

# Load Balancing in CDN using Cooperative Algorithm

M.SANDHINI<sup>1</sup>, S. KOWSALYA<sup>2</sup>, S. THAMIZHARASAN<sup>3</sup>, D.VINOTH<sup>4</sup>, K. GUGAN<sup>5</sup>, S.SARAVANAN<sup>6</sup>

M.Tech Student , CSE Dept, Rajiv Gandhi College of Engineering and Technology, Puducherry, India<sup>1,2</sup>

Asst. Professor , MCA Dept, Rajiv Gandhi College of Engineering and Technology, Puducherry, , India<sup>3</sup>

Asst. Professor, CSE Dept, Rajiv Gandhi College of Engineering and Technology, Puducherry, , India<sup>4,5</sup>

Asst. Professor (SL.G), CSE Dept, Rajiv Gandhi College of Engineering and Technology, Puducherry, , India<sup>6</sup>

**Abstract:** The challenging issue of defining and implementing an effective law for load balancing in Content Delivery Networks (CDNs). The proposal on a formal study of a CDN system, carried out through the exploitation of a fluid flow model characterization of the network of servers. Starting from such characterization, it derives and proves a lemma about the network queues equilibrium. The 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. The discrete formulation of the proposed balancing law is eventually discussed in terms of its actual implementation in a real-world scenario. Finally, the overall approach is validated by means of simulations.

**Keywords:** Load balancing, Content Delivery Network, Cooperative Algorithm

## I. INTRODUCTION

Networking is the word basically relating to computers and their connectivity. It is very often used in the world of computers and their use in different connections. The term networking implies the link between two or more computers and their devices, with the vital purpose of sharing the data stored in the computers, with each other. The networks between the computing devices are very common these days due to the launch of various hardware and computer software which aid in making the activity much more convenient to build and use. Structure of networking between the different Computers as shown in Figure 1.

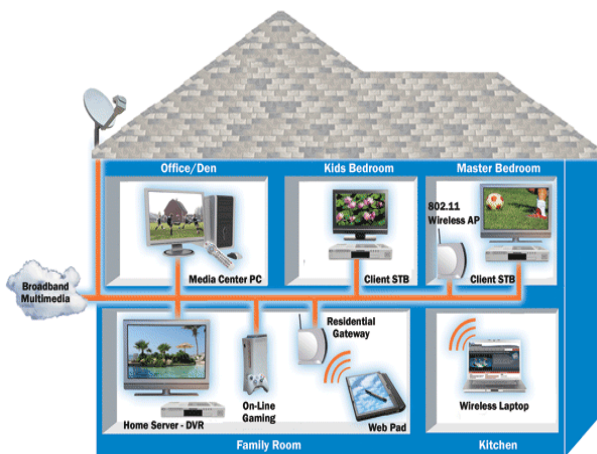


Figure 1 Structure of networking between the different Computers

## II LITERATURE SURVEY

From Reference [1] The challenging issue of defining and implementing an effective law for load balancing in Content Delivery Networks. The formal study of a CDN system, carried out through the exploitation of a fluid flow model characterization of the network of servers. Starting

from such characterization, it can derive and prove a lemma about the network queues equilibrium. This result is then leveraged in order to devise a novel distributed algorithm for load balancing. The overall approach is validated by means of simulations showing the effectiveness of the proposed algorithm in terms of both fair load distribution and limited service time from reference 1.

From reference [2] with the rapid development of network applications, the Internet has evolved from a content-based communication infrastructure to a social-based community network. The emerging applications require the Internet to preserve not only the existing advantages of simplicity and scalability, but also demand varying amounts of capability, availability, reliability, flexibility, and differentiated quality of service. Therefore, it is of paramount importance to bridge the gap between these applications and the IP networks which were originally designed and developed for supporting one-size-fits-all functionality. An efficient solution is to build a virtual network on top of a generic IP transport layer in order to provide additional functionality and flexibility. The content delivery networks technique is one of the successful virtual networks rapidly developed over the last decade with the specific advantage of optimizing the Internet. Nowadays, the CDN has become one of the most important parts of the Internet architecture for content distribution. In this article we highlight the innovative technologies in CDNs and present their evolution triggered by ever newer emerging applications. By presenting in-depth discussion about the architecture, challenges, and applications of CDNs, we demonstrate their importance for the future Internet.

From reference [3] Improvement of Multiplayer Online Game (MOG) performance has been addressed by different means. The number of simultaneous online users,

the variable latency of the network owing to overload in the servers and the bandwidth required to ensure a good quality-of-service are the principal problems in MOG. This paper proposes an architecture that could improve the performance of previous proposals, using a Content Distribution Network (CDN) infrastructure. The core element in a CDN is the redirection algorithm which will send the players to the most adequate server according to network conditions. CDN provides a reduction in latency and at the same time a load balancing mechanism among players.

From Reference [4] the problem of balancing load in a large-scale distributed system when information about server load may be stale. It is well known to sending each request to the machine with the apparent lowest load can behave badly in such systems, yet this technique is common in practice. Other systems use round robin or random selection algorithms that entirely ignore load information or that only use a small subset of the load information.

From reference [5] describes a prototype scalable and highly available web server, built on an IBM SP-2 system, and analyze its scalability. The system architecture consists of a set of logical front-end or network nodes and a set of back-end or data nodes connected by a switch, and a load balancing component. A combination of TCP routing and Domain Name Server (DNS) techniques are used to balance the load across the Front-end nodes that run the Web (httpd) daemons. The scalability achieved is quantified and compared with that of the known DNS technique. The load on the back-end nodes is balanced by striping the data objects across the back-end nodes and disks. High availability is provided by detecting node or daemon failures and reconfiguring the system appropriately. The scalable and highly available web server is combined with parallel databases, and other back-end servers, to provide integrated scalable and highly available solutions.

From reference [6] The Web is becoming the standard interface for accessing remote services of information systems, hosting data centers and application service providers. Demands placed on Web-based services continue to grow and Web-server systems are becoming more stressed than ever. The performance problems of Web-based architecture will even worsen because of the proliferation of heterogeneous client devices, the need of client authentication and system security, the increased complexity of middleware and application software, and the high availability requirements of corporate data centers and e-commerce Web sites. Because of the complexity of the Web infrastructure, performance problems may arise in many points during an interaction with Web-based systems. For instance, they may occur in the network because of congested Internet routers, as well as at the Web and database server, either because of under provisioned capacity or unexpected surge of requests. Although, in recent years, both network and server capacity have improved and new architectural solutions have been deployed, response time continues to challenge Web-system-related research.

From reference [7] introduce a control theoretical analysis of the closed-loop congestion control problem in packet networks. The control theoretical approach is used in a proportional rate controller, where packets are admitted into the network in accordance with network buffer occupancy. A Smith Predictor is used to deal with large propagation delays, common to high speed backbone networks. The analytical approach leads to accurate predictions regarding both transients as well as steady-state behavior of buffers and input rates. Moreover, it exposes tradeoffs regarding buffer dimensioning, packet loss, and throughput.

From reference [8] the goal of this paper is to present a system able to support the network delivery of live events in an expected, future Digital Cinema scenario. This service can consume a large amount of network bandwidth, due to the large volume of transmitted data and to the number of receivers, thus multicast transmission proves to be very useful. The system is the request routing algorithm, the goal of which is to optimize the QoS-guaranteed delivery of live streams in the backbone, each one towards a set of theatres. We consider the Multi Protocol Label Switching, which has emerged as an elegant solution to meet traffic engineering and resource reservation requirements, and focus on the overall request routing procedure, the mathematical modeling of the problem, and relevant solving algorithms. We present the comparative performance evaluation of these algorithms by means of an extensive simulation campaign performed with the OMNET++ simulation platform.

From reference [9] use a previously developed nonlinear dynamic model of TCP to analyze and design Active Queue Management (AQM) control systems using RED. First, we liberalize the interconnection of TCP and a bottlenecked queue and discuss its feedback properties in terms of network parameters such as link capacity, load and round-trip time. Using this model, we next design an AQM control system using the random early detection (RED) scheme by relating its free parameters such as the low-pass filter break point and loss probability profile to the network parameters. We present guidelines for designing linearly stable systems subject to network parameters like propagation delay and load level. Robustness to variations in system loads is a prime objective. We present ns simulations to support our analysis.

From Reference [10] defines active Queue Management (AQM) is the process of signaling TCP sources from core routers with the objective of managing queue utilization and delay. It is essentially a feedback control problem. Based on a recently developed dynamic model of TCP's congestion-avoidance mode, this paper does three things. First, it relates key network parameters such as the number of TCP sessions, link capacity and round-trip time to the underlying feedback control problem. Secondly, it analyzes the present de facto AQM standard: random early detection (RED) and determines that RED's queue-averaging is not beneficial. Finally, it recommends alternative AQM schemes which amount to classical proportional and proportional-integral control. We

illustrate our results using ns simulations and demonstrate the practical impact of proportional-integral control on managing queue utilization and delay.

In existing system in a queue-adjustment strategy, the scheduler is located after the queue and just before the server. The scheduler might assign the request pulled out from the queue to either the local server or a remote server depending on the status of the system queues. In a rate-adjustment model strategy, instead the scheduler is located just before the local queue. Upon arrival of a new request, the scheduler decides whether to assign it to the local queue or send it to a remote server. In a hybrid-adjustment strategy for load balancing, the scheduler is allowed to control both the incoming request rate at a node and the local queue length. Thus in Existing systems, upon arrival of a new request, indeed, a CDN server can either elaborate locally the request or redirect it to other servers according to a certain decision rule, which is based on the state information exchanged by the servers. Such an approach limits state exchanging overhead to just local servers.[11]

Disadvantage of existing system A critical component of CDN architecture is the request routing mechanism. It allows to direct users' requests for content to the appropriate server based on a specified set of parameters. The proximity principle, by means of which a request is always served by the server that is closest to the client, can sometimes fail. The routing process associated with a request might take into account several parameters (like traffic load, bandwidth, and servers' computational capabilities) in order to provide the best performance in terms of time of service, delay, etc. Furthermore, an effective request routing mechanism should be able to face temporary, and potentially localized, high request rates (the so-called flash crowds) in order to avoid affecting the quality of service perceived by other users.

In proposed system defines in a similar way, 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. The most notable implementation issues associated with the proposed load-balancing strategy. A new mechanism for redirecting incoming client request to the most appropriate server, thus balancing the overall system requests load. The mechanism leverages local balancing in order to achieve global balancing. This is carried out through a periodic interaction among the system nodes. Advantages of proposed system tells The proposed mechanism also exhibits an excellent average Response Time, which is only comparable to the value obtained by the 2RC algorithm. The excellent performance of our mechanism might be paid in terms of a significant number of redirections. Since the redirection process is common to all the algorithms analyzed, and then evaluate the percentage of requests redirected more than once over the total number of requests generated.

### III ALGORITHM DESCRIPTION

#### A. Control Law Balancing (CLB)

CLB for each algorithm evaluated each server's queue length behavior over time, together with the average value

among all servers. A parameter represents an excellent indicator of the request distribution degree achieved by the CDN. Another important parameter is the Response Time (RT), which evaluates the efficiency of the algorithm in terms of end-user's satisfaction. For such a parameter can be evaluated both the average value and the standard deviation. Also introduce an Unbalancing Index to estimate the capability of the algorithms to effectively balance requests among the available servers.

CLB is the best algorithm in terms of capability to recover from the overload situation due to the presence of excess traffic generated during the flash-crowd interval. However, this is achieved by the CLB with a more fair balancing among the available servers, as it is further confirmed by the analysis of the unbalancing index. The capability of the algorithms to effectively balance requests among the available servers again the low degree of unbalancing exhibited by our solution with respect to the evaluated counterparts. Such a result confirms that the algorithm provides an optimized balancing mechanism.

#### B. Cooperative Algorithm

Nearest Neighbour Protocol (NNP) Scheme

- Selects the nearest neighbor from the source as the relay node
- Considers only the uplink (toward the BS/AP) Neighbour discovery
- Each relay sends out "Hello" message
- Each source node can know its distance to the BS/AP using TOA (Time of Arrival)

The cooperative algorithm as shown in Figure 2.

#### C. CDN DESIGN

Architecture of LANC-based CDN, how it processes requests, and how requests are redirected between servers. Distinguish between the content server, the client and the origin web server. Clients generate requests and forward them to content servers. Servers receive requests and handle them by delivering the requested content, potentially fetching an authoritative copy of the content from the origin web server. Content servers have three components:

- The cache manager provides an interface to store and retrieve content from the local disk. It also defines the cache replacement strategy (e.g., LRU or LFU) when disk space is low and content must be discarded.
- The server HTTP proxy is the point of entry for HTTP requests by providing a proxy interface. For each request, the proxy decides to:
  - (a) Retrieve content locally if the cache manager indicates that the requested content exists in the local cache;
  - (b) Request content from a nearby server with the help of the server coordinate manager (or) Return content from the origin web server to the requester, while caching it locally for future access.
- The server coordinate manager maintains the

LANC. It takes application-level delay measurements to random other servers and clients and updates the LANC accordingly. It also maintains a routing table of neighbors that is used for mapping clients to servers. Clients generate HTTP requests with a regular web browser. To redirect requests to content servers, clients run two components as a separate process

- The client HTTP proxy provides a local proxy. The local browser is configured to send all HTTP requests through this proxy. The proxy interacts with the client coordinate manager to redirect requests to content servers.
- The client coordinate manager is similar to the one on the server-side. It makes delay measurements to maintain a LANC. It also manages a fixed-sized neighbor set with nearby content servers used for mapping requests to servers. The co-operative algorithm Structure as shown in Figure 2.

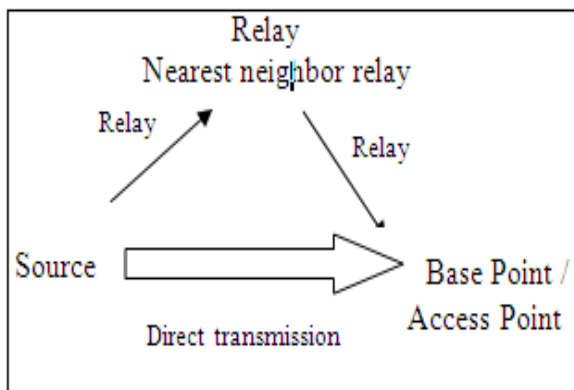


Figure 2. Co-operative algorithm structure

#### D. Request Mapping

Clients map requests to the content server with the closest LANC compared to their own. Each client coordinate manager must dynamically keep track of a small set of nearby servers. For this, we use a geometric routing approach in the coordinate space created by the LANCs of all servers.

#### E. Request Redirection

LANC-based CDN is only populated with new content when a server fetches content from the origin web server after a cache miss (local-only redirection). We propose two simple coordinate-based techniques for cooperation between content servers: Server-centric redirection. When a primary content server receives a request that it cannot satisfy with its local cache, it forwards the request to a set of secondary servers in parallel. The secondary servers are neighbors in its geometric routing table within a distance  $d$  in the LANC space. The secondary servers then.

(a) Return the requested content to the primary server if available. The primary server then forwards the content to the client while caching it; or.

(b) responds with cache misses. If all secondary servers had cache misses, the primary server fetches the content from the origin web server. A low value of  $d$  ensures that retrieving content from secondary servers gives better performance than fetching it directly from the origin web server.

#### F. Client-centric redirection

A client sends requests in parallel to nearby servers in the LANC space. Fundamentally, server-centric redirection spreads popular content more effectively in the CDN because primary servers retrieve it from secondary servers without involving origin web servers. Therefore evaluation in Section is server-centric redirection.

### IV. SOFTWARE DESCRIPTION

#### A. The network Simulator 2.33(NS2)

Network Simulator (NS2) is a discrete event driven simulator developed at UC Berkeley. Versions are available for FreeBSD, Linux, Solaris, Windows and Mac OS X.

#### B. Structure of NS2

NS2 is built using object oriented methods in C++ and OTcl (object oriented variant of Tcl. NS2 interprets the simulation scripts written in OTcl. The Simplified User's View Of NS2 as shown in Figure 3

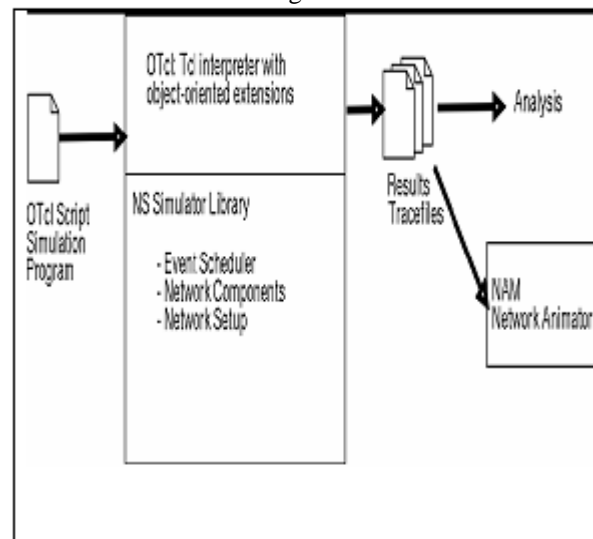


Figure 3. Simplified User's View Of NS2

#### C. Functionalities NS 2.33

Functionalities for wired, wireless networks, tracing, and visualization are available in NS2. Support for the wired world include

- Routing DV, LS, and PIM-SM.
- Transport protocols: TCP and UDP for unicast and SRM for multicast.
- Traffic sources: web, ftp, telnet, CBR (constant bit rate), stochastic, real audio.
- Different types of Queues: drop-tail, RED, FQ, SFQ, DRR.
- Quality of Service: Integrated Services and Differentiated Services.
- Emulation.

Support for the wireless world include

- Ad hoc routing with different protocols, e.g. AODV, DSR, DSDV, TORA.
- Wired-cum-wireless networks.



- Mobile IP.
- Directed diffusion.
- Satellite.
- Senso-MAC.
- Multiple propagation models (Free space, two-ray ground, shadowing).
- Energy models

**Tracing Visualization**

- Network Animator (NAM).
- Trace Graph.

**Utilities**

- Mobile Movement Generator.

**D. Mobile network in NS2.33**

The wireless model that was originally ported as CMU's Monarch group's mobility extension to NS2. The first section covers the original mobility model ported from CMU/Monarch group. The internals of a mobile node, routing mechanisms and network components that are used to construct the network stack for a mobile node.

**E. The basic wireless model in NS**

The wireless model essentially consists of the Mobile Node at the core, with additional supporting features that allows simulations of multi-hop ad-hoc networks, wireless LANs etc. In this section we shall describe the internals of Mobile Node, its routing mechanisms, the routing protocols dsdv, aodv, tora and dsr, creation of network stack allowing channel access in Mobile Node, brief description of each stack component, trace support and movement/traffic scenario generation for wireless simulations.

**F. Mobile Node for creating wireless Topology**

Mobile Node is the basic *ns* Node object with added functionalities like movement, ability to transmit and receive on a channel that allows it to be used to create mobile, wireless simulation environments. The class Mobile Node is derived from the base class Node. Mobile Node is a split object. The mobility features including node movement, periodic position updates, maintaining topology boundary etc are implemented in C++ while plumbing of network components within Mobile Node itself (like classifiers, dmux, LL, Mac, Channel etc) have been implemented in Otc.

**G. Implementation of environment**

Network simulator2 is used as the simulation tool in this project. NS was chosen as the simulator partly because of the range of features it provides and partly because it has an open source code that can be modified and extended. There are different versions of NS and the latest version is ns-2.1b9a while ns-2.1b10 is under development. Network simulator (NS) is an object-oriented, discrete event simulator for networking research. NS provides substantial support for simulation of TCP, routing and multicast protocols over wired and wireless networks.

One of the advantages of this split-language program approach is that it allows for fast generation of large scenarios. To simply use the simulator, it is sufficient to know OTcl. On the other hand, one disadvantage is that modifying and extending the simulator requires programming and debugging in both languages. NS can simulate the following:

- Topology: Wired, wireless.
- Scheduling Algorithms: RED, Drop Tail.
- Transport Protocols: TCP, UDP.
- Routing: Static and dynamic routing.
- Application: FTP, HTTP, Telnet, Traffic generators.

The Block diagram of architecture of NS2as shown in Figure 4

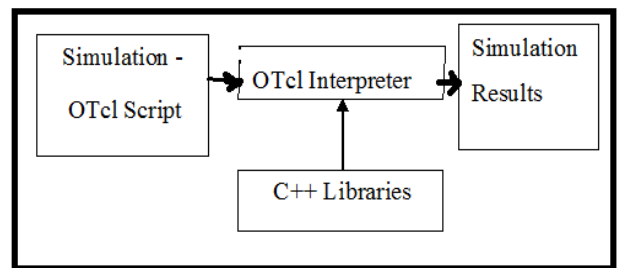


Figure 4. Block diagram of architecture of NS2

**G. Network components**

The NS components, mostly compound network components. A partial OTcl class hierarchy of NS, which will help understanding the basic network components. The root of the hierarchy is the Tcl object class that is the super class of all OTcl library objects (scheduler, network components, timers and the other objects including NAM related ones). The network and objects that have only one output DATA path are under the Connector class, and switching objects that are possible multiple output DATA paths are under the Classifier class.

**V. EXPERIMENTAL RESULTS AND DISCUSSION**

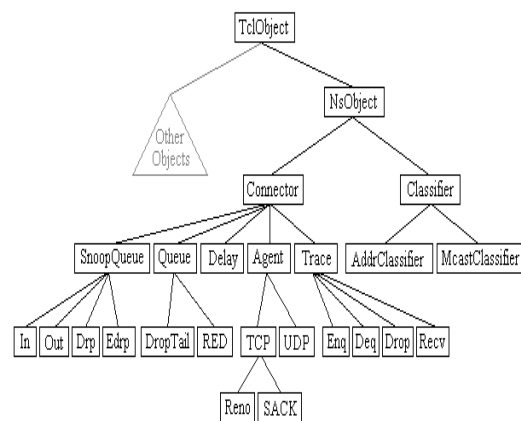


Figure 6 Getting Request From Client

load is getting from the client and then the server have receive the request as shown in Figure 6. The Load sharing servers as shown in Figure 7.

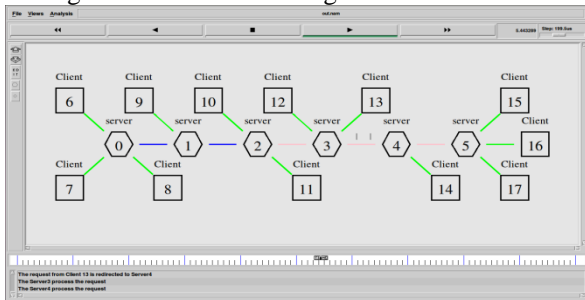


Figure 7 Load Sharing Among Servers

Getting the load from the client and give it to the server. If traffic occurs, the load is shared among the neighbouring servers. The Load display as shown in Figure 8.

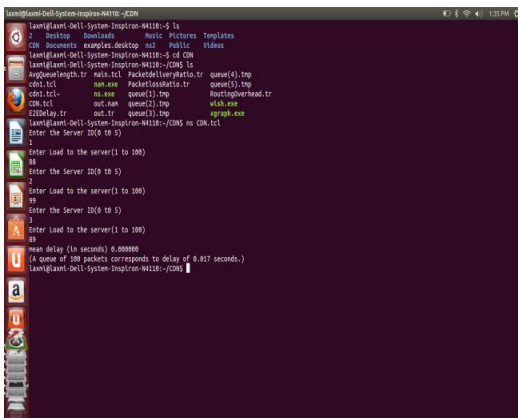


Figure 8 Load is display in server

In these terminal, it identifies which server has to be given the load.

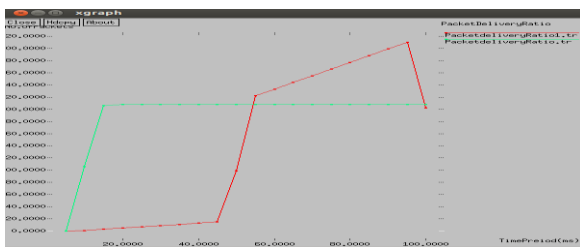


Figure 8 Comparison of Existing and Proposed System Packet Delivery Ratio. In packet delivery ratio, it is the time period to the number of packet arrived in bits per milliseconds as shown in Figure 8. System end to end delay as shown in Figure 9. And average Queue length as shown in Figure 10.

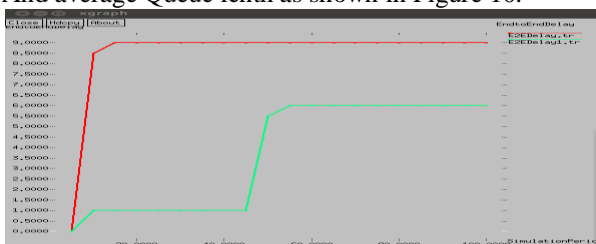


Figure 9 Comparison Of existing and propose system End to End Delay

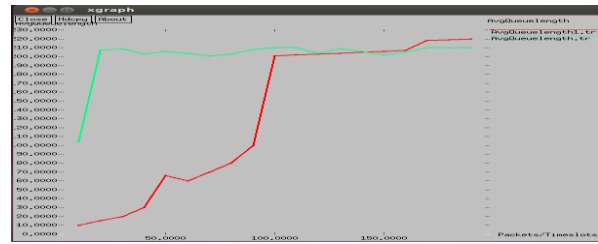


Figure 10 Comparison of Existing And Proposed Graph for Average Queue Length

In end to end delay, the client and server response time.

## V. CONCLUSION AND FUTURE ENHANCEMENT

Load-balancing law for cooperative algorithm in CDN networks can defined a model for such networks based on a fluid flow characterization. Algorithm that define to 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, both the scalability and the effectiveness of our proposal, which performs most of the potential alternatives that have been proposed in the past. The present work represents the first step towards the realization of a complete solution for load balancing in a cooperative, distributed environment. The existing load balancing technology in the content distribution network (CDN) only emphasizes even load distribution among servers, and it does not make the best of network topological information and the historical information of file accessed, so user request can not get timely response. To solve the above problem, the paper presents a load balancing a algorithm based on the Distributed binning strategy. The algorithm can make full use of network topological information and file access history as well as server load information, analyzing the popularity of files with the access history of the clusters of clients from the servers, efficiently finishing distribution and routing of the high popularity files among servers so that users can closely get their requested content, relieve the congestion situation in the Internet and then enhance the speed of response when they accessing the websites.

## REFERENCES

- [1] S. M. Metev and V. P. Veiko, Laser Assisted Microtechnology, 2nd ed., R. M. Osgood, Jr., Ed. Berlin, Germany: Springer-Verlag, 1998.
- [2] J. Breckling, Ed., The Analysis of Directional Time Series: Applications to Wind Speed and Direction, ser. Lecture Notes in Statistics. Berlin, Germany: Springer, 1989, vol. 61.
- [2] S. Zhang, C. Zhu, J. K. O. Sin, and P. K. T. Mok, "A novel ultrathin elevated channel low-temperature poly-Si TFT," IEEE Electron Device Lett., vol. 20, pp. 569–571, Nov. 1999.
- [3] M. Wegmuller, J. P. von der Weid, P. Oberson, and N. Gisin, "High resolution fiber distributed measurements with coherent OFDR," in Proc. ECOC'00, 2000, paper 11.3.4, p. 109.
- [4] R. E. Sorace, V. S. Reinhardt, and S. A. Vaughn, "High-speed digital-to-RF converter," U.S. Patent 5 668 842, Sept. 16, 1997.

- [5] (2002) The IEEE website. [Online]. Available: <http://www.ieee.org/>
- [6] M. Shell. (2002) IEEEtran homepage on CTAN. [Online]. Available: <http://www.ctan.org/tex-archive/macros/latex/contrib/supported/IEEEtran/>
- [7] FLEXChip Signal Processor (MC68175/D), Motorola, 1996.
- [8] "PDCA12-70 data sheet," Opto Speed SA, Mezzovico, Switzerland.
- [9] A. Karnik, "Performance of TCP congestion control with rate feedback: TCP/ABR and rate adaptive TCP/IP," M. Eng. thesis, Indian Institute of Science, Bangalore, India, Jan. 1999.
- [10] J. Padhye, V. Firoiu, and D. Towsley, "A stochastic model of TCP Reno congestion avoidance and control," Univ. of Massachusetts, Amherst, MA, CMPSCI Tech. Rep. 99-02, 1999.
- [11] Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification, IEEE Std. 802.11, 1997.

## BIOGRAPHIES



**Ms.M. Sandhini** has received B.Tech (I.T) in Pondicherry University. She is student of M.Tech (Computer Science Engineering ) from 2014 to 2016 batch in Rajiv Gandhi college of Engineering and Technology, Puducherry, India. She has interested areas in computer Networking, Operating system ect.



**Mr.S.Thamizharasan**, Assistant Professor, Department of Computer Applications in Rajiv Gandhi College of Engineering and Technology. He holds M.S in Software Engineering and pursuing his Ph.D in Mobile Technologies from Manonmaniam Sundaranar University, India. His areas of Interest are Mobile Technologies and Wireless Sensor Networks.



**Mr. D.VINOTH** is received B.Tech (I.T) in 2011 at Christ College of Engineering & Technology, Pondicherry and M.Tech (I.T) in 2013 at Anna University, Regional Centre, Coimbatore. He is working as Assistant Professor in computer science Department at Rajiv Gandhi college of Engineering and Technology, Puducherry from January 2014. He has interested in computer communication network and mobile ad hoc network.



**Mr. K. GUGAN** has received B.Tech (I.T), M.Tech (CSE) in Pondicherry University, Puducherry. He has interest in areas of Image processing, Grid computing, cloud computing, Data Mining. Telemedicine network. He has done Data Mining Project in UG level and Image Processing Project in PG level



**Mr..S.SARAVANAN** Assistant Professor (Selection Grade) in computer science Department in Rajiv Gandhi college of Engineering and technology, puducherry, India. He has completed B.E( Electronics and communication) in 1998, M.S (Information technology) in 2003, M.E ( computer science Engineering) in 2008, MBA(Education management)in 2010, M.Tech ( Communication System) in 2012, Ph.D pursuing from 2009. He has teaching experience from 1998 in various Department of Electronics and communication, Biomedical Engineering and Computer science Engineering