Computer Networks 57 (2013) 197–212

Contents lists available at SciVerse ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

An update-based step-wise optimal cache replacement for wireless data access

Hui Chen [a], Yang Xiao [b,*], Susan V. Vrbsky [b]

[a] Dept. of Math & Computer Science, Virginia State University, Petersburg, VA 23806, USA
[b] Dept. of Computer Science, Univ. of Alabama, Tuscaloosa, AL 35487, USA

ARTICLE INFO

Article history:
Received 7 December 2011
Received in revised form 30 June 2012
Accepted 12 September 2012
Available online 21 September 2012

Keywords:
Cache
Replacement policy
Access scheme
Data update
Wireless data access

ABSTRACT

Many network applications requires access to most up-to-date information. An update event makes the corresponding cached data item obsolete, and cache hits due to obsolete data items become simply useless to those applications. Frequently accessed but infrequently updated data items should get higher preference while caching, and infrequently accessed but frequently updated items should have lower preference. Such items may not be cached at all or should be evicted from the cache to accommodate items with higher preference. In wireless networks, remote data access is typically more expensive than in wired networks. Hence, an efficient caching scheme considers both data access and update patterns can better reduce data transmissions in wireless networks. In this paper, we propose a step-wise optimal update-based replacement policy, called the *Update-based Step-wise Optimal* (*USO*) policy, for wireless data networks to optimize transmission cost by increasing *effective hit ratio*. Our cache replacement policy is based on the idea of giving preference to frequently accessed but infrequently updated data, and is supported by an analytical model with quantitative analysis. We also present results from our extensive simulations. We demonstrate that (1) the analytical model is validated by the simulation results and (2) the proposed scheme outperforms the Least Frequently Used (LFU) scheme in terms of effective hit ratio and communication cost.

1. Introduction

Wireless networks are one of the essential technologies to realize ubiquitous data access, i.e., data access at any time, at any place, and in any form. With recent deployment of high bandwidth cellular wireless networks (3G, 3.5G, and 4G) and wireless LANs, there is an growing demand on data access in wireless networks from large user community [1]. In an information retrieval system, databases and files are hosted at a remote server, conventionally located on the wired networks. Each database or file server hosts a number of data items. Users access these data items through mobile terminals, termed as clients, in wireless networks. Whenever these mobile terminals need to access a data item from the remote server, all the communications have to pass through the wireless network.

In a wireless data access application, wireless links are the most scarce and the most expensive resources. For the best utilization of these resources, a client has to be very economic about the usages. In many cases, the problem is handled by developing network-aware applications, such as retrieving images with adaptive quality, adjusted to the available bandwidth. Beside developing adaptive applications, one novel and ubiquitous way to solve the problem is to via caching (e.g., [2–11]). In general, data access applications employ caching of recurrently accessed data locally. Caching contributes in three ways to increase the performance of such applications and network systems. Firstly, the average access latency is reduced as many data items

* Corresponding author. Tel.: +1 205 348 4038.
E-mail addresses: huichen@ieee.org (H. Chen), yangxiao@ieee.org (Y. Xiao), vrbsky@cs.ua.edu (S.V. Vrbsky).

1389-1286/$ - see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.comnet.2012.09.012

are fetched locally, instead of fetching from the remote server. Secondly, the network load is reduced, as without cache each data item has to be passed through the network from the server to the client. Reduced network load potentially decreases the cost of data access. Thirdly, as the server gets fewer request from a client, the server becomes more scalable without additional computing and network resources. In wireless networks, by cutting down the number of communication transmissions, caching techniques not only saves on expensive wireless link access but also saves power and prolongs battery life, especially, those proposed for wireless sensor networks (e.g., [12]).

A cache mechanism includes two aspects – (1) a cache *access algorithm* and (2) a *replacement policy*. In general, a cache access algorithm describes how a client–server system uses the cache and maintains consistency between original data items at the server and copies in the caches of the clients. Sometimes it is called a cache *consistency algorithm* or a cache *invalidation scheme*, if investigated from data consistency or data invalidation perspective, respectively. Different applications have different data consistency requirement. To many applications, an old copy of a data item is of no use. These applications require access to data items along with their most recent updates. In literature, this kind of consistency is called the *latest value consistency*[13]. A cache consistency algorithm which ensures the latest value consistency is called a *strongly consistent* caching algorithm [5,13]. When a strongly consistent caching algorithm is used, an obsolete cached data item becomes invalid and cannot be used. In this case, the client has to retrieve the data item from the server. Various cache access algorithms have been proposed and studied for mobile/wireless data access. Among those, strongly consistent algorithms are *Invalidation Report (IR)* schemes [3,6,7,10, 13–21], *Poll-Each-Read*[5] and *Call-back*[5,21–24]. To some other applications, an old copy of a data item is usable, i.e., the latest value consistency or strong consistency is not required. In literature, this kind of consistency is called *weak consistency*. A cache consistency algorithm which only ensures weak consistency is called a *weakly consistent* caching algorithm. The weak consistency algorithms typically rely on a time-to-live (TTL) value assigned to a cached item [25–27]. Most recent work in cooperative caching also falls into this category (e.g., [2,4]). During the time period specified by the TTL value, the cached data item is usable to the application although the item may be obsolete. Many weakly cache consistent algorithms are proposed, such as those in [11,28].

A replacement comes into play when the cache is full and an accessed data item has to be accommodated. Hence, one or more cached data items have to be evicted to make room for the newly arrived item. In general, a cache access algorithm can be combined with any replacement policy. Most researchers have studied cache access algorithms with the *Least Recently Used* (*LRU*) [3,5–7,10,13–21] and *Least Frequently Used* (*LFU*) [29] replacement policies for wireless data access. Note that, a cache access algorithm performs differently when combined with different replacement policy. It is important to choose the appropriate replacement policy for a particular cache access algorithm and vice versa to attain a better performance out of a caching system.

Update process plays an important role because it makes the locally cached data items obsolete. It is beneficial for a replacement policy to utilize update and access information together in mobile wireless data access. The studies in [5] investigates LRU as replacement policy without considering the effects of the update process. In other researches [8,9], replacement policies have been proposed to utilize update information. However, these replacement policies are studied with IR schemes and their efforts are spent to reduce the *stretch*[1] [30]. The IR schemes perform well under the following assumptions: (1) the channel is a broadcast channel, (2) all the clients are within one wireless cell in a Personal Communication Service (PCS) network, and (3) the server is also local to the wireless cell, i.e., at the base station in a PCS network. However, in a realistic wireless network, none of the above assumptions may be satisfied. In other words, the server is more likely to be at a remote site, and subscribed clients are scattered all over the PCS network or even in remote wireless networks. For example, if all the clients are in different wireless cells, the IR schemes become very expensive, since IR reports include too much information that is irrelevant to clients in different wireless cells. Furthermore, IR schemes may require broadcasting in entire network to support clients distributed over different wireless cells [5]. A suitable implementation of an IR scheme requires cross layer support for efficiency, and hence, IR schemes may not always be realistic in practical wireless networks. In addition, in an IR scheme, an MS has to wait for the next invalidation report before answering a query [14]. The waiting can introduce large latency as a result of large IR period. Applications that require near real time access, such as a Web 2.0 application that requires responsive user interactions may not afford to such a large latency.

In this study, we are interested in strongly consistent cache access algorithms and update-based replacement policies that are applicable to practical wireless networks with more than one wireless cells. In particular, we introduce a *Server-Based Poll-Each-Read* (*SB-PER*) and a *Revised Call-back* (*R-CB*) access algorithms which can provide both access and update histories to the replacement policy. These two cache access algorithms allow prompt answers to queries and can be applied to support applications that require near real time access. Our replacement policy is based on the intuition that infrequently accessed but frequently updated data items should be evicted to accommodate new items. In this paper, we also present analytical analysis for both the SB-PER and R-CB. We demonstrate that our replacement policy is step-wise optimal. The design goals of the proposed caching scheme are – (1) to increase the effective hit ratio, and (2) to reduce transmission cost over wireless links. Simulations are performed to validate our proposal and claims.

The rest of the paper is organized as follows. We present our system model and performance metrics in Section 2. The SB-PER and the R-CB cache access policies, and our cache replacement policy are introduced in Section 3. Quantitative analysis with an analytical model is provided

[1] Stretch is defined as the ratio of average response time to service time, where service time is the response time as if there exists no other job in the system.