

# Content Delivery Network

## Using Community Information

Takuya Kitano\*  
cstk11041@g.nihon-u.ac.jp

Shun-ichi Kurino\*  
kurino@math.cst.nihon-u.ac.jp

Noriaki Yoshikai\*  
yoshikai.noriaki@nihon-u.ac.jp

Toshio Takahashi\*\*  
Toshiozoo@gmail.com

(\*)  
Collage of Science and Technology  
Nihon University  
Tokyo Japan

(\*\*)  
Japan Organization for Employment of the Elderly, Persons  
with Disabilities and Job Seekers  
Nagano Japan

**Abstract**—A content delivery network (CDN) using community information that is extracted from the data of a social networking service (SNS) is proposed in this paper. The structure of the load characteristics of the CDN in terms of the number of servers, communities, and downloads is also shown as a tool for evaluating network performance.

**Keywords**—CDN, Community, Social Network Analysis

### I. INTRODUCTION

CDNs have been widely adopted for content delivery services. A CDN has many servers (CDN servers) scattered on the network, and keeps a great deal of content in these servers. When a client requests some content from a CDN, a CDN server mediates this request. If the CDN server has the requested content, the client can get it directly from this CDN server. If the content is not present on the server, the server requests the origin server to send the content. After getting the content, the CDN server sends the requested content to the client. To speed the downloading of the requested content, a CDN is expected to be able to predict the content that will be frequently used by clients, and to place such content in the servers prior to client requests. However, this forecasting function has not been realized yet.

It is possible, though, to investigate human community activity in cyberspace by analyzing social network services (SNS) such as Facebook and Twitter. The network community composed of SNS users can yield information that includes the connections among users and their degree of affinity. Generally, it is considered that users in a community with high affinity tend to have the same interests, so they are likely to use the same content. Based on this hypothesis, a novel CDN using such community information is proposed in this paper.

In Section II, the problems in existing CDNs are reviewed. Section III depicts the algorithm by which community information is extracted from SNS data. And Section IV explains the network architecture for our proposed CDN. Section V shows how to evaluate the network performance of the proposed CDN.

### II. CHALLENGES OF CDN

There are three kinds of server in existing CDNs<sup>[1]</sup>: edge server, storage server and origin server. When a client requests content, an edge server receives the request from the client. If the edge server has the requested content, it delivers the content to the client directly. Otherwise, the edge server forwards the request to either a storage server or an origin server, which supplies the requested content, which the edge server then delivers to the client. Content holders place their content in origin servers. Content that is likely to be used frequently by clients is copied from the origin server to the storage servers, even in advance of client requests. Although this mechanism can avoid traffic jams caused by numerous requests for popular content at origin servers, the content held in storage servers is usually limited to that already held by clients. Since it is very difficult to forecast which content will be popular, unused content might be stored in storage servers.

In another type of CDN, there is a network that copies all content in origin servers to all edge servers. Google Cache is a good example. This CDN enables every client to get all requested content from its servers in a short time. But this CDN is extremely costly because it requires a very large number of very high capacity servers.

In our proposed system, each CDN server stores only content that its clients are likely to want. The exact content stored in each CDN server depends on community information that describes the clients, as explained in section III. Since the total volume of memory in servers can be smaller than that of an existing CDN like Google Cache, this is expected to reduce the system cost.

### III. COMMUNITY EXTRACTION ALGORITHM

In order to get community information from a SNS's big data, a high-speed network community computing algorithm using two-stage clustering has been proposed<sup>[2]</sup>. The main flow is shown in Figure 1.

First, the SNS users are counted, using the HITS (Hyperlink-Induced Topics Search) algorithm<sup>[3]</sup>, and the users who participate in communities are identified. Next, data is gathered on networks whose links are bidirectional. Then, the communities can be extracted from the data by using two

clustering algorithms; CNM (Clauet Newman Moore) [4] and n-Clan.

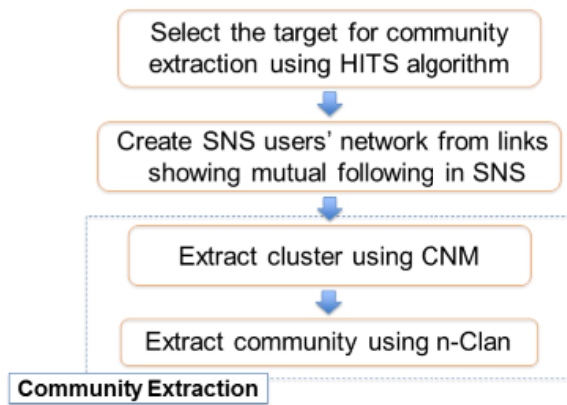


Figure 1. Community Extraction Algorithm

A community extracted by this algorithm is defined as a set of nodes that are clustered in the connection data. It is assumed that the members of a community are interested in the same things. From these conditions, we hypothesize that SNS users in the same community are likely to request the same content when part of a CDN. Based on this hypothesis, a novel CDN using community information is considered.

#### IV. NETWORK ARCHITECTURE OF NOVEL CDN

##### A. System Concept

According to our hypothesis, any content that is requested by one community member is likely to be requested by others. Therefore, the proposed system also locates content that a member has requested from one CDN Server on other CDN servers for the same community.

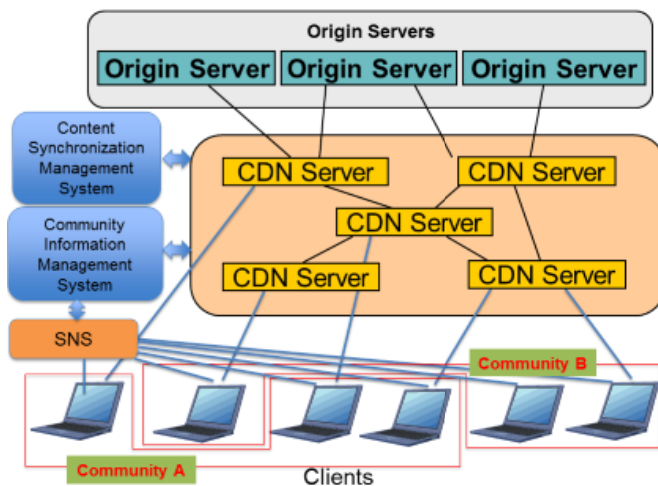


Figure 2. Framework of CDN using Community Information

A CDN that uses community information is configured by CDN servers, a community information management system (CIMS), and a content synchronization management system (CSMS), as shown in Figure2. CDN servers mediate the clients' requests and content, and cache the content like an http proxy server.

##### B. Community Information Management System(CIMS)

A CIMS manages the community information and community members' network information, such as IP addresses. Initially, the CIMS gets big data from a SNS and extracts relevant community information from it. Next, the CIMS assigns the members address in the network. Using this network information, the CIMS can find the CDN server that is nearest to each community member. Finally, the CIMS gives these CDN server lists to the CSMS.

##### C. Content Synchronization Management System(CSMS)

The CSMS manages content synchronization using the information received from the CIMS.

When a member (A) in a community requests content from a CDN server, the CDN server requests and gets that content from the origin server, if the CDN server does not already have the requested content. At the same time, the CDN server reports to the CSMS that is getting this requested content from the origin server. After getting this report from the CDN server, the CSMS talks to other CDN servers connected to members of A's community. The CDN servers that talked with the CSMS then request and get the content from the origin server. As a result of these actions, all CDN servers connected to members of that community have all the content that any member has requested, almost simultaneously. We call this process *content synchronization*. It enables all community members to get the same content that any member has requested, almost immediately.

##### D. Message Flow of Content Synchronization

An example of message flow in content synchronization is shown in Figure3.

A set of two SNS users, User1 and User2 are judged to be in a community by CIMS. CIMS assigns these users network addresses, and finds CDN Server1 and CDN Server2, which are the nearest CDN servers to User1 and User2.

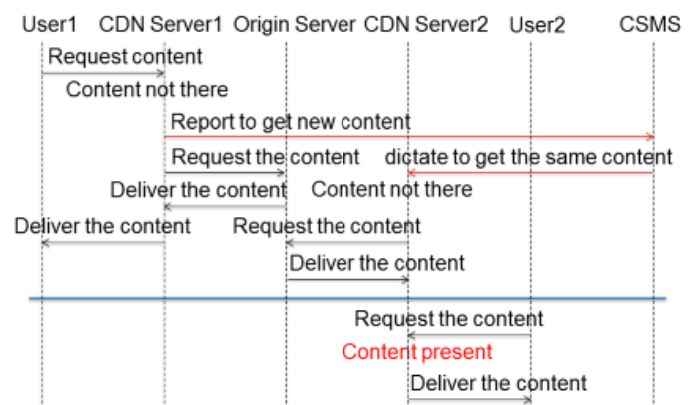


Figure 3. Message Flow of Content Synchronization

In Figure3, User1 requests content from CDN Server1. CDN Server1 gets this request, and checks its cache to see whether it has the required content or not. If not, CDN Server1 requests the content from the origin server. At the same time

that CDN Server1 requests the content, it reports to the CSMS that it is getting new content from the origin server. Getting this report, the CSMS requests the same content from CDN Server2, causing that server to check its cache for that content. If CDN Server2 does not have the content, it requests transmission of this content from the origin server. At this time, CDN Server1 and CDN Server2 get the content that was requested by user1. From then on, if User2 requests that content from CDN Server2, CDN Server2 already has the content, and User2 can get it from CDN Server2 immediately.

## V. SYSTEM PERFORMANCE

We have formulated expressions for the load of the system to allow evaluation of its performance. The load of the proposed CDN is compared with that of a system that does not use community information, using a formulation for the load characteristics that includes the number of servers, the number of communities, and the number of downloads.

The load is defined as the product of data size and the distance that the data flows.

The maximum load for downloading a particular content  $i$  times is defined in Equation (1).

$$(i - T_i)h + T_i\{H + (n - 1)C\} \cdots (1)$$

H: Maximum load for receiving content that flows from origin servers to all clients through CDN Servers

h: Maximum load for receiving content that flows from CDN servers to all clients

C: Maximum load for content synchronization

n: Number of CDN servers that are used by a community

$T_i$ : Number of content items that are synchronized in each CDN server, after the CDN carries out a download of content  $i$  times

The first term in Equation (1) defines the maximum load for downloading content from CDN servers. The second term defines the maximum load for downloading content from origin servers and for synchronizing the content among the CDN servers connected to community members.

Next, the maximum load for the model that requests a content download  $i$  times from the origin server directly to members is given by Equation (2).

$$Hi \cdots (2)$$

Figure 4 shows the general form of the maximum load, normalized by the number of download repetitions ( $i$ ). To the right of the intersection of Equations (1) and (2), the maximum load imposed by community members'  $i$  repetitions of a particular content download using our proposed system as defined in Equation (1) is lower than the maximum load imposed by the same downloads coming directly from the origin server as defined in Equation (2).

In the future we will demonstrate the benefits of our proposal compared with existing CDNs.

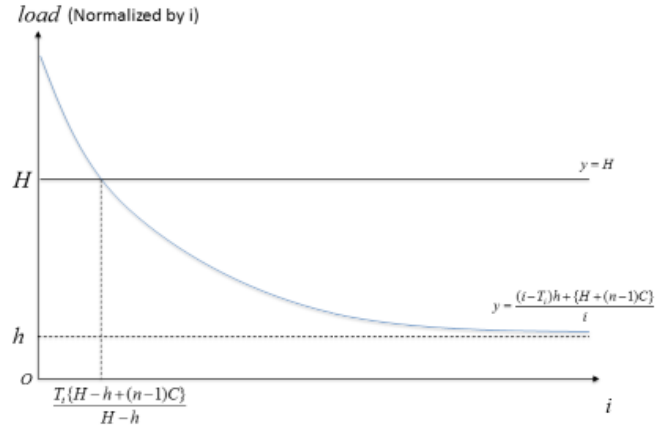


Figure 4. General form of the load

## VI. CONCLUSION

A content delivery network using community information is proposed, and its characteristics are explained. Many items remain to be specifically studied. In the CIMS, the method to assign the SNS user information to its network information must be clarified. A method to elicit location information from network information should also be shown. We will build a model system and evaluate the character of the CDN by experiment.

## ACKNOWLEDGEMENT

This research was supported by a grant from Scientific Research in Japan (Project no. 26330386).

### References

- [1] [http://www.akamai.com/dl/technical\\_publications/GloballyDistributedContentDelivery.pdf](http://www.akamai.com/dl/technical_publications/GloballyDistributedContentDelivery.pdf)
- [2] Mayako. Sato, Noriaki Yoshikai, Shun-ichi Kurino, Study on Network Community Analysis. –High Speed Computing Algorithm, IEICE SITE2014-53 pp.39-43,2014.
- [3] J.M.Kleinberg, “Authoritative sources in a hyperlinked environment”, Journal of the ACM, 46(5);pp.604-632,1999.
- [4] Aaron Clauset, M.E.J.Newman, Cristopher Moore, “Finding community substructure in very large networks”, Phys.Rev.E, 70:066111,Dec 2004.