Proceedings of the International MultiConference of Engineers and Computer Scientists 2011 Vol I,
IMECS 2011, March 16 - 18, 2011, Hong Kong

Content Anycasting Applications for Future Internet

Othman Othman M.M., and Koji Okamura

***Abstract*—In this paper, we show our design and implementation of two applications built using our Contents Anycasting [14], the two applications are popular large file distribution and an enhanced P2P video streaming. Content Anycasting aims to providing more flexible and dynamic redirection of contents that is done with the help of OpenFlow's new potential abilities and thus giving new opportunities to the future internet that are currently not available. Both of the popular large file distribution system and the enhanced P2P video streaming applications tends to attract large number of user clients, especially if their content was a popular one; like some distributions of an operating system or a live video of a sport event. And thus, those applications can take advantage of the content anycasting by using it to reduce the load off the server or as an alternative method to construct the P2P multicast tree that has a low startup delay and has a less tendency to fail due to flash crowds.**

***Index Terms*—Future internet, OpenFlow, Content Delivery Networks, Content based networking, Anycast.**

I. Introduction

At the beginning of the Internet, it served the role of delivering contents using the simple client server model. But as time passes number of users grew fast, along with that also the user's needs became broader and more diverse, causing an increase of server loads thus giving place to new content delivery models to take place like the Peer to Peer. And again as time passes with the introduction of new technologies, services and with social changes that followed the wide adoption of the Internet user's needs became more and more demanding. As shown by studies in AKARI [8] that the traffic size increases by factor of 1.7 every year. Moreover the needs became more complicated, by having wide variety of contents available on the Internet ranging from simple web pages and emails to online video and large file transfer. This led to an increase in server loads, causing many difficulties for the current delivery models to cope with, along many other difficulties that rose due to the current Internet's architecture and design limitations. While on the other hand, (the clients side) clients are having stable internet connections with considerably high bandwidth specially with the introduction of the Fiber To The Building FTTB and the Fiber To The House FTTH, as an example in Japan the number of FTTH users is more than the number of DSL users as shown in [7] (see figure 1).

And due to the current Internet limitations researchers have embarked on studying the Future Internet. Where researchers

Manuscript received December 28, 2010; revised February 5, 2011.
O. Othman M.M. is with the Department of Advanced Information Technology, Graduate school of Information Science and Electrical Engineering, Kyushu University, 6-10-1 Hakozaki, Higashi-ku, Fukuoka 812-8581, Japan e-mail: omo_5@ale.csce.kyushu-u.ac.jp
K. Okamura is withthe Research Institute for Information Technology, Kyushu University. 6-10-1 Hakozaki, Higashi-ku, Fukuoka 812-8581, Japan. e-mail: oka@ec.kyushu-u.ac.jp

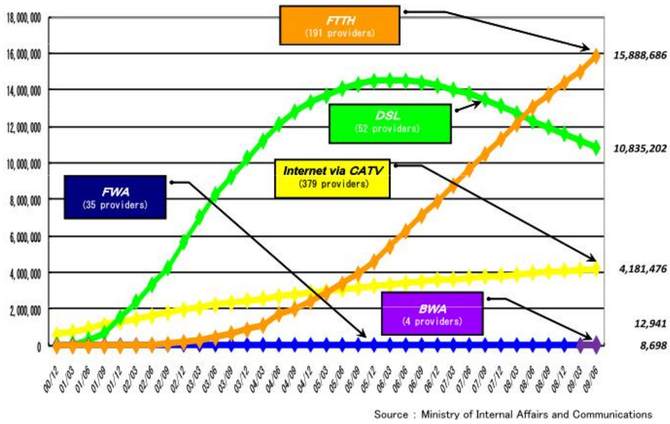

Fig. 1. Number of services subscribers in Japan from 2000 to 2009 as shown in [7].

from all over the world as in FIND [5] and GENI [6] in the United States, AKARI [8] in Japan, FIF [4] in Korea, and EIFFEL [3] in Europe, are studying challenges of the current Internet and are proposing ways to study and solve those challenges along with proposing new functionalities that were not possible on the current internet.

Among the future Internet research, content anycasting [14] was proposed aiming to provide a more flexible, dynamic way of redirection, and to tip off the imbalance between the heavily loaded server side and the demanding client side having a high bandwidth, and to improve the content server side by increasing the number of clients that the server can serve. This is done with the help of OpenFlow [10] project that holds a great potential for developing systems that have more freedom in acting. This makes content anycasting a good candidate to be used in cases where a large number of clients demand getting some large contents form the content server, such as a popular large file in case of file download, or a popular video streaming like in case of some sports events.

In case of popular large file download many solutions were used to solve the server overloading problem like increasing the bandwidth link, having redundant servers with load balancers or using Content Delivery Network which was introduced to overcome this problem and to provide high availability of the contents. And to enable the CDN many technologies were introduced like the Anycast [11] and Peer to Peer networks as in [9]. But those technologies suffer from some drawbacks like; that anycast acts on fixed servers making it a static way of providing contents, while peer to peer networks impose extra communications forced by their protocols, overlay nature, and their lack of knowledge about the network topology. While content anycasting can increase the number of clients downloading large files from servers by having clients acting as servers to other clients,

ISBN: 978-988-18210-3-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

and by distributing the load of arranging this upon the content server, the network and the clients the thing that gives it an advantage over other CDN solutions.

And in case of popular video streaming, many technologies are used like IP multicasting [15] and P2P multicasting [16] to provide live video service. But also those technologies suffers from some limitations related to management, and the overheads due to extra communication and maintenance of the P2P multicast. Where content anycasting can reduce the over heads of constructing the P2P multicast tree also by distributing this load among the content server, the network and the clients.

The remainder of this paper is organized as follows. We first explain about Content Anycasting redirection system showing its components and its content requesting method in section number 2. We then show how Content Anycasting can be utilized to implement a popular large file distribution system in section 3. And also in section 4 we show how Content Anycasting can be used to make an enhanced P2P video streaming system. Finally we conclude our work in section 5.

II. The Content Anycasting Redirection System

A. *Content Anycasting System overview*

Content Anycasting system consists of content server that provides contents, anycast manager, OpenFlow routers or switches and user clients (see Figure 2).

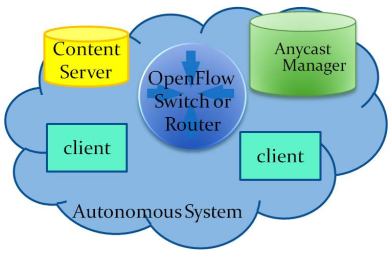

Fig. 2. System overview.

1) Content Server: The content server is almost a regular content server that provides contents like large files or videos, where each content has a unique identifier. The content server is responsible for sending redirection request in order to manage redirections. This redirection request includes the contents ids, the list of user clients that are currently getting this content and the number of redirections for each one of those user clients (See Figure 3).

Redirection request header		
Content id: 12345	192.168.10.1	1
	192.169.1.1	2
	192.168.10.4	3
⋮		
Content id: 33333	192.200.10.1	2
	192.200.1.1	1
	192.210.10.4	1

Fig. 3. Redirection Request example.

2) Anycast Manager: The anycast manager is responsible for creating redirections and managing those redirections. The redirection system might require more than one anycast manager for example one for each autonomous system, which means one anycast manager for each network of independent organization that implements BGP (Border Gateway Protocol) (see Figure 4.).

The first function that the anycast manager provides is managing redirection requests which means deciding which is the best anycast manager to handle the redirection request and forwarding the redirection request to it.

The second function it provides is creating the redirection by sending instructions to the OpenFlow routers or switches to redirect the content requests that are originally sent to the content server to other user clients that are near to the requesting client.

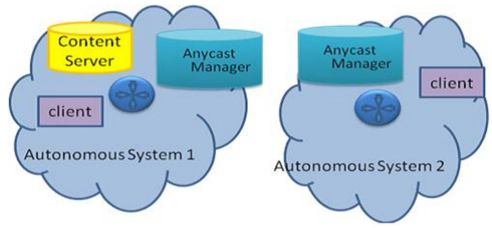

Fig. 4. An example of a redirection system with two autonomous systems and tow anycast managers.

3) OpenFlow routers or switches: The role of the OpenFlow routers or switches is to perform the redirections that were created by the anycast manager. Where the redirections are OpenFlow's flow table entries, where each incoming packet is checked if its destination IP address and its content id matches the content server IP address and the content id in one of the flow table entries. And if a match occurs the OpenFlow router or switch uses one of the Openflow functionalities to change the destination IP address in the first packet of the content request from the IP address of the content server to the IP address of the nearest client.

4) User Clients: User clients in our system have a modified behavior; that they will act as servers for the contents they are currently getting from the content server.

B. *Overview of Requesting Content Using Content Anycasting*

In order to make use of our system a special method for getting the content is required. This new method is divided into three phases; the first one is related to finding the content id, while the second is related to sending the content request, and the third one aims to establish the connection and downloading the content.

1) Phase 1: Getting Content ID: On this phase the client finds the desired content that has the content id as part of the URL for that content, the client side software (web browser for example) will store the content id in order to use it in the second phase.

2) Phase 2: Content Request: This phase takes place after the client gets the content id and before starting to download the content. In this phase a 3 way handshake (see Figure 5) using the content id is used before starting a TCP session to get the content. And in order to enable this, a new Probe protocol is introduced (see Figure 6).

ISBN: 978-988-18210-3-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

According to the Probe protocol, the client initiates this probe handshake by sending a START packet this packet will be redirected by the OpenFlow switches or routers to the nearest client that is currently getting the same content from the content server. This is done by having the OpenFlow router or switch changing the destination IP address to be that of the current client instead of the content servers IP address. And in response to this initiation the other client that received the packet will respond with START/ACK packet. Finally the new client will respond by acknowledging the received acknowledgement (ACK/ACK).

3) Phase 3: Getting the Content: At this time the new client realizes that it has been redirected to another client and initiates a TCP session using the IP address of the other client it received in the second phase directly without relying on the redirection system. Figure 9 shows the Probe's 3 way handshake and the TCP session in case of a match (see Figure 5).

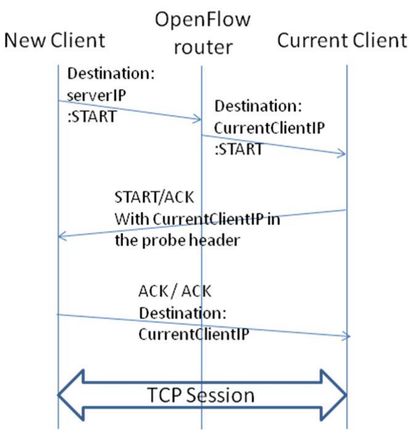

Fig. 5. Content request in case of redirection.

Bit offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	Content id																															
32	Length																Checksum															
64	Source Port																Destination Port															
96	Command																															
128	IP address																															
160	Xid																															

Fig. 6. The Probe header (modified UDP), where the highlighting shows the modified fields.

III. Popular Large File Distribution

In this section we will show how content anycasting can be used to increase the number of clients getting the popular large file. This is done through having the content server sending redirection request that has the IP address of any client that is downloading the popular file form the content server.

A. Example of using Content Anycasting in Popular Large File Distribution System

First, Figure 7 show the content server, anycast manager, client A which is currently downloading the desired content,

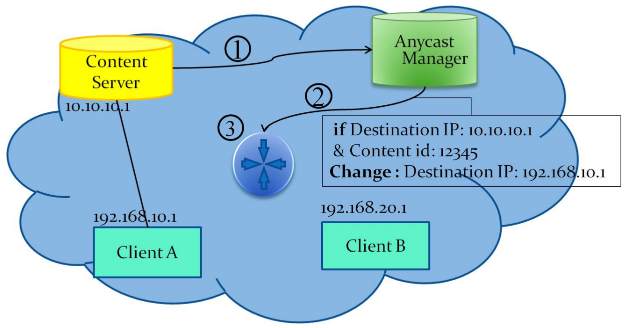

Fig. 7. System initiation.

client B which is a new client that is intersected in downloading the same content and an OpenFlow router shown as the circle in the figure.

In step number 1 the content server sends a redirection request to the anycast manager. In step number 2 the anycast manager analyzes the request and prepares the required redirections and sends them to the OpenFlow router. And in step 3 the OpenFlow router saves the redirection, which is an OpenFlow table entry to its flow table.

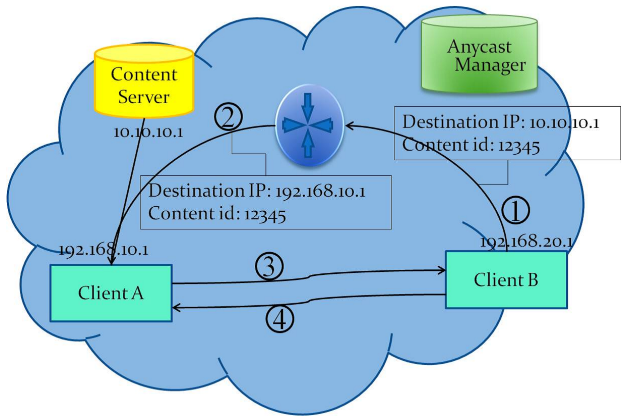

Fig. 8. Content requesting.

Next, Figure 8 shows phase 2 of the content request.

In Figure 8 step 1 client B sends the Probe protocol's START handshake that contains the content id. In step 2 the OpenFlow router redirects the packet sent by client B to go to client A instead of the content server after matching it to the redirection it received from the anycast manager by changing the destination IP to be that of client A and sending it on the right port to be delivered to client A. But in case of not finding a matching redirection the router will process the packet in the conventional way by sending the packet to the content server. Then in step 3 client A will acknowledge the START of step 1. And finally, client B will confirm that it received the acknowledgement.

Finally, Figure 9 shows the third phase of requesting the content; where client B and client A completes the handshake of the Probe protocol. Then client A will start sending the content to client B directly without using the redirection on the router using a direct TCP session.

ISBN: 978-988-18210-3-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

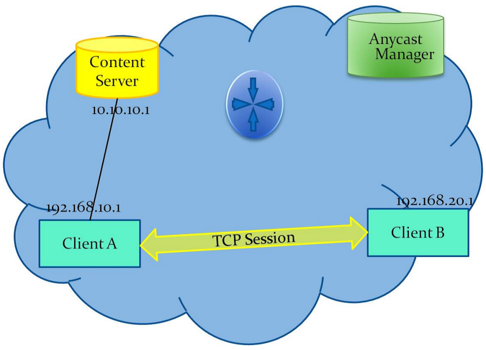

Fig. 9. Getting the content.

B. *Evaluation of using Content Anycasting in Popular Large File Distribution System*

And in order to judge the effectiveness and advantages of using Content Anycasting in building popular large file distribution system, we designed and implemented a simple simulator as a preliminary method of evaluation. This simulator was built using Java programming language. It is designed to compare the content server load in case of using Anycasting as a method of constructing the Large file distribution system with the case of using Content Anycasting for that purpose. In more detail this simulator runs using a topology of 5 areas resembling 5 autonomous systems.

Then running simulation using anycasting where one replica content server is located in each one of the areas. This resulted in 20% of the load being distributed on the 5 replica servers (see figure 10).

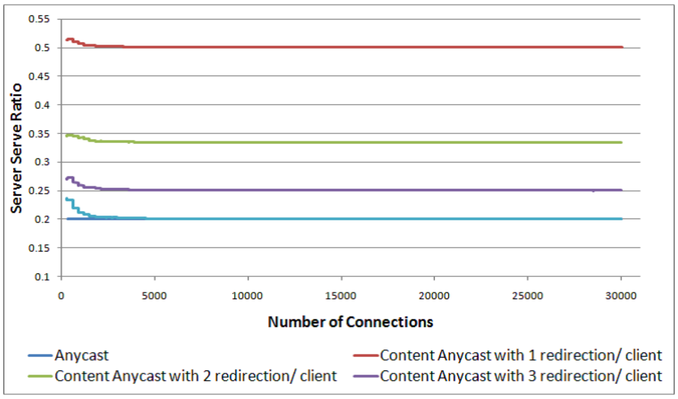

Fig. 10. The content server load in case of using current anycast, and content any casting with 4 cases; where each client is capable of serving 1,2,3,4 other clients.

And then we used the same topology to build a file distribution system using content anycasting with only one content server. Different cases were studied, where each client is capable of serving 1or 2 or 3 or 4 other clients (see figure 10).

This resulted in having the content server load being 50%, 33%, 25% and 20% respectively for the previous mentioned cases (each client is capable of serving 1, 2, 3, 4 other clients).

This shows that using content anycasting is capable of achieving the same load on one server rather than 5 in case of anycasting if the all of the clients where capable of serving 4 other clients.

IV. Enhanced P2P Video Streaming

In this section we will show how content anycasting can be used to enhance the P2P Video streaming by decreasing the joining delay of new clients. This is done on the assumptions that clients have a stable internet connection with a high bandwidth (for example as a requirement by the IPTV provider) and also that they are less likely to leave a video stream after joining it (specially in case of popular sport events).

This is done by having the clients reporting that they have joined the video streaming to the server, so that the server will know about all of the client that are currently joining the video streaming even the clients that have joined after their probe was redirected and are getting the stream through other clients. And thus the server can use those clients in order to make them server others after checking the desired depth of the multicast tree and sending the appropriate ones in a redirection request to the anycast manager. This will be illustrated in the following example.

This has the advantage of bringing the multicast tree construction one step ahead. For example when a new client wants to join the video stream it does not have to contact the server to find which client to get the connection from, instead the client request will be directly connected to the client that he will be connecting to. And after the clients joins the multicast tree this way he will inform the server of this. This means that the new client will first join the multicast and then contacts the server. While in case of other P2P video streaming, the new client have to contact the server before joining the multicast tree.

A. *Example of using Content Anycasting to make an enhanced P2P video streaming*

First, Figure 11 show the content server, anycast manager, client A which is currently downloading the desired content, client B which is a new client that is intersected in downloading the same content and an OpenFlow router shown as the circle in the figure.

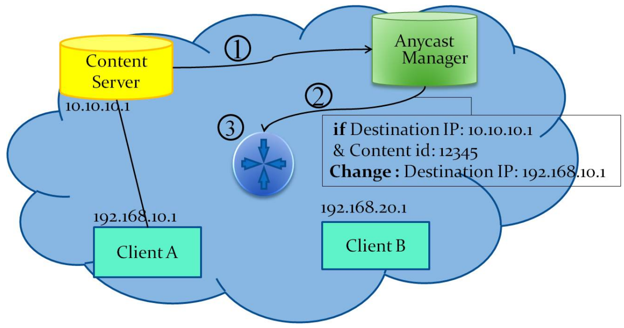

Fig. 11. System initiation.

In step number 1 the content server sends a redirection request to the anycast manager. In step number 2 the anycast manager analyzes the request and prepares the required redirections and sends them to the OpenFlow router. And in step 3 the OpenFlow router saves the redirection, which is an OpenFlow table entry to its flow table.

Next, Figure 12 shows phase 2 of the content request. In Figure 12 step 1 client B sends the Probe protocol's START handshake that contains the content id. In step 2

ISBN: 978-988-18210-3-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

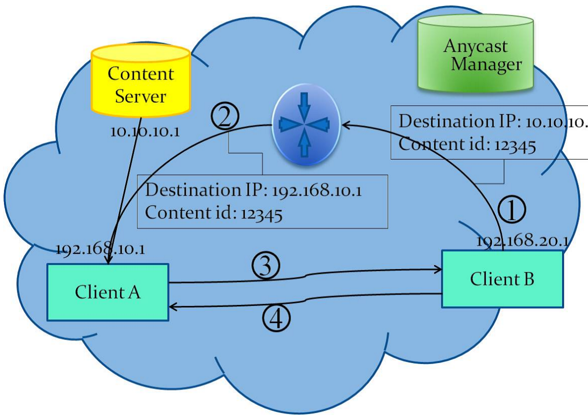

Fig. 12. Content requesting.

the OpenFlow router redirects the packet sent by client B to go to client A instead of the content server after matching it to the redirection it received from the anycast manager by changing the destination IP to be that of client A and sending it on the right port to be delivered to client A. But in case of not finding a matching redirection the router will process the packet in the conventional way by sending the packet to the content server. Then in step 3 client A will acknowledge the START of step 1. And finally, client B will confirm that it received the acknowledgement.

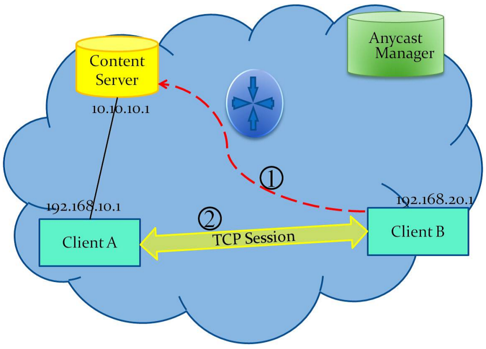

Fig. 13. Getting the content.

Finally, Figure 13 shows the third phase of requesting the content; where client B and client A completes the handshake of the Probe protocol. Then client B will send to report to the server that it will join the video stream through client A as shown in step 1. And then client B will start receiving the content from client A directly without using the redirection on the router using a direct TCP session in step 2.

B. Evaluation of using Content Anycasting to make an enhanced P2P video streaming

In order to judge the effectiveness and advantages of using Content Anycasting in making an enhanced P2P video streaming system, we designing and implemented a simple simulator as a preliminary method of evaluation. This simulator was built using Java programming language. It is designed to compare the content server load, and the overhead of joining the multicast tree in case of using Anycasting as a method of constructing the enhanced P2P video streaming system with the case of using Content Anycasting for that purpose. In more detail this simulator runs using a topology of 5 areas resembling 5 autonomous systems.

Which we will use to compare the overhead of joining the multicast tree for the regular P2P video streaming with that of the enhanced P2P video streaming using content anycasting. And also we will use it to compare the server load in case of flash crowd (large number of clients' requests content in a short period of time).

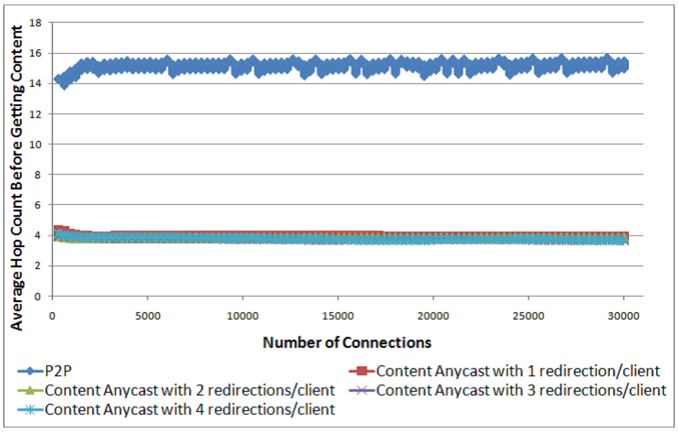

Fig. 14. The average distance in hops that the client request have to travel prior to get the content in case of using central management P2P, and content any casting with 4 cases; where each client is capable of serving 1,2,3,4 other clients.

In order to assess the overhead of joining the multicast tree we counted the average total number of hops that the clients' request and their reply packets had to travel across the network before the client is able to get the content. Figure 14 shows the results that we obtained by comparing the case of regular methods of constructing a P2P network (using a central management entity for the P2P network) with the case of using content casting (again with four cases where each client serves 1, 2, 3, and 4 other clients). This figure shows that the average hop count needed before getting content according to the topology we used was about 15 hops in case of the regular method of constructing the P2P network, while the average was around 5 hops in case of using Content Anycasting. Which means; that using content anycasting to construct a P2P network would improve the overhead of joining the P2P network or the P2P multicast tree.

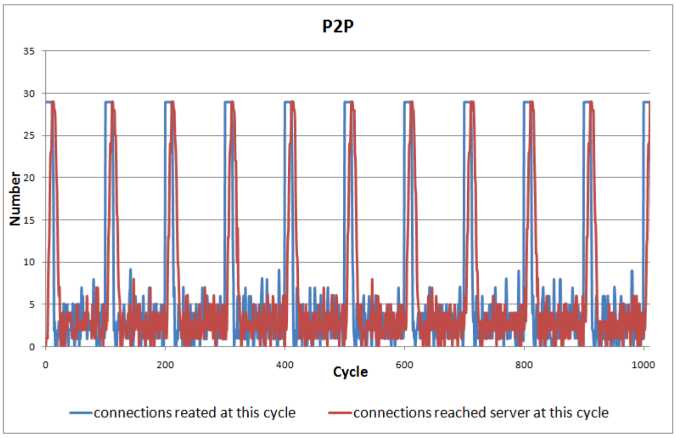

Fig. 15. Connections created by clients and connections served by the server in each cycle of the simulation in case of regular P2P with central management.

ISBN: 978-988-18210-3-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

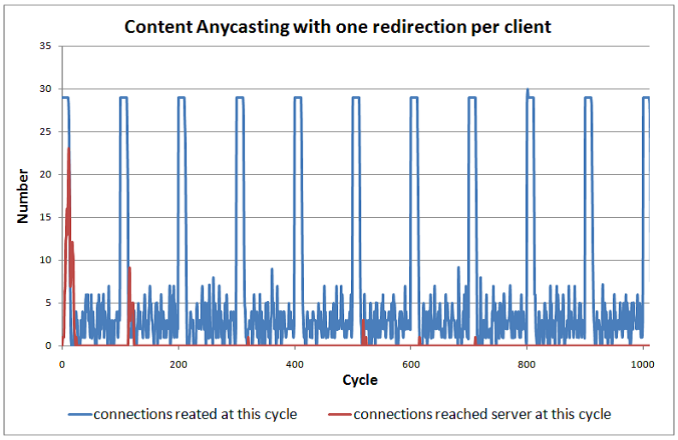

Fig. 16. Connections created by clients and connections served by the server in each cycle of the simulation in case of content anycasting.

And in order to assess the server load in case of flash crowd, we set our simulator to periodically bust a large number of connections and then used it to count the number of connections that reached the server (or the P2P management entity) at each simulation cycle. Next we ran this simulation two times; first, using the regular method of constructing P2P network (see figure 15), second using content anycasting to construct the P2P network (see figure 16).

Those figures show that in case of using the regular methods to construct P2P network; all of the requests sent by the clients have to reach the server (or the central management entity) to be processed and replied back to the clients. This is due to the way they operate; in which the client have to consult the server or the management entity so that thy reply with the information needed to contact other peers. While in case of using content anycasting the server does not have to process all of the requests because the new clients requests will be redirected by the network towards the other peers, and so there is no need to consult the server (or the management entity) prior to getting the content,

V. Conclusion

Providing new opportunities to the future internet is very important. This requires the creation and adoption of new technologies. In this paper we described two applications that will take benefit of the content anycasting, which aims make use of the content anycasting flexible and dynamic redirection of contents, to improve the availability of popular large files, to increase the number of clients downloading them, and to reduce the delays imposed by P2P multicasting in case of the enhanced P2P video streaming. Through this paper we showed how to provide an enhanced popular large file distribution and an enhanced P2P video streaming that makes use of content anycasting which relies on both the destination address and the content id to take the redirection decision. Our preliminary simulation shows that using content anycasting with only a single content server; load can be reduced to 50%, 33%, 25% and 20% respectively for the previous mentioned cases (each client is capable of serving 1, 2, 3, 4 other clients) compared to 20% load for the case of having 5 replica content servers in case of anycast. More over content anycasting showed that it is capable of reducing the average hop count needed prior getting the content, because it makes use of the network to redirect packets and so helps to redirect requests to a peer within the same network rather than randomly choosing peers regardless of their location. Also, content anycasting showed it can be used to take the process of querying about peers one step ahead by pre installing redirections on the network rather than having the server to reply to all of the queries. And thus it improves the availability of the server.

References

[1] BitTorrent Protocol Specification v1.0. http://wiki.theory.org/BitTorrentSpecification
[2] Cerf,V., Dalal, Y., Sunshine, C. “SPECIFICATION OF INTERNET TRANSMISSION CONTROL PROGRAM.” December 1974.
[3] EIFFEL, http://www.future-internet.eu .
[4] FIF, http://fif.kr
[5] FIND, http://www.nets-find.net
[6] GENI, http://www.geni.net
[7] FUJINO, M., “National Broadband Policies: 1999–2009, Japan”. October 2009. http://www.soumu.go.jp/main_sosiki/joho_tsusin/eng/presentation /pdf/091019_1.pdf
[8] Hirabaru, M., Inoue, M., Harai, H., et. al. October 2008. “New Generation Network Architecture AKARI Conceptual Design (ver1.1)”. AKARI Arechitecture Design Project. http://akari-project.nict.go.jp/eng/concept-design/AKARI_fulltext_e_translated_version_1_1.pdf
[9] Karagiannis, T., Rodriguez, P., and Papagiannaki, K. 2005. “Should internet service providers fear peer-assisted content distribution?”. In Proceedings of the 5th ACM SIGCOMM Conference on internet Measurement (Berkeley, CA, October 19 - 21, 2005). Internet Measurement Conference. USENIX Association, Berkeley, CA, 6-6.
[10] Nick McKeown, Tom Anderson, Hari Balakrishnan, Guru Parulkar, Larry Peterson, Jennifer Rexford, Scott Shenker, and Jonathan Turner. 2008. “OpenFlow: enabling innovation in campus networks”. SIGCOMM Comput. Commun. Rev. 38, 2 (March 2008), 69-74. DOI=10.1145/1355734.1355746 http://doi.acm.org/10.1145/1355734.1355746
[11] Partridge, C., Mendez, T., and Milliken, W., RFC 1546 - “Host Anycasting Service”. November 1993.
[12] Teemu Koponen, Mohit Chawla, Byung-Gon Chun, Andrey Ermolinskiy, Kye Hyun Kim, Scott Shenker, and Ion Stoica. 2007. “A data-oriented (and beyond) network architecture”. SIGCOMM Comput. Commun. Rev. 37, 4 (August 2007), 181-192. DOI=10.1145/1282427.1282402 http://doi.acm.org/10.1145/1282427.1282402
[13] Van Jacobson, Diana K. Smetters, James D. Thornton, Michael F. Plass, Nicholas H. Briggs, and Rebecca L. Braynard. 2009. “Networking named content”. In Proceedings of the 5th international conference on Emerging networking experiments and technologies (CoNEXT ’09). ACM, New York, NY, USA, 1-12. DOI=10.1145/1658939.1658941 http://doi.acm.org/10.1145/1658939.1658941
[14] Othman Othman M.M., Koji Okamura 2010. “Improvement of Content Server with Contents Anycasting Using OpenFlow”. In Proceedings of the 30 APAN of Network Research Workshop.
[15] R. Perlman, “Models for IP Multicast”, in Proceedings of IEEE International Conference on Networks, 2004.
[16] Xiaofei Liao, Hai Jin, Yunhao Liu, Lionel M. Ni, and Dafu Deng. 2006. “AnySee: Peer-to-Peer Live Streaming”. in Proceedings of INFOCOM 2006. 25th IEEE International Conference on Computer Communications

ISBN: 978-988-18210-3-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)