International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878, Volume-8 Issue-3, September 2019

Enhanced Methods of Mobile Cache Sharing and Pre-fetching for Required Web Contents

P. Amudha Bhomini, Jayasudha J.S

***Abstract:** Availability of content over the web is increasing exponentially. The demand for content by users is also increasing rapidly. The problem of making the right content available to user at the right time will continue to be a crucial issue. As variety of contents are available and variety of users are involved, there is no single way of matching the availability versus need and deliver content instantly especially in a limited mobile environment. Hence a hybrid method is proposed in this paper by combining the different techniques such as caching, pre-fetching and cache sharing with noise reduction to improve the overall performance of mobile for optimal cache memory utilisation, efficient bandwidth utilization, network traffic reduction and latency reduction. Efficiency of mobile caching and pre-fetching is improved using Enhanced Bloom Filter technique and data is shared among cooperative users by establishing a voluntary hub. The unwanted contents in the web page can be considered as noise which is removed when storing the web pages in cache or pre-fetch area. The success of the proposed method greatly depends on the hit ratio of contents rendered locally rather than getting it from server. In order to reduce server hits, sharing the contents of cache and pre-fetch area amongst mobile users is effective. Whenever any user requires new content, even if it is not available in browser cache or local cache of that user, the content can be rendered from the cache or pre-fetch area of collaborative mobile users rather than hitting the server. This hybrid cooperative cache sharing and pre-fetching for accessing the required contents improve the overall performance and hit ratio than the existing methods.*

***Keywords:** bandwidth, cache sharing, cooperative users, latency reduction, mobile caching, pre-fetching.*

I. INTRODUCTION

The caching and pre-fetching techniques have been used since the client server era. The number of users accessing the internet is increasing exponentially and their dream is to access the data instantly. With content explosions, implementation of caching alone was found to be insufficient, and caching and pre-fetching were combined for better content delivery. Even the existing modern techniques were found to be not very effective for users who keep browsing new contents. With MANET there is a huge opportunity of collaboration amongst network nodes.

Manuscript published on 30 September 2019
* Correspondence Author
P. Amudha Bhomini*, Department of Computer Applications, Nesamony Memorial Christian College, Marthandam – 629 165, India. Email: amudhabhomini@yahoo.co.in
Dr. Jayasudha J.S, Department of Computer Science and Engineering, Sree Chitra Thirunal College of Engineering, Trivandrum, India. Email: jayasudhajs@gmail.com

Each device as a node can share and collaborate with each others, especially in content sharing to reduce network traffic, sever hits and latency. However, if these collaborative techniques were not implemented wisely, it may have a contrary effect as well.

Another area of improvement is to remove noise contents while caching/pre-fetching. If contents are stored and shared without noise, the hit ratio of caching/pre-fetching can be increased and the performance of cooperative users' content sharing can be improved.

There are many research works on caching, pre-fetching and content sharing to improve hit rate which reduces latency and network traffic. Some methods adopted the sharing of cache/pre-fetch area without any central system to monitor it. Alternatively, nodes may collaborate amongst themselves using a central node to facilitate/manage the caches. In the proposed method, cooperative users are established and cache/pre-fetch area contents are shared between them using a hub. While storing data in cache/pre-fetch area, the unwanted contents are identified and the prevalent noise is removed for efficient memory utilization. A voluntary cooperative environment is established to enable relevant content sharing which reduces the disadvantages of random content sharing or central node processing as well as reduces the complexity of keeping the content consistent and current. The performance of caching and pre-fetching is enhanced using Bloom Filters and the consistency is maintained using Least Recently Used (LRU) combined with First In First Out (FIFO) algorithms.

The rest of the paper is organized as follows: Section II discusses the related research works by reviewing the existing techniques, Section III describes the proposed methodology, Section IV presents the analysis of results and discusses the performance evaluation and Section V concludes the paper.

II. BACKGROUND RESEARCH

A. *Collaboration amongst nodes*

The collaboration amongst nodes can be achieved and the contents stored in any willingly participating node can be accessed by any other requesting node, if available. There is no dependency on central node/hub. Only if there is a local miss, contents are delivered by sever and then stored locally by the requesting node, which can be served in future locally. Consistency is maintained by the node itself[1]. Similar approach is used in cooperative caching in Vehicular networks, each device in moving vehicle decides and gets content from sever which can be shared among other vehicles to satisfy their content based need[2].

Contents are downloaded into base stations(nodes) close to users and these base stations request contents amongst themselves before hitting the sever enabling faster content delivery [3].

MuNCC consists of content request nodes, routers and content providers, when content is requested, each provider serves set of contents and routers cache some of the contents based on path coordination.The caching content in every node is also possible by global coordination. In this method, cache replacement could be tricky and implemented with coordinated cache eviction and coordinated caching strategy with refreshing cache summary intermittently[4]. In content based information centric caching for fog networking, nodes around the end user forms a fog network connection to perform caching, communication and even computation with each other directly to get content locally whenever available. Content based centrality is used for collaborative caching in information centric fogs[5].

Graph based cooperative caching in Fog RAN optimises clustering content caching using graphs based on local content popularity[6]. In Global Cluster Cooperative (GCC) caching, cache content is stored in each node and the contents from each cluster node is also stored in each node, which is not memory efficient. In case of local cache miss, content within cluster cache is searched, if not found (cluster cache miss) then the GCS (Global Cluster State) is searched. If found in other clusters (remote cluster hit) then it is served else fetched from sever. In Non cooperative caching future request pattern is predicted based on TTNR (Time To Next Reference) value. Cache consistency is maintained by each node using adoptive caching scheme[7].

CBRP is run in network layer and implemented using non overlapping cluster of nodes. One node is elected as CH (Cluster Head) and maintains local and cluster cache table. In this method, maintaining cluster consistency and keeping the tables update in all CH is a challenge[8]. In CCDP each autonomous nodes cooperate with each other to exchange data by multi hop communication and cache is maintained in each node/selected nodes. This method is specially used for image data items[9].

B. Sharing the cache contents using central node/mechanism

Predictive caching is done by clustering users as per their usage pattern and then pre-fetch contents based on pre-fetching schemes. Clustering and association using frequency support pruning improves web proxy servers' performance. Cache/pre-fetch contents are shared and managed using proxy server[10]. Cluster users and proxy servers are also grouped into clusters. Meta data is maintained in all proxy servers including neighbours, meta data is stored in the cluster to enable faster fetching (if content available in own/neighbours cluster) [11]. Five central modules are used in Cluster based COCA and pre-fetching [12]. Clustering, stack profile, information searching, cache management and pre-fetching are used for clustering and rendering content from within clusters. In CCSP, some nodes are elected as Service Cache provides (SCP), which receives summarisation of nodes and provide content when requested. If content is not available, then neighbouring SCPs are referred. Cache consistency check and replacement policies need to be done at each cached node. This approach may lead to more latency in case of server hit as all SCPs need to be referred before server hit. Load balancing while electing SCP may also be a process to be considered[13]. Maintaining consistency for cooperative caching is addressed using APCC and CBDC[14]. In these studies the benefits and improvements were studied and challenges of keeping content consistent and relevant are addressed.

C. Noise removal from web pages

Tag based noise removal is used to remove background images, advertisements, copyright information, unimportant links, search paneland audio, video plug ins[15]. The approach for removing Primary Noise – navigation bars, panels, frames, page header and footers, copy right and privacy notices, advertisements and other unimportant links like voice, video, multiple links etc. are discussed. Duplicate contents and noise content as per block importance are removed [16]. Global and local noises are removed[17]. Noise is removed by parsing the web page using Kdwd system and contents are tagged and many filters areused to remove unwanted contents[18)]. Noise is reduced using feature analysis, in which certain prominent features of noise / main content are used to separate noise from content[19]. There are plenty of research on removing noise for data mining [20], [21]. Researchers have identified techniques to remove noises using machine learning and neural networks [22], [23],[24]. There also exists user centric approach for noise removal based on dynamic user interests [25], [26].

III. PROPOSED METHODOLOGY

The proposed methodology contains the following

a. Effective caching and pre-fetching using EBF
b. Data sharing among cooperative users
c. Removal of noise
d. Fetching required contents of web page

A. Effective caching and pre-fetching using EBF

Caching and pre-fetching in individual nodes are done using Enhanced Bloom Filter (EBF). For caching and pre-fetching, Bloom filter array is created with 256 bits, out of which first 128 bits are used for domain name and remaining bits are used for storing path and file name of a URL. In the EBF array, all bits are initialised to 0 in the beginning and as user access web pages, MD5 and hash functions are applied on the URL and position of 1s in the outcome are identified and make changes in the EBF array. For each URL, the same MD5 and hash functions are applied and the resultant value is compared with the bit positions in EBF array, if the corresponding bit positions in both resultant value and EBF array are 1, it means that URL is available in cache area and it is rendered, if not available in cache, it is searched in pre-fetch area. If it is available in pre-fetch area, it is rendered. If it is not available in cache or pre-fetch area, then it is rendered from web server and it is stored in EBF by turning corresponding bit positions to 1. The content of the web page is processed, noises are removed and only required content is stored in cache. Also related contents (links) from web page are fetched into pre-fetch area based on user access pattern. Fetching content into pre-fetch area happens only when network is relatively free by monitoring the existing the bandwidth utilization.

The flow chart for enhanced bloom filter technique used for web caching is given in Fig. 1.

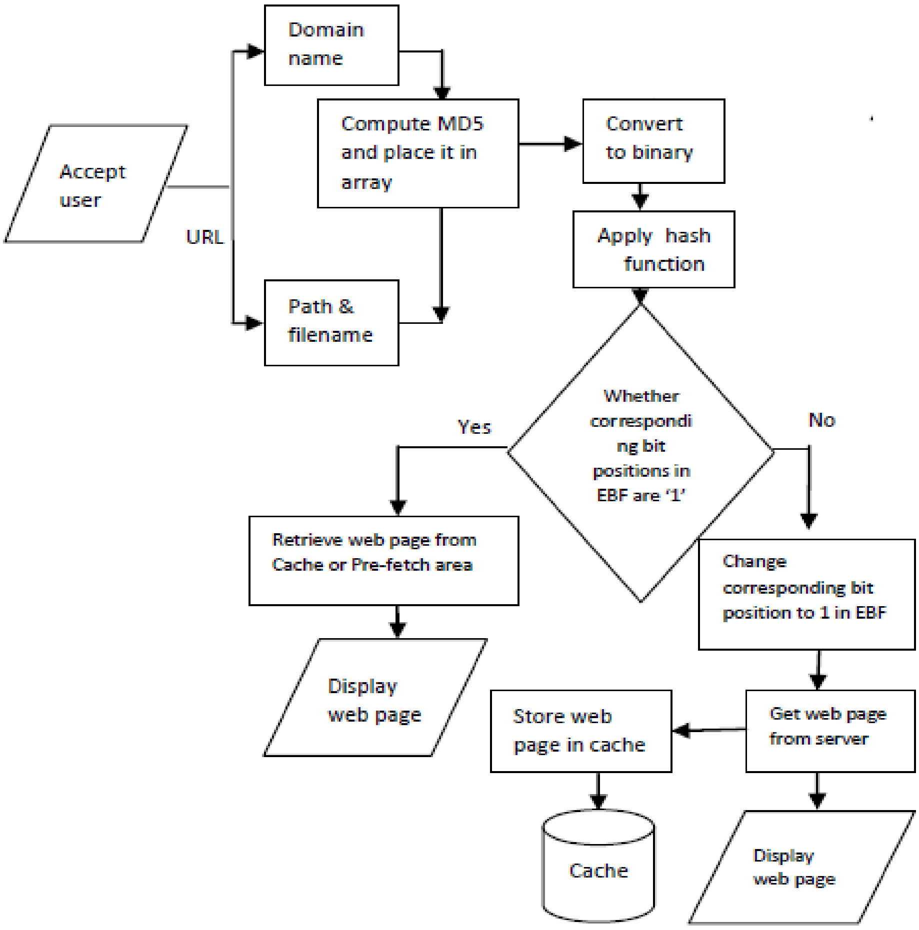

Fig. 1. Flowchart for Web Caching using Enhanced Bloom Filter Technique

When a new web page is fetched from the server and if cache/pre-fetch area is full, the LRU (Least Recently Used) combined with FIFO (First In First Out) is applied to replace cache with fresh contents. Also consistency with sever is checked and maintained while rendering contents from cache/pre-fetch area based on TTL (Time To Live) and If-modified-since-header values. If content is obsolete, then it is rendered from sever and replaced in cache/pre-fetch area. Cache replacement process is explained in Fig.2.

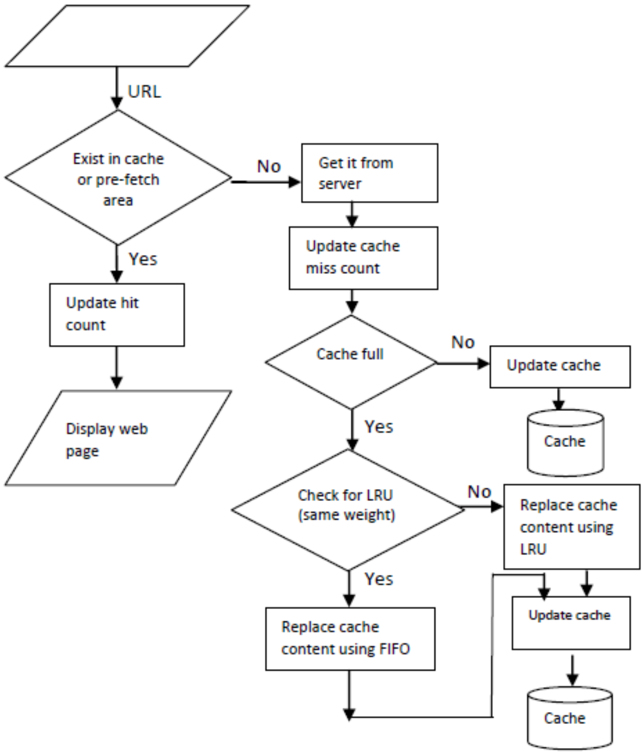

Fig.2.Flowchart for Cache Replacement using hybrid LRU and FIFO

B. Cache /Peftech data sharing among cooperative users

A voluntary hub has been established for cooperative users to share only required content among them using notifications without hitting the server. Only required contents are provided from local cache and pre-fetch area if available, otherwise hit the hub to find content availability with cooperative users and fetch it from cooperative users' cache/pre-fetch area instead of sending the request to the server.

Cooperative Users Cache Content Sharing using SignalR

SignalR is an ASP.NET library, which uses existing transport technologies based on the infrastructure. SignalR has the capability of real-time communication with wide range of clients including mobile devices. SignalR is a library for ASP.NET to add real-time web functionality to applications. Real-time web functionality has the ability to push content to the connected clients as it happens, in real-time. The creation of hub, creation of hub proxy, Cooperative Users (CUs) registration, CUs Group join, Start Connection, Send Notification, Send/Receive data and Update EBF array are the different process implemented using SignalR. The context diagram for establishing hub connection with cooperative users (CUs) and sharing data among them is given in Fig.3.

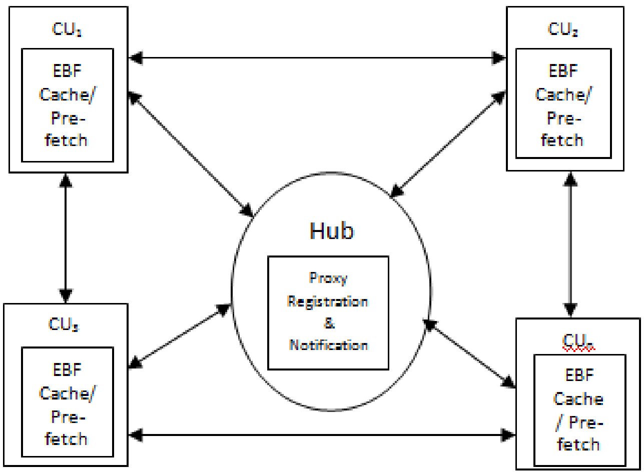

Fig. 3. Context Diagram for Data Sharing between Hub and Cooperative Users (CU)

In this case of cooperative cache content sharing, content replacement and consistency check are done by the CU which serves the content, hence it simplifies the complexity of consistency mechanism in cooperative/collaborative environment. Once the content is received from the CUs/server, then it is stored in local cache of the requesting CU, hence forth maintaining consistency and replacement will be done by CUs cache. Hub is used only to identify and route request to the right CU. This eliminates the complexity of maintaining cache consistency in multiple places.

C. Removal of noise

Memory utilization for caching and pre-fetching are improved by getting only required contents without any noise. Noise contents are to be identified with the help of HTML tags and these noise contents are to be removed from web pages. Different categories of noise pattern may be present in the web pages and removal of these contents saves the memory required for caching the contents. Care should be taken to remove noise while storing them in local cache/pre-fetch area. Noise pattern such as banners, search panels, advertisements, decorative noises like background images, copyright and privacy notices can be identified by using HTML tags. The HTML tags are used to detect the noise patterns in a web page. The Fig. 4 shows the phases involved in the process of noise removal.

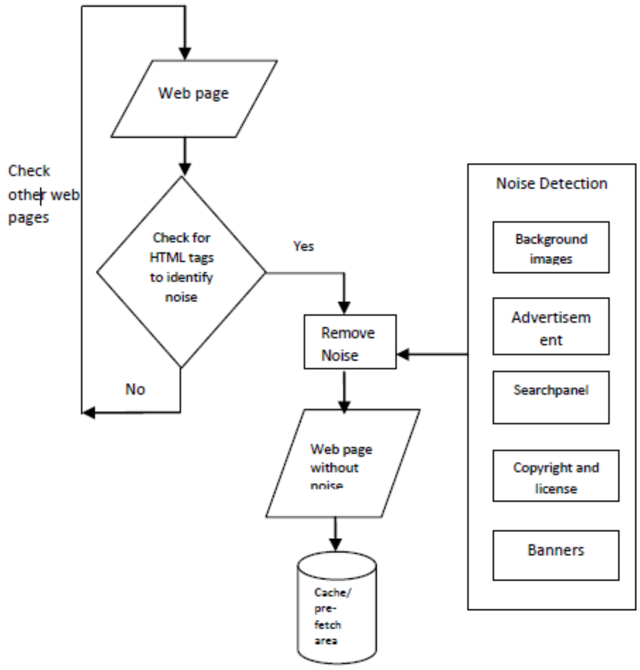

Fig. 4. Block Diagram for Noise Removal

D. Fetching required contents of web page using hybrid techniques

Removing noise is usually done while data mining, data analysis etc. However, removing local noise and reducing unwanted content can be applied while doing mobile caching/pre-fetching as well. It reduces cache memory usage and if it is used when cooperative users share their cache/pre-fetch content (without noise) with each other, it improves the performance as many times the reduced content is shared, compared to sharing cache data with noise. In this methodology, noise is removed every time content is fetched from original server and it is stored in local cache/pre-fetch area without noise. Hence while sharing amongst CUs only noiseless content is shared. The steps involved in fetching required contents of web pages using hybrid techniques are given below.

1. When user searches a URL, it is checked in cache, if found content is checked for consistency.
2. If the content is latest then it is rendered from cache and the count is updated.
3. If the content is stale, update miss count, fetch the web page from server, remove noise and store it into cache.
4. If the requested web page is not in cache but available in pre-fetch area, content is checked for consistency and if it is latest then it is rendered from pre-fetch area and updates its weight.

5. If the requested page is not in cache or pre-fetch area, the notifications in the hub are checked for availability.
6. If the notifications in the hub indicated that the requested document is available in cooperative users' cache then respective CU ID is taken from the hub and transfer CU ID to the cooperative user. The content is served from the respective cooperative user. The count is updated.
7. When content is not available in cache/pre-fetch area/CUs cache then miss count is incremented and content is fetched from web server and a copy is stored in cache using EBF.

Hybrid techniques are applied for cache replacement, maintaining consistency and noise removal. The requested web page is cached and links are pre-fetched using EBF. CU's hub is notified to indicate the availability of the recently fetched web page in the cache.The flow chart used for the implementation of the overall hybrid techniques is given in Fig.5.

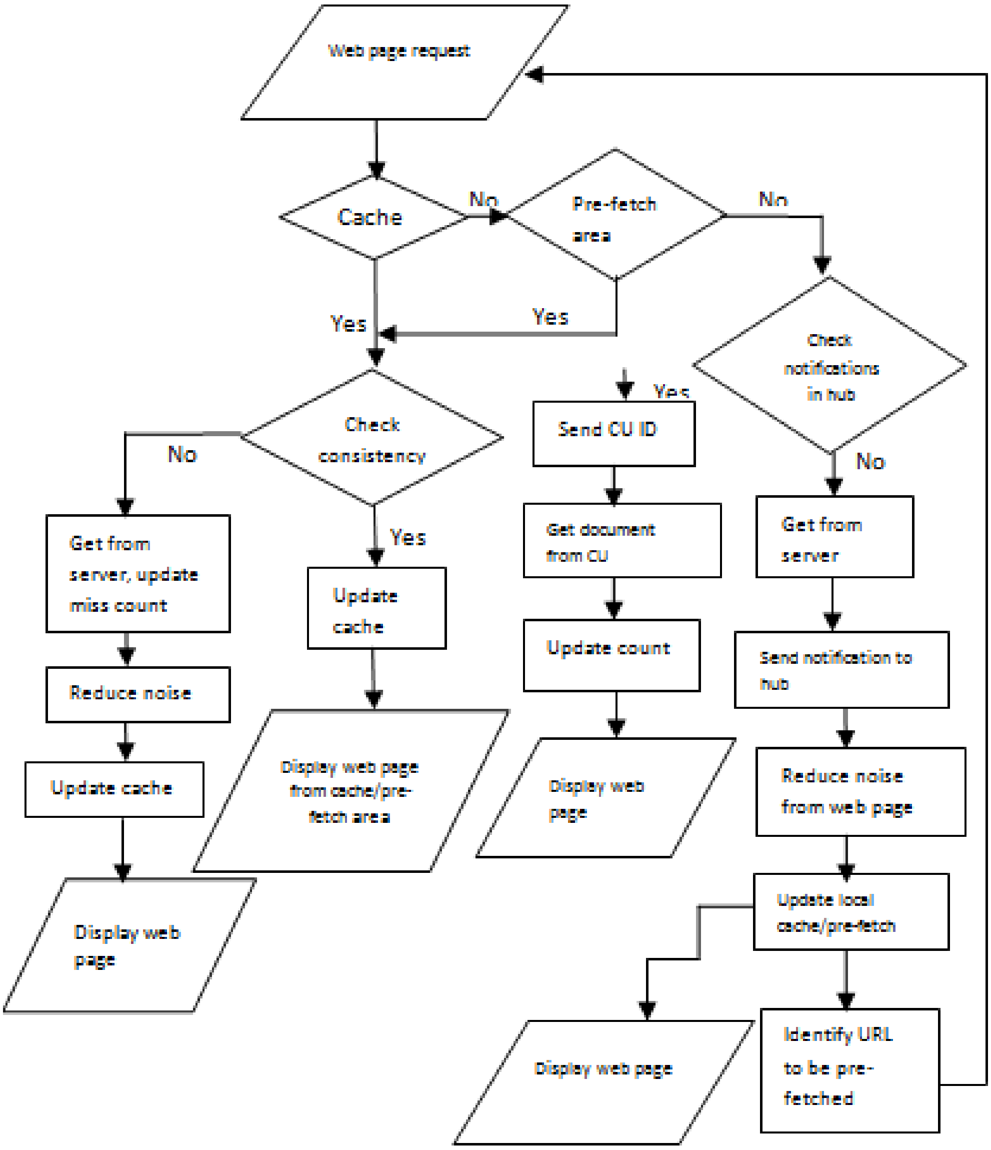

Fig. 5. Flowchart for fetching the Required Contents of Web Pages using Hybrid Techniques

IV. RESULTS AND DISCUSSION

The performance of the enhanced methods are analysed in terms of the parameters such as hit ratio, latency, bandwidth usage and memory utilization etc.

A. Hit Ratio analysis with content sharing amongst CUs

Cooperative content sharing is analyzed by repeating the URLs as well as with new URLs. Usually hit ratio improves when many cooperative users are repeatedly searching for similar content. If all CUs are searching only new content then hit ratio drops to individual cache and pre-fetch level and no benefits of using CUs cache. It is because not much content is shared. In Fig.6, it has shown that usually hit ratio improves over a period of time, as wider content will be available amongst CUs; hence chance of hitting within CU improves.

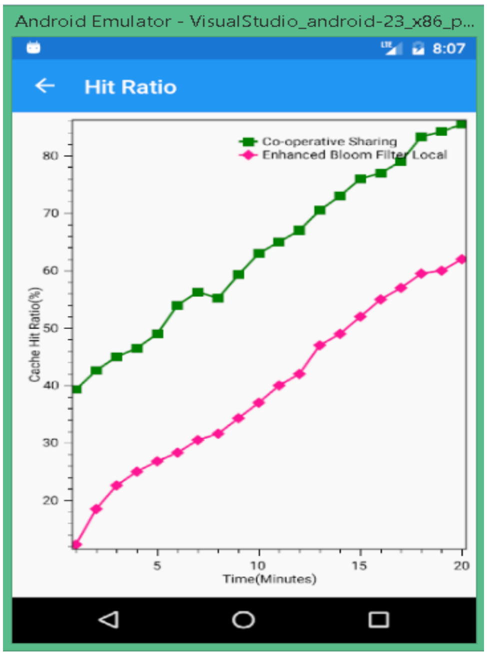

Fig.**6. Analysis of Cache Hit Ratio when Accessing Web Pages using EBF and CUs**

B. Latencyreduction in mobile caching using EBF

Latency when served from local cache is the lowest and rendering from server is much higher as in Fig. 6, however, on top of it when CUs performance is plotted, latency is much lower than serving the page from server and it is only slightly higher than local cache, if page is available with CUs and shared amongst them. Significant reduction in latency compared to server hit is observed, when page is served from any CUs. If more CUs and more contents are available amongst CUs then also latency will be slightly higher than the latency when serving the document from local cache and latency will be more when serving the content from the server. The following Fig. 7 shows the analysis of latency reduction between local cache and CUs.

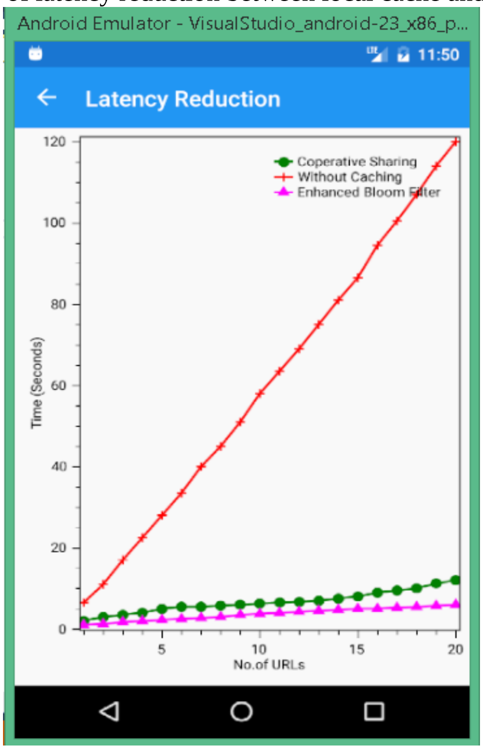

Fig.7. Analysis of Latency Reduction between Local Cache and CUs

C. Memory Reduction after Noise Removal

The amount of memory reduction by removing the noises from web pages depends on the amount of noise present in the web pages. When noise is removed, in some cases 78% content was reduced or in other words original content got reduced to 22%. In some cases, only 21% noise is reduced and content got reduced only by 79%. Table 1 shows memory reduction before and after noise removal.

Table 1 Comparison of Memory Reduction when Removing Noise from Web Pages

Web Page	Before noise removal (KB)	After noise removal (KB)	Memory Reduction (%)
Msdn	175	81	46.3
Nasscom	140	111	79.3
Sify	963	794	82.5
Wordpress	317	71	22.4
Bbc	204	147	72.1

D. Latency improvements using hybrid method

In hybrid method, noise is removed while storing it in local cache/pre-fetch area as well as in CUs cache and pre-fetch area. Only required content without noise is rendered. This improves hit ratio as more content can be stored in cache and reduces latency as content is rendered faster as well as reduces bandwidth while content is shared amongst CUs. The Fig.8 shows the overall improvements in latency when noise is reduced.

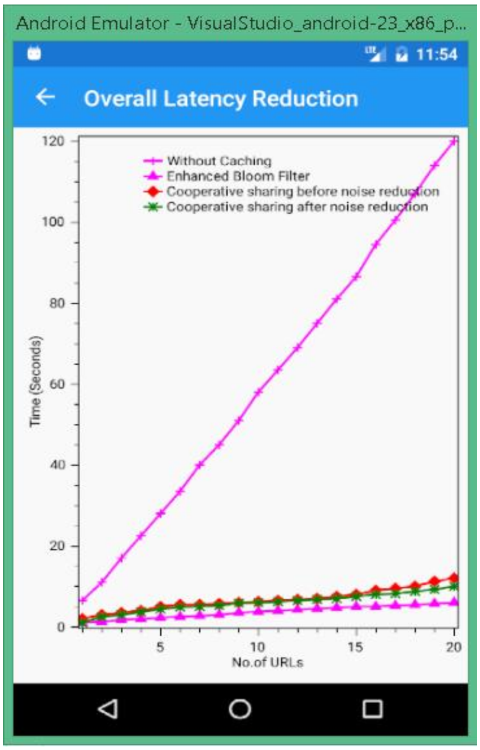

Fig. 8. Analysis of Latency Reduction when Accessing Documents without Noise

Retrieval Number: C6310098319/2019©BEIESP
DOI:10.35940/ijrte.C6310.098319
Journal Website: www.ijrte.org

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

E. Bandwidth utilisation

The Fig.9 shows the analysis of bandwidth utilization of five cooperative users before and after noise reduction. Bandwidth utilization is better if more noise content is reduced. Depends on the noise content in web page, bandwidth is reduced varying from negligible amount to up to 40%.

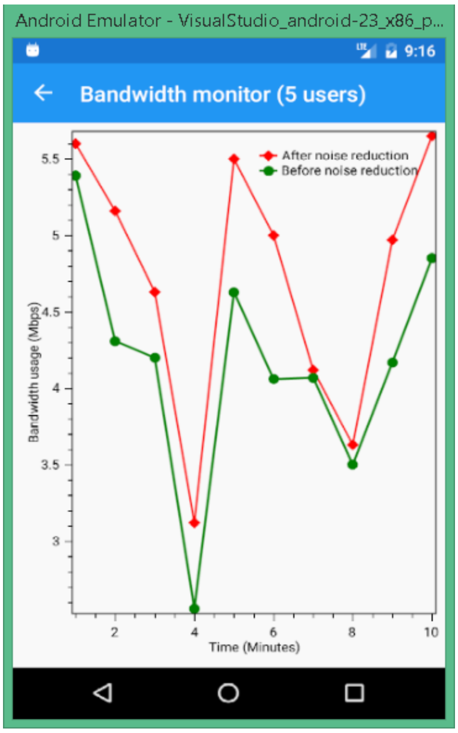

Fig.9. Analysis of Bandwidth Utilization of Five Cooperative Users Before and After Noise Reduction

V. CONCLUSION

The novel hybrid technique enhanced the overall performance in hit ratio, latency reduction, cache memory utilization and better bandwidth usage. It is observed that hit ratio is above 50% for a single user and improved further up to 80 to 90% when contents are shared between cooperative users with similar usage pattern. Latency is another key parameter, which drops from 120 seconds tofew seconds in case of content served from cache using EBF and around 10 to 20 seconds if served from CUs content. There is a huge reduction in latency if unwanted contents are removed from web page and served from local cache with reduced noise. Noise reduction by itself may not be much significant for latency though it reduces few seconds delay, but it is mostly based on the content. However noise reduction has got many other improvements in terms of memory utilization and usability. In hybrid method, latency is considerably reduced by integrating caching and pre-fetching using EBF and CUs content sharing without noise. Latency drops from as high as 120 seconds to 10 seconds. Depends on the noise content in web page, bandwidth requirement is reduced varying from negligible amount to up to 40%. Pre-fetching is done when network is relatively free, however it may utilize bandwidth to provide better hit ratio and latency reduction. However usage pattern of users plays a major role in increasing CUs hit rather than number of CUs.

REFERENCES

1. Sibren Isaccman and Margaret Matonosi, “Potential for collaborative caching and pre-fetching in largely disconnected villages”, ACM,2008.
2. Saeid Akhavan Bitaghsirabd Ahmed Khonsari, “Cooperative caching for content dissemination in vehicular networks”, International Journal of communication systems, 2018,pp. 1-22.
3. Abdullah Khreishah and Jacob Chakareski, “Collaborative caching for multicell-coordinated systems”, IEEE conference on computer communications, 2015.
4. Tavis Mick , Reza Tourani , Satyajayant Misna, “MuNCC: Multi-hop neighborhood collaborative caching in information centric networks”, Research Gate, 2016, pp. 93-101.
5. Junaid Ahmed Khan, Cedric Westphal and Yacine Ghamri-Doudane, “A content based centrality metric for collaborative caching in information-centric fogs”, 1705.01343v1 [cs.NI], 2017.
6. Xiaoting Cui, Yanxiang Jiang , “ Graph based cooperative caching in Fog-RAN”, arXiv:1806.0909v1 [cs.NI], 2018.
7. R.Sarada, Shirin Banu Koduri, Dr. M. Seetha, “A cluster based cooperative caching mechanism in MANET”, International Journal of Computer Science and Technology, vol. 3, issue 2, 2012, pp. 1057-1062.
8. Abdulaziz Zam, N. Movahedinia, “Performance improvement of cache management in cluster based MANET”, International Journal of Computer Network and Information Security, vol. 10, 2013, pp. 24-29.
9. A. Bathsheba Parimala , B. Jefferson, K. Appasamy, “Data caching and pre-fetching using collaborative caching and pre-fetching algorithms in wireless ADHOC networks”, International Journal of Engineering and Techniques, vol. 3, issue 6, 2017, pp. 35-41.
10. Nanhay Singh, Arvind Panwar and Ram Shringar Raw, “Enhancing the performance of web proxy server through cluster based pre-fetching techniques”, IEEE, 2013, pp. 1158 –1165.
11. Niklesh Rathore, Ms. Swati Tiwari, “New framework for web proxy server through cluster based pre-fetching”, International Journal of Software & Hardware Research in Engineering, vol. 2, issue 10, 2014, pp. 1-4.
12. Shashidhara DN, Chandrappa DN, “A literature survey on caching and pre-fetching techniques in MANET”, Journal of Information Technology and Software Engineering, vol. 6, issue 5, 2016.
13. Ali Larbi, Louiza Bouallouche-Medjkoune and Djmil Aissani, “Improving cache effectiveness based on cooperative cache management in MANETs”, Volume 98, issue 3, pp 2497–2519, 2018.
14. Kuppusamy P and B. Kalaavathi, “Cluster based data consistency for cooperative caching over partitionable mobile adhoc network”, American Journal of Applied Sciences, Science Publications, vol. 9, issue 8, 2012, pp. 1307 – 1315.
15. Mayajohn, Jayasudha J.S., “Tag based noise removal from web pages”, Journal of theoretical and applied information technology”, vol.95, issue.2, 2017, pp. 6336-6346.
16. P. Sivakumar, “Efficient web content mining using noise removal from web pages”, Springer, Wireless Personal Communications, vol. 84, issue 1, 2015.
17. Surabhi Linngwal, “Noise reduction and content retrieval from web pages”, International journal of compute applications”, vol. 73, issue. 4, 2013, pp. 24-30.
18. Hitesh Kumar Azad, Rahul Raj, Rahul Kumar, Harshit Ranjan, Kumar Abhishekand M.P. Sing, “ Removal of noisy information in web pages”, ICTCS, ACM, 2014.
19. Kun Jung and Yuexiang Yang, “Noise reduction of web pages via feature analysis”, International conference on information science and control engineering, IEEE, 2015, pp. 345 – 348.
20. S.S. Bhamare and B.V. Pawar, “An efficient method of web page noise cleaning for effective web mining”, International journal of computer applications, vol. 146, issue, 3, 2016, pp. 18-22.
21. Yogita K Patel, Nrendasinh Limbad, “Noise removal from web pages for web content mining”, IJAIIE, vol. 2, issue 3, 2016, pp. 2293 – 2299.
22. Neetu Nawal and Dr. Sanjay Kuma Sharma, “Machine learning for web page adaptation”, International journal of computer science and information technologies, vol. 7, issue 1, 2016, pp. 60- 65.
23. Hamza Yunis, “Content extraction from web pages using machine learning”, Master’s thesis, 2016.

24. Thanda Htwe and Khin Haymar Saw Hla, “Noise removing from web pages using neural network”, IEEE, vol. 1, 2010, pp. 281 – 285.
25. Julius Onyancha and Valentina Plekhanova, "A user centric approach towards learning noise in web data", 12th International conference on intelligent systems and knowledge engineering, IEEE, 2017.
26. Julius Onyancha and Valentina Plekhanova, “Noise reduction in web data: A leaning approach based on dynamic user interests”, International Journal of Computer and Information Engineering”, vol. 12, issue.1, 2018, pp 7-14.

AUTHORS PROFILE

P. Amudha Bhomini received the B.Sc. (Computer Science) degree from Madurai Kamaraj University, Madurai in 1990, M.C.A. degree from University of Madras, Madras in 1993 and M.Phil (Computer Science) degree from Mother Teresa Women’s University, Kodaikanal in 1999. She is working as an Associate Professor in Computer Applications Department, Nesamony Memorial Christian College, Marthandam since 1999. She has presented and published 6 papers in National and International Conferences and Journals. Her current research interest focuses on Computer Networks, Wireless Communications and Network Security.

Jayasudha J. S. received her B. E. degree from Madurai Kamaraj University in 1994, M. E. degree from National Institute of Technology, Trichy in 2002 and Ph.D from Kerala University in 2008. She has been working in Sree Chitra Thirunal College of Engineering in the department of Computer Science & Engineering since November 1996. She has 22 years of teaching experience. Now she is working as Professor and Dean (Academic). She has presented and published many papers in National and International Conferences and Journals. Her current research interest focuses on Computer Networks, Image Processing and Cloud Computing.