Efficient Mobile Content Delivery by Exploiting User Interest Correlation

Tao Wu, Sadhna Ahuja and Sudhir Dixit
Nokia Research Center
5 Wayside Road, Burlington, MA 01803 USA
{tao.wu, sadhna.ahuja, sudhir.dixit}@nokia.com

Abstract

Accelerating multimedia content delivery to mobile devices is challenging because we must not only address long and unpredictable latency created by the error-prone air interface, but also meet mobility-specific requirements such as efficient terminal power consumption. In this paper, we present a novel Architecture for Content delivery in the Mobile Environment (ACME) that significantly accelerates interactive mobile services such as web browsing. ACME exploits the key concept of user interest correlation (the interest in the same content among different users) to push content to mobile users efficiently. Trace-driven simulations show that ACME achieves 50% effectiveness of content broadcasting by, on average, pushing content to only 0.7% to 6% of all users, thus substantially reducing mobile terminal power consumption. Furthermore, ACME has superior scalability as the push group size increases by only 62% when the total number of users increases by 16 times.

1 Introduction

Mobile multimedia content is emerging as one of the most compelling services in 2.5G and 3G mobile networks as mobile users begin to consume rich content such as mobility-optimized but full-featured World Wide Web and multimedia messaging. A key success factor for mobile content services is that they must achieve satisfactory user experience. A positive user experience not only consists of sufficient QoS for real-time and streaming media applications, but also includes performance requirements for interactive applications such as web browsing and multimedia content discovery. In these interactive services, minimizing end-to-end latency is key to maintain good interactivity and overall user experience.

However, achieving low latency in interactive mobile content services is uniquely challenging due to the fundamental limitations of mobile networks. Most significantly, wireless networks have error-prone air interface and consequently, high frame corruption rate [4] [3]. Link-layer retransmissions of corrupted frames are usually necessary to provide the reliability that most data applications require. These procedures, however, result in long and unpredictable latency and severely affect application interactivity and user experience. In practice, the round-trip time in a GPRS network is close to one second [5], and it has been shown that the PING round-trip time from a terminal to a web server increased by more than 20 times when a wireline link was replaced by an emulated heavily used GPRS link [3]. Generally, the air interface is the bottleneck of performance and user experience in mobile services.

In this paper, we use content delivery techniques to address these performance issues in mobile networks. Content delivery techniques have been quite effective in scaling the Internet backbone to accommodate the rapid growth of the web traffic. In the first few years of the web’s existence, the Internet backbone (especially ISP peering points) and origin servers were the bottleneck of performance. By utilizing web caches at the “edge” of the network to store and serve popular web content, as illustrated in Figure 1(a), content delivery techniques reduce the usage of the backbone and significantly enhance user experience.

However, directly applying wireline content delivery techniques in mobile networks would provide only limited QoS improvement. In contrast to the wireline Internet, the performance bottleneck of mobile web applications is at the air interface, as discussed above (and shown in Figure 1(b)). As a result, user-perceived latency cannot be effectively improved even if content is cached and served from the access network (the “edge” in traditional content delivery). Recognizing this fundamental difference, we implement the edge caching functionality in ACME Caches in terminals, as shown in Figure 2. This approach minimizes the use of the air interface bottleneck and substantially reduces user perceived latency.

Because edge caching is a superset of client caching and stores popular content requested by all users, it is necessary for ACME to push non-requested content to terminals.
However, implementing full-fledged edge caching functionality in terminals would require pushing or broadcasting each requested web object to all other terminals, as done in satellite networks [7]. For mobile terminals, this approach would consume excessive terminal power and storage resources. Toward this end, we make the key observations that user interest correlation, or common interest in the same content among different users, is the foundation of edge caching effectiveness, and that we only need to push content to users who share the same interest with the current requesting user. The ACME Director, shown in Figure 2, quantifies user interest correlation using conditional access probability, and uses this information to selectively push the web object being requested by one user to other users who are most likely to request it in the future. In our trace-driven simulations, we find that to achieve 50% Efficiency of full edge caching, ACME only needs to push content to 0.7% to 6% of all users, substantially reducing average terminal power consumption. Furthermore, ACME has superior scalability because the push group size increases by only 62% when the total number of users increases by 16 times. ACME provides a scalable and mobility-optimized system that effectively improves mobile user experience.

2 Using User Interest Correlation for Efficient Mobile Content Delivery

As explained in Introduction, implementing full edge caching at terminals would require pushing every requested object to all terminals. If there are $N$ terminals, then ACME’s average terminal processing, power consumption and storage costs would be $O(N)$. This is generally too expensive and non-scalable for mobile terminals. The key idea of ACME is to push content only to users who are most likely to request the same content after the current requester based on user interest correlation. In this section, we present the ACME Director’s user interest correlation algorithm and evaluate its performance using trace-driven simulations.

2.1 The ACME Director Algorithm

We use conditional probability to measure user interest correlation. For $N$ users, let $P$ be the $N$-by-$N$ matrix such that $P_{i,j} = p(j|i)$, where $p(j|i)$ denotes the probability that user $j$ will access a web object in the future, given user $i$ has accessed that object. $P_{i,i}$ is defined to be zero because this case is handled by conventional client caching and is irrelevant to ACME. In practice, $P$ can be trained using available web access traces, as done in Section 2.2. For a certain web object encountered in the training process, if user $i$ accesses it before user $j$, then $p(j|i)$ is increased accordingly. In addition, we need to make sure that after $P$ is trained, it should be normalized such that $\sum_{j=1}^{N} p(j|i) = 1$, $i = 1, 2, \ldots, N$, which has obvious probabilistic interpretation.

Once the ACME Director has a properly trained $P$ matrix, it performs selective push as follows. First, we use a tunable parameter called the “selectiveness factor” $\alpha \ (0 \leq \alpha \leq 1)$.
$\alpha \leq 1$ to control push “selectiveness”. Suppose user $i$ requests a web object, and $p(j|i)$ is among the largest $\alpha N^2$ elements in $P$, then the ACME Director pushes the content to user $j$. Clearly, $\alpha = 0$ represents the case of conventional client caching, because no other user receives the content. On the other hand, $\alpha = 1$ represents the case of full edge caching, i.e., every terminal receives a copy of the requested content. The design goal of the ACME Director is to achieve high hit ratio (close to that of full edge caching) with a small $\alpha$ to reduce terminal power consumption. We note that in practice, content pushing can take place asynchronously with the corresponding request. For example, it is straightforward to design a system that aggregates content to be pushed to individual terminals and performs pushing either periodically or when content to be pushed exceeds a size threshold.

2.2 Performance Evaluation

We now perform trace-driven simulations to evaluate ACME’s performance, especially the ACME Director’s ability to accurately push content to relevant users.

2.2.1 Traces

We first describe the traces we use in the simulations. Due to the lack of publicly available mobile web traces, we use three wireline proxy traces [6] for simulations and exclude all dynamically generated and non-cacheable responses. The UCB trace is the first 300,000 accesses in UC Berkeley’s home IP access, representing about 17 hours of activity from 2155 IP addresses. The BU trace is the concatenated access log of Room B19 in Boston University’s computer science department in March and April 1995, with a total of 553 users and 437,861 web accesses. The NLANR trace is the access log of the pb node of the National Laboratory for Advanced Network Research on November 13, 2000, with a total number of 127 user IP addresses and 878,085 web accesses.

2.2.2 Simulations

To quantitatively evaluate ACME’s performance, we note that its hit ratio is upper-bounded by full edge caching and lower-bounded by client caching. The hit ratio is also dependent on $\alpha$. So we define the Director Effectiveness, $E(\alpha)$, as

$$E(\alpha) = \frac{H_D(\alpha) - H_C}{H_F - H_C}$$

where $H_D(\alpha)$ is the average ACME Cache hit ratio when the selectiveness factor is $\alpha$, and $H_F$ and $H_C$ are the average hit ratios for full edge caching and client caching, respectively. Obviously, we have $0 \leq E(\alpha) \leq 1$, with $E(0) = 0$ and $E(1) = 1$. This normalized performance metric allows us to compare the ACME Director’s performance between different traces.

We define the push group size $S(\alpha)$ as the average number of terminals that receive pushed web objects when the selectiveness factor is $\alpha$. Because average terminal power consumption is proportional to $S(\alpha)$, we use $S(\alpha)$ to represent average terminal power consumption. Again, to compare the group size between different traces, we define relative push group size, $s(\alpha)$, as

$$s(\alpha) = \frac{S(\alpha)}{S(1)}$$

So $s(\alpha)$ is the push group size relative to that of full edge caching.

Now we run simulations driven by the traces described above to determine $E(\alpha)$, $S(\alpha)$ and $s(\alpha)$. We partition each trace into a training part (odd-numbered accesses) and a test part (even-numbered accesses). After training $P$ using the training part of each trace, we run the test part in an ACME Director simulator and measure the average ACME Cache hit ratios under different selectiveness factors. $E(\alpha)$ is then computed based on $H_D(\alpha)$, $H_F$ and $H_C$. We obtain $S(\alpha)$ by counting the number of terminals receiving the pushed object, for all requested objects, and compute $s(\alpha)$ by normalization.

Figure 3 depicts Director Effectiveness $E(\alpha)$ against relative push group size $s(\alpha)$ by varying $\alpha$ from 0 to 1 for all three traces. ACME’s effectiveness is evident from the figure, as $E(\alpha)$ increases rapidly with very small group size. In all three traces, a Director Effectiveness of 80% is achieved at a relative push group size smaller than 20%. Furthermore, as shown in Table 1, ACME Director achieves
50% Effectiveness when the relative push group size is as small as 0.7% (UCB trace) to 6% (NLANR trace) of the full edge caching push group size $S(1)$. The push group size ranges from 6.74 (NLANR trace) to 10.96 (UCB trace). The reduction in average terminal power consumption is calculated as the inverse of the relative push group size, so the ACME Director reduces the terminal power consumption by 17 (NLANR trace) to 144 times (UCB trace) while maintaining 50% Effectiveness.

Another interesting finding from our simulations is that for a wide range of Director Effectiveness values, the relative push group size decreases with the increasing number of users among traces. This indicates that the ACME push group size is much less sensitive than the full edge caching push group size, to the vastly different number of users in the three traces. Indeed, at 50% Effectiveness, the push group size increases by only 62%, from 6.74 to 10.96, as $N$ increases by 16 times from 127 in the NLANR trace to 2155 in the UCB trace. This is probably because for a given user, her interest in content can be best approximated by a small group of users whose interests are closest to hers. The size of this interest group grows much slower than the general user group because it only keeps the users with the closest interest in the general user group. More broadly, we note that user interest correlation has been applied in a variety of applications, including peer discovery in P2P networks [8] and making add-on recommendations to consumers based on other people’s purchase patterns in e-business [2]. This may suggest that user interest correlation may be at the gist of a group of human-centric applications, and better understanding of the concept would be important to enhance these applications.

Finally, we note that ACME’s performance gain is incremental and transparent. While having more memory always increases hit ratio, resource-constrained terminals with a little available storage benefit most from ACME’s concave hit ratio curve in Figure 3.

### 3 Implementation Issues

ACME is a versatile architecture and can be implemented in a variety of networks. It can utilize radio multicast (such as MBMS in 3GPP [1]) for efficient delivery of popular content, if such capability exists. In other networks, ACME can utilize existing push mechanisms to implement edge caching. The split proxy architecture of ACME makes it possible to implement optimal transport protocols between the ACME Director and ACME caches, and provides full transparency to web applications at the origin server and the clients.

### 4 Conclusions

In this paper, we developed ACME, an efficient content delivery architecture for mobile networks, which extends the functionality of edge caching to the terminal using selective content push. ACME effectively addresses several challenges in enhancing user experience in mobile networks, such as excessive over-the-air latency and stringent requirements on terminal power consumption. We identified that the effectiveness of web caching lies in user interest correlation, and exploited this fact to selectively push content only to users who share the same interest. ACME also exhibits superior scalability in trace-driven simulations, as the push group size grows much slower than the total number of users.

### References