Available online at www.elixirpublishers.com (Elixir International Journal)

Computer Engineering

Elixir Comp. Engg. 79 (2015) 30309-30314

ISSN: 2229-712X

Improving the QoS, and Memory Utilization of Proxy Server by Identifying and Removing the Abnormal Web Traffic using Connection Graph

Subhash Chand[1] and Sanjay Mathur[2]

[1]Department of Computer Engineering, College of Technology, G.B. Pant University of Agriculture & Technology, Pantnagar-263145, India.

[2]Department of Electronics & Communication Engineering, College of Technology, G.B. Pant University of Agriculture & Technology, Pantnagar-263145, India.

ARTICLE INFO

Article history:

Received: 10 December 2014;

Received in revised form:

20 January 2015;

Accepted: 5 February 2015;

Keywords

Client, Server, Graph, In-degree, Out-degree, QoS.

ABSTRACT

Providing better Internet services to users' becomes a challenge to researchers' and Internet Service Provider (ISP). The Quality of Service (QoS) may be improved by using web caching at server level. Web caching catch the frequently requested web objects into the memory of server. The memory is limited in size. Therefore web object replacement policy, which selects the web objects to be replaced from memory play an important role in web caching. In this paper web objects are selected on the basis of client Out-degree, server In-degree and frequency of URLs' by using directed graph method between clients and servers. An experiment is performed on the real access.log file data of proxy server of central computing facility of the University. The results of experiment are satisfactory, which improve the QoS and best utilize the memory.

Introduction

The Internet plays an important role in the progress of developing nations and developed nations. Internet plays a vital role in operational, education, research, personal communications. Due to lack of public IT infrastructure in developing nations, Internet becomes an expensive resource [1]. Internet becomes more popular because of its availability on mobile phones and its variety of services like email, file transfer, chatting, instant messaging, remote login, website surfing, social communication, etc. Due to increasing number of users and services provided by Internet the QoS becomes a challenge to researchers.

The rapid growth of Internet users has put a lot of pressure on Internet Service Providers (ISP), website owners etc., to improve the Quality of Service (QoS) [2]. Web caching is the way to improve the QoS [3]. The heart of a caching system is the page replacement policy, which selects the pages to be replaced from the proxy cache when a request arrives [4]. The pages identified for replacement may be termed as outliers. The QoS is also improved by identifying and stopping the abnormal or unwanted web traffic on the network. This abnormal web traffic generated by non-human activity such as bot-nets or worms, delude peoples or by any activity which are not allowed on the network. This abnormal web traffic is identified by using connection graph between clients and servers, Client Out-degree C_{out} and Server In-degree S_{in} [5].

In this paper regarding QoS seven parameters are considered. These parameters are known as Average hit percentage, Average number of server, Average number of clients, Average number of URLs, Average retrieve time, Average frequency of URLs, Average data transfer rate [6].

The proxy server normally has limited number of resources and each resource has limited capacity. Therefore best utilization of hardware resources becomes a challenge. In this study memory is considered as a hardware resource which is limited in its capacity. For better utilization of memory we first identify the web objects which can be deleted and denoted as outliers onwards. These outlier web pages are chosen on the basis of C_{out} , S_{in} and frequency of the web page, where C_{out} and S_{in} are defined as client Out-degree and server In-degree respectively. In case C_{out} and S_{in} values becomes 1 then concern network behave like peer to peer (P2P). The term outlier is normally denoted to an object which occupies a place different from its class. Some approaches based on classification and neural networks have been applied to identify the misclassified web objects for removal from the cache [7].After identification of these outliers they are removed from memory and average of remaining web objects are calculated for all the parameter i.e. HIT percentage, number of servers, number of URLs retrieval time, retrieved data, frequency of URLs data transfer rate..

Internet is working on client server model. The server keeps the web pages, files, programmes etc. which are demanded by many users. There is no limit of distance between client and server, but it is obvious that more distance between client and server will take more time to get the information from server to client. Using the caching technique this retrieval time is reduced or minimized for the cached web pages. The web page caching may be done at client level, network server level, ISP server level. If the demanded page is available in the cache memory of server then it is known as HIT, otherwise MISS. In case of HIT the webpage is available to client from the cache itself and not from the original web server. In HIT case the retrieval time of web page is reduced to minimum and the traffic on network and load on original web server is reduced. Proxy servers are designed to decrease network traffic, reduce user perceived lag, and reduce load on original server[8]. There are several ways of evaluating performance of proxy server. Some approaches use information concerning utilization of resources, such as memory, disk space, cpu usages, etc. Others consider bandwidth utilization or latency perceived by end user [9].

Tele:

E-mail addresses: subiphd@gmail.com

The general perception that a server demanded by more clients will be genuine and the server demanded by less clients will be doubtful is applied here. To calculate the In-degree of web server and Out-degree of clients the connected graph between clients and servers are developed [4]. Average hit percentage, average number of servers, average number of URLs', average retrieval time, average retrieval data, average frequency of URLs, and average data transfer rate are also the parameters reflecting the QoS of a proxy server and hence considered as performance measures in this study.

A graph $G = (V,E)$ consists of two objects known as set of vertices $V = \{v_1, v_2 \ldots.\}$ and set of edges $E = \{ e_1, e_2 \ldots. \}$, such that each edge e_k is identified with a pair of vertices (v_i, v_j). Graph is represented by means of a diagram, in which vertices are represented as points and each edge as a line joined with its end vertices [10] as shown in fig 1.

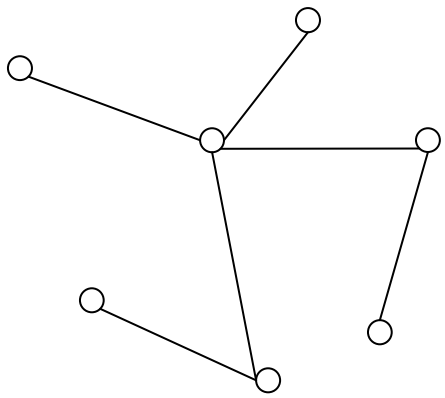

Fig 1 : A Graph with six vertices and five edges

A directed graph G is a graph with a mapping Ψ that maps every edge onto some ordered pair of vertices (v_i, v_j), directed from vertex v_i to vertex v_j . the number of edges incident out of a vertex v_i is called Out-degree of v_i and written as $d^+(v_i)$ and the number of edges incident into vertex v_i is called In-degree of v_i and written as $d^-(v_i)$. In any directed graph G the sum of all In-degree is equal to sum of all Out-degree [10].

i.e. $\sum_{i=1}^{n} d^+(v_i) = \sum_{i=1}^{n} d^-(v_i)$

A typical directed graph is shown in fig 2.

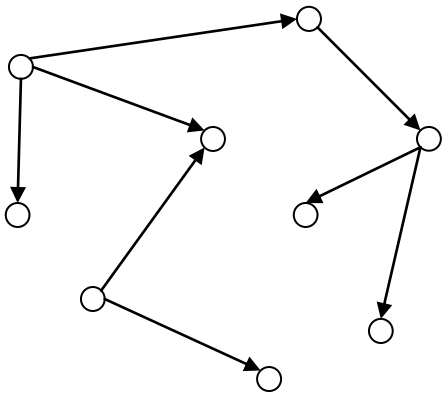

Fig 2 : Directed Graph with nine vertices and eight edges having eight In-degree and eight Out-degree

A connected graph between clients and servers are formed in which client and server are represented by vertices and URLs are represented by edge between the vertices. The thickness of the edge represents the number of URLs requested by a client to the server. After making this connected graph we identify the servers having the minimum In-degree and remove all traffic coming from these servers.

Assumption :- It is assumed that all the URLs available in the cleaned access. log file are stored into the cache of the proxy server and server memory is full. To make room available for the newly cached web object it is necessary to sacrifice some cached objects from the cache of proxy server. These objects are known as *outliers*. These outliers are identified on the basis of minimum Out-degree, minimum In-degree and minimum frequency of clients, servers and URLs respectively. It is also observed that minimum Out-degree clients, minimum In-degree servers and minimum frequency URLs contains the malicious traffic [5]. After preventing these clients, servers and URLs the abnormal web traffic will be minimized and users will get better services of Internet. Consequently the QoS will be improved.

Methodology :- The steps taken to complete the experiment are given as follows :

Step 1 : Take the log file known as access.log from the proxy server.

Step 2: To clean the log file store it into database or open it in a editor.

Step 3 : Clean the data by treating the null values and remove the URLs having the symbols like '?', '#','%' etc.

Step 4: Group the data to draw the connected graph between clients and servers by representing it as a vertex and URLs as the edge of graph.

Step 5 : Calculate the In-degree for each web server (S_{in}) and Out-degree for each client (C_{out}).

Step 6: Calculate the average hit percentage, average retrieval time, average retrieval data, average frequency of URLs, average number of servers and average data transfer rate.

Step 7: Delete all traffic from log file coming from web server having minimum(S_{in}) value.

Step : 8 Repeat step 6 & 7 up to the desired value of (S_{in})

Step 9 : Draw the graph of calculated values

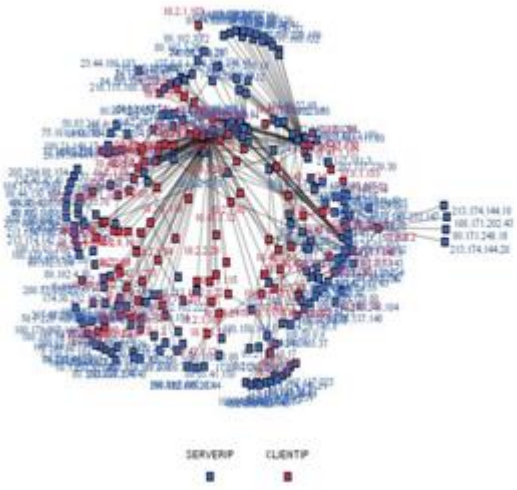

Fig 3: Connected graph between client and server (Network Model)

Experimental Data: Initially the number of records taken from the access. log file was 10,48,576. This was the two days records of one proxy server taken from one proxy server of Central Computing Facility of the University. The proxy server serves to around 3000 nodes. Following step 3 we clean the data and after cleaning the data the number of record was reduced to 6,34,930. Using these records the clients and servers are identified. The details of record are given in table 1.

Table 1 : Number of Client, Server and URLs before and after data cleaned

No. of records in log file	Clients	Server	URLs
Initially	163	12872	1048576
After cleaning	162	9991	634930

The connected graph between 162 clients and 9991 servers was developed as shown in fig 3 & 4. Fig 3 represents the network model and fig 4 represent the circular model between client and server.

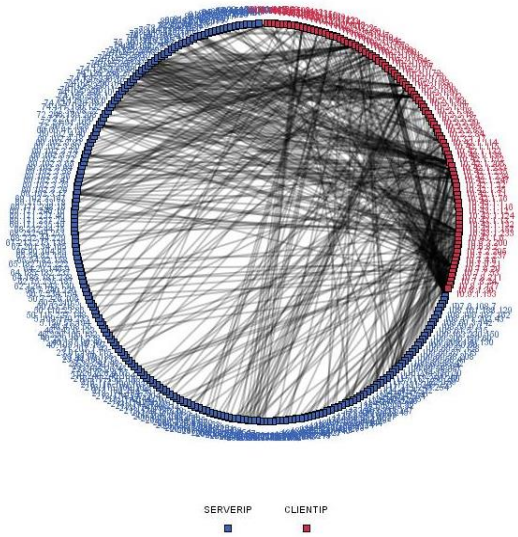

Fig 4: Connected graph between client and server (circular Model)

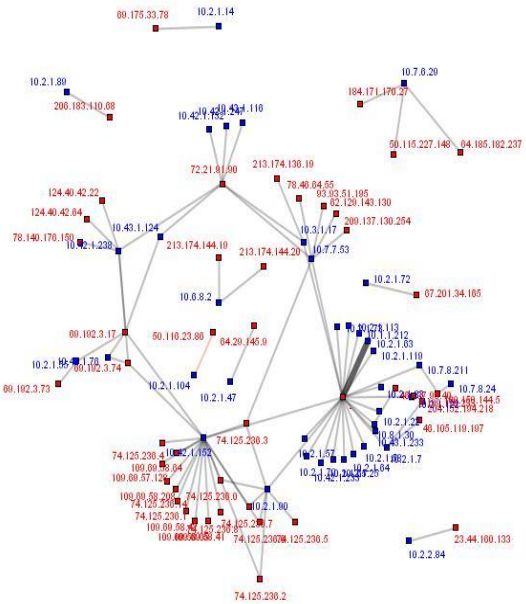

Fig 5. Connected Graph between client and server after using filtration criteria

Fig. 3 & 4 have only one graph each which represents all the connections between clients and servers. Fig. 5 has more than one graph and each graph represents the connection between clients and servers. Fig. 5 is created from fig 3 after applying the condition on number of links (URLs). The smallest graph contains only one client and one server. Some graph contains one client and more than one servers. Using the graph the servers In-degree(S_{in}) and client Out-degree(C_{out}) of each client and server is calculated.

Table 2 : Number of Server associated with S_{in} range.

Server In-degree Range (S_{in})	No. of server	Overall percentage of server
1 - 1	2511	25.13262
2 - 2	1114	11.15004
3 - 3	653	6.535882
4 - 4	449	4.494045
5 - 5	385	3.853468
6 - 6	322	3.222901
7 - 15	1459	14.60314
16 - 100	2261	22.63037
101 - 634930	838	8.387549

The table 2 represents the servers In-degree(S_{in}) range and number of server used in this range also the overall percentage of used server for each range.

Table 3. Out-degree of each client with its percentage and data used

Client Name	Total Out-degree of Client (C_{out})	Client Percentage	Data Used by Client (MB)
102163	51290	8.0780558	2332.613405
10421152	32435	5.1084372	2690.994417
107753	26339	4.1483313	1255.682919
103117	17724	2.7914888	333.8220959
10421238	16974	2.6733656	611.6369247
10431233	14681	2.3122234	131.0630608
1021104	13088	2.0613296	92.37670326
102190	12561	1.9783283	106.3026075
102155	12331	1.9421039	280.1132135
102157	11794	1.8575276	3462.634989
102173	11421	1.798781	1972.794016
1021113	11315	1.7820862	146.0707178
102131	10588	1.6675854	307.8092308
102179	10485	1.6513631	1165.750974

The table 3 represents the client name , client Out-degree (C_{out}), client percentage and data used by the client.

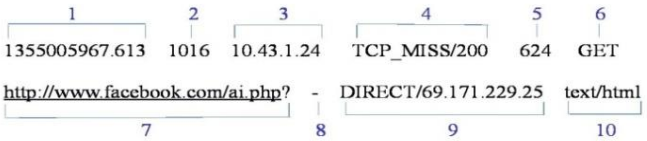

Fig 6: One record of access.log file, having ten fields

In fig 6, one record of access.log is shown where field 1 is the time stamp indicating when the request was made by the client, field 2 is the time (in milliseconds) indicating time consumed in retrieval of requested web page, field 3 is the client IP address who made the request for web page, field 4 represents the http result code and cache result, field 5 is the amount of data (in bytes) consumed by the client in retrieving the webpage, field 6 represents the data request method made to the server, field 7 indicates the address of the requested web object also known as Uniform Resource Locator (URL), field 8 shows the details about user authentication, filed 9 is the data about requested server and field 10 is the details about requested web object [7].

In our experiment field 2, field 3, field 4, field 5, field 9 are used and two more fields known as frequency(number of occurrence) of each URL and data transfer rate (data size divided by time taken) was calculated from the existing fields.

Input to the experiment:- The input to the experiment is the cleaned data of access.log file. In the client IP field all the dots are removed to perform the sorting, selecting etc. operations

smoothly and accurately. From the field 9 only server IP extracted and all the dots of server IP are removed. One more field known as the frequency of each URL is calculated and added to the input data. Two more fields known as the In-degree of server (S_{in}) and Out-degree of client (C_{out}) are also calculated and added to the input data. Finally out of 13 fields 7 fields have been chosen as input to the experiment.

The experiment is performed on the same data three times by using the different criteria each time. First time it was based on client second time it was based on server and third times it was based on URL. These three cases are known as Case 1, Case 2 and Case 3 respectively.

Case 1 (based on client) :- In this case only Out-degree of each client is considered. The lowest number of Out-degree client is identified as outlier and remove all the data related to these clients from the cache of proxy server. After removal of these outlier all the six output fields are calculated and this process is repeated 30 times.

Case 2 (based on server) :- In this case only In–degree of each server is considered. The lowest number of In-degree server is identified as outlier and remove all the data related to these servers from the cache of proxy server. After removal of these outlier all the six output fields are calculated and this process is repeated 30 times.

Case 3 (based on URL) :- In this case only the frequency of each URL is considered. The lowest number of frequency URL is identified as outlier and remove all the data related to these URL from the cache of proxy server. After removal of these outliers all the six output fields are calculated and this process is repeated 30 times. The input data for all the cases look like as shown in table 4-6.

Table 4. Sample Input (Case 1) based on Client

Client IP	Hit Percentage	Servers Used	(C_{out})	Average Time Taken	Average Data Used	Average Frequency
109127	0.004310156	82	225	62117.7153	85916.461	90.833898
10111158	0.00794822	53	236	2932.90809	19602.3842	145.88786
108172	0.008810251	58	253	9008.78939	6956.73632	90.723051
102113	0.008678755	79	275	892.055556	12719.1296	339.27104
1081214	0.004792309	93	297	15095.6768	13897.8018	286.55182

Table 5 : Sample Input (Case 2) based on Server

Server IP	Hit Percentage	Clients Used	(S_{in})	Average Time Taken	Average Data Used	Average Frequency
1652545864	0.00233771	34	160	12624.225	42153.712	27
1652545883	0.00251304	34	172	19803.308	37124.116	37.59302
2031901241	0.00127113	34	87	610.16092	535.36781	1.494252
2361194216	0.00245459	34	168	11244.631	43951.636	35.55357
2361194248	0.00222082	34	152	7294.2960	36213.407	28.92105

Table 6 : Sample Input (Case 3) based on URL

URL	Hit Percentage	No. of Server	No. of Client	Avg. Time Taken	Avg. Data Used	Avg. Freq.
0.drive.google.com:443	1.46E-05	1	1	19201	44214	1
0d90ca51-a-62cb3a1a-s-sites.googlegroups.com:443	1.46E-05	1	1	11616	4912	1
1.0.244.191:23003	1.46E-05	1	1	0	1391	1
1.165.193.190:54651	1.46E-05	1	1	0	1395	1
1.205.6.4:23843	1.46E-05	1	1	0	1387	1

Output of the experiment :- Each case of input set has one output set which contains six parameters. The seventh parameter known as average data transfer rate is calculated by dividing average data to average time. i.e.

Average data transfer rate = $\frac{\text{Average data used (i.e.Column number 5 of output tables)}}{\text{Average time taken (i.e.Column number 4 of output tables)}}$

Case 1 (based on client) :- For this case six output fields are calculated which are known as Average hit percentage, Average number of server used by client, Average number of URL requested by client, Average time taken(in milliseconds) by client, Average data (in bytes) used by client, Average frequency of URL used by client. One more field known as Average data transfer rate for client is also calculated by dividing the total number of bytes consumed by the client to the total time consumed in retrieving these bytes [6].

Table 7. Sample Output (Case 1) based on Client

Avg. Hit Percentage	Avg. No. of Server	Avg. No. of URLs	Avg. Time Taken (ms)	Avg. data used(bytes)	Avg. Frequency of URLs
0.058344	291.5975	2448.05	21445.29	179385.1	428.1555
0.059087	295.2866	2479.21	20537.3	181618.6	433.1702
0.060236	300.9805	2527.448	15825.81	184773.1	441.5274
0.06063	302.9216	2543.941	15627.27	185966.5	444.4001

Case 2 (based on server) :- For this case six output fields are calculated which are known as Average hit percent, Average number of client served by server, Average number of URL served by server, Average time taken(in milliseconds) by server, Average data (in bytes) served by server, Average frequency of URL served by servers. One more field known as Average data transfer rate for server is also calculated [6].

Table 8. Sample Output (Case 2) based on Server.

Avg. Hit Percentage	Avg. No. of Client	Avg. No. of URLs	Avg. Time Taken (MS)	Avg. data used(bytes)	Avg. Frequency of URLs
0.000929	4.641377	63.5502	32121.65	1032128	72.7601
0.001846	9.645675	126.3391	18863.34	443546.5	136.9263
0.002598	13.39674	177.8105	15609.72	147613.2	165.5725
0.003288	16.61949	225.0747	15950.72	58849.55	188.0048
0.00383	19.26951	262.1145	16341.58	42933.47	210.1735

Case 3 (based on URL) :- For this case six output fields are calculated which are known as Average hit percentage, Average number of server requested by URL, Average number of client requesting for URL, Average time taken(in milliseconds) by URL, Average data (in bytes) used by URL, Average frequency of URL requested by client. One more field known as Average data transfer rate for URL is also calculated [6].

Table 9. Sample Output (Case 3) based on URL.

Avg. Hit Percentage	Avg. No. of Server	Avg. No. of Client	Avg. Time Taken (ms)	Avg. data used(bytes)	Avg. Frequency of URL
3.16E-05	1.195915	1.327758	2639.052	100864.2	2.165829
0.000129	2.338632	3.239483	4615.819	99662.35	8.840858
0.000127	2.988024	3.712392	3615.27	27013.07	8.713638
0.000279	4.851264	6.730197	4167.948	28476.4	19.07112
0.000438	6.546986	9.588517	4072.831	23003.97	29.96689

Finally as an output we have graphs corresponding to seven attributes and three cases. In the following seven graph the x-axis is common and represents the number of iterations and y-axis represents the average HIT percentage (fig. 7), average number of server used by client(fig. 8), average number of URL used by client(fig. 9), average time in milliseconds consumed to get the web object from the server(fig. 10), average data in bytes consumed by the client(fig. 11), average number of frequency of URL requested by the client(fig. 12), average data transfer rate between client and server(fig. 13).

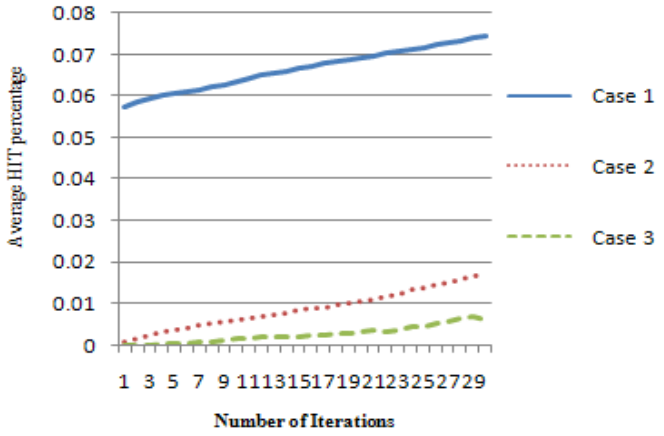

Figure 7 : Variation of Average HIT percentage Vs Number of Iterations

Fig 7 indicates that average HIT percentage is improved for all the three case. The improvement in case 1 and 2 is batter than in case 3. Imprroment in the hit percentage means the probability of availabilty of web object into cache is higher. On the basis of quantitative measure it is observed that approach applied here has better performance and this is also an indicator of improvement in the QoS [11].

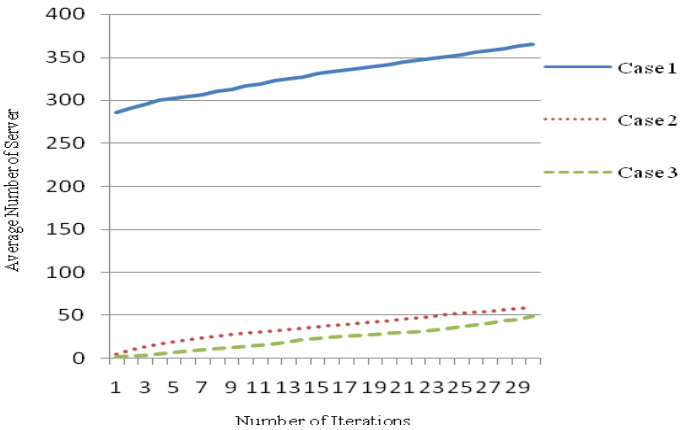

Figure 8 : Variation of Average Number of server used by client Vs Number of Iterations

Fig 8 indicates that average number of servers used by the clients is increasing. It is increased because the server having minimum In-degree is removed. The minimum In-degree servers are assumed to be involved in malicious activity therefore removal of all the traffics coming from these server will stop the abnormal web traffic passing through the proxy server. After this removal the quality of service (QoS) will also be improved.

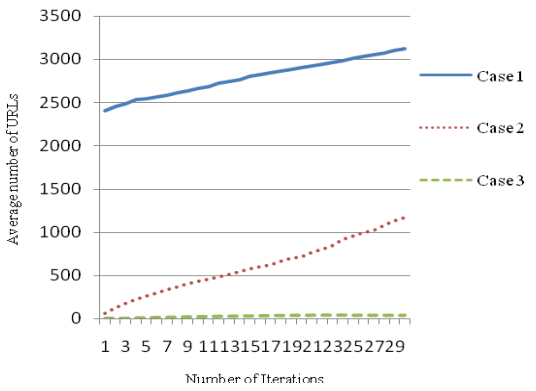

Figure 9 : Variation of Average Number of URL used by client Vs Number of Iterations

Fig 9 indicates that average number of URL used by the clients is increasing i.e. during the same time clients can access more URL. This is also a good sign to the improved QoS.

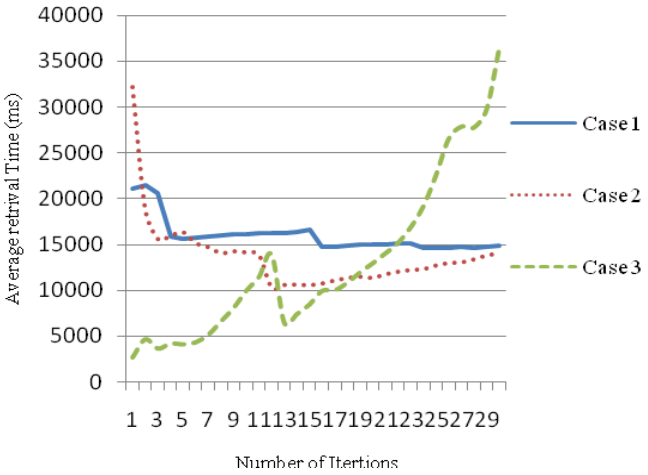

Figure 10 : Variation of Average Retrieval Time consumed to get web object Vs Number of Iterations

In fig 10 the trend of graph is not same. Case 1 and 2 are showing decrease in the web object retrieve time but case 3 shows increase in web object retrieval time. This retrieve time is measured by the proxy server in millisecond. In case 1 and 2 the web traffic associated with minimum Out-degree client and web traffic associated with minimum In-degree server are removed from the web traffic. Therefore average retrieve time goes down which is a good sign. In case 3 the average retrieve time goes high because initially all the URL having lower frequency are removed from the web traffic. Therefore average retrieve time goes high.

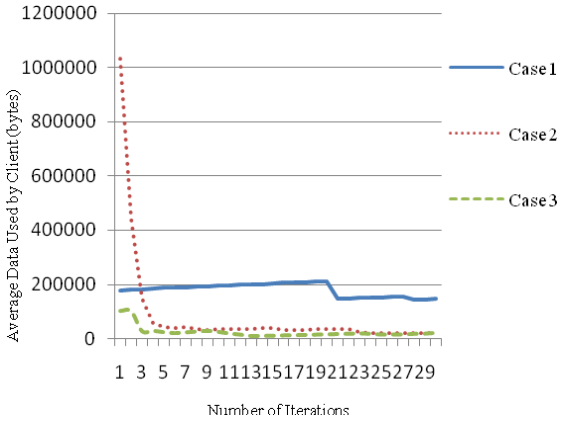

Figure 11. Variation of Average Data used by client Vs Number of Iterations

Fig 11 indicates that average retrieve data reduced in all three case. The retrieve data is measured in bytes by the proxy server.In case 2 the average retrieve data reduced upto a singificant leve but in case 1 and 3 the retrieve data is reduced very minimum which indicates that data associated with minimum Out-degree client and minimum frequency URL are also minimum in size. In case of server the minimum In-degree server delevering the significantly large data to the client which may be considered as abnormal data. Initially by removing this large data from log file the graph goes down rapidly and after a certain level it also becomes constant. Therefore this is also a proof that the server having minimum In-degree are involved in malicious activity and producing the abnormal traffic. The

prohibition of the communication between minimum In-degree server and associated client improve the QoS.

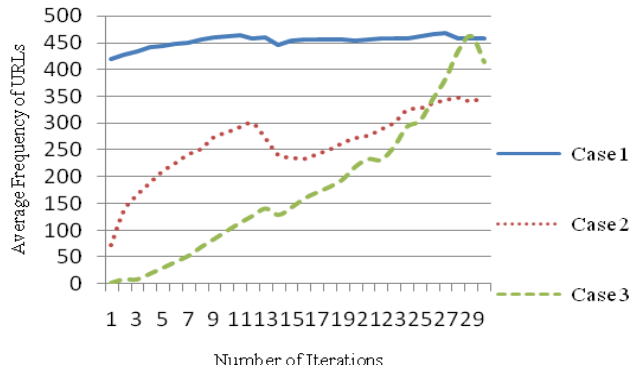

Figure 12. Variation of Average Frequency of URL Vs Number of Iterations

Fig 12 shows that average frequency of URL is increasing in all three cases. It is increased little in case 1 and more in case 2 and 3. The improvement in case 2 is not uniform from starting point to end point but over all it is significant. In case 3 the improvement is very fast because lower frequency URL are removed from first from the log file for each iteration and consequently the average frequency of URL goes high. The improvement in average frequency of URL is the proof that the request made to the minimum In-degree server is very less. User making these web request may be categorized as abnormal user. Therefore prohibition of this type of communication between client and server improve the average frequency of URL and also the QoS.

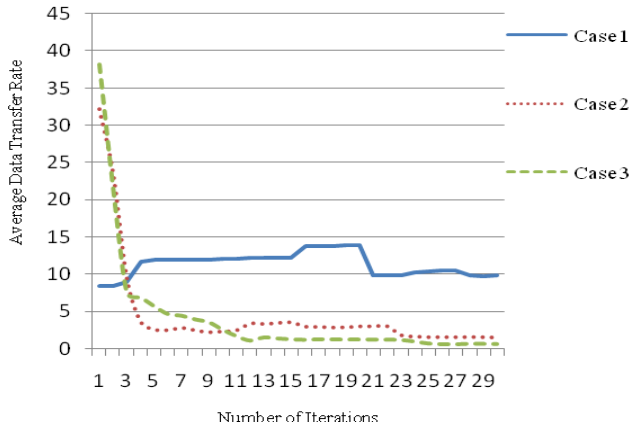

Figure 13. Variation of Average Data Transfer Rate Vs Number of Iterations

Fig 13 shows that data transfer rate improves slightly in case 1 and reduced significantly in case 2 and 3. In case 2 and 3 average data transfer rate is reduced significantly at start but after a certain level it becomes constant or reduced very minimal. This shows that the data associated with the minimum In-degree server and minimum frequency URL are bigger in size and consuming the higher bandwidth. It also creating the congestion on the network. This type of web traffic may be categorized as abnormal web traffic. Therefore stoping or removel this type of web traffic from the log file will improve the quality of serverce (QoS).

Conclusion :- The above experiment produces an evidance to the assumption that web server having minimum In-degree are not used by the normal user as a normal web server. These are the servers used by abnormal users and involved in producing the abnormal web traffic to the normal users and create congestion on the network. Consequantely the quality of service (QoS) is reduced. After identifying these minimum In-degree web server the users involved with these server are also identified and may be categorized as abnormal users. During the analysis of log files it is also observed that percentage of this type users is very small but consuming significant amount of Internet bandwidth.

To provide the better Internet service to all the user it is necessery that HIT percentage must be improved by caching new objects and removing the old cached objects from the cache to make available space for the newely cached web objects. The outliers are web objects which are removed from the cache of the proxy server. Secondely on the basis of minimum Out-degree and minimum In-degree, identify the users and web server which are involved in receving and producing the abnormal web traffic. After identification, prohibit the communication between these client and server to improve the quality of Service (QoS).

References

[1] A.C. Odinma, S. Butakov and E. Grakhov, ”Improving the browsing experience in a bandwidth limited environment through traffic management”, Information Technology for Development, Vol. 17, No. 4, page 306-318, October 2011.

[2] V Sathiyamoorthi, Murali Bhaskaran, “Data Preprocessing Techniques for Pre-Fetching and Caching of Web Data through Proxy Server”, International Journal of Computer Science and Network Security, Vol. 11 No. 11, page 92-98, November 2011.

[3]] M.K. Liu, F.Y Wang and D.D. Zeng, “Web Caching: A Way to Improve Web QoS”, J. Comput. Sci. & Technol., Mar. 2004, Vol. 19 No. 2 page 113-127.

[4] Qiang Yang, Henry Haining Zhang, Ian T.Y. Li and Ye Lu, “Mining Web Logs to Improve Web Caching and Prefetching”, Springer-Verlag Berlin Heidelberg, page 483-492, 2001.

[5] M.C. Tran, Lee Heejeong and Y. Nakamura, “Abnormal Web Traffic Detection Using Connection Graph”, Bulletin of Networking, Computing, Systems and Software, Vol 3, NO. 1, Page 57-62, 2014.

[6] Martin Arlitt, Rich Friedrich, Tai Jin, “Performance evaluation of Web proxy cache replacement policies”, ELSEVIER, Performance Evaluation 39 (2000) 149-164.

[7] Subhash Chand, Sanjay Mathur, “Squid Proxy Server Cache Management using K-means Algorithm”, International Journal of Computer Science and Information Technology, Vol. 5 (2), 2014, 1918-1923.

[8] P. Jomsri, “Improving the Performance of Proxy Server by Using Data Mining Technique”, World Academy of Science, Engineering and Technology, Vol. 7, page 1249-1253, 2013.

[9] Manuel Afonso, Alexandre Santos, Vasco Freitas, “QoS in Web caching”, Computer Networks and ISDN Systems 30 (1998), 2093-2103.

[10] Narsingh Deo: Graph Theory: with application to engineering and computer sciences, PHI

[11] Yin-Fu Huang, Jhao-Min Hsu, “Mining web logs to improve hit ratios of prefetching and caching”, ELSEVIER, Science Direct, Knowledge-Based Systems 21 (2008), 62-69.