

Evaluating the Performance of Equitable Dominating based Content Distribution Network Design

Amutharaj Joyson and Radhakrishnan Shanmugasundaram
Department of Computer Science and Engineering
Faculty of Arulmigu Kalasalingam College of Engineering
Anandnagar, Krishnankoil, Srivilliputtur (Via), Tamil Nadu, India

Abstract: Problem statement: In this study, we considered an efficient and resilient large file content push problem in a large scale distributed content delivery networks and investigated the Quality of Service (QoS) requirements for content distribution. We investigated the effect of equitable dominating set in SON formation and how it was useful in reducing the redundancy? **Approach:** At first, we constructed an equitable dominating set based semantic overlay network of surrogate servers to form the logical infrastructure of the CDN by choosing the optimal number of surrogate servers. Then we proposed a novel Efficient Fault Resilient Replica Algorithm (EFRRA) to replicate the content from the origin server to the dominant set of surrogate servers in an efficient and reliable manner. **Results:** We assessed the efficiency and resiliency of the proposed EFRRA algorithm by conducting simulation experiments and compared its performance with traditional content replication algorithms stated in the literature. We extended the simulation experiments to analyze the role of EDSON in maintaining uniform CDN utility of above 0.9. **Conclusion:** It also observed that equitable dominating set based SON was useful in keeping the average replication time stable and much more predictable. We also investigated the quality of service requirements for the content distribution and evaluated the performance of EDSON based CDN in terms of mean response time, mean CDN utility, latency and hit ratio percentage.

Key words: Content Distribution Network (CDN), Dominating set, EDSON, EFFRA, optimal fast replica, tornado codes, adjacent vertices, latency, replication algorithm, taxonomy, hit ratio percentage

INTRODUCTION

Content Distribution Network (CDN) consists of many surrogate servers located at different locations which can be clustered or grouped together to form a surrogate server site, so that a client has a good connectivity to at least one of the surrogate servers. These surrogate servers have to cooperate with each other to enhance the performance of the content delivery network and meet the user perceived Quality of Service (QoS). In this study, we use equitable dominating set for selecting the replication set of surrogate servers and apply EFRRA content replication algorithm to disseminate the content among the surrogate servers in the CDN. Performance measurement is carried out to estimate the values of performance metrics such as Mean Response Time, Latency and hit ratio percentage which gives an indication of system conditions and used to identify the factors that influence the design of CDN and its performance, assisting the content providers in decision making and achieve load balancing in large systems.

Pathan and Buyya (2007) presented a comprehensive taxonomy with a broad coverage of CDNs in terms of organizational structure, content distribution mechanisms, request redirection techniques and performance measurement methodologies. Their surveys focused on understanding the existing CDNs in terms of their infrastructure, request-routing mechanisms, content replication techniques, load balancing and cache management.

Geetha and Vasumathi (2011) presented a survey on current trends and methods in video retrieval which focus on research issues such as shot segmentation, key frame extraction, feature extraction, clustering, indexing and video retrieval-by similarity, probabilistic, transformational, refinement and relevance feedback. This study assisted the upcoming researchers in the field of video retrieval and facilitate them in know about the techniques and methods available for video retrieval.

Malarvizhi Nandagopal and Rhymend Uthariaraj, 2011, proposed Multi Criteria Resource Selection (MCRS) algorithm which considered multiple criteria

Corresponding Author: Amutharaj Joyson, Department of Computer Science and Engineering, Arulmigu Kalasalingam College of Engineering Anandnagar, Krishnankoil, Srivilliputtur (Via), Tamil Nadu, India Tel: 91-4563-289042/91- 98423-77635 Fax: 91-4563- 289322

such as processing power, workload and network bandwidth of the resource during resource selection.

Ramadoss and Rajkumar (2007) outlined the underlying XML Schema based content description structures of DMAR and proposed a quality metric, fidelity to evaluate the expressive power of the dance annotations.

Cherksova and Kee (2002) proposed Fast Replica algorithm to distribute the content, in which a user downloads different parts of the same file from different servers in parallel. Once all the parts of the file are received, the user reconstructs the original file by reassembling the different parts.

Lu *et al.* (2008) proposed a novel content push policy, called TRRR i.e., Tree-Round-Robin-Replica which yields an efficient and reliable solution for distributing large files in the content delivery networks environment. They carried out simulation experiments to verify TRRR algorithm in small scale and demonstrated that TRRR significantly reduces the file replication time as compared with traditional policies such as sequential unicast and multiple unicast.

Dominating sets have been used by Han and Jia (2005) in topology control for wireless Ad hoc networks. Ma *et al.* (2005) used Dominating sets for virtual backbone creation in sensor networks.

Liu *et al.* (2010) proposed a new method called ATISA (Approximation Two Independent Sets based Algorithm) for constructing CDS is proposed. The ATISA has three stages. The first stage is constructing a connected set CS (connected set) and the second stage is constructing a connected dominating set CDS and the third stage is pruning the redundant dominators of CDS.

Liu *et al.* (2010) presented a taxonomy and general classification of CDS construction algorithms. They formed virtual backbone by constructing a Connected Dominating Set (CDS). The CDS of a graph representing a network has a significant impact on an efficient design of routing algorithms in WSN. They also found that a good CDS should be small, additionally, it should have other characteristics such as robustness to node failures and low stretch.

Shakkottai and Johari (2010) proposed a hybrid content distribution system that combines the features of peer-to-peer and a centralized client-server content distribution system.

Pathan and Buyya (2009) presented architecture to support peering arrangements between CDNs, based on a Virtual Organization (VO) model. They presented a Quality of Service (QoS)-driven performance modeling approach for peering CDNs in order to predict the user perceived performance. Their approach has the provisions for an overloaded CDN to return to a normal

state by offloading excess requests to the peers and also providing concrete QoS guarantee for a CDN provider.

Amutharaj and Radhakrishnan (2008) constructed a dominating set based overlay network to optimize the number of servers for replication. They investigated the use of Fast Replica algorithm to reduce the content transfer time for replicating the content within the semantic overlay network and compared its performance with sequential unicast, multiple unicast content distribution strategies in terms of content replication time and delivery ratio.

Amutharaj and Radhakrishnan (2010) proposed EFRRA algorithm which combined the features of both fast replica and tornado coding algorithm. It provides an efficient and resilient content replication solution. They performed both analytical study and empirical study to analyze the performance of EFRRA and proved that EFRRA algorithm outperforms other algorithms in terms of replication time and also maintain the competent delivery ratio.

Amutharaj and Radhakrishnan (2010) proposed Equitable Dominating Set Based Semantic Overlay Network (EDSON) and applied optimal fast replica algorithm to disseminate the content among the surrogate servers in the EDSON. They investigated the use of EDSON in reducing redundancy.

MATERIALS AND METHODS

Design of equitable dominating set based semantic overlay network: Semantic Overlay Network 'G' can be defined as follows:

$$G = \{V, E\} \quad (1)$$

Where $V = \{V_1, V_2, V_3, \dots, V_n\}$ be the set of surrogate servers and E is the set of edges between i^{th} surrogate server and j^{th} surrogate server i.e., $E = (V_i, V_j)$ such that $V_i \neq V_j$. Let D be the dominating set of G and $D \subset G$, the server not in D is adjacent to at least one surrogate server in D . Hence, all the surrogate servers are either member of D or $V \setminus D$. Equitable Dominating set D is a set of 'r' dominating vertices in V since $|D| = r$ and $V \setminus D$ is the set of all the adjacent vertices of dominating server set D such that the difference between the degrees of all the vertices in D can differ utmost by 1. Each vertex v in D has more or less same number of neighbor nodes which are members of $V \setminus D$. So contents are only replicated in the set of surrogate servers D which contains 'r' surrogate servers or less than 'r' number of surrogate server's i.e., $|D| < |V|$.

Quitable Dominating Set based SON (EDSON) construction Algorithm:

- Step 1: Mark all the vertices of the graph white
- Step 2: Select the vertex with the maximal number of white neighbors
- Step 3: The selected vertex is marked black and its neighbors are marked gray
- Step 4: The algorithm then iteratively scans the gray nodes and their white neighbors and selects the gray node or the pair of nodes (a gray node and one of its white neighbors), whichever has the maximal number of white neighbors
- Step 5: The selected node or the selected pair of nodes is marked black, with their white neighbors marked gray
- Step 6: Once all the vertices are marked gray or black, the algorithm terminates. All the black nodes form a Connected Dominating Set (CDS)
- Step 7: After forming the CDS, check the degree of each vertices of the connected dominating set
- Step 8: If the degree of any vertex varies more than one then mark that vertex gray and find the suitable alternate vertex as the member of the dominating set and mark it black. If no alternate node is found then leave as it is

The equitable dominating set formation algorithm is applied to form the semantic overlay network of surrogate servers which are connected logically to provide the logical infrastructure of the CDN, in which any replication algorithm can replicate the content.

Working principle of EFRRA: A novel algorithm called EFRRA is proposed for an efficient and fault resilient replication of large files in the CDN. Working mechanism of EFRRA can be summarized as follows. In order to replicate a large file among ‘n’ nodes, the original file is partitioned into ‘n’ sub files of equal size and each sub file is transferred to a different node in the group. After that, each node propagates its sub file to the remaining nodes in the group. Thus instead of the typical replication of an entire file to ‘n’ nodes by using ‘n’ internet paths connecting the original node to the replication group, this replica algorithm exploits n*n diverse internet paths within the replication group where each path is used for transferring 1/nth of the file. Hence, the bandwidth requirement is reduced by a factor of 1/n.

Step 1: Distribution of content to surrogate servers: As shown in “Fig. 1,” the originator node N_0 opens n concurrent network connections to nodes $\{N_1...N_n\}$ and sends each recipient node N_i ($1 \leq i \leq n$) the following;

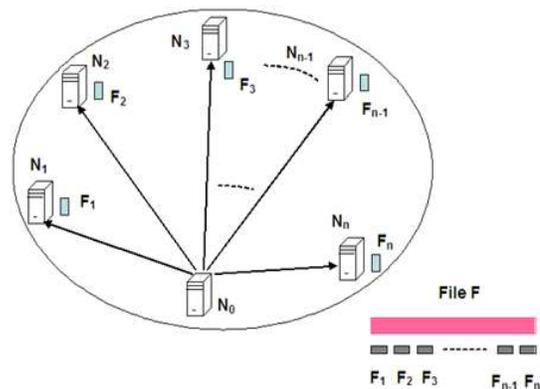


Fig. 1: Distribution step in EFRRA

- A distribution list of nodes $R = \{N_1, \dots, N_n\}$ to which sub file F_i has to be sent on the next step.
- Sub file F_i .

Step 2: Adding fault resiliency to EFRRA: It keeps the main structure of the EFRRA replication algorithm practically unchanged while adding the desired property of resilience to node failure.

In order to maintain the resiliency, the surrogate servers in the network are exchanging the heartbeat messages with their origin server. The heartbeat messages from surrogate servers to their origin server are augmented with additional information on the corresponding algorithm. Once the content is distributed in the network, the receiver has to recollect all the content from the network in a parallel manner. For example, if surrogate server N_1 fails during transfer, then this may impact all surrogate servers N_2, \dots, N_n in the network because each node depends on node N_1 to receive sub file F_1 . In the described scenario as shown in Fig. 2, surrogate server N_i is acting as a recipient server in the replication set. If a surrogate server fails when it acts as the origin surrogate server N_i , this failure impacts all the surrogate servers in the replication group which may be the replication sub tree rooted in surrogate server N_i .

Step 3: Adding resiliency during content collection at the receiver: Once the entire file is distributed to all the surrogate servers in the overlay network of surrogate servers using step1 then the recipient node or client node has to recollect all the sub files or blocks of the requested file from the overlay network of surrogate servers in a parallel manner.

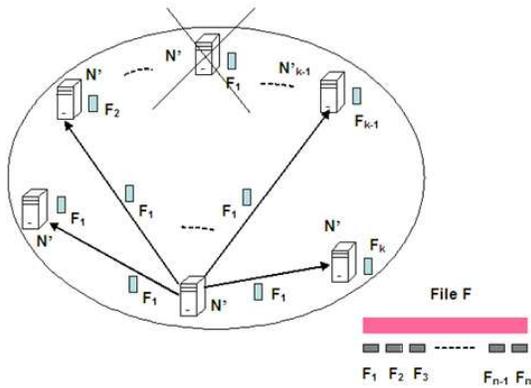


Fig. 2: Adding resiliency to EFRRA

Recipient node retrieves the original source file, in the form of a sequence of ‘k’ encoded packets, along with additional redundant packets, are transmitted by the sender and the redundant data can be used to recover lost source data at the receivers. Here retransmission of lost packets will not be needed. In this collection step also, EFRRA algorithm maintains resiliency against surrogate server failure and link outages.

In the ideal case, when $k = m$, every surrogate server N_i holds all of m sub-files of original file F and reorganizes them to form the Original file F in the local node. When the user requests file F from the origin server, the request will be redirected to one surrogate server in the list $\{N_1, N_2, \dots, N_m\}$ and download the whole file F .

Simulation test bed and performance measurement: We used the simulation tool CDNSim developed by K. Stamos *et al.*, (2010) to create and customize the simulation environment named EDSONCDN which includes the following five modules.

EDSON based CDN model: To evaluate the performance of the proposed EDSONCDN simulation environment developed using the simulation tool CDNSim, which simulates a main CDN infrastructure based on equitable dominating set and is implemented in the C programming language. In EDSON based CDN infrastructure where surrogate servers are logically connected based on equitable dominating semantic overlay network. So all the surrogate servers are either member of dominated semantic overlay network of surrogate servers or member of adjacent surrogate server set which is one hop connected with the EDSON. Each surrogate server maintains the neighbourhood information and knowledge about the file objects stored in all the other surrogate servers.

If user’s request is missed on a surrogate server which is not a member of EDSON then the content will

be searched on the adjacent surrogate server which is the member of EDSON and served. If the content is not available in the adjacent surrogate server then the content will be searched in the other surrogate server in the EDSON and served. If the content is not available in the entire EDSON then it is pulled from origin server. By default, CDNSim simulates a cooperative push-based CDN infrastructure, where each surrogate server has knowledge about what content (which has been proactively pushed to surrogate servers) is cached to all the other surrogate servers. If a user’s request is missed on a surrogate server, then it is served by another surrogate server. In this framework, the CDNSim simulates a CDN with 200 surrogate servers which have been located all over the world. The default size of each surrogate server has been defined as the 40 percent of the total bytes of the Web server content. Each surrogate server in CDNSim is configured to support 1,000 simultaneous connections.

Web server content generator: This Web server content generator module includes modelling the file object, its size and semantic characteristics such as type of content mentioning static or dynamic. Usually Web server content generator module creates two files. The first one is the graph and the second one records the produced communities.

Client request stream generator and network topology generator: This captures the main characteristics of Web users’ behaviour and built-in network topology generator to generate AS, Random, Transit_Stub and Waxman topologies. In this study, we have generated a maximum of 1 million users’ requests and each request is for a single object. We consider that the requests arrive according to a Poisson distribution with rate equal to 30. Then, the Web users’ requests are assigned to CDN’s surrogate servers taking into account the network proximity and the surrogate servers’ load, which is the typical way followed by CDNs’ providers. Finally, concerning the network topology, we used an AS-level Internet topology with a total of 3,037 nodes. This topology captures a realistic Internet topology by using BGP routing data collected from a set of seven geographically dispersed BGP peers.

Content distribution algorithm simulator: This Content Distribution Algorithm Simulator module is developed in OMNET++ to simulate the working of content replication algorithm. It collects the entire file object and its semantic information from the origin server, maintains the neighborhood information, decision making logic and disseminates the object

according to the content replication algorithms such as sequential unicast, multiple unicast, fast replica, Resilient Fast Replica and Optimal Fast Replica, Tornado Codes and EFRRRA.

Account manager: Account manager module is developed in the simulation test bed using OMNETT++, used to capture the traffic information at each and every moment and maintain the trace files and logs. These log information contains the number of file objects stored in the surrogate servers, number of blocks generated during block level replication, number of packets lost during transmission, number of redundant blocks generated and transmitted and time of initiation, time taken to reach the destination etc. These log information can be used by the account manager to compute the Quality of Service metrics such as net utility, mean surrogate server utilization, average content replication time, delivery ratio, reception efficiency, mean response time, latency and hit ratio percentage.

CDN network simulation setup: The distribution and arrangement of servers, routers and clients in the network affects the performance of the CDN (Table 1). Different network backbone types result in different “neighbourhoods” of the network elements. Therefore, the redirection of the requests and ultimately the distribution of the content are affected. In CDNSim simulation test bed, there are four different network backbone flavours: AS, Waxman, Transit_stub and Random. Each of them contains 3037, 1000, 1008 and 1000 routers respectively. The routers retransmit network packets using the TCP/IP protocol between the clients and the CDN. All the network phenomena such as bottlenecks and network delays, packet routing protocols, content distribution policies, EDSON formation mechanism are simulated.

RESULTS AND DISCUSSION

Analytical study: Let $Time_i$ denote the transfer time of file F from the origin server N_0 to surrogate server N_i as measured at N_i . Average replication time is considered as a performance measure to evaluate the performance of content replication algorithm.

Average replication time:

$$Time_{avg} = 1/n * \sum_{i=1}^{i=n} Time_i \quad (2)$$

In idealistic setting all the nodes and links are homogeneous and let each node can support ‘n’ network connections to other nodes at B bytes/sec. Then:

$$Time_{distribution} = Size (F) / (nxB) \quad (3)$$

$$Time_{collection} = Size (F) / (nxB) \quad (4)$$

Performance of content distribution algorithms in an ‘n’ server semantic overlay network: Time taken for distributing the content over the Semantic Overlay Network by different content distribution algorithms are presented in Table 2. Therefore, Replication Time proportion of different content distribution algorithms can be expressed as follows:

$$\begin{aligned} &Time_{SU} : Time_{MU} : Time_{FR} : Time_{R-FR} : Time_{O-FR} \\ &: Time_{TC} : Time_{FER} \\ &n : 1 : 2/n : (2+m/n) * 1/n : \\ &((k+n)/n * n * k) \\ &: 2 * c/n : 2+c/n \end{aligned} \quad (5)$$

Where:

- n = total number of surrogate servers in the replication set
- m = Number of surrogate servers in which replication of contents carried out
- k = Number of redundant blocks generated

Performance of content distribution algorithms in equitable dominating set based semantic overlay network: Equitable Dominating set D is a set of ‘r’ dominating surrogate servers in surrogate server set V and $V \setminus D$ is the set of all the adjacent vertices of dominating node set D such that the difference between degree of all the vertices in D can differ utmost by 1. Each vertex ‘v’ in V has more or less same number of neighbor nodes which are members of the adjacent servers set $V \setminus D$. So contents are only replicated in the equitable dominated set of surrogate servers D instead of V. Suppose Cardinality of D is ‘r’ or a value less than ‘r’ then the contents will be replicated in utmost ‘r’ number of surrogate servers which is always less than ‘n’. i.e., $|D| \leq |V|$.

Therefore, Replication Time proportion of different content distribution algorithms such as sequential unicast, multiple unicast, Fast Replica (FR), Resilient Fast Replica(R-FR) and Optimal Fast Replica (O-FR), Tornado Codes and EFRRRA in EDSON can be expressed as follows:

$$\begin{aligned} &r : 1 : 2/r : (2+m/r) * 1/r : ((k+r)/r * r * k) \\ &: 2 * c/r : 2+c/n \end{aligned} \quad (6)$$

Performance of different content distribution schemes in SON in terms of average replication time: We experimented with 12 different sized files;

Table 1: Technical specification of CDNSim simulation environment

Simulation parameters	Specification
Network topology	AS, WAXMAN, Transit_Stub, Random
Surrogate server	SON, DSON, EDSON
Cooperation mechanisms used	
Number of surrogate Servers	200
Number of clients	100000
Website size	1GB
Number of file objects in website	50000
Max. Number of requests	1000000
Link speed	1Gbps
QoS parameters measured at surrogate server	Average replication time, Delivery ratio, Reception efficiency, Encoding time and Decoding time
QoS Parameters measured at client side	Mean Response Time, Mean CDN Utility, Latency, Hit ratio percentage.

Table 2: Content distribution times of different content distribution algorithms

Algorithm	Content distribution time (T_D)
Sequential unicast	$n * \text{Size (F)} / B$
Multiple unicast	$\text{Size (F)} / B$
Fast replica	$2 * \text{Size (F)} / (nxB)$
Resilient fast replica without Node failure	$2 * \text{Size (F)} / (n * B)$
Resilient fast replica with Failure of 'm' servers	$(2+m/n) * \text{Size (F)} / (nxB)$
Optimal fast replica	$((k+n) / n * n * k) * \text{Size (F)} / B$
Tornado codes	$2 * c / n * \text{Size (F)} / B$
EFRRRA	$(2 + c) / n * \text{Size (F)} / B$

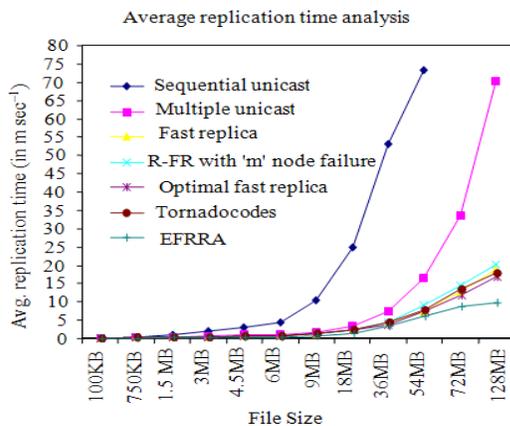


Fig. 3: Average content replication times for various schemes

100 KB, 750 KB, 1.5 MB, 3 MB, 4.5 MB, 6 MB, 7.5 MB, 9 MB, 36 MB, 54 MB, 72 MB, 128 MB in the SON based CDN infrastructure.

Figure 3 shows the average replication time measured by different individual surrogate servers for different file sizes of 100 and 750 KB, 1.5, 3, 4.5, 6, 7.5, 9, 36, 54, 72 and 128 MB when SON based replication set of surrogate servers. High variability of average replication time under Multiple and Sequential Multicast is identified for larger file sizes. Average content replication time of EFRRRA algorithm across large file sizes in SON based replication set is much more stable and predictable. Hence, EFRRRA algorithm outperforms all the traditional content distribution schemes.

Performance of EFRRRA algorithm during Surrogate Server Failure:

The delivery ratio is defined as the ratio of the number of data packets successfully received by the recipient surrogate server to the number of data packets sent by the source surrogate server. The worst case delivery ratio of different content distribution schemes such as sequential unicast, multiple unicast, fast replica(FR), Resilient fast replica(R-FR), Optimal Fast Replica(O-FR), Tornado Codes and EFRRRA Content Distribution algorithms when the number of simultaneous surrogate server failures in the CDN has been analyzed and its performance is shown in Fig. 4. From the delivery ratio analysis shown in Fig. 4, we found that the delivery ratio of EFRRRA algorithm is consistent during the surrogate server failure.

Delivery ratio of traditional algorithms such as Fast Replica (FR), Resilient Fast Replica (R-FR), Optimal Fast Replica (O-FR) and Tornado Codes degrades gracefully with respect to surrogate server failure. It is also observed that delivery ratio of Sequential Unicast and Multiple Unicast content distribution algorithms are degrades drastically with respect to surrogate server failure.

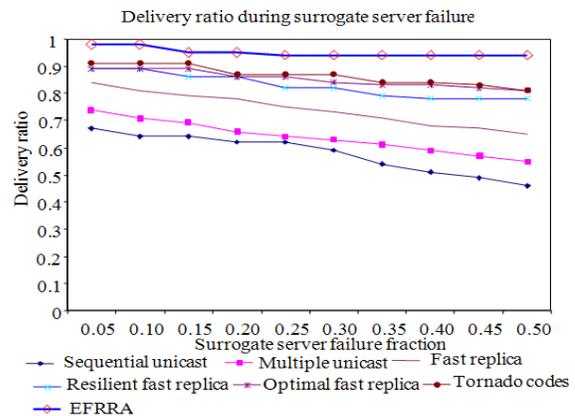


Fig. 4: Delivery ratio during surrogate server failure

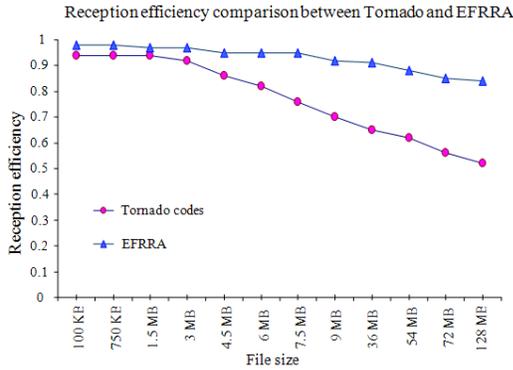


Fig. 5: Reception Efficiency Comparison between Tornado and EFRRA

Table 3: Encoding/decoding times of tornado and EFRRA

File Size	Tornado codes		EFRRA	
	Encoding time (in ms)	Decoding time (in ms)	Encoding time (in ms)	Decoding time (in ms)
100 KB	0.09	0.14	NIL	0.13
750 KB	0.25	0.25		0.22
1.5 MB	0.32	0.36		0.24
3 MB	0.46	0.58		0.54
4.5 MB	0.59	0.79		0.73
6 MB	0.76	0.96		0.88
7.5 MB	1.27	1.32		1.12
9 MB	2.25	2.49		1.27
36 MB	4.32	4.36		2.23
54 MB	7.56	7.42		4.32
72 MB	12.83	11.83		10.32
128 MB	18.94	16.47		11.23

Performance Comparison between Tornado code and EFRRA in terms of Reception Efficiency:

Tornado and EFRRA algorithms are implemented based on digital foundation strategy to distribute the content over the SON. First we split the entire file F into a set of ‘k’ data blocks or packets and produce a set of ‘c’ redundant blocks or packets for a total of n=c+k encoding packets all of a fixed size P. Here n/c is called the stretch factor. Decoding time in the collection step is proportional to k(1+x).P where P is the size of the data block or packet and x is the number of source data blocks not received by the receiving surrogate server and which therefore must be restituted or reconstructed from the redundant data received.

Reception efficiency of a receiving surrogate server can be defined as the ratio between the total number of source data blocks or packets sent by the sender and total number of data blocks received before reconstruction at receiver. It contains two components. First is the decoding efficiency which can be defined as the ratio between the total number of source data blocks

(or) packets and the total number of distinct data blocks received before the reconstruction phase. Second component is the distinctness efficiency which captures the loss in efficiency due to the reception of redundant packets usually caused due to the addition of resiliency in content collection step of EFRRA algorithm. So distinctness efficiency is defined as the ratio between the total number of distinct blocks received at the receiver and the total number of packets received at the receiver. Reception efficiency is measured for both tornado codes and EFRRA algorithms by the account manager module and their comparison is depicted in Fig. 5. It is observed that reception efficiency of EFRRA is better than the reception efficiency of tornado Codes algorithm.

Encoding and Decoding times of Tornado code and EFRRA:

Tornado codes and EFRRA content replication strategies produce a total of ‘n’ encoded packets from a k packet source. To reconstruct the source data, it is necessary to recover Ck packets from the total ‘n’ encoding packets, where C > 1. Encoding and decoding times in the idealistic setting is shown in the Table 3.

$$\text{Encoding time} = (k + c) \ln (1/ \epsilon). P$$

$$\text{Decoding time} = (k + c) \ln (1/ \epsilon). P$$

Where:

- P = Size of the encoding packet
- k = Number of data blocks in source
- c = Number of redundant data blocks
- ε = Stretch factor

In Tornado Codes, an entire file is fragmented in to a ‘k’ packets or blocks of equal size and encodes it into ‘n’ encoded packets where n=2^A-1 such that A is the length of the symbol. A random set of blocks of a file will be replicated in multiple surrogate servers. During content collection, the receiver can run the Tornado decoding algorithm in real-time as the encoding packets arrive and reconstruct the original file as soon as it determines that sufficiently many packets have arrived. But, in EFRRA content replication strategy, the file fragments F_i’s are distributed to a list of surrogate servers i.e., R = {N₁, N₂,... N_n} based on round robin fashion. Hence, no need for encoding the file fragments during distribution. But the receiver has to run the decoding algorithm to reassemble the file fragments to form the original file F. Encoding and Decoding time of Tornado Codes and EFRRA content replication strategies are measured using the information available with the log file and tabulated in Table 3.

Analyze the impact of equitable dominating set based SON in CDN construction:

We constructed the

logical infrastructure of the content distribution network using different overlay construction methodologies such as Semantic Overlay Network (SON), Dominating set based Semantic Overlay Network (DSON) and Equitable Dominating set based Semantic Overlay Network (EDSON) and analyzed their performance in terms of number of surrogate servers in which the content is replicated instead of original number of surrogate servers. It is also observed that equitable dominating set based Semantic Overlay Network are useful in reducing the average number of surrogate servers for content replication to 55 percentages or less. Hence, EDSON based CDN contains lesser number of replica servers compared to SON based CDN and DSON based CDN. This is depicted in Fig. 6.

Performance of EFRR algorithm in SON, DSON and EDSON: We measured the average replication time of EFRR to replicate the different sized files in the SON, DSON and EDSON and the performance graph is depicted in Fig. 7. It is observed that average replication time of EFRR is very less in EDSON based replication set.

Analyze the role of Equitable Dominating Set in surrogate server utilization: CDN utility is the mean of the individual net utilities of each surrogate server in a CDN. Net utility is a value that expresses the relation between the number of bytes of the served content against the number of bytes of the pulled content from origin or other surrogate servers. Net Utility (U_i) of a surrogate server can be given by the formula.

$$U_i = 2 / \pi * \arctan(\alpha) \tag{7}$$

α - ratio between uploaded bytes to downloaded bytes.

The resulting utility value ranges to [0..1].

The value U_i can be

$U_i = 1$ if the surrogate server uploads the content

$U_i = 0$ if the surrogate server downloads the content

$U_i = 0.5$ if upload and downloads are equal

We evaluate the performance of CDN in terms of Mean CDN Utility (U_{Mean}) which can be computed using the following formula.

$$U_{Mean} = 1 / n * \sum_{i=1}^{i=n} U_i \tag{8}$$

We investigated the use of different overlay construction methodologies such as Semantic Overlay Network (SON), Dominating set based SON (DSON) and Equitable Dominating Set based SON (EDSON) in terms of Mean CDN Utility.

It is observed that Mean CDN Utility (U_{Mean}) is uniform in the Equitable Dominating Set based Semantic Overlay Network based CDN of surrogate servers. This is depicted in Fig. 8.

Mean response time vs file size: Mean Response Time is defined as the expected time for a request to be satisfied. It is the summation of all requests' times divided by their quantity. This measure expresses the users' waiting time in order to serve their requests. Lower values indicate fast served content. The overall response time consists of many components, namely, DNS delay, TCP setup delay, network delay between the user and the server, object transmission delay, encoding and decoding times of block level replication and so on. Our response time definition implies the total delay due to all the aforementioned components. We analyzed the Mean Response Time experienced by the users to download different sized files in SON, DSON and EDSON based CDN is depicted in Fig. 9.

Mean response time vs number of clients: It is observed that when client number increases in a network the mean response time always increases. But the mean response time in EDSON based CDN is uniform and is always less than the mean response time in DSON based CDN and SON based CDN which is depicted in Fig. 10.

Mean response time and number of requests: Another finding is, when number of clients is fixed in a network and the number of requests increases then the mean response time of EDSON based content distribution network is always less than DSON based and SON based CDN which is depicted in Fig. 11.

Latency Vs. file size: Latency is defined as the interval between the time the user requests for certain content and the time at which it appears in the user browser or is available at client machine. The end user perceived latency is a useful metric to select the suitable surrogate for that user. In our CDN system, each CDN node determines its set of neighbors using latency information. However, different file sizes have different latencies and web objects can essentially be of any size. Hence, we need techniques to estimate the latency of downloading an object as a function of file size using only a limited number of probes. Fortunately, our measurements show that the average network latency of downloading a file is roughly proportional to its size when the file size is between 100KB and 128 MB and is depicted in Fig. 12.

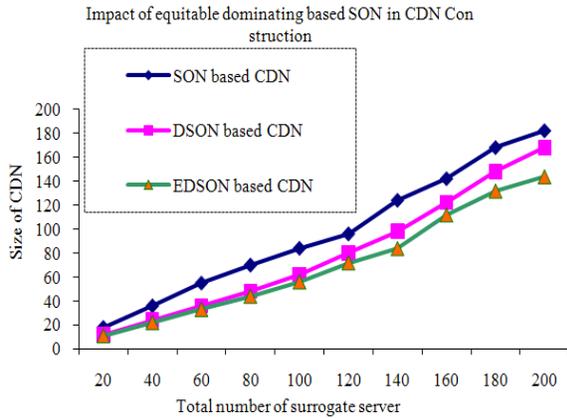


Fig. 6: Reduction in replication set due to the impact of equitable dominating set in CDN formation

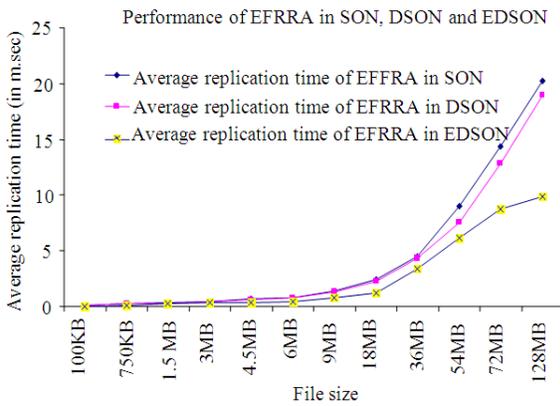


Fig. 7: Performance of EFRRA in SON, DSON and EDSON

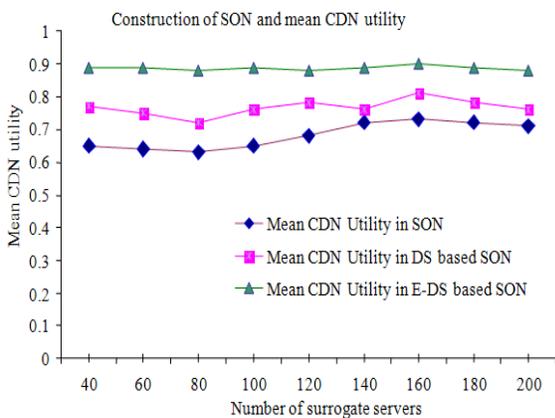


Fig. 8: Construction of SON Vs Mean CDN Utility

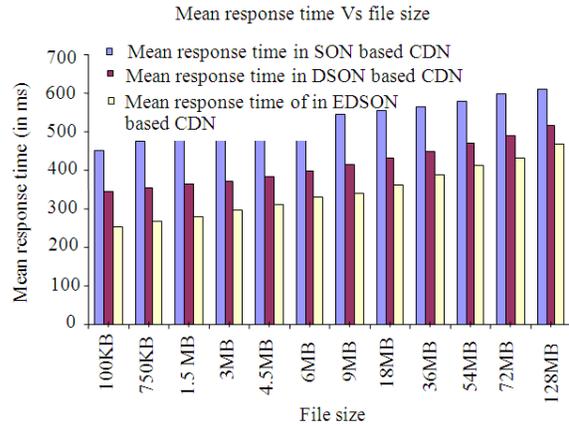


Fig. 9: Mean response time and file size

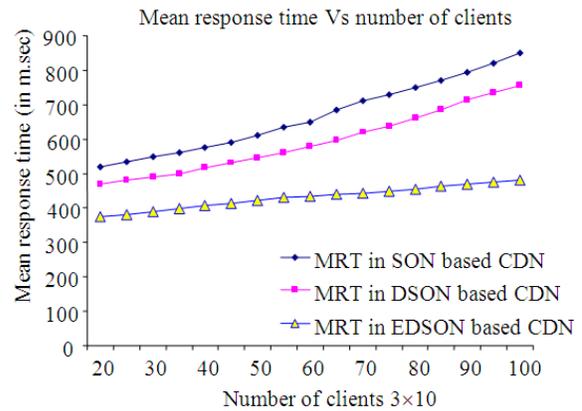


Fig. 10: Mean response time and number of clients

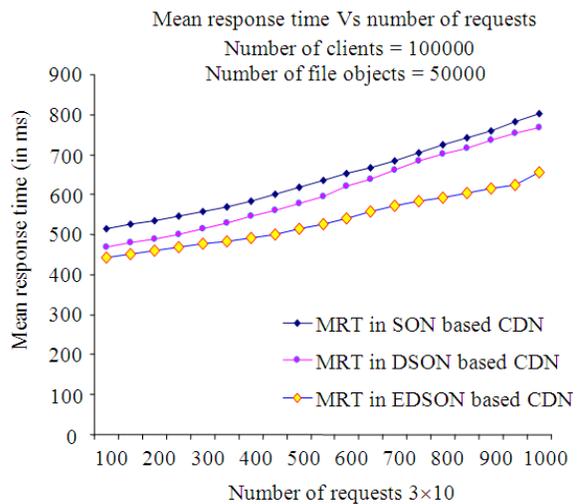


Fig. 11: Mean response time and number of requests

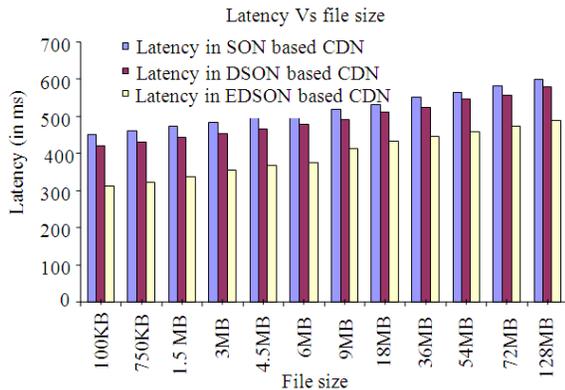


Fig. 12: Latency and file size

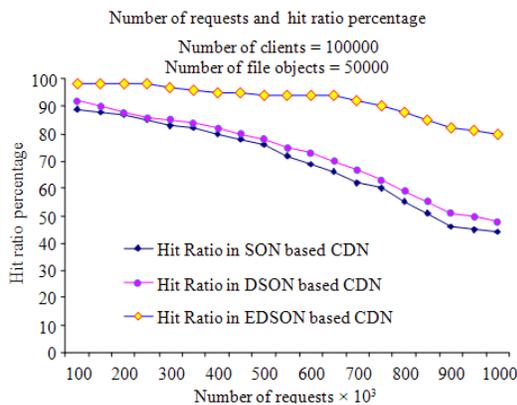


Fig. 13: Number of requests and hit ratio percentage

Number of requests Vs hit ratio percentage: Generally surrogate servers serve contents to the clients from its cache. Hit ratio percentage is the ratio between the number of contents a surrogate is serving and the number of content request it is receiving. A high hit ratio indicates an effective cache management policy, content distribution policy and surrogate server cooperation. It improves network performance and bandwidth saving. From Fig. 13, we can see that for particular number of request, hit ratio percentage of EDSON based CDN is always higher than hit ratio percentages of DSON based CDN and SON based CDN. Also in EDSON based CDN infrastructure most of the time the surrogates are able to serve the request, as load is almost equally balanced among the surrogates, so redirection probability is less.

But in DSON and SON based CDN request redirection probability is higher and in worst case it may happen that there is no requested content in a surrogate. So the surrogate redirects the requests to other surrogates that have those contents or sometimes to the origin server itself.

CONCLUSION

In this study, first we constructed equitable dominating set based semantic overlay network (EDSON) of surrogate servers and applied EFRRR content replication algorithms for replicating the content from the origin server to a set of surrogate servers.

Both analytical study and empirical study were carried out to analyze the performance of the content distribution algorithms.

The effect of equitable dominating set in SON formation and how it was useful in reducing the redundancy was investigated. It is also observed that equitable dominating set based SON is useful in keeping the average replication time stable and much more predictable and further noticed that Mean CDN Utility is uniform. We evaluated the performance of EDSON based CDN in terms of Mean Response Time, Latency and hit ratio percentage. Our future work includes design of virtual organization based peering of cooperative and coordinated CDNs and evaluation of its performance.

ACKNOWLEDGMENT

The researchers acknowledge TIFAC-CORE in Network Engineering (established under the Mission REACH Program of Department of Science and Technology, Govt. of India) for providing necessary facilities in Open Source Technologies Laboratory for working on this project. The authors would also thank Kalasalingam Anandam Ammal Charities for providing professional development allowance to support this study.

REFERENCES

- Amutharaj, J. and S. Radhakrishnan, 2010. EFRRR: An efficient-fault-resilient-replica-algorithm for content distribution networks. *Int. J. Comput. Sci. Inform. Security*, 8: 189-197. <http://www.docstoc.com/docs/53650353/EFRRR-An-Efficient-Fault-Resilient-Replica-Algorithm-for-Content-Distribution-Networks>
- Amutharaj, J., M. Gomathynayagam and S. Radhakrishnan, 2010. Design of equitable dominating set based semantic overlay network with optimal fast replica for resilient and balanced content distribution. *Proceedings of the ACEEE International Conference on Advances in Computer Science, Trivandrum, Dec. 22-23*, pp: 1-6. <http://www.searchdl.org/conference/ACS2010/24.PDF>

- Amutharaj, J and S. Radhakrishnan, 2007. Dominating set based semantic overlay networks for efficient content distribution. Proceedings in IEEE-ICSCN, 1, Madras Institute of Technology, Anna University, Chennai, pp: 228-232. DOI: 10.1109/ICSCN.2007.350736
- Amutharaj, J. and S. Radhakrishnan, 2008. Dominating set theory based semantic overlay networks for efficient and resilient content distribution. J. Networks, Acad. Publishers, 3: 42-48. <http://www.academypublisher.com/jnw/vol03/no03/jnw03034248.html>
- Cherksova, L. and J. Kee, 2002. Fast replica: Efficient large file distribution within content delivery networks. Proceedings of the 4th SENIX symposium on Internet Technologies. http://www.hpl.hp.com/personal/Lucy_Cherkasova/papers/usits-fr.pdf
- Geetha, P. and V. Narayanan, 2011. Survey of content-based video retrieval. J. Comput. Sci., 4: 474-486. DOI: 10.3844/jcssp.2008.474.486
- Han, B. and W. Jia, 2005. Design and analysis of connected dominating set formation for topology control in wireless adhoc networks. Proceedings of 14th International Conference on Computer Communication and Networks, DOI: 10.1109/ICCCN.2005.1523797
- Lu, Z.H., W.M. Fu, S.Y. Zhang and Y.P. Zhong, 2008. TRRR: A tree-round-robin-replica content replication algorithm for improving fast replica in content delivery networks. Proceedings of the 4th International Conference on Wireless Communications, Networking and Mobile Computing, DOI: 10.1109/WiCom.2008.1352
- Liu, Z., B. Wang and Q. Tang, 2010. Approximation two independent sets based connected dominating set construction algorithm for wireless sensor networks. Inform. Technol. J., 9: 864-876.
- Liu, Z., B. Wang and L. Guo, 2010. A survey on connected dominating set construction algorithm for wireless sensor networks. Inform. Technol. J., 9: 1081-1092.
- Ma, C., Y. Yang and Z. Zhang, 2005. Constructing battery-aware virtual backbone in sensor networks. Proceedings of the International Conference on Parallel Processing, IEEE Computer Society. DOI: 10.1155/2007/40154
- Nandagopal, M. and R. Uthariaraj, 2011. Performance analysis of resource selection algorithms in grid computing environment. J. Comput. Sci., 7: 493-498. ISSN: 1549-3636
- Pathan, A.M.K. and R. Buyya, 2007. A taxonomy and survey of CDNs. Technical Report, GRIDS-TR-2007-4, The University of Melbourne, Australia. <http://www.gridbus.org/reports/CDN-Taxonomy.pdf>
- Pathan, M. and R. Buyya, 2009. Architecture and performance models for QoS-driven effective peering of content delivery networks. Technical Report, Grid Computing and Distributed Systems (GRIDS) Laboratory, The University of Melbourne. <http://www.buyya.com/papers/QoS-Peering-CDN2009.pdf>
- Ramadoss, B and K. Rajkumar, 2007. Semi-automated annotation and retrieval of dance media objects. J. Cybernetics Syst., Taylor and Francis, PA, USA. CBS 38(4). DOI. 10.1080/01969720701291189
- Shakkottai, S. and R. Johari, 2010. Demand-Aware content distribution on the internet. IEEE/ACM Trans. Network., 18. DOI: 10.1109/TNET.2009.2035047
- Stamos, K, G. Pallis, A. Vakali, D. Katsaros and A. Sidiropoulos *et al.*, 2008. CDNSim: A simulation tool for content distribution networks. ACM Trans. Model. Comput. Simulation, 20: DOI: 10.1145/1734222.1734226