



HAL
open science

Anycast-based context-aware server selection strategy for VoD services

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

► **To cite this version:**

Soraya Ait Chellouche, Daniel Négru, Eugen Borcoci, Eric Lebars. Anycast-based context-aware server selection strategy for VoD services. IEEE GLOBECOM Workshops, Dec 2010, Miami, FL, United States. pp.1513 - 1517, <10.1109/GLOCOMW.2010.5700192>. <hal-00877597>

HAL Id: hal-00877597

<https://hal.science/hal-00877597v1>

Submitted on 28 Oct 2013

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.



HAL Authorization

Anycast-based Context-Aware Server Selection Strategy for VoD Services

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—Internet video services have experienced an immense popularity in the recent years and will probably be the dominant applications of Future Internet. Therefore, the design of scalable video distribution systems that on the one hand, improve the end-users experience and deal with their context heterogeneity and on the other hand, optimize the cost of these services, is becoming a very important and challenging issue. One common approach for this purpose is the server replication. Nevertheless, this approach brings new issues such as the server selection (how to direct the user request to the closest replica), the servers' placement, etc.

This paper deals with the server selection issue. Thus, we present a two-level filtering based server selection. The policy-based filtering retrieves the list of servers whose services matches the client context while the metric-based filtering select from this list the most convenient server that on the one hand, avoid the servers overload and in the other hand, copes with the underlying network state and capabilities. The paper also presents the preliminary evaluation of the proposed video provisioning scheme. The promising results led to more extensive evaluation of the approach and later to the launch of real developments in the framework of a large-scale European project which objective focuses on Future Media Internet.

Index Terms—Context-awareness, video delivery, application-layer anycast, server selection, future media Internet.

I. INTRODUCTION

THE Internet multimedia services (VOD, IPTV, etc) have been experiencing an explosive growth in the recent years, phenomenon which will continue and become one of the main requirements of the Future Internet¹. The popularity of these services has led to a high demand for network bandwidth and servers performance. According to [1], with the deployment of these new video services, the existing infrastructures will be pushed to their limits. To accommodate the needs of these services, networks must be able to scale and to provide quality of service (QoS) and security on an end-to-end basis.

One common technique permitting to increase the scalability of network services is the server replication. It enables 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, new issues arise

concerning the server placement and the server selection. In this paper, the focus is put on the server selection issue for which a new anycast-based server selection process is proposed.

Anycast was originally introduced by Partridge et al. in [2] within a specific IPv4 class of address. The IP anycasting is defined as a stateless best effort routing service able to deliver the anycast datagram to at least one of the hosts that serves the 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. The authors present it as a solution for service discovery and hosts auto-configuration which makes it suitable for pervasive environments [3].

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. 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. Many investigations were made on network-layer anycasting scalability issues [4]-[5], other focused on the design of routing algorithms based on active routers [6]-[7] and papers such as [8] and [9] proposed proxy-based infrastructures to address network-layer anycast issues like scalability or session-based services support.

The network-layer anycasting limitations led the researchers to define the anycast paradigm at the application layer. In [10]-[11]-[12], 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 maintained servers' performance information at its side. Paper [10] identified and tested four approaches for this: (1) remote server performance probing, (2) server push, (3)

¹ www.fi-nextmedia.eu

probing for locally-maintained server performance and (4) user experience.

The work performed in [13] 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. [10]. The conclusions arising from paper [13], based on the queuing theory, are the following: first, the requirement-based algorithm is better than the periodical probing algorithm for 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 [14] and [15], the authors propose concepts of application-layer anycasting in the framework of multimedia services. [14] 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. [15] 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 [15], 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 [15] is only designed and evaluated in small-scale environment.

In this paper, we propose an efficient context-aware application-layer anycast multimedia delivery scheme. The innovation in the proposed approach consists in mapping the video service requirements to both the user and server contexts including the underlying network conditions. Indeed, the proposed server selection strategy provides accurate network distance information, using not only the delay metric but also the load one. This approach takes in consideration the path capacity, the available bandwidth and permits to obtain the best paths which may or may not correspond to the least path delay, but which definitely offers better bandwidth and packet loss rates.

The rest of this paper is organized as follows. Section II first presents the system architecture into which the proposed solution is aimed to be effective and second describes the proposed communication protocol between the infrastructure nodes. Section III details the proposed approach: the server selection strategy. Section IV presents the performance evaluation of our application-layer anycast server selection by depicting and analyzing the results of the simulation process. Section V 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 ALICANTE².

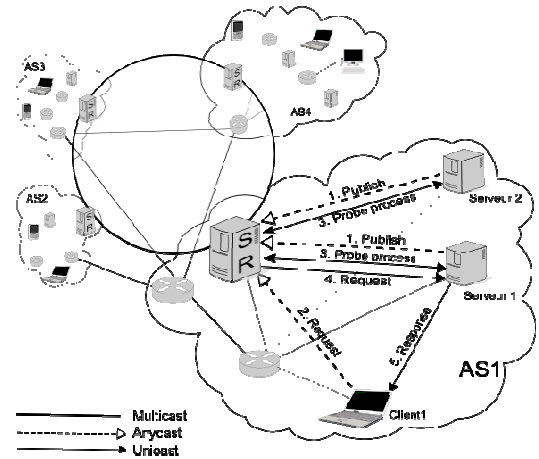


Fig. 1. The Application-Layer Anycast System Architecture .

II. SYSTEM ARCHITECTURE

The aim of the anycast system architecture depicted in figure 1 is to provide end users with an efficient context-aware video delivery service. For this three types of nodes take part in the communication: the client nodes that request and consume the service, the server nodes that provide the service and the Service Registry (SR) nodes that act as intermediates between the clients and the servers. This latter are distributed through the network.

Each server that enjoys the service has to publish the description of its contents to the SR. When the client requests the service, its request is sent to the SR. The latter's role is to redirect the request to the server that best suits the client context. The way in which this redirection is done will be explained in section III. The following sub-sections present the detail of the sub-processes that compose the video session establishment.

As illustrated in Figure 2, the communication protocol between the different nodes of the infrastructure is based on SIP (Session Initiation Protocol). SIP is a widely used signaling protocol to establish and manage a multimedia session. Originally designed by the Internet word [16]-[17], it is also accepted as a 3GPP signaling protocol [18] and a part of IP Multimedia Subsystem IMS architecture. Therefore, it appears as the predominant candidate for Future Internet signaling.

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 server should send a *SIP Publish request* 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.

² 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>

B. Servers probing

To maintain its server's context information, the SR must probe the servers. It can do it in either a proactive or a reactive manner. To ensure the accuracy of the retrieved information, the probing is done in a proactive manner. In addition, one of the metrics on which our server selection strategy is based is the server to client delay, for which we need to transmit the client's address to the server. 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 evaluates its weighted network distance to the client and sends the result to the interrogating SR in a *SIP 200 ok response*. Thanks to this probing, the SR obtains accurate information about the servers and network performance and can thus select the most suitable server for the client request. The comprehensive detail of the used network distance and 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 concludes, after processing, that the received client request can't be responded or the required QoS cannot be ensured), it multicasts it to the other SRs. Each SR processes the received request and selects, if possible, a suitable server among its registered servers and responds to the original SR of the request with a *200 OK response* containing the IP address of the selected server and its network distance evaluation. Otherwise, it returns 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 when joining the service;
- 2) The Client that wants to access the service initiates a session with a simple *anycast Invite request* that contains its context and the requested content;
- 3) The request is then directed to the nearest SR which will try to select, alone or in collaboration with the others SRs, the most suitable server among a set of candidate servers based on the servers' information retrieved in the probing stage;
- 4) If the SR succeeds to select a server, it forwards to it the client request. Otherwise, it notifies the client that its request has failed with an error response;
- 5) In the case where the selection succeeds the selected server receives the client *Invite request* and confirms the establishment of the session. Finally, the media streaming can be started.

III. SERVER SELECTION STRATEGY

The proposed solution for context-aware VoD delivery is mainly based on the selection of the *best server* for each *server*, we mean the server that best suits both the client

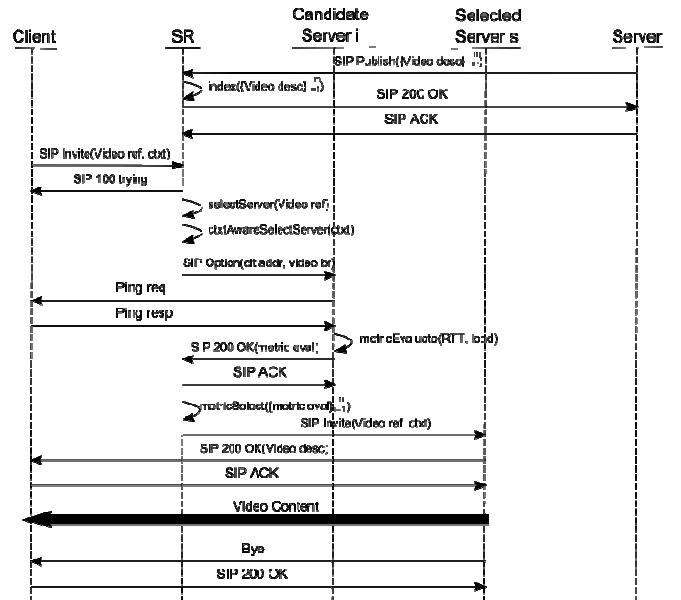


Fig. 2. Message sequence for server selection by the SR and media delivery session.

environment (connectivity and terminal characteristics) and the requirements of the underlying network conditions for finally improving the perceived Quality of Experience QoE at the client side. In this respect, a two-level filtering technique has been conceived: the first is based on “policy-based filtering”, the second on “metrics-based filtering”.

The first filtering level aims to select the server that both deliver the requested content and meet the user context and requirements. For this purpose, in addition to the list of the video contents delivered by the service, and the associated video streaming servers which published them, the SRs maintain for each video its description (the technical parameters that characterize it).

When requesting a service, the client specifies in addition to the requested video, context information such as his preferences, the serving terminal characteristic and the access network parameters.

The context-aware selection is done by mapping, from one side, the maintained servers' information and, from the other side, the client context and requirements. This mapping is done according to *Boolean* policies. At the end of this phase, the SR disposes of the subset of servers satisfy the set of predefined policies.

As for the second filtering level, it aims to select one server (the best) from the constructed sub-list in the previous level. This selection is made based on a combination of metrics evaluated by probing the servers of the sub-list. The choice of the metrics and their exploitation strategy are directly related on the triggered application.

The VoD service, addressed in this paper is known to be very sensitive to the packet loss. 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 context. At this level (metrics-based), we consider the congestion at both the server and network levels. Thus, the defined filter is a weighted function of two metrics: the server load and the server-to-client delay. The

combination of these two metrics permit to avoid congestion at both the server side by avoiding overload and at the network side by considering the current client-to-server delay. The evaluation of this function is as in the 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 network distance between it and the client. For this purpose, the probe request must contain the client address A_c and the required video bitrate Rbr_c . 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.

Each probed server evaluates the distance function F . As the server is not overloaded, it only takes into account the client-to-server delay d_{sc} . But as soon as the server is overloaded, the distance involves both the two metrics and the priority is inverted as follows:

$$F(A_c, Rbr_c) = \left(d_{sc} * \left(\left(\sum_{i=1}^n Rbr_i + Rbr_c \right) / br \right) \right) \quad (2)$$

Where, A_c is the address of the client c and is the bitrate that well match the context of the client c .

After receiving the evaluations of the distance from the probed servers, the SR selects the best server s such as:

$$F_s(A_c, Rbr_c) = \min_{1 < i < m} (F_i(A_c, Rbr_c)) \quad (3)$$

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.

IV. PERFORMANCE EVALUATION

A. Simulation environment

In this section, we evaluate the proposed context-aware

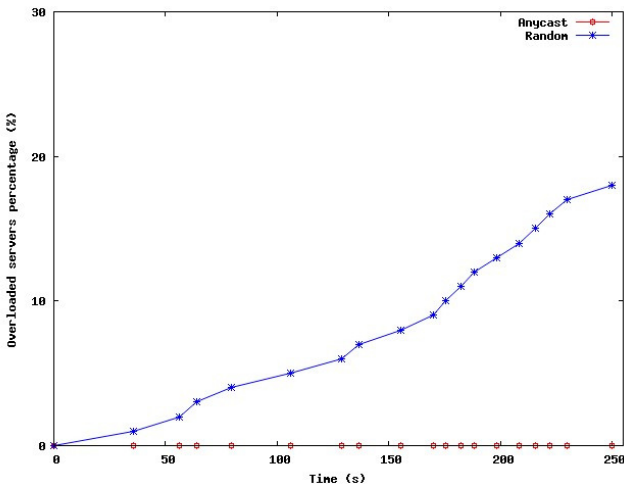


Fig. 3. The overloaded servers' percentage.

anycast video delivery approach. The simulations were done using the Network Simulator NS2. The network consists of 5 Autonomous Systems (AS). Each AS contains 200 clients, 100 servers, one server registry and 20 routers. The connectivity between routers is constant at 2 Gb/s. Servers also have a constant connectivity of 20Mb/s. Clients connectivity however, varies in $[0.512, 100]$ Mb/s.

The simulated video streaming service is providing clients with 10 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 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 5 ASes. All the videos have a minimum duration of 250s in order to keep active all the 1000 video sessions simultaneously.

The proposed solution is compared to the random server selection scenario that we have simulated in the same conditions. The evaluation is done according to different metric parameters: the percentage of overloaded servers, the average packet loss, the average path distance (media streaming delay) and some additional metrics.

1) *Overload*: Figure 3 depicts the evolution of the number of overloaded servers versus simulation time. As illustrated in the figure, no server was experiencing overload during all the simulation time under our approach. By contrast, the

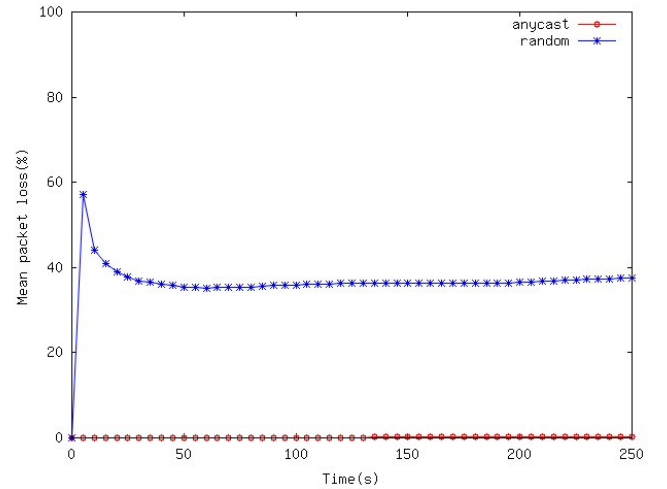


Fig. 4. The mean packet loss at all active sessions.

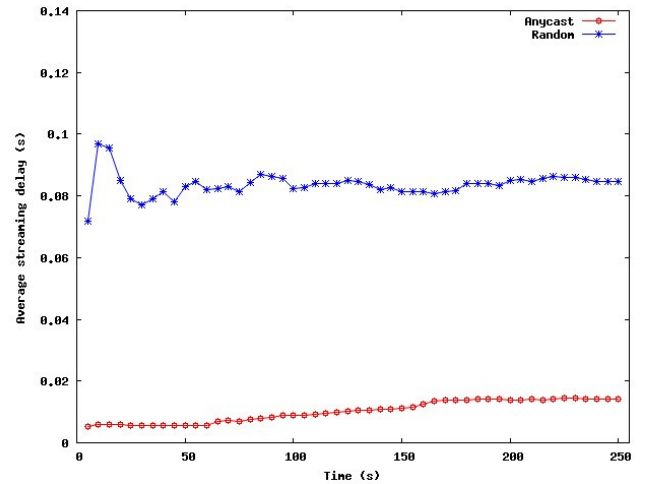


Fig. 5. The mean RTP packet delay.

overloaded servers' percentage increases throughout all the simulation time to reach 18% after initiating all the sessions under the random approach. The mean overload of these servers varies between 120% and 150%, which fact induces congestion at the servers' side.

2) *Packet loss*: Figure 4 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 random selection scenario reaches a loss percentage greater than 35% during all the simulation time. The effectiveness of our approach is the result of better congestion avoidance at all levels by considering (1) the user context : the delivered video content always meet the client context contrarily to the random selection scenario where the percentage is 33.7% ; the servers load as seen in Figure 3 and the underlying network conditions by considering the server-to-client delay.

3) *Average path distance*: Figure 5 represents the average path distance, represented in term of path delay. 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 [5, 14]ms while, under the random scenario, the mean path delay varies in the interval [71, 96]ms . 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.

The server selection process induces overhead in term of 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) and signaling throughput. Indeed, the SIP throughput is greater under our approach. Its average is 74.59 KB/session. However, it is not significant comparing to the total service media throughput whose average is 219.8 MB/session. As for service response time, whose average is 1.05 s, it is far from reaching the 32 s fixed in [17] to conclude that the SIP session has expired.

V. CONCLUSION

In this paper, we presented a context-aware video delivery solution based on the application-layer anycast model to be used within the Future Internet architectural context. The enhancement consists in the design of an efficient server selection strategy that takes into account both the video services requirements and the end user context. Indeed, the proposed strategy relies on context-aware policies and accurately measured metrics for selecting the most suitable server for each request. The simulation results confirm the effectiveness of the solution and this with a non significant traffic and response time overhead.

In addition, the proposed solution can be easily integrated in the next generation networks platforms. Further work will be dedicated, first, to fully explore the solution with more intensive simulation 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".

VI. REFERENCES

- [1] Cisco Systems White Paper, "Optimizing Video Transport in your IP Triple Play Networks", 2006. Available: www.cisco.com
- [2] C. Partridge, T. Mendez and W. Milliken, "Host Anycasting Service", RFC1546, 1993.
- [3] Zhang Li Shi Zhen-lian Shen Qi, "A Service Discovery Architecture based on Anycast in Pervasive Computing Environments", Computer Software and Applications Conference, 2007.
- [4] Dina Katabi, and John Wroclawski, "A Framework for Scalable Global IP-Anycast (GIA)", SIGCOMM'00, Stockholm, Sweden, 2000.
- [5] Wang Xiaonan Qian Huanyan, " An Anycast Communication Model on Solving Scalability in IPv6", Control, Automation, Robotics and Vision, 2006.
- [6] Hirokazu Miura and Miki Yamamoto, "Server Selection Policy in Active Anycast", IEICE Trans. Commun., Vol. E84.B, No. 10 October 2001.
- [7] H.B. Hashim, J.A. Manan, "An active anycast rtt-based server selection technique", Proceedings of 13th IEEE International Conference on Networks I (2005) 145–149.
- [8] 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.
- [9] 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.
- [10] Samrat Bhattacharjee, Mostafa H. Ammar, Ellen W. Zegura, Viren Shah, and Zongming Fei, "Application-Layer Anycasting," IEEE INFOCOM'97, Kebe, Japan, April 1997.
- [11] 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.
- [12] 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.
- [13] 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.
- [14] Zheng da Wu, Chris Noble, and D. Huang. "Optimal Video Distribution Using Anycasting Service", Information Technology papers, 1999.
- [15] 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.
- [16] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, R. Sparks, M. Handley, E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, June 2002.
- [17] A. Niemi, "Session Initiation Protocol (SIP) Extension for Event State Publication", RFC 3903, October 2004.
- [18] 3GPP Forum, "IP Multimedia Subsystem (IMS)", Stage 2, 3GPP TS 22 228 v8.11, 2009-12-14, 2009.