Mobile Information Systems 3 (2007) 1–18
IOS Press

Cooperative proxy caching for wireless base stations

James Z. Wang*, Zhidian Du and Pradip K. Srimani
Department of Computer Science, Clemson University, Box 340974, Clemson, SC 29634, USA
Tel.: +1 864 656 7678; Fax: +1 864 656 0145; E-mail: {jzwang, zdu, srimani}@cs.clemson.edu

Abstract. This paper proposes a mobile cache model to facilitate the cooperative proxy caching in wireless base stations. This mobile cache model uses a network cache line to record the caching state information about a web document for effective data search and cache space management. Based on the proposed mobile cache model, a P2P cooperative proxy caching scheme is proposed to use a self-configured and self-managed virtual proxy graph (VPG), independent of the underlying wireless network structure and adaptive to the network and geographic environment changes, to achieve efficient data search, data cache and date replication. Based on demand, the aggregate effect of data caching, searching and replicating actions by individual proxy servers automatically migrates the cached web documents closer to the interested clients. In addition, a cache line migration (CLM) strategy is proposed to flow and replicate the heads of network cache lines of web documents associated with a moving mobile host to the new base station during the mobile host handoff. These replicated cache line heads provide direct links to the cached web documents accessed by the moving mobile hosts in the previous base station, thus improving the mobile web caching performance. Performance studies have shown that the proposed P2P cooperative proxy caching schemes significantly outperform existing caching schemes.

Keywords: Wireless internet, base station, proxy caching, VPG, mobile cache model, network cache line

1. Introduction

The popularity of wireless networks grows with the advances in wireless technologies and Internet applications. Many interesting web applications, such as multimedia streaming, have flourished with the growth of the Internet and WWW. New generation wireless networks, such as G4 and WiMAX, have brought these web applications into wireless. Figure 1 depicts a typical wireless Internet architecture. In this architecture, mobile hosts access the mobile network through base stations, which are interconnected by access routers to form wireless LANs, and in turn are connected to the Internet through gateway routers. Among numerous studies on enhancing the wireless Internet performance, caching popular web documents at locations close to the mobile clients is an effective solution to improving the quality of wireless web applications [1–7]. Web caching can be implemented in various points of a wireless network. Wireless providers often install cache appliances on the edge of a wireless network to act as the proxy to the Internet. With this approach, the wireless web performance is often compromised by the long latency between mobile clients, proxy cache, and original web servers, because the proxy server is located outside the wireless network. A single proxy cache at the edge of a wireless network can also be overloaded and become the bottleneck.

*Corresponding author.

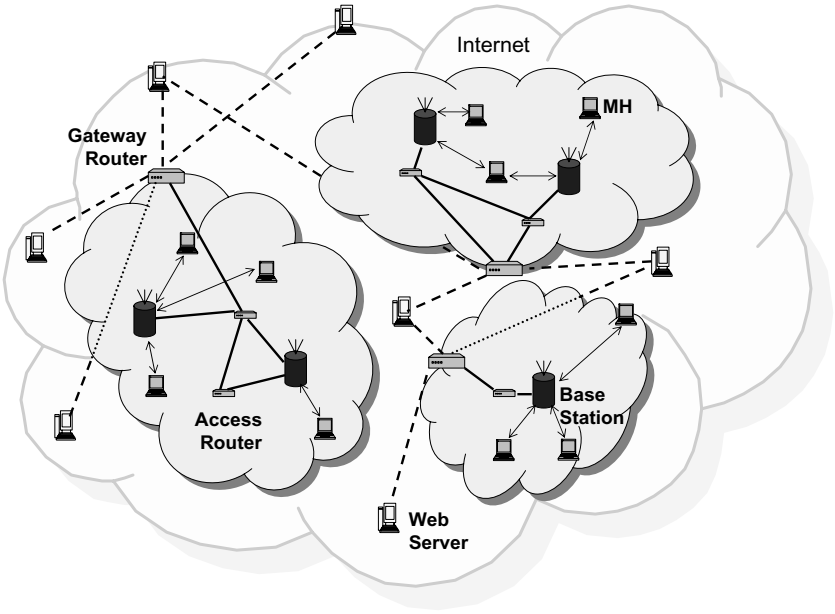

Fig. 1. Wireless Internet architecture.

To address these drawbacks, wireless base stations are often used as caching proxies for mobile hosts [8]. Caching popular web documents in wireless base stations can ease network traffic between base stations and web servers, and reduce user web request latencies. Providing caching mechanisms in wireless base stations can also reduce the connection time between the mobile host and the base station due to the reduced waiting time experienced in fetching remote web documents, thus saving the limited wireless bandwidth.

However, independent proxy caching in wireless base stations may not be optimal for wireless web applications because mobile devices often switch wireless services between adjacent base stations. It is desirable to make the wireless base stations to cooperate in proxy caching [8,9]. A simple solution to designing a cooperative proxy cache system for wireless base stations is to adopt an existing cooperative proxy caching scheme that was originally designed for wired network environments. There are two kinds of cooperative proxy caching architectures in wired network environments. The hierarchical cooperative proxy caching schemes [10–12] require proxy servers being placed at key access points of networks and configured as a hierarchy. The hierarchical cooperative proxy caching is not suitable for wireless environments due to the non-hierarchical nature of the distributed wireless base stations. On the other hand, existing distributed cooperative proxy cache systems [3,4,13–19] in wired network environments employ sophisticated caching and searching schemes, such as centralized or distributed directory lookup, distributed hashing, and multicasting, to distribute and search the cached web documents. These proxy caching schemes all inherit the vertical cache model used in operating systems and storage systems. In a vertical cache mode, the proxy cache acts solely as the agent between the server and the client. However, in a distributed cooperative proxy cache system, web documents may come from cooperative proxy servers. It is necessary to design a new cache model that can take advantage of the cooperation among distributed proxy servers. In addition, the objective of caching web documents in wireless base stations is to reduce the user request latencies so that the wireless network bandwidth and mobile battery power are saved. Nonetheless, most of these existing caching schemes are originally designed for

cluster environments where proxy servers are physically close to each other. They often concentrate on increasing the cache hit ratio (the ratio of web requests being satisfied by the proxy cache system) while ignoring the cost of retrieving web documents from proxy servers where they are cached [20,21]. A cooperative proxy caching scheme should emphasize on moving the cached web documents closer to interested clients through data replication.

Besides the aforementioned problems, new challenges exist in the cooperative proxy caching for wireless base stations. First, mobile hosts in wireless environments often switch services from one base station to another. Mobile host handoffs increase the complexity of the cooperative proxy cache system. Second, reducing the user request latency (the time for a user to wait for the first portion of the requested document to arrive) is more critical in wireless environments since wireless links have limited bandwidth and mobile hosts have limited battery power. Finally, in wired networks, a caching proxy server usually serves clients within an organization, in which clients usually share common interests. However, in wireless environments, clients connecting to the same base station might come from different regions and have different backgrounds, hence having different web access interests. Such diversity of client interests directly affects the cache strategies and replacement policies. To address these challenges in the wireless proxy caching, we propose a P2P cooperative proxy caching scheme that uses a mobile cache model to facilitate the proxy server cooperation, and a virtual proxy graph to assist the P2P data cache, data search and data replication.

The rest of the paper is organized as follows. In Section 2, we propose the P2P cooperative proxy caching scheme, including the mobile cache model and virtual proxy graph, and discuss the data cache, data search and data replication strategies. After analyzing the performance advantages of the proposed schemes in Section 3, we verify our formal analysis using simulation studies in Section 4. Finally we have our concluding remarks and discuss future studies in Section 5.

2. A P2P cooperative proxy caching scheme

2.1. Mobile cache model

Designing a cache model has to be based on the characteristics of the cached data and the specific caching environment. In wireless proxy caching environments, the cached web documents may be migrated to this base station proxy server from other base stations instead of being directly fetched from the original web servers. To cope with the characteristics of cached web documents and wireless environments, we design a network cache line as depicted in Fig. 2.

The network cache line consists of two portions. The cache line body is the cached web document. The head portion includes *ID, Tag, State bytes, Link fields, Client list* and *Origin*. The *ID* field contains a *UUID* which the *URL* of the cached web document is hashed to. *Tag* is the name of the cached web document. *State bytes* are used to store caching state information about the cached web document. Because the cached web document might be migrated from other base station proxy servers, *link fields* are needed to provide links to proxy servers that previously owned or searched for this document, so that subsequent web requests in nearby base stations can easily find this cached web documents. *Link field* is organized as a pair of integer $< NID, Dist >$. *NID* is an integer used to lookup the neighbor table for a neighbor base station's IP address. *Dist* is round-trip network distance to reach the cached web document through the base station specified by *NID*. *Client List* contains the IDs of the mobile hosts that are currently accessing the cached web document. *Origin* indicates whether this current web replica is fetched from the original web server or is migrated from another proxy server in the P2P cooperative proxy cache system.

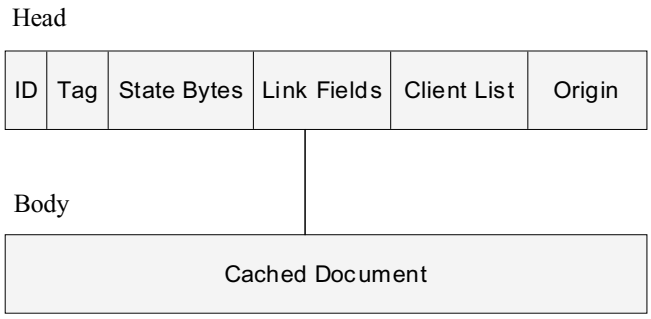

Fig. 2. Network cache line.

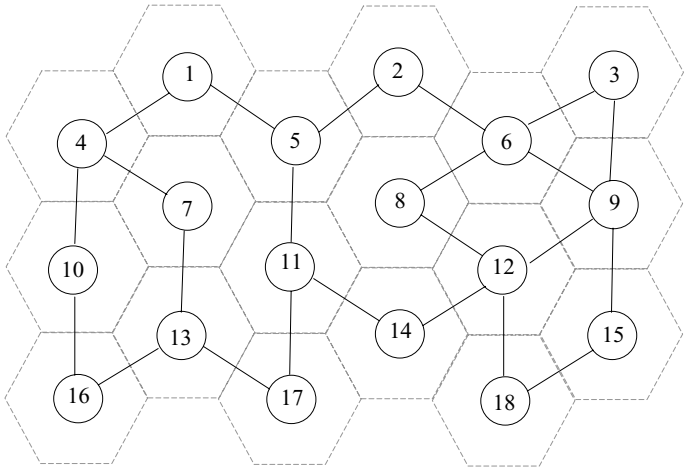

Fig. 3. Virtual proxy graph.

2.2. Virtual proxy graph

Since the cached web replicas in nearby base station proxy servers can be linked using network cache lines, we propose a virtual proxy graph (VPG), an overlay network independent of the underlying wireless network structures, to facilitate the data linkage and data exchange among the caching proxy servers. A VPG forms the foundation for data search, data cache and data replication in our P2P cooperative proxy cache system. With the VPG, A proxy server only exchanges data and information with its neighbor proxy servers. Figure 3 demonstrates a VPG with 18 nodes. In this VPG, proxy node 12 has 4 neighbors. They are proxy nodes 8, 9, 14, and 18.

In the wireless environment, there is no centralized controller to build the virtual proxy graph among the distributed wireless base stations. Ideally, a VPG should be automatically configured by individual base station proxy servers through simple information exchange. It is also desirable that the VPG can be automatically adjusted to fit the dynamic changes of the wireless network and web caching requirement. We propose an individual-based VPG configuration process for proxy servers to automatically configure and manage the VPG for the P2P cooperative proxy cache system in following two phases:

Initialization phase:

– When a base station proxy server wants to join the P2P cooperative proxy cache system, it first

broadcasts a request message to all other base stations in the wireless network, asking for the permission token.
- If within certain period of time this base station does not get any response from any other base stations or no proxy servers in the wireless network has been found holding the permission token, it creates a token and assumes itself to be the first proxy server in the P2P cooperative proxy cache system.
- If there are existing proxy servers in the P2P cooperative proxy cache system, one of them must hold the token. The proxy server holding the permission token must pass the token to the requesting proxy server after finishing its own configuration. All other proxy servers already in the P2P cooperative proxy cache system must notify the requesting proxy server that they are part of the cache system.
- After getting the permission token, the joining proxy server collects characteristic information, such as storage capacity, network bandwidth, number of connections, user tolerance to request latencies, from the exiting proxy servers.
- The maximum number of virtual links a proxy server can have with other proxy servers should be proportional to its computing and network resources. The joining proxy server selects its neighbor proxy servers so that the network distance between a pair of neighbor proxy servers does not exceed the tolerable user request latencies in both servers. After the new connections, the proxy servers connecting to the joining proxy server have to make sure that their numbers of virtual links are still proportional to their processing powers. During initialization, a neighbor table will be built in this joining proxy server. An entry in the neighbor table includes an ID which is normally a small integer and the IP address of the neighbor proxy server. The proxy servers connecting to the joining proxy server also add this proxy server to their neighbor tables.

Self-adjusting phase:

After becoming a port of the VPG, a proxy server enters into the self-adjusting phase. In this phase, a proxy server continuously monitors its data and information exchanges with its neighbor proxy servers.

- If a proxy server finds that its data exchange with one of its neighbor proxy servers is lower than a threshold for a certain period of time, it can propose to that neirghbor proxy server to drop the virtual connection. The connection is dropped only if both servers agree to.
- If a proxy server finds that many cached web replicas routed to this server are originally from a certain proxy server, it can negotiate with that proxy server to form a new virtual link if their network distance does not exceed the tolerable user request latencies in both servers.

This dynamic self-adjusting process makes sure that the VPG is always effective for data search, data cache and data replication.

2.3. Data cache and data search strategies

An critical task of our P2P cooperative proxy caching scheme for wireless base stations is to design a data search and data cache strategies so that individual proxy servers can manage the cached data based on their local knowledge of the global caching state using the self-configured VPG and the mobile cache model. The objective is to allow individual proxy servers making data search, data cache and data replication decisions independantly, hoping the aggregate effect of individual proxy server actions generate the group behavior to manage the global caching state of the P2P cooperative proxy caching system.

When a new web request reaches a certain proxy server from its mobile client, the proxy server takes actions based on three different situations as follows:

1. **The entire network cache line is cached in this proxy server:** In this case, the cached web document is sent to the mobile client and the web request is satisfied.
2. **Only the head of the network cache line is cached in this proxy server:** This proxy server checks the head of the network cache line for the shortest distance to the cached web replicas. If the distance to the nearest web replica is larger than its expected response time by directly requesting from the original web server, a query is sent to the original web server. Otherwise, a query message will be propagated along the links in the heads of network cache lines to the proxy server that holds the nearest web replica.
3. **Nothing is cached at this proxy server:** This proxy server first sends a query message to all its neighbor proxy servers and waits for responses from them. The query message will be disseminated to other proxy servers through neighbor relay until the message life time expires or a matching cache line (maybe head only) is reached. The proxy server that has the matching cache line responds the query with the meta information about the cached replica in a responding message, which is routed back to the requesting proxy server along the query path. The proxy servers on the routing path create or update the appropriate network cache line heads based on the information in the responding message to form links to the cached web replica. The proxy server that initiates the query message collects all responding messages arriving within a specified time period and creates a network cache line for the web document based on the information in these responding messages. After that, the search process become the same as in case 2.

When a query locates a cached web document in the P2P cooperative proxy cache system, the cached web document will be transferred to the requesting proxy server directly from the proxy server where it is located. Unlike the existing hashing based schemes in which data replication is avoided, the web document is replicated at the requesting proxy server expecting an increasing demand for this web document. Replication of the web document is based a supply-demand management principle similar to that used in the real-world economic systems, i.e., increasing supplies based on demand, a poven mechanism to automatically manage the complexity of distributing goods among customers [22].

2.4. Advantages of the P2P cooperative proxy caching scheme

The following excellent features of the proposed P2P cooperative proxy caching scheme demonstrate the advantages of the proposed data cache, data search and data replication strategies:

- **Data Cohesion:** Because individual proxy servers on the search path create and update the heads of network cache lines to establish links to the cached web replicas, nearby cached web replicas will be linked together to form a cohesive group. The data cohesion enable the nearby proxy servers to effectively cooperate in making cache replacement decisions.
- **Search Alignment:** By caching and updating the network cache line heads in individual proxy servers during search, later web requests can quickly locate the cached web replicas by following the traces of previous web requests. This search alignment feature can not only reduce the user request latencies but also reduce the network overhead generated by search messages.
- **Replication-by-demand:** When a web request is satisfied by a cooperative proxy server in the P2P proxy cache system instead of the local proxy server, the web document is replicated at the local proxy server, expecting a increasing demand for this document. The replication-by-demand feature not only moves the cached web documents to the interested clients but also automatically balances the workload by distributing highly demanded web documents to multiple caching locations.

2.5. Adapting to wireless environments

Due to the existence of mobile host handoffs in a wireless network, the VPG self-configuration and self-adjustment has to consider the base station's geographic locations. In general, the proxy servers on two geographically close base stations should be connected as neighbors in the VPG since handoffs might happen between them. However, a proxy server itself does not know which other proxy servers in the wireless network are geographically close to it. One solution is to feed the VPG self-configuration process in individual proxy servers with the geographical parameters of all base stations provided by the wireless service provider. Nonetheless, relying on a base station distribution map to determine neighbor proxy server is not reliable. For instance, there might not be any handoff between two base station proxy servers separated by a new construction although they are geographically close to each other on the map. Connecting these two proxy servers as neighbors might not be the best interest of the P2P cooperative proxy cache system. Furthermore, the geographic environment changes dynamically from time to time. Road blocking or detouring might reduce the data and information flow between two connected neighbor proxy servers for a long period of time. The self-adjustment phase of the automatic VPG configuration process is designed to address this issue.

Furthermore, when a mobile host switches services from one base station to another, it is highly possible that the mobile host will access the same web information at the new base station after the handoff. To accelerate the web browsing, Hadjiefthymiades and Merakos [23] proposed a proxy cache relocation scheme, which tries to predict the next base station a mobile host might move to, and copies the cached web documents in a dedicated cache space for this mobile host to the predicted base stations before the handoff actually takes place. Due to the prediction inaccuracy, this scheme copies 100% of cached data to the best predicted neighbors, 70% of cached data to the second best predicted neighbors, and 30% to the rest neighbors.

There are two problems in this proxy cache relocation scheme:

1. This caching scheme assigns each mobile host a dedicated cache space in the proxy server. Recent studies found that user access to web documents follows a Zipf-like distribution with high Zipf factors [22,24]. It means user access to web documents is highly skewed to a few web documents. Although dedicated cache space simplifies the proxy cache management, it wastes the limited proxy cache space since the interest locality of mobile clients causes duplication of the same web document in the same proxy cache.
2. The inaccuracy predictions cause unnecessary data copy among the base station proxy servers. Copying web documents based on prediction of the mobile host movement may introduce data oscillation in wireless network.

Although new cache relocation techniques [25] are proposed to deal with the issue of poor path prediction by temporarily moving data objects to a common parent node prior to a handover. These cache relocation schemes still replicate web documents that may not be needed in the new base station. In addition, they did not solve the data oscillation problem due to copying web documents based on the prediction of the mobile host movement and, hence, generate excessive traffics in the wireless network. Furthermore, using a common parent node to temporarily store the web documents increases the complexity of the cache relocation and causes extra workload in the common parent node. To improve the caching performance, our proposed P2P cooperative proxy caching scheme addresses these two problems naturally. The mobile cache model enables the proxy server cooperation among the neighbor base stations without the need of assigning dedicated cache spaces for individual mobile hosts. The P2P proxy server cooperation among neighbor base stations can increase the cache hit ratio and

reduce the web request latency. When the mobile host moves from one base station to its neighbor base station, it is not necessary to move its cached web documents to the new base station prior to or at the handoff. Our P2P data search scheme can quickly find a cached web document for a mobile host from the proxy server which the mobile host previously connected to, if the web document has not been removed by the cache replacement. The user request latency of querying a cached web document from a neighbor proxy server is much less than that of fetching the data from the original web server. Furthermore, the heads of the network cache lines currently accessed by the moving mobile host can be copied to the new base station during the handoff. We call this strategy as cache line migration (CLM). Using this simple method, we can effectively reduce the cost of searching the previously accessed web documents for a mobile host because the links in the network cache line heads can directly lead to the cached web replicas in the previous base station proxy server. The overhead of migrating the network cache line heads should not be significant because the a network cache line head is usually much smaller than a cached web document.

3. Performance study

To evaluate the performance of the proposed P2P cooperative proxy caching scheme in wireless environments, we compare it with some existing proxy caching schemes using formal analysis and simulation study. The following five schemes are evaluated:

- **Cache Relocation (CR):** The proxy cache relocation scheme proposed in [23] is evaluated with the assumption of 100% accuracy in prediction of the mobile host movement, although it is unlikely for any prediction to be 100% accurate.
- **Non-cooperative Caching (NC):** In this scheme, each individual proxy server acts independently and there is no cooperation among the base station proxy servers. This scheme can be used to evaluate how proxy server cooperation impacts the performance of the wireless proxy caching.
- **Multicast-based Cooperative Caching (MCC):** This cooperative proxy caching scheme uses multicast to retrieve the cached web documents in a cooperative proxy cache system. Since it was originally designed for distributed wired network environments, it is one of the best existing proxy caching schemes that can be easily adapted into the wireless base stations. In this scheme, the application-level multicast is employed because IP multicast may not be supported in heterogeneous networks. For fair comparison, the self-configured VPG is used as the overlay network for the application-level multicast. In an multicast-based scheme, a TTL value is usually used to control the number of hops that a query message may travel in the wireless network. Without the TTL value, query messages may flood the wireless network and cause the network instable. When a query message is issued by a proxy server, a redefined TTL value is assigned to the message. The TTL value is decremented by 1 when a proxy server receives the query message. A proxy server stops forwarding the query message when the message's TTL value reaches 0. In our performance study, we choose TTL $= 3$ for a query message.
- **P2P Cooperative Caching (PCC):** This is our basic P2P cooperative proxy caching scheme which includes the individual-based data caching, searching and replicating methods using the proposed network cache model and the self-configured VPG. To ensure a fair comparison, we also set a TTL value equal to the one used in the MCC scheme for query messages in our P2P cooperative proxy caching scheme. When a proxy server can not find a network cache line matching the requested web document, it forwards the query message to their neighbor proxy servers if and only if the message's TTL is not equal to 0.

– **PCC with Cache Line Migration (PCC-CLM):** To take advantage of the mobile host handoff, this scheme enhances our P2P cooperative proxy caching scheme by migrating the network cache line heads associated with the moving mobile host during handoffs.

3.1. Performance analysis

Although it is hard to mathematically derive the absolute performance value of a cooperative proxy cache system, it is possible to evaluate the relative performance of different proxy caching schemes through formal analysis. To simplify the analysis, we assume that all caching schemes use the same network topology and all proxy servers have the same proxy cache space. We also assume that if a local cache miss happens, a proxy server sends a query message to its neighbor proxy servers and waits for the responding message for a time $t_{\max}$ before it sends a web request directly to the original web server. For the P2P cooperative proxy caching scheme, the network cache line heads are used to link the cached web documents together through data search, data cache and data replication. Thus, it is reasonable to assume that a cached web document can always be retrieved by following the links in the network cache line heads as long as the query message reaches a proxy server having a network cache line matching the requested web document.

We first compare the performance of non-cooperative caching (NC) scheme with the cache relocation (CR) approach (CR). If we assume the probability of finding a cached web document locally is P_{loc}, then the local cache hit ratio is P_{loc} as well. Thus the probability of finding a cached web document in a single non-cooperative cache is $P_{nc} = P_{loc}$. In the CR scheme, each mobile host is allocated an independent cache space in the proxy server. Assume the probability of finding a cached web document using the CR scheme is P_{cr}. If there are N mobile hosts, then the total cache space is partitioned into N partitions ($N > 0$). Since all N partitions in the CR scheme are independent, the probability of finding the cached web document in one of the N partitions obeys the Bernoulli trail. Therefore, the probability of finding the cached web document in the entire cache space follows a binary distribution:

$$P^*_{cr} = 1 - \binom{N}{0} P^0_{cr}(1 - P_{cr}) = 1 - (1 - P_{cr})^N \tag{1}$$

Thus, we have $P^*_{cr} - P_{cr} = (1 - P_{cr}) - (1 - P_{cr})^N$.
Since $N > 0$, we have $P^*_{cr} \leqslant P_{loc} = P_{nc}$, and

$$P_{cr} \leqslant P^*_{cr} \leqslant P_{loc} = P_{nc} \tag{2}$$

Inequality Eq. (2) shows that the NC scheme results in higher cache hit ratio than the CR scheme does. The CR scheme does not perform well because it assigns a dedicated cache space for each individual mobile host, resulting duplicate copies of popular documents in the same proxy server.

For the MCC scheme, if there is a local cache miss in a proxy server, the probability of finding a cached web document from another caching proxy server during $t_{\max}$ is:

$$P_{mcc} = P_{loc} + (1 - P_{loc}).P_{bc\_mcc} \tag{3}$$

where P_{bc_mcc} is the probability of finding the cached web document by broadcasting query messages to neighbor proxy servers. We always have

$$P_{mcc} \geqslant P_{loc} = P_{nc} \tag{4}$$

In the basic PCC scheme, if there is a local cache miss in a proxy server, this server broadcasts a query message to their neighbor proxy servers. We assume that the probability of finding a network cache line head within time $t_{\max}$ is P_{hd}, and the probability of finding a cached web document within the same time is P_{bc_pcc}. Then the probability of finding a cached web document using the basic PCC scheme is,

$$P_{pcc} = P_{loc} + (1 - P_{loc}).(P_{hd} + (1 - P_{hd})P_{bc\_pcc}) \tag{5}$$

Since the MCC scheme uses the VPG automatically configured by the PCC scheme as the multicast overlay network, and both the PCC scheme and the MCC scheme use the same TTL value for query messages, we have $P_{bc\_pcc} = P_{bc\_mcc}$. Based on Eqs (3) and (5), we have,

$$\begin{aligned} & P_{pcc} - P_{mcc} \\ & \quad = (1 - P_{loc}).(P_{hd} + (1 - P_{hd})P_{bc\_pcc} - P_{bc\_mcc}) \\ & \quad = (1 - P_{loc}).(P_{hd} + (1 - P_{hd})P_{bc\_mcc} - P_{bc\_mcc}) \\ & \quad = (1 - P_{loc}).(P_{hd}(1 - P_{bc\_mcc}) \\ & \quad \geqslant 0 \end{aligned}$$

Thus, we have

$$P_{pcc} \geqslant P_{mcc} \tag{6}$$

Now we study the effect of cache line migration. Assuming the probability of finding a network cache line head in a proxy server is now P_{hdmv}, we have $P_{hdmv} \geqslant P_{hd}$ due to the cache line heads moved in with mobile hosts. Thus the probability of finding a cached web document under the PCC-CLM scheme is,

$$P_{clm} = P_{loc} + (1 - P_{loc}).(P_{hdhm} + (1 - P_{hdhm})P_{bc\_clm}) \tag{7}$$

Because network cache line heads are usually much smaller than the cached web documents, these cache line heads do not take much proxy cache space. We can reasonably assume that P_{loc} in the PCC-CLM scheme is still the same as that in the basic PCC scheme, i.e., $P_{bc\_clm} = P_{bc\_pcc}$. Based on Eqs (5) and (7), we have

$$\begin{aligned} & P_{clm} - P_{pcc} \\ & \quad = P_{loc} + (1 - P_{loc}).(P_{hdhm} - P_{hd})(1 - P_{bc\_clm}) \\ & \quad \geqslant 0 \end{aligned}$$

so we have

$$P_{clm} \geqslant P_{pcc} \tag{8}$$

Based on Eqs (2), (4), (6) and (8), the cooperative proxy cache system using the CR scheme has the least cache hit ratio while the P2P cooperative proxy cache system with cache line migration have the best performance in terms of the cache hit ratio. The PCC scheme outperforms the MCC scheme due to the mobile cache model and P2P data search, data cache and data replication strategies. As expected, the non-cooperative proxy caching scheme performs worse than any of the cooperative proxy caching schemes in terms of the cache hit ratio.

However, our performance analysis only offers relative ranks of different proxy caching schemes in terms of the cache hit ratio. It is necessary to quantitatively evaluate the performance difference of these proxy caching schemes. In addition, we need to study the performance of the proxy caching schemes in terms of other performance metrics, including the user request latency and network overhead. Thus, we use simulation to further evaluate the performance of these proxy caching schemes.

4. Simulation study

4.1. Simulation model

We assume that there are 100 base stations forming a 10×10 mesh-like grid in the wireless network. We also assume the automatically configured VPG has the same structure in which all base station proxy servers have 4 neighbors, excepting those on the edge and the corner. The proxy servers on edges and corners of the 10×10 mesh have 3 and 2 neighbors respectively. We further assume the VPG structure will not change during the entire simulation.

To evaluate the impact of the cache line migration, we use a variant random waypoint movement pattern [26] to simulate the movement of the mobile hosts. The events of the mobile host movement are independent of the events of web requests. After a mobile host stays in the coverage area of one base station for some time, it may move to one of its neighbor base stations, or jump to a non-neighbor base station (this is the case that the user turns off the mobile device and turns on again after driving for a while), or simply choose to stay with the same base station. The mobile host has equal probability to make any of these moves. The time for a mobile host to stay with one base station follows the exponential distribution with an average duration of 180 seconds. This time is based on the assumption that the distance between two adjacent base stations is 2 miles, the average speed of vehicles is 40 miles/hour, and handoff happens at center point of two adjacent base stations. Similar assumptions are also used in [27,28].

We assume the user request latency is 100 ms ignoring the other overheads if the requested web document is cached in the proxy server that the mobile host is currently connected to. The round trip network distance between a pair of neighbor proxy servers is assumed to be 100 ms. So if a requested web document is found in a neighbor proxy server, the user request latency is assumed to be 200 ms, ignoring other overheads. If a requested web document is two hops away from the current proxy server, the user request latency will be 300 ms. We further assume that the average user request latency for a cache miss (the web document has to be fetched from the original web server) is 2000 ms, ignoring all other overheads. These assumptions are based on statistic data collected by running the benchmark Polygraph [29] on a web caching appliance developed by Swell Technology [30].

We also assume that there are 10,000 distinct web documents on the Internet and the average web document size is 60 K. We assume 35% of the web documents have size less than 10 K, 60% of them have size in the range from 10 KB to 100 KB, the sizes for the rest web documents are in the range of 100 KB to 1 MB. The assumption on web document sizes is based on the latest web access statistics obtained from several web servers [31,32]. It is assumed in [33] that, on average, when people surf the Internet they click a web link once every 12.5 seconds. Thus in our simulation, we assume the arrivals of web requests generated by a mobile host follow an exponential distribution and the average inter-arrival time equals to 12.5 seconds. As shown in previous studies [22,24], the user access frequencies of web documents follow a Zipf-like distribution. The user access frequency for a web document i can be calculated as:

$$f_i = \frac{1}{i^z \cdot \sum_{j=1}^{m} \frac{1}{j^z}} \tag{9}$$

where m is the number of the distinct web documents in the system, and $0 \leqslant z \leqslant 1$ is the Zipf factor [24]. A larger z value corresponds to a more skewed data access pattern, i.e., some documents are accessed considerably more frequently than others. When $z = 0$, the user access distribution is uniform, i.e., all

Table 1
Simulation Parameters

Parameter	Value
Proxy cache space	6000 KB
Number of distinct web pages	10,000
Total number of Web requests	400,000
Zipf Factor	0.75
Mean request interval time	12.5 seconds
Mean handoff interval time	180 seconds
Number of base stations	100
Number of mobile hosts	500
Average size of the web page	60
Average size of the query/reply message	0.1 KB
Average size of the cache line head	0.1 KB

documents are accessed at equal frequency. Total of 400,000 web requests are issued to the simulated cooperative proxy cache system.

To avoid the inaccurate statistics due to the simulation startup, we start to collect statistic data after the first 20,000 web requests being processed. A simple LRU cache policy is used for every tested proxy caching schemes in our simulation. We list most of our default simulation parameters in Table 1.

We use these default parameters throughout the simulation study unless explicitly stated otherwise. As in previous studies [10,11], we choose the cache hit ratio and the user request latency as the performance metrics. In addition, we also observe the query cost in terms of the average data exchange per web request between a pair of proxy servers. The data exchanged between proxy servers include messages and documents. In our P2P cooperative proxy caching scheme, moving and replicating network cache lines also contribute to the query cost.

4.2. Performance under various cache sizes

In this simulation, we study the performance of various proxy caching schemes under different cache spaces. We vary the cache size in each proxy server from 3000 KB to 18000 KB, and observe the cache hit ratios and average user request latencies under various cache sizes. We also monitor the data exchange between neighbor proxy servers and calculate the average data exchange per web request. The simulation results are depicted in Figs 4, 5 and 6.

As we expected, when the cache size at the individual proxy server increases, the cooperative proxy cache system can cache more web documents, resulting in higher cache hit ratios and lower average user request latency. Among the evaluated proxy caching schemes, the cache relocation scheme performs the worst due to its dedicated cache model. The proxy server cooperation improves the cache system performance tremendously as shown by the performance of both the MCC scheme and the PCC scheme, compared with the non-cooperative proxy caching approach. For instance, when the cache size is 9000 KB, the MCC and PCC schemes outperform the NC scheme by 175% and 235% respectively in terms of the cache hit ratio. In the meantime, the average user request latency using the NC scheme is 31% and 45% higher than that using the MCC and PCC schemes respectively. Among the cooperative proxy caching schemes, our P2P cooperative proxy caching schemes demonstrate better caching performance in term of the cache hit ratio and user request latency. For instance, when the cache size is 9000 KB, the cache hit ratio using the PCC scheme is 22% higher than that using the MCC scheme. On the other hand, the average user request latency using the MCC scheme is 11% higher than that using the PCC scheme.

Most importantly, compared to the MCC scheme, our PCC scheme requires much less network bandwidth within the wireless network while achieving better caching performance in terms of the cache

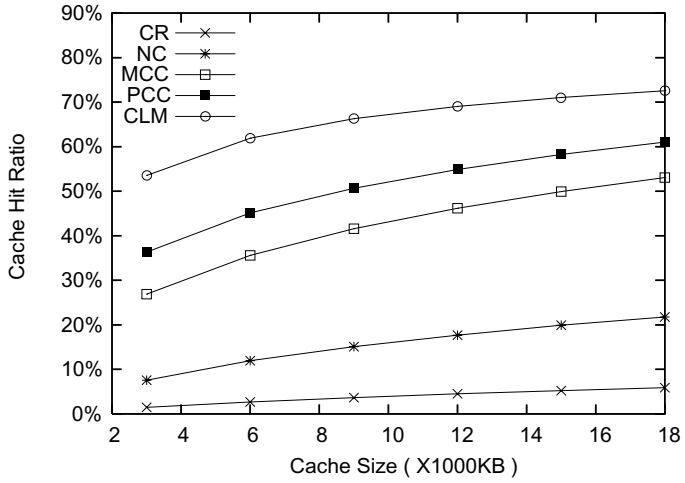

Fig. 4. Cache hit ratio under various cache sizes.

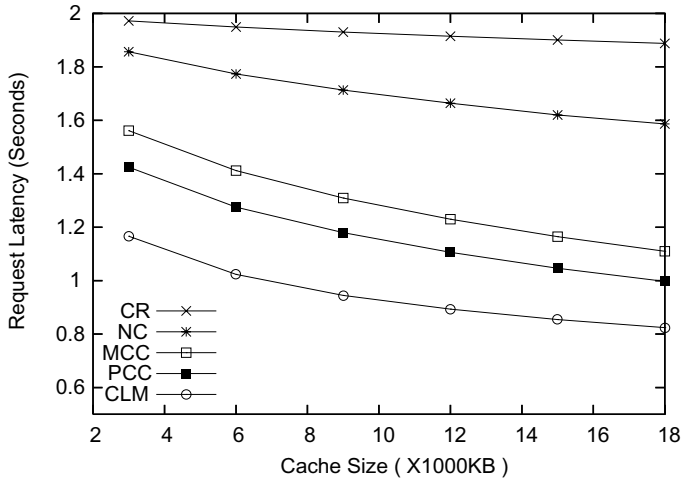

Fig. 5. Average request latency under various cache sizes.

hit ratio and average user request latency. As shown in Fig. 6, the data exchange between neighbor proxy servers using the MCC scheme is much higher than that using our PCC scheme. For instance, when cache size is 9000 KB, the average data exchange per web request using the MCC scheme is 167% higher than that using our PCC scheme. The superiority of our PCC scheme is due to its individual-based data search, data cache and data replication strategies, which can not only create links among nearby cached web replicas but also replicate the cached web documents based on demand and move them closer to interested clients. Because our P2P cooperative proxy caching scheme creates and updates the network cache line heads based on information in query responding messages, no extra network traffic is generated during the replication of network cache line heads.

Based our analysis, we can improve the performance of our P2P cooperative proxy caching scheme through cache line migration. As shown in Figs 4 and 5, our PCC-CLM scheme outperforms the other schemes by very large margins in terms of the cache hit ratio and average user request latency. For

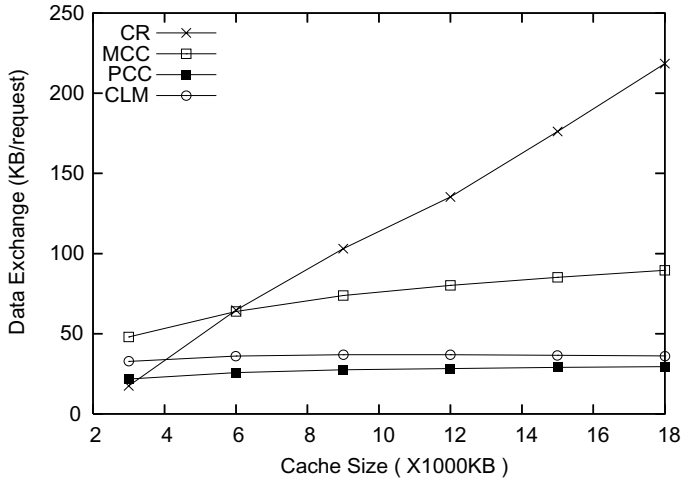

Fig. 6. Data exchange under various cache sizes.

instance, when the cache size is 9000 KB, the cache hit ratio using the PCC-CLM scheme is 59% better than that using the MCC scheme, and the average user request latency using the MCC scheme is 39% worse than that using the PCC-CLM scheme. Although moving cache line heads with the mobile hosts in the PCC-CLM scheme causes slightly more data exchange between neighbor proxy servers compared to the basic PCC scheme, this extra cost is worthwhile because the PCC scheme is only 22% better than the MCC scheme in terms of cache hit ratio while the PCC-CLM scheme is 59% better. On the other hand, the PCC-CLM scheme tremendously outperforms the MCC scheme in terms of data exchange between neighbor proxy servers. As shown in Fig. 6, when cache size is 9000KB, the MCC scheme generates 2 times of data traffic in wireless network compared to the PCC-CLM scheme. Cache relocation scheme shows the worst performance in terms of data exchange between proxy servers. We must note here that the non-cooperative cache approach does not need extra communication among proxy servers.

4.3. Impact of data access skew condition

Data access skew condition, i.e, the locality of user interests, determines the probability of new web requests referring to previously accessed web documents. As discussed before, user access frequencies to web documents follow Zipf-like distribution with the Zipf factor varying between 0.64 and 0.83. In this simulation, we study how these evaluated proxy caching schemes perform under various Zipf factors by varying the Zipf factor from 0.4 to 0.9. The simulation results are reported in Figs 7, 8 and 9.

When the Zipf factor increases, the cache hit ratio should increase because more web requests concentrate on fewer web documents. As shown in Fig. 7, cache hit ratios increase for all proxy caching schemes when the Zipf factor increases. Once again our PCC-CLM scheme significantly outperforms other proxy caching schemes in terms of the cache hit ratio. For instance, when the Zipf factor equals to 0.7, the cache hit ratio using the PCC-CLM scheme is 73% higher than that using the MCC scheme, and 465% higher than that using the non-cooperative approach (NC). Usually, retrieving web documents from the original web servers results in much higher user request latencies than fetching the cached web documents in the P2P cooperative proxy cache system. When the Zipf factor increases, the average user request latencies decrease for all proxy caching schemes because more web requests are satisfied by the proxy cache system. Once again, our P2P cooperative proxy caching scheme and its cache line migration

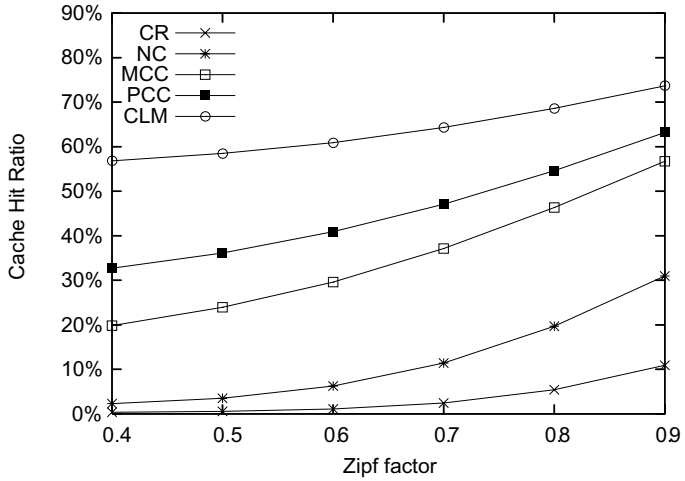

Fig. 7. Cache hit ratio under different Zipf factors.

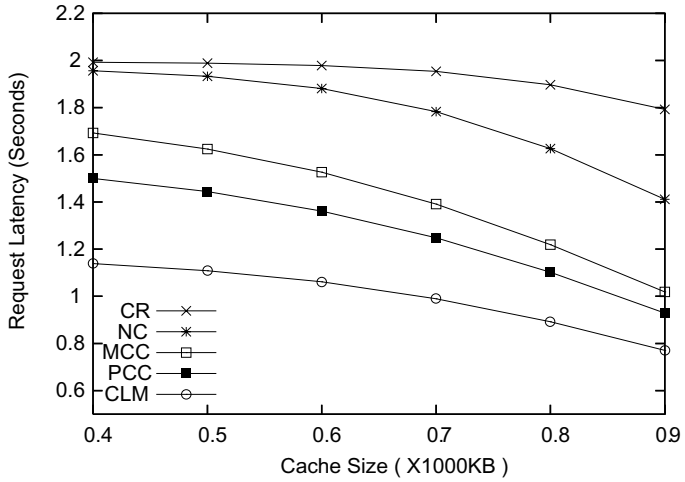

Fig. 8. Average request latency under various Zipf factors.

enhancement outperform the other schemes in large margins. When Zipf factor is 0.7, the average user request latency using MCC scheme is 41% higher than that of PCC-CLM scheme. In the meantime, the average user request latency using NC scheme is 2 times of that using the PCC-CLM scheme.

In terms of communication overhead, our basic P2P cooperative proxy caching scheme (PCC) generates less data exchange between proxy servers than other cooperative proxy caching schemes do. Although our PCC-CLM scheme generates slightly more data exchange between neighbor proxy servers than the basic PCC scheme does, its outstanding performance in terms of the cache hit ratio and user request latency makes the extra cost worthwhile. The MCC scheme generates too much extra traffic in wireless network that may affect the performance of the network itself. Once again, the cache relocation (CR) scheme performs the worst in terms of all performance metrics.

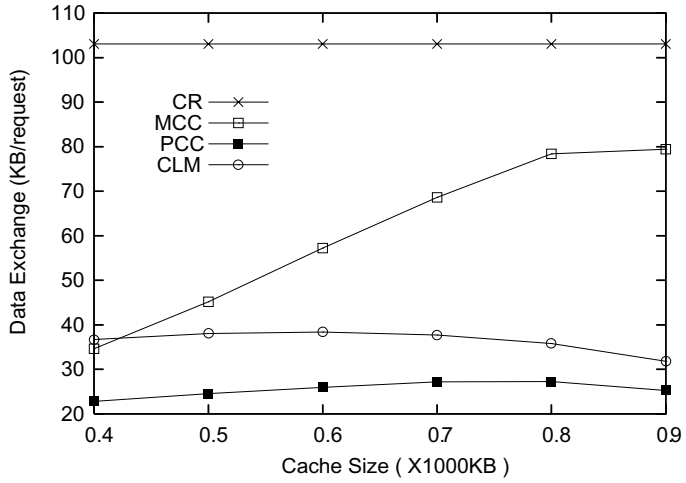

Fig. 9. Data exchange under different Zipf factors.

5. Concluding remarks and future studies

In this paper, we propose a novel P2P cooperative proxy caching scheme for wireless base stations. We design a mobile cache model to facilitate the cooperative proxy caching. Based on this model, we use a self-configured virtual proxy graph (VPG), which is independent of the underlying network structure and adaptive to the network and geographic environment changes, to achieve efficient data search, data cache and date replication. The aggregate effect of data caching, searching and replicating actions by individual proxy servers automatically distributes cached web documents closer to the interested mobile clients. Furthermore we propose a cache line migration strategy which replicates the heads of network cache lines associated with a moving mobile host to the new base station during the mobile host handoff to enhance the P2P cooperative proxy caching scheme. These replicated cache line heads provide direct links to the cached web documents previously accessed by the moving mobile host in neighbor proxy servers, thus, improving the mobile web caching performance. Our performance analysis and simulation study demonstrate that our proposed P2P cooperative proxy caching schemes outperform other proxy caching schemes in terms of different performance metrics.

In our simulation studies, we used a simple LRU algorithm for the cache replacement. Some other factors, such as document size, date replication rate, and mobile host handoff also affect the proxy cache space management. We are currently studying the collective impact of these factors to the cache replacement decision so that we can design a cache replacement algorithm specifically for our P2P cooperative proxy cache system. To further improve the system performance, we are also investigating the possibility of integrating the prediction of the mobile host movement with our proposed P2P cooperative proxy caching scheme.

References

[1] J. Wang, A survey of web caching schemes for the internet, *ACM SIGCOMM Computer Communication Review archive* **29**(5) (1999), 36–46.

[2] A. Chankhunthod, P.B. Danzig, C. Neerdaels, M.F. Schwartz and K.J. Worrell, A hierarchical internet object cache, In USENIX Annual Technical Conference, 1995.

[3] D. Povey and J. Harrison, *A distributed internet cache*, In Proceedings of the 20th Australasian Computer Science Conference, 1997.
[4] L. Zhang, S. Michel, S. Floyd, V. Jacobson, K. Nguyen and A. Rosenstein, in: *Adaptive web caching: Towards a new global caching architecture*, In Proceedings of the Third International Caching Workshop, 1998.
[5] P.S. Yu and E.A. MacNair, Performance study of a collaborative method for hierarchical caching in proxy servers, *Computer Networks and ISDN System* (1998), 215–224.
[6] P. Rodriguez, C. Spanner and E.W. Biersack, *Web caching architectures: Hierarchical and distributed caching*, In Proceedings of the 4th International Web Caching Workshop, 1999.
[7] J. Jayaputera and D. Taniar, Invalidation for corba caching in wireless devices, *Embedded and Ubiquitous Computing, Lecture Notes in Computer Science* **3207** (2004), 460–471.
[8] Z. Jiang, L.F. Chang, B.J. Kim and K.K. Leung, Incorporating proxy services into wide area cellular ip networks, *In Proceedings of the IEEE WCNC* (2000).
[9] J. Wang, Z. Du and P. Srimani, *Network cache model for wireless proxy caching*, in EEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS), 2005.
[10] A. Luotonen and K. Altis, World wide web proxies. Special Issue of Computer Networks and ISDN systems, *First International Conference on WWW* **27**(2) (1994), 147–154.
[11] D. Wessels and K. Claffy, Icp and the squid, *IEEE Journal on Selected Areas in Communication* **16**(3) (1998), 345–357.
[12] D. Katsaros and Y. Manolopoulos, *Caching in web memory hierarchies*, In Proceedings of the 2004 ACM symposium on Applied computing table of contents, 2004, 1109–1113.
[13] R. Tewari, M. Dahlin, H. Vin and J. Kay, Beyond hierarchies: Design consideration for distributed caching on the internet, In Technical Report: TR98-04, Department of Computer Science, University of Texas at Austin, 1998.
[14] J. Touch, *The lsam proxy cache ?a multicast distributed virtual cache*, In the 3rd International WWW Caching Workshop, 1998.
[15] J. Almeida, L. Fan, P. Cao and A. Broder, Summary cache: A scalable widearea web cache sharing protocol, *IEEE/ACM Transactions on Networking* **8** (2000), 281–293.
[16] A. Rousskov and D. Wessels, Cache digests, *Computer Networks and ISDN Systems* **30** (1998), 2155–2168.
[17] J. Xu, Q. Hu, W.-C. Lee and D.L. Lee, Performance evaluation of an optimal cache replacement policy for wireless data dissemination, *IEEE Transactions on Knowledge and Data Engineering* **16**(1) (2004), 125–139.
[18] D. Katsaros and Y. Manolopoulos, Web caching in broadcast mobile wireless environments, *IEEE Internet Computing* **8**(3) (2004), 37–45.
[19] J. Yang, W. Wang, R. Muntz and J. Wang, *Access driven web caching*, Technical report, UCLA, 990007.
[20] M. Rabinovich, J. Chase and S. Gadde, Not all hits are created equal: Cooperative proxy cache over a wide-area network, *Computer Networks and ISDN Systems* **30**(22–23) (1998), 2253–2259.
[21] A. Wolman, G.M. Voelker, N. Sharma, N. Cardwell, A.R. Karlin and H.M. Levy, On the scale and performance of cooperative web proxy caching, *In Symposium on Operating Systems Principles* (1999), 16–31.
[22] J.Z. Wang and R.K. Guha, Proxy ecology – cooperative proxies with artificial life, *Web Intelligence and Agent Systems: An International Journal* **2**(3) (2004).
[23] Stathes Hadjiefthymiades and Lazaros Merakos. Using proxy cache relocation to accelerate web browsing in wireless/mobile communications. In WWW10, May 2001.
[24] L. Breslau, P. Cao, L. Fan, G. Phillips and S. Shenker, Web caching and zipf-like distributions: Evidence and implications. In INFOCOM1999, 1999, 126–134.
[25] K.Y. Lai, Z. Tari and P. Bertok, Supporting user mobility through cache relocation, *Mobile Information Systems* **1**(4) (2005), 275–307.
[26] W. Su, S.-J. Lee and M. Gerla, Mobility prediction in wireless networks, *In Proceedings of the IEEE Military Communications Conference* (*MILCOM*) (2000).
[27] S.-J. Lee, Routing and Multicasting Strategies in Wireless Mobile Ad hoc Networks. PhD thesis, University of California, Los Angeles, 2000.
[28] E.M. Royer, Routing in Ad hoc Mobile Networks: On-Demand and Hierarchical Strategies. PhD thesis, University of California, Santa Barbara, 2000.
[29] The measurement factory. http://www.measurement-factory.com/, 2006.
[30] Tsunami. http://www.swelltech.com/products/caching.html, 2006.
[31] Web server statistics. http://www.lescroupiersrunningclub.org.uk/ ace/logfile.php, 2006.
[32] Web server statistics for vsohp. http://satobs.org/ps.html, 2006.
[33] Microsoft internet security & acceleration server: Isa server scales out to meet enterprise-class caching demands. http://www.microsoft.com/isaserver/techinfo/deployment/ScaleOutCachingwithISAServer.doc, 2001.

James Z. Wang received the B.S. and M.S. degrees in Computer Science from University of Science and Technology of China, and the Ph.D. degree in Computer Science from University of Central Florida. He is currently an assistant professor in Department of Computer Science at Clemson University. He previously worked as senior software engineer in Veritas Corp. and Computer Associate. His research interests include multimedia systems, database, distributed computing, Internet technologies, data mining and bioinformatics. He has published more than 30 papers in refereed journals and conference proceedings. He is a member of IEEE and ACM. He served as program committee member or session chair in several international conferences.

Zhidian Du received his B.S. degree in Computer Science from Hebei Science and Technology University. He received his M.S. degree in Computer Science from New Mexico State University. He is a PhD student in Department of Computer Science at Clemson University. He is a student member of IEEE.

Pradip K. Srimani is a professor and chair of computer science at Clemson University. He has previously served the faculty of the India Statistical Institute, Calcutta, Gesselschaft fuer Mathematik und Datenverarbeitung, Bonn, West Germany, the Indian Institute of Management, Calcutta, India, and Southern Illinois University, Carbondale, Illinois, Colorado State University, Ft. Collins, Colorado, and the Technical University of Compiegne, France. He was the editor-in-chief of the IEEE Computer Society Press during 1996-2000 and is currently an associate editor of IEEE Transaction on Data and Knowledge Engineering and Journal of Parallel Computing. His research interests include mobile computing, distributed computing, parallel algorithms, networks, and graph theory applications. He has published more than 100 papers in archival journals and more than 100 papers in conference proceedings. He is a coeditor of two books on software reliability and distributed mutual exclusion algorithms by IEEE CS Press. He has guest edited special issues for IEEE Computer, IEEE Software, VLSI Design, Journal of Systems & Software, and Journal of Computer & Software Engineering, IEEE Transactions on Software Engineering, Parallel Computing, International Journal of Systems Science, IEEE Transactions on Computers, MONET, WINET, and others. He is a member of the ACM/IEEECS Steering Committee on Curricula 2001 for computer science and computer engineering. He is a fellow of the IEEE and a member of the ACM.