A SURVEY ON WEB PRE-FETCHING AND WEB CACHING TECHNIQUES IN A MOBILE ENVIRONMENT

Greeshma G. Vijayan[1] and Jayasudha J. S.[2]

[1]Department of Computer Science & Engineering, SCT College of Engineering, Kerala
greeshmagvijayan@gmail.com
[2]Department of Computer Science & Engineering, SCT College of Engineering, Kerala
jayasudhajs@gmail.com

ABSTRACT

As the Internet continues to grow in size and popularity, web traffic and network bottlenecks are major issues in the network world. The continued increase in demand for objects on the Internet causes severe overloading in many sites and network links. Many users have no patience in waiting more than few seconds for downloading a web page. Web traffic reduction techniques are necessary for accessing the web sites efficiently with the facility of existing network. Web pre-fetching techniques and web caching reduces the web latency that we face on the internet today. This paper describes about the various prefetching and caching techniques, how they predict the web object to be pre-fetched and what are the issues challenges involved when these techniques are applied to a mobile environment.

KEYWORDS

Prefetching, cache, latency

1.INTRODUCTION

Web prefetching is fetching web pages in advance by proxy server/client before a request is send by a client/proxy server. The major advantage of using web prefetching is reduced latency. When a client makes a request for web object, rather than sending request to the web server, it may be fetched from a pre-fetch area. The main factor for selecting a web prefetching algorithm is that its ability to predict the web object to be pre-fetched in order to reduce latency. Web prefetching exploits the spatial locality of web pages, ie. pages that are linked with current page will be accessed with higher probability than other pages. Web prefetching can be applied in a web environment as between clients and web server, between proxy servers and web server and between clients and proxy server[3]. If it is applied between clients and web server, it is helpful in reducing user perceived latency, but the problem is it will increases network traffic. If it is applied between proxy server and web server, can reduce the bandwidth usage by prefetching only a specific number of hyper links. If it is applied between clients and proxy server, the proxy starts feeds pre-fetched web objects from its cache to the clients so there won't be extra internet traffic. This paper describes about the various prefetching techniques, how they predict the web object to be pre-fetched and what are the issues involved in these techniques. Web pre-fetching techniques are categorised into three- probability based, clustering based and using weight-functions[21]. In

Natarajan Meghanathan, et al. (Eds): ITCS, SIP, JSE-2012, CS & IT 04, pp. 119–136, 2012.
© CS & IT-CSCP 2012 DOI : 10.5121/csit.2012.2111

the probability based pre-fetching, probabilities are calculated using the history access data. This method assumes that the request sequence follows a pattern and the probabilities are trying to follow this pattern. Clustering based pre-fetching methods make decisions using the information about the clusters containing pages that have been fetched previously, assumes that pages that are close to those previously fetched pages are more likely to be requested in the near future.

The rest of the paper is organized as follows, Section 2 discusses various web prefetching techniques, Section 3 deals with web prefetching and caching in a mobile environment ,Section 4 discusses various web caching techniques Section 5 provides some conclusion remarks.

2. Web Pre-fetching Techniques

2.1. Interactive Prefetching Scheme

In this scheme all the hyper links and inline images in linked pages are fetched. Since it retrieves all the hyper links, hit rate of 80% is possible. The disadvantage is it increases the load on to the host, and requires a lot of memory to store the pre-fetched web pages.
Interactive Pre-fetching Algorithm(from [1])

```
procedure Prefetching(base_page : URL)

    var   A, I : URL;
     rpage_set, image_set : set of URL;
begin

          {gather A-tag by parsing of HTML }

rpage_set := GatherReferences(base_page);

          foreach A in rpage_set do

          begin

 {get HTML part of referenced page via HTTP }

           Get(A);
          {gather IMG-tag by parsing of HTML }

          image_set := GatherImages(A);

          foreach I in image_set do

                    begin

{get in-line images of referenced page via HTTP }

                              Get(I);
                    end;
          end;
end;
```

2.2 Link Pre-fetching

This mechanism, utilizes browser idle time to download documents that the user might visit in the future. A web page provides a set of pre-fetching hints to the browser and after the browser finishes loading the page, it starts pre-fetching specified documents and stores them in its cache. When the user visits one of the pre-fetched documents, it can be served up quickly out of the browser's cache. Fisher et. al proposed a server driven approach for ink prefetching[15].In this approach ubrowser follows special directives from the web server or proxy server that instructs it to pre-fetch specific documents. This mechanism allows servers to control the contents to be pre-fetched by the browser. The browser looks for either HTML <link> tag or an HTTP Link: header tag to pre-fetch the subsequent links. The Link: header can also be specified within the HTML document itself by using a HTML <meta> tag[16].When the browser is idle, it observes these hints and queues up each unique request to be pre-fetched.

2.3. Top 10 Approach

Evangelos P. Markatos et al. proposes a top 10 approach to prefetching on the web , in which the server calculates the list of most popular documents[2].This approach is easy to implement in a client server architecture. It consider frequency of access for predicting the web object,not the client characteristics on the web.

2.4. Domain Top Approach

Seung Won Shin et al. proposes a domain top approach for web prefetching, which combines the proxy's active knowledge of most popular domains and documents[3].In this approach proxy is responsible for calculating the most popular domains and most popular documents in those domains, then prepares a rank list for prefetching.

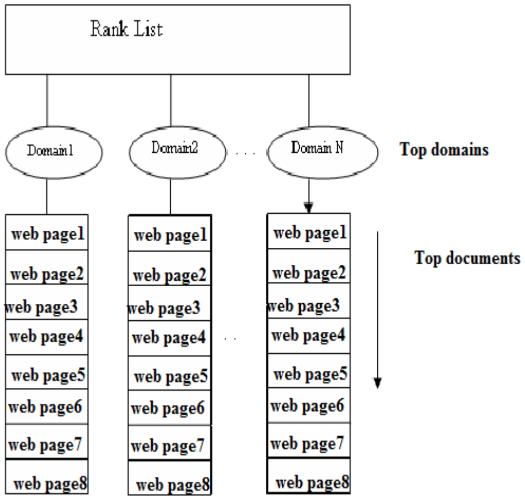

Figure 1. Structure of rank list for Domain Top Approach

2.5. A Keyword based semantic prefetching approach in internet news services

This proposes a key word based sematic prefetching ,in which prediction of future requests are based on semantic preferences of past retrieved web documents.This technique is applied to internet news services , it finds out sematic preferences by analyzing keywords in URL anchor text of previously accessed documents in different news categories[4][17].

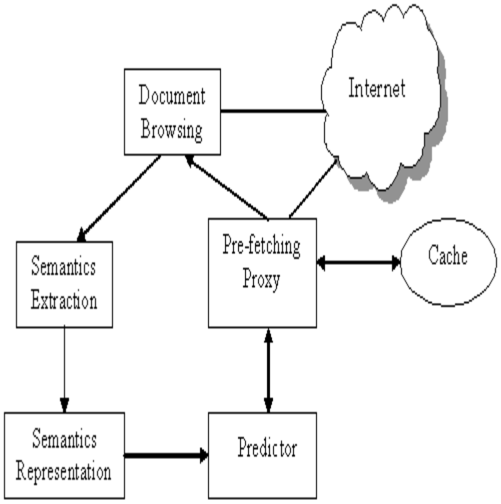

Figure 2. Architecture of semantic prefetching scheme

Figure 2 shows architecture of client side semantic prefetching scheme.It assumes a proxy is running behind the web browser keeps track of clients characteristics and find out semantic relation between the documents[4].

The semantic representation is represented in an open self learning capable model which collects the knowledge about the client preferences. Client future request predictions are based on this knowledge. The prefetched documents are stored in internal cache. When client makes a request the web document is fetched from cache if it is available otherwise it fetches the documents from internet. This technique uses a neural network based semantics model ,that has the capability of self learning. So this technique capable of prefetching URLs have never been accessed. But it has to handle the related issues with key words: synonymy and polysemy.

2.6. Dynamic web prefetching

In dynamic web pre-fetching technique[5], each user can keep a list of sites to access immediately called user's preference list. The preference list is stored in proxy server's database. Intelligent agents are used for parsing the web page, monitoring the bandwidth usage and maintaining hash table, preference list and cache consistency. It controls the web traffic by reducing pre-fetching at heavy traffic and increasing pre-fetching at light traffic. Thus it reduces the idle time of the existing network and makes the traffic almost constant. A hash table is maintained for storing the list of accessed URLs and its weight information. Depending upon the bandwidth usage and weights in the hash table, the prediction engine decides the number of URLs to be pre-fetched and gives the list to pre-fetch engine for pre-fetching the predicted web pages. After pre-fetching, the proxy server keeps the pre-fetched web pages in a separate area called pre-fetch area.

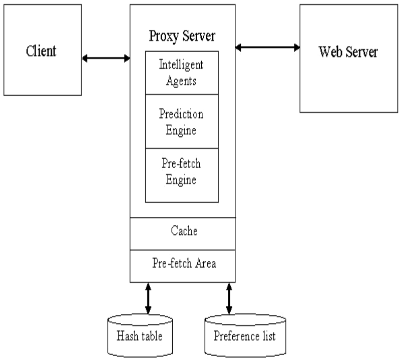

Figure 3. Dynamic Prefething model

The major advantage of using this technique is that, number of links to be prefetched is based on the current band width usage. This technique increases overhead to proxy/client.

2.7. WebCompanion

This approach employs a highly selective prefetching strategy based on estimated round-trip times for web resources.

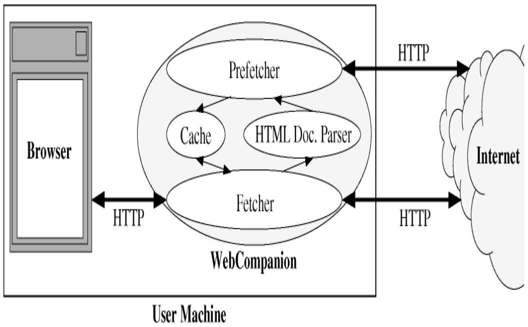

Figure 4.Web Companion acts as an http proxy between web browser and the internet

Figure 4 shows how WebCompanion interacts with a browser and the Internet. Every request for a Web resource from the browser is routed through a WebCompanion component called Fetcher . WebCompanion inspects its document cache and returns the requested resource if it is stored in the cache[6]. If not, WebCompanion accesses the addressed origin server, fetches the resource, stores it in its cache, and forwards it to the browser. If the resource is an HTML document, an HTML document parser in WebCompanion compiles a list of all embedded hyperlinks. When the process of loading this HTML document and all embedded resources is completed, WebCompanion activates its prefetching engine and prefetches a subset of the documents addressed by the embedded hyperlinks. All prefetched documents are transferred to WebCompanion's cache. The advantage of using this technique is that it reduces network overhead. But it is limited to prefetching of static web pages.

2.8. Markov model for predicting web access

Markov model and hidden markov model are used for predicting the web pages to be prefetched based on the web access probabaility. Traditional markov model takes a sequence of web pages accessed by the user as an input and predict the next page to be accessed by the user. Low order markov models can't accurately predict the subsequent requests of user. If higher order markov models are used the memory required increased rapidly with order.To solve this Xing Dongshan et. al proposed hybrid order tree like Markov model, which can predict web access precisely, providing high coverage and good scalability[12].

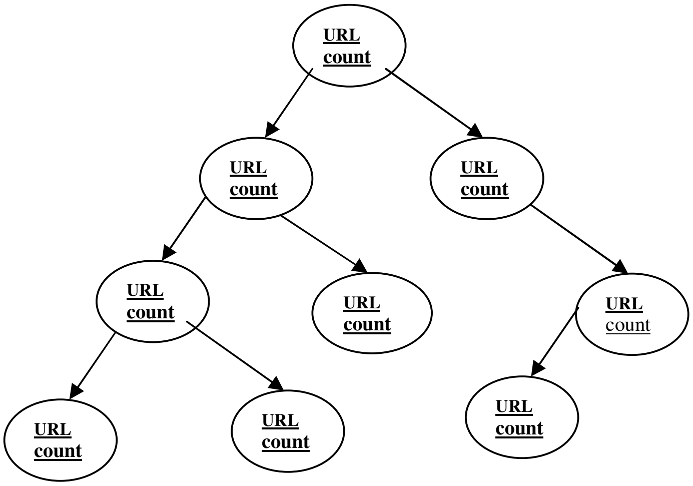

Figure 5. Structure of a tree like markov model[12]

Figure 5 shows a K-order tree like markov model(TMM) constructed from past web access sequence. Each node in the TMM is a visited page and and each branch is the path to the node.The number of times the user has visited the node through the same route is recorded in the variable count in each node. If the user's current access sequence is $u_1u_2...u_{m,}$, the K-order TMM predicts user's access patterns by extracting k suffixes, $u_{m-k+1}u_{m-k}...u_m$.

Xin Jin et. al[13] proposed a novel web prefetching approach based on Hidden Markov Model. The HMM is used for analyzing user's browsing history and make semantic based web prefetching decisions. These models are based on historial data, so these are unable to predict a new web request.

2.9. Data Mining Algorithms for Web Pre-fetching

Zhang et. al[14] proposed data mining algorithms for web prefetching, which uses a simple WWW data model to represent data to mine the association rules.The mined association rules are stored in a knowledge base which is used to predict the user's actions. It contains an intelligent agent ,responsible for mining user's interest and prefetching web pages.

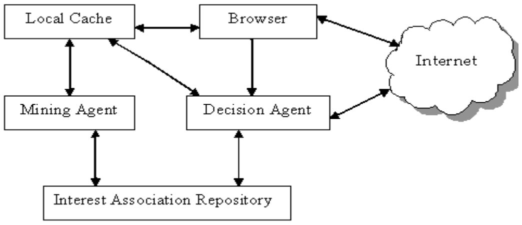

Figure 6. Data mining agent for prefetching system[14]

Figure 5 shows agent based web pre-fetching system, in this browser side is attached with Interest Association Repository, mining agent and decision agent. The major task of mining agent is to update the interest association rules periodically. The responsibility of decision agent is to detect and predict the user activities, also it does tasks such as getting the current web page from browser, analyzing the current web page, interact with interest association repository and prefetching web pages and storing into the local cache. This technique is based on the analysis of great amount of historical data so unable to prefetch a new web object.

Yang et. al proposed an application of web log mining to obtain web-document access patterns to extend the caching and prefetching policies[30]. Hung et al. proposed an access sequence miner for mining web logs to improve hit ratios of prefetching and caching[31]. Rule assisted prefetching identifies set of association rules from the web server's access log[32]. Its access miner analyzes and discovers acess patterns in the form of association rules. Nanopoulos et al. developed a new algorithm called WMo, which is based on data mining and is proven to be a generalization of existing ones[33]. Songwatta proposed Mining Web Logs for Prediction in Pre-fetching and Caching in which web mining and machine learning techniques are used for prediction in pre-fetching and caching[34].

2.10. Prediction by Partial Match

Palpanas et. al[18] proposed prediction by partial match model ,in which prefetching decisions are taken by reviewing the URLs that clients on a particular server have accessed over some period[18].This algorithm has three parameters: order, depth and threshold. The algorithm uses order, which is the length of the history substring to find a match. Depth represents the number of accesses into the future the algorithm attempts to predict. Threshold is the minimum probability ,in which a pre-fetch candidate should have. The standard prediction by partial match model uses multiple markov models to store the historical URL paths[19].In this client sessions are represented by each markov model, it uses any URL for a root node and records any subsequent URL in the tree rooted by the first URL. Every node represents an accessed URL. It consists of a counter ,which records the number of times the URL occurs in the path from the root node. The advantage of standard PPM model is that it is not very complex.

2.11. Model based Predictive Pre-fetching

Yang et. al proposed a model based predictive pre-fetching, in which an integrated web-caching and web-pre-fetching model is used[22]. The prediction model used in this is based on the statistical correlation between web objects. The prediction model is time based, prediction window represents some specific time period than number. The algorithm constructs a logical graph called correlation graph,which shows the correlation between web objects and prefetch prefetch web objects that are highly correlated to a currently requested object. They developed an integrated caching and prefetching algorithm, Pre-GDF. This algorithm is based on the algorithms GD-Size[23] and its enhancement GDSF[24]. The key components in the algorithm are replacement manager, pre-fetching agent, prediction queue and cache.

2.12. Adaptive pre-fetching Scheme

Adaptive pre-fetch scheme are developed to adapt user's browsing history and habits [25]. Jiang and et al. proposed an adaptive pre-fetch scheme , in which the number of files to be pre-fetched depends on user access history and network conditions[26]. This scheme consists of two modules: prediction module and threshold module. The prediction module updates the history and compute the access probability of each file. Files whose access probabilities greater than or equal to the pre-fetch threshold are only pre-fetched. Chen and et. al[27] proposed an adaptive pre-fetch scheme, in which dynamically adjust the pre-fetch aggressiveness in web servers and uses a threshold to adjust the aggressiveness of pre-fetching. Fagni and et. al[28] proposed uses an approach for boosting the performance of search engine by exploiting the spatial and temporal locality present in the stream of processed queries.

2.13 Web Pre-fetching using Display Based Prediction

This sscheme constructs link graph, which contains both nodes and links by gathering usage information of a client inorder to analyze the reference patterns and predict the next request based on overall displayed documents in the web browser. The node indicates a HTML document with unique URL and the link represents hyperlink or embed link on it. The access counter of a node is increased by one whenever a user fetches or follows it[35].

2.14 Web Prediction based on Evolutionary Algorithm

Bonino et. al proposed a an evolutionary algorithm for evolving a population of prediction machine to provide a prediction model for user’s next request at each session[36]. The general architecture of prediction system is shown in Figure . composed of two modules: an evolutionary algorithm and request predictor.

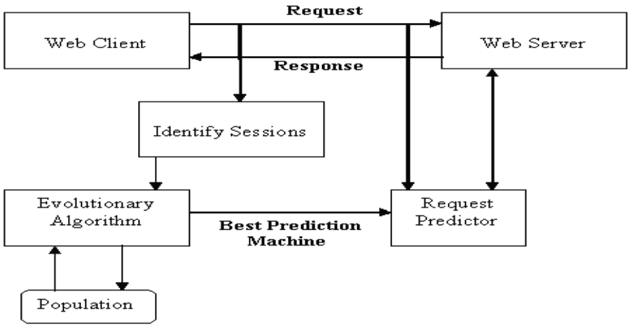

Figure 7. General Architecture of prediction Model using Evolutionary Algorithm

This scheme exploits the user navigational path behaviour for predicting future requests in real time.

2.15. Non-interfering Prefetching

A non-interfering system avoids interference by effectively utilizing spare resources on the servers and network[29]. In this system prediction and resource management are two important tasks. Predictor prepares prioritized lists of high-valued documents for prefetching and resource manager limits the number of documents to pre-fetch and schedules the pre-fetch request to avoid interference.

3. WEB PREFETCHING AND CACHING IN A MOBILE ENVIRONMENT

Recently, mobile devices such as personal digital assistants, smart phones, and hand held media players are powerful enough to be employed as service providers and participate within the ubiquitous computing environment[7]. So web prefetching can be applied to a mobile environment to improve overall performance.

Zhimei Jiang et al. proposes web prefetching in mobile environment, in which it uses an adaptive prefetch scheme. In this approach prefetch decision is able to adapt different network conditions. This proposes two modules: prediction module and prefetch module, computes the access probabilities and prefetch thresholds respectively. The access probabilities indicate how likely files will be requested by the user, and the prefetch thresholds determine whether the performance may be improved by prefetching certain files. As a user changes network in a mobile environment, it is the pre-fetch threshold, which is computed based on system conditions as well as costs of bandwidth and time, that adjusts the number of pre-fetched files accordingly[8]. When using access probabilities from server, successful prediction rate is generally lower. This approach is based on user's access history, so unable to predict a new web object.

Sarina Sulaiman et al. proposes a new hybrid technique based on combination of ANN and Particle Swarm Optimization (PSO) for classification Web object either to cache or not and generate rules from log data by using Rough Set technique on proxy server (Rough Neuro-PSO). This approach is needed because mobile context has limited resources like speed and memory. This method is by using XML file for prefetching which is saved into mobile memory. Prefetching that used xml file is much faster to be searched or accessed. In Web caching side, enhance the speed by using Rough Neuro-PSO in order to choose the log[9]. his approach uses PSO to adjust weight on ANN in order to find an optimal weight. This ANN is designed to train Web caching until it is able to find which object that very popular to be accessed. Then, Rough Set is used to get the Web caching rules.The proposed framework is designed to rise up the access of social network using mobile devices.

Siti Mariyam Shamsuddin et al.This paper proposes a new technique known as an Intelligent Mobile Web Pre-fetching (IMWeP) that creates pre-fetching in a mobile environment using the Extensible Markup Language (XML) technology. This technique proves to be faster in searching or accessing process in mobile applications. A case study on Facebook Mobile investigates the loading process by creating priority ranking on XML files using the proposed IMWeP technique[10]. This approach is mainly for social network site facebook. It calculates the cumulative number of hits for each facebook mobile feature (inbox, home, profile, and friends) when the user vists in future.

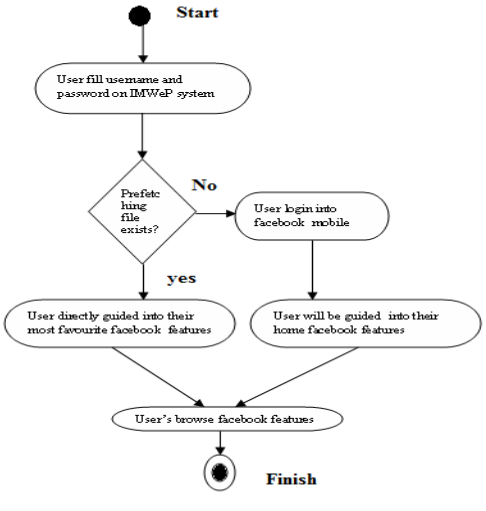
Start
User fill username and password on IMWeP system
Prefetching file exists?
No
User login into facebook mobile
yes
User directly guided into their most favourite facebook features
User will be guided into their home facebook features
User's browse facebook features
Finish

Figure 8. User's activity in the proposed system

Major issues for web prefetching in mobile environment is the limited resources, like memory screen size etc. Another issue is related to bandwidth- efficient use of bandwidth is required. For better performance bandwidth usage must be limited. In the case of a client side prefetcher security is one major issue.Some security mechanism is required to be implemented to protect the cached or prefetched data. The security mechanism thus implemented shall not increase the latency too much. Most of the related works is based on access history. Improvement in this can be done using some learning approach. Studies shows that integrating caching and prefetching can better reduce the latency. In a mobile environment, if we are integrating caching and prefetching redundancy is another issue.

4. WEB CACHING TECHNIQUES

Cache collaboration is important to improve caching performance. When one of the clients sharing the proxy generates a request, the proxy searches its local storage for the requested object. If the object is available locally it is sent to the client otherwise the request is passed on to the remote cooperative proxy servers or cache servers. If there is no remote hit, the client requests are forwarded to the origin server. Web caching techniques can be characterized as back-end or front-end [36]. Back-end approaches are deployed inside the site infrastructure at the original server side. They can reduce content generation delays but they do not address network related delays and bandwidth consumption issues. Front-end approaches concern caching outside the site infrastructure at a proxy side or a cache that resides at the edge of a Content Delivery Network (CDN). Front-end web caching techniques are serving at client side and they have difficulty in handling dynamic web pages due to their strong dependency on the back-end site infrastructure.

In proxy caching, proxy server intercepts HTTP requests from clients and if it finds the requested object in its cache, it returns the object to the client. If the object is not found in proxy's cache, it forwards the request to the original server, gets the object, deposits it in cache and returns the object to the user. One disadvantage of this design is that the cache represents single point failure in the network [37]. This drawback can be avoided by sharing the caches of proxy servers [38]. In reverse proxy caching, caches are deployed near the servers. This mechanism is useful for servers that expect a high number of requests and want to assure a high level of quality of service [39]. In application-level caching, caches are explicitly managed by application. It is designed with an API which allows an application writer to explicitly manage the cache contents [40]. The information about proxy caches that is used in filtering the request is called hint information. Hint allows clients to make decisions based on local state [41]. A shared cache solution for the home Internet gateway is outlined in [42]. For avoiding proxy overhead, site-based dispatching (SBD) technique forwards only hot-site requests to the proxy. SBD forwards request to less popular sites directly to the remote web sites [43]. Summary of some web caching architectures [44] are given in table 4.1.

Table 4.1 Summary of web caching architectures([44])

Architecture	Description	Advantage	Disadvantage
Proxy (Known as forward proxy caching)	Deployed at the edge of the network	Easy to deploy	Single point of failure
Reverse proxy caching	Deployed near origin	Server farm management	Single point of failure
Transparent proxy caching	Intercepting HTTP request	Eliminate Single point of failure	Violates end to end statement

Active web caching	Applets, Caching for dynamic documents	Caching for dynamic documents and personalized cache	Issue of privacy
Architecture	Description	Advantage	Disadvantage
ADAPTIVE WEB CACHING	Optimizing global data distribution, Consists of multiple, distributed caches which dynamically join and leave cache groups; CGMP, CRP	Tackling “hotspot” phenomenon	Assumption: Development of cache clusters across administrative boundaries is not an issue
PUSH CACHING	To keep cached data close to those clients requesting that information (Concept of mirror site)	Targeted providers	Assumption: Ability to launch caches which may cross administrative boundaries
INTELLIGENT WEB CACHING	Applying neural network and network analysis	Adaptable to environment	High computational. Applied n 3-tier design
Mobile Environment for Intelligent Genetic search an Proxy Caching	Implementation of mobile agents with genetic search an proxy caching algorithms	Efficient search for group of people share interests in some subject	Spend lot of time for fetching documents from the internet onto the local disk
Hybrid cache index forwarding for mobile WWW	Hybrid Mowgli WWW and CINDEX schemes to send the caching data information on mobile hosts per document to the base station and transfers all of the cache index data from the old base station to the new one during the handover phase	Supports for high mobility of mobile hosts and provides a high cache hit ratio	Suffer from wireless network delays. Cache index transmission is performed by the base station

4.1 CO-OPERATE CACHING

In cooperative caching, caches communicate with peers over the network before demanding files from the web as shown in Figure 2.3. Cooperative proxy caching is the sharing and coordination of caches among multiple caching proxies [45, 46,47]. Cooperation among caches can be performed in horizontal and vertical orthogonal dimensions. Horizontal cooperation is performed by geographically clustered caches that have equal distant to the web servers. Vertical cooperation is performed by geographically distributed caches that have unequal distant to the web servers.

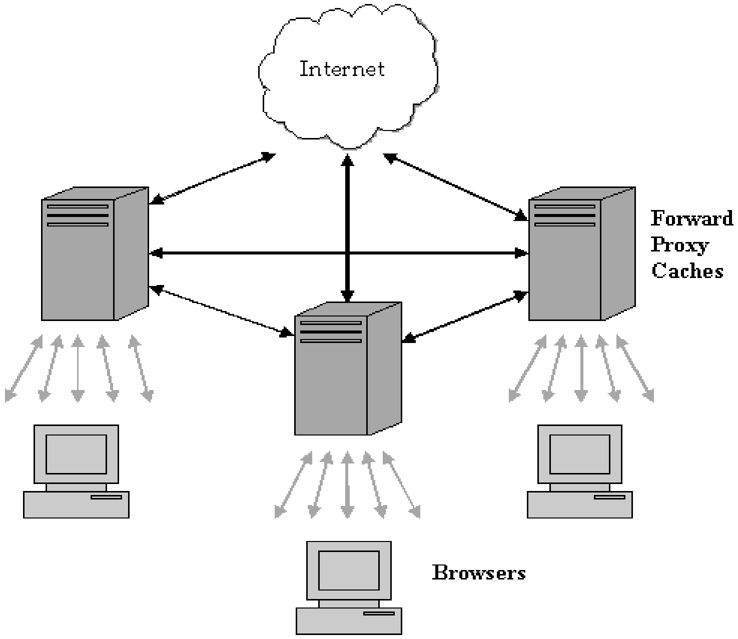

Figure 9. Co operative caching

An expiration age-based scheme is proposed in [48] for reducing the replication of documents across the cache group, while ensuring that a copy of the document always resides in a cache where it is likely to stay for the longest time. In buddy Cache approach, a group of close-by collaborating clients are connected to a storage repository. They can avoid interactions with the server if needed objects and coherency information are available in any client in the group [49]. Cooperative caching architectures can be divided into hierarchical, distributed and hybrid caching architectures [50].

4.2 TRANSPARENT WEB CACHING

Transparent web caching uses network devices to redirect HTTP traffic to cache servers. This technique is called transparent because Web browsers do not have to be explicitly configured to point to a cache server [51]. There are two ways to deploy transparent proxy caching: at the switch level and at the router level. Router based transparent caching uses policy-based routing to direct requests to the appropriate caches. In switch based transparent caching, the switch acts as a dedicated load balancer [54]. The clients and cache servers are connected to the Layer 5 switch as in Figure 10. Layer 5 switches use information in the TCP and HTTP request headers. This improves performance by ensuring that non- cacheable HTTP requests bypass the cache servers. LB-L5 Web caching scheme uses the Layer 5 switching based technique to support distributed web caching [52]. LB-L5 uses a weighted bloom filter to represent cache content and access frequency information. LB-L5 extends ICP to support communication between cache servers and L5 switches [53]. In this method, when the number of URL's in the cache directory increases, there is a chance of increasing false positive. Thus remote hit ratio is reduced.

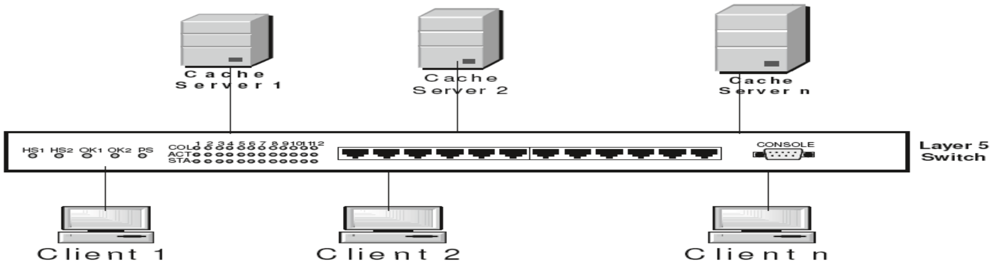

Fig 10. Transparent web caching

4.3 En-route Caching

In this architecture, the caches are associated with routing nodes in the network. These routing nodes are called en-route caches [55]. An en-route cache intercepts all client requests passing through the associated routing node as seen in Figure 11. If the requested object is in the cache, the object is sent to the client and the request is not propagated further upstream. Otherwise, the routing node forwards the request along the regular routing path towards the origin server. If no en-route cache has the target object, the request is eventually serviced by the origin server. En-route caching has the advantage that it is transparent to both origin servers and clients. Since no request is detoured off the regular routing path, the additional bandwidth consumption and network delay for cache misses are minimized. The extra overhead needed for locating the cache server by sending broadcast queries or maintaining cache summaries are eliminated in en-route caching.

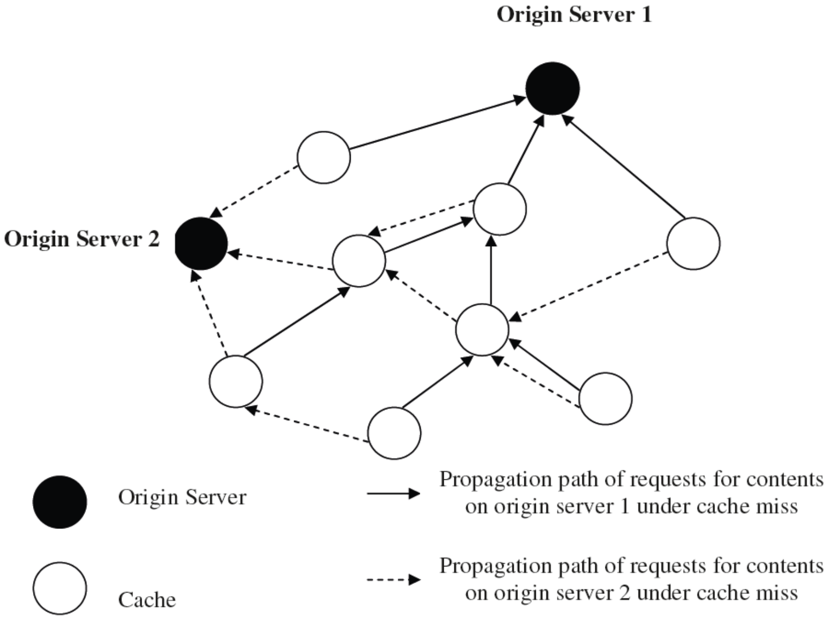
Origin Server 1
Origin Server 2
Origin Server
Cache
Propagation path of requests for contents on origin server 1 under cache miss
Propagation path of requests for contents on origin server 2 under cache miss

Figure 11. En-route caching[55]

4.4 Cascaded caching

Cascaded caching is also known as multi-layer caching. Multiple copies of same object maybe available in many caches placed at different locations. There can be cache servers both at the client location and server location. This allows a cache server at client location to get a requested document from the cache server at the server location. The overall performance of cascaded caching depends on how the cache contents are managed, including object placement and replacement algorithms [56]. Figure 12 shows that normally more than one cache is located between the origin servers and the clients.

Dynamically determining the appropriate number of object copies and placing them in suitable caches are challenging tasks. Allocating too many copies of unpopular objects in the network is wasteful of cache space while assigning too few copies of popular objects may not reduce their access latency sufficiently. Cascaded caching has the advantage that URLs can be processed by the content server aliases, so that requests from firewall proxies can be forwarded to their destinations as directly as possible.

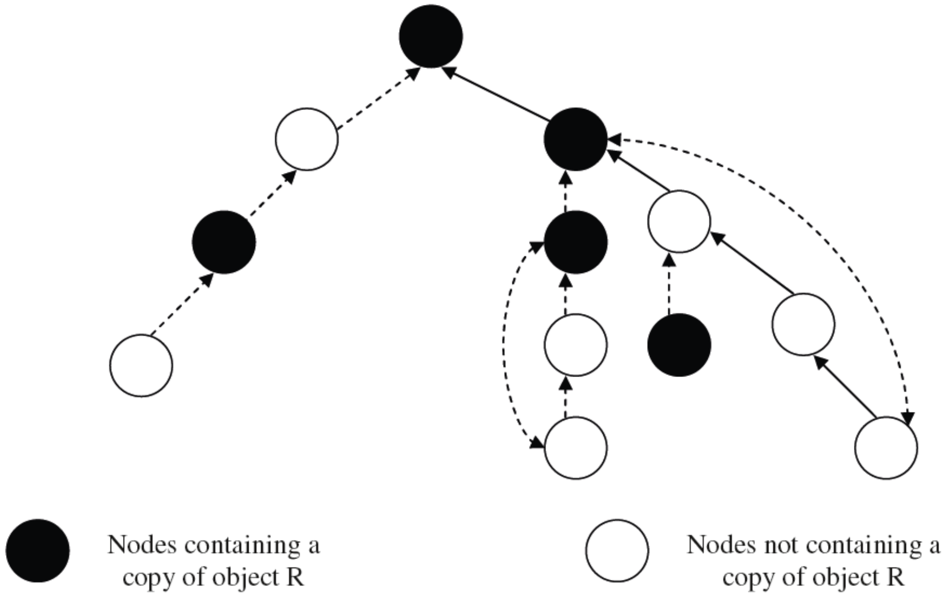
Nodes containing a copy of object R
Nodes not containing a copy of object R

Figure 12. Cascaded caching[56]

4.5 Mobile Caching

Web caching is necessary for sharing content over a mobile network and to reduce the latency in accessing the objects. When downloading, users can store content in a local cache and share it with other users they meet, e.g., via Bluetooth or WiFi. The storage capacity of mobile devices is typically limited. Which content a user should store in the cache is to be identified. Psephos is a novel mechanism for determining the caching policy of each mobile user [61]. Psephos is fully distributed: users compute their own policies individually, in the absence of a central authority. It is designed for a heterogeneous environment, in which demand for content, access to resources, and mobility characteristics may vary across different users. An Integrated Prefetching and Caching Scheme for Mobile Caching System is proposed in [57]. Neighbor Graph Cache (NGC) mechanism is proposed in [58] to reduce scanning latency while a mobile station tries to make a link-layer handover. Energy Efficient Cache Replacement Policies for Cooperative Caching in Mobile Ad-hoc Networks are proposed in [59]. With the rapidly increasing use of mobile personal devices, efficient mobile collaborative virtual environment (MCVEs) applications based on Gnutella network are to be built. An optimization mechanism for Gnutella network to meet the mobility and the network requirements of collaborative virtual environments (CVEs) is proposed in [60]. The proposed work comprises an enhancement of the Gnutella network through an efficient overlay formation protocol, and a novel caching optimization technique implemented in specific nodes selected through a Gnutella Ultrapeer system (GUS) mechanism.

5. CONCLUSIONS

Lot of research is going on in web prefetching in various directions. In this paper various web prefetching techniques and other directions of web prefetching are analyzed and discussed. The web prefetching scheme focus on the property spatial locality of web objects. These techniques are applied to reduce the network traffic and improve the user satisfaction. Web prefetching and caching can also be integrated to get better performance. At present, web prefetching and caching in mobile environment opens a vast scope in the research area.

REFERENCES

[1] http://shika.aistnara.ac.jp/products/wcol/tec h/p_concept.html dated 7/9/2011.

[2] Evangelos P. Markatos & Catherine E. Chronaki, (1998)"A Top-10 Approach to Prefetching on the Web, Proceedings of INET' 98 (The Internet Summit).

[3] Seung Won Shin, Byeong Hag Seong & Daeyeon Park, (2000)"Improving World-Wide- Web Performance Using Domain-Top Approach to Prefetching", Fourth International Conference on High-Performance Computing in the Asia-Pacific Region vol. 2, pp. 738-746.

[4] Cheng-Zhong Xu & Tamer I. Ibrahim, (2004) "A Keyword-Based Semantic Prefetching Approach in Internet News Services", in IEEE Transactions on Knowledge and data engineering, vol.16, pp. 601-611.

[5] Achuthsankar S. Nair & J. S. Jayasudha,(2007) "Dynamic Web Pre-fetching Technique for Latency Reduction", International Conference on Computational Intelligence and Multimedia Applications.

[6] Reinhard P. Klemm,(1999)" WebCompanion: A Friendly Client-Side Web Prefetching Agent", IEEE Transactions on knowledge and data engineering, pp. 577-594.

[7] Chii Chang, Sea Ling & Shonali Krishnaswamy,(2011) "Proactive Mobile Web Service Provision Using Context-Awareness", in 8th IEEE International Workshop on Managing Communications and Services, pp. 69-74.

[8] Zhimei Jiang & Leonard Kleinrock,(1998) "Web Prefetching in a Mobile Environment", in IEEE Personal Communications, pp. 25-34.

[9] Sarina Sulaiman, Siti Mariyam Shamsuddin & Ajith Abraham, (2009)"Rough Neuro-PSO WebCaching and XML Prefetching for Accessing Facebook from Mobile Environment", in World Congress on Nature & Biologically Inspired Computing.

[10] Sarina Sulaiman, Siti Mariyam Shamsuddin & Ajith Abraham,(2010) "Intelligent Mobile Web Pre-fetching (IMWeP) Using XML Technology", 6th IEEE International Conference on Next Generation Web Services Practices, pp. 475-480.

[11] Sarina Sulaiman, Siti Mariyam Shamsuddin, Ajith Abraham, (2009)"Rough Web Caching", Springer Verlag, pp. 187-211.

[12] Xing Dongshan & Shen Junyi,(2002)"A new markov model for web access prediction", pp. 34-39.

[13] Xin Jin & Huanqing Xu, (2003)"An Approach to Intelligent Web Pre-fetching Based on Hidden Markov Model", in IEEE Conference on Decision and Control , pp. 2954-2958.

[14] Weifeng Zhang, Baowen Xu, William Song, Hongji Yang & Kecheng Liu,(2000) "Data Mining Algorithms for Web Pre-Fetching", Proceedings of IEEE International Conference on Web Information Systems Engineering.

[15] Darin Fisher & Gagan Saksena, (2003)"Link Prefetching in Mozilla : A Server-driven Approach", Proceedings of 8th International Workshop on Web Content Caching and Distribution.

[16] http://www.w3.org/protocols/rfc2616/rfc2616.html dated 01/02/2011.

[17] Cheng-Zhong Xu &Tamer I. Ibrahim,(2003) "Towards Semantics-Based Prefetching to Reduce Web access latency '', Proceedings of the Symposium on Applications and the Internet.

[18] Themistoklis Palpanas & Alberto Mendelzon,(1998) “Web Prefetching Using Partial Match Prediction”, Proceedings of 4th Web Caching Workshop.

[19] Xin Chen & Xiaodong, Zhang,(2003) “A Popularity-Based Prediction Model for Web Prefetching”,IEEE Computer, Vol. 36, No. 3, pp. 63-70.

[20] Venkata N. Padmanabhan & Jeffrey C. Mogul,(1993) “Using Predictive Prefetching to Improve World Wide Web Latency”, Computer Communication Review journal, Vol. 26, No. 3, pp. 22-36.

[21] Yingyin Jiang, Min-You Wu & Wei Shu,(2002) “Web Prefetching: Costs, Benefits and Performance”, Proceedings of 7th International Workshop on Web Caching & Distribution.

[22] Qiang Yang & Zhen Zhang,(2001) “Model Based Predictive Prefetching”, Proceedings of IEEE hInternational Workshop on Database and Expert Systems Applications.

[23] P.Cao & S. Irani ,(1997) “Cost-aware WWW proxy and caching algorithms ”,in USENIX Symposium on Internet Technologies and Systems.

[24] L.Cherkasova,(1998) “ Improving WWW proxies performance with Greedy-Dual-Size-Frequency Caching Policy”, in HP Technical report.

[25] Brian D. Davison,(1999) “Adaptive Web Prefetching”, Proceedings of 2nd Workshop on Adaptive Systems and User Modelling on the World Wide Web, Toronto,pp. 105-106.

[26] Zhimei Jiang & Leonard Kleinrock,(1998) “An Adaptive Network Prefetch Scheme”, IEEE Journal on Selected Areas in Communications, Vol. 16, No. 3, pp. 358-368.

[27] Xin Chen, Xiaodong Zhang,(2003) “Accurately Modeling Workload Interactions for Deployinger Prefetching in Web Servers”, Proceedings of IEEE International Conference on Parallel Processing.

[28] Tiziano Fagni, Raffaele Perego, Fabrizio Silvestri, Salvatore Orlando, “Boosting the Performance of Web Search Engines: Caching and Prefetching Query Results by Exploiting Historical Usage Data”, ACM Transactions on Information Systems, Vol. 24,No. 1, pp. 51-78.

[29] Ravi Kokku, Praveen Yalagandula & Arun Venkataramani, Mike Dahlin,(2003) “NPS : A Non-Interfering Deployable Web Prefetching System”, Proceedings of USENIX Symposium on Internet Technologies and Systems.

[30] Qiang Yang, Haining Henry Zhang & Tianyi Li,(2001) “Mining Web Logs for Prediction Models in WWW Caching and Prefetching”, Proceedings of 7th ACM International Conference on Knowledge Discovery and Data Mining, pp. 473-478.

[31] Yin-Fu Huang & Jhao-Min Hsu,(2005) “Mining Web Logs to Improve Hit Ratios of Prefetching and Caching”, Proceedings of IEEE/WIC/ACM International Conference on Web Intelligence.

[32] Bin Lan, Stephane Bressan, Beng Chin Ooi & Kian-Lee Tan, (2000) “Rule-Assisted Prefetching in Web-Server Caching”, Proceedings of ACM Conference on Information and Knowledge Management, pp. 504-509.

[33] Alexandros Nanopoulos, Dimitrios Katsaros & Yannis Manolopoulos,(2003) “A Data Mining Algorithm for Generalized Web Prefetching”, IEEE Transactions on Knowledge and Data Engineering, Vol. 15, No. 5, pp. 1155-1168.

[34] Areerat Songwatta,(2008) “Mining Web Logs for Prediction in Pre-fetching and Caching”, Proceedings of IEEE International Conference on Convergence and Hybrid Information Technology, pp. 1006-1011.

[35] Yuna Kim & Jong Kim,(2003) “Web Prefetching Using Display-Based Prediction”, Proceedings of IEEE International Conference on Web Intelligence.

[36] Manolis Veliskakis, John Roussos, & Panos Georgantas,(2005) Timos Sellis, “DOMProxy: Enabling Dynamic-Content Frontend Web Caching”, Proceedings of 10th International Workshop on Web Content Caching and Distribution, IEEE Computer Society September, pp. 56-61.

[37] Athena I. Vakali & George E. Pallis,(2011) “A Study of Web Caching Architectures and Performance”, Proceedings of 5th World Multi-conference on Systemics, Cybernetics and Informatics.

[38] Clinton L. Jeffery & Samir R. Das,(1996) Garry S. Bernal, “Proxy-Sharing Proxy Servers”,Proceedings of 1st IEEE Conference on Emerging Technologies and Applications in Communication, pp. 116-119.

[39] Greg Barish & Katia Obraczka, (2000) “World Wide Web Caching: Trends and Techniques”,IEEE Communication, Vol. 38, No. 5, pp. 178-184.

[40] Louis Degenaro, Arun Iyengar & Isabele Rouvellou,(2001) “Improving Performance with Application-Level Caching”, Proceedings of International Conference on Advances in Infrastructure for Electronic Business, Science and Education on the Internet.

[41] Prasenjit Sarkar, & John H. Hartman, (2000)“Hint-Based Cooperative Caching”, ACM Transactions on Computer Systems, Vol. 18, No. 4, pp. 387-419.

[42] Hykyung Bahn,(2004) “A Shared Cache Solution for the Home Internet Gateway”, IEEE Transactions on Consumer Electronics, pp. 168-172.

[43] King Yeung Wong & Kai Hau Yeung,(2001) “Site–Based Approach to Web Cache Design”, IEEE Internet Computing Journal, Vol. 5, pp. 28-34.

[44] Sarina Sulaiman, Siti Mariyam Shamsuddin, Ajith Abraham & Shahida Sulaiman, (2008) “Web Caching and Prefetching: What, Why and How?”, Proceedings of IEEE International Symposium on Information Technology, pp. 26-28.

[45] Jean Marc Menaud, Valerie Issarny & Michel Banatre,(2000) “A Scalable and Efficient Cooperative System for Web Caches”, IEEE Concurrency, Vol.8, No. 3, pp.56-62.

[46] Radhika Malpani, Jacob Lorch & David Berger, (1995)“Making World Wide Web Caching Servers Cooperate”, Proceedings of 4th International World Wide Web Conference, Boston.

[47] Sandra G. Dykes, Kay A. Robbins,(2001) “A Viability Analysis of Cooperative Proxy Caching”,Proceedings of Annual Joint Conference of the IEEE Computer and Communications Socities,pp. 1205-1213.

[48] Lakshmish Ramaswam & Ling Liu,(2004) “An Expiration Age-Based Document PlacementvScheme for Cooperative Web Caching”, IEEE Transactions on Knowledge and Data Engineering, Vol. 16, No. 5, pp. 585-592.

[49] Magnus E. Bjornsson, Liuba Shrira, (2002)“BuddyCache: High-Performance Object Storage for Collaborative Strong-Consistency Applications in a WAN”, Proceedings of the 17th ACM conference on Object-oriented programming, systems, languages and applications, pp. 26-39.

[50] Mingkuan Liu, Fei-Yue Wang & Daniel Zeng, Lizhi Yang,(2001) “An Overview of World Wide Web Caching”, Proceedings of IEEE International Conference on Systems, Man and Cybernetics, USA, Vol. 5, pp. 3045-3050.

[51] Zhengang Liang, Hossam Hassanein & Patrick Martin,(2001) “Transparent Distributed Web Caching”, Proceeding of 26th Annual IEEE Conference on Local Computer Networks, pp. 225-231.

[52] Qing Zou, Patrick Martin & Hossam Hassanein,(2003) “Transparent Distributed Web Caching with Minimum Expected Response Time”, Proceedings of IEEE International Conference on Performance, Computing and Communications, pp. 379-386.

[53] Hossam Hassanein, Zhengang Liang & Patrick Martin,(2002) “Performance Comparison of Alternative Web Caching Techniques”, Proceedings of 7th IEEE International Symposium Computers and Communications.

[54] Athena I. Vakali, George E. Pallis, (2001)“A Study of Web Caching Architectures and Performance”, Proceedings of 5th World Multi-conference on Systemics, Cybernetics and Informatics.

[55] Xueyan Tang & Samuel T. Chanson, (2002)“Coordinated En-Route Web Caching”, IEEE Transactions on Computers, Vol. 51, No. 6.

[56] Xueyan Tang & Samuel T. Chanson,(2003) “Coordinated Management of Cascaded Caches for Efficient Content Distribution”, Proceedings of 19th IEEE International Conference on Data Engineering, pp. 37-42.

[57] Beihong Jin, Sihua Tian, Chen Lin, Xin Ren & Yu Huang, (2007)“An Integrated Prefetching and Caching Scheme for Mobile Caching System”, Proceedings of IEEE International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, pp. 512-527.

[58] Chung-Sheng Li, Yung-Chih Tseng, Han-Chieh Chao & Yueh-Min Huang, (2008)“A neighbour caching mechanism for handoff in IEEE 802.11 wireless networks”, The Journal ofSupercomputing , Volume 45, Number 1, pp. 1-14.

[59] Wenzhong Li, Edward Chan & Daoxu Chen,(2007) “Energy Efficient Cache Replacement Policies for Cooperative Caching in Mobile Ad-hoc Networks”, Proceedings of IEEE International Conference on Wireless Communications and Networking, pp. 3349-3354.

[60] Boukerche, A., Zarrad, A. & Araujo, R.B.,(2008) “Novel Optimized Caching Technique for Mobile Gnutella Based Network to Support Large-Scale Collaborative Virtual Environment”, Proceedings of IEEE International Symposium on Simulation, pp.289-297.

[61] Stratis Ioannidis, Laurent Massoulie, Augustin Chaintreau, “ Distributed Caching Over Heterogeneous Mobile Networks”, ACM SIGMETRICS Performance Evaluation Review, Vol. 38, Issue 1, 2010.

Authors

Greeshma G. Vijayan, received her B.Tech degree in Computer Science from Cochin University of Science and Technology, Kerala. Currently she is doing her M.Tech in Computer Science and Engineering under Kerala University at SCT College of Engineering , Kerala. Her fields of interest are web pre-fetching, web caching and data mining

Jayasudha J. S., received her B.E degree in Computer Science from R.V.S. College of Engineering, Dindigul, M.E degree in Computer Science from NIT, Trichy and Ph.D degree in Computer Science from Kerala University, Kerala in 2008. She is now the Associate Professor of Computer Science Department at SCT College of Engineering , Kerala. She is an active researcher in web prefetching, web caching and has published more than 10 papers in journals and Conference Proceedings.