



Content-Aware Distributed Multimedia Provisioning based on Anycast Model towards Future Media Internet

Soraya Ait Chellouche, Daniel Negru, Eugen Borcoci, Eric Lebars

► To cite this version:

Soraya Ait Chellouche, Daniel Negru, Eugen Borcoci, Eric Lebars. Content-Aware Distributed Multimedia Provisioning based on Anycast Model towards Future Media Internet. Consumer Communications and Networking Conference (CCNC), 2011 IEEE, Jan 2011, Las Vegas, France. pp.880 - 885. hal-00999570

HAL Id: hal-00999570

<https://hal.science/hal-00999570v1>

Submitted on 3 Jun 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Context-Aware Distributed Multimedia Provisioning based on Anycast Model towards Future Media Internet

Soraya Ait Chellouche¹, Daniel Négru¹, Eugen Borcoci², Eric LeBARS³

¹CNRS LaBRI Lab. – University of Bordeaux 1
351 Cours de la Libération, 33400 Talence, FRANCE
{soraya.ait-chellouche,daniel.negru}@labri.fr

²University "Politehnica" of Bucharest (UPB),
1-3, Iuliu Maniu Ave., 061071, Bucharest 6, ROMANIA
Eugen.Borcoci@elcom.pub.ro

³Thomson Grass Valley France
40, rue de Bray, 35510 Cesson-Sevigné, FRANCE
Eric.LeBars@grassvalley.com

Abstract—The Future Internet is foreseen to fully handle multimedia services which have experienced an explosive growth in the recent years. Therefore, improving the scalability of these services is getting crucial. One common approach for this purpose is the deployment of multiple replicas of a server throughout the network. The resulting issue is to direct the user request to the *best* server. Methods based on application-layer anycast service constitute good means to (1) support the server's replication strategy and (2) process an appropriate server selection algorithm for the client request according to different metrics.

This paper presents an efficient context-aware anycast multimedia provisioning scheme. In addition to the clients and servers nodes, the adopted architecture involves enhanced Service Registry (SR) nodes, in a distributed mode. SRs have two main roles: first, they collect and maintain the servers' contexts and their content description; second, they perform the mapping of the anycast address of the client request to the unicast address of the most convenient server based on both the client and servers contexts, including the underlying network conditions. The paper presents the evaluation of the proposed video provisioning scheme through intensive simulations. The promising results led to the launch of real developments to be undertaken in the framework of a large-scale European project ALICANTE¹ which objective focuses on Future Media Internet.

Index Terms—Context-awareness, multimedia services provisioning, application-layer anycast, server selection, load balancing, future media Internet.

I. INTRODUCTION

THE rapidly growing of video services known to be very costly services and the heterogeneity of end-users context will place high demand on the Future Internet² in term of network and servers' performance and scalability, content contextualization and management on per-service basis. According to the Cisco Visual Networking Index [1], Global IP traffic will increase by a factor of four from 2009 to 2014 and video alone (TV, VoD, Internet Video, and P2P) will

exceed 91 percent of global consumer traffic. According to same source [1], mobility is one of the main trends on networking. Mobile data traffic will double every year through 2014 and 66 percent of it will be video.

One common technique permitting to increase the scalability of network services is the server replication. It consists on the deployment of multiple replicas of a server throughout the network. The objectives are to increase the service availability and efficiency and also to provide load distribution and fault tolerance. However, this technique gives rise to new challenging issues in the fields of server placement and server selection. In this paper, the focus is put on the server selection issue. The proposed solution is based on the application-layer anycast model.

Anycast was originally introduced by Partridge et al. in [2] within a specific IPv4 class of address. Anycasting in IP environment is defined as a best effort delivery of an anycast datagram to one, but possibly more than one, of the hosts that serve the destination anycast address. An anycast IP address is then assigned to a group of servers that provide the same service. A client trying to reach one of these servers sends a datagram with the anycast address as a destination address. The sent datagram will be delivered to the "nearest" server (according to the routing protocol metric) identified by the anycast address. Anycast model constitutes then a good mean to support server replication.

In this paper, we propose an efficient context-aware application-layer anycast multimedia distribution scheme. The innovation in the proposed approach consists in filtering the multimedia services and contents for selecting the *best* server to handle the request and the related content to stream according to both the user and servers' contexts including the underlying network conditions. Indeed, the proposed server selection strategy permits a preliminary filtering based on the clients' contexts and provides accurate network distance information, using not only the end-to-end delay metric but also the servers' load one. This approach takes in consideration the path capacity and permits to obtain the best paths which may or may not correspond to the least path in term of number of hops or delay, but which definitely offers end-users better Quality of Experience (QoE).

The rest of this paper is organized as follows. Section II gives an overview of related works. Section III first presents

¹ This work is supported by the European research project ALICANTE within the framework of the EU FP7 in ICT, under grant agreement n° 248652/ /ICT-ALICANTE/. <http://www.ict-alicante.eu>

² www.fi-nextmedia.eu

the functional architectural context into which the proposed solution is aimed to be effective and second describes the proposed communication protocol between the infrastructure nodes, along with the resulting functionalities. Section IV presents in detail the core of the proposed approach: the server selection strategy. Section V presents the performance evaluation of our application-layer anycast server selection by depicting and analyzing the results of the simulation process. Section VI brings out the conclusion and the future work, especially the integration of the proposed approach within a large-scale European project for Future Internet and Networked Media convergence, named ALICANTE.

II. RELATED WORKS

DNS[3] deployment and “6to4”[4] router constitute the most widely publicized uses of unicast addressing scheme. However, the anycast model constitutes a good solution for service discovery and hosts auto-configuration which makes it suitable for pervasive environments [5].

The original idea of anycasting was on network layer [2]. Thus, many investigations were made on network-layer anycasting. [6]–[7] focus on the scalability issue, other such as [8]–[9] focus on the design of routing algorithms based on active routers and papers such as [10] and [11] proposed proxy-based infrastructures to address network-layer anycast issues like scalability or session-based services support.

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

The network-layer anycasting limitations led the researchers to define the anycast paradigm at the application layer. In [12]–[13]–[14], the authors examined the definition and support of the anycasting paradigm at the application layer, providing a service that maps anycast domain names into one or more IP addresses using anycast resolvers. Application-layer anycasting appears then as a good solution for distributed Internet services provisioning, especially when it requires no modification in the existing infrastructure. Another motivation to use application-layer anycasting is the ability to manage QoS and define service requirements on a per-service basis.

These papers also presented metrics to measure the anycast performance. The resolver decides which server among the replicated servers is the best one based on the server information. For this purpose, the resolver maintains the servers’ performance information. Paper [13] identified and tested four approaches for this: (1) remote server performance probing, (2) server push, (3) probing for locally-maintained server performance and (4) user experience.

The work performed in [15] overviews the anycast researches. It also proposes a requirement-based probing algorithm and compares it to the periodical probing algorithm proposed by Bhattacharjee et al. [12]. The conclusions arising from paper [15], based on the queuing theory, are the following: first, the requirement-based algorithm is better than the periodical probing algorithm for

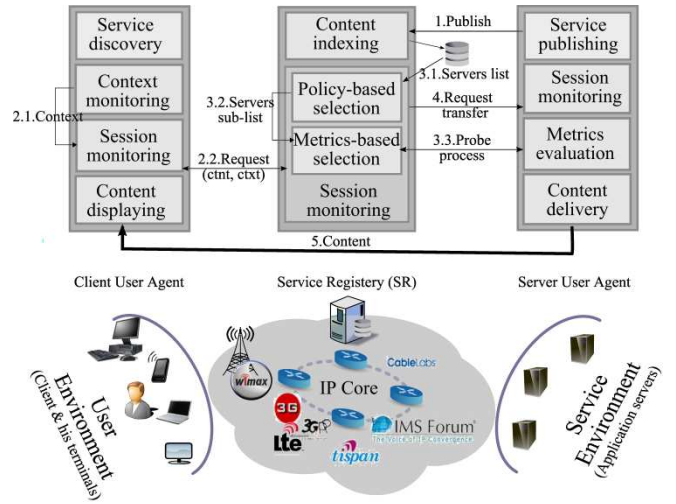


Fig. 1. Functional System Architecture towards Context-Aware Anycast Model Deployment.

each query; second, the average waiting time for all the queries of the requirement-based probing algorithm is shorter than the periodical probing algorithm one.

In [16] and [17], the authors propose concepts of application-layer anycasting in the framework of multimedia services. [16] presents an algorithm theoretically related to an economic model with a queuing theory based on the available free buffer, the available bandwidth, the average arrival rate of requests and the call blocking probability. [17] proposes three anycast-based multimedia distribution architectures, namely the identical, the heterogeneous and the semi heterogeneous architectures, to identify the best media server selection for different application domains.

Except in [17], all the above works have based their selection strategies on the servers’ performance and have considered neither the client context nor the network conditions which hardly influence the multimedia services. However, the solution presented in [17] is only designed and evaluated in small-scale environment.

III. CONTEXT-AWARE ANYCAST MODEL AND SYSTEM ARCHITECTURE

The application-layer anycast architecture adopted in this paper aims to provide clients with an efficient and transparent multimedia provisioning service. We assume that each Autonomous System (AS) hosts its own Service Registry (SR) node which will receive the requests of its related clients. The SR has two main roles: first, it will perform the mapping of the anycast address of the client request to the unicast address of the most appropriate server and, second, it will retrieve and maintain the servers’ contexts and their contents descriptions. We also assume that the set of SRs can collaboratively perform the server selection feature. The infrastructure, as illustrated in Figure 1, is then based on three types of nodes: the *client’s nodes*, in the user environment that request the service, the *server’s nodes*, in the service environment, that provide the service and the *SRs nodes* that handle the client’s requests and perform the server selection strategy. This strategy, and its joint algorithm, will be detailed in the next section (section III). Figure 1, also illustrates a functional architecture of the

proposed solution in a Future Internet context. For each node type, it gives the different functional blocs that compose it. It also depicts the dialogue engaged between the different agents in order to establish the media session. This process is described in detail in the following sub-sections.

The agents are communicating using SIP (Session Initiation Protocol) [18] which is a signaling protocol to handle multimedia session and its extension for event state publication [19]. SIP is foreseen to become the key signaling protocol for Next Generation Networks (NGN) platforms. Even though coming from the Internet world, it has already taken over the Telecommunication world (e.g. Tispan, IMS, 3GPP [20]) and operators place high expectations on it. Therefore, it appears as the predominant candidate for Future Internet signaling. Following this conjecture, the designed communication protocol has been based on SIP as illustrated in Figure 2.

A. Service publishing

We consider in this work that the servers have heterogeneous contexts (output link bandwidth, location, etc.) and provide clients with different contents. Each node that wants to act as a server should send a *SIP Publish* message within the anycast service address as illustrated in Figure 2. The request contains its context and the list of the contents that it will deliver with their descriptions. In the case of Video Streaming service, the description corresponds to the video content reference, the available coding formats, the resolution, the bitrate and the language of the video. The request will then be directed to the nearest SR by the underlying routing protocol. This SR caches the received information and replies to the server with a final 200 ok response.

B. Servers probing

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

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

C. SRs collaboration

The SRs are interconnected in a multicast scheme. When a SR receives a client request and concludes, after processing it, that the content cannot be delivered by its registered

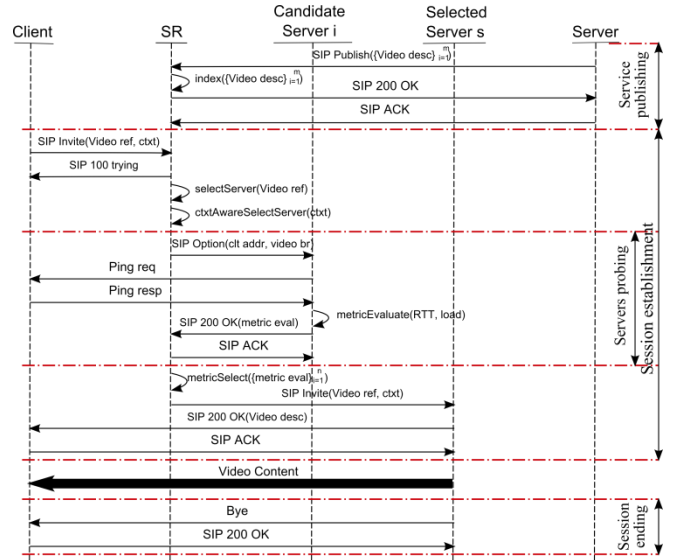


Fig. 2. Message sequence for server selection by the SR and media delivery session.

servers (or that the required QoS cannot be ensured), it multicasts the request to the other SRs. Each SR processes the received request and selects, if possible, a suitable server among its registered servers. Then, if the selection has led to a result, it responds to the original SR of the request with a *200 OK response* containing the IP address of the selected server and its evaluation of the selection function. Otherwise, it responds with a *404 not found* response. The original SR selects then the most suitable server among the received responses and forwards the client request to it. In case the SR receives only *Not Found* responses, the request fails and the client is notified.

D. Session establishment

A client/server video session establishment process, illustrated in Figure 1 and detailed in Figure 2, is described in the following:

- 1) We assume that the servers have published their contents as explained in the session publishing sub-section;
- 2) The Client trying to access the anycast service initiates a session with a simple *anycast Invite request* that contains its context and the requested content;
- 3) The request is then routed to the nearest SR which will retrieve the client context and requirements and try to select, alone or in collaboration with the others SRs, the most suitable server among a set of candidate servers. The result of the selection is based on the information retrieved in the probing stage;
- 4) If the SR succeeds to select a server, it forwards to the latter the client request. Otherwise, it sends to the client an error response to inform it that its request has failed;
- 5) In the case where the selected server receives the client *Invite request*, it processes it and confirms the establishment of the session. It finally starts the streaming of the requested content to the client.

IV. CONTEXT-AWARE SERVER SELECTION ALGORITHM FOR VIDEO SERVICE DISTRIBUTION

As explained previously, our proposed approach for video distribution is mainly based on the selection of the *best*

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

A. Policy-based filtering

Each SR maintains at its side all the list of video contents published by the attached video streaming servers. For each video, it maintains the set of servers that deliver it and for each of these servers the set of technical parameters that characterize the video at the server side.

When requesting a service, the client specifies in addition to the requested video reference, its context (available bandwidth, terminal resolution, etc.) and preferences like the video language by including them in the *SIP Invite request* that initiate the service. The registered services at the SR side are then filtered by a set of predefined policies in order to only keep the services that deliver contents that match the user context. The policies define the mapping, from one side, the maintained servers’ contexts and related contents information and, from the other side, the client context and requirements. These policies are typically *Boolean*, in the sense that a service is either match the context or not.

B. Metrics-based filtering

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

In this paper, we address the video streaming service, which is known to be very sensitive to the packet loss metric. Therefore, the main requirement that we have considered when designing our server selection strategy is to avoid congestion and this at different levels. At the policy-based step, we have considered the congestion at the client level by taking into account the client available bandwidth.

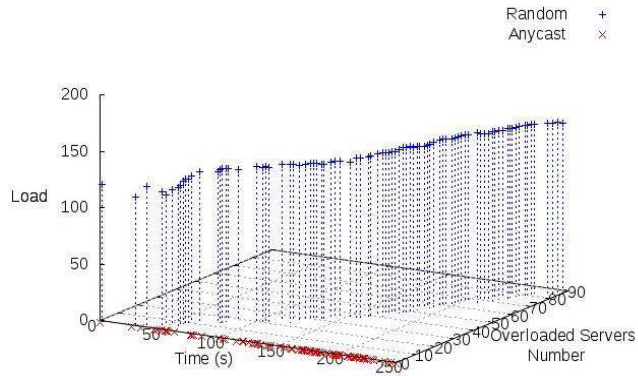


Fig. 4. The overloaded servers’ number and average load.

```

F(Ac, Rbrc) {
    dsc = delay(Ac);
    if ( ( ( ∑i=1n Rbri + Rbrc ) / br < 1 ) then {
        return dsc;
    } else {
        return ( α * dsc * ( ( ∑i=1n Rbri + Rbrc ) / br ) );
    }
}

```

Fig. 3. The server selection function algorithm.

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

The SR probes all the servers that constitute the retrieved sub-list, from step 1, in order to evaluate for each of them the server selection function F described in Figure 3. Because F combines The server-to-client delay and the server load, the he probe *SIP Option request* must contain the client address A_c and the required video bitrate Rbr_c . On the other side, the server must also maintain its load. Indeed, whenever, the server accepts the establishment of a multimedia session or ends one of its current sessions, it must update its load. The current load is calculated as follows:

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

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

The probed servers evaluate then the function F . As illustrated in Figure 3, F is based on the server-to-client delay d_{sc} and the server load. As the server is not overloaded, it only takes into account the client-to-server

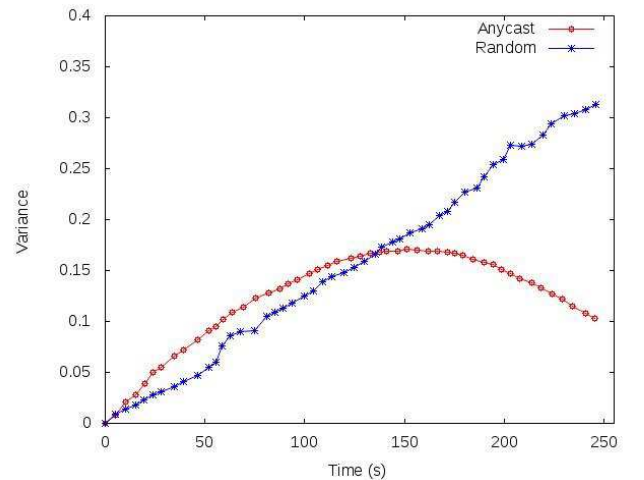


Fig. 5. The servers’ load variance.

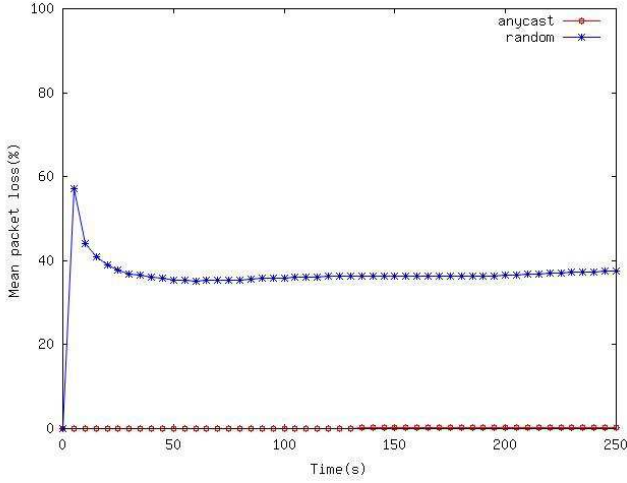


Fig. 6. The mean packet loss at all active sessions.

delay. But as soon as the server is overloaded, the function involves both the two metrics and the priority is inversed. The α parameter should be fixed by the network administrator according to the network topology in order to give the server load metric the top priority.

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

$$F_s(A_c, Rbr_c) = \min_{1 \leq i \leq m} (F_i(A_c, Rbr_c)) \quad (2)$$

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

V. PERFORMANCE EVALUATION

A. Simulation environment

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

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

B. Simulation metrics and results

For evaluating the effectiveness of our video distribution protocol, we compare it to the random server selection

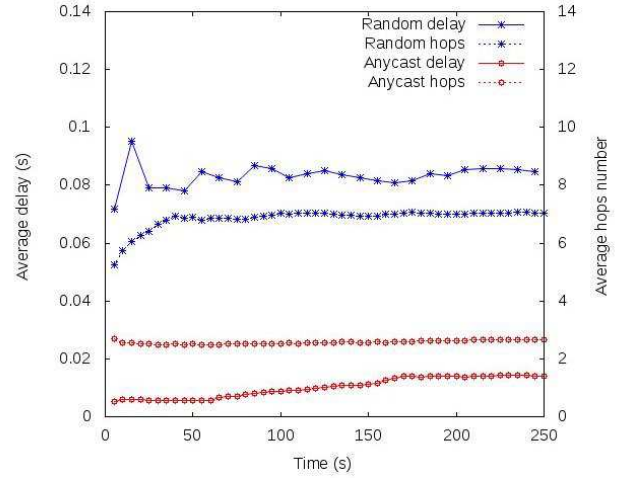


Fig. 7. The average path distance (delay & hops number).

TABLE I
ADDITIONAL SIMULATION RESULTS

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

scenario based on the uniform distribution that we simulated in the same environment. In this paper, the comparison is done according to different metric parameters: the average servers' overloads, the server load variance, the average packet loss, the average path distance (delay and hop number) and some additional results such as the selection overhead, the services throughput, etc.

1) *Overload*: Figure 4 depicts the evolution of both the number of overloaded servers and their mean charge. As illustrated in the figure, no server was experiencing overload during all the simulation time under our approach. By contrast, the overloaded servers' percentage reaches 17.4% after initiating all the sessions under the random approach. The mean overload of these servers varies between 120% and 150%, which induces congestion at the servers' side.

2) *Servers' load variance*: Figure 5 depicts the servers' loads variance versus time. In the first stage of the simulation [0s, 137s], the random selection approach is experiencing a less servers' loads variance than our approach, reflecting the fact that in our approach we give the top priority to the path delay in the calculating of the network distance as the servers are not overloaded. However, in the second stage [137s, 250s], the variance in the random selection continues to increase while it decreases in our approach. This is the result of giving the priority to the servers' load when calculating the network distance in this stage. The curves illustrate a more efficient spread of the clients' requests in our approach.

3) *Packet loss*: Figure 6 depicts the average packet loss of all the initiated sessions versus time. We can note that the packet loss under our approach is almost none. It approaches 0.22% with 1000 active sessions. On the contrary, the selection scenario reaches a loss percentage greater than 35% during all the simulation time, and this with less traffic to manage (cf. table 1). The effectiveness of our approach is

the result of better congestion avoidance at all levels. Especially, at the client where the loss is known to be important. Indeed the *last-mile* is known to be the most common causes of packet loss and consequently, the video service degradation over the end-to-end path. The proposed server selection corrects this problem in its first stage by selecting only the services that cope with the user context.

4) *Average path distance*: Figure 7 represents the average path distance, represented in term of path delay and path hops number. We can clearly note that the mean path distance is reduced by our solution for both metrics. Indeed, under our approach, the mean path delay varies in the interval [5ms,14ms] and the mean hops number is almost 2.5 while, under the random scenario, the mean path delay varies in the interval [71ms,96ms] and the mean hops number varies in [5.2, 7.06]. These results reflect the consideration of the server-to-client delay metric in our selection strategy. Indeed, the SR always selects the nearest server among the non overloaded candidate servers. Thus we can conclude that our approach ensure a better service delay but also a better distribution of clients' requests on servers and consequently a better congestion avoidance at network level as seen in Figure 6.

Table 1 summarizes some additional results such as the delivered content matching to the client context, the percentage of inter AS sessions, the average SIP overhead, the service throughput and the service response time. We can note that thanks to the context-aware selection, the delivered video content always meet the client context contrarily to the random selection scenario where the percentage is 33.7%. We can also note that although the SIP traffic overhead (due to the exchanged messages in order to perform the server selection) is greater under our approach. However, it is not significant comparing to the total service throughput. The additional processes necessary to perform the selection strategy also induce a longer service response time (the duration between the requesting of the service by the client and the establishment of a media session between the client and the selected server), as seen in Table 1. However, the service average response time under our approach whose average is 1.05 s is far from reaching the 32 s fixed in [18] to conclude that the SIP session has expired.

The evaluation of our solution detailed in this section proves its effectiveness and this with a non significant traffic and response time overhead. These promising results led to the launch of real developments in the framework of a large-scale European project ALICANTE which objective focuses on Future Media Internet.

VI. CONCLUSION

This paper proposed a context-aware video delivery solution based on an anycast model to be used within the Future Internet architectural context. The major enhancement consists in the design of an efficient server selection strategy that copes with the video services requirements and the end user experience. Indeed, the proposed strategy combines multiple filters based on both context-aware policies and accurately measured metrics for selecting the most suitable server for each client request. The simulation results presented in the previous section confirm the effectiveness of this selection strategy.

Another advantage of the proposed video delivery scheme is its ability to be easily integrated in the next generation networks platforms. Indeed, the obtained results provide promising aspects towards the evolution of media servers reaching home users devices and permitting users to become content providers. To this end, further work will be dedicated, first, to expand to other multimedia-related applications and, second, to achieve real developments and large scale evaluations of the solution in the framework of the large-scale European project ALICANTE that directly deals with future media networks, aimed to deploy a concept for a new user-centric "Networked Media Ecosystem".

VII. REFERENCES

- [1] Cisco Visual Networking Index, 2009. Available : www.cisco.com
- [2] C. Partridge, T. Mendez and W. Milliken, "Host Anycasting Service", RFC1546, 1993.
- [3] T. Hardie, "Distributing Authoritative Name Servers via Shared Unicast Addresses", RFC 3258, April 2002.
- [4] C. Huitema An Anycast Prefix for 6to4 Relay Routers, RFC3068, June 2001
- [5] Zhang Li Shi Zhen-lian Shen Qi, "A Service Discovery Architecture based on Anycast in Pervasive Computing Environments", Computer Software and Applications Conference, 2007.
- [6] Dina Katabi, and John Wroclawski, "A Framework for Scalable Global IP-Anycast (GIA)", SIGCOMM'00, Stockholm, Sweden, 2000.
- [7] Wang Xiaonan Qian Huanyan, "An Anycast Communication Model on Solving Scalability in IPv6", Control, Automation, Robotics and Vision, 2006.
- [8] Hirokazu Miura and Miki Yamamoto, "Server Selection Policy in Active Anycast", IEICE Trans. Commun., Vol. E84.B, No. 10 October 2001.
- [9] H.B. Hashim, J.A. Manan, "An active anycast rtt-based server selection technique", Proceedings of 13th IEEE International Conference on Networks 1 (2005) 145–149.
- [10] H. Ballani, P. Francis, "Towards a global IP Anycast service", Proceedings of the 2005 conference on Applications, technologies, architectures, and protocols for computer communications 301–312, 2005.
- [11] Tim Stevens, Joachim Vermeir, Marc De Leenheer, Chris Develder, Filip De Turck, Bart Dhoedt, Piet Demeester, "Distributed Service Provisioning Using Stateful Anycast Communications", lcn, pp.165–174, 32nd IEEE Conference on Local Computer Networks, 2007.
- [12] Samrat Bhattacharjee, Mostafa H. Ammar, Ellen W. Zegura, Viren Shah, and Zongming Fei, "Application-Layer Anycasting," IEEE INFOCOM'97, Kobe, Japan, April 1997.
- [13] E. Zegura, M. Ammar, Z. Fei, and S. Bhattacharjee. "Application-layer anycasting: A server selection architecture and use in a replicated web service". ACM/IEEE Transactions on Networking, 8(4):455–466, 2000.
- [14] Zongming Fei, Samrat Bhattacharjee, Ellen W. Zegura and Mostafa Ammar, "A Novel Server Selection Technique for Improving the Response Time of a Replicated Server", INFOCOM'98, 1998.
- [15] S. Yu, W. Zhou, Y. Wu, "Research on network anycast", Proceedings of Fifth International Conference on Algorithms and Architectures for Parallel Processing 154–161, 2002.
- [16] Zheng da Wu, Chris Noble, and D. Huang. "Optimal Video Distribution Using Anycasting Service", Information Technology papers, 1999.
- [17] Hsu-Yang Kung, Chung-Ming Huang, Hao-Hsiang Ku, Ching-Yu Lin, "Efficient multimedia distribution architecture using anycast", Computer Communications, v.30 n.2, p.288–301, January, 2007.
- [18] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, R. Sparks, M. Handley, E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, June 2002.
- [19] A. Niemi, "Session Initiation Protocol (SIP) Extension for Event State Publication", RFC 3903, October 2004.
- [20] 3GPP Forum, "IP Multimedia Subsystem (IMS)", Stage 2, 3GPP TS 22 228 v8.11, 2009-12-14, 2009.