Flexible and Fine-Grained Optimal Network Bandwidth Utilization Using Client Side Policy

Samay S. Omanwar[1], Dr. Kailas R. Patil[2], Prof. Narendra P. Pathak[3]

Abstract— According to World Wide Web Consortium (W3C) Over 3 billion people i.e, 40% of the world population has an Internet connection. Internet users prefer to access web pages that range from static to dynamic, HD videos to 3D videos. Due to the dynamic nature of the website and rich user interface a lot of network bandwidth is wasted in various web contents like advertisement, images, redirection, audios and videos. Such heavy websites introduces latency in loading of web pages due to low network bandwidth. To reduce the latency a lot of research work is proposed on prefetching and parallel loading of web page content but they do not provide personalization. In this paper, we propose a novel network bandwidth conservative technique for personalize web access that reduces web latency on low network bandwidth devices or users. In particular, our approach uses object reordering, priotarization and compression algorithm to reduce the web latency. The approach allows user to specify his preferences to improve browsing experience with filtering unnecessary advertisements and unwanted content on website and compress images and videos. Thus, low latency network user's perceived latency can be minimized and network bandwidth consumption can be reduced.

Index Terms— Bandwidth Saver, Browser, Compression, Network Bandwidth, nsiContentPolicy, Personalization, Priotarization.

1 INTRODUCTION

According to W3C number of end users browsing (3 billion) and amount of data has been downloaded is increasing exponentially. User prefers to browse, download and buffer the contents from various web services.

Web sites are evolved from static web pages into dyanamic web applications and include heavy contents. but, the per user bandwidth has not grown as per the growth of the websites. Web sites are now including contents like images, videos, audios, popups and redirection that consumes lots of bandwidth but growth of the network bandwidth is not as per requirement.

Now, dynamic web pages provide rich user interface, and includes third party contents such as advertises, pop-ups, automatic redirection, audios and videos. This consumes more user's bandwidth because, website publisher includes third party contents. Thus, it consumes more user bandwidth in loading third party contents, because web browser loads all contents. For getting web pages faster to the user, lots of research work is done with help of combining different techniques like web prefetching [5] [15] [3], content delivery networks [5], web caching [5] [17]and neural network [6]. By applying these techniques with proper collaboration, user get web pages faster, but front-end optimization is not provided by any system or framework. There are some techniques available, but they are all-or-none approach.

The detail problem is as follows.

A website developer develops a website, and for financial gain he includes various third party contents such as advertisements, audios and videos. However, the inclusion of third party contents requires more network bandwidth and increases web page loading time. Delay in loading of client side effect is perceived by the low bandwidth users. To address these problems, we develop a framework for personalize web access. In particular, we provide freedom to the user to choose the contents and its order of loading on the website. In addition to that our approach compresses image objects on webpages to save user's network bandwidth.

A nsiContentPolicy [1] interface provided by Firefox aims to solve above problems, it provides users to control loading of contents. In addition to that, we incorporate compression technique to provide compress images on web pages. This saves user's network bandwidth while browsing on the Internet.

In summary, this paper makes the following contributions:

1. We identified limitations in the traditional approach by analyzing different existing techniques and recent work.
2. We design and propose a novel approach to allow user personalization of web page object loading. In addition our approach allows compression og image object in website by using thorad party proxy server.
3. We evaluate and compare the result of websites from various categories to show effectiveness and efficiency of our approach.

The rest of the paper is organized as follows: Section II provides motivation; Section III provides detailed overview of recent developments. In Section IV provides observations from the study and Section V provides propose technique Section VI provides implementation, Section VII provides Results, Section VIII provide discussion, Section IX and Section X provide discussion and Section XI provides conclusion of the paper.

- [1]*Samay Omanwar is currently pursuing ME degree in Computer Engineering in Viswakarma Institute of Infromaton Technology, from Savitribai Phule University of Pune, India. E-mail:samayomanwar@gmail.com*
- [2]*Dr. Kailas Patil is currently professor in Department of Computer Engineering in Viswakarma Institute of Infromaton Technology, from Savitribai Phule University of Pune, India. E-mail: kailas.patil@viit.ac.in*
- [3]*Prof. Narendra Pathak is currently Associate Professor in Department of Information Technology in Viswakarma Institute of Infromaton Technology, from Savitribai Phule University of Pune, India. E-mail: narendra.pathak@viit.ac.in*

ISSN 2229-5518

2 MOTIVATION

Due to rapid development in Internet, web pages are becoming more interactive and contain hundreds of resources such as images, audios, videos and animations. But, relatively slow advancement in bandwidth affects the user's quality of service and its browsing experience in various ways. Because, the contents are enforced by the web publisher and user has no control to decide which contents to be loaded on browser.

The normal web page contains 0 to 40% dynamic contents such as images, advertises, pop-ups, automatic redirection, audios and videos.

Figure 1 shows the snaps of popular websites where we see that advertises and images covers 40% part of the web page.

Fig. 1. Regular Webpage

According to Wireshark traces 50% user's network bandwidth consumed by loading advertises only [4]. So we need a framework that allows user to decide which contents of a web page should load and which are not and provide personalized web access to the user.

3 LITERATURE SURVEY

The goal of literature survey is to study and analyze techniques which are enhancing Internet quality and user experience.

The present modernization of web browser is targeted towards faster access, better performance, and high user interface and bandwidth conservation. For this enhancement lot of techniques are carried out in back-end. Like web prefetching, web caching, content delivery network, artificial neural network etc.

Every technique has its own advantages and disadvantages. So for minimizing disadvantages, researchers combine various techniques for better performance. A lot of research work is done by combining above techniques to get web pages faster to users. The survey is divided in three categories.

A] Research work on web prefetching, web caching, and content delivery network.

B] Research regarding neural network and other recent techniques.

C] Basic survey of image compression techniques.

3.1 Research work on web prefetching, web caching, and content delivery network.

Web prefetching is a technique that predicts future requests, in 2011 by combining web prefetching technique with integrated technique, a mechanism was designed for reducing user latency [3]which predicts future requests for user or group of users. It totally depends on how many times users can access a particular page. This framework is based on the client server model and uses the technique of web caching and web prefetching. However, the limitation of this technique is that the prediction is made on the basis of, user or group of users but the scope of this is only local. It can access links, which are locally available and not globally. In 2014, for web service enhancement [15]the advancement is done in web prefetching. This technique combines FP growth rule mining concept, the weighted mining concept and Markov model. The technique takes a decision on the ratio between caching and prefetching accessed document. For better performance or response it is necessary to build a proxy server, by combining techniques of FP growth rule mining concept, the weighted mining concept and Markov model. FP growth rule mining concept is used to find frequent pages without generating candidate set. However, the limitation is that it requires some extra time to decide for sending response to the user as it evaluates a particular request from which cluster it comes and send back.

Web caching is a technique that stores frequently accessed content, but suffers the problem of disconnection with servers. In 2012 the technique called dynamic web caching [17] was proposed. In this technique, problem of frequent disconnection is overcome by using the clustering concept. However, the problem of the swamping of server occurred that is the number of users requesting at a time to one cluster.

The content delivery network is used to provide content to the host. In 2013, the technique called LACDN [5] is proposed to get pages faster to the user. This technique combines three techniques that are web prefetching; web caching, and content delivery network. Web perfecting gives hint about a document that can be prefetched. Web caching is a mechanism for web documents and temporary storage; it reduces bandwidth usage and content delivery network serve content to end user with high performance and high availability. However, prefetching policy used in this technique is based on a combination of history based and content based perfecting strategies.

3.2 Research regarding neural network and other recent techniques.

The neural network is working on prediction so by combining this technique with various existing technique user get faster access in 2013, the technique called ART1NN [18] is proposed, in this they combine ART1NN clustering algorithm with web prefetching. The advantage of this is, it predicts pages more correctly for users or group of users. The term prototype vector gives a generalized representation about web pages which are most frequently requested by users in a cluster. The main advantage of approach is to utilize better network resources by prefetching the URLs for group or cluster rather than a single host or user. However, it depends critically upon the order in which the training data are processed. The effect can be re-

duced to some extent by using a slower learning rate. In 2014 the method called Recurrent Neuro Evolutionary Technique [6] is proposed. It predicts future frame size by analyzing traffic size and historical access data, afterwards in 2014 the technique for web link prioritization [7] is proposed in this it analyses user web usage and content and decides priority for that particular user. This is achieved by combining genetic algorithm technique and different mining techniques that are web content mining, web usage mining and web structure mining. However, if the user is new on the web then this technique can't predict most relevant web pages to dedicated user. Technique needs some training data with respect to user to find more relevant web pages.

The techniques like HTTP acceleration and Green Ajax which helps to enhance the quality of Internet bandwidth. The idea behind HTTP acceleration technique [9] is to enhance Internet access for remote communities. This is achieved by introducing HTTP Performance Enhancing Proxy (HTTPEP) it improves the speed and utilizes satellite resources. And the technique called Green Ajax [16] is used to reduce web server's load and data transfer between user and web server because, it only loads a specific part of a web page. The idea of this approach is to receive signals from the web server that will trigger the web application to renew only new data from the web server so that the traffic from the server and client is reduced. But, this approach totally depends on the trigger so if trigger lost, then all updates regarding change in information also lost. The technique, called Smart Mobile Web Browsing [1], this is the framework for mobile browsing that delivers webpages to the smartphone, based on current network type and battery level. The technique, called UserCSP [13] User Specified Content Security Policies is a mechanism for browser security it protects the website from content injection attack and gives authority to user to enforce client side policies.

3.3 Basic survey of image compression techniques.[2] [12]

The need of the survey is to analyze basic techniques which are used to compress images. Basic techniques such as; delta encoding, quantization, baseline and progressive encoding. Delta encoding: The idea behind delta encoding is, it will check relative pixel of the images which are same or little different and replace it with code as an example given below. This technique used in Portable Network Graphics (PNG).

1. Remove metadata from images.
2. eg: if pixel contain value = 0000002 then on the place of six zeros we can place one A ie; 000000 = A the it becomes A2.

Similarly, in Joint Photographic Experts Group (JPG or JPEG) technique,

1. Split images into 8*8 blocks, represent each of these pixels as a delta and find similarity with the help of DCT or average caller.
2. Nearby picture pixel have similar color.

Hence, Quantization [8]: The goal of the technique is, pixel those are less significant visually that are merged with the same color. It provides lossy compression, so there are some quality issues.

Baseline encoding: Baseline means the full data of every pixel block is returned to the file one after another, it renders images top left line by line.

Progressive encoding: Progressive means only part of blocks is returned to the file and then another until the full data send. In this we see that image gradually go sharper and sharper.

Discrete cosine transform (DCT)[10,11}: The DCT helps to separate the image into parts (or spectral sub-bands) of differing importance (with respect to the image's visual quality). It is specially use for lossy compression and use in JPEG compression technique.

The significant advantage of DCT technique is that it uses quantization process as we discuss above, which is an efficient method to eliminate redundant data and also provides flexibility.

4 Our Observations

Observations are:

1. Web pages are brought faster, but there is no technique that provides front end optimization.

Because of interactive environment provided by web browsers with various inbuilt techniques such as web prefetching, web caching, content delivery network and artificial neural networks. Which provides faster access to all autonomous users, but all this technique is providing web pages with relevant search and faster rate. But this served pages has lots of unnecessary stuff like advertises, animations, various social activities, popups and unnecessary redirections. These activities are not controlled by techniques that we discuss so it is necessary to provide front end optimization and secure web computing to the user.

2. As we found from literature survey little work is done on object reordering and prioritization based on the personal preference.
3. No personalization provided on per site basis.

5 Proposed Architecture

The proposed architecture includes three parties that are web browser, third party server and main server.

The figure 2 shows an enhancement in traditional browser architecture, by adding Bandwidth Sever block.

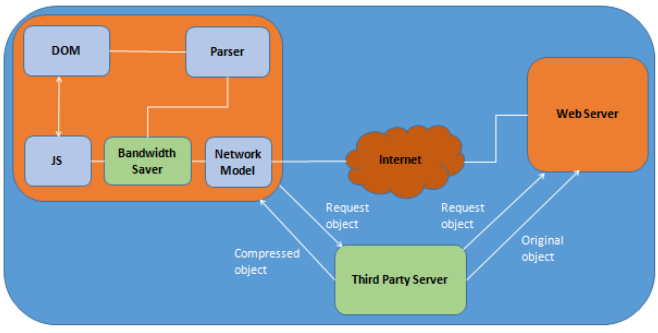

Fig. 2. General Architecture

server. It is a virtual machine that runs the JavaScript program which is embedded in HTML documents. It also understands

CSS rules for layout or to manage the web page with different specifications. Browser functionality mainly divided in to four parts.

5.1.1 Document Object Module (DOM)

DOM is nothing but the tree structure of a web page, it sort outs all HTML tags by its functionality and priority.

5.1.2 Parser

It is a process for rendering the web pages, parsing a document what actually means is, convert the document into a significant format i.e. it can be understood by code or computer program. The result of parsing is a tree of nodes that represent the structure of the document. It uses a context free grammar to parse because it's a deterministic grammar.

5.1.3 JavaScript (JS)

It is a kind of virtual machine which interprets (translates) and executes JavaScript. It allows easier testing and implementation. It also provides an access for developers to use functionality needed to control browser that is DOM handling, networking, data storage, external events, HTML 5 videos, etc.

5.1.4 Network Module

This module is responsible for periphery activity between the user and server. It manages, send and receive requests. This are browser's functionalities, we are proposing an additional functionality for efficient bandwidth access as follows.

5.1.5 Bandwidth Sever

We propose to add this functionality before network module. Because, from that position we decide which requests are allowed or not allowed.

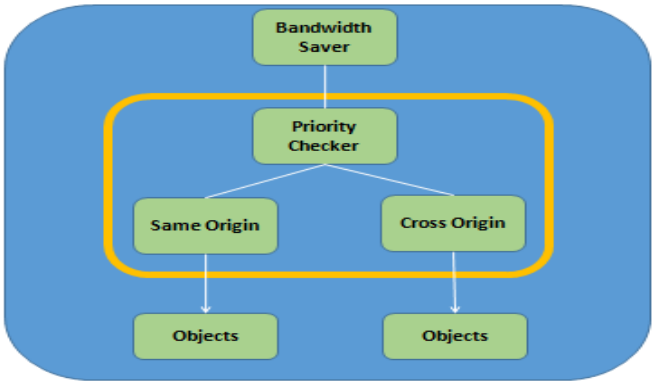

Fig. 3. Bandwidth Sever Architecture

Bandwidth saver architecture in fig 3 help users to apply priority on every website by providing them with a GUI to set priority on websites. This approach monitors the browser's internal events such as HTML parsing, same origin object and cross origin objects. Then it dynamically analyzes the content type loaded by the particular web page and the source of that content. This browser rendering information is useful for inferring nsiContentPolicy to web page automatically.

5.2 Third Party Server

The third party server focuses on compression of images. We compress images with our novel compression algorithm and serve that image to the user to save his network bandwidth. The flow is as follows.

1. Bandwidth sever sends request to third party server.
2. Third party server will forward that request to main server.
3. Main server will respond to the third party server.
4. Third party server compresses the requested image by using our proposed compression algorithm.
5. Then third party server responds compress image to user.

In this way the third party server saves user network bandwidth by providing images which are 50% compress than original.

5.3 Web Server

It is a system or a program that uses client/server model and process request via HTTP, It serves the web pages or files via World Wide Web. The important and primary function is to store, process and delivered the web pages to the client, which are requesting specific content, files or web pages.

5.2 ALGORITHM FOR BANDWIDTH SEVER

The need of the algorithm is to understand flow and working of extension. Algorithm focuses on validating users to our extension if users are not valid it allows to validate users and set their priority to websites using our approach. We explain step by step execution as follows.

Algorithm 1 : Bandwidth Sever

1) Get URL from web browser
2) Request go to the site specific profile checker
   if (User = Valid)
   Sort request with respect to user specific profile (Priority)
   I. For same origin
   A] if (Priority = Text)
   Request sends directly to the server and response send back to the client.
   B] if (Priority = image/video)
   - Request comes to the third party server.
   - Third party server sends request to the domain server.
   - Domain server sends back response to third party server.
   - Compression algorithm is applied on response.
   - Response send to client

   II. For cross origin
   - Repeat steps A and B as above.

   if (Priority = Advertise)
   - Enable advertise.

3) if (User = Invalid)
   - Save its preference priority at client side.
   - Repeat step 2 for the same.

5.3 ALGORITHM FOR COMPRESSION

The focus of the algorithm is to minimize the size of image, i.e.; n*m (dimension) to n' * m' (dimension). We achieve 50% compression of image by removing alternate rows and column

of any image pixel data.

Algorithm 2 : Compression

1. Get requested image from user.
2. Any n * m image could become n' * m'
3. Value of a depends on
   - Pixel difference threshold.
   - Number of such pixels.
4. Skip alternate row and column from raw image data.

7 IMPLIMENTATION

We did an introductory implementation for proposed system using Jetpack framework provided by Mozilla. We implement nsiContentPolicy for blocking and unblocking contents on websites. By using this we can observe the content that is being loaded into the browser. This interface is very useful for content-aware plug-in. By using this we can allow or stop user-browsed URLs. For our proposed technique we created a GUI as following fig 4.

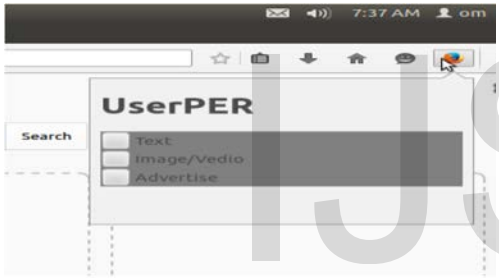

Fig. 4. GUI for extention

A nsiContentPolicy [1]have two methods shouldLoad() and shouldProcess() which are responsible for implementing nsiContentPolicy.

1. shouldLoad() : This method will call before loading the resources, it decides the resources at this location are loaded or not.
2. shouldProcess() : This method will be called once, it is used to allow a response from a server to be handled or ignore.

We can accept and reject requests by specifying its type and source. TYPE_SCRIPT, TYPE_IMAGE, TYPE_STYLESHEET etc. this are the content defined in nsiContentPolicy.

Example:-

Content type: TYPE_IMAGE = REJECT OR ACCEPT,

TYPE_SCRIPT = REJECT OR ACCEPT. In this way we define policy pragmatically, results are as follows.

After implementing nsiContentPolicy we get following results which allow user to block his contents on a website. In particular, we take two scenarios i.e; $www.facebook.com$ having multiple images on website and $www.yahoo.com$ having an images and advertise on the website. The output shows the result of blocking images and advertisements on the website.

Fig. 5. Blocking images

Fig 5 shows a scenario of web pages while user loads www.facebook.com, left image is the output of normal scenario and right image is output after implementing nsiContentPolicy which blocks images.

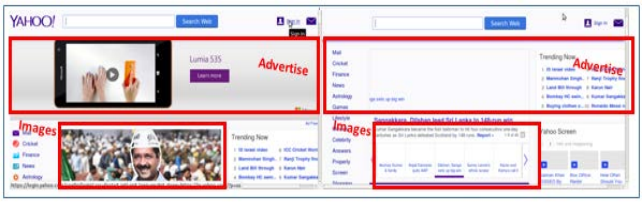

Fig. 6. Blocking images and advertise.

Fig 6 shows a scenario of web pages while user loads www.yahoo.com; left image is an output of normal scenario which display all images and advertisers. And right image is output after implementing nsiContentPolicy which blocks images and advertisers.

Fig. 7. Output of compression algorithm.

Fig 7 shows the output of the compression algorithm. It gives the size difference between two images. In output, right side image in figure is 50% compress than left one.

8 RESULT ANALYSIS

We implemented and tested the results of proposed system and compression algorithm which is develop using Matlab 2012 on PC with Intel(R) core i3 processor, four GB RAM and Ubuntu 12.04 operating system for addon sdk developement and Windows 7 for compression algorithm. These results were analyzed with respect to performance, speed and quality factors.

8.1 Performance overhead of network bandwidth saver extension

We compare and analyze the results with respect to normal loading time, load only visible area and load a web page without image and advertise.

As shown in the table 1 we calculate the loading time for 25 popular website with respect to three factors as follows.

1. Normal time to load web page.
   In this we load websites as it's own defined policy and calculate the loading time to load whole website. We compare this factor with other two factors which focuses on user defined policy.
2. Time to load web page without image and advertise.
   In this we defined policy to reject images and adverties on websites so that it only loads text.
3. Time to load web page with visible area of web page
   In this we defined policy to load only visible area on screen. So that it only loads the visible region insted of total loading.

Sr. no	Website	Normal time	With out image	With first page
1	www.google.com	1.85	1.72	1.45
2	www.facebook.com	3.29	5.97	3.24
3	www.yahoo.com	8.065	7.02	7.15
4	www.Youtube.com	1.028	2.48	2.94
5	www.baidu.com	3.38	4.61	3.60
6	www.amazon.com	2.09	2.58	3.14
7	www.twitter.com	9.29	6.37	6.94
8	www.live.com	6.95	5.20	9.92
9	www.linkdin.com	2.69	2.93	3.32
10	www.ebay.com	3.83	4.63	5.46
11	www.flipkart.com	1.93	1.72	2.93
12	www.snapdeal.com	0.95	1.53	0.76
13	www.myntra.com	1.01	0.40	0.74
14	www.being.com	2.38	1.21	0.59
15	www.tumbler.com	8.51	5.71	7.11
16	www.makemytrip.com	6.96	4.97	4.50
17	www.apple.com	2.58	1.49	3.51
18	www.msn.com	2.94	4.09	4.64
19	www.wordpress.com	11.03	7.32	7.45
20	www.ask.com	5.37	0.76	2.39
21	www.hotmail.com	8.64	7.00	4.33
22	www.paypal.com	7.49	10.46	4.65
23	www.microsoft.com	7.15	4.01	2.87
24	www.gmail.com	5.75	2.45	2.70
25	www.rediff.com	1.70	1.57	0.91

Table. 1. Analysis of Content Loading of Popular Website

8.2 Quality measure of compression algorithm

We compare and analyze the results with respect to overhead and quality of image. The table 2 shows the factors we are considering for measuring quality of image.

The quality measure factors are as follows:

1. Compression ratio (CR): It is the difference between original image size and compress image size.
2. Peak signal-to-noise ratio (PSNR): It is used for finding the ratio between maximum power signal and the corrupting power signal with respect to orignal one. PSNR is easily defined by mean square error. It is most commonly used to measure or calculate the quality of reconstruction of lossy compression.

$$PSNR(x,y) = 10 * log10\left(\frac{\text{MAX}\,(\text{MAX}(x)\,,\text{MAX}\,(y))^2}{|x-y|^2}\right)$$

3. Root Mean squared error (RMSE): It is the average of sequence of errors. In particular, it is the difference between estimator and what is estimated.

Following table shows the comparison of existing DCT technique [10] and our propose technique. We compare three stranded images which are in matlab as listed below.

Test Image	DCT			Proposed Technique		
-	PSNR	RMSE	CR	PSNR	RMSE	CR
Lena.tif (256 X 256)	27.25	11.06	11.33	30.12	7.94	50
Cameraman.tif (512 X 512)	31.23	6.99	13.65	23.70	16.64	50
Mandrill.tif (512 X 512)	26.78	11.67	8.35	21.43	21.62	50

Table. 2. Result analysis of compression technique

We say that our compression ratio is more than existing system but quality is affected.

8.3 Performance overhead of compression algorithm

The time to execute an compression algorithm is approximate 0.50 ms. We calculate this by using tic () and toc () functions provided by Matlab 2012. Hence, the additional time required to provide images on web page is time required to compress an image and sending and receiving time overhead of requested image.

8.4 Network bandwidth saved using compression algorithm

We are providing 50% compression to the user in other words we say that the compression ratio is directly proportional to network bandwidth. i.e, lesser a size of an image more a network bandwidth saves.

9 FUTURE WORK

We did an introductory implementation of proposed system which allows users to block or unblock the contents on websites. In addition to that we also develop an image compression algorithm which provides 50% compression of any image.

Now we are focusing on further developments such as, the object reordering and priotarization. After implementing this, we are able to provide personalize web access to the users.

10 DISCUSSION

After implementation of complete proposed system it performs or takes following actions:

1. If the website has predefined user priority rules then browser renders contents with respect to user rules.
2. If the website hasn't had a predefined user priority rule, then our extension will pop out a GUI for setting priority to the website.
3. If the user doesn't want to set the priority to a particular website then it loads the contents as website defined policy.
4. The extension is also responsible for distinguishing the

same origin and cross origin contents by evaluating its content type and source from where it loaded.

5. The extension allows resetting priority of website.

11 CONCLUSION

In the end, we conclude that recently there are many developments in web technology and techniques related to the Internet. As most of the research work is done on how to get relevant web pages faster to the user related search query. For this, researcher combines various techniques like web prefetching, web caching, content delivery network, artificial neural network as we discuss above but, no one technique provides user personalization or front end optimization. Hence we develop a framework for user personalization where users can specify its preference on per site basis. And in addition to that we provide 50% compress image than original on web page which are requested by the user. In short, we can say that users can paint his website and save its network bandwidth while browsing websites.

REFERENCES

[1] Mozilla Devloper Network link. https://developer.mozilla.org/en-us/

[2] Wikipedia for introductory study of compression techniques. https://en.wikipedia.org.

[3] N. Ahmad, O. Malik, M. ul Hassan, M.S. Qureshi, and A. Munir. Reducing user latency in web prefetching using integrated techniques. In Computer Networks and Information Technology (ICCNIT), 2011 International Conference on, pages 175–178, July 2011.

[4] A. Albasir, K. Naik, and T. Abdunabi. Smart mobile web browsing. In Awareness Science and Technology and UbiMedia Computing (iCAST-UMEDIA), 2013 International Joint Conference on, pages 671–679, Nov 2013.Towards Optimal Network Bandwidth Approach Using Client Side Poicy

[5] L.R. Ariyasinghe, C. Wickramasinghe, P.M.A.B. Samarakoon, U.B.P. Perera, R.A.P. Buddhika, and M.N. Wijesundara. Distributed local area content delivery approach with heuristic based web prefetching. In Computer Science Education (ICCSE), 2013 8th International Conference on, pages 377–382, April 2013.

[6] R. Arshad, G.M. Khan, and S.A. Mahmud. Smart bandwidth management using a recurrent neuro-evolutionary technique. In Neural Networks (IJCNN), 2014 International Joint Conference on, pages 2240–2247, July 2014.

[7] K. Chaudhary and S.K. Gupta. Prioritizing web links based on web usage and content data. In Issues and Challenges in Intelligent Computing Techniques (ICICT), 2014 International Conference on, pages 546–551, Feb 2014.

[8] P.C. Cosman, K.L. Oehler, E.A. Riskin, and R.M. Gray. Using vector quantization for image processing. Proceedings of the IEEE, 81(9):1326–1341, Sep 1993.

[9] P. Davern, N. Nashid, A. Zahran, and C.J. Sreenan. Http acceleration over high latency links. In New Technologies, Mobility and Security (NTMS), 2011 4th IFIP International Conference on, pages 1–5, Feb 2011.

[10] A.H. Karode Dipeeka O. Kukreja, S.R. Suralkar. Performance analysis of various image compression techniques, 2012.

[11] M.M. Dixit and Priyatamkumar. Comparative analysis of variable quantization dct and variable rank matrix svd algorithms for image compression applications. In Computational Intelligence and Computing Research (ICCIC), 2010 IEEE International Conference on, pages 1–5, Dec 2010.

[12] CTO of the Web Experience BU. Guy Podjarny. Image compression tecniques survey by Akamai. https://www. youtube.com/watch?v=k4Xkpawt1EU.

[13] Frederik Braun Mark Goodwin Zhenkai Liang Kailas Patil, Tanvi Vyas. Poster: Usercsp- user specified content security policies, 2013.

[14] M. Thomson M. Belshe Twist, R. Peon and A. Melnikov. Hypertext transfer protocol version 2.0 draft-ietf-httpbis-http2-07., July 2013.

[15] M.B. Pal and D.C. Jain. Web service enhancement using web pre-fetching by applying markov model. In Communication Systems and Network Technologies (CSNT), 2014 Fourth International Conference on, pages 393–397, April 2014.

[16] R. Sanjaya. Web traffic reduction for infrequent update application using green ajax. In Information Management and Engineering (ICIME), 2010 The 2nd IEEE International Conference on, pages 170–176, April 2010.

[17] R. Tiwari and N. Kumar. Dynamic web caching: For robustness, low latency amp; disconnection handling. In Parallel Distributed and Grid Computing (PDGC), 2012 2nd IEEE International Conference on, pages 909–914, Dec 2012.

[18] H.K. Yogish and G.T. Raju. A novel art1nn clustering and pre-fetching technique for reducing web latency. In Computational Intelligence and Communication Networks (CICN), 2013 5th International Conference on, pages 327–330, Sept 2013.