A Cooperative User-System Approach for Optimizing Performance in Content Distribution/Delivery Networks

---

Citation:

Hiroki Nishiyama, Hiroshi Yamada, Hideaki Yoshino, and Nei Kato, "A Cooperative User-System Approach for Optimizing Performance in Content Distribution/Delivery Networks," IEEE Journal on Selected Areas in Communications, vol. 30, no. 2, pp. 476-483, Feb. 2012.

URL:

http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6136833

A Cooperative User-System Approach for Optimizing Performance in Content Distribution/Delivery Networks

Hiroki Nishiyama, *Member, IEEE*, Hiroshi Yamada, Hideaki Yoshino, *Member, IEEE*,
and Nei Kato, *Senior Member, IEEE*

***Abstract*—Recently, the demand for content delivery in wired/wireless heterogeneous networks is increasing at a rapid pace. Content Distribution/Delivery Networks (CDNs) are considered to be one of the best solutions for dealing with this increasing demand. In this paper, we point out that the performance of a CDN typically degrades in such heterogeneous environments due to the changes in not only user demand but also wireless mobility, which triggers unexpected fluctuations in traffic. Wireless users, roaming between different access networks, may contribute to sudden and unexpected demand spikes in certain parts of the content delivery system. To address this issue, we develop a cooperative server selection scheme, which is designed to maximize robustness to such changes with the cooperation between the content delivery system and its users. The performance of our proposal is evaluated by extensive computer simulations. The evaluation results demonstrate that our proposed scheme effectively makes the considered content delivery system resilient against request fluctuations while minimizing system overloading.**

***Index Terms*—Content distribution/delivery networks, server selection, cooperative system, heterogeneous networks, and load balancing.**

I. INTRODUCTION

In the last decade, the ways of accessing Internet entered an amazing transformation featuring a heterogeneous mixture of wired and wireless connectivity, which promises to offer an unprecedented level of ubiquitous broadband access. Such recent advances in network technology have contributed to an upsurge in demand for next generation content services. For example, next generation wireless technologies such as Worldwide Interoperability for Microwave Access (WiMAX) and third/fourth-generation (3G/4G) cellular systems, as well as the current generation wireless technologies (e.g., Wireless Fidelity or WiFi), are enabling users with mobile access to real-time data-intensive applications, e.g., streaming movies and video conferencing, with a greater ease. In addition to these infrastructure-based technologies, infrastructure-less wireless networking technologies can also allow us to construct flexible networks [1]. As these new technologies and services continue to develop further and gain more popularity, it is crucial to ensure that their underlying content delivery system also goes through similar advancement.

The Content Distribution/Delivery Network (CDN), one of the content delivery paradigms, has been widely embraced by the industry as the standard method for disseminating large amounts of contents to the users. As the demand for real-time streaming applications grows, end-users may tend to more increasingly depend upon CDNs in procuring these services. The nature of real-time traffic and/or on-demand contents gives rise to a new set of challenges in dealing with the highly variable requests from the users. Seemingly insignificant content at one moment, may surge to the peak of popularity in the next, triggering a sharp rise in demand, which can bring network performance to a halt, and even cripple unprepared servers. Furthermore, mobility of users presents complicated challenges to CDNs [2]. Heterogeneous access environments allow mobile users to switch the network freely according to different situations. This change can also trigger similar fluctuations in demand.

Although a number of schemes for improving the performance of CDNs exist in literature, a cooperative approach, which may consider the needs of both the the CDN system and its users is essential in order to maximize the remaining network resources to deal with unexpected changes in demand. In order to address this issue, content server selection technique holds much promise. If servers are not selected and assigned to the users with appropriate care, some of the servers will not only become overloaded, but also as they are inundated with traffic, local hot-spots will appear in those areas resulting in a further decline in performance. With this in mind, in this paper, we propose a new "cooperative user-system" approach, which uses server selection to not only satisfy the user demand and system constraints, but also ensures high resource availability to combat the unexpected growth in demand by effectively taking the hot-spot problem into account.

The remainder of the paper is structured as follows. Section II provides an overview of current server selection methods in CDNs. Following the problem formulation in Section III, Section IV presents our proposed approach. In Section V, we present our simulation results and analysis to evaluate the performance of the proposed approach. Finally, Section VI concludes the paper.

Manuscript received February 15, 2011; revised September 25, 2011.

H. Nishiyama and N. Kato are members of the Graduate School of Information Sciences at Tohoku University, Japan. They may be contacted at bigtree@it.ecei.tohoku.ac.jp and kato@it.ecei.tohoku.ac.jp, respectively.

H. Yamada works for the NTT Information Sharing Platform Laboratories in Tokyo, Japan. He may be contacted at yamada.hiroshi@lab.ntt.co.jp.

H. Yoshino works for the NTT Service Integration Laboratories in Tokyo, Japan. He may be contacted at yoshino.hideaki@lab.ntt.co.jp.

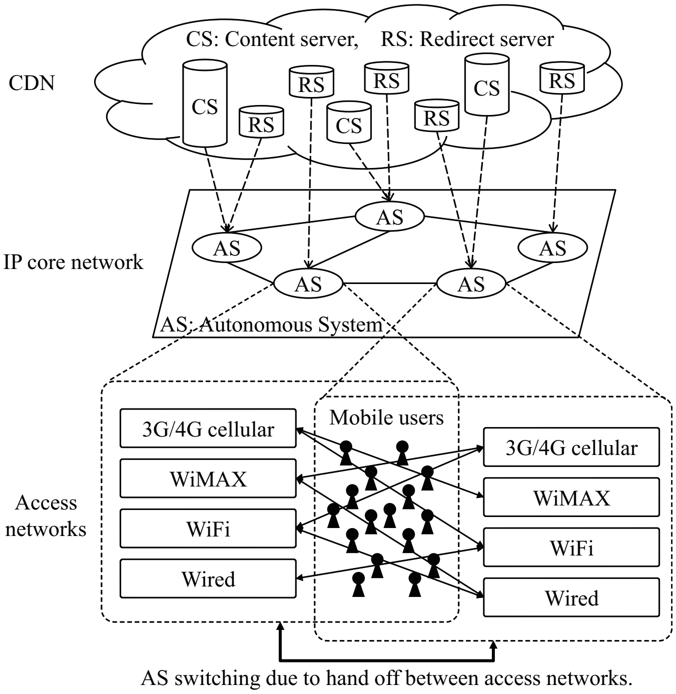


Fig. 1. A CDN system over the Internet.

II. CONVENTIONAL SERVER SELECTION APPROACHES

To date, networks based upon Internet Protocol (IP) account for most of the current Internet communications. Autonomous Systems (ASs) and collections of IP networks such as Internet Service Providers (ISPs) are connected together and collectively form the Internet. CDNs are also no exception – a CDN is implemented as a network built on top of the existing IP infrastructure as depicted in Fig. 1. The content-delivery infrastructure, which is responsible for distributing the content to the users, consists of a number of content servers. Redirect servers also play an essential role to distribute user requests to the appropriate content servers. Since routing is performed at the underlying IP layer and operated by third-party companies, redirect servers are only used for selecting the appropriate content server(s).

For server selection, there are two approaches based on a static criteria and a dynamic set of criteria, respectively. In the static server selection approach, the server is chosen based on non-changing parameters (e.g., the number of hops, link bandwidth, content delivery server performance, and so forth), which are configured during the deployment of the network. The static approach is simple and achieves good performance as demonstrated in the research work conducted in [3] when traffic fluctuation is low. It is also ideal for situations where real-time network estimation techniques are not available. However, due to its static nature, it cannot deal with dynamic fluctuations in traffic. Recent researches have dedicated lots of effort on dynamic server selection [4]–[7], in which the server is chosen based upon dynamic parameters subject to abrupt fluctuation, e.g., the number of user requests, network link load, and the content server load. Although these dynamic metrics can be exploited to improve the server selection, preparing for future fluctuations without leading to a substantial overhead remains a major issue.

Furthermore, how to optimally select the server (i.e., following the user-centric way or adopting the server-oriented strategy) also raises an issue. Individual user-centric approaches aim at minimizing transmission delays and/or throughput maximization, such as covered in the work in [8]–[10]. Although user-centric approaches optimize the performance of individual users, they do not guarantee on achieving the optimal performance of the entire system. In contrast, system-oriented approaches [3], [5], [11], [12] strive to make system-wide optimizations, such as increasing the total throughput, and balancing the network and/or server load.

Note that the afore-mentioned approaches do not adequately take into account the level of tolerance of the sudden fluctuations in requests from the users. If excessive requests are dispatched to a particular content server, the server will not be affected itself alone, but the overall performance in the vicinity of the server will degrade. This phenomenon can severely degrade the performance of the entire CDN. In fact, traffic fluctuations, which lead to such situations, may be represented as a function of the demand of users. While most of the previous research works have assumed that users demand can be estimated from demand-time history or gradual pattern in dynamic changes, it is not a reasonable methodology in the CDN context, whereby the number of users may suddenly and significantly change due to users mobility and frequent switching of the connected network.

Recent advances in IP networking technology have given rise to a new heterogeneous access environment built on wireless and wired technology, particularly in the last-mile, closest to the end-users. As a consequence, the next generation content delivery services must be robust enough to such traffic fluctuations, which occur unexpectedly and can cause drastic degradation in performance of the CDN. In order to address this issue, in this paper, we design a new dynamic server selection method, which provides a load balancing mechanism to account for sudden changes in traffic. By avoiding traffic interference, which creates problems for over-used links, a CDN can be made resilient against dynamic changes in user requests. Our system also comprises cooperative interaction between the CDN system and its users in terms of solving the problem by minimizing denial of service while maintaining an acceptable level of system load.

III. PROBLEM FORMULATION

In this section, we first describe our research motivation, and present an overview of the considered system model. Then, we formulate our research objective as an NP-hard optimization problem.

A. Motivation behind our research

Our research aims at maximizing the robustness of a CDN system to encounter the effect of unexpected sudden requests fluctuations. Note that our motivation differs from the existing ones, which attempt at maximizing the number of users and the available network resources at the same time. Most of the previous research work have focused on optimizing the performance of CDN under the scenarios whereby some servers or network links are overloaded due to the concentration of requests or traffic due to inappropriate selection of servers. In such scenarios, different types of optimization schemes may be considered from the points of view of different entities (i.e., server, network, and end-users) by using various criteria. In contrast, our research focuses on the scenario that there is no overloaded equipment in the system, but the distribution of the remaining resources may not be optimal for satisfying further demands. The performance of CDNs in such scenarios have not been sufficiently addressed in literature due to the assumption that the distribution of request onsets in time and space is, although subject to change at some point of time, remains consistent for a certain period following a change. This is owing to the fact that the request event is assumed to be just a function of users demand. Therefore, much research effort has been dedicated towards developing methodologies that consider distributing requests or traffic from crowded areas to uncrowded ones.

However, in the next generation CDNs, most of the users are supposed to access the CDNs by using mobile terminals via different access networks, which are parts of different ASs. In such situations, users' mobility and hand-off from the access network may contribute to demand spikes in certain ASs. It is impossible to forecast when and where such request fluctuations or spikes may occur. Also, it is quite difficult to dynamically control the server selection method by following such sudden and instantaneous changes. Therefore, we focus on maximizing the robustness of the CDNs for unexpected request fluctuations before such changes may cause actual congestions.

B. Considered System model

Since CDNs continue to grow in size in order to meet the rising demands and to more effectively serve the users, we consider a distributed CDN system. This consideration is similar, in spirit, with numerous research works on CDNs [6], [8], [13]. In our distributed model as shown in Fig. 1, CDN providers usually deploy content servers in multiple ASs. CDN providers also place a redirect server per AS. Content delivery servers are designed to both store and serve a considerable amount of content to the users (i.e., the customers who will subscribe and obtain the content). In our model, we assume that each content server is capable of establishing multiple connections and its capacity is defined by the maximum number of connections available simultaneously. Redirect servers are endowed with the responsibility of redirecting user requests to the appropriate content servers (i.e., the available servers having the desired content).

When a user issues a request for a particular content, the request is sent to the local redirect server in the same AS as user. Upon receiving the request, the redirect server forwards the request to the appropriate content server. This decision is based on the information collected and maintained by the redirect servers in various ASs. It is worth noting that although this information is useful for making intelligent decisions based on the current state of the network, since past trends do not predict future demands, it cannot be used with ease to anticipate imminent user requests. This is why we stress on introducing a new server selection algorithm on each redirect server by utilizing their information on content servers and AS networks. Indeed, information sharing between the content and redirect servers is easy and practical since they are usually deployed by the CDN providers in partnership. On the other hand, necessary information on AS networks, network link loads and routing information, can be obtained as described in detail later in Section IV-D.

C. Problem definition

As mentioned earlier, our approach attempts to maximize the robustness of the considered CDN to unexpected changes in users requests by appropriately selecting content servers even when none of the servers and/or network links is under a high congestion level. Note that we assume not having any prior knowledge in forecasting when and where (i.e., at which content server) the next request convergence will occur. In this work, we suppose that the spike in demand is not likely to happen in different ASs at the same moment. Also, we assume that, in each AS, the available capacity of the content server(s) for meeting the unexpected increase of incoming requests is considered to derive the maximum number of further requests, which can be successfully served by the content servers distributed over the network. Therefore, our objective is to maximize the minimum available capacity for the incoming requests spike (i.e., the sudden increase in demand) in the ASs. This can be formulated as an optimization problem including max-flow problems as described below.

Server selection policy can be expressed as a vector, $\boldsymbol{P} = (p_1, p_2, ..., p_k, ..., p_{|\boldsymbol{S}|})$, where p_k indicates the probability of selecting the k^{th} server from the set of servers, $\boldsymbol{S}$. It follows trivially that $\sum p_k = 1$. It depends on the employed selection technique if $\boldsymbol{P}$ is similar among redirect servers or not. Users get content streams from the servers through the network, $\boldsymbol{G'}(\boldsymbol{N}, \boldsymbol{L})$, where $\boldsymbol{N}$ and $\boldsymbol{L}$ are the sets of ASs and links, respectively. Suppose that all content streams have the same rate. The capacity and utilization of the link l are denoted by c_l^L and u_l^L, respectively. Similarly, the capacity and utilization of the server s are indicated by c_s^S and u_s^S, respectively. These values have a unit of connections. It should be noted that $\boldsymbol{u^L}$ and $\boldsymbol{u^S}$ are affected by $\boldsymbol{P}$. Here, we consider the residual graph of $\boldsymbol{G'}$ denoted by $\boldsymbol{G}$, which is a function of $\boldsymbol{P}$, $\boldsymbol{c^L}$, and $\boldsymbol{c^S}$. The maximum flow, $MaxFlow(\boldsymbol{G}, i)$, from servers to i^{th} AS in $\boldsymbol{G}$ can be obtained by solving the max-flow problem where the source is the set of servers and the destination is the AS. Consequently, the objective of our approach can be defined by the following optimization problem, which derives the optimal server selection policy, i.e., $\boldsymbol{P}$:

$$\max_{\boldsymbol{P}} \ \min_{i \in \boldsymbol{N}} \ MaxFlow\left(\boldsymbol{G}(\boldsymbol{P}, \boldsymbol{c^L}, \boldsymbol{c^S}), i\right), \tag{1}$$

$$\text{s.t.} \quad 0 \leq u_l^L \leq c_l^L, \qquad \forall l \in \boldsymbol{L}, \tag{2}$$

$$0 \leq u_s^S \leq c_s^S, \qquad \forall s \in \boldsymbol{S}, \tag{3}$$

$$0 \le p_s \le 1, \qquad \forall s \in \boldsymbol{S}, \tag{4}$$

$$\sum_{s \in \boldsymbol{S}} p_s = 1. \tag{5}$$

Although the individual max-flow problems can be solved in actual time, the entire max-min optimization problem is, unfortunately, NP-hard. In order to overcome this shortcoming, in the next section, we propose a heuristic server selection algorithm, which attempts to approximately optimize the problem.

IV. COOPERATIVE SERVER SELECTION RESILIENT TO REQUEST FLUCTUATION

In this section, we propose a cooperative server selection algorithm in a heuristic manner for approximately solving the optimization problem defined in the previous section.

A. *Different policies for different scenarios*

In the problem formulation in the previous section, we only consider the situation that the CDN system is not overloaded and still has sufficiently available resources to serve the users. However, the considered server selection scheme should include an algorithm for scenarios whereby the system is partially reaching its performance limit following a demand spike (i.e., several content servers and/or CDN links are congested). In other words, our scheme adopts two different server selection policies for congested and non-congested areas of the considered CDN, respectively. While it aims to make the system resilient to unexpected requests spike where the system equipments are still not congested, it aims at minimizing the risk of denial of service in the congested zone. Actually, in the proposed scheme, which makes a decision based on server selection costs, the different cost functions (i.e., f_1 or f_2) are utilized according to whether each content server is under non-congested or congested situations, respectively. If selecting a certain content server results in employing any congested equipment for the content delivery, the content server is regarded as in the congested situation. Otherwise, the content server is deemed to be under the non-congested situation.

As criteria to distinguish whether servers or links are congested or not, thresholds for each content server and network link that indicate their maximum acceptable utilization ratios are introduced. Utilization ratios exceeding the thresholds represent congestion. Let θ_k^S and θ_j^L denote the threshold for the k^{th} server and the j^{th} link, respectively. Since the values of these thresholds may be determined according to the system management policies in CDNs or ISPs, our scheme assumes that these values have been pre-set and remain the same throughout the course of content distribution.

In the proposed scheme, when a redirect server receives a content delivery request from a user belonging to the same AS as the redirect server, it determines an adequate content server based on the cost of selecting each content server. One of the content servers having the smallest cost is randomly selected. Before calculating the selection cost of a content server, the proposed scheme, at first, makes sure if any congested equipment exists in the path between the content server and the user's AS, and accordingly determines the appropriate cost function (i.e., either f_1 or f_2). As defined in the following, we design the cost functions so that the range of f_1 is below that of f_2. This design represents that content servers under non-congested situations are to be preferentially selected because the proposed scheme chooses a content server having the smallest cost.

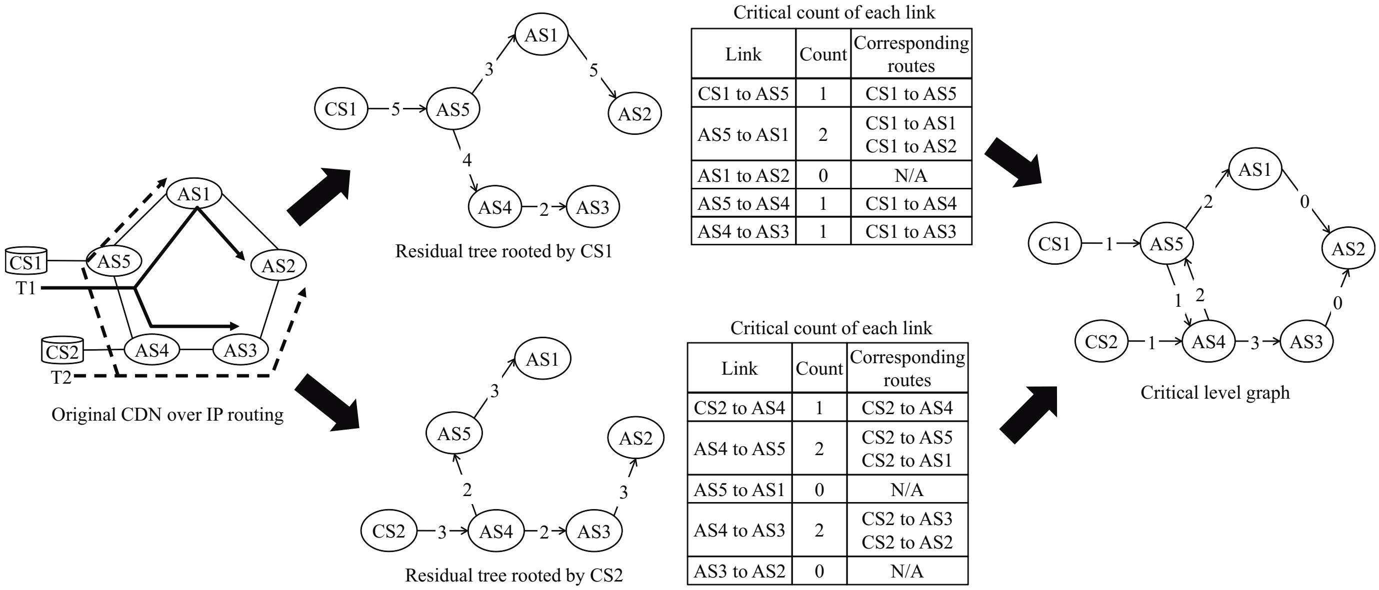
B. *Server selection cost for non-congested situations*

When the CDN system has highly available resources, it is important to ensure its resiliency to unexpected request spike in the future. Since the original problem formulated in the previous section is NP-hard, we need to develop a heuristic approach. One of the most straightforward approximate solutions is to sequentially select a content server to avoid reducing the minimum $MaxFlow$ at the time for every request in each AS. However, it involves computing the maximum flow for every new request in every AS. Instead, the proposed scheme utilizes the concept of "critical link".

Critical links, defined based on maxflow and mincut operations [14], represent those links in the CDN that, when over-used, decrease the maximum flow of other ASs. As a consequence, critical links must be used as sparingly as possible to provide leftover capacity to absorb fluctuations in future requests. Therefore, content servers should be selected so that the load on critical links is reduced. As strictly identifying critical links is difficult for large scale networks, we note that critical links are essentially the bottleneck links in the network. The server selection cost function is defined by using the *critical level* derived from critical links. Details of this is presented in the following.

As shown in Fig. 2, the critical level graph, used for computing server selection costs, can be obtained from residual trees rooted by individual content servers. Residual trees are generated from content delivery trees, which are determined by IP routing. In the residual tree, ASs and the content server are represented as nodes, and a number associated with each edge indicates its residual bandwidth (i.e., the remaining/available bandwidth). Residual bandwidths of edges connecting different ASs are derived from the maximum acceptable traffic rates defined by their thresholds and their latest utilization. The capacity of the edge between the content server and its AS can be defined in the same manner except that the capacity of the content server represented as the number of connections is transformed to the bandwidth by exploiting the average content stream rate.

For each route between the root content server and an individual AS in the residual tree, the link in the route with the minimum bandwidth is counted as the critical link of the route. The critical level of each link is defined as the total number of times the link is counted as a critical link. For example, tables of critical count shown at the center of Fig. 2 can be summarized into the critical level graph as depicted on the right side in the figure. Based on the critical level graph, the cost function of selecting i^{th} content server at j^{th} AS is



Critical count of each link

Link	Count	Corresponding routes
CS1 to AS5	1	CS1 to AS5
AS5 to AS1	2	CS1 to AS1 CS1 to AS2
AS1 to AS2	0	N/A
AS5 to AS4	1	CS1 to AS4
AS4 to AS3	1	CS1 to AS3

Critical count of each link

Link	Count	Corresponding routes
CS2 to AS4	1	CS2 to AS4
AS4 to AS5	2	CS2 to AS5 CS2 to AS1
AS5 to AS1	0	N/A
AS4 to AS3	2	CS2 to AS3 CS2 to AS2
AS3 to AS2	0	N/A

Fig. 2. Computation of a sample critical level graph.

defined as follows.

$$f_1(i|j) = \frac{\sum_{l \in R_{i,j}} x_l}{\max\limits_k \left(\sum_{l \in R_{k,j}} x_l \right)}, \qquad 0 \leq f_1 \leq 1, \tag{6}$$

where $R_{i,j}$ represents the route from the i^{th} content server to the j^{th} AS, and x_l denotes the critical level of the link l included in the corresponding route. The closer f_1 is to one, the more sparingly it should be selected.

C. Server selection cost for congested situations

Under the congested situations, the proposed scheme aims to minimize the occurrence of denial of service, i.e., rejection ratio of requests, because reducing service denials is an essential requirement of users, CDNs, and ISPs. The denial of service can be mainly caused by two different reasons, namely (i) no available connection remaining at the selected content server, and/or (ii) traffic overflow at the link on the path from the selected content server to the user's AS. For sake of simplicity, we consider a link as overloaded if the traffic rate traversing the link exceeds its bandwidth.

Since the request rejection probability at each content server can be considered as similar to the rate of loss calls represented by the Erlang-B formula [15], the request acceptance probability at the i^{th} content server is expressed as follows:

$$P_S(i) = 1 - \frac{u_i^S}{1 + u_i^S}, \tag{7}$$

where u_i^S is the connection utilization ratio of the i^{th} content server. On the other hand, the risk of traffic overflow occurrence at each link can be estimated based on queuing theory. Regardless of the arrival and service distributions, the probability of having no overflow at the link l is illustrated in the following equation:

$$P_L(l) = 1 - u_l^L, \tag{8}$$

where u_l^L denotes the utilization ratio of the link l.

As mentioned earlier, denial of service may be caused by the excess of network link load and/or server load. Thus, the cost function of selecting the i^{th} content server at the j^{th} AS is defined by the following equation based on the probability of successfully serving a content stream without any excess load.

$$f_2(i|j) = 1 - P_S(i) \times \prod_{l \in R_{i,j}} P_L(l) + bias. \tag{9}$$

Here, $R_{i,j}$ denotes the route from the i^{th} content server to the j^{th} AS. The $bias$ parameter is intended to maintain a larger value of f_2 in contrast with that of f_1 at any time, i.e., its value should not be below one. It is worth reminding that each redirect server determines the appropriate cost function (i.e., either f_1 or f_2) for every content server according to the load conditions at the server and the links on the route, and eventually selects the content server having the smallest cost. The proposed server selection algorithm is summarized in Algorithm 1.

D. Information sharing among redirect servers

In the proposed scheme, each redirect server regularly receives the server load information and network load information within its AS via a content server and an *AS Border gateway Router* (ASBR), respectively, which are deployed in the same AS. The static information on IP routing in the entire network is also available from ASBR. The redirect servers share these information with one another. As a consequence, all redirect servers can have access to the information of the entire CDN. Redirect servers must be regularly updated with fresh statistics (e.g., network and server loads) to function effectively. However, it should be noted that the information do not necessarily have to be updated with high frequency to detect sudden changes in demand, because the proposed scheme never aims at adaptively controlling such request fluctuations.

Algorithm 1 Server selection in j^{th} AS

```
for all content server $i$ do
    $Flag \leftarrow false$
    if utilization of $i$, $u_i^S >$ threshold for $i$, $\theta_i^S$ then
        $Flag \leftarrow true$
    end if
    if $Flag = false$ then
        for all link $l$ composed in $R_{i,j}$ do
            if utilization of $l$, $u_l^L >$ threshold for $l$, $\theta_l^L$ then
                $Flag \leftarrow true$
                break
            end if
        end for
    end if
    if $Flag = false$ then
        Calculate the cost of $i$ by using $f_1$
    else
        Calculate the cost of $i$ by using $f_2$
    end if
end for
return $i$ having the smallest cost
```

Instead, our scheme attempts at adequately distributing content streaming traffic before unexpected request spikes actually take place. Thus, it is sufficient to periodically obtain the latest information with a relatively coarse time interval.

V. PERFORMANCE EVALUATION

In this section, we evaluate the performance of the proposed method through extensive computer simulations conducted using MATLAB. First, the considered simulation environment is delineated. Then, the results and performance comparisons with conventional methods are presented.

A. Simulation Setup

In our conducted simulations, contents are delivered over the AS network consisting of thirty ASs. The AS network topology is generated by using a scale-free network generator [16] because it is widely known that the Internet is scale-free. Average node degree of the network is set to two. We assume that the routes between ASs are determined based on the hop count between them, i.e., the shortest path routing is utilized for this purpose. Three content delivery servers are used in the simulations, and each content server is deployed in an arbitrarily AS without duplication. To avoid the possibility that content allocation techniques may affect the performance, we further assumed that all content servers serve the same content, the rate and length of which are set to 1 Mbps and 300 seconds, respectively. For simplicity, thresholds of all links and content servers are set to the same value (i.e., 0.9). It should be noted that these values are also used to evaluate congested probability, which is described later. Simulation configuration parameters are listed in Table I. The capacities of individual links and content servers are randomly selected from within the given range.

TABLE I
SIMULATION CONFIGURATION PARAMETERS.

Parameter	Value
Number of ASs	30
Number of content servers	3
Content rate	1 Mbps
Content view time	300 sec
Range of link bandwidth	100Mbps – 200Mbps
Range of server capacity [connections]	200 – 400
Threshold for server utilization ratio θ^S	0.9
Threshold for link utilization ratio θ^L	0.9

TABLE II
SIMULATION SCENARIO SETTINGS.

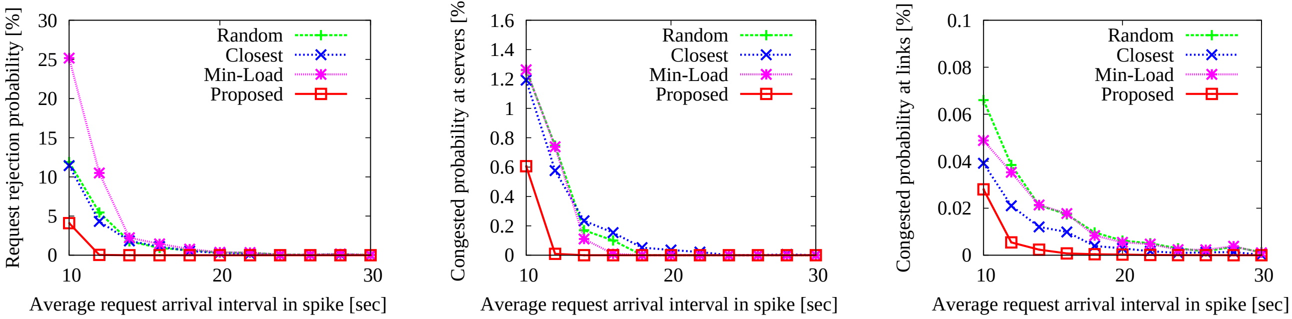
		Scenario I	Scenario II
Average request arrival interval [sec]			
	In normal	30	
	In spike	10 – 30	10
Number of ASs suffering a request spike		30	1 – 30

We evaluate our proposed approach under the two scenarios involving a request spike, which significantly impacts the performance of the simulated CDN. The settings of these scenarios are summarized in Table II. User requests are generated according to Poisson process, and thus request spikes are simulated by changing the average request arrival interval. Simulation time is set to ten times the content viewing time, and request spikes are invoked during the last one-tenth period of the entire course of simulation. All the simulation results are obtained from the average value of trials carried out by varying the network topology, link capacity, capacity and allocation of content servers, and request occurrences.

We evaluate our proposed method against the three server selection schemes, namely random select, closest select, and min-load select algorithms. The random select algorithm, which is also the most common server selection method, chooses a server randomly by using Round-Robin strategy. The closest-selection is another popular server selection method, which chooses a server based on the distance between the server and its users. The hop count is used as a metric of the distance. If more than one content server with the minimum value of the metric exist, one of the servers is randomly selected. In the min-load selection method, information about the loads on the content servers are used to decide which content server will be selected, i.e., one of the servers with the minimum load is randomly chosen. To exclude the influence of the implementation techniques of observing time changes in loads upon the performance of the considered approaches, we assume that there is no latency in load measurements in both the min-load and proposed methods.

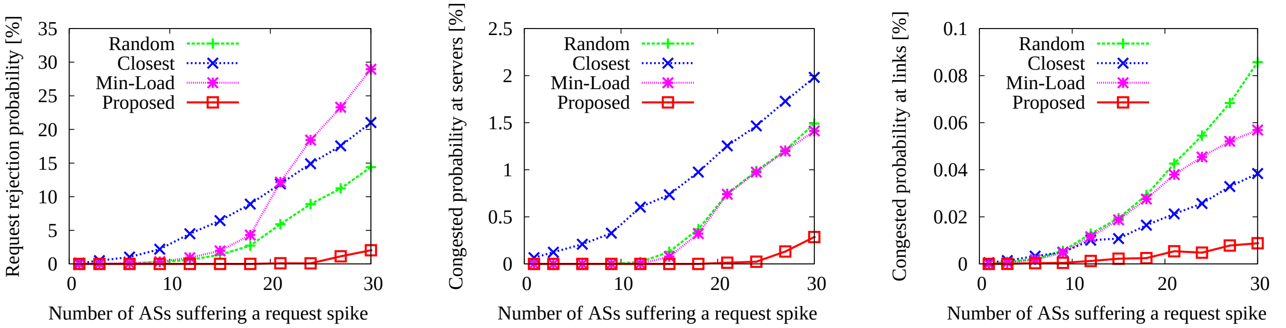
B. Simulation results

Figs. 3 and 4 demonstrate the results of simulations using $Scenarios\ I$ and II, respectively. In both the scenarios, we use the same performance metrics, request rejection probability, and congested probability at servers and links. All metrics are computed by using data measured during the period with



(a) Request rejection probability (b) Congested probability at servers (c) Congested probability at links

Fig. 3. Performance comparison in $Scenario\ I$.



(a) Request rejection probability (b) Congested probability at servers (c) Congested probability at links

Fig. 4. Performance comparison in $Scenario\ II$.

spikes. Congested probability can be calculated based on the duration under congestion and the excess amount of load at that time, by averaging it over the measurement period. Whether links or servers are congested or not is distinguished by using the thresholds.

$Scenario\ I$ is designed to model a CDN environment where request spikes simultaneously occur in all ASs. For example, this may be seen just after the appearance of a new content attracting much attention. In the simulations, the average request arrival interval in the spike is varied from 10 to 30 seconds. The results clearly demonstrate that our proposed approach provides the lowest probability of request rejection compared with the other conventional methods. While smaller request rejection probabilities represent that more traffic are passing through the network, the proposed scheme also succeeded in maintaining low congested probabilities at both the servers and links. This result implies that the proposed scheme can achieve highly efficient utilization of the remaining server and link capacities. Although the closest-selection and min-load selection methods are able to retain small congested probability at the links and servers, respectively, they fail to reduce the request rejection due to their one-sided strategy. It is clear that load balancing between the servers and links is essential to accept many more requests from the users.

In $Scenario\ II$, only a part of the ASs suffer the request spikes, which can occur in real situations by not only the changes in the popularity of the content but also due to mobility of the users. The number of ASs suffering a request spike is varied from one to 30. The results reveal that our proposed server selection strategy leads to the smallest request rejection probability, and a significant improvement in the congested probabilities in the considered servers and links. On the other hand, in the other three methods, all the considered metrics significantly increase with the large number of requests. In the comparison between the random and min-load selection methods, the request rejection ratio of the min-load selection method rapidly increases when the number of ASs suffering a request spike exceeds 20 while its congested probabilities at the servers and links are similar to or less than those of the random selection method. This result implies that the requests converge to specific servers and/or links. Therefore, it is evident that the requests must be distributed to avoid formation of a bottleneck. Indeed, this is what the proposed scheme manages to achieve. By allowing the content distribution system to accept many more user-requests, the proposed scheme utilizes the residual limited resources of the servers in an efficient and even manner, and thus avoids creating bottlenecks.

To summarize, through the conducted simulations, it is verified that the proposed method can successfully improve the request rejection probability, i.e., users' experiences, by mitigating the congestion events in both the servers and links, i.e., CDNs and ISPs, even in situations in which unexpected sudden request fluctuations may occur.

VI. CONCLUSION

As the demand for real time ubiquitous access to progressively larger amounts of content increases, Content Delivery

Networks or CDNs will play a significant role in next-generation networks in providing the customers with their desired contents. To withstand the highly complex dynamics of CDNs in the near future, the design and optimization of CDNs require to be thoroughly considered. Toward this end, in this paper, we proposed a new cooperative server selection method to improve the overall performance of CDNs by considering the needs of both the CDN system and its users. We have also addressed the problem of unexpected traffic fluctuations and provided an adequate solution for the same. The performance of our proposed server selection strategy is verified by computer simulations. The results of the simulations clearly demonstrate the effectiveness of the proposed strategy in terms of the service acceptance, and the servers/network load balancing.

REFERENCES

[1] J. Chen, W. Xu, S. He, Y. Sun, P. Thulasiraman, and X. Shen, "Utility-based asynchronous flow control algorithm for wireless sensor networks," *IEEE Journal on Selected Areas in Communications*, vol. 28, no. 7, pp. 1116–1126, Aug. 2010.

[2] M. Uyar, J. Zheng, M. Fecko, S. Samtani, and P. Conrad, "Evaluation of Architectures for Reliable Server Pooling in Wired and Wireless Environments," *IEEE Journal on Selected Areas in Communications*, vol. 22, no. 1, pp. 164–175, Jan. 2004.

[3] Y. Hu, J. Lan, and C. Wan, "An Algorithm for Unsplittable Flow Problem in Flexible Reconfigurable Network," in *Proc. of Fourth International Conference on Frontier of Computer Science and Technology*, Dec. 2009, pp. 543–547.

[4] L. Cai, J. Ye, J. Pan, X. S. Shen, and J. W. Mark, "Dynamic server selection using fuzzy inference in content distribution networks," *Computer Communications*, vol. 29, no. 8, pp. 1026–1038, May 2006.

[5] M. Menai, F. Fieau, A. Souk, and S. Jaworski, "Demonstration of Standard IPTV Content Delivery Network Architecture Interfaces: Prototype of Standardized IPTV Unicast Content Delivery Server Selection Mechanisms," in *Proc. of IEEE 6th Consumer Communications and Networking Conference*, Jan. 2009, pp. 1–2.

[6] J.-B. Chen and S.-J. Liao, "A Fuzzy-Based Decision Approach for Supporting Multimedia Content Request Routing in CDN," in *Proc. of International Symposium on Parallel and Distributed Processing with Applications*, Sep. 2010, pp. 46–51.

[7] R. Cohen and G. Nakibly, "A Traffic Engineering Approach for Placement and Selection of Network Services," *IEEE/ACM Transactions on Networking*, vol. 17, no. 2, pp. 487–500, Apr. 2009.

[8] Y. Bai, B. Jia, J. Zhang, and Q. Pu, "An Efficient Load Balancing Technology in CDN," in *Proc. of Sixth International Conference on Fuzzy Systems and Knowledge Discovery*, vol. 7, Aug. 2009, pp. 510–514.

[9] Y.-T. Han, M.-G. Kim, and H.-S. Park, "A Novel Server Selection Method to Achieve Delay-Based Fairness in the Server Palm," *IEEE Communications Letters*, vol. 13, no. 11, pp. 868–870, Nov. 2009.

[10] W. Jiang, R. Zhang-Shen, J. Rexford, and M. Chiang, "Cooperative Content Distribution and Traffic Engineering," in *Proc. of the 3rd international workshop on Economics of networked systems*, Aug. 2008, pp. 7–12.

[11] S. Ranjan and E. Knightly, "High-Performance Resource Allocation and Request Redirection Algorithms for Web Clusters," *IEEE Transactions on Parallel and Distributed Systems*, vol. 19, no. 9, pp. 1186–1200, Sep. 2008.

[12] M. Pathan and R. Buyya, "Resource Discovery and Request-Redirection for Dynamic Load Sharing in Multi-Provider Peering Content Delivery Networks," *ACM Journal of Network and Computer Applications*, vol. 32, pp. 976–990, Sep. 2009.

[13] J. Famaey, B. De Vleeschauwer, T. Wauters, F. De Turck, B. Dhoedt, and P. Demeester, "Dynamic QoE Optimisation for Streaming Content in Large-Scale Future Networks," in *Proc. of IFIP/IEEE International Symposium on Integrated Network Management-Workshops*, Jun. 2009, pp. 128–134.

[14] K. Kar, M. Kodialam, and T. Lakshman, "Minimum interference routing of bandwidth guaranteed tunnels with MPLS traffic engineering applications," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 12, pp. 2566 –2579, Dec. 2000.

[15] L. Kleinrock, *Queueing Systems, Vol. I: Theory*. John Wiley and Sons, 1975, vol. 1, ch. 3.

[16] M. N. George, "B-A Scale-Free Network Generation and Visualization." [Online]. Available: http://www.mathworks.com/matlabcentral/fx_ files/11947/1/content/html/pubfile.html.

Hiroki Nishiyama received his M.S. and Ph.D. in Information Science from Tohoku University, Japan, in 2007 and 2008, respectively. He was a Research Fellow of the Japan Society for the Promotion of Science (JSPS) until finishing his Ph.D, following which he went on to become an Assistant Professor at the Graduate School of Information Sciences at Tohoku University. He has received Best Paper Awards from the IEEE Global Communications Conference 2010 (GLOBECOM 2010) as well as the 2009 IEEE International Conference on Network Infrastructure and Digital Content (IC-NIDC 2009). He was also a recipient of the 2009 FUNAI Foundation's Research Incentive Award for Information Technology. He is a member of the Institute of Electronics, Information and Communication Engineers (IEICE) and an IEEE member.

Hiroshi Yamada received his B.S. degree in Mathematics from Nagoya University in 1985 and his Ph.D in network engineering from Tokyo Institute of Technology in 1996. He joined NTT laboratories in 1985, and works at NTT laboratories as a senior research engineer and supervisor. His recent work focuses on network design and management including ITIL, network performance analysis, and analyzing network and protocol performance through simulation in OPNET.

Hideaki Yoshino (M'03-) received the B.Sc., M.Sc. and D.Sc. degrees in information science from the Tokyo Institute of Technology, Tokyo, Japan, in 1983, 1985 and 2010, respectively. He joined Nippon Telegraph and Telephone Corporation (NTT) Laboratories in 1985 and has been engaged in communication traffic and service quality research. As a visiting scholar, he stayed at the University of Stuttgart, Germany, during 1990 - 1991. He is currently an executive researcher, supervisor and serves as a project manager of NTT Service Integration Laboratories, where he is conducting research and management on traffic and quality for future communications networks and services. Dr. Yoshino is a member of IEEE, IEICE Japan, and the Operations Research Society of Japan.

Nei Kato has been a full professor at GSIS, Tohoku University, since 2003. He has been engaged in research on computer networking, wireless mobile communications, and smart grid, and has published more than 200 papers in journals and peer-reviewed conference proceedings. Dr. Kato currently serves as the Chair of IEEE SSC TC, the Secretary of IEEE AHSN TC, the Vice Chair of IEICE Satellite Communications TC, and editor for three IEEE Transactions. His awards include Satellite Communications Award from the IEEE Communications Society, SSC TC in 2005, the IEICE Network System Research Award in 2009, and best paper awards from IEEE GLOBECOM 2010. He also serves on the expert committee of Telecommunications Council, Telecommunications Business Dispute Settlement Commission Special Commissioner, Ministry of Internal Affairs and Communications, Japan, and as Chairperson of ITU-R SG4, SG7, Japan.