Chapter 12
Peer-Based Collaborative Caching and Prefetching in Mobile Broadcast

Wei Wu
Singapore-MIT Alliance, and School of Computing, National University of Singapore, Singapore

Kian-Lee Tan
Singapore-MIT Alliance, and School of Computing, National University of Singapore, Singapore

ABSTRACT

Caching and prefetching are two effective ways for mobile peers to improve access latency in mobile environments. With short-range communication such as IEEE 802.11 and Bluetooth, a mobile peer can communicate with neighboring peers and share cached or prefetched data objects. This kind of cooperation improves data availability and access latency. In this chapter the authors review several cooperative caching and prefetching schemes in a mobile environment that supports broadcasting. They present two schemes in detail: CPIX (Cooperative PIX) and ACP (Announcement-based Cooperative Prefetching). CPIX is suitable for mobile peers that have limited power and access the broadcast channel in a demand-driven fashion. ACP is designed for mobile peers that have sufficient power and prefetch from the broadcast channel. They both consider the data availability in local cache, neighbors’ cache, and on the broadcast channel. Moreover, these schemes are simple enough so that they do not incur much information exchange among peers and each peer can make autonomous caching and prefetching decisions.

INTRODUCTION

Mobile broadcast is a scalable data dissemination model for mobile computing (Acharya & Alonso, 1995; Imielinski, 1997; Tan, 2000). In mobile broadcast, a server broadcasts data objects on a wireless channel and (a large number of) mobile peers get their required data objects by tuning into the broadcast channel and retrieving the data objects when they appear. Data broadcast differs
Peer-Based Collaborative Caching and Prefetching in Mobile Broadcast

from traditional point-to-point access in that the broadcast channel is open to all mobile clients and one transmission of a data object on the broadcast channel can satisfy the needs of potentially many clients. Mobile broadcast is especially suitable for data dissemination in asymmetric communication environments where the client to server ratio is large and there is a high degree of commonality among client interests. Information interesting to the majority of the clients is more suitable for broadcast. Many projects and systems are based on the data broadcast technology (Acharya & Franklin, 1995; Acharya, 1997; Acharya, 1998; Altinel, 1999; Hughes, 2008; Gifford, 1990; Imieliński, 1997; Microsoft, 2008; Zheng, 2005). They are sometimes referred to in the literature as Dissemination-Based Information Systems (DBIS) (Franklin, 1996).

Mobile peers in broadcast environments sometimes suffer from long access latency (the time elapsed from the moment a client has a query for a data object to the point when the client gets the data object), especially when the broadcast cycle is long due to large volume of data or limited broadcast channel. When the broadcast cycle is long, a mobile peer has to wait a long time before their required data objects appear on the broadcast channel.

Caching and prefetching are two effective ways to improve response time. They both store copies of data objects locally for future use. The difference is that caching happens after data access while prefetching stores data objects that are not currently under demand but believed to be useful in the future. In other words, caching is driven by data accesses, and prefetching is driven by anticipation of future accesses. In the environments of mobile broadcast, caching is the mechanism used to store a data object after it is taken from the broadcast channel to fulfill a pending request, and prefetching is to actively listen to the broadcast channel to grab objects that are anticipated to be useful. Thus in prefetching the mobile peer listens to the broadcast channel even when there is no pending request and stores interesting objects locally. A carefully designed prefetching scheme results in better access latency than a caching scheme does, while consuming more energy (Acharya, 1996).

The fact that the mobile peers have small local storage space limits the effectiveness of caching and prefetching. Local storage constraint makes it impossible to hold all interesting data objects that may be accessed.

With short-range wireless communication technologies, such as IEEE802.11 and Bluetooth, a mobile peer is able to communicate with other mobile peers in its communication range. Figure 1 is an illustration of cooperative mobile peers in a mobile broadcast environment. A line between two mobile peers means they can communicate directly and share contents in a simple peer-to-peer fashion. This enables the mobile peers to share cached or prefetched data objects: when a mobile peer needs a data object, it can request it from its neighbors (we define a mobile peer’s neighbors as the mobile peers within its communication range, i.e. one hop away). Such cooperation improves applications’ response time, because a mobile peer now can probably get its required data object from its neighbors before getting it on the broadcast channel. This may even reduce energy consumption because getting a data object from

---

Figure 1. Cooperation among mobile peers. A line between two peers means they can communicate directly and share contents in a simple peer-to-peer fashion.
Related Content

Stabilization of Large-Scale Fuzzy Interconnected System
(2017). Large-Scale Fuzzy Interconnected Control Systems Design and Analysis (pp. 34-83).
www.igi-global.com/chapter/stabilization-of-large-scale-fuzzy-interconnected-system/181988?camid=4v1a

Integrated Software Testing Learning Environment for Training Senior-Level Computer Science Students
www.igi-global.com/chapter/integrated-software-testing-learning-environment/29493?camid=4v1a

Enhancing Testing Technologies for Globalization of Software Engineering and Productivity
www.igi-global.com/chapter/enhancing-testing-technologies-globalization-software/37024?camid=4v1a

The STECC Framework - An Architecture for Self-Testable Components
www.igi-global.com/chapter/stecc-framework-architecture-self-testable/30753?camid=4v1a