A Novel Server Selection Method to Achieve Delay-Based Fairness in the Server Palm

Young-Tae Han, *Student Member, IEEE,* Min-Gon Kim, *Member, IEEE,* and Hong-Shik Park, *Member, IEEE*

***Abstract*—It is pivotal to achieve delay-based fairness when users access the same content, especially in real-time services, from content-replicated servers based upon the client-server communication model. To resolve this issue, this letter proposes a novel server selection method of providing users with delay-based fairness from all the corresponding servers by applying the delay-based dynamic deficit round robin (D^3RR) scheme. The deficit counter of the D^3RR is adjusted based upon round trip times (RTTs) of data traffic measured by passive-based probing which ensures low resource waste. Performance evaluation results highlight that the delay variance among all the corresponding servers can be maintained as low as possible and delay-based fairness is guaranteed without unnecessary waste of resources.**

***Index Terms*—Load balancing, server selection, delay-based fairness, deficit round robin.**

I. Introduction

TO support scalable and reliable real-time Internet services such as Web, Video on Demand (VoD), games, and Internet Protocol Television (IPTV) services, the same content, especially in real-time services, is replicated in client-server communication model-based multiple servers, which are geographically placed in a single or multiple districts. Generally, when users access the content, a single or multiple server selection nodes forward the connection requests of users to an appropriate server. The dominant server selection method is a round-robin (RR)-based method [1] for achieving throughput-based fairness with respect to the amount of traffic or the number of connections without concern about *the end-to-end delay* from a client to the its corresponding server. Therefore some users might be dissatisfied by delay variances among servers according to states of each duplicated server.

Basically, the end-to-end delay is mainly affected by the delay from the server selection node to the corresponding server in the server palm, but not the backbone network environment, because backbones are usually over-provisioned [2]. In particular, server processing capability and different connection requests from clients can cause delay variance among servers. If a high level of delay variance occurs in a case where users access the same content from different servers, then Quality of Experience (QoE) of users will be significantly degraded [3]. Therefore, a method of selecting an appropriate server for a client request is necessary for achieving an target performance regarding delay-based fairness.

Previous studies on this aspect have just proposed an Anycast Domain Names (ADN)-based server selection method with the active probing mode [3], which can cause additional overheads to the network and a probing process, and a Shared Passive Network Performance Discovery (SPAND)-based server selection method with the passive probing mode [4], which requires additional hardware devices and complex processes to keep consistency of information between the measurement point and server selection node. The contribution of these studies [3], [4] is to keep a low level of delay variance for guaranteeing delay-based fairness, but those are restricted to HTTP applications. Considering operational complexity, architectural simplicity, and various applications, we proposes a novel server selection method, which can provide users with delay-based fairness from replicated servers by applying the delay-based dynamic deficit round robin (D^3RR) scheme on the server selection node. The deficit counter of the D^3RR is calculated based upon round trip times (RTTs) measured with TCP connections [5] by the passive mode [6] measuring data traffic for a low resource waste differing from the active mode [3]. The performance evaluation results substantiate the fact that the proposed method can keep as low delay variance as possible and thus delay-based fairness can be achieved contrary to the RR-based method. The following section explains the detailed operation of the proposed method including the RTTs measurement and the D^3RR, and then performance evaluation results prove its effectiveness.

II. The Proposed Server Selection Method

A. Measurement of RTTs with Passive Mode

Basically, in order to acquire information about the server states, probing modes are divided into two types: (i) the active mode, which additionally sends probing packets to the target servers whenever information is required (i.e., timely information can be acquired, but there is a deterioration of network performance and additional overheads to the target servers occur) [3] and (ii) the passive mode, which measures with data packets; which allows for a lower network burden and higher network resource efficiency [6]. To capitalize on the benefits of the passive mode, it is adopted as a part of the proposed server selection method to measure the RTTs.

The basic principle of the RTTs measurement is based upon the transmission time of the observed data from a client to

Manuscript received April 27, 2009. The associate editor coordinating the review of this letter and approving it for publication was S. Pierre.

This work was supported by the MKE, Korea, under the ITRC support program supervised by the NIPA (NIPA-2009-(C1090-0902-0036)), and in part by Daegu Gyeongbuk Institute of Science and Technology (DGIST) Research Program of the MEST, the Development Man-made Disaster Prevention Technology grant funded by the Korea government (NEMA; National Emergency Management Agency) (No.Nema-09-MD-06).

Y.-T. Han and H.-S. Park are with the Dept. of Information and Communications, Korea Advanced Institute of Science and Technology (KAIST), Daejeon, S. Korea (e-mail: han0tae@gmail.com, parkhs@ee.kaist.ac.kr).

M.-G. Kim is with the Public & Original Technology Research Center, DGIST, Daegu, S. Korea (e-mail: kmg0803@gmail.com).

Digital Object Identifier 10.1109/LCOMM.2009.090967

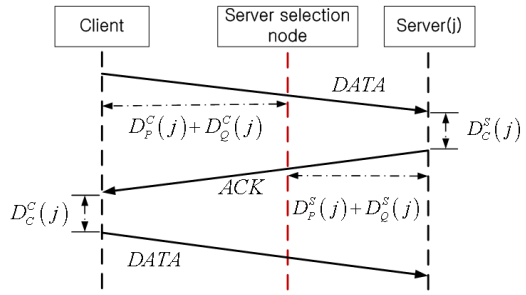


Fig. 1. Delineation of the delay within a single TCP connection.

the corresponding server(j). As shown in Fig. 1, the end-to-end RTT between clients and the corresponding server(j) is decomposed as follows: (1) $D_P^C(j)$: the propagation delay between a client and the server selection node, (2) $D_Q^C(j)$: the queuing delay of intermediate nodes between a client connected to server(j) and the server selection node, (3) $D_C^C(j)$: the processing delay of a client connected to server(j), (4) $D_P^S(j)$: the propagation delay between the server selection node and server(j), (5) $D_Q^S(j)$: the queuing delay of intermediate nodes between the server selection node and server(j), and (6) $D_C^S(j)$: the processing delay of server(j). Then, the RTT between the server selection node and the corresponding server(j) ($D_R(j)$) is given by:

$$D_R(j) = 2 \cdot D_P^S(j) + 2 \cdot D_Q^S(j) + D_C^S. \tag{1}$$

Finally, the average RTT between the server selection node and the corresponding servers(j) ($\hat{D}_R(j)$) is obtained by:

$$\hat{D}_R(j) = \sum_{j=1}^{N} \frac{D_R(j)}{N}, \tag{2}$$

where *N* is the number of sampled RTTs for the corresponding server(j) at the server selection node. Therefore, the average RTT between the server selection node and all the corresponding servers is given by:

$$\hat{D}_R = \sum_{k=1}^{M} \frac{\hat{D}_R(j)}{M}, \tag{3}$$

where M is the number of corresponding servers.

B. Delay-based Dynamic Deficit Round Robin Scheme (D³RR)

Based upon the average RTTs obtained from the aforementioned passive measurement (e.g., $\hat{D}_R(j)$ and $\hat{D}_R$), a scheme for appropriately selecting a corresponding server for a request of the connection is proposed by modifying the conventional deficit round robin (DRR) [1], which has a low time complexity (i.e., O(1)) and has achieved fairness regarding traffic load and the number of connections. In particular, the deficit counters (*DC*s) of the D³RR for each corresponding server are in inverse proportion to its average RTT to consider delay variance but not throughput and the number of connections. Thus, the value of *DC* of server(j) ($DC(j)$) is defined as:

Algorithm 1 The operation of the D³RR scheme

Require: DC_{init} and D_{th}

```
while a connection request do
  if no established connection to the server palm, then
    DCs for all the corresponding server are set to DC_init, and switch to the first server (j=0).
  else
    while DC(j) = 0 do
      Switch to the next corresponding server (j=j+1).
    end while
    Allocate the connection request to server(j) and DC(j) = DC(j) - 1.
    Update D̂_R(j).
    if |D̂_R − D̂_R(j)| > D_th or All DCs = 0, then
      Update D̂_R and DCs for all the corresponding servers.
      Switch to the first corresponding server (j=0).
    else
      if j==N then
        Switch to the first corresponding server (j=0).
      else
        Switch to the next corresponding server (j=j+1).
      end if
    end if
  end if
end while
```

$$DC(j) = \left\lceil \frac{\sum_{k=1}^{M} \hat{D}_R(k)}{\hat{D}_R(j)} \right\rceil, \tag{4}$$

where M is the number of corresponding servers.

Algorithm 1 elucidates the overall operation of the D³RR, as follows. In the beginning of the operation (no established connection to the server palm), if a connection request arrives at the server selection node, the values of DCs for all the corresponding servers are set to the initial value of the deficit counters (DC_{init}), because there is no information about the state of each server. After requests are evenly distributed to all the corresponding server at the beginning, the connection request is allocated to the server whose *DC* is greater than 0, which causes its *DC* to decrease. Then, to check the state of the RTTs between the server selection node and the corresponding servers, the values of $\hat{D}_R(j)$ are updated. After that, if the delay difference between the $\hat{D}_R$ and $\hat{D}_R(j)$ is greater than D_{th} (i.e., the threshold value for checking the delay variance of server(j) significantly increases compared

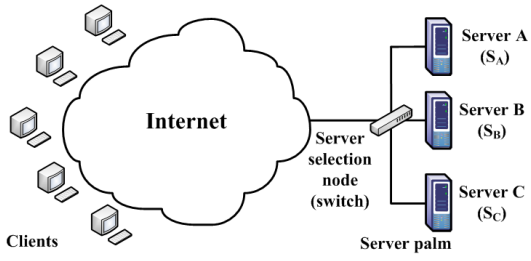
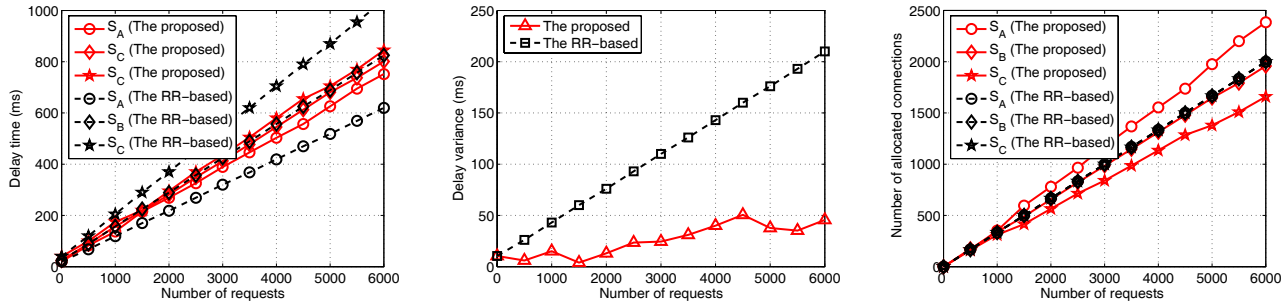


Fig. 2. Topology for simulation.



(a) Delay times between the server selection node and all the corresponding servers

(b) Delay time variances between the server selection node and all the corresponding servers

(c) The number of the allocated connections for each corresponding server

Fig. 3. Performance comparison of the proposed and the RR-based methods.

to other corresponding servers) or DCs are 0 for all the corresponding servers, $\hat{D}_R$ and the values of DCs for all the corresponding servers are newly updated; otherwise, after switching to the next corresponding server, the server selection node will wait to receive another connection request. As a consequence of the D^{3}RR, the connections already supported and newly allocated can be treated under a similar delay depending on the observed RTTs.

III. Performance Evaluation and Discussions

This section presents the performance comparison of the proposed method and the RR-based method by using computer simulation. Simulation results are obtained with the following assumptions: (i) there are three different types of corresponding servers in the server palm: Server A (S_A) whose transmission delay time including propagation delay and queuing delay to the server selection node is 20ms and processing delay time is increased per 10 connections by 3ms; Server B (S_B) whose transmission delay time to the server selection node is 25ms and processing delay time is increased per 10 connections by 4ms; and Server C (S_C) whose transmission delay time to the server selection node is 40ms and processing delay time increased per 10 connections by 5ms, as depicted in Fig. 2, (i.e., processing capability and background processes of each server are put into consideration), (ii) the queuing delay of intermediate nodes between the server selection node and the corresponding servers is not considered as a part of the performance evaluation (because generally the server selection node and the corresponding servers are close to each other in the server palm), (iii) only DATA-ACKs are taken into account in the sample data but not SYN-ACKs (because the RTTs of DATA-ACKs, including the processing time of the server, are more appropriate for inferring state of the server than those of SYN-ACKs [7]), and (iv) D_{init} and the D_{th} are set to the number of all the corresponding servers and 10ms, respectively.

Fig. 3 shows the performance results in terms of (i) the delay times between the server selection node and all the corresponding servers, (ii) the delay time variances between the server selection node and all the corresponding servers, and (iii) the number of the allocated connections for each corresponding server. First of all, differing from the RR-based method, the proposed method can maintain similar delay times for all the corresponding servers, as presented in Fig. 3(a). Thus, the delay time variance can be kept as low as possible, as shown in Fig. 3(b). This results come from the fact that the proposed method distributes connection requests differently to each corresponding server based on the delay time periodically measured by the passive probing, whereas the RR-based method evenly distributes connection requests to each corresponding server, as presented in Fig. 3(c).

In order to control delay time variance, we can adjust the value of D_{th}, which can control the initiation time of newly updating $\hat{D}_R$ and DCs for all the corresponding servers. If the value is set to a smaller one, delay time variance will decrease but more overhead for frequent update process will occur. On the other hand, if the value is set to a higher one, delay time variance will increase but less overhead will occur. Consequently, depending on the characteristics of applications supported in the server palm, D_{th} can be decided adaptively.

For the resource and operational efficiencies, the proposed method can keep resource waste and time complexity low by applying the passive probing and the conventional DRR. Therefore, it is possible to provide a similar quality of Internet services regarding the perspective of delay performance, and thus QoE of users will be well guaranteed due to a low level of delay variance in a case where users access the same contents from replicated servers.

References

[1] M. Shreedhar and G. Varghese, "Efficient fair queuing using deficit round-robin," *IEEE/ACM Trans. Networking*, vol. 4, no. 3, pp. 375-385, June 1996.

[2] S. Ranjan, R. Karrer, and E. Knightly, "Wide area redirection of dynamic content by Internet data centers," in *Proc. IEEE INFOCOM*, Mar. 2004.

[3] E. Zegura, M. Ammar, Z. Fei, and S. Bhattacharjee, "Application-layer anycasting: a server selection architecture and use in a replicated web service," *IEEE/ACM Trans. Networking*, vol. 8, no. 4, Aug. 2000.

[4] M. Stemm, R. Katz, and S. Seshan, "A network measurement architecture for adaptive applications," in *Proc. IEEE INFOCOM*, Mar. 2000.

[5] H. Jiang and C. Dovrolis, "Passive estimation of TCP round-trip times," *ACM Comp. Commun. Rev.*, vol. 32, no. 3, July 2002.

[6] S. G. Dykes, K. A. Robbins, and C. L. Jeffery, "An empirical evaluation of client-side server selection algorithm," in *Proc. IEEE INFOCOM*, no. 3, pp. 1361-1370, Mar. 2000.

[7] S. Jaiswal, G. Iannaccone, C. Diot, J. Kurose, and D. Towsley, "Inferring TCP connection characteristics through passive measurements," in *Proc. IEEE INFOCOM*, Mar. 2004.