J. Parallel Distrib. Comput. 69 (2009) 559–572

Contents lists available at ScienceDirect

J. Parallel Distrib. Comput.

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

A proxy-based integrated cache consistency and mobility management scheme for client–server applications in Mobile IP systems

Weiping He *, Ing-Ray Chen

Department of Computer Science, Virginia Tech, Northern Virginia Center, 7054 Haycock Road, Falls Church, VA 22043, United States

ARTICLE INFO

Article history:
Received 1 June 2008
Received in revised form
4 November 2008
Accepted 20 February 2009
Available online 6 March 2009

Keywords:
Mobile IPv6
Cache consistency
Proxy
Regional registration
Integrated mobility and service management
Performance analysis
Petri net

ABSTRACT

In this paper, we investigate a proxy-based integrated cache consistency and mobility management scheme for supporting client–server applications in Mobile IP systems with the objective to minimize the overall network traffic generated. Our cache consistency management scheme is based on a stateful strategy by which cache invalidation messages are asynchronously sent by the server to a mobile host (MH) whenever data objects cached at the MH have been updated. We use a per-user proxy to buffer invalidation messages to allow the MH to disconnect arbitrarily and to reduce the number of uplink requests when the MH is reconnected. Moreover, the user proxy takes the responsibility of mobility management to further reduce the network traffic. We investigate a design by which the MH's proxy serves as a gateway foreign agent (GFA) as in the MIP Regional Registration protocol to keep track of the address of the MH in a region, with the proxy migrating with the MH when the MH crosses a regional area. We identify the optimal regional area size under which the overall network traffic cost, due to cache consistency management, mobility management, and query requests/replies, is minimized. The integrated cache consistency and mobility management scheme is demonstrated to outperform MIPv6, no-proxy and/or no-cache schemes, as well as a decoupled scheme that optimally but separately manages mobility and service activities in Mobile IPv6 environments.

1. Introduction

With the advances of IP-based wireless networks, and the growth in the number of mobile hosts (MHs) carrying wireless devices, it is widely speculated that Mobile IPv6 (MIPv6) [14] will become prevalent in next generation all-IP networks to allow users to maintain service continuity while on the go [19]. A major challenge is to maintain service continuity even if an MH disconnects and then reconnects at will. An MH may disconnect voluntarily simply to reduce the connection cost and/or to avoid power consumption. An MH may also disconnect involuntarily due to handoff or wireless link problems [20].

Two major sources of traffic in MIPv6 systems are mobility management [1,17] and service management [8]. Traditionally, mobility management has been considered separately from service management, as mobility management mainly deals with mobility handoff, location update and location search, while service management mainly deals with data delivery, and application servers can always query the underlying mobility management system to know the location of the MH. Over the years, many micro-mobility protocols extending MIP have been proposed with the goal to minimize mobility management overheads, including MIP Regional Registration [10,18] HMIPv6 [23], Cellular IP [25], IDMP [16], and HAWAII [21].

In this paper we consider integrated mobility and cache management in the context of MIPv6 environments, considering *cache management* as a form of service management. In particular, we propose a Proxy-based Integrated Cache and Mobility Management (PICMM) scheme in MIPv6 networks to support mobile client–server database applications in which the MH queries the server for dynamic data. For example, an MH may query dynamic data such as stock prices, dynamic web pages, weather reports, or traffic information. To avoid sending a query to the server and receiving a reply through the expensive and often unreliable wireless communication network, an MH can cache data objects on the local storage and then answer queries for data that are up-to-date. Caching reduces the server access cost and improves the user-perceived response time [12]. Cache management for such mobile client–server database applications essentially is service management since it deals with efficient data delivery issues.

To process user queries correctly based on caching, an MH must ensure that its cached data are up-to-date. Our integrated cache consistency and mobility management scheme proposed in this paper is based on the stateful strategy [24,27] by which a cache invalidation message is asynchronously sent by the server to the MH, whenever there is an update to a data object cached at the

* Corresponding author.
E-mail addresses: weiping@vt.edu (W. He), irchen@vt.edu (I.-R. Chen).

0743-7315/$ – see front matter © 2009 Elsevier Inc. All rights reserved.
doi:10.1016/j.jpdc.2009.02.003

MH. The MH uses invalidation reports received to determine the validity of its cache content before answering a query. If a query asks for a data object that has been invalidated, then a request is sent uplink to the sever to ask for a fresh copy of the data object accessed by the query before the query can be answered. Moreover, to support service continuity in cases the MH is disconnected and then reconnected again, we use a per-user proxy to buffer invalidation messages to allow the MH to disconnect arbitrarily and to reduce the number of uplink requests to check the cache status when the MH is reconnected. The generated network traffic cost associated with cache consistency management thus includes the cost of receiving and buffering invalidation messages at the proxy and the cost of forwarding them from the proxy to the MH, as well as the cost of sending requests to the server and receiving responses in case cached data objects are not up-to-date to answer queries. This cost is considered as part of the "service" management cost which we like to minimize. Here we note that a large body of research in the literature on cache invalidation for mobile environments is based on the *stateless* strategy [31,13, 5,30] by which the server has no knowledge of cache contents of MHs and will broadcast information on data objects that have been updated either periodically or asynchronously. However, the stateless strategy is not suitable for deployment over large wireless networks where users can roam from one network to another.

Our approach utilizes a per-user proxy to keep invalidation reports of the MH's cache. Whenever the proxy moves it informs the application servers of its whereabout, so invalidation reports can be sent directly from the application servers to the proxy. To minimize both the "service" and "mobility" network traffic costs, the proxy furthermore takes the responsibility of mobility management for integrated mobility and cache management. We investigate a design by which the MH's proxy serves as the MH's gateway foreign agent (GFA) as in the MIP Regional Registration micro-mobility protocol [10,18] to keep track of the address of the MH in a region. The proxy migrates with the MH when the MH crosses a regional area. We aim to identify the *optimal* regional area size under which the overall network traffic generated due to cache consistency management, mobility management, and query requests/replies is minimized. The idea can be extended to other micro-mobility management protocols such as HMIPv6 [23], Cellular IP [25], IDMP [16], and HAWAII [21], although in this paper we only consider the Regional Registration protocol.

The basic idea of our approach is that we use a client-side proxy to support caching and mobility management in Mobile IPv6. The proxy has three functions: (1) working as a GFA as in regional registration to keep track of the MH's location; (2) acting as a service proxy for services engaged by the MH; (3) allocating an extended cache space to store service context information including cache invalidation reports with timestamps for each MH. The proxy will receive invalidation reports from the server on behalf of the MH. If the MH is connected, the proxy will forward the invalidation report to the MH and then discard the invalidation report. If the MH is disconnected, the proxy will store invalidation reports in the proxy's extended cache. Once the MH is reconnected, the MH will get the latest invalidation reports from the proxy. The benefit of the proxy-based scheme lies in the fact that the proxy as an integral part of mobility management knows the MH's location information at all times and thus can efficiently manage the MH's cached data and perform data delivery on behalf of server applications.

The client-side proxy is created when the MH starts in MIPv6, acting as a GFA for the MH. We note that a cross-layer design such as MobileMAN [3] could be used for the implementation of the client-side proxy to run on the access router (AR). The cross-layer design has two components. The first component is at the network layer dealing with mobility management. The second component is at the application layer dealing with service management such as storing invalidation reports or service context information in its extended cache. We also note that Proxy Mobile IPv6 (PMIPv6) [15,9], a recent mobility management protocol for all-IP mobile networks, also proposes to run proxies on ARs for network-based mobility management. However only mobility management is addressed. The security issue of deploying proxies to run on ARs can be handled in a similar way as in PMIPv6. Finally, modern ARs are very powerful devices with ample memory space to run our proxies with extended cache functions. As the proxies of MHs (and hence their GFAs) will move from time to time as they cross regional areas, the load on ARs will be distributed according to the MHs' mobility patterns. In the case of random mobility, the load would spread out to ARs evenly.

The contributions of the paper are (1) the notion of integrated mobility and cache management to minimize the overall network traffic cost for supporting mobile client–server query applications in future MIPv6 systems; (2) identifying the optimal proxy setting including the region area size that will minimize the overall network traffic generated due to mobility and cache consistency management; and (3) demonstrating the benefit of integrated mobility and cache consistency management in MIPv6 compared with basic MIPv6, no-proxy and/or no-caching schemes, as well as a decoupled scheme under which mobility management and cache management are separate but optimally run. This work extends from our earlier preliminary work [11] to cover a comprehensive comparative analysis with existing cache invalidation and mobility management approaches in MIPv6 environments. The extension includes identifying conditions under which PICMM performs better than existing schemes, comparing PICMM with a decoupled scheme that optimally but separately manages cache and mobility activities, and analyzing the optimal regional area size as a function of a mobile user's sleep pattern and the server database size. We note that the use of per-user proxies for integrated mobility and service management was discussed in [6]. This work differs from [6] in that it specifically addresses cache consistency issues in query-based client–server mobile applications in MIPv6 environments. By integrating cache consistency management, query management and mobility management based on stateless cache invalidation protocols, it supports frequent MH disconnections and achieves cost minimization.

The rest of the paper is organized as follows. Section 2 describes the system model and mobile database applications, Section 3 describes the proposed integrated cache and mobility management scheme. Section 4 develops a performance model to analyze the cost incurred under the proposed scheme. Section 5 identifies optimal conditions under which our proposed proxy-based algorithm performs the best in terms of network traffic minimization. We compare the performance of our proposed scheme with no-proxy and/or no-caching schemes, as well as with a decoupled scheme that separately manages mobility and service activities, to demonstrate the benefit of integrated cache and mobility management in Mobile IPv6 systems. Finally, Section 6 summarizes the paper and discusses the applicability.

2. Preliminaries

2.1. Querying data objects in mobile client–server applications

We consider mobile client–server applications in Mobile IPv6 environments in which mobile clients query database servers for data objects. To speed up query processing, a client stores a copy of frequently used data objects in its local cache. When a query is issued by the client, the validity of cached data objects accessed by the query is checked first. If these data objects are valid, then the query is immediately processed with very little access time. Otherwise, a fresh copy of invalid data objects will be retrieved to the local store before the query is answered.