Cache Signatures for Peer-to-Peer Cooperative Caching in Mobile Environments*

Chi-Yin Chow Hong Va Leong Alvin Chan

Department of Computing, Hong Kong Polytechnic University, Hong Kong

E-mail: {*cscychow, cshleong, cstschan*}@*comp.polyu.edu.hk*

Abstract

Caching is a key technique for improving data retrieval performance of mobile clients in mobile environments. The emergence of robust and reliable peer-to-peer (P2P) technologies now brings to reality what we call "cooperative caching" in which mobile clients can access data items from the cache in their neighboring peers. This paper considers a ***CO****operative* ***CA****ching scheme for mobile systems, called COCA. A cache signature scheme is devised for COCA that provides hints for the mobile clients to determine whether a required data item is cached by their neighboring peers based on their local state. The trade-off between the improvement in system performance and the overheads of the cache signature scheme in COCA is discussed. The performance of COCA with and without the cache signature scheme is evaluated through a number of simulated experiments. COCA is shown to be capable of effectively reducing the number of server requests and power consumption, as well as shortening the access latency as the number of neighboring peers increases. The inclusion of cache signature scheme further improves on the access latency.*

1. Introduction

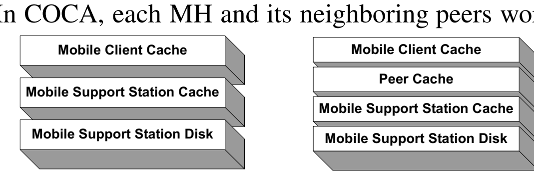
With the recent widespread deployment of new peer-to-peer (P2P) communication technologies, such as IEEE 802.11 and Bluetooth, there is a new alternative for information sharing among clients over the standard client/server model in a mobile environment. Coupled with the accelerated computation power and improved storage capacity of mobile devices, mobile clients can now directly communicate among themselves to share cached information rather than having to rely on their connection to the server for each request. This new information sharing alternative is known as *mobile cooperative caching*.

In the past, cooperative caching schemes were extensively studied in wired networks [3, 4, 10, 12]. Recently, cooperative caching schemes in mobile environments have been drawing increasing attention. In [6, 7, 11], cooperative caching schemes were proposed in mobile ad hoc networks (MANETs). Our work is different from previous works in that we propose a cooperative caching scheme in the context of conventional mobile environments, in the *presence* of mobile support stations (MSSs) and is based on single-hop communication within their servicing cells.

Signature techniques have been applied on information filtering in mobile broadcast environments [8] and on efficiently searching P2P networks [9]. In this paper, we propose a *cache signature* scheme, that is applied to cooperative caching. It provides hints for the mobile hosts (MHs) on the cache contents in their neighboring peers. Importantly, the hints allow an MH to effectively determine whether a required data item is residing in the cache of its neighboring peers. Thus, the MH can make a *local* decision on whether to search for the required data item from its neighboring peers or directly requesting the data item from the MSS.

In this paper, a **CO**operative **CA**ching (COCA) scheme is proposed for mobile systems. In conventional mobile systems, the storage hierarchy consists of three layers: Mobile Client Cache, MSS Cache and MSS Disk, as depicted in Figure 1(a). If an MH cannot find the required data item in its cache (called a *cache miss*), it sends a request to the MSS. The MSS grabs the required data item from its disk if the item does not reside in the MSS cache. In COCA, a new logical layer is inserted between the Mobile Client Cache layer and the MSS Cache layer. This layer is called the *Peer Cache* layer, as depicted in Figure 1(b). In COCA, when an MH suffers from a cache miss (called a *local cache miss*), it looks up the required data item from its neighboring peers' cache before enlisting the MSS for help. Only when it cannot find the data item from its peers' cache (called a *global cache miss*) will it request the data item from the MSS, as exercised in the conventional system.

In COCA, each MH and its neighboring peers work to-



(a) Conventional (b) COCA

Figure 1. Storage hierarchy of mobile systems.

*Research supported in part by the Research Grant Council and the Hong Kong Polytechnic University under grant numbers PolyU 5084/01E and H-ZJ86.

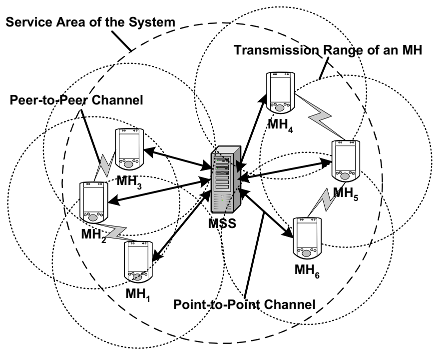


Figure 2. System architecture of COCA.

gether as a *dynamic group* to share their cached data items cooperatively via the established P2P channels, as illustrated in Figure 2. All MHs residing within the transmission range of an MH are enlisted as members of that MH's group. Meanwhile, an MH can belong to more than one dynamic group. If MH_2 finds the required data item from its local cache, it constitutes to a *local cache hit*. When MH_2 encounters a local cache miss, it attempts to request the required data item from its members in its dynamic group (MH_1 and MH_3), before requesting the data item from the MSS. If its peers can turn in the data item, a *global cache hit* is resulted; otherwise, MH_2 has to obtain the data item from the MSS.

COCA is appropriate for systems in which a group of MHs possesses similar access patterns, with individual hot spots. For example, in an exhibition, the common access pattern is defined by the information provided by the booths or the organizers. The MHs are likely to access general information, as well as information pertaining to their own interest more frequently. Conversely, they are likely to access information about other booths with lower frequency. In short, when the MHs share a common access pattern, there exist higher probabilities for them to obtain the required data items from their neighboring peers.

The rest of this paper is organized as follows. Section 2 describes the COCA model. Section 3 delineates the cache signature scheme. Section 4 defines the simulation model and studies the performance of standard COCA and COCA with the cache signature scheme through a number of simulated experiments. Finally, Section 5 offers brief concluding remarks.

2. COCA Model

In COCA, we assume that each MH has its own cache space with the same capacity ($ClientCacheSize$). The MHs access data items stored in the MSS database, and each has its own hot spot. COCA is based on the system architecture as depicted in Figure 2. For simplicity, we further assume that there is no update to data items. This assumption will be relaxed in our next piece of work. The communication protocol, power control and power consumption measurement for COCA are next described.

2.1. Communication Protocol

For each request, an MH first locates for the required data item from its local cache. If it cannot find the data item, it broadcasts a *request* message to its neighboring peers. The peers that cache the required data item reply to the MH with a *reply* message via the P2P communication. When the MH receives replies from some of its neighboring peers, it selects one of them to return the required data item by considering the distance between them. The MH chooses the peer with the shortest distance, and the transmission power is adjusted correspondingly to the minimal required level to communicate with the target peer. It then sends a *retrieve* request to the target peer via the P2P channel. Finally, the target peer receiving the *retrieve* request sends the required data item to the MH.

In case of more than one neighboring peer caching the required data items and the minimum transmission power is the same, the one with the highest power capacity is selected. If no neighboring peer caches the required data item, the MH has to request it from the MSS.

2.2. Power Control

All MHs are assumed to possess the same power capacity and adopt the same wireless network interface. Each MH is equipped with an omnidirectional antenna so that all MHs within the transmission range of a transmitting MH can receive its transmission. Furthermore, when an MH communicates with other MHs, it is able to adjust the transmission range by controlling the transmission power to the minimum range that can cover the target peers [1, 13].

For the power control mechanism, the transmission power is divided into several discrete levels. Each discrete level is associated with a predefined equi-width transmission range. The minimum and maximum power levels produce the shortest transmission range R_{min} and the longest transmission range R_{max} respectively. The distance of each equi-width transmission range is R_{range}. When an MH (MH_i) establishes a P2P connection to another MH (MH_j), the distance between them is calculated by using an Euclidean distance equation $D_{ij} = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2}$, where ($x_i$,$y_i$) and ($x_j$,$y_j$) are the locations of MH_i and MH_j respectively. The required minimum transmission range can be calculated by using the equation:

$$R_{ij} = \begin{cases} R_{min}, & \text{for } D_{ij} \leq R_{min} \\ R_{min} + \tau \times R_{range}, & \text{for } R_{min} < D_{ij} \leq R_{max} \end{cases} \quad (1)$$

where $\tau = \lceil (D_{ij} - R_{min})/R_{range} \rceil$.

2.3. Power Consumption Measurement

Both the MH and the MSS are assumed to operate in an ad hoc mode, and the non-destination MH maintains the wireless network interface in an idle mode during the data transmission. The calculation of power consumption of the MH is based on [5] which uses linear formulas to measure the power consumption of the source MH, S, the destination MH, D and other *remaining* MHs residing in the transmission range of the source MH, S_R and the destination MH, D_R, as depicted in Figure 3(a). The P2P power consumption is measured by Equation 2.

$$P_{p2p} = \begin{cases} v_{send} \times |m| + f_{send}, & \text{for } MH = S \\ v_{recv} \times |m| + f_{recv}, & \text{for } MH = D \\ v_{sd\_disc} \times |m| + f_{sd\_disc}, & \text{for } MH \in S_R \wedge MH \in D_R \\ v_{s\_disc} \times |m| + f_{s\_disc}, & \text{for } MH \in S_R \wedge MH \notin D_R \\ v_{d\_disc} \times |m| + f_{d\_disc}, & \text{for } MH \notin S_R \wedge MH \in D_R \end{cases} \tag{2}$$

where f is the fixed power consumption for sending, receiving and discarding a message, and v is the variable power consumption based on the size of a message m in byte ($|m|$).

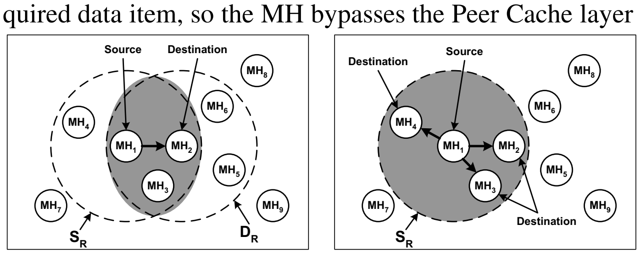
Power consumption of source MH, S, and other MHs residing in S_R of a broadcast communication as depicted in Figure 3(b), can be measured by Equation 3.

$$P_{bc} = \begin{cases} v_{bsend} \times |m| + f_{bsend}, & \text{for } MH = S \\ v_{brecv} \times |m| + f_{brecv}, & \text{for } MH \in S_R \end{cases} \tag{3}$$

3. Cache Signature Scheme

A *cache signature* is a bit string that summarizes the content of a local cache in an MH. All cache signatures have the same length λ. The cache signature is generated by superimposing all *data signatures* of the data items in the local cache. The data signature for a data item is generated by a double-hashed mechanism, as illustrated in Figure 4, where an MH caches five data items and $\lambda = 10$. Each data item is first hashed by a hash function **HASH()**. The hash value of the first data item (ID=18) is 0000110000. A modulus function **MOD()** is used to yield the hash value within the bound defined by λ. As the result, we get a hash value which is between 0 and $\lambda - 1$. The data signature 0100000000 is produced by setting the bit at the position indicated by the hash value. After this procedure is repeated for each data item in the local cache, the cache signature can be obtained by superimposing all data signatures.

When an MH encounters a local cache miss, it generates a data signature for the required data item, called *search signature*. It also computes a *peer signature* by superimposing the peers' cache signatures maintained in the local cache. The data signature is then compared with the peer signature by performing a bitwise AND operation on them. If the result is zero, it indicates that no peer caches the required data item, so the MH bypasses the Peer Cache layer



(a) P2P communication (b) Broadcast communication

Figure 3. Power consumption measurement.

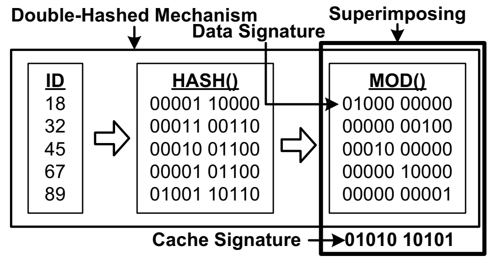


Figure 4. Cache signature generation.

and requests the data item from the MSS. If the result is the same as the search signature, the neighboring peers are likely to cache the data item, so the MH broadcasts the request to them. When the MH can obtain the data item from some of its peers, it is called a *true positive*; otherwise, if no peer caches the required data item, it is called a *false positive*. False positive can arise under three situations. First, there is a delay upon updating a cache signature. Second, there is a delay upon detecting the *departure* of a neighboring peer. Third, two or more data items produce the same data signature.

The MHs exchange their cache signatures with their neighboring peers actively or passively. With *passive exchange*, the MHs piggyback their cache signatures onto the *request* and *reply* messages. With *active exchange*, MHs that have not generated any request to their neighboring peers for a period of time μ are required to broadcast their cache signatures to their neighboring peers.

An MH removes a cache signature from its local cache when it has not received the cache signature from the corresponding MH for a period of time μ. Since each neighboring peer is expected to broadcast its cache signature periodically, the departure of a neighboring peer can be detected.

There are two sources of overheads for the cache signature scheme in COCA. The cache signature scheme suffers from *false negative*, that can be caused by two situations. First, there is a delay upon updating a peer's cache signature, so an MH cannot find a match in the peer signature. However, some peers do indeed cache the required data item. Second, the delay on detecting the *join* of a neighboring peer, in which that peer is the *only* peer caching the required data item. There are also computation overheads of generating cache signatures and communication overheads of exchanging them among peers. Since the computation overheads are less significant and do not affect other peers, only communication overheads are considered in power consumption measurement.

4. Simulation Model

In this section, we present the simulation model used to evaluate the performance of COCA with and without the cache signature scheme. The simulation model is implemented in C++ using CSIM. The simulated mobile environment is composed of a single MSS and 100 MHs. The MHs can move freely in a 500 m $\times$ 500 m space ($Area$) where

Table 1. Simulation default parameters.

Parameters	Values
$NumData$	1000 items
$ClientCacheSize$	10 % of *NumData*
$Speed\,(v_{min} \sim v_{max})$	$1 \sim 5$ m/s
λ	500 bits
μ	5.0 s
θ	0.5

is also the service area provided by the MSS. The database in the MSS contains $NumData$ equal-sized (1 KB) data items. The MSS provides a bandwidth 10 Mbit/s for the MHs; there is also a P2P channel with a bandwidth 1 Mbit/s, through which an MH communicates with its neighboring peers. The MHs' devices adopt the same wireless network interface, with a transmission range of 50 meters.

Client Model. The time interval between two consecutive requests generated by the MHs follows an exponential distribution with a mean of one second. The MHs access the data items stored in the MSS database and each of them has its own hot spot. The position of the hot spot of an individual MH is uniformly distributed. Each MH generates accesses to data items following a Zipf distribution with a skewness parameter θ. In our simulation, the MHs move according to the "random waypoint" model [2]. In the beginning, the MHs are randomly distributed in $Area$. Each MH randomly chooses its own destination with a randomly determined speed s from a uniform distribution $\mathcal{U}(v_{min}, v_{max})$. It then travels with constant speed s. When it reaches the destination, it comes to a standstill for one second to determine its next destination. It then moves towards its new destination with another random speed $s' \sim \mathcal{U}(v_{min}, v_{max})$.

Server Model. There is a single MSS in the system. The MSS receives and processes requests from the MHs with a FCFS service policy. When the MSS is busy, an infinite queue is used to buffer the requests from the MHs. The MSS has a cache capacity of 50 percent of $NumData$, and it adopts LRU as the cache replacement policy. Each cache miss in the MSS incurs disk I/O access time (10 ms) for retrieving the required data item from the disk.

Table 1 shows the default parameter settings used in the simulated experiments. Table 2 and Table 3 show the parameter settings for MHs playing different roles, as used in power consumption measurement for P2P and broadcast communication respectively [5].

4.1. Simulation Experiments

In our simulation, we compare the performance of COCA schemes with a conventional caching approach that does not involve cooperation among peers. This serves as a base case for comparison. LRU replacement strategy is applied on the base case (LRU) and standard COCA (COCA). Likewise, we evaluate the performance of COCA with the cache signature scheme (COCA-SIG) by comparing it with standard COCA. All simulation results are recorded after the system reaches a stable state. Each MH generates 20,000 queries, where 2,000 are warm-up queries in order to avoid a transient effect. We conduct the experiments by varying the parameters: cache size, transmission range and signature length. The performance metrics in the experiments include access latency, server request ratio, power consumption, false positive ratio and false negative ratio.

In order to obtain fair and consistent comparison with other caching strategies, the cache signature is stored in the local cache and consumes a certain amount of storage.

Table 2. Power consumption measurement - P2P.

Condition	$\mu W \cdot s/byte$	$\mu W \cdot s$
$MH = S$	$v_{send} = 1.9$	$f_{send} = 454$
$MH = D$	$v_{recv} = 0.5$	$f_{recv} = 356$
$MH \in S_R \wedge MH \in D_R$	$v_{sd\_disc} = 0$	$f_{sd\_disc} = 70$
$MH \in S_R \wedge MH \notin D_R$	$v_{s\_disc} = 0$	$f_{s\_disc} = 24$
$MH \notin S_R \wedge MH \in D_R$	$v_{d\_disc} = 0$	$f_{d\_disc} = 56$

Table 3. Power consumption measurement - broadcast.

Condition	$\mu W \cdot s/byte$	$\mu W \cdot s$
$MH = S$	$v_{bsend} = 1.9$	$f_{bsend} = 266$
$MH \in S_R$	$v_{brecv} = 0.5$	$f_{brecv} = 56$

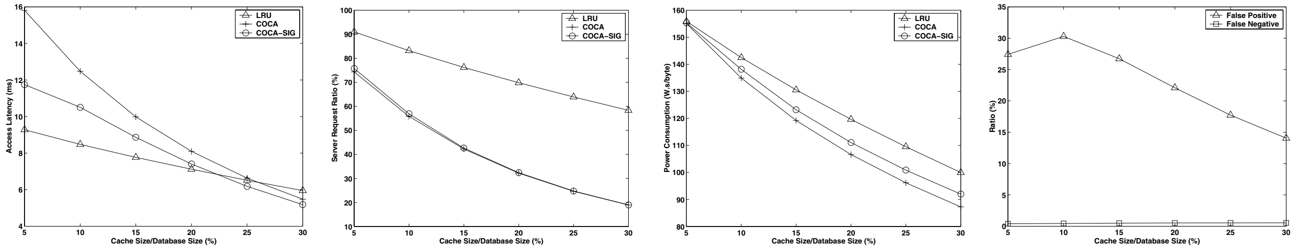
4.2. Effects of Cache Size

Our first experiment studies the effect of cache size on the system performance by varying the ratio of cache size to database size from 5 percent to 30 percent.

Figure 5(a), Figure 5(b) and Figure 5(c) show that all schemes exhibit better system performance with increasing cache size. This is because more required data items can be found in the local cache as the cache size gets larger. When the cache size is larger than 25 percent of the database size, COCA-SIG is the best performer in terms of access latency.

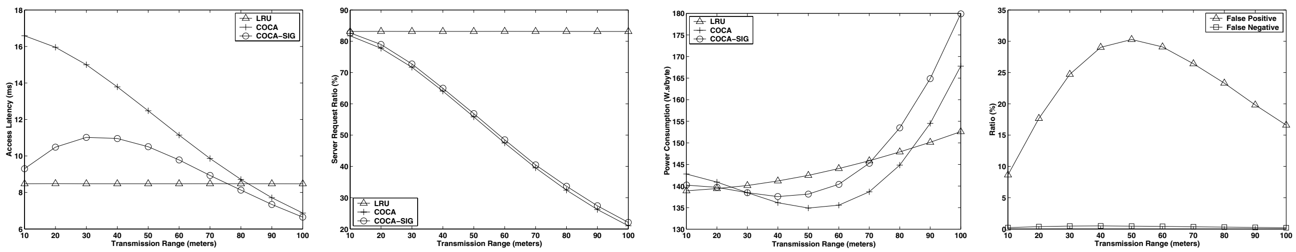
In Figure 5(b) and Figure 5(c), COCA schemes are found to consistently perform better than LRU. COCA-SIG incurs slightly higher overheads than COCA because false negatives generate more server requests and the exchanges of cache signatures consume more battery power.

Figure 5(d) shows the effect of cache size on the false positive and false negative ratios. The false positive ratio peaks and then drops as the cache size increases. On the other hand, the false negative ratio slightly increases as the cache size gets larger. Initially, the increase in cached data items can increase the chance of data items being mapped to the same signature, leading to a slight increase in false positive. As the cache size increases further, the time that a data item resides in the local cache becomes longer, and more neighboring peers could cache the required data item, the chance of encountering a false positive is correspondingly reduced. As an MH caches more data items, the probability of obtaining the required data item from a neighboring peer increases. The impact due to the delay on detecting a newly admitted neighboring peer also increases, leading to an increase in false negative.



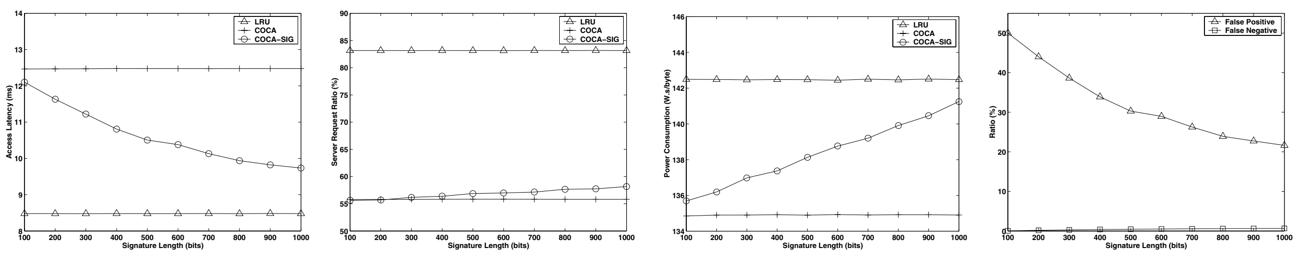
(a) Access Latency (b) Server Request Ratio (c) Power Consumption (d) False Positive & False Negative

Figure 5. Performance studies on various cache size.



(a) Access Latency (b) Server Request Ratio (c) Power Consumption (d) False Positive & False Negative

Figure 6. Performance studies on various transmission ranges.



(a) Access Latency (b) Server Request Ratio (c) Power Consumption (d) False Positive & False Negative

Figure 7. Performance studies on various signature length.

4.3. Effects of Transmission Range

In this series of simulated experiments, we examine the influence of system performance on various transmission ranges, from 10 to 100 meters.

Figure 6(a) and Figure 6(b) indicate that the performance of COCA schemes improves as the transmission range increases. Since there is no collaboration among peers in LRU, its performance is not affected by varying the transmission range. In Figure 6(a), it can be observed that COCA-SIG peaks at a transmission range of 30 to 40 meters and then drops. For COCA-SIG, the initial increase is due to the problem of false positive and false negative (Figure 6(d)), both impacting the performance.

Similar to the study on the cache size, the overheads of COCA-SIG are generally higher than those of COCA, as depicted in Figure 6(b) and Figure 6(c). This is also due to the effect of false negative and more power consumption on the exchange of cache signatures. In Figure 6(c), the power consumption of LRU increases with a larger transmission range. The larger the transmission range the more neighboring peers an MH can enlist, thus the more peers are affected when the MH communicates with the MSS. The power consumption of COCA and COCA-SIG also increases after they reach the transmission range of 50 meters and 40 meters respectively. COCA and COCA-SIG reduce the power consumption as the transmission range increases initially. This is because more neighboring peers can return the required data item as the transmission range increases. Thus, the MHs can save the power required to communicate with the MSS. This is also due to the fact that with a relatively small transmission range, only a low power level is needed and the number of MHs affected by the MH broadcasting message having to discard unintended messages is still limited. The power saving due to more MHs holding the required data item is more dominant. However, as the transmission range further increases, the power consumption of COCA and COCA-SIG gets larger. The MH has to dissipate more battery power on receiving the message broadcast and discarding unintended message sent by its neighboring peers, and this growth offsets the savings with more neighboring peers contributing to the required data item.

Figure 6(d) shows that the false positive and false negative ratios peak at a transmission range of 50 meters and 40 meters respectively, and then they drop, as the transmission range gets larger. Since there is a delay upon detecting a leaving neighboring peer, the accuracy of a peer signa-

Proceedings of the 18th International Conference on Advanced Information Networking and Application (AINA'04)
0-7695-2051-0/04 $ 20.00 © 2004 IEEE

ture decreases as there are more neighboring peers for an MH. After the transmission range reaches 50 meters, the false positive ratio drops. This is because the probability of more than one neighboring peer caching the required data item increases, as an MH has more neighboring peers. The number of departures of neighboring peers also drops with a larger transmission range. Thus, the problem of late detection on the departure of neighboring peers is alleviated. The transmission range has a very slight effect on the false negative ratio. The false negative ratio increases when the transmission range increases, due to a decrease in the accuracy of a peer signature because of the delay upon detecting a newly admitted peer. When the transmission range becomes larger, the false negative ratio drops because more neighboring peers cache the required data item and the number of joins of neighboring peers decreases, alleviating the effect of the late detection of newly admitted peers on the false negative ratio.

4.4. Effects of Signature Length

In our final experiment, we study the effect of the cache signature length on the system performance of COCA-SIG by varying the signature length from 100 bits to 1000 bits.

Figure 7(a) indicates that the access latency decreases as the signature length gets longer. This is because the MHs can more precisely bypass the Peer Cache layer with increasing signature length. They do not have to waste the timeout period on searching through the peers' cache, when the peer signature indicates that the required data item is not in the cache of their neighboring peers.

As shown in Figure 7(b), the number of server requests increases with a longer cache signature, since there is a slight increase in the false negative ratio (Figure 7(d)), which induces a small increase in number of server requests. In Figure 7(c), it shows that the power consumption increases with the increasing signature length because the MHs have to consume more power to exchange their cache signatures.

Figure 7(d) depicts that the false positive ratio decreases as the signature length gets longer; however, the false negative ratio increases slightly as the signature length increases. As the signature length increases, the cache signature gives more accurate information. Thus, the probability of more than one data item being represented by the same bit in a cache signature decreases. The MHs can more precisely bypass the Peer Cache layer with a lower chance of encountering false positive. However, when a cache signature represents more precise hints, the delay in updating the cache signature increases the probability of not recognizing the fact that some peers already cache the required data item.

5. Conclusion

In this paper, we have described a cooperative caching scheme, called COCA, and a cache signature scheme adapted to COCA for mobile environments. In COCA, the MHs share their cache with one another to reduce the number of server requests and power consumption. The cache signature scheme provides hints for the MHs to determine whether they should bypass the Peer Cache layer based on their local state. The performance of standard COCA and COCA with the cache signature scheme is evaluated through a number of simulated experiments, which show that they reduce the number of server requests and power consumption, compared with LRU; however, in general, they incur longer access latency when the MHs encounter a global cache miss. The cache signature scheme can effectively reduce the access latency with little extra overheads.

References

[1] S. Agarwal, R. H. Katz, S. V. Krishnamurthy, and S. K. Dao. Distributed power control in ad-hoc wireless networks. In *Proc. of the 12th IEEE PIMRC*, pages 59–66, September 2001.

[2] J. Broch, D. A. Maltz, D. B. Johnson, Y.-C. Hu, and J. Jetcheva. A performance comparison of multi-hop wireless ad hoc network routing protocols. In *Proc. of the 4th MobiCom*, pages 85–97, October 1998.

[3] M. Dahlin, R. Y. Wang, T. E. Anderson, and D. A. Patterson. Cooperative caching: Using remote client memory to improve file system performance. In *Proc. of the 1st USENIX OSDI*, pages 267–280, November 1994.

[4] M. J. Feeley, W. E. Morgan, F. H. Pighin, A. R. Karlin, H. M. Levy, and C. A. Thekkath. Implementing global memory management in a workstation cluster. In *Proc. of the 15th ACM SOSP*, pages 201–212, December 1995.

[5] L. M. Feeney and M. Nilsson. Investigating the energy consumption of a wireless network interface in an ad hoc networking environment. In *Proc. of the 20th INFOCOM*, pages 1548–1557, April 2001.

[6] T. Hara. Cooperative caching by mobile clients in push-based information systems. In *Proc. of the 11th CIKM*, pages 186–193, November 2002.

[7] W. H. O. Lau, M. Kumar, and S. Venkatesh. A cooperative cache architecture in support of caching multimedia objects in MANETs. In *Proc. of the 5th ACM WoWMoM*, pages 56–63, September 2002.

[8] W.-C. Lee and D. L. Lee. Signature caching techniques for information filtering in mobile environments. *Wireless Networks*, 5(1):57–67, January 1999.

[9] M. Li, W.-C. Lee, and A. Sivasubramaniam. Neighborhood signatures for searching P2P networks. In *Proc. of the 7th IDEAS*, pages 149–159, July 2003.

[10] L. Ramaswamy and L. Liu. A new document placement scheme for cooperative caching on the Internet. In *Proc. of the 22nd ICDCS*, pages 95–103, July 2002.

[11] F. Sailhan and V. Issarny. Cooperative caching in ad hoc networks. In *Proc. of the 4th MDM*, pages 13–28, January 2003.

[12] P. Sarkar and J. H. Hartman. Hint-based cooperative caching. *ACM TOCS*, 18(4):387–419, November 2000.

[13] J. E. Wieselthier, G. D. Nguyen, and A. Ephremides. Energy-efficient broadcast and multicast trees in wireless networks. *MONET*, 7(6):481–492, December 2002.