**IMPACT: International Journal of Research in Engineering & Technology (IMPACT: IJRET)**
**ISSN(E): 2321-8843; ISSN(P): 2347-4599**
**Vol. 2, Issue 6, Jun 2014, 165-174**
**© Impact Journals**

# WEB CONTENT AGING AND FILTERING OF STATIC HTML OBSOLESCENCE *(GLOBAL CONTENT DELIVERY PLATFORM FOR KNOWLEDGE CREATION AND DISSEMINATION)*

**MITHILESH KUMAR MISHRA[1] & ANURIKA VAISH[2]**

[1]Research Scholar, UP Rajarshi Tandon Open University, Allahabad, Uttar Pradesh, India

[2]Associate Professor, Indian Institute of Information Techonology, Allahabad, Uttar Pradesh, India

## ABSTRACT

This paper illustrates the diversity in the publication and delivery of content of any website. The issue of web content obsolescence [1] is growing with reach and popularity of web technology. One of various kind of web obsolescence is time based content obsolescence There have been several approaches to curb the obsolescence in the cyber space, such as user feedback based marking of obsolete content and use of language processing techniques. In this paper we have a new method of detecting and filtering of web obsolescence from Internet traffic by defining age of web content at paragraph level using newly introduced two attributes. Proposed content obsolescence detection/filtering technique will reduce the effort of website designers. By introducing such content aging techniques in markup language, we can make obsolescence detection and filtering more automated that eventually mitigates the manual work of updating the website on a regular basis [2]. In this paper a framework has been designed to demonstrate the modus operandi of content filtering on the basis of the age of web content in a static HTML [3] document file, present in the web root of any web server.

**KEYWORDS:** Web Content, Framework, Obsolescence, Content Aging, Filtering, Future Web Authoring

## INTRODUCTION

System efficiency is observed to be meliorating with rich content filtering capability. Various attributes mainly responsible for content filtering are audience, the content itself, style sheets, artifacts which are already established while other potential attributes not evidenced so far are also available. Time is one of those potential attributes as Aging techniques are applied on everything to uncover important features. So, it can also be used in filtering the content of websites or real time systems to make them more productive, progressive and vigorous in nature. Web sites often have content which is relevant for a specified period and should not appear after a particular time stamp. For example, admission notice for a university should be displayed during a specified period of time after which it should not be available in the web site. To deal with such situations the web pages have to be updated regularly.

Using the framework discussed in this paper, the age of content available may be defined in web pages so that a particular content is visible for a specified duration and when the mentioned timestamp is over, the content is not displayed without any modification by an administrator. The content delivery period could be modified by the administrator without any code rework.

To illustrate the importance of aging, it is important to identify the scenarios in which aging techniques can be used. As an example, the link for online application for an exam has to be available for a defined duration. In this case the web site has to be updated timely so that the specified timeline is adhered. There is no method for future programing i.e the content is available in the web site but will be displayed only when the defined period starts. Such a system is discussed in this paper.

## OVERVIEW OF THE SYSTEM

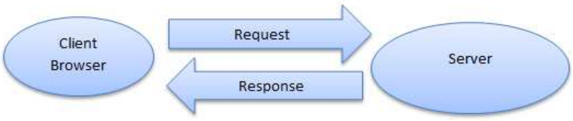


**Figure 1: Client/Server Architecture [4]**

Above figure shows the client server architecture. When a client browser request for any static html file say Index.html, the request is listened and processed by the server.

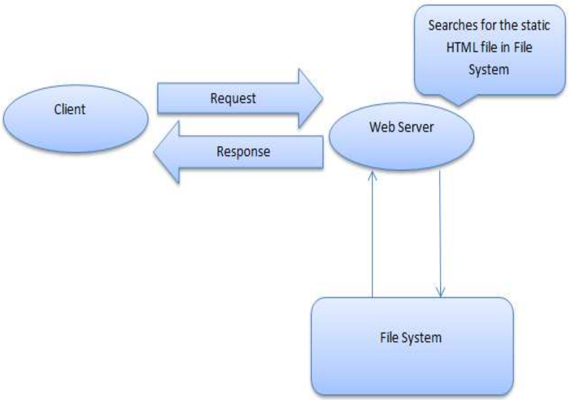


**Figure 2: Static HTML Delivery Mechanism [5]**

When a client requests for a static content to a sever, the server searches for the requested file in the server's local file system and if the file is found the requested file is served to the client else error code 404 (file not found) **[6]** is sent as a response.

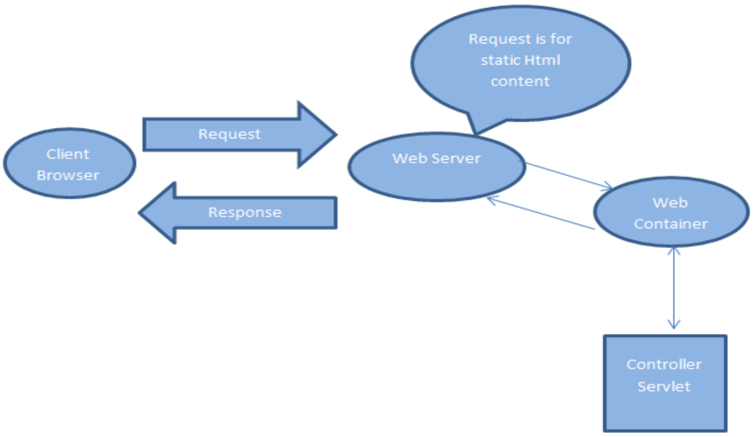


**Figure 3: Proposed Framework**

The idea of the framework is shown in the above figure. Whenever a request for dynamic content comes it is forwarded to the web container for handling the request. The container creates **Request [7]** and **Response [8]** objects, finds the servlet to be invoked, spawns a new thread for the request and passes the **Request** and **Response** object to the servlet. The servlet then processes the request and generates a response which is passed to the web server via web container and the client receives the corresponding response.

Our framework is based on above concept of serving dynamic content **[8]**. Every request to web server is forwarded to a controller servlet **[9]** which then invokes Model **[9]** class for the implementation of the required logic.

## DETAILED DESCRIPTION

HTML is a markup language and is used for web page designing. It consists of various tags like <HTML> <BODY> <TABLE> <P> **[10]** etc. These tags are used for different markup purposes. HTML tags have many attributes for changing the default rendering properties of web content inside the web browsers.

The main idea of **Content Obsolescence** is to define new attributes in the tags so that the age of content could be specified. In our case we have used <P> tag for paragraph declaration and defined new attributes **“pubDate”** to define date of publication of web content in an HTML document and **“expDate”** to define expiry of web content published in an HTML document using a <P> tag. This way we have defined age of web content under any <P> tag.

Now using this we can create a paragraph which is to be displayed in future. So, at the time of designing of web page we need to know what the valid time for a particular paragraph is, the need to update or remove the paragraph form the web page is not there anymore. The content could be redisplayed just by modifying dates in the new attributes to meet the age requirement. This facilitates the automated delivery of valid static HTML content over Internet.

**Example**

```
 <P pubDate=”24-Aug-2013” expDate=”24-Sep-2013”>
         This information is for a limited period only.
</P>
```

would mean that the text

**This information is for a limited period only.**

should be delivered by delivery platform between 24-Aug-2013 to 24-Sept-2013 only. Beyond this period delivery platform should automatically filtered the content from the response. However,

```
<P pubDate=”20-Nov-2013”>
        Perpetual information
</P>
```

will not be filtered by delivery platform as there is no expiry date set for this paragraph.

Under the proposed framework, whenever a request comes to our web server it is handled by a controller servlet which inspects the age of all paragraphs in the requested HTML file and generates a new HTML file in cache memory, filtering all the paragraphs which are obsolete according to current system date on web server. All the other tags present in the requested HTML file will be unaffected. This way our proposed framework ensures delivery of only valid content based on their age and ensures knowledge creation and dissemination more efficient and effective over Internet.

## ANALYSIS

The proposed framework requires an algorithm that scans the file once in order to find the <P> tags with the proposed attributes. The scanning requires O(n) **[11]** time whereas removing the obsolete data requires time of O(1). Size of the file is directly proportional to no. of packets n for transmission i.e if size of the file increases then n will also increase which in turn increases the time to scan the file. The graph in Figure 1 demonstrates the same.

Performance **[12]** can be measured in many ways, including transit time **[13]** and response time **[14]**. Transit time is the amount of time required for a message to travel from one device to another. Response time is the elapsed time between a request and a response. The performance of a network depends on a number of factors, including the number of users, the type of transmission medium, the capabilities of the connected hardware, type of content and the efficiency of the delivery software.

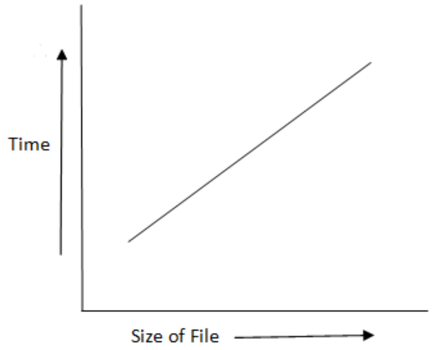


**Figure 4: Time vs Size**

Performance is often evaluated by two networking metrics throughput **[15]** and delay **[16]**. We often need more throughputs and less delay.

Assume that we have an HTML file containing the mandatory data (say x Bytes) as well as expired or obsolete data (say y Bytes). So, for delivery we have HTML content of size z bytes which is equivalent to **x + y** Bytes. Assuming, that Average packet size is p bytes, total no. of packets that are to be sent in order to send the complete HTML file in response would be z/p bytes. However, if we apply our framework, the total no. of packets that are required to be sent will be x/p bytes thus a saving of (z-y) bytes per request is achieved. The reduction in disseminated packets is (y/p) packets per request.

It is clear from above discussion that the saving in bandwidth utilization [**17]** (in terms of number of packets sent) would be more in case our framework is integrated with any delivery platform. The efficient bandwidth utilization in turn would increase the throughput of the system.

The graph in Figure 2 clearly shows that the proposed framework definitely gives better results if we compare this to the present content delivery mechanism. Here, α is the threshold value of number of packets sent after which the congestion **[18]** grows in the system. As, we are removing the obsolete data from our file, it decreases the file size that is to be sent which in turn reduces the number of packets to be sent and eventually reduces the demand of bandwidth in content delivery.

In our case the delay will heavily depend on the filtering time required by our module (assuming other conditions ideal).

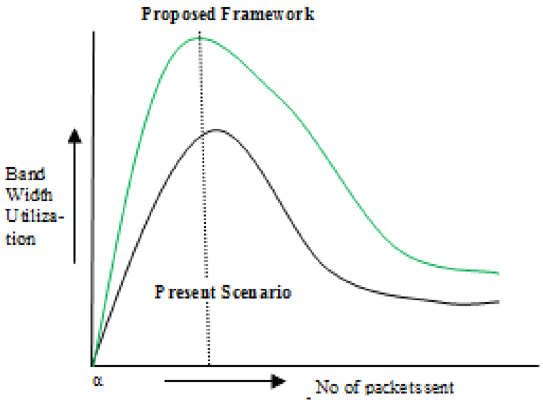


**Figure 5: Bandwidth Utilization**

In normal situation the total time elapsed from the time user makes a request is:

$T_{1} = T_{response} = T_{request}+T_{processing}+T_{delivery}$

However, in proposed framework it is

$T_{2}=T_{response}= T_{request}+T_{processing}+T_{filtering}+T_{delivery}$

$T = T_2 - \triangle T_1$ and the challenge is to$\triangle$keep T minimum

Assuming $\mathbf{T_{request}}$ as **constant,** the delay experienced by the user depends mainly on the time in filtering whereas delivery time reduces due to less number of packet to delivery over Internet. i.e $\mathbf{T_{filtering}}$ and $\mathbf{T_{delivery}}$ are the main factors which determine the overall experience of web content consumers ($\mathbf{T_{response}}$) over the Internet.

As discussed above, the response time of a website using our framework depends mainly on the time taken by the filtering technique. In case when significant content is detected as obsolete and filtering is done by the framework, the total amount of data sent to the client (browser) as a response reduces, thus the cost in terms of time invested in detecting and filtering obsolescence in web content at the time of delivery will be much less than the gain in terms of number of packets sent over Internet to the content consumer.

Let the user web surfing experience in terms of response time be denoted by $W_x$, then following cases may arise:

$T_{filtering} > W_x$

i.e. poor performance due to proposed mechanism of detection/filtering of obsolete content

$T_{filtering} = W_x$

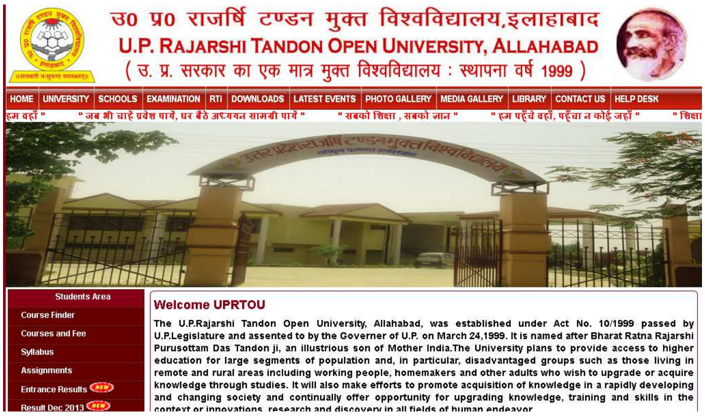
i.e. same performance with proposed mechanism of detection/filtering of obsolete content

$T_{filtering} < W_x$

i.e. Good performance with proposed mechanism of detection/filtering of obsolete content

In this case the framework will disseminate desired knowledge and information stored in the form of static HTML file efficiently and effectively for which currently server-side scripting is used by the web authors having good scripting and programming skills.

A test case **[19]** demonstrates the above claims when proposed framework got implemented on a web server and the website got evaluated over Internet.


उ० प्र० राजर्षि टण्डन मुक्त विश्वविद्यालय,इलाहाबाद
U.P. RAJARSHI TANDON OPEN UNIVERSITY, ALLAHABAD
( उ. प्र. सरकार का एक मात्र मुक्त विश्वविद्यालय : स्थापना वर्ष 1999 )
HOME
UNIVERSITY
SCHOOLS
EXAMINATION
RTI
DOWNLOADS
LATEST EVENTS
PHOTO GALLERY
MEDIA GALLERY
LIBRARY
CONTACT US
HELP DESK
हम वहाँ "
" जब भी चाहें प्रवेश पायें, घर बैठे अध्ययन सामग्री पायें "
" सबको शिक्षा , सबको ज्ञान "
" हम पहुँचे वहाँ, पहुँचा न कोई जहाँ "
" शिक्षा
Students Area
Course Finder
Courses and Fee
Syllabus
Assignments
Entrance Results NEW
Result Dec 2013 NEW

Welcome UPRTOU

The U.P.Rajarshi Tandon Open University, Allahabad, was established under Act No. 10/1999 passed by U.P.Legislature and assented to by the Governer of U.P. on March 24,1999. It is named after Bharat Ratna Rajarshi Purusottam Das Tandon ji, an illustrious son of Mother India.The University plans to provide access to higher education for large segments of population and, in particular, disadvantaged groups such as those living in remote and rural areas including working people, homemakers and other adults who wish to upgrade or acquire knowledge through studies. It will also make efforts to promote acquisition of knowledge in a rapidly developing and changing society and continually offer opportunity for upgrading knowledge, training and skills in the context or innovations, research and discovery in all fields of human endeavor

**Figure 6: Homepage of Test Website**

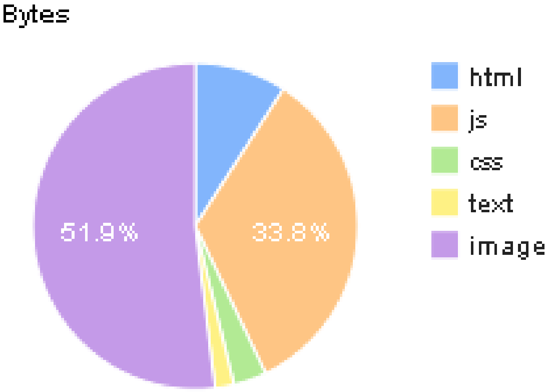


**Figure 7: Breakup of Content of above Website**

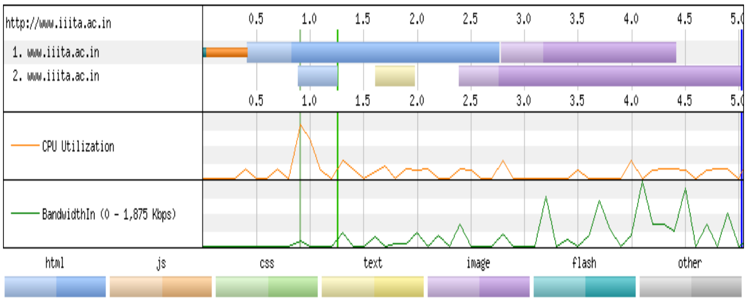


**Figure 8: Delivery Time Parameter Values**

Time taken in delivery of whole web content of homepage of this website is 6.434 seconds; however, through our proposed delivery platform time taken is considerably less i.e. 5.020 second as the few bytes got filtered due to obsolescence. For testing image was put inside a paragraph and declared obsolete by setting expDate=”” attribute of

<P> tag. Here, CPU utilization and during delivery total bandwidth consumed is also depicted in the graph. For this testing a circuit of 1 Gbps for Internet access has been used.

## BENEFITS OF THE FRAMEWORK

### Benefits to Web Users (Content Consumers)

- The users will always get the right content
- Obsolete contents are detected/filtered automatically by the delivery platform.
- Chance to see filtered obsolete content
- Timeline navigation of web content
- Quicker response time compared to existing delivery platform

### Benefits to Web Authors (Content Suppliers)

- Deciding and defining the age of web content in HTML files using pubDate and expDate attributes with <P> tag.
- Putting automatically obsolete content offline
- Putting obsolete content online by redefining the expDate attribute of <P> tag
- Providing timeline navigation to content consumers
- Declaring detectiong/filtering of content through use of suitable attribute in DOCTYPE delaration or use of <HTML ContentAging=”yes”> tag
- <HTML Content Aging=”no”> will bypass scanning of HTML file for detection/filtering of obsolescence content from the response at the time of delivery of knowledge bearing content that never expires
- Auto alert in the form of email or SMS by delivery platform to web authors intimating the content to expire or expired content on the website
- Futuristic web authoring where pubdate and expdate both are ahead can be achieved under this framework

## FUTURE SCOPE

- We can use similar attributes for other tags so that age of every HTML element can be defined for detection and filtering from the delivery.
- To make delivery platform more effective a background demon process can be utilized to scan static HTML document files periodically for obsolescence detection and regeneration of cache HTML document file with valid content only for delivery over Internet.
- We can include some header information in the pages to notify that the page is of temporal nature and contains content that will be expired in due course of time depending on their ages. This way proposed delivery platform will automatically enforce desired scanning for obsolescence detection and filtering of web content. This would also help avoid time taken in detection/filtering at each hit.

- Since the obsolete content remains in the static HTML document web authors can facilitate web users a timeline navigation mechanism using pubDate, expdate with respect to System Date as below:

  pubDate <= sysDate <= expdate

- Presently pubDate and expdate are defined with date value only using format dd-MMM-YYYY, however, time value can also be incuded for fast moving contents
- With evaluation of system date against pubDate and expDate dynamism up to some extent can be achieved even without use of client/server-side scripting such as JavaScript or VBScript. For example, to greet visitor of website, one can define two paragraphs:

  ```
  <P pubDate=”00:00:00” expDate=”12:00:00”>Hello! Good Morning</P>
  ```

  and

  ```
  <P pubDate=”12:00:01” expDate=”24:00:00”>
      Hello! Good Evening
  </P>
  ```

- Browser side module can also be used in the form of plug-ins for detection and filtering of obsolete content delivered by any delivery platform.
- Cache management features of proxy servers and browsers can be optimized by purging the obsolete contents stored in these caches periodically **[20]**.
- Integrity of web data can be ensured with filtering of obsolete content from the web.

## ACKNOWLEDGEMENTS

The implementation of this concept was done with the help of our two PG students Mr. Prashant Harbola and Mr. Kumar Vaibhav along and two UG student Mr. Jatin Mehta and Mr. Aditya Dev Chaturvedi. Their programming skill made our efforts demonstrative for data collection and analysis in support of our claim. Their effort is heartily appreciated by authors of this paper.

## CONCLUSIONS

It can be concluded that the proposed framework reduces Internet traffic by filtering obsolete content from delivery. It also alerts web authors on obsolescence of web content so that timely updation of content can be ensured. For delivery content can be specified by “pubDate” and “expDate” using suitable HTML tag, a futuristic web authoring can be achieved under proposed framework. Detection of expiry date of content by proxy servers, web browsers and search engines eliminates chance of delivery of obsolete content over Internet. This also promises green content and green web delivery. Performance of delivery of bigger size pages with high nature of obsolescence in content is better than the smaller size pages. This framework redefines the content delivery over Internet in order to address the problem of web obsolescence.

## REFERENCES

1. http://christianaboutros.wordpress.com/2013/07/01/planned-obsolescencewhen-will-it-stop/
2. http://webdesign.about.com/od/htmlxhtmltutorials/p/what-are-markup-languages.htm
3. http://www.w3schools.com/html/html_intro.asp
4. http://www.utdallas.edu/~chung/SA/2client.pdf
5. http://computer.howstuffworks.com/web-server6.htm
6. http://www.404errorpages.com
7. http://docs.oracle.com/javaee/6/api/javax/servlet/http/HttpServletRequest.html
8. http://docs.oracle.com/javaee/5/api/javax/servlet/http/HttpServletResponse.html
9. http://www.tutorialspoint.com/struts_2/basic_mvc_architecture.htm
10. http://www.w3schools.com/tags/tag_p.asp
11. http://www.csd.uwo.ca/courses/CS1037a/notes/topic13_AnalysisOfAlgs.pdf
12. http://technet.microsoft.com/en-us/library/cc976785.aspx
13. http://pubs.research.avayalabs.com/pdfs/ALR-2003-051-paper.pdf
14. Network Response Time for efficient interactive use.

    [http://www.ewp.rpi.edu/hartford/~rhb/cs_seminar_2004/Addendum/flemming.pdf]
15. Throughput analysis of IEEE 802.11 wireless networks with network coding [Bo Kyung Jang, Dept of Electrical Engineering, Kyung Hee University, Youngin, South Korea]
16. Delay Based Network Utility Maximization [Neely, MJ, University of Southern California, Los Angeles, CA, USA]
17. Eureka: A methodology for measuring bandwidth usage of networked applications [Vaishnavi, I.; Centrum Wiskunde en Inf., Amsterdam, Netherlands]
18. http://www.nyu.edu/classes/jcf/g22.2262-001_sp10/slides/session9/NetworkCongestion-Causes-Effects-Controls.pdf
19. http://www.webpagetest.org
20. Cao P, Zhang J, Beach K. Active cache: caching dynamic contents on the Web. In: Proceedings of IFIP international conference on distributed systems platforms and open distributed processing, 1998. p. 373-88 [http://www.cs.wisc.edu/~cao/papers/active-cache.ps].