International Journal of Modern Engineering Research (IJMER)
www.ijmer.com Vol. 3, Issue. 4, Jul - Aug. 2013 pp-2514-2521 ISSN: 2249-6645

An Efficient Distributed Control Law for Load Balancing in Content Delivery Networks

S. K. Mehertaj[1], K. V. Subbaiah[2], P. Santhi[3], T. Bharath Manohar[4]

[1,3]*M. Tech 2ndyr, Dept of CSE, PBRVITS (Affiliated to JNTU Anantapur), Kavali, Nellore. Andhra Pradesh. India.*
[2] *Assoc. Prof., Dept of CSE, PBRVITS (Affiliated to JNTU Anantapur), Kavali, Nellore. Andhra Pradesh.India.*
[4] *Asst. Professor, Dept of CSE, CMR College of Engineering &Technology,*
(Affiliated to JNTU Hyderabad)Hyderabad.Andhra Pradesh.India.

Abstract: *Content Delivery Networks (CDN) aim at overcoming the inherent limitations of the Internet. The main concept at the basis of this technology is the delivery at edge points of the network, in proximity to the request areas, to improve the user's perceived performance while limiting the costs. This paper focuses on the main research areas in the field of CDN, pointing out the motivations, and analyzing the existing strategies for replica placement and management, server measurement, best fit replica selection and request redirection. In this paper, we face the challenging issue of defining and implementing an effective law for load balancing in Content Delivery Networks .A formal study of a CDN system, carried out through the exploitation of a fluid flow model characterization of the network of servers.This result is then leveraged in order to devise a novel distributed and time-continuous algorithm for load balancing, which is also reformulated in a time-discrete version.*

Keywords: *CDN's Fluid flow model, Load balancing Algorithm.*

I. INTRODUCTION

Content Delivery Network (CDN) represents a popular and useful solution to effectively support emerging Web applications by adopting a distributed overlay of servers. By replicating content on several servers, a CDN is capable to partially solve congestion issues due to high client request rates, thus reducing latency while at the same time increasing content availability

- In this paper, we face the challenging issue of defining and implementing an effective law for load balancing in Content Delivery Networks.
- A formal study of a CDN system, carried out through the exploitation of a fluid flow model characterization of the network of servers.
- This result is then leveraged in order to devise a novel distributed and time-continuous algorithm for load balancing, which is also reformulated in a time-discrete version.

II. MOTIVATION

- In this paper, we face the challenging issue of defining and implementing an effective law for load balancing in Content Delivery Networks.
- A formal study of a CDN system, carried out through the exploitation of a fluid flow model characterization of the network of servers.
- This result is then leveraged in order to devise a novel distributed and time-continuous algorithm for load balancing, which is also reformulated in a time-discrete version.

SCOPE:

The most important performance improvements derived from the adoption of such a network concern two aspects:

1) Overall system throughput, that is, the average number of requests served in a time unit (optimized also on the basis of the processing capabilities of the available servers);
2) Response time experienced by clients after issuing a request. The decision process about these two aspects could be in contraposition. As an example, a "better response time" server is usually chosen based on geographical distance from the client, i.e., network proximity; on the other hand, the overall system throughput is typically optimized through load balancing across a set of servers. Although the exact combination of factors employed by commercial systems is not clearly defined in the literature, evidence suggests that the scale is tipped in favor of reducing response time.

Drawbacks of Existing System

- Goal of CDN can be achieved in many different ways, not all of which provide local stability guarantees, as well as balancing of the servers' queues.
- Indeed, it might happen that the overall condition is met, but one or more local server's queues overflow, thus bringing to packet losses and unavailability of the overloaded servers.

PROBLEM STATEMENT

- We focus exclusively on critical conditions where the global resources of the network are close to saturation.
- This is a realistic assumption since an unusual traffic condition characterized by a high volume of requests, i.e., a flash crowd, can always overfills the available system capacity.
- In such a situation, the servers are not all overloading which we have local instability conditions where the input rate is greater than the service rate.

NEW SYSTEM PROPOSAL

- We first design a suitable load-balancing law that assures equilibrium of the queues in a balanced CDN by using a fluid flow model for the network of servers.
- We present a new mechanism for redirecting incoming client requests to the most appropriate server, thus balancing the overall system requests load.
- Our mechanism leverages local balancing in order to achieve global balancing. This is carried out through a periodic interaction among the system nodes.

III. LITERATURE SURVEY

The commercial success of the Internet and e-services, together with the exploding use of complex media content online has paved the way for the birth and growing interest in Content Delivery Networks (CDN). Internet traffic often encounters performance difficulties characteristic of a non dedicated, best effort environment. The user's urgent request for guarantees on quality of service have brought about the need to study and develop new network architectures and technologies to improve the user's perceived performance while limiting the costs paid by providers. Many solutions have been proposed to alleviate the bottleneck problems and the most promising are based on the awareness of the content that has to be delivered. The traditional *"content-blind"* network infrastructures are not sufficient to ensure quality of service to all users in a dynamic and ever increasing traffic situation. New protocols and integrated solutions must be in place both on the network and on the server side to distribute, locate and download contents through the Internet. The enhancement of computer networks by means of a content aware overlay creates the new architectural paradigm of the CDN. Today's CDN act upon the traditional network protocol stack at various levels, relying on dynamic and proactive content caching and on automatic application deployment and migration at the edge of the network, in proximity to the final users. Content replicas in a CDN are geographically distributed, to enable fast and reliable delivery to any end-user location: through CDN services, up-to-date content, can be retrieved by end-users locally rather than remotely.

CDNs were born to distribute heavily requested contents from popular web servers, most of all image files. Nowadays, a CDN supports the delivery of any type of dynamic content, including various forms of interactive media streaming. CDN providers are companies devoted to hosting in their servers the content of third-party content providers, to mirroring or replicating such contents on several servers spread over the world, and to transparently redirecting the customers requests to the 'best replica' (e.g. the closest replica, or the one from which the customer would access content at the lowest latency). Designing a complete solution for CDN therefore requires addressing a number of technical issues: which kind of content should be hosted (if any) at a given CDN server (replica placement), how the content must be kept updated, which is the 'best replica' for a given customer, which mechanisms must be in place to transparently redirect the user to such replica. A proper placement of replica servers shortens the path from servers to clients thus lowering the risk of encountering bottlenecks in the non-dedicated environment of the Internet. A request redirection mechanism is provided at the access routers level to ensure that the best suited replica is selected to answer any given request of possibly different types of services with different quality of service agreements. The CDN architecture also relies on a measurement activity that is performed by cooperative access routers to evaluate the traffic conditions and the computational capacity and availability of each replica capable of serving the given request. Successfully implemented, a CDN can accelerate end user access to content, reduce network traffic, and reduce content provider hardware requirements.

IV. SYSTEM ANALYSIS

DISTRIBUTED LOAD-BALANCING ALGORITHM

We want to derive a new distributed algorithm for request balancing that exploits the results are presented. First of all, we observe that it is a hard task to define a strategy in a real CDN environment that is completely compliant with the model proposed. As a first consideration, such a model deals with continuous-time systems, which is not exactly the case in a real packet network where the processing of arriving requests is not continuous over time. For this reason, in the following of this section, we focus on the control law are described. The objective is to derive an algorithm that presents the main features of the proposed load-balancing law and arrives at the same results in terms of system equilibrium through proper balancing of servers' loads, as assessed by Lemma.

Algorithm Description

The implemented algorithm consists of two independent parts: a procedure that is in charge of updating the status of the neighbors' load, and a mechanism representing the core of the algorithm, which is in charge of distributing requests to a node's neighbours. The pseudocode of the algorithm is reported. Even though the communication protocol used for status information exchange is fundamental for the balancing process, in this paper we will not focus on it. Indeed, for our simulation tests, we implemented a specific mechanism: We extended the HTTP protocol with a new message, called *CDN*, which is periodically exchanged among neighboring peers to carry information about the current load status of the sending node. Naturally, a common update interval should be adopted to guarantee synchronization among all interacting peers. For this purpose, a number of alternative solutions can be put into places, which are nonetheless out of the scope of the present work.

```
// peer status update
prob_space[0]=0; load_diff = 0; load_diff_sum = 0;
for(j=1; j<=n; j++){
     if(load_i – peer[j].load){
          load_diff = load_i – peer[j].load;
          //insert the new difference
          build_prob_space(load_diff, prob_space);
          load_diff_sum = load_diff_sum + load_diff;
     }
     //normalize the vector elements
     update_prob_space(load_diff_sum, prob_space);
}

// balancing process
if(prob_space[] == NULL) //no neighbors with lower load
     //serve locally the request
     serve_request();
else{
     float x = rand(); //random number generator
     int req_sent = 0; int i = 0;
     while(prob_space[i] == 1 or req_sent == 1){
          if(prob_space[i-1] <= x < prob_space[i]){
               //send request to the chosen peer
               send_to(peer[i-1].addr);
               req_sent = 1;
          }
          i++;
     }
}
```

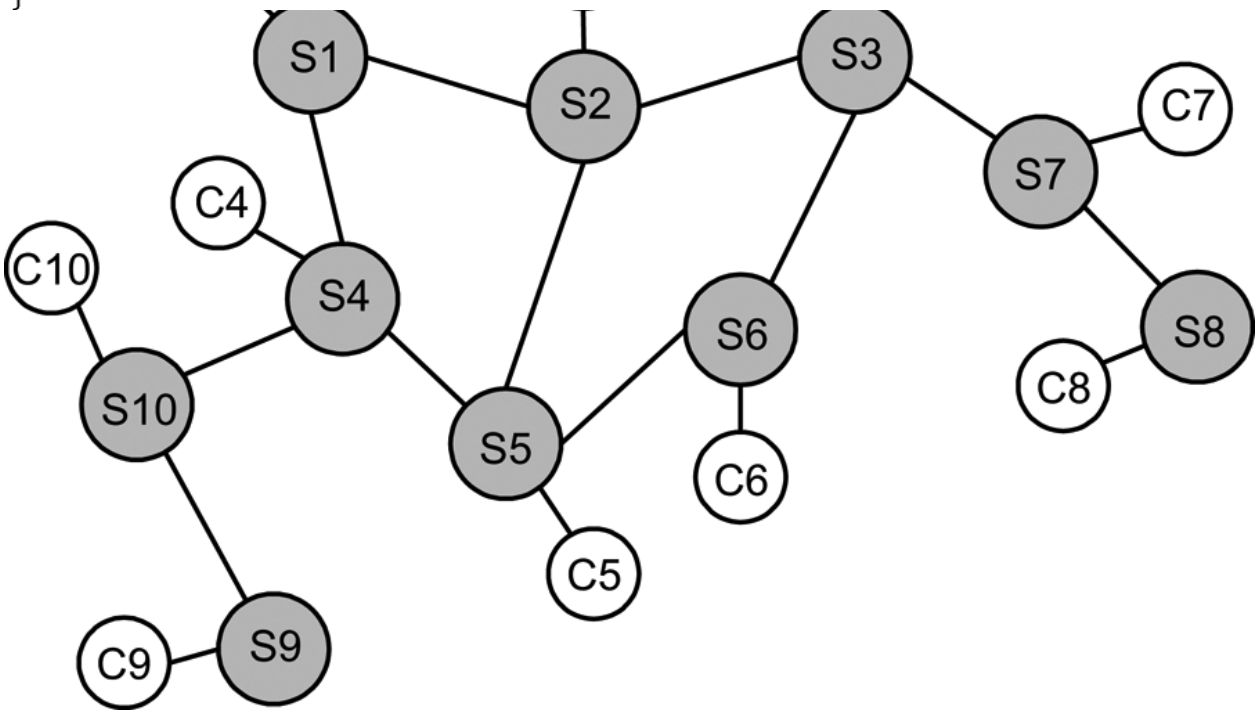


Fig: Pseudocode description of the proposed algorithm

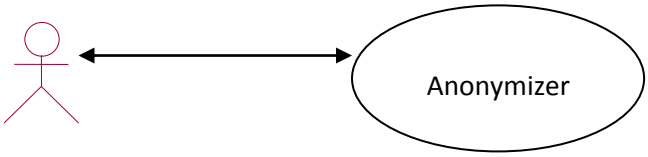
Functional Requirements Specification

This section outlines the use cases for each of the active readers separately. The reader, the author and the reviewer have only one use case apiece while the editor is main actor in this system

Different use cases in the system

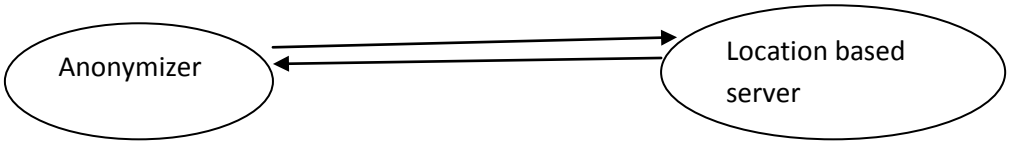
1. Admin

He/she enters into the mobile sink they are end mobile agents who participate in the communication .mobile agents passes /submits query to anonymizer and it sends to location-based service.



2. Anonymizer

It is an interface between user i.e. mobile agent and main server .It receives messages from user and signals to LBS/BS and then passes information to mobile nodes.It also selects local database and checks the databases for localization of mobile agents.



Mobile user (MU), issues the query and public key to the nearby anonymizer, Anonymizer passes the query and key to the base station (BS). As the LBS server is deployed at the BS, LBS processes the query and returns the result back to the BS , the BS knows that the user is in its region and gives a active signal to all the anonymizers in that particular region.

3. LBS (LOCATION BASED SERVER)

It is the main server acts as a base station and processes the requests from clients and sends message to mobile agents. It signals the anonymizer about mobile locations.

Overall Description

In this paper, we presented a novel load-balancing law for cooperative CDN networks. We first defined a model of such networks based on a fluid flow characterization. We hence moved to the definition of an algorithm that aims at achieving load balancing in the network by removing local queue instability conditions through redistribution of potential excess traffic to the set of neighbors of the congested server. The algorithm is first introduced in its time-continuous formulation and then put in a discrete version specifically conceived for its actual implementation and deployment in an operational scenario. Through the help of simulations, we demonstrated both the scalability and the effectiveness of our proposal, which outperforms most of the potential alternatives that have been proposed in the past.

External Interface Requirements

The only link to an external system is the link to the Historical Society (HS) Database to verify the membership of a Reviewer. The Editor believes that a society member is much more likely to be an effective reviewer and has imposed a membership requirement for a Reviewer. The HS Database fields of interest to the Web Publishing Systems are member's name, membership (ID) number, and email address (an optional field for the HS Database).

- The *Assign Reviewer* use case sends the Reviewer ID to the HS Database and a Boolean is returned denoting membership status. The *Update Reviewer* use case requests a list of member names, membership numbers and (optional) email addresses when adding a new Reviewer. It returns a Boolean for membership status when updating a Reviewer.
- Product Functions

Logical database requirement

DATA BASE ARCHITECTURE

There are three types of architecture: ONE TIER ARCHITECTURE TWO-TIERED THREE-TIERED ONE TIER ARCHITECTURE

The application and the data reside together logically. These are not usually database programs. The logic and its data reside together. Figure below shows a model of a single-tier application.

TWO-TIERED

The application resides in a different logical location than the data. These are usually database applications. Most client/Server applications fit into this category. figure shows a model of a two-tier application.

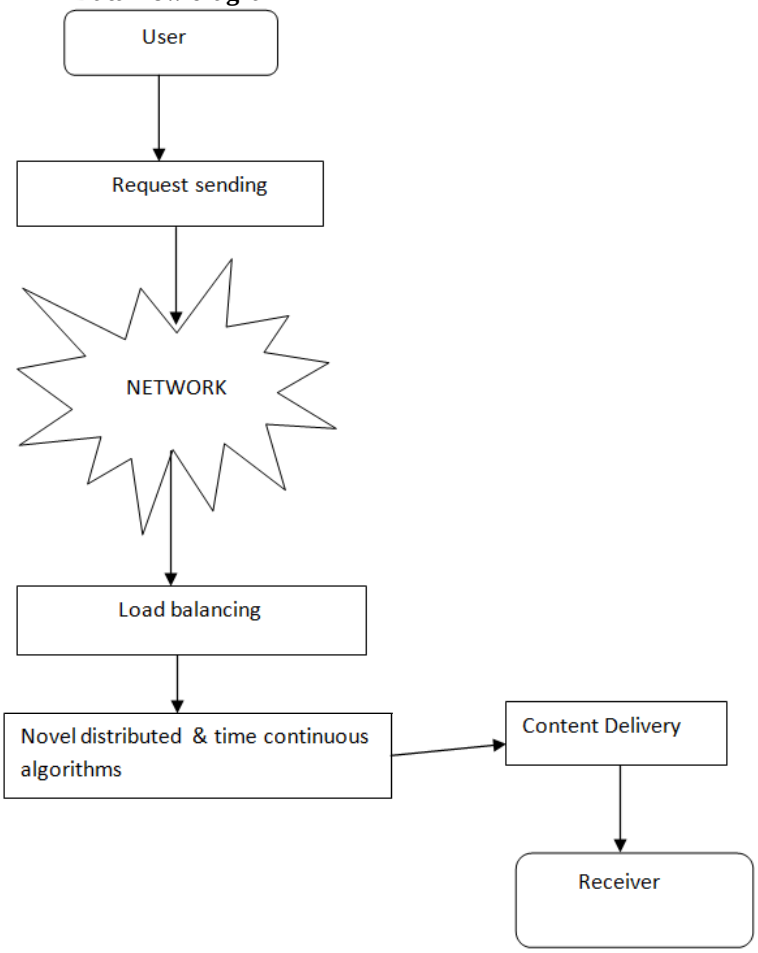
THREE-TIERED

In a three-tiered system, the application resides in a different logical location than the logic of the application and the data.

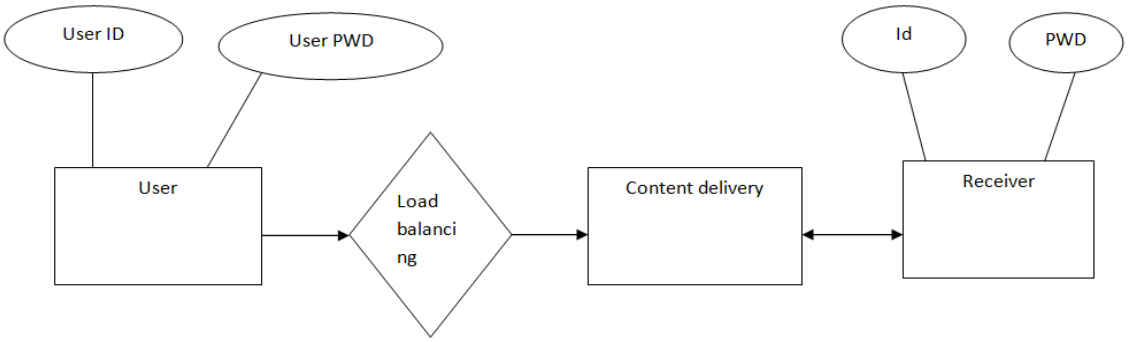
To put it another way, the client software makes a call to a remote service. that remote service is responsible for interacting with the data and responding to the client. the client has no knowledge of how and where the data is stored. All it knows about is the remote service has no knowledge of the clients that will be calling it. It only knows about the data.

E-R DIAGRAMS AND DATA FLOW DIAGRAMS

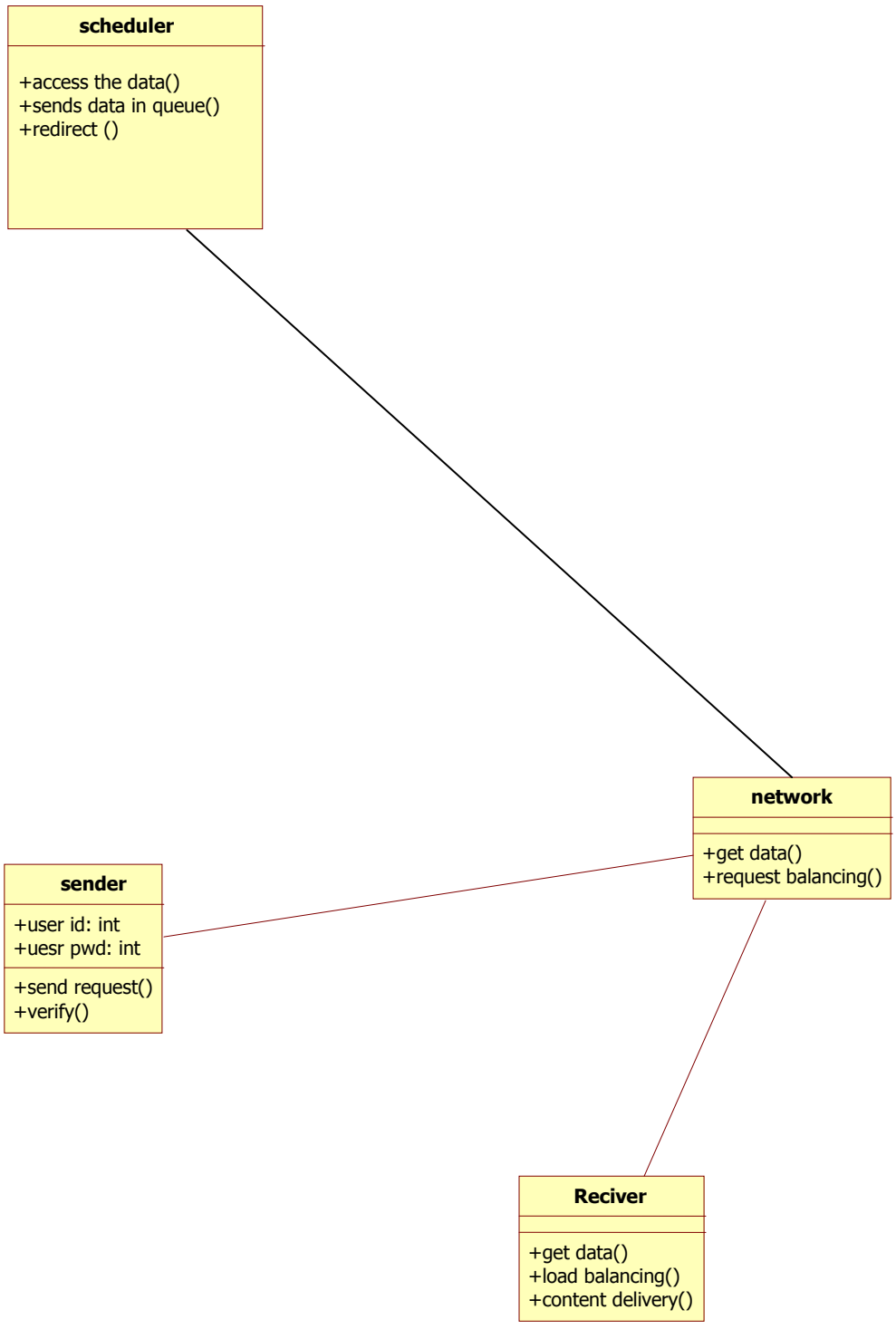
Data flow diagram



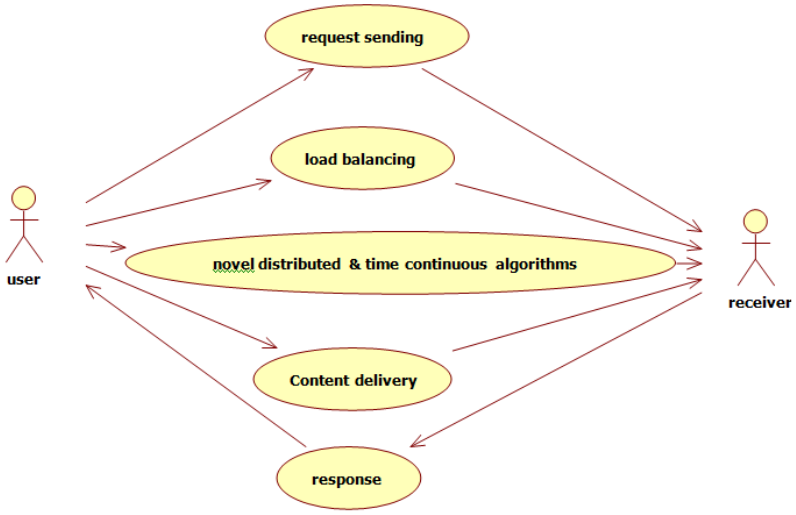
ENTITY RELATIONSHIP MODELS



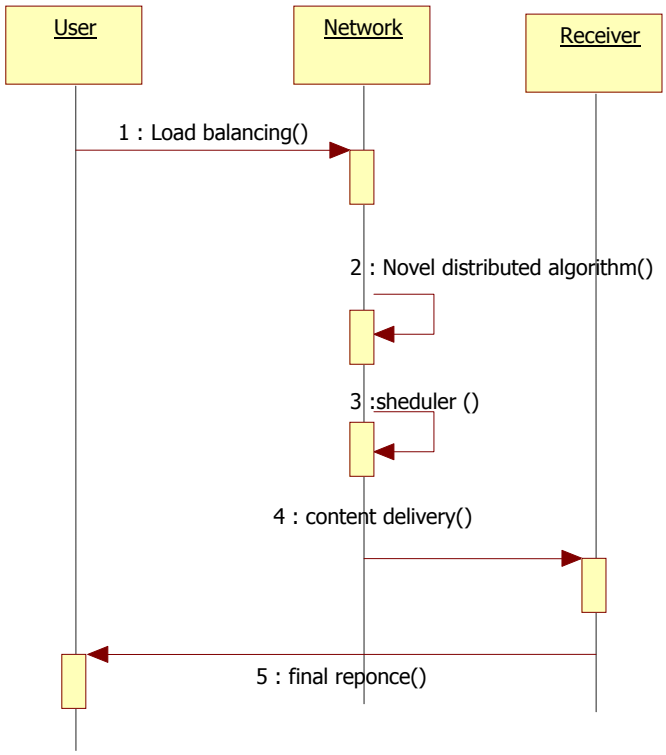
CLASS DIAGRAM



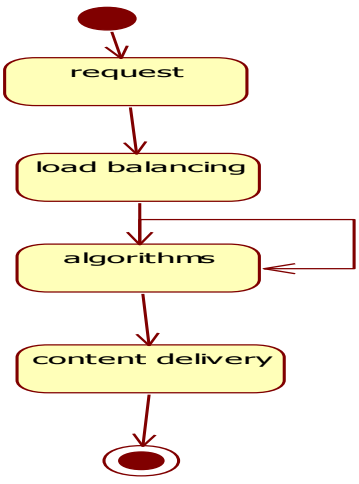
USE CASE DIAGRAM



SEQUENCE DIAGRAMS



STATE CHART DIAGRAMS



V. CONCLUSION

In this paper, we presented a novel load-balancing law for cooperative CDN networks. We first defined a model of such networks based on a fluid flow characterization. We hence moved to the definition of an algorithm that aims at achieving load balancing in the network by removing local queue instability conditions through redistribution of potential excess traffic to the set of neighbors of the congested server. The algorithm is first introduced in its time-continuous formulation and then put in a discrete version specifically conceived for its actual implementation and deployment in an operational scenario. Through the help of simulations, we demonstrated both the scalability and the effectiveness of our proposal, which outperforms most of the potential alternatives that have been proposed in the past. The present work represents for us a first step toward the realization of a complete solution for load balancing in a cooperative, distributed environment. Our future work will be devoted to the actual implementation of our solution in a real system, so to arrive at a first prototype of a load-balanced, cooperative CDN network to be used both as a proof-of-concept implementation of the results obtained through simulations and as a playground for further research in the more generic field of content-centric network management.

ACKNOWLEDGMENT

I would like to express my sincere thanks to my Guide and my Co-Authors for their consistence support and valuable suggestions.

REFERENCES

[1] M. Arlitt and T. Jin, *A Workload Characterization Study of 1998 World Cup Web Site*, IEEE Network, pp. 30-37, May/June 2000.
[2] J. Dilley, B. Maggs, J. Parikh, H. Prokop, R. Sitaraman and B. Weihl, *Globally Distributed Content Delivery*, IEEE Internet Computing, pp. 50-58, September/October 2002.
[3] M. Zukerman, T. D. Neame and R. G. Addie, *Internet Traffic Modeling and Future Technology Implications*, IEEE INFOCOM, 2003.
[4] Akamai Technologies, Inc. www.akamai.com, 2006
[5] F. Douglis, M. F. Kaashoek, *Scalable Internet Services*, IEEE Internet Computing, vol. 5, no. 4, 2001, pp.36-37.
[6] G. Pallis and A. Vakali, *Insight and Perspectives for Content Delivery Networks*, Communications of the ACM, vol. 49, no. 1, ACM Press, NY, USA, January 2006. pp. 101-106.
[7] S. Jamin, C. Jin, Y. Jin, D. Raz, Y. Shavitt and L. Zhang, *On the placement of Internet Instrumentation*, In Proceedings of IEEE INFOCOM conference, pp. 295-304, Tel-Aviv, Israel, March 2000.
[8] P. Krishnan, D. Raz, Y. Shavitt, *The Cache Location Problem*, IEEE/ACM Transaction on Networking, 8(5), 2000.
[9] L. Qiu, V. N. Padmanabhan, G. M. Voelker, *On the Placement of Web Server Replicas*, In Proceedings of IEEE INFOCOM conference, pp. 1587-1596, Anchorage, Alaska, USA, April 2001.
[10] S. Jamin, C. Jin, A. R. Kure, D. Raz and Y. Shavitt, *Constrained Mirror Placement on the Internet*, In Proceedings of IEEE INFOCOM Conference, Anchorage, Alaska, USA, April 2001.
[11] Y. Chen, R. H. Katz and J. D. Kubiatowicz, *Dynamic Replica Placement for Scalable Content Delivery*, In Proceedings of International Workshop on Peer-to-Peer Systems (IPTPS 02), LNCS 2429, Springer-Verlag, pp. 306-318, 2002.
[12] A. Vakali and G. Pallis, *Content Delivery Networks: Status and Trends*, IEEE Internet Computing, IEEE Computer Society, pp. 68-74, November-December 2003.
[13] N. Fujita, Y. Ishikawa, A. Iwata and R. Izmailov, *Coarse-grain Replica Management Strategies for Dynamic Replication of Web Contents*, Computer Networks: The International Journal of Computer and Telecommunications Networking, vol. 45, issue 1, pp. 19-34, May 2004.
[14] G. Peng, *CDN: Content Distribution Network*, Technical Report TR-125, Experimental Computer Systems Lab, Department of Computer Science, State University of New York, Stony Brook, NY 2003. [20] J. Kangasharju, J Roberts and K. W. Ross, *Object Replication Strategies in Content Distribution Networks*, Computer Communications 25(4), pp. 367-383, March 2002.
[15] M. Day, B. Cain, G. Tomlinson and P. Rzewski, *A Model for Content Internetworking (CDI)*, Internet Engineering Task Force RFC 3466, February 2003.
[16] W. Y. Ma, B. Shen and J. T. Brassil, *Content Services Network: Architecture and Protocols*, In Proceedings of 6th International Workshop on Web Caching and Content Distribution(IWCW6), 2001.
[17] M. Hofmann and L. R. Beaumont, *Content Networking: Architecture, Protocols, and Practice*, Morgan Kaufmann Publishers, San Francisco, CA, USA, pp. 129-134, 2005.