An Analytical Study of Cooperative Data Dissemination in Push-based Mobile Environments

Chengcheng Dai*, Chi-Yin Chow*, Hong Va Leong† and Alvin T. S. Chan†
* Department of Computer Science, City University of Hong Kong, Hong Kong, China
Email: chengcdai2@student.cityu.edu.hk, chiychow@cityu.edu.hk
† Department of Computing, The Hong Kong Polytechnic University, Hong Kong, China
Email: cshleong@comp.polyu.edu.hk, cstschan@comp.polyu.edu.hk

***Abstract*—Caching is an important technique for improving the data retrieval performance of mobile clients who store frequently needed data items in their local cache, often of a limited size. With the concept of cooperative data dissemination (CDD), the mobile clients not only retrieve data items from mobile support stations, but also from the cache in their peers. This paper describes a CDD framework in a push-based mobile environment and proposes a generic analytical model for evaluating the performance of the CDD framework in terms of the access latency and the tune-in time. Analytical results show that CDD effectively reduces the data access latency and tune-in time in the push-based mobile environment. The results of this work provide requisite knowledge and insights for designing new CDD frameworks in the push-based mobile environment.**

I. Introduction

With the advances of new **peer-to-peer** (known as P2P throughout this paper) wireless communication technologies and portable devices, a new information sharing paradigm, known as mobile P2P information access has rapidly taken shape. In this new information sharing paradigm, mobile clients can obtain their desired data items not only from the mobile support stations (MSSs), but also from the cache in their peers rather than relying solely on the server to search data items and disseminate from the information source to the requesting clients. This paradigm is referred to as *mobile* P2P *Cooperative Data Dissemination* CDD, which is appropriate for an environment where a group of mobile hosts (MHs) have a common access interest. For example, in an exhibition, the MHs crowded near to the same booth are probably interested in information related to the booth. In a shopping mall, MHs residing in the same shop are likely to access information pertaining to the shop. At a conference, the participants staying in the same session room also likely possess common research interests, e.g., they may download similar documents of the proceedings or related articles from the server. When the MHs share a common access interest, there is a higher probability for them to find their required data items from the cache in their peers.

The traditional data dissemination models in mobile systems can be classified into two types of architecture, *infrastructure*- and *ad-hoc-based*. In the infrastructure-based mobile system, the wireless network connecting MHs can retrieve their desired data items from the MSSs, either by retrieving them over shared point-to-point channels, i.e., a *pull-based* data dissemination model, or catching them from scalable broadcast channels, i.e., a *push-based* data dissemination model, or through the utilization of both these types of channels, i.e., a *hybrid* data dissemination model. For an ad-hoc-based mobile system, also known as a Mobile Ad hoc NETwork (MANET), the MHs can share information among themselves without any help of the MSS, i.e., a P2P data dissemination model.

For the *push-based* data dissemination model, although there is no scalability bottleneck as the uplink channel bandwidth in the *pull-based* data dissemination model when serving an enormous number of MHs [1], the MHs generally suffer longer access latency and higher power consumption than those in the *pull-based* one. This is because the MHs need to tune in to the broadcast channel and wait for the index or their desired data items to appear. Furthermore, since the data items are broadcast sequentially, the MHs experience longer access latency with an increase of the number of data items being broadcast. While in MANETs, the MHs could suffer from long access latency or access failure, when the peers holding the desired data items are far way or unreachable due to frequent network partitioning [2] or client disconnection.

In reality, long access latency or access failure could possibly cause the abortion of valuable transactions or the suspension of critical activities, so that it is likely to reduce user satisfaction and loyalty, and potentially bring damages to the organization involved. The drawbacks of the existing mobile data dissemination models motivate us to evaluate the performance of mobile information access applications using CDD in the push-based mobile environment.

In this paper, we describe a CDD framework for a push-based mobile environment that combines the conventional infrastructure-based push-based mobile system with P2P communication techniques. We also provide an analytical models for both conventional push-based data dissemination and the CDD framework. Mobile P2P cooperative data dissemination strategies for mobile clients have been well studied. However, there still lacks a generic analytical model for studying the system performance of the CDD framework in the push-based mobile environment. For example, given local and global cache hit ratios of a mobile P2P cooperative data dissemination strategy, we can derive the access latency and tune-in time from our analytical model with respect to various system settings, e.g., the number of users and the P2P communication bandwidth. Analytical results show that CDD effectively reduces the data access latency and tune-in time in the push-based mobile environment. The results of this work provide requisite knowledge and insights for designing new CDD frameworks in the push-based mobile environment.

The rest of this paper is organized as follows. Section II highlights related work. The system model and analytical model of the CDD framework in the push-based mobile environment are described in Sections III and IV, respectively. Section V presents analytical results. Finally, Section VI concludes this paper.

II. Related Work

The related work in CDD mainly focuses on how to search data items, and how to disseminate them from the source MH or the MSS to the requesting MH. An intuitive cooperative data dissemination framework has been proposed for MANETs, in which if an MH can connect directly to an MSS, it retrieves the required data item from the MSS; otherwise, it has to enlist other peers that are closer to the MH than the MSS for help to turn in the required data item [3]. If no peer caches the required data item, the request is routed to the nearest MSS. Another similar cooperative data dissemination framework has been proposed to support continuous media access in MANETs [4]. In this framework, *proactive* and *on-demand* data location schemes are designed for the MHs to determine the nearest data source to retrieve their required multimedia objects. Wang and Li have proposed three cooperative data dissemination schemes, namely, *CacheData*, *CachePath*, and *HybridCache* [5]. The key idea of *CacheData* is that an MH caches a passing-by data item, if the data item satisfies certain conditions. The MH adopting *CachePath* caches the path information of the passing-by data item instead of the data item. *HybridCache* is a hybrid scheme which combines both *CacheData* and *CachePath*. The 7DS scheme has been proposed as a complement to the infrastructure support with power conservation [6]. When an MH fails to connect to the Internet to retrieve the desired data item, it attempts to search its neighboring 7DS peers for the data item. The proposed power conservation scheme adjusts the MHs' degree of activity or participation in cooperative data dissemination based on their battery levels. Another power conservation, called ECOR, has been proposed for MHs [7], where they exchange the cache content and the optimal search radius (in hops) of each cached data item among them. The MH can get its desired data item from its peers within the optimal search radius. A cocktail cache resolution scheme is proposed for MHs to locate their desired data items in MANETs [8]. Resource discovery services have also been designed for MANETs [9]–[11], where they simply disseminate queries and resource information in the network. Once a match between a query and an available resource is found, the resource information is sent back to the sender of the matched query. There are also cooperative semantic caching schemes that are designed to support location-based query processing [12]–[15].

This paper distinguishes itself from previous work that it presents an analytical model to evaluate the performance of the CDD framework in the push-based mobile environment in terms of the access latency and the tune-in time to the broadcast channel. The results of this work provide requisite knowledge and insights for designing new mobile applications using CDD in the push-based mobile environment.

III. System Model

In this section, we present the system model of our CDD framework and a P2P (i.e., peer-to-peer) data dissemination protocol. There are three possible outcomes of each client request in the CDD framework. (1) **Local Cache Hit (LCH)**. If an MH finds its required data item in its local cache, a local cache hit takes place; otherwise, it is a local cache miss. (2) **Global Cache Hit (GCH)**. When the MH encounters a local cache miss, it attempts to retrieve the data item from its peers. If a peer can turn in the data item to the MH, it constitutes a GCH. (3) **Cache Miss (or Server Request)**. If the MH fails to achieve neither a LCH nor a GCH, i.e., a cache miss, it accesses the required data item from the MSS.

In the CDD framework, each MH is equipped with two wireless network interface cards (NICs) supporting dual bands, i.e., 2.4GHz 802.11b/g and 5GHz 802.11a (e.g., Linksys wireless dual-band adapter [16]). These two NICs are tuned to different, non-interfering channels [17]–[19]; one is dedicated to communicate with the MSS, while the other one is devoted to communicate with other MHs. Furthermore, there are two P2P communication paradigms: *point-to-point* and *broadcast*. In the P2P point-to-point communication, there is only one destination MH for the message being sent from a source MH. In the P2P broadcast communication, all MHs residing in the transmission range of a source MH receive the broadcast message.

The proposed P2P data dissemination protocol consists of three types of messages: request, reply and retrieve. A request message contains a unique identifier that is formed by a combination of the user identifier (*ID*) and the request time (*TS*), i.e., $\langle ID, TS\rangle$. The peers only process a request with the same identifier once, and they simply drop duplicate requests. The key idea of the data dissemination protocol is that when an MH encounters a local cache miss, it broadcasts a request message to its neighboring peers. If

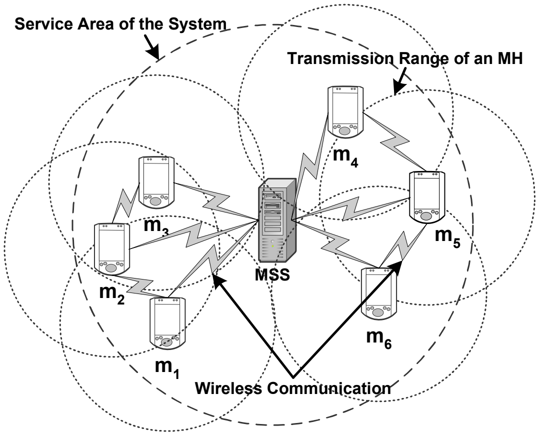


Fig. 1. System architecture of the CDD framework.

the neighbour caches the required data item, it sends a reply message to the MH directly. Otherwise, it simply drops the request message. After the requesting MH receives the reply message from the peer, it sends a retrieve message to the peer, and then the peer turns in the required data item to the MH. If the requesting MH receives multiple reply messages, it selects one peer to turn in the required data item based on certain criteria, e.g., response time and battery power. Figure 1 depicts the system architecture of CDD. When an MH m_2 encounters a *local cache miss*, m_2 broadcasts a request to its neighboring peers, i.e., m_1 and m_3. If m_1 or m_3 can turn in the required data item to m_2, a GCH is recorded; otherwise, m_2 encounters a *cache miss*, so it enlists the MSS for help.

IV. Analytical Models

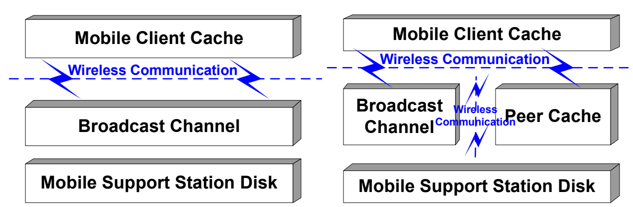
In this section, we first describe the storage hierarchy and the CDD framework in the push-based mobile environment and analyze its data access latency and tune-in time. Table I depicts the meaning of the symbols used in the analytical models.

A. Storage hierarchy

The storage hierarchy of the push-based mobile environment consists of three layers: Mobile Client Cache, Broadcast Channel and MSS Disk, as depicted in Figure 2a. The MSS grabs the data items from a disk or a database server, and allocates them to the broadcast channel. If an MH encounters a local cache miss, it tunes in to the broadcast channel, and catches the required data item when the data item appears in the broadcast channel. In CDD, we insert the Peer Cache layer as a supplementary component of the Broadcast Channel layer, as shown in Figure 2b. When an MH encounters a local cache miss, it tunes in to the broadcast channel; meanwhile, it searches the Peer Cache layer for the required data item. A GCH occurs, when some neighbours can return the required data item to the MH before the data item appears in the broadcast channel. Otherwise, the MH has to wait for the data item to appear in the broadcast channel, as in the conventional scheme.

TABLE I
SYMBOLS USED IN THE ANALYTICAL MODEL

Symbol	Meaning
P_L	The probability of a local cache miss
P_G	The probability of a global cache miss
P^{InitR}	The probability of accessing a data item from the broadcast channel before the index
|request|	The size of a request message
|reply|	The size of a reply message
|retrieve|	The size of a retrieve message
|data|	The size of a data item
BW_{Push}	The broadcast channel bandwidth in the *push* mobile environment
TP_{P2P}	Peer-to-peer throughput capacity per MH
$InitR$	The time from the MH tunes in to the broadcast channel to the time when the required data item appears in the channel before the index
$Probe$	The time from the MH tunes in to the broadcast channel to the time when the next index is broadcast
$Search$	The time from the MH gets the index to the time when the MH finds the arrival time of the required data item
$Retrieve$	The time the MH spends on downloading the required data item from the broadcast channel
$Doze$	The time duration of the MH in the doze mode
$Index$	The number of buckets for one complete index tree
$Data$	The number of buckets for data items between two successive complete index trees
L^{Hit}_{P2P}	The latency of a global cache hit
L^{Reply}_{P2P}	The latency of getting a reply message for a global cache search
D	The number of data items
B	The number of buckets for a data item
B_c	The number of buckets for the data items in a broadcast cycle (excluding the index)
S	Bucket size
h	The height of an index tree
n	The number of index attributes in a bucket
N	The number of MHs
m	The number of complete index trees in a broadcast cycle
$\|disk\|$	The number of broadcast disks
$disk\_size_i$	The size of a broadcast disk i
rel_freq_i	The relative frequency of a broadcast disk i
Δ	The number of hottest data items switched to the slowest broadcast disk
θ	Skewness parameter for Zipf distribution
f_i	The broadcast frequency of a data item i
P_i	The access probability of a data item i



(a) Conventional (b) CDD

Fig. 2. Storage hierarchy of the push-based mobile environment.

B. Broadcast scheduling algorithm

The conventional scheme and CDD in the push-based mobile environment are analyzed with the *broadcast disk* [20] broadcast scheduling algorithm. To reduce power consumption, we employ the $(1, m)$-Indexing technique [21], which is an index allocation scheme that evenly broadcasts a complete index m times in a broadcast cycle. The access latency (*Access*) and tune-in time (*Tune*) of the broadcast disk algorithm are:

$$Access = InitR \text{ or } (Probe + Search + Doze + Retrieve)$$
$$Tune = InitR \text{ or } (Probe + Search + Retrieve)$$

where *InitR* is the duration from the time when the MH tunes in to the broadcast channel to the time when the required data item appears in the channel before the index; *Probe* is the duration from the time when the MH tunes in to the broadcast channel to the time when the next index is broadcast; *Search* is the duration from the time when the MH gets the index to the time when it figures out the arrival time of the required data item by following a list of pointers; *Doze* is how long the MH stays in the doze mode, after it determines the arrival time of its required data item; and *Retrieve* is the time for the MH to download the required data item from the broadcast channel.

Let S denote a bucket size, D denote the total number of data items, B denote the number of buckets for a data item (i.e., $B = \frac{|\text{data}|}{S}$), and B_c denote the total number of buckets for the data items in a broadcast cycle (i.e., excluding the buckets for the index). Furthermore, let h and n be the height of the index tree and the number of index attributes that a bucket can hold, respectively. Finally, let $Index$ and $Data$ be the number of buckets occupied by the index tree and the number of buckets occupied by the data items between two successive index trees, respectively. Since a complete index is evenly broadcast m times in a broadcast cycle, i.e., $Data = \frac{B_c}{m}$. When an index tree is fully balanced, $h = \lceil \log_n(D) \rceil$ and $Index = \sum_{i=0}^{h-1} n^i$.

C. Analytical Models

We here consider the following two parameters to evaluate the efficiency of data dissemination: *a. Access Latency.* The expected time from the time a client requests a data item to the time the client obtains the data item. *b. Tuning time.* The amount of time spent by a client listening to the channel. This will determine the power consumption of a client to retrieve the required data.

Conventional scheme. The *broadcast disk* algorithm partitions all data items into several ranges. Each range that contains the data items with similar access probabilities is referred to as a $disk$. Also, each disk, $disk_i$, with size, $disk\_size_i$, has its own relative broadcast frequency, rel_freq_i. The larger the relative broadcast frequency rel_freq, the higher the disk spinning speed. The hottest data items are allocated to the $disk$ with the highest rel_freq. Likewise, the coldest data items are arranged to the $disk$ with the slowest rel_freq. Since most MHs cache the hottest data items, the most Δ hottest data items are shifted to the slowest rel_freq to obtain better performance [20]. For simplicity, we assume the access pattern follows the Zipf distribution with a skewness parameter θ [22], so the access probability of a data item d_i is $P_i = \frac{1}{i^\theta \sum_{j=1}^{D} \frac{1}{j^\theta}}$. $|disk|$ disks are used, i.e., the relative frequencies of $disk_1, disk_2, \ldots, disk_{|disk|}$ are $rel\_freq_1, rel\_freq_2, \ldots, rel\_freq_{|disk|}$, respectively, where $rel\_freq_1 \geq rel\_freq_2 \geq \ldots \geq rel\_freq_{|disk|}$. The broadcast frequency f_i of each data item d_i based on the index i is as follows: For $\Delta < i \leq disk\_size_1 + \Delta$, $f_i = rel\_freq_1$. For $\sum_{j=1}^{k-1} disk\_size_j + \Delta < i \leq \sum_{j=1}^{k} disk\_size_j + \Delta$, $f_i = rel\_freq_k$ where $1 < k < |disk|$. For $\sum_{j=1}^{|disk|-1} disk\_size_j + \Delta < i \leq \sum_{j=1}^{|disk|} disk\_size_j$ or $1 \leq i \leq \Delta$, $f_i = rel\_freq_{|disk|}$.

The expected number of buckets before the arrival of the desired data item is $\frac{Index \times m + B_c}{2 \times f_i}$ [23]. The number of buckets for data items in a broadcast cycle excluding the index trees is $B_c = Data \times m = B \times \sum_{i=1}^{D} f_i = B \times \sum_{j=1}^{|disk|} disk\_size_i \times rel\_freq_i$. Hence, $InitR$, $Probe$, $Search$, $Doze$ and $Retrieve$ in the push-based mobile environment are:

$$InitR = (\frac{1}{2} \times (Index + Data) + B) \times \frac{S}{BW_{Push}}$$
$$Probe = \frac{1}{2} \times (Index + Data) \times \frac{S}{BW_{Push}}$$
$$Search = (h+1) \times \frac{S}{BW_{Push}}$$
$$Doze = \frac{Index \times m + B_c}{2 \times f_i} \times \frac{S}{BW_{Push}}$$
$$Retrieve = B \times \frac{S}{BW_{Push}}$$

Since some data items are broadcast more than once in a broadcast cycle, P^{InitR}, the probability of accessing a data item from the broadcast channel before the index, is based on the broadcast frequency f_i of a data item d_i:

$$P^{InitR} = \frac{f_i}{m} \times [\frac{Index}{Index + Data} + \frac{Data}{Index + Data} \times \sum_{i=1}^{Data/B} (\frac{1}{Data/B} \times \frac{i}{Data/B})]$$

Access$_{nc}$ and *Tune*$_{nc}$ of the conventional scheme in the push-based mobile environment are:

$$Access_{nc} = \sum_{i=1}^{D} \{P^{InitR} \times InitR + (1 - P^{InitR}) \times (Probe + Search + Doze + Retrieve)\} \times P_i$$
$$Tune_{nc} = \sum_{i=1}^{D} \{P^{InitR} \times InitR + (1 - P^{InitR}) \times (Probe + Search + Retrieve)\} \times P_i$$

Cooperative data dissemination (CDD). In CDD, we consider that each MH's device has the same transmission range with a fixed bandwidth in a network with random traffic patterns, hence the throughput capacity per MH is $TP_{P2P} = \Theta(\frac{BW_{P2P}}{\sqrt{N \log N}})$ [24]. We further assume that the processing time of handling a global cache query is negligible and the latency of a global cache access depends on the transmission time. Thus, the latency of a global cache hit is $L_{P2P}^{Hit} = \frac{|\text{request}| + |\text{reply}| + |\text{retrieve}| + |\text{data}|}{TP_{P2P}}$.

TABLE II
PARAMETER SETTINGS OF THE ANALYTICAL MODELS.

$D = 10\text{K}$, $P_L = 0.05$, $P_G = 0.1$ to 0.5, $\theta = 0.5$
$n = 10$, $\Delta = 100$, $m = 30$, $S = 64\text{bytes}$
$rel\_freq_{1,2,3} = 3, 2, 1$, $disk\_size_{1,2,3} = 1\text{K}, 3\text{K}, 6\text{K}$
$\lvert disk \rvert = 3$, $h = 4$, $N = 500$, $\lvert \text{data} \rvert = 4\text{KB}$
$\lvert \text{reply} \rvert = \lvert \text{request} \rvert = \lvert \text{retrieve} \rvert = 64\text{bytes}$
$BW_{Push} = 10\text{Mbps}$, $BW_{P2P} = 2\text{Mbps}$

If the MH of CDD can get its required data item from its neighbour before the data item appears in the broadcast channel, the access latency can be improved. Likewise, the tune-in time can be reduced, if the MH receives a reply message before the index is being broadcast. Thus, the access latency and tune-in time of CDD with the *broadcast disk* algorithm are:

$$Access_{cdd} = (1 - P_L) \times \{P_G \times \min(L_{P2P}^{Hit}, Access_{nc}) + (1 - P_G) \times Access_{nc}\}$$
$$Tune_{cdd} = (1 - P_L) \times \{P_G \times \min(L_{P2P}^{Reply}, Tune_{nc}) + (1 - P_G) \times Tune_{nc}\},$$

where $L_{P2P}^{Reply} = \frac{|\text{request}| + |\text{reply}|}{TP_{P2P}}$. Since L_{P2P}^{Reply} is relatively much smaller than $Tune_{nc}$ and L_{P2P}^{Hit} is also substantially smaller than $Access_{nc}$, we consider that L_{P2P}^{Reply} (L_{P2P}^{Hit}) is the minimum value compared with $Tune_{nc}$ ($Access_{nc}$). Thus, $Access_{cdd}$ and $Tune_{cdd}$ can be simplified, without loss of generality, as follows:

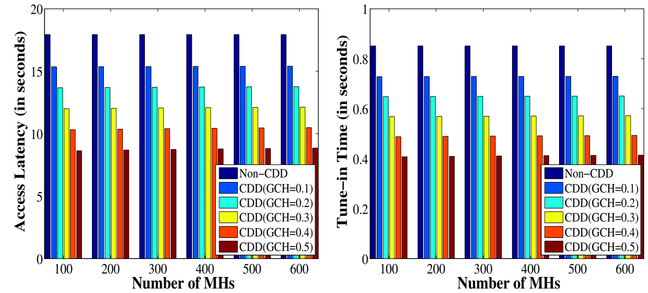
$$Access_{cdd} = (1 - P_L) \times \{P_G \times L_{P2P}^{Hit} + (1 - P_G) \times Access_{nc}\}$$
$$Tune_{cdd} = (1 - P_L) \times \{P_G \times L_{P2P}^{Reply} + (1 - P_G) \times Tune_{nc}\}$$

V. ANALYTICAL RESULTS

In this section, we present the analytical results for the conventional approach (i.e., non-CDD) and the CDD framework in the push-based mobile environment with the *broadcast disk* scheduling algorithm [20] and the $(1, m)$-Indexing technique [21]. The performance of CDD is evaluated with respect to various system settings and given global cache hit (GCH) ratios in terms of the access latency and the tune-in time to the broadcast channel. Table II gives the parameter settings for the analytical models.

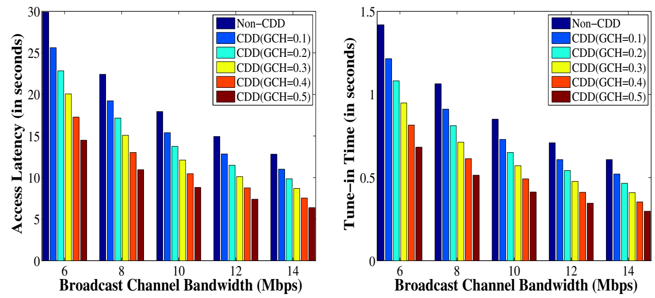
A. Effect of the Number of MHs

Figures 3a and 3b depict the access latency and the tune-in time of non-CDD and CDD with respect to various numbers of MHs from 100 to 600, respectively. Figure 3a shows that CDD gives better access latency than non-CDD as the GCH ratio gets higher. However, the access latency of CDD slightly gets worse, when there are more MHs in the system. This is due to the fact that the throughput of the P2P communication reduces as the number of MHs rises. Thus, CDD is a scalable and effective technique for improving the data access latency. Figure 3b shows



(a) Access Latency (b) Tune-in Time

Fig. 3. The number of MHs.



(a) Access Latency (b) Tune-in Time

Fig. 4. The broadcast channel bandwidth (BW_{Push}).

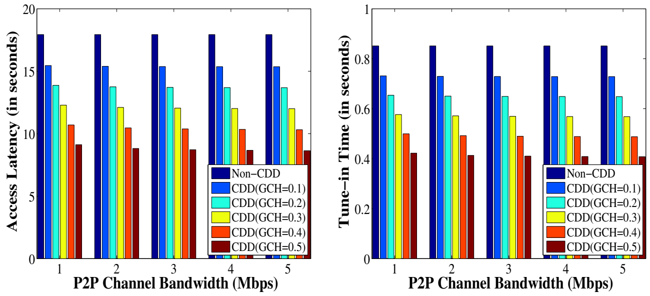
that CDD also reduces the tune-in time, as the GCH ratio increases. However, increasing the number of MHs degrades the throughput of the P2P communication that leads to a higher tune-in time. Since the power consumption is proportional to the tune-in time, the MHs adopting CDD experiencing a shorter tune-in time can save more power. Therefore, CDD is also a power conservation technique in the push-based mobile environment.

B. Effect of Broadcast Channel Bandwidth

Figures 4a and 4b depict the access latency and the tune-in time of non-CDD and CDD with an increase of the bandwidth of the broadcast channel (i.e., BW_{Push}) from 6 to 14 Mbps, respectively. Figure 4a shows that both non-CDD and CDD give better access latency when BW_{Push} increases. This is because it is faster for the MSS to broadcast both data items and indexes.

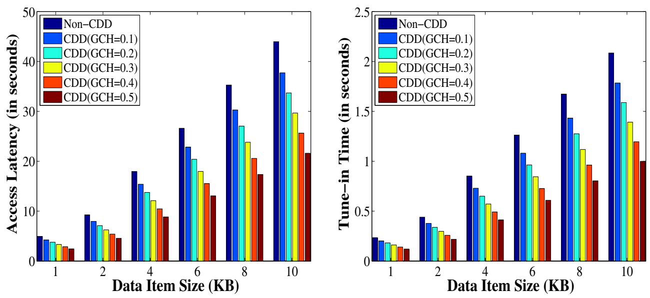
C. Effect of Peer-to-Peer (P2P) Channel Bandwidth

Figures 5a and 5b depict the access latency and the tune-in time with an increase of the bandwidth of the P2P channel (i.e., BW_{P2P}) from 1 to 5 Mbps, respectively. Since non-CDD does not involve any peer cooperation, its performance is not affected by varying BW_{P2P}. Figure 5a shows that the access latency of CDD decreases as the BW_{P2P} increases. This is because the throughput of P2P communication increases as BW_{P2P} rises. CDD gives better access latency than non-CDD as the GCH ratio gets higher because the MH is more likely to get its required



(a) Access Latency (b) Tune-in Time

Fig. 5. The P2P channel bandwidth (BW_{P2P}).



(a) Access Latency (b) Tune-in Time

Fig. 6. The size of data items (|data|).

data item from peers before the data item appears in the broadcast channel. Figure 5b shows that CDD also reduces the tune-in time when the GCH ratio increases.

D. Effect of the Size of Data Items

Figures 6a and 6b depict that both the access latency and the tune-in time of non-CDD and CDD get higher as data item size increases from 1 to 10 KB, respectively. The main reason is that when the data item size gets larger, both the MSS and MHs take more time to broadcast and send a data item, respectively. Since retrieving a data item or getting a reply message from peers is usually before the data item or index appears in the broadcast channel, CDD gives better access latency and tune-in time than non-CDD as the GCH ratio rises.

VI. Conclusion

In this paper, we have presented the cooperative data dissemination (CDD) framework in the push-based mobile environment. In CDD, the mobile hosts can get their required data items not only from the mobile base station, but also from their peers. We have also proposed the analytical models to evaluate the system performance of the conventional approach and the CDD framework. Analytical results show that CDD is more efficient than the conventional approach in the push-based mobile environment in terms of both the access latency and the tune-in time to the broadcast channel. Our future directions will focus on the analytical study of CDD in the pull-based and hybrid mobile environment and the empirical study of the CDD framework.

Acknowledgement

The work described in this paper was partially supported by grants from City University of Hong Kong (Project No. 7002867, 7002722, and 7002686).

References

[1] S. Acharya, M. Franklin, and S. Zdonik, “Balancing push and pull for data broadcast,” in *ACM SIGMOD*, 1997.
[2] K. H. Wang and B. Li, “Efficient and guaranteed service coverage in partitionable mobile ad-hoc networks,” in *IEEE Infocom*, 2002.
[3] F. Sailhan and V. Issarny, “Cooperative caching in ad hoc networks,” in *MDM*, 2003.
[4] W. H. O. Lau, M. Kumar, and S. Venkatesh, “A cooperative cache architecture in support of caching multimedia objects in MANETs,” in *WoWMoM*, 2002.
[5] L. Yin and G. Cao, “Supporting cooperative caching in ad hoc networks,” *IEEE TMC*, vol. 5, no. 1, pp. 77–89, 2006.
[6] M. Papadopouli and H. Schulzrinne, “Effects of power conservation, wireless coverage and cooperation on data dissemination among mobile devices,” in *ACM MobiHoc*, 2001.
[7] H. Shen, S. K. Das, M. Kumar, and Z. Wang, “Cooperative caching with optimal radius in hybrid wireless network,” in *IFIP NETWORKING*, 2004.
[8] Y. Du and S. K. S. Gupta, “COOP - A cooperative caching service in MANETs,” in *IARIA ICNS*, 2005.
[9] U. Lee, J. Lee, J.-S. Park, and M. Gerla, “FleaNet: A virtual market place on vehicular networks,” *IEEE TVT*, vol. 59, no. 1, pp. 344–355, 2010.
[10] O. Wolfson, B. Xu, and R. M. Tanner, “Mobile peer-to-peer data dissemination with resource constraints,” in *MDM*, 2007.
[11] X. Zhu, B. Xu, and O. Wolfson, “Spatial queries in disconnected mobile networks,” in *ACM GIS*, 2008.
[12] C.-Y. Chow, M. F. Mokbel, and H. V. Leong, “On efficient and scalable support of continuous queries in mobile peer-to-peer environments,” *IEEE TMC*, vol. 10, no. 10, pp. 1473–1487, 2011.
[13] W.-S. Ku, R. Zimmermann, and H. Wang, “Location-based spatial query processing in wireless broadcast environments,” *IEEE TMC*, vol. 7, no. 6, pp. 778–791, 2008.
[14] F. J. Maymi, M. Rodriguez-Martinez, and W. Rolke, “A cooperative spatial-aware cache for mobile environments,” in *ACM MobiDE*, 2010.
[15] Q. Zhu, D. L. Lee, and W.-C. Lee, “Collaborative caching for spatial queries in mobile P2P networks,” in *IEEE ICDE*, 2011.
[16] Linksys AE2500, “Wireless dual-band USB adapter,” http://home.cisco.com/en-mea/products/adapters/AE2500; Last accessed: August 22, 2012.
[17] R. Draves, J. Padhye, and B. Zill, “Routing in multi-radio, multi-hop wireless mesh networks,” in *ACM MobiCom*, 2004.
[18] B.-J. Ko, V. Misra, J. Padhye, and D. Rubenstein, “Distributed channel assignment in multi-radio 802.11 mesh networks,” in *IEEE WCNC*, 2007.
[19] J. Zhao, P. Zhang, and G. Cao, “On cooperative caching in wireless p2p networks,” in *IEEE ICDCS*, 2008.
[20] S. Acharya, R. Alonso, M. Franklin, and S. Zdonik, “Broadcast Disks: Data management for asymmetric communication environments,” in *ACM SIGMOD*, 1995.
[21] T. Imielinski, S. Viswanathan, and B. R. Badrinath, “Data on Air: Organization and access,” *IEEE TKDE*, vol. 9, no. 3, pp. 353–372, 1997.
[22] G. K. Zipf, *Human Behavior and the Principle of Least Effort*. Addison-Wesley, 1949.
[23] Q. Hu, W.-C. Lee, and D. L. Lee, “A hybrid index technique for power efficient data broadcast,” *Distributed and Parallel Databases*, vol. 9, no. 2, pp. 151–177, 2001.
[24] P. Gupta and P. Kumar, “Capacity of wireless networks,” *IEEE Trans Inf Theory*, vol. 46, no. 2, pp. 388–404, 2000.