ACEEE Int. J. on Information Technology, Vol. 3, No. 2, June 2013

Fast Distribution of Replicated Content to Multi-Homed Clients

Mohammad Malli
Arab Open University, Beirut, Lebanon
Email: mmalli@aou.edu.lb

Abstract—**Clients can potentially have access to more than one communication network nowadays due to the availability of a wide variety of access technologies. On the other hand, service replication has become a trivial approach in overlay networks to provide a high availability of data and better QoS. In this paper, we consider such a multi-homed client seeking a replicated service in overlay network (e.g., CDN, peer-to-peer). Our aim is to improve the content distribution by proposing a new model for being applied at the application-level and in a fully distributed way. Basically, our model proposes to determine the best mirror server that could be reached through each client's network interface based on application utility function. Then, it consists of downloading the requested content from the determined best servers simultaneously through their associated interfaces. Each best server should deliver a specific estimated range of bytes (i.e., content chunk) to an independent TCP socket opened at the client side for being finally aggregated at the application-level. Our real experiments show that our model is able to considerably improve the QoS (e.g., content transfer time) perceived by the client comparing to the traditional content distribution techniques.**

Index Terms—**content distribution, service replication, multi-homing.**

I. Introduction

Service replication is a scalable solution for the distribution of digital content over the Internet. The need for this replication is caused by the increasing number of Internet users and by the desire to improve the QoS. Also, it is important for achieving a high availability of data. Many overlay networks are proposed and installed to realize this replication: (i) Content Distributed Networks (CDN), where client requests are forwarded by request redirectors, and where the contents are stored in mirror servers geographically distributed over the Internet. Many companies, like Akamai [1], provide CDNs to content providers. (ii) Peer-to-peer networks (e.g., bitTorrent [2], where peers behave as clients and servers. On the other hand, one can profit from multi-homed clients to achieve bandwidth aggregation by striping data across the multiple network interfaces of the clients.

In this paper, we address the problem of improving the transfer time perceived by multi-homed clients when requesting digital content replicated in the mirror servers of one CDN network (resp. peer-to-peer network) or in multiple ones (i.e., content multi-homing [3], [4], [5]). In the following discussion, we consider a server as being either a server among a set of replicated servers in a CDN or a peer in a peer-to-peer network that hold the requested content. The best server is the one which is able to provide the requested service to the client with a better QoS than all other servers. Also, we mean by client a standard client in the client/server paradigm, or a peer that requests content in a peer-to-peer network; these terms are used in the paper interchangeably. Clearly, the best server varies from one client to another based on many parameters as the performance on the path connecting the client to the server through each network interface.

For enhancing the content distribution, many solutions [6], [7], [8], [9] rely on particular network infrastructure nodes (e.g., load-aware Anycast router, route controller, peer coordinator, etc.) taking into account network-specific constraints (e.g., traffic engineering constraints) perceived by the ISP or the overlay network operator to solve the problem as a global optimization problem. While such approaches could be of great benefit for traffic engineering purposes, end-systems solutions is able to provide better enhancement to the performance perceived by the clients. Besides, in some scenarios, end-systems solutions are able to achieve a better traffic engineering outcome than the ISPs can by themselves as shown in [10]. Moreover, one can avoid the deployment limitations (e.g., network overhead) of the existing solutions by solving the problem at the end-user level in a fully distributed way. On the other hand, although the concept of multipath-capable end systems is interesting to be applied at the transport level [10], [11], [12], [13], [14], there is no protocol that simultaneously uses multiple paths has ever been standardized let alone widely deployed to replace the most widely used existing protocol TCP.

Therefore, we propose a new model to be applied at the application-level and in a fully distributed way for improving the QoS perceived by multi-homed end-users. It consists in client downloading a replicated content from a certain set of best mirror servers simultaneously through his/her different network connections. Firstly, it proposes to determine the best server that could be communicated through each network interface based on application utility function. Then, it consists of downloading the requested content from the determined best servers simultaneously but with different estimated amounts.

This must be achieved by opening a TCP connection with each best server through its associated network interface to download a specific estimated range of bytes.

The size of this range depends on the weight assigned to the best server; function of the performance status on its path to the client's associated network interface. Thus, our model is able to improve the QoS perceived by the client

DOI: 01.IJIT.3.2.1

through achieving bandwidth aggregation by striping data across multiple TCP sockets (i.e., one per network interface) that download the content chunks from their associated best servers simultaneously.

This paper is organized as follows. The next section elaborates the problem of best server selection in replicated service environment. Then, we present, in Section III, a new model for distributing replicated content to multi-homed client. The experiments that show the performance enhancement provided by this model are presented in Section IV. Finally, the conclusion is presented in Section V.

II. Service Replication

Service replication is a scalable solution for the distribution of digital content over the Internet. The need for this replication is caused by the increasing number of Internet users and by the desire to improve the QoS. Also, it is important for achieving a high availability of the service. Many overlay networks (e.g., Content Distributed Networks (CDN), and peer-to-peer networks) are proposed and installed to realize this replication. The first stage of our approach consists of determining the best server to be communicated through each client's network interface.

Many policies have been studied in the literature for best server selection. The mostly used approaches can be classified to the following three categories:

- Using the DNS (Domain Name System) to get the IP address of the best server. This widely used technique is simple: the DNS servers distribute the IP addresses of multiple servers associated to a unique name with a round robin algorithm. It is clear that this solution is not designed to improve the QoS since it does not consider any static or dynamic performance limitations. It only ensures basic load balancing.
- Offering the client a list of servers and let him choose manually the best server to contact. The client choice in this case is based on his own criteria, for example the geographical proximity.
- Choosing the closest server in terms of delay. Inferring the delay closeness between client and servers can be done using one of the scalable approaches presented in the literature [15], [16], [17], [18], [19], [20], [21]. Most of these solutions are based on the network embedding. Such approaches are based either on network coordinates or on distance matrix factorization. Also, the closeness can be determined by identifying the bin of the client and each server (see [22]). This can be done by measuring their RTT (Round-Trip Time) to a set of landmark points. By knowing the bins of the client and servers, the DNS server can classify the servers (from the best one to the worst one) based on the distance between their bins and the client's one.

Thus, most of the existing solutions for best server selection are based on simple metrics such as the delay, and the geographical locations which are uncorrelated with other network characteristics (e.g., available bandwidth, loss rate) as perceived in the literature [23]. Hence, these metrics are not enough to characterize the proximity given the heterogeneity of the Internet in terms of path characteristics and access link speed, and the diversity of application requirements.

We have realized, in [24], that the proximity must be characterized in a CHESS space where it is determined at the application level taking into consideration the network metrics that decide on the application performance. Therefore, we have proposed to do that using a utility function that models the quality perceived by peers at the application level. In this framework, a peer is closer than another one to some third peer if it provides a better utility function, whatever the position of each peer in the geographical and delay spaces.

For example, take the case where the service consists of clients downloading digital content from a set of replicated servers using the TCP protocol and where the QoS provided to clients is maximized if the transfer time is minimized. In this case, choosing the best server amounts to downloading the file from the server that is able to provide the minimum transfer time. This improves the QoS provided to clients and avoids network and server congestion by distributing the load over servers and network paths that are less loaded than others.

While the characterization of the proximity in CHESS [24] has a good impact on application performance, it is a challenging task due to the two following major requirements. First, it requires the identification of the appropriate utility function for each application in a first stage. To solve this problem, many interesting models have been proposed in the literature (e.g., transfer time prediction [25], speech quality prediction [26], [27]). The second challenging task is the measurement of the different network parameters that impact the utility function. This is difficult to achieve in large scale networks where the number of peers can be huge. In such case, the cost of the direct probing among peers may outweigh the profit of the characterized proximity. Hence, the estimation of the network parameters, impacting the utility function, must be achieved in an easy and scalable way. In other terms, this should be achieved with a small measurement overhead and a limited cooperation among nodes. Particularly, the determination of the network parameters, on the paths joining a large number of peers, must be achieved in a way that avoids the direct probing among them as have been proposed in the literature [15], [16], [17], [18], [19], [20], [21], [24], [28].

III. Enhanced Model For Distributing Replicated Content In Multi-Homing Environment

A. Proximity Model

The major contribution that we present in this paper is an extended model of the previously proposed one CHESS [24] which has been very briefly described in the previous section. In the new model, we take advantage of the presence of multi-homing environment where multiple network connections held at the peer side to improve the perceived performance when downloading content from a set of mirror servers. In this setting, we propose to construct one CHESS space per

DOI: 01.IJIT.3.2.1

ACEEE Int. J. on Information Technology, Vol. 3, No. 2, June 2013

network connection. Thus, one can consider the new model as multi-dimensional CHESS where peers are ranked in each CHESS space based on the proximity characterized in this space; we recall that the proximity characterization is based on the application utility function.

Then, a peer could position itself in an overlay network or decide on the size of the content portions to download from the best peer of each CHESS space based on the proximity characterization in such overlay networks; our focus in this paper is on content distribution and not on overlay construction. Particularly, downloading content from multiple best servers (i.e., pooling capacity over the space) through the different network connections simultaneously (i.e., pooling capacity over the time) could be achieved obviously in shorter time delay than that achieved by the trivial point-to-point content delivery techniques. Besides, this can provide better availability of content and resiliency of service.

To the best of our knowledge, there is no model that pools capacity over the space and the time as efficient as the one proposed in this paper although the concept of resource pooling has been widely elaborated in the literature. Modern approaches rely on end systems for managing the network traffic patterns and enhancing the content distribution. In this scope, many solutions [6], [7], [8], [9] rely on particular network infrastructure nodes (e.g., load-aware Anycast router, route controller, peer coordinator, etc.) taking into account network-specific constraints (e.g., traffic engineering constraints) perceived by the ISP or the overlay network operator to solve the problem as a global optimization problem.

While such approaches could be of great benefit for traffic engineering purposes, end-systems solutions is able to provide better performance enhancement perceived by the clients. Besides, in some scenarios, end-systems solutions are able to achieve a better traffic engineering outcome than the ISPs can by themselves as shown in [10]. Moreover, one can avoid the deployment limitations (e.g., network overhead) of the existing solutions by solving the problem at the end-user level in a fully distributed way. On the other hand, although the concept of multipath-capable end systems is interesting to be applied at the transport level [10], [11], [12], [13], [14], there is no protocol that simultaneously uses multiple paths has ever been standardized let alone widely deployed to replace the most widely used existing protocol TCP.

Therefore, we take advantage of these perceptions to propose a new model for improving the distribution of replicated content to multi-homed peer. Basically, it proposes to determine the best server that could be communicated through each network interface based on application utility function (i.e., the closest peer in each CHESS space). Then, it consists of downloading the requested content from the different best servers simultaneously but with different estimated amounts.

This must be achieved by opening a TCP connection with each best server through its associated network interface to download a specific estimated range of bytes that could be considered as a chunk of the requested content. The size of such chunk depends on the weight assigned to the best server; function of the performance status on its path with the client's associated network interface. Thus, our model is able to improve the QoS perceived by the client through achieving bandwidth aggregation by striping data across multiple TCP sockets (i.e., one per network interface) that download the content chunks from their associated best servers simultaneously.

More formally, suppose that the network contains n peers $p=\{p_1,p_2,...,p_n\}$ where each peer could play the role of a client seeking a content or a server holding the requested content. Obviously, the content is replicated in multiple peers (resp. mirror servers). The utility function (e.g., delay, available bandwidth, predicted download time) on the paths joining peers p_i and p_j $(i,j=\{1...n\})$ on top of network connection c $(c=\{1...k\})$ is represented by an $n \times n$ matrix U^c, where U_{ij}^c is the estimated utility function from p_i to p_j through the network connection c. The fact that peers could have different number of network connections and thus different values of k does not affect the functionality of our model since we are presenting a distributed algorithm to be executed at each peer independently. In case that every peer has one network connection, the system converges to one CHESS space where the content must be transferred to the client's unique network interface from the best server selected as described in Section II.

Thus, for every multi-homed peer p_i,

- the rest of peers p_j are ranked through every network connection c based on the estimations U_{ij}^c. We assume in this model that the larger the utility function value, the better the quality of service (e.g., available bandwidth on the network path connecting peers) and the closer the peers to each other in this space. Obviously, in case where the utility function is in contrast significant for small values (e.g., delay, predicted download time), peers must be ranked according to the increasing order of U_{ij}^c.
- its closest peer in the CHESS space c (i.e., best peer reachable through network connection c) is the peer P_i^c that satisfies $MAX_{j=\{1..n\}} U_{ij}^c$ (resp. $MIN_{j=\{1..n\}} U_{ij}^c$ in case the utility function is significant for small values). The best peer P_i^c must be different than the ones determined through the other network interfaces even if it is the closest peer to p_i in the different CHESS spaces. Thus, if the closest peer in the CHESS space c is the same one selected as the best peer through another network connection, then P_i^c must be selected as the next closer peer (based on the previously presented ranking) that is not yet selected as the best peer through another network connection.

Hence, each best server can upload only one content chunk to a peer through one of its network connection. In this way, we are able to pool the capacity over the space by relying on a good number of best servers (i.e., equal to the number of client's network connections) instead of uploading the chunks from a fewer number of best servers (i.e., smaller than the number of client's

DOI: 01.IJIT.3.2.1

network connections). Besides, the content must be delivered from the selected best servers in a concurrent way to pool the capacity over the time as well. Moreover, for more efficient pooling of the servers' capacity, the size of a content chunk to be downloaded from each best server must be proportional to the performance on its end-to-end path with the associated client's network connection (as will be evaluated in this model).

- its best CHESS space is the space C that satisfies $max_{c=\{1..k\}} U_{i(Pi^c)}$. Then, the closest peer to p_i in the best CHESS space C is denoted by P_i^C. Thus, P_i^C is closer to p_i than the other best peers of the rest CHESS spaces since the estimated utility function between peer p_i and P_i^C has the greatest value.

Our algorithm can be useful for improving the overlay construction and content distribution. For overlay construction, a peer could infer its closest peer in each CHESS space to better position itself in an overlay network; this issue will be explored and examined in a future work. As for content delivery purpose which is our focus in this paper, it consists of determining the correspondent proportion to download DP_i^c by peer p_i through its network connection c from its associated best server c. Such proportion depends on the relative value of the performance on the path connecting p_i to P_i^c through the network connection c with respect to those on its path with the other best servers reachable through the remaining network connections. Therefore, we propose to evaluate DP_i^c as the fraction between the weight w_i^c assigned to P_i^c and the overall weights assigned to all the selected best servers:

$$DP_i^c = \frac{w_i^c}{\sum_{c=1}^{k} w_i^c}, \quad \text{for } c = \{1, .., k\} \tag{1}$$

where,

$$\sum_{c=1}^{k} DP_i^c = 1, \tag{2}$$

and the weight w_i^c evaluates how good is the predicted quality between p_i and P_i^c comparing to the best case estimated between p_i and P_i^C. Therefore, one can evaluate it as the fraction of the utility function evaluated between p_i and the best peer in the CHESS space c (i.e., to be communicated

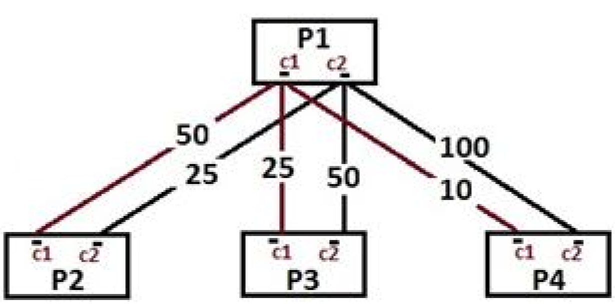

Figure 1. An example of multi-homed peer seeking a replicated content

through the network interface c) over the utility function between p_i and its closest peer in the best CHESS space:

$$w_i^c = \frac{U_{i(P_i^c)}^c}{U_{i(P_i^C)}^C}, \quad w_i^c \leq 1 \tag{3}$$

Thus, we propose that peer p_i opens a TCP connection with the best server selected in each CHESS space P_i^c ($c=\{1...k\}$) to download the content chunk having the following proportion from the whole content:

$$DP_i^c = \frac{\frac{U_{i(P_i^c)}^c}{U_{i(P_i^C)}^C}}{\sum_{c=1}^{k} \frac{U_{i(P_i^c)}^c}{U_{i(P_i^C)}^C}}, \quad \text{for } c = \{1, .., k\} \tag{4}$$

If these proportions do not divide the file size into finite ranges, the residual value is added to the range of bytes allocated to the closest peer in the best CHESS space P_i^C. Finally, the receiving application at the client side does the re-sequencing using a buffer having the size of the requested content to achieve reliable in-sequence data deliv.ery.

B. Case study

Take the example of Figure 1 where peer p_1 would like to download a content of *999B* replicated in peers $p_2, p_3,$ and p_4. In this scenario, we assume that there are two network connections c_1 and c_2 that could be used by peers to communicate with each other. Thus, there are two correspondent CHESS spaces. Besides, the utility function is assumed to be a simple metric which is the available bandwidth on the end-to-end network path for simplicity seeking. As shown in the figure, the available bandwidth values on the paths connecting p_1 to the other peers through network connection c_1 are:

- p_1 to p_2 is *50 Mbps*. So, $U_{12}^{c1} = 50$,
- p_1 to p_3 is *25 Mbps*. So, $U_{13}^{c1} = 25$,
- p_1 to p_4 is 10 Mbps. So, $U_{14}^{c1} = 10$.

For the second CHESS space which is reachable through the network connection c_2, the available bandwidth values are:

- p_1 to p_2 is *25 Mbps*. So, $U_{12}^{c2} = 25$,
- p_1 to p_3 is *50 Mbps*. So, $U_{13}^{c2} = 50$,
- p_1 to p_4 is 100 Mbps. So, $U_{14}^{c2} = 100$.

In this case, p_2 is the best peer for p_1 in CHESS space c_1 with $U_{12}^{c1} = 50$ and p_4 is its best peer in CHESS space c_2 with $U_{14}^{c2} = 100$. Thus, the best CHESS space for p_1 is c_2 (i.e., $C= c_2$).

Then, peer p_1 assigns the following weights to the best servers determined in the two CHESS spaces c_1 and c_2 respectively:

$$w_1^1 = \frac{U_{1(P_1^{c1})}^{c1}}{U_{1(P_1^C)}^C} = \frac{U_{12}^{c1}}{U_{14}^C} = 0.5, \tag{5}$$

and,

$$w_1^2 = \frac{U_{1(P_1^{c2})}^{c2}}{U_{1(P_1^C)}^C} = \frac{U_{14}^{c2}}{U_{14}^C} = 1. \tag{6}$$

Thus, peer p_1 should download from peer p_2 through the network interface of CHESS space c_1 a content chunk having the following proportion:

$$DP_1^1 = \frac{w_1^1}{\sum_{c=1}^{k} w_i^c} = \frac{0.5}{1.5} = \frac{1}{3}, \quad (7)$$

and subsequently the range of bytes *[1,333]* from the whole content to download.

Concurrently, peer p_1 should download from peer p_4 through the interface of CHESS space c_2 a content chunk having the following proportion:

$$DP_1^2 = \frac{w_1^2}{\sum_{c=1}^{k} w_i^c} = \frac{1}{1.5} = \frac{2}{3}, \quad (8)$$

and subsequently the range of bytes *[334,999]* from the whole content to download.

If the file size is *1000B* which could not be divided to a finite ranges of bytes in this case. Then, the residual extra byte from the division is allocated to the range allocated to the closest peer p_4 in the best CHESS space c_2. Then, the range of bytes to be downloaded from p_4 becomes *[334, 1000]* and the range of bytes to be downloaded from p_2 remains the same.

IV. Enhanced Content Delivery Perceived by The Application

For evaluating the improvement that can be achieved by applying the presented approach for content distribution, we have conducted real experiments having the following settings. We take *10* multi-homed c lients spread in the networks of four ISPs where each client is plugged simultaneously to two network connections which are DSL and WiMax. The DSL and WiMax connections have the same dedicated bandwidth of *512kbps* download speed and *128kbps* upload speed.

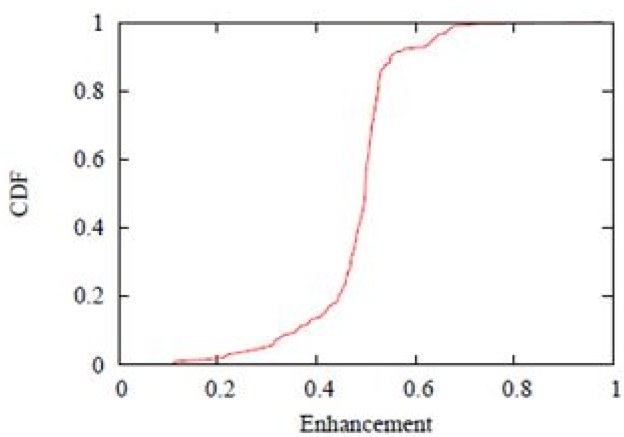

Figure 2. Enhancement of the content delivery time

Each trace consists in multi-homed client downloading a digital content of particular size from one of *20* selected CDN networks having worldwide distributed sets of mirrors. This leads to a total of *200* traces where in each trace we measure the following two content delivery times:

- T_m which is the latency measured when applying our new model. In this case, the content is delivered concurrently from the two best servers identified through the two network connections according to the multidimensional CHESS model presented in Section III.
- T_b which is the latency measured when downloading the content in point-to-point mode from the best server identified through the DSL connection according to the CHESS model published in [24] and briefly presented in Section II.

The two ways of content delivery applied in each trace have been achieved in a sequential manner. The whole traces have taken place in different dates and times during the month of July 2012. We assume in these experiments that the utility function, for best server selection, is the end-to-end available bandwidth on the network path connecting the client to the server. One can rely on the predicted transfer time metric [25] as a more optimized tool in this scope. However, this choice of utility function does not affect the observations of our results since our aim is to compare our new model of content delivery with the classic way of point-to-point content delivery despite the way of determining the best servers.

Hence, to measure the available bandwidth, we have applied the probe rate model [29], [30]. This is achieved by sending a stream of packets from the client to the server at a rate greater than *512kbps* since the end-to-end available bandwidth is surely smaller (or equal) than this value which is the maximum download rate of the client per network interface. Then, the end-to-end available bandwidth is calculated as the rate of the receiving stream's echoes.

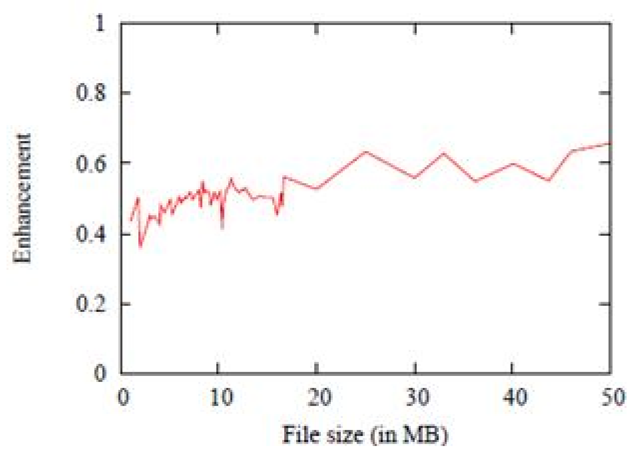

Figure 3. Enhancement variation with respect to the file size

Thus, to compare our new content delivery model with the point-to-point one, we evaluate the enhancement as a metric having the following expression:

$$Enhancement = \frac{T_b - T_m}{T_b}. \quad (9)$$

In Figure 2, we plot the CDF (i.e., Cumulative Distribution Function) of the metric enhancement estimated from the *200* traces. The figure shows that for around *85%* of the traces, our new model for content distribution improves the delivery time by a proportion between *0.4* and *0.9*. The rest of traces

DOI: 01.IJIT.3.2.1

(i.e., around *15%*) shows an enhancement between *0.1* and *0.4*. The considerable enhancement of content delivery time can be also observed from the expected value of the enhancement metric (averaged over the whole traces) which is obtained equal to *0.49*. In other terms, our traces show that with high probability, the delivery time could be decreased to an amount smaller than roughly half of its value when applying our model for content delivery instead of the classic point-to-point delivery method.

To study the variation of the enhancement with respect to the file size, we plot it in Figure 3 where the first metric is on the *y-axis* and the second one is on *x-axis*. Every point in the graph represents the average enhancement of several traces transferring contents of particular size in megabytes. One can observe from the figure that while the enhancement fluctuates around the value *0.5*, the graph shows a positive correlation between the enhancement metric and the file size. Moreover, other measures show that when the file size increases considerably over *50MB* till the value *250MB*, the enhancement fluctuates between *0.6* and *0.7*. This interval of file sizes has not been plotted in Figure 3 since its traces contain only one trace every increase of *10MB*; thus, we are unable to plot the average enhancement values over this interval. However, a positive correlation has been also clearly perceived through the coefficient which obtained equal to *0.47*.

Hence, one can realize that our model for content delivery is able to decrease more considerably the latency when transferring larger digital contents by striping the data across multiple TCP sockets (i.e., one per network interface). This observation can be due to the fact that when the content size increases the congestion avoidance phase becomes more dominant than the slow start phase of the download connection; we notice that the congestion window size increases exponentially in the slow start phase and then linearly in the congestion avoidance phase. In this case, it is obvious to observe better enhancement when reducing greater number of rounds in the congestion avoidance phase and spending more rounds in the slow start phase.

V. Conclusions And Perspectives

In this paper, we propose a new model to improve the content distribution in overlay networks. Our model takes advantage from the content replication and multi-homing facilities which are widely available nowadays. This is done by determining the best peer that could be reached through each network interface based on the estimation of the application utility function. Then, it consists of achieving bandwidth aggregation by striping data across multiple TCP sockets (i.e., one per network interface) that download the content chunks from their associated best servers simultaneously.

Our extensive real measurements show clearly how our solution outperforms the existing solutions by decreasing considerably the content distribution time. Our traces show that with high probability, the delivery time could be decreased to an amount smaller than roughly half of its value when applying our model for content delivery instead of the classic point-to-point content delivery methods. This is due to the fact that it combines the best server selection scheme with the bandwidth aggregation facility in multi-homing environment. Also, the results show that it is able to provide better quality of service when distributing larger content.

Besides, it is more flexible to be deployed than the solutions that depend on network infrastructure nodes and those proposed at the transport layer. Thus, our approach does not require the deployment of any special network node and does not impose any change to the existing reliable transport protocol TCP. This is ensured by working at the application-level and in a fully distributed way.

Concerning the challenge of determining the best server, many solutions have been proposed for estimating the application utility function in a scalable way. This could be done by firstly defining a function of the parameters impacting application performance and then relying on a scalable approach for inferring these parameters using a limited set of measurements.

Regarding our future work, we will test the efficiency of the presented model when used for overlay construction. Besides, we will investigate how our approach reacts to congestion.

Acknowledgment

The author wishes to thank Mr. Ahmad Moosa for his valuable help in the testing phase.

References

[1] Akamai, http://www.akamai.com.

[2] BitTorrent, http://www.bittorrent.com.

[3] H. H. Liu, Y. Wang, Y. R. Yang, H. Wang, and C. Tian, "Optimizing cost and performance for content multihoming", *ACM SIGCOMM*, 2012.

[4] V. K. Adhikari, Y. Guo, F. Hao, M. Varvello, V. Hilt, M. Steiner, and Z.-L. Zhang, "Unreeling netflix: Understanding and improving multi-CDN movie delivery", *IEEE INFOCOM*, 2012.

[5] G. Bertrand, E. Stephan, G. Watson, T. Burbridge, P. Eardley, and K. Ma, "Use cases for CDNi", *IETF Draft*, 2012.

[6] H. A. Alzoubi, S. Lee, M. Rabinovich, O. Spatscheck, and J. Van Der Merwe, "A practical architecture for an anycast CDN", *ACM Transactions on the Web*, vol. 5, pp. 17-29, 2011.

[7] R. S. Peterson, B. Wong, and E. G. Sirer, "A content propagation metric for efficient content distribution", *ACM SIGCOMM*, 2011.

[8] I. Poese, B. Frank, B. Ager, G. Smaragdakis, and A. Feldmann, "Improving content delivery using provider-aided distance information", *ACM IMC*, 2010.

[9] R. S. Peterson and E. G. Sirer. Antfarm, "efficient content distribution with managed swarms", *NSDI*, 2009.

[10] D. Wischik, M. Handley, M. Braun, "The resource pooling principle", *ACM SIGCOMM CCR*, vol. 38, pp. 47-52, 2008.

[11] D. Wischik, M. Handley and C. Raiciu, "Control of multipath TCP and optimization of multipath routing in the Internet", *NetCOOP*, 2009.

[12] M. Zhang, J. Lai, A. Krishnamurthy, L. Peterson, and R. Wang, "A Transport Layer Approach for Improving End-to-End

Performance and Robustness Using Redundant Paths”, *USENIX*, 2004.

[13] H. Hsieh and R. Sivakumar, “pTCP: An End-to-End Transport Layer Protocol for Striped Connections”, *IEEE ICNP*, Paris, 2002.

[14] H. Hsieh and R. Sivakumar, “A Transport Layer Approach for Achieving Aggregate Bandwidths on Multi-homed Mobile Hosts”, *ACM MOBI-COM*, Atlanta, GA USA, 2002.

[15] E. Ng and H. Zhang, “Predicting Internet network distance with coordinates-based approaches”, *IEEE Infocom*, 2002.

[16] L. Tang, and M. Crovella, “Virtual Landmarks for the Internet”, *ACM IMC*, 2003.

[17] B. Wong, A. Silvkins, and E. G. Sirer, “Meridian: A Lightweight Network Location Service without Virtual Coordinates”, *ACM SIGCOMM*, 2005.

[18] P. Francis, S. Jamin, C. Jin, Y. Jin, D. Raz, Y. Shavitt, and L.Zhang, “IDMaps: A Global Internet Host Distance Estimation Service”, *IEEE/ACM Transactions on Networking*, 2001.

[19] Y. Shavitt, and T. Tankel, “Big-Bang Simulation for Embedding Network Distances in Euclidean Space”, *IEEE Infocom*, 2004.

[20] H. Lim, J. Hou, and C. Choi. “Constructing Internet Coordinate System based on Delay Measurement”, *ACM IMC*, 2003.

[21] J. Ledlie, P. Gardner and M. Seltzer, “Network Coordinates in the Wild”, *NSDI*, 2007.

[22] S. Ratnasamy and M. Handly, R. Karp and S. Shenker, “Topologically-Aware Overlay Construction and Server Selection”, *IEEE Infocom*, 2002.

[23] M. Malli, C. Barakat, and W. Dabbous, “Application-level versus Network-level Proximity”, *Asian Internet Engineering Conference*, Thailand, 2005.

[24] M. Malli, C. Barakat, and W. Dabbous, “CHESS: An Application-aware Space for Enhanced Scalable Services in Overlay Networks”, *IEEE Computer Communication Journal*, vol. 31, pp. 1239-1253, 2008.

[25] M. Malli, C. Barakat, and W. Dabbous, “An Efficient Approach for Content Delivery in Overlay Networks”, *IEEE CCNC*, 2005.

[26] L. Ding, and R. Goubran, “Speech Quality Prediction in VoIP Using the Extended E-Model”, *IEEE Globecom*, 2003.

[27] R. Cole, and J. Rosenbluth, “Voice over IP Performance Monitoring”, *ACM SIGCOMM CCR*, vol. 31 , pp. 9-24, 2001.

[28] N. Hu, P. Steenkiste, “Exploiting Internet Sharing for Large Scale Available Bandwidth Estimation”, *ACM IMC*, 2005.

[29] N. Hu and P. Steenkiste, “Evaluation and Characterization of Available Bandwidth Techniques”, *IEEE JSAC Special Issue in Internet and WWW Measurement, Mapping, and Modeling*, 2003.

[30] V. J. Ribeiro, R. H. Riedi, R. G. Baraniuk, J. Navratil, and L. Cottrell, “PathChirp: Efficient Available Bandwidth Estimation for Network Paths”, *Passive and Active Measurement Workshop*, 2003.

DOI: 01.IJIT.3.2.1