer	is_egress_source boolean	is_ingress_source boolean	address_source inet	address_destination inet
1	8	125	80	131	TRUE	FALSE	10.1.7.0/24	10.1.4.0/24
2	9	125	25	132	TRUE	FALSE	10.1.7.0/24	10.3.6.0/24
3	10	125	28	129	TRUE	FALSE	10.1.7.0/24	10.2.6.0/24
4	11	128	18	131	TRUE	FALSE	10.2.1.0/24	10.1.4.0/24
5	12	128	25	132	TRUE	FALSE	10.2.1.0/24	10.3.6.0/24
6	13	128	29	129	TRUE	FALSE	10.2.1.0/24	10.2.6.0/24
7	14	125	70	126	TRUE	FALSE	10.1.7.0/24	10.1.1.0/24
8	15	128	17	126	TRUE	FALSE	10.2.1.0/24	10.1.1.0/24
*								

Figure 5. The list of network distances between the access networks known by the SP



	ID [PK] integer	description text	resource_id integer	node_id integer	address inet	mpd_url text	gw_id integer
1	2	Big_Buck_Bunny	100	131	10.1.4.131	http://10.1.4.131/bunny_1s/BigBuckBunny_1s_isoffmair	131
2	3	Big_Buck_Bunny	100	132	10.3.6.132	http://10.3.6.132/bunny_1s/BigBuckBunny_1s_isoffmair	132
3	4	Big_Buck_Bunny	100	129	10.2.6.129	http://10.2.6.129/bunny_1s/BigBuckBunny_1s_isoffmair	129
4	5	Big_Buck_Bunny	100	126	10.1.1.126	http://10.1.1.126/bunny_1s/BigBuckBunny_1s_isoffmair	126
*							

Figure 6. List of resources hosted by the content servers

VII. Conclusions and Future Work

This paper continued the work presented in [1]. It detailed the design concepts, and exposed the guidelines for system implementation of a *media delivery system having a light-architecture and working on top of the current Internet connectivity*. Functional testing for the Control Plane has been described.

Given the complexity of the overall DISEDAN system, the focus of this work was on a part of the system, i.e., the initial content server selection functionality, as preliminary phase of the media session (including media adaptation and/or CS switching) for media consumption. The architectural specification has been refined, then the Control Plane design is described. Implementation characteristics are shortly presented. Finally, some samples of conformance tests results (performed on a real-life testbed) are given, proving the correctness of the Control Plane behavior. It has been checked that the SP correctly performs the initial server selection by running an MCDA algorithm, and delivers to the client an ordered list of servers. Monitoring-related functions have been also tested for conformance.

Experimental results on performances will be reported in other papers.

Future work will be done to evaluate the possibility to extend the DISEDAN architecture for media distribution in the context of wireless and 4G/5G environment, where distributed caching in the access networks is envisaged. Such an environment will be favorable for DISEDAN-like solution in media streaming applications.

Acknowledgment

This work has been partially supported by the Research Project DISEDAN, No.3-CHIST-ERA C3N, 2014- 2015.

References

[1] E. Borcoci, C. Cernat, and R. Iorga, “Control Plane Design for a Content Streaming System with Dual Adaptation,” AICT 2015 Conference, https://www.thinkmind.org/index.php?view=article&articleid=aict_2015_6_40_10154

[2] P. A. Khan and B. Rajkumar, “A Taxonomy and Survey of Content Delivery Networks,” Department of Computer Science and Software Engineering, University of Melbourne. Australia: s.n., 2008. www.cloudbus.org/reports/CDN-Taxonomy.pdf.

[3] J. Choi, J. Han, E. Cho, T. Kwon, and Y. Choi, “A Survey on Content-Oriented Networking for Efficient Content Delivery,” IEEE Communications Magazine, March 2011, pp. 121-127.

[4] “Information-Centric Networking-3,” Dagstuhl Seminar, July 13-16 2014, http://www.dagstuhl.de/en/program/calendar/semhp/?seminar=14291, [retrieved: February, 2015].

[5] http://wp2.tele.pw.edu.pl/disedan/ [retrieved:May, 2015]

[6] T. Dreier, "Netflix sees cost savings in MPEG DASH adoption," 15 December 2011. [Online]. Available: http://www.streamingmedia.com/Articles/ReadArticle.aspx?ArticleID=79409, [retrieved: October, 2014].

[7] I. Sodagar, "The MPEG-DASH Standard for Multimedia Streaming Over the Internet," MultiMedia, IEEE, vol. 18, no. 4, 2011, pp. 62 - 67.

[8] ISO/IEC 23009-1, "Information technology -- Dynamic adaptive streaming over HTTP (DASH) -- Part 1: Media presentation description and segment formats," ISO/IEC, Geneve, second edition, 2014.

[9] 3GPP TS 26.247 v10.1.0, “Transparent End-to-End Packet Switched Streaming Service (PSS); Progressive Download and Dynamic Adaptive Streaming Over HTTP (3GP-DASH),” Release 10, June 2011.

[10] ETSI TS 126 247 V11.7.0 (2014-07) (UMTS); LTE; Transparent end-to-end Packet-switched Streaming Service (PSS); Progressive Download and Dynamic Adaptive Streaming over HTTP (3GP-DASH), (3GPP TS 26.247 version 11.7.0 Release 11, 2014).

[11] A. Beben, J. M. Batalla, W. Chai and J. Sliwinski, “Multi-criteria decision algorithms for efficient content delivery in

content networks," Annals of Telecommunications - annales des telecommunications, Springer, vol. 68, Issue 3, 2013, pp. 153-165.

[12] E. Borcoci, M. Vochin, M. Constantinescu, J. M. Batalla, and D. Negru, "On Server and Path Selection Algorithms and Policies in a light Content-Aware Networking Architecture," ICSNC 2014 Conference, http://www.elcom.pub.ro/disedan/docs/ICSNC%202014%20 Conf.pdf.

[13] J. M. Batalla, C.X. Mavromoustakis, G. Mastorakis, D. Négru and E. Borcoci, "Evolutionary Multiobjective Optimization algorithm for two-phase content source selection process in Content Aware Networks," submitted to Springer 4OR - A Quarterly Journal of Operations Research.

[14] E. Borcoci, ed., et al., D2.1 "System requirements and comparative analysis of existing solutions for media content server selection and media adaptation," July 2014, http://wp2.tele.pw.edu.pl/disedan.

[15] O. Oyman and S. Singh,"Quality of Experience for HTTP Adaptive Streaming Services," IEEE Communications Magazine, April 2012, pp. 20-27.

[16] C. Alberti, D. Renzi, C. Timmerer, C. Mueller, S. Lederer, S. Battista and M. Mattavelli, "Automated QoE Evaluation of DASH," 2013 Fifth International Workshop on Quality of Multimedia Experience (QoMEX), http://infoscience.epfl.ch/record/188813/files/p20-Alberti.pdf, [retrieved: February, 2015].

[17] C. Müller, S. Lederer and C. Timmerer, "An Evaluation of Dynamic Adaptive Streaming over HTTP in Vehicular Environments," www-itec.uni-klu.ac.at/dash/?p=517.

[18] http://www.postgresql.org, [retrieved: March, 2015].

[19] http://nodejs.org, [retrieved: March, 2015].

[20] S.Obreja, E.Borcoci ed., et al., D3.3 "Results of conformance tests," January 2016, http://wp2.tele.pw.edu.pl/disedan