

Design of QoS Aware Dominating set based Semantic Overlay Network (QADSON) for Content Distribution

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

Abstract: Problem statement: This study addressed the user's perceived quality of service requirements in content distribution and investigated the role of QoS. Aware Dominating set based Semantic Overlay Network (QADSON) in surrogate server selection to achieve the specified quality of service. **Approach:** At first, we constructed the QoS aware dominating set based semantic overlay network which was a virtual network of surrogate servers that was built on top of existing physical network with the purpose to implement new network services and features such as efficiency, exact-1 domination, controlled redundancy and fault tolerance that are not available in the existing network. We applied EFRRA content replication algorithm to disseminate the content among the surrogate servers and evaluated its performance in QADSON. **Results:** We assessed the efficiency, exact-1 domination, controlled redundancy and fault resiliency of QADSON in terms of mean response time, mean CDN utility, hit ratio percentage, rejection rate and CDN load. We extended the simulation experiments to analyze the role of QADSON in maintaining uniform CDN utility of above 0.95. **Conclusion:** We also investigated the quality of service requirements for the content distribution and evaluated performance of QADSON based CDN in terms of mean response time, latency, hit ratio percentage, mean CDN utility, rejection rate and CDN load.

Key words: Content Distribution Network (CDN), QoS Aware Dominating set based Semantic Overlay Network (QADSON), Efficient and Fault Resilient Replication Algorithm (EFRRA)

INTRODUCTION

A 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). A CDN provider mainly focuses on providing the following services and functionalities: storage and management of content, distribution of content among surrogate servers, cache management, delivery of static, dynamic and streaming content, back up and disaster recovery solutions, monitoring services, performance monitoring and reporting.

This paper dealt with the implementation of Quality of Service aware dominating set for selecting the replication set of surrogate servers. Then EFRRA

content replication algorithm is applied to disseminate the content among the surrogate servers in the QADSON based CDN. Performance measurement is carried out to estimate the values of performance indices such as size of CDN, mean response time, mean CDN utility, latency, hit ratio percentage, number of completed requests, rejection rate and CDN load 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 efficiency, load balancing and fault tolerance in massive content distribution systems.

Pathan and Buyya (2008) 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

Corresponding Author: Amutharaj Joyson, Department of Computer Science and Engineering,
Arulmigu Kalasalingam College of Engineering, Anandnagar, Krishnankoil, Srivilliputtur (Via),
Tamil Nadu, India

mechanisms, content replication techniques, load balancing and cache management.

Rodriguez and Biersack (2002) proposed a dynamic parallel-access scheme to access multiple mirror servers. They showed that their dynamic parallel downloading scheme achieves significant downloading speedup with respect to a single server scheme. However, they studied only the scenario where one client uses parallel downloading. The effect and consequences when clients choose to adopt the same scheme is not addresses.

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.

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. Xia *et al.* (2009) considered a two-tier content distribution system for distributing massive content and proposed popularity-based file replication techniques within the CDN using multiple hash functions.

Ozkasap *et al.* (2009) proposed and designed a peer-to-peer system; SeCond, addressing the distribution of large sized content to a large number of end systems in an efficient manner. It employed a self-organizing epidemic dissemination scheme for state propagation of available blocks and initiation of block transmissions. They showed that SeCond is a scalable and adaptive protocol which took the heterogeneity of the peers into account.

Stamos *et al.* (2008) have presented a generic non-parametric heuristic method that integrates caching and content replication to improve the performance of CDN in terms of availability and cache hit ratio. They developed a placement similarity approach, called SRC, for evaluating the level of integration.

Byers *et al.* (1999) proposed a parallel accessing scheme based on tornado codes in which a client is allowed to access a file from multiple mirror sites in parallel to speed up the download.

Pathan and Buyya (2009) presented architecture to support peering arrangements between CDNs, based on a Virtual Organization (VO) model. Performance can be achieved through proper policy management of negotiated Service Level Agreements (SLAs) between peers. They also 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.

Geetha and Narayanan (2011) presented a survey on current trends and methods in video retrieval. The major themes covered by their study include 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.

Nandagopal and Uthariaraj (2011) proposed Multi Criteria Resource Selection (MCRS) algorithm which considered multiple criteria such as processing power, workload and network bandwidth of the resource during resource selection. They conducted simulation experiments to evaluate the performance of the algorithm and compared its performance with conventional single criteria algorithm.

Ramadoss and Rajkumar (2007) considered a system for the semiautomatic annotation of an audio-visual media of dance domain, DMAR (Dance Media Annotation, authoring and Retrieval system). Their work outlined the underlying XML Schema based content description structures of DMAR and discussed the merits and demerits of their approach of evolving semantic network as the basis for the audio-visual content description. Further, they proposed a quality metric, fidelity to evaluate the expressive power of the dance annotations. Evaluation results are presented to depict the performance of the dance video queries in terms of precision and recall.

Caviglione and Cervellera (2011) proposed an overlay Content Distribution Network (CDN) which can able to sustain the real-time delivery of data streams. They modeled a predictive control scheme to enhance utilization of resources and evaluated the effectiveness of the proposed solution during multimedia streaming and interactive grid data.

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 a hybrid replication strategy named EFRR 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 EFRR and proved that EFRR algorithm outperforms other algorithms in terms of replication time and also maintain the competent delivery ratio.

MATERIALS AND METHODS

Design of QoS aware 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 $V \setminus D$.

QoS Aware Dominating set is formed with an aim to construct a dominating set with the properties such as Efficiency, Controlled Redundancy and fault-tolerance. Efficiency can be achieved by implementing the principle of dominate every vertex in the adjacent set at least once. Cardinality Redundancy of a graph $R(G)$ can be defined as the minimum number of vertices in the adjacent set dominated more than once by a dominating set. Controlled redundancy can be incorporated with the dominating set by maintaining the minimum cardinality redundancy value. Fault Tolerance is defined as the ability of the network to provide service even when it contains a faulty component or components. Fault tolerance property can be ensured by modeling the behavior of a network in the presence of a fault and can be analyzed by determining the effect that removing an edge(link) or a vertex(server failure) from its underlying graph.

QoS aware dominating Set based SON (QADSON) 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).

// QoS aware dominating set formation steps

Step 7: After forming the DS, check the degree of each vertices of the connected dominating set.

Step 8: If the degree of any vertex is more than one then marks that vertex gray and find the suitable alternate vertex as the member of the dominating set:

// exact-1 domination

Step: 9: Check whether the following criterions are satisfied by the DS:

- $|N[u] \cap D| = 1$ for every $v \in V(G)$ // Efficiency Criterion
- $R(G) = \min\{\sum_{v \in V(G)} |N[v] \cap D| : D\} = \min\{I(D)\}$

where, D is a dominating set // Redundance (Control) Criterion:

- For any tree T in D with $n \geq 2$, there exists a vertex $v \in V$, such that $\gamma(Y-v) = \gamma(T)$ // Fault Tolerance criterion

The QoS Aware 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.

Design of an efficient and Fault Resilient Replication Algorithm (EFRR):

Working principle of EFRR: A novel algorithm called EFRR is proposed for an efficient and fault resilient replication of large files in the CDN. Working mechanism of EFRR 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 'n' different nodes in the group. Each node is provided with a distribution list of 'm' surrogate servers. In the second round, each node propagates its sub file to 'm' surrogate servers in the distribution list. This propagation is repeated until 'k' number of file fragments would be reached by all the 'n' recipient surrogate servers in the surrogate server site. 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 $n \times n$ diverse internet paths within the replication group where each path is used for transferring $1/n^{\text{th}}$ 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 to each recipient node N_i ($1 \leq i \leq n$) the following:

- A distribution list of nodes $R = \{N_1, \dots, N_m\}$ 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 resiliency to node failure. 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 has to recollect all the sub files or blocks of the requested file from the overlay network of surrogate servers in a parallel manner.

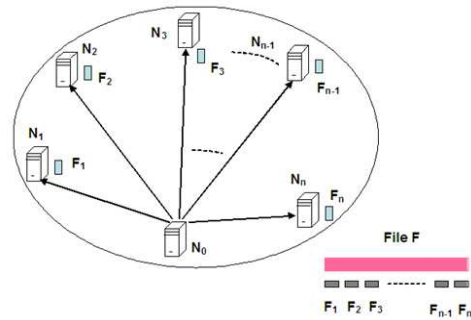


Fig. 1: Distribution step in EFRRA

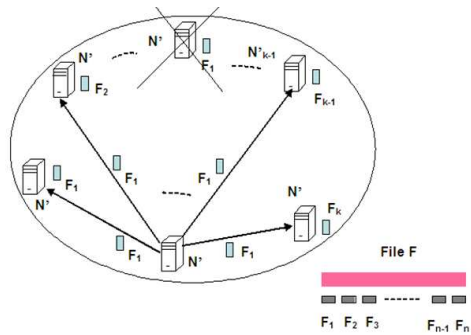


Fig. 2: Adding resiliency to EFRRA

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 $\{N_1, N_2... N_m\}$ and download the whole file F .

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.

Average replication time:

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

Table 1: 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	$2 \times \text{Size (F)}/(n \times B)$
Resilient fast replica without node failure	$2 \times \text{Size (F)}/(n \times B)$
Resilient fast replica with failure of 'm' servers	$(2+m/n) * \text{Size (F)}/(n \times B)$
Optimal fast replica	$((k+n) / n * n * k) * \text{Size(F)}/B$
Tornado codes	$2 * c / n * \text{Size (F)}/B$
EFRRRA	$(2+c) / n * \text{Size(F)}/B$

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:

$$\text{Time}_{\text{distribution}} = \text{Size (F)} / (n \times B) \quad (3)$$

$$\text{Time}_{\text{collection}} = \text{Size (F)} / (n \times B) \quad (4)$$

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 1.

Therefore, Replication Time proportion of different content distribution algorithms can be expressed as follows:

$$\text{Time}_{\text{SU}} : \text{Time}_{\text{MU}} : \text{Time}_{\text{FR}} : \text{Time}_{\text{R-FR}} : \text{Time}_{\text{O-FR}} : \text{Time}_{\text{TC}} : \text{Time}_{\text{EFRRRA}}$$

$$n : 1 : 2/n : (2+m/n) * 1/n : ((k+n) / n * n * k) : 2 * c / n : (2+c) / n \quad (5)$$

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 QoS aware dominating set based semantic overlay network: QoS Aware 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 dominating set would satisfy QoS requirement criterions such as exact-1 domination, efficiency, controlled redundancy and fault tolerance. Each vertex 'u' in dominating set 'D' has to dominate every vertex atleast once. So each

vertex in D 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 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 QADSON can be expressed as follows:

$$r : 1 : 2/r : (2+m/r) * 1/r : ((k+r) / r * r * k) : 2 * c / r : (2+c) / r \text{ where } r < n \quad (6)$$

Simulation test bed and performance measurement:

We used the simulation tool CDNSim [25] to create and customize the simulation environment named QADSONCDN which includes the following five modules:

- QADSON based CDN Model : To evaluate the performance of the proposed QADSONCDN simulation environment developed using the simulation tool CDNSim [25], which simulates a main CDN infrastructure based on QoS Aware dominating set and is implemented in the C programming language. In QADSON based CDN infrastructure where surrogate servers are logically grouped to form the QoS aware content distribution network backbone based on QoS constraint dominating set construction algorithm.. 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 QADSON. 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 QADSON then the content will be searched on the adjacent surrogate server which is the member of QADSON 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 QADSON and served. If the content is not available in the entire QADSON 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 work, 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, number of completed requests, number of requests rejected, number of requests arrived etc. These log information can be used by the account manager to compute the Quality of Service metrics such as mean CDN utility, average content replication time, delivery ratio, reception efficiency, mean response time, latency, hit ratio percentage, number of completed requests, rejection rate and mean CDN load.

CDN network simulation set up: The distribution and arrangement of servers, routers and clients in the network affects the performance of the CDN. 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 flavors: 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, QADSON formation mechanism are simulated.

Performance of different content distribution schemes in QADSON in terms of average replication time: We experimented with 12 different sized files; 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 QADSON based CDN infrastructure.

Figure 3 shows the average replication time measured by different individual surrogate servers for different file sizes of 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 and 128 MB when QADSON 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 QADSON based replication set is much more stable and predictable. Hence, EFRRRA algorithm outperforms all the traditional content distribution schemes.

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

Delivery ratio Vs surrogate server failure fraction: 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 EFRR in SON, DSON, EDSON and QADSON when the number of simultaneous surrogate server failures in the CDN has been analyzed and its performance is shown in Fig. 5.

From the delivery ratio analysis shown in Fig. 5, we found that the delivery ratio of EFRR algorithm in QADSON based CDN is above 0.97 always, even though the surrogate server failure fraction reaches 0.5. Hence, QADSON based CDN is found to be fault tolerant and efficient during surrogate server failure.

Analyze the impact of QoS aware dominating set based SON in CDN formation: By the implementation of QoS Aware dominating set for the clustering of surrogate servers in the SON, the average number of surrogate servers for content replication is reduced to 55 percentages or less. This is depicted in Fig. 6.

Figure 6 Reduction in Replication Set due to the impact of Equitable Dominating Set in CDN Formation

Analyse the role of QoS aware 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 only content
- $U_i = 0$ if the surrogate server downloads only 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), Equitable Dominating Set based SON (EDSON) and QoS Aware Dominating Set based SON in terms of Mean CDN Utility.

It is observed that QADSON based CDN is very useful in maintaining uniform Mean CDN Utility (U_{Mean}) of 0.95. It is also found that QADSON based CDN outperforms other CDN construction methodologies such as SON, DSON and EDSON in terms of Mean CDN Utility. This is depicted in Fig. 7.

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, EDSON and QADSON based CDN is depicted in Fig. 8. It is observed that Mean Response Time experienced by the users to download different sized files in QADSON is lesser than SON based CDN, DSON based CDN and EDSON based CDN. So users quickly received the content in QADSON based CDN.

Mean response time Vs number of clients: Simulation experiment is conducted by fixing the input values for following parameters:

- Maximum number of requests = 1,000,000
- Number of file objects = 50000
- Maximum website size = 1GB

Simulation parameters	Specification
Network topology	AS, WAXMAN, Transit_Stub, Random
Surrogate server cooperation mechanisms used	SON, DSON, EDSON, QADSON
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
QoS parameters measured at Client side	Mean response time, mean CDN Utility, Latency, hit ratio percentage, rejection rate and CDN load

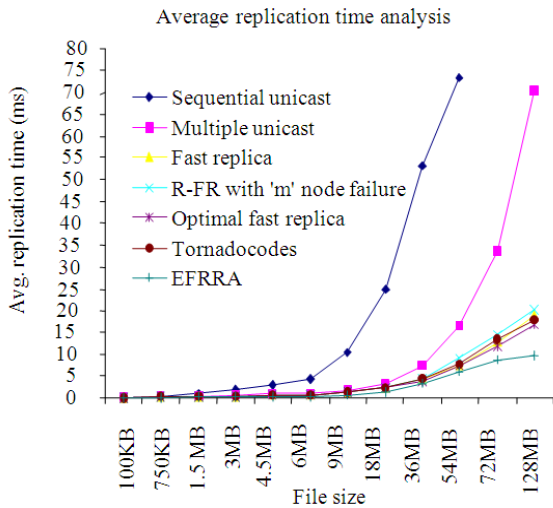


Fig. 3: Average content replication times for various schemes

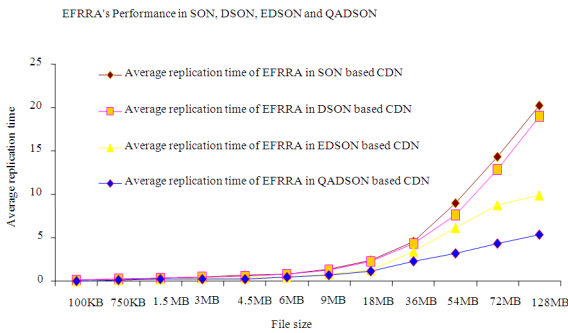


Fig. 4: Performance of EFRRRA in SON, DSON, EDSON and QADSON in terms of Average Replication Time

From the experimental results, we found that when the number of clients increases in a network the mean response time always increases. But the increase in mean response time in QADSON based CDN is uniform and is always less than the mean response times in EDSON based CDN, DSON based CDN and SON based CDN which is depicted in Fig. 9.

Mean response time Vs 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 QADSON based content distribution network is always less than EDSON based CDN, DSON based CDN and SON based CDN which is depicted in Fig. 10.

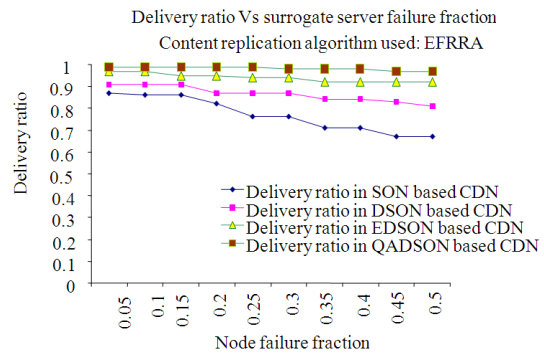


Fig. 5: Delivery ratio during surrogate server failure

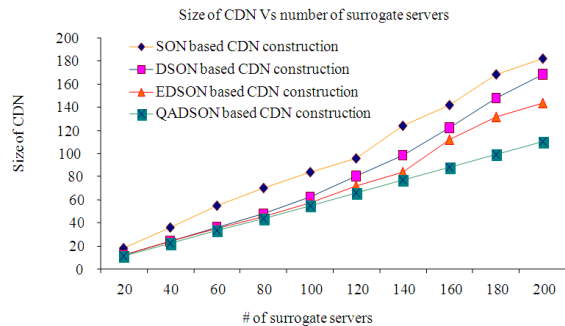


Fig. 6: Reduction in replication set due to the impact of equitable dominating set in CDN formation

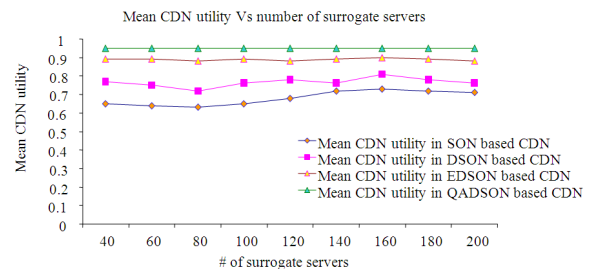


Fig. 7: Construction of SON Vs mean CDN utility

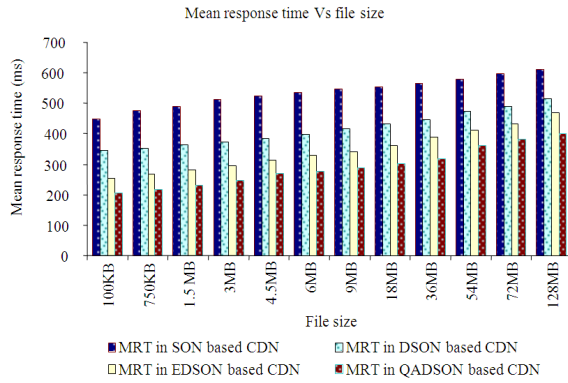


Fig. 8: Mean response time Vs file size

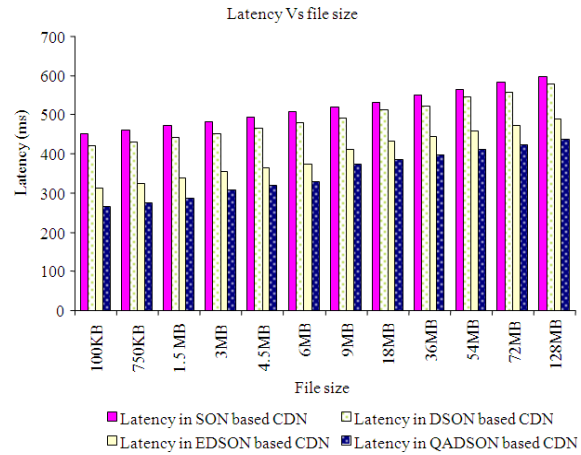


Fig. 11: Latency Vs file size

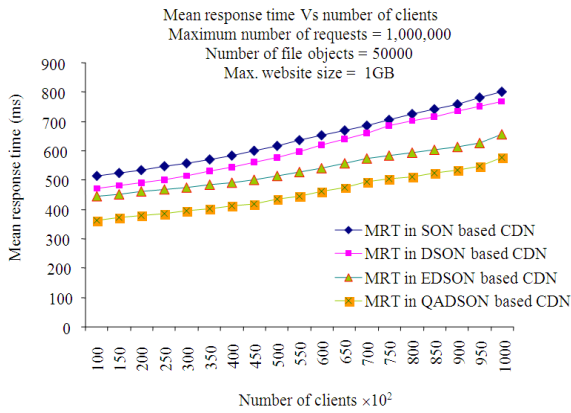


Fig. 9: Mean response time Vs number of clients

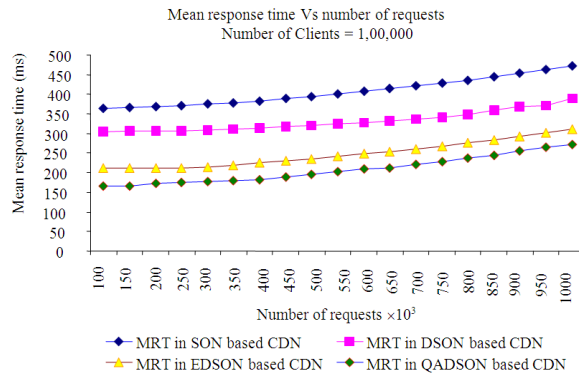


Fig. 10: Mean response time Vs number of requests

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 neighbours using latency information. However, different sized files 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. Our measurements from simulation experiments showed 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. 11. From the measured values of latency, it is found that latency in QADSON based CDN is lesser than EDSON based CDN, DSON based CDN and SON based CDN for different sized files ranges from 100KB and 128 MB.

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.

Simulation experiment is conducted by fixing the input values for following parameters:

- Number of clients = 100000
- Number of file objects = 50000

From experimental results plotted in Fig. 12, we can see that for particular number of request, hit ratio percentage of QADSON based CDN is always higher than hit ratio percentages of EDSON based CDN, DSON based CDN and SON based CDN.

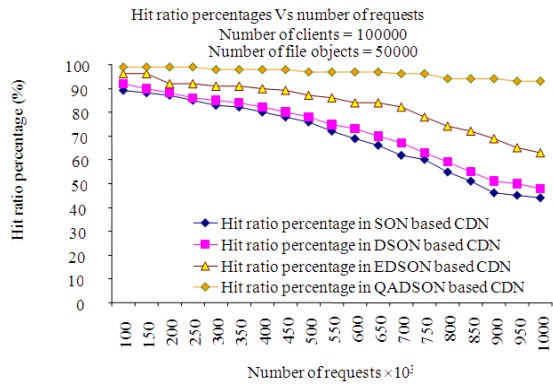


Fig. 12: Number of requests Vs hit ratio percentage

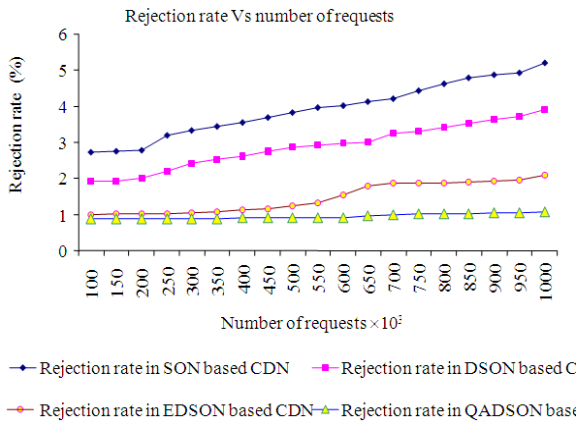


Fig. 13: Rejection rate Vs number of requests

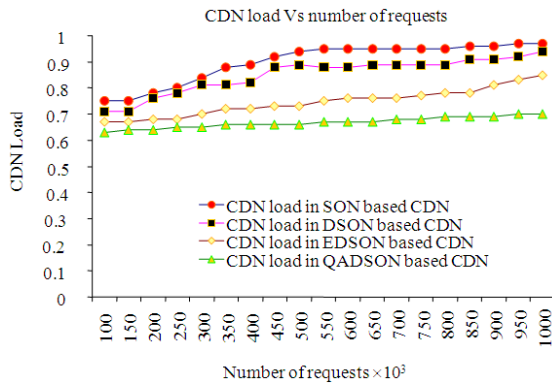


Fig. 14: CDN Load Vs number of requests

Also in QADSON 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 EDSON, 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.

Rejection rate Vs number of requests: Rejection rate is defined as the percentage of dropped requests due to service unavailability. It depends on the number of disruptions due to service unavailability in the network. Low rejection rate indicates that users experiencing the high service availability. From Fig. 13, it is observed that QADSON based CDN has low rejection rate lesser than 1.08% due to fault tolerant property of QADSON which is very low compared to the rejection rate observed in EDSON based CDN, DSON based CDN and SON based CDN.

CDN load vs total number of requests: CDN load can be defined as the ratio of mean request arrival rate (i.e. number of requests arrived per second) and mean service rate. From the experimental results, we found that when the number of requests increases in the CDN, the CDN load always increases. But the increase in load in QADSON based CDN is uniform and in between 0.6 to 0.7. It is also observed that the CDN load of QADSON based CDN is always less than the CDN load of EDSON based CDN, DSON based CDN and SON based CDN which is depicted in Fig. 14.

CONCLUSION

In this study, first we constructed QoS Aware Dominating set based Semantic Overlay Network (QADSON) of surrogate servers to form the logical infrastructure of CDN. Then we applied EFRRRA content replication algorithm to disseminate the content among the surrogate servers in QADSON.

We have conducted simulation experiments using CDNSim and analyzed the performance of EFRRRA algorithm in terms of average replication time, delivery ratio in SON, DSON, EDSON and QADSON based CDN.

The effect of QoS Aware dominating set in SON formation and how it was useful in reducing the redundancy, improving the efficiency and maintaining fault tolerance were investigated. It is also observed that QoS aware dominating set based SON is useful in keeping the mean response time stable and much more predictable and further noticed that Mean CDN Utility is uniform and above 0.95. We also evaluated the performance of QADSON based CDN in terms of Size of CDN, Latency, hit ratio percentage, rejection rate

and CDN load. 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 generously providing professional development allowance to support this study.

REFERENCES

- Pathan, A.M.K. and R. Buyya, 2008. A Taxonomy and Survey of CDNs. The University of Melbourne.
- Rodriguez, P. and E. Biersack, 2002. Dynamic parallel access to replicated content in the Internet. *IEEE/ACM Trans. Network.*, 10: 455-465. DOI: 10.1109/TNET.2002.801413
- Cherksova, L. and J. Kee, 2002. Fast Replica: Efficient Large file Distribution within Content Delivery Networks. Proceedings of 4th SENIX Symposium on Internet Technologies, (USITS'03), USENIX Association Berkeley, CA, USA, pp: 7-7.
- 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 4th International Conference on Wireless Communications, Networking and Mobile Computing, Oct. 12-14, IEEE Xplore Press, pp: 1-4. DOI: 10.1109/WiCom.2008.1352
- 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, Oct. 17-19, IEEE Xplore Press, USA, pp: 7-12. DOI: 10.1109/ICCCN.2005.1523797
- Lu, K. J.A. Bolla, and D.T. Huynh, 2006. Adapting connected d-hop dominating sets to topology changes in wireless adhoc networks. Proceedings of 25th IEEE International Conference on Performance, Computing and Communication, April 10-12, IEEE Xplore Press, Phoenix, AZ, pp: 214-222. DOI: 10.1109/.2006.1629409
- Ma, C., Y. Yang and Z. Zhang, 2005. Constructing battery-aware virtual backbone in sensor networks, Proceedings of the International Conference on Parallel Processing, June 14-17, IEEE Xplore Press, USA, pp: 203-210. DOI: 10.1109/ICPP.2005.27
- Shakkottai, S. and R. Johari, 2010. Demand-aware content distribution on the internet, *IEEE/ACM Trans. Network.*, 18: 476-489. DOI: 10.1109/TNET.2009.2035047
- Xia, Y, S. Chen, C. Cho and V. Korgaonkar, 2009. Algorithms and performance of load-balancing with multiple hash functions in massive content distribution. *Comput. Networks*, 53: 110-125. DOI: 10.1016/J.COMNET.2008.10.003
- Ozkasap, O., M. Caglar and A. Alagoz, 2009. Principles and performance analysis of SeCond: A system for epidemic peer-to-peer content distribution. *J. Network Comput. Appl.*, 32: 666-683. DOI: 10.1016/j.jnca.2008.07.005
- Byers, J., M. Luby and M. Mitzenmacher and A. Rege, 1999. Accessing multiple mirror sites in parallel: Using tornado codes to speed up downloads, *Proc. IEEE INFOCOM*, 1: 275-283. DOI: 10.1109/INFCOM.1999.749293
- Pathan, M. and R. Buyya, 2009. Architecture and performance models for QoS-driven effective peering of content delivery networks. *Multiagent Grid Syst.*, 5: 165-195. pdf
- Stamos, K., G. Pallis, A. Vakali, D. Katsaros, A. Sidiropoulos and Y. Manolopoulos, 2010. CDNsims: A simulation tool for content distribution networks. *ACM Trans. Modell. Comput. Simulat.*, 5: 1-0.
- 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
- Nandagopal, M. and R. Uthariaraj, 2011. Performance analysis of resource selection algorithms in grid computing environment. *J. Comput. Sci.*, 7: 493-498. DOI: 10.3844/jcssp.2011.493.498
- Ramadoss, B. and K. Rajkumar, 2007, Semi-automated annotation and retrieval of dance media objects. *J. Cybernet. Syst.*, Taylor Francis, 38: 349-379. DOI: 10.1080/01969720701291189
- Caviglione, L. and C. Cervellera, 2010. Design, optimization and performance evaluation of a content distribution overlay for streaming, *Comput. Commun.*, 34: 1497-1509. DOI: 10.1016/J.COMCOM.2010.04.047

- Amutharaj, J and S. Radhakrishnan, 2008. Dominating set theory based semantic overlay networks for efficient and resilient content distribution. *J. Networks Acad. Pub.*, 3: 42-48. DOI: 10.4304/jnw.3.3.42-48
- Amutharaj, J. and S. Radhakrishnan, 2010. EFRR: An efficient-fault-resilient-replica-algorithm for content distribution networks. *Int. J. Comput. Sci. Inform. Security*, 8: 189-197.
- 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 ACEEE International Conference on Advances in Computer Science, (ACE '10), ACEEE, India*, pp: 1-6.