

Multiuser Prefetching with Queuing Prioritization in Heterogeneous Wireless Systems

Ben Liang and Stephen Drew
Department of Electrical and Computer Engineering
University of Toronto
10 King's College Road
Toronto, Ontario, M5S 3G4, Canada
{liang, drews}@comm.utoronto.ca

Abstract

We study the performance of a multi-user prefetching strategy in a two-tier heterogeneous wireless network. A predictive framework was previously introduced for mobility-aware document prefetching to enhance the experience of a mobile user roaming between heterogeneous wireless access networks. However, an undesirable effect of multiple prefetching users is the potential for system instability due to the racing behavior between document access delay and user prefetch quantity. This phenomenon is particularly acute in the heterogeneous environment. We propose to alleviate the system traffic load through optimizing a prefetch thresholding algorithm, accounting for server queuing prioritization. We evaluate the performance of the proposed algorithm through numerical analysis and simulation. We show that stability can be maintained even under heavy usage, providing both the same scalability as a non-prefetching system and the performance gains associated with prefetching.

Keywords: mobile prefetching, heterogeneous wireless networks, WLAN/3G integration, performance modelling, queuing analysis

1 Introduction

The future wireless information system will consist of heterogeneous radio access networks, including wide-area cellular networks, wireless metropolitan area networks (WMANs), wireless local area networks (WLANs), and infrastructure-less wireless networks [3]. Since no single access technology meets the ideal of high bandwidth, universal availability, and low cost, they should be strategically integrated to provide optimal services. In such heterogeneous systems, a mobile device roaming across differ-

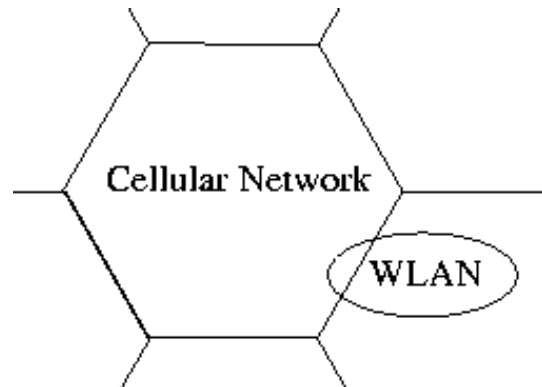


Figure 1. A two-tier wireless heterogeneous network.

ent access networks should dynamically adapt and make intelligent choices to balance the trade-offs between various performance factors [14]. In this work, we study network aware document prefetching by a mobile device in a two-tier wireless system comprised of a universal basic coverage network, and within it a preferred high speed network with lower access cost but limited coverage. Throughout this paper, we use wide-area cellular and WLAN as examples for these two networks, respectively, as shown in Fig. 1.

Prefetching is a technique in which the client device proactively fetches from the server documents that are predicted to be accessed in the near future. For example, the Mozilla-based Web browsers [1] have support for a prefetch tag, which allows a Web page to specify the subsequent documents that should be automatically fetched by the browser. The main benefit of prefetching is that it can reduce the user perceived access delay to the much shorter time of a cache lookup. Prefetching is most effective when there exist items that will be accessed with high probability, and there are de-

lays or down-times between consecutive access requests.

There exists much research on Web document prediction and prefetching [15, 11, 12, 5, 2, 7, 4]. Most of the proposed methods make use of user histories to make informed predictions. Traditional Web caching systems without prefetching have been shown to achieve a maximum hit rate of around 40% to 50% on static Web pages, whereas aggressive prefetching schemes can increase the hit rate to the order of 80%. In addition, we have previously examined the benefit of prefetching using Web browsing as an example and showed that mobility-awareness can lead to significant performance gain [9]. However, the major side effect of Web prefetching is a considerable increase in traffic due to prefetching stale documents that will not be accessed [6]. Furthermore, in the wireless environment, prefetching decisions may be constrained by device power [10, 19] or storage capacity [18].

In this work, we observe that prefetching is particularly valuable for users in a two-tier wireless network because the cost of access in WLAN is generally much less expensive than the surrounding cellular network. A successful prefetch when the user is about to leave the coverage area of the WLAN displaces a potential cellular network access with a cheaper WLAN access. Network awareness is thus built into the prefetching decision process.

We extend the previous results in [9] and examine the effect of multiple prefetching users on system performance. A unique challenge that arises in the multi-user scenario is the feedback of prefetching strategy amongst the users. When multiple users are competing for the available bandwidth, each user may have to wait much longer for its request to be serviced, and will adjust its prefetching strategy accordingly. This introduces more requests to the server, and further increases the time to service requests. The increase in traffic delays due to prefetching is well known [6], but in the two-tier network it results in a more significant problem because of the increased level of prefetching.

We propose a novel analysis framework to evaluate and optimize the performance of prefetching in a two-tier network. Furthermore, the analysis framework accounts for queuing prioritization and reneging at the document server, allowing service differentiation between regular and prefetching requests. The amount of prefetching, or the aggressiveness of the algorithm, can be adjusted by each user based on their current mobility and application characteristics. Other metrics that are used in the prefetching decision process include the network bandwidth, data access costs, and the user perceived value of time. We develop the analysis framework using these QoS metrics such that the effects of any mobility pattern, network topology, or access pattern could be used as inputs into optimal network-aware prefetching over heterogeneous networks.

The rest of this paper is organized as follows. In Section

2 we describe the system model and user prefetching strategy. In Section 3 we present a recursive queuing analysis framework to evaluate prefetching performance. In Section 4 numerical and simulation results are presented. Finally, conclusions are given in Section 5.

2 Multi-user Network Aware Prefetching

In this section, we describe the system model for multi-user network aware prefetching and present the user prefetching strategy.

2.1 Network Model and Document Access

We consider mobile users in a two-tier network, comprised of WLANs surrounded by a ubiquitous cellular network. Users may roam anywhere and are not constrained to any one network. Users are mobility-aware, such that they have an estimate of which networks they may roam to in the near future [9, 13]. Users access documents, and are provided with a mechanism to estimate the access probabilities, p_a , for their next set of possible documents [15, 11, 12, 5, 2, 7, 4].

Prefetch requests are sent to a central server while users are reading their current page, and the prefetched documents are placed in a cache on the mobile device. Each new user request is served by first examining the cache for a successful prefetch, and if none is found, a normal document request is sent to the server. It was shown in [9] that gains from prefetching within the cellular network are minimal, and thus we only consider the case of prefetching from within the WLAN. Each user will establish a *prefetching threshold*, H , and will prefetch in a single batch request all documents with access probabilities greater than the threshold (i.e., $p_a > H$).

The central server is modelled as a queue servicing requests from all users in the system. Since normal requests are more time sensitive, while prefetch documents can be returned any time within the inter-request interval, we study a two-priority system where web documents are given high priority (HP), and all prefetch documents are given low priority (LP). We choose a preemptive resume system [16], which operates so that when an HP request arrives while an LP request is in service, it is serviced immediately and causes the LP request to be preempted to the front of its waiting queue. The LP request returns to service only when all HP requests have been serviced, and resumes from where it left off. The preemptive resume model fits well with a packet-based system.

Furthermore, the server queue supports reneging. The HP and LP requests are dropped from the server when the user departs from the WLAN. It is also reasonable to purge the stale prefetch requests. When a user submits a new HP

or LP batch request, any prior LP requests by the same user are deemed stale and renege.

2.2 Prefetching Strategy

We define the cost of access as the sum of access cost and the penalty for access delay. Then, the prefetching threshold is based on a decision function that compares the expected costs of requesting and not requesting to prefetch a document with access probability p_a , denoted c_p and c_{np} , respectively.

If a document is not prefetched and if it is indeed requested by the user at time t , or if the document is not prefetched in time, then the user's document request will be forwarded to the server with high priority. The expected cost of this is

$$c_{HP}(t) = (1 - F_{t_w}(t)) (\alpha_{BW}s + \alpha_T E[S_{HP}]) + F_{t_w}(t) (\alpha_{BC}s + \alpha_T s/b_C), \quad (1)$$

where t_w denotes the user's residual WLAN residence time and $F_{t_w}(t)$ its cdf, α_{BW} and α_{BC} denote the price per byte of access to the WLAN and to the cellular network¹, respectively, s denotes the average document size, b_C denotes the constant bit rate provided by the cellular network, α_T denotes the cost of lost time, where lost time is defined as the duration in which a user is waiting for the server to service a HP request, and S_{HP} denotes the HP request sojourn time in the server queue, given by

$$S_{HP} = \min(S_{HP}^0, t_w), \quad (2)$$

where S_{HP}^0 is the *touched* sojourn time of a HP request if there were no renegeing due to the user moving out of the WLAN. Note that α_{BW} and α_{BC} may represent both the monetary cost paid to the service providers and the energy cost of wireless access. Hence, we have

$$c_{np} = \int_0^\infty f_{t_r}(t) p_a c_{HP}(t) dt, \quad (3)$$

where t_r denotes the time for the user to request the next document and $f_{t_r}(t)$ its pdf.

The value of c_p depends on whether the prefetch request is successfully serviced. Let S_{LP}^0 be the *untouched* sojourn time of a LP request if there were no renegeing due to a new request by the user or the user moving out of the WLAN, so that the LP request sojourn time is given by

$$S_{LP} = \min(S_{LP}^0, \min(t_r, t_w)). \quad (4)$$

¹These prices may account for both the monetary cost charged by the service provider and the communication energy cost to the mobile device.

Then, we have

$$c_p = \int_0^\infty f_{t_r}(t) \left(P\{S_{LP}^0 > t\} p_a c_{HP}(t) + P\{S_{LP}^0 < \min(t, t_w)\} \alpha_{BW}s + P\{t_w < S_{LP}^0 < t\} p_a (\alpha_{BC}s + \alpha_T s/b_C) \right) dt. \quad (5)$$

The optimal prefetching threshold is determined by

$$H = \min\{p_a | c_p < c_{np}\}. \quad (6)$$

Clearly, it depends on the network characteristics and user access patterns. Next, we present an analytical framework to evaluate the effect of various system parameters on the optimization of prefetching.

3 Performance Analysis Framework

This section provides an analytical framework for multi-user network aware prefetching. We present a method to compute the distributions of S_{HP}^0 and S_{LP}^0 , which in turn depend on the prefetching quantity by the other users and hence H . Then, a recursive procedure can be used to approach the optimal prefetching threshold. This recursion converges as long as the feedback generated from an increase in traffic is less than the increase in traffic [8]. For most traffic loads in our numerical analysis, the system converges after a few iterations.

3.1 Steady State Server Queue Distribution

The state of the preemptive resume priority queue can be described by the doublet of state variables ($\#LP, \#HP$). We first determine the arrival rates of different types of requests. To obtain tractable analysis results, we assume that the user's document inter-request time, t_r , and the time for the server to transmit a document, t_s , are both exponential, with rates λ and μ , respectively. We further assume the WLAN residence time, t_w , of a user is exponential with rate γ . This is a common model for cell residence time in the literature. Later, in the simulation section, we demonstrate that the computed optimal prefetching threshold provides a close approximation even for non-memoryless document access patterns.

We consider the queue state only at instants when a document is viewed, and hence an HP request and its associated LP requests can be regarded as one batch request. If the last batch request occurred in the WLAN, and the request was successful in prefetching the next document viewed by the user, the user will generate a new batch of only prefetch requests. If the last batch request was unsuccessful, or was from a different network, then the current batch must contain an HP request for the document the user wishes to view.

Therefore, each batch contains a variable amount of LP requests and one or zero HP requests. To find the probability that a batch request contains k LP documents, denoted $P_L[k]$, we count the number of documents with access probability exceeding the prefetching threshold:

$$P_L[k] = \sum_a P\{\delta = a\} \cdot P\left\{ \sum_{x \in x_D} 1(p_a(x) > H) = k | \delta = a \right\}, \quad (7)$$

where δ represents all document access probability distributions and x_D represents all possible documents to be requested.

To simplify analysis, we further assume that if one prefetch request from a batch is dropped, then all of the requests are dropped, and similarly if one prefetch request is returned, then all of the requests are returned. This assumption is reasonable because most of the time the LP requests are served in quick succession. We ignore this assumption in our simulation, such that individual documents within a batch can be either dropped or received. As can be seen later, this approximation is acceptable, and it significantly reduces the analysis complexity.

We first consider request batches with no HP request. This is possible only if the previous LP requests were not dropped and they include the document actually intended by the user. Thus, the arrival rate of request batches that cause a queue state net movement of $(k, 0)$, where $0 \leq k \leq |x_D|$, given the current state (j, n) , is

$$\lambda_{k,0|j,n} = \lambda p_W P_L[k] \sum_{i=1}^{|x_D|} P_L[i] P\{C|i\} (1 - P\{D_L|j,n\}), \quad (8)$$

where p_W is the probability that the inter-request time is less than the WLAN residence time, i.e.,

$$p_W = P\{t_r < t_w\} = \frac{\lambda}{\lambda + \gamma}, \quad (9)$$

$P\{C|k\}$ is the sum of access probabilities in the last batch of k LP requests, and $P\{D_L|j,n\}$ is the probability that the last batch of LP requests are dropped due to staleness given current queue state (j, n) . When j is not too small, $P\{D_L|j,n\}$ can be approximated by the probability that any batch of LP requests are dropped due to staleness, i.e.,

$$P\{D|j,n\} = P\{t_r < S_{LP}^0|j,n\}. \quad (10)$$

Otherwise, some normalization may be necessary.

Batch requests with one HP request can be due to three possible outcomes of the previous LP batch request: they were dropped due to staleness, they were dropped due to user moving out of WLAN, or they were received but did

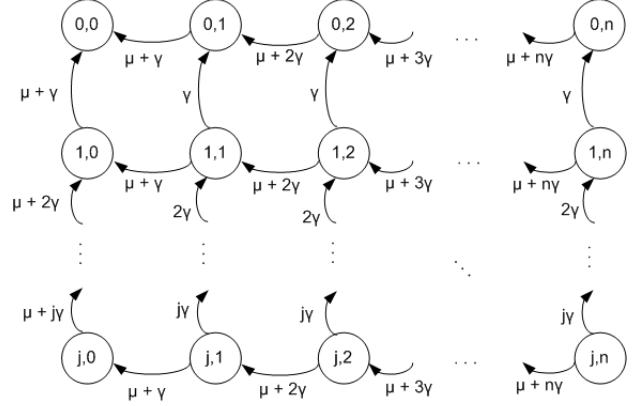


Figure 2. Priority service rate model.

not include a successful prefetch. The probability for the last case, or the probability the last prefetch batch resulted in a miss, is

$$P\{M|j,n\} = P_L\{0\} + \sum_{k=1}^{|x_D|} P_L\{k\} (1 - P\{C|k\}) (1 - P\{D_L|j,n\}). \quad (11)$$

For there to be a net movement of $k > 0$ LP requests and one HP request in the queue, there is either a batch of k LP requests inducing no LP dropping, or a batch of $i \leq k + 1$ LP requests inducing $i - k$ dropped LP requests. Hence,

$$\lambda_{k,1|j,n} = \lambda P_L[k] \left(p_W P\{M|j,n\} + 1 - p_W \right) + \lambda p_W \sum_{i=k+1}^{|x_D|} P_L[i] P_L[i - k] P\{D|j,n\}. \quad (12)$$

For the net movement of LP requests to be less than zero, there must be dropped LP requests. Thus, we have the following for $0 < k \leq |x_D|$:

$$\lambda_{-k,1|j,n} = \lambda p_W \sum_{i=0}^{|x_D|-k} P_L[i] P_L[i + k] P\{D|j,n\}. \quad (13)$$

The arrival rates in (8), (12), and (13) are then combined with service rates for the preemptive-resume priority model, as shown in Fig. 2, to define a continuous-time Markov chain that represents the server queue. This Markov chain is clearly ergodic. We denote $u_{j,n}$ the steady state distribution of this Markov chain. We further define λ' as the sum of all outgoing rates from a specific state

$$\lambda'_{j,n} = \sum_{k=1}^{|x_D|} \lambda_{k,0|j,n} + \sum_{k=0}^{|x_D|} \lambda_{k,1|j,n} + \sum_{k=1}^{\min(|x_D|-1,j)} \lambda_{-k,1|j,n}. \quad (14)$$

Then, the following balance equations can be used to numerically compute $u_{j,n}$ [17]:

$$\begin{aligned}
\lambda'_{0,0}u_{0,0} &= (\mu + \gamma)u_{1,0} + (\mu + \gamma)u_{0,1} \\
(\lambda'_{0,n} + \mu + n\gamma)u_{0,n} &= (\mu + (n+1)\gamma)u_{0,n+1} + n\gamma u_{1,n} \\
&\quad + \lambda_{0,1|0,n-1}u_{0,n-1} + \sum_{k=1}^{|x_D|-1} \lambda_{-k,1|k,n-1}u_{k,n-1} \\
(\lambda'_{j,0} + \mu + j\gamma)u_{j,0} &= (\mu + (j+1)\gamma)u_{j+1,0} \\
&\quad + (\mu + \gamma)u_{j,1} + \sum_{k=1}^{\min(|x_D|,j)} \lambda_{k,0|j-k,0}u_{j-k,0} \\
(\lambda'_{j,n} + \mu + (j+n)\gamma)u_{j,n} &= (\mu + (n+1)\gamma)u_{j,n+1} \\
&\quad + (j+1)\gamma u_{j+1,n} + \sum_{k=1}^{\min(|x_D|,j)} \lambda_{k,0|j-k,n}u_{j-k,n} \\
&\quad + \sum_{k=0}^{\min(|x_D|,j)} \lambda_{k,1|j-k,n-1}u_{j-k,n-1} \\
&\quad + \sum_{k=1}^{x_D-1} \lambda_{-k,1|j+k,n-1}u_{j+k,n-1}.
\end{aligned} \tag{15}$$

3.2 HP and LP Sojourn Time Distribution

Since the server queue is preemptive, S_{HP} is simply the waiting time of an M/M/1 queue with reneging due to the user moving out of WLAN at rate γ . We first consider the untouched (i.e., not reneged) HP sojourn time S_{HP}^0 . It can be shown that, given n HP request in the queue, the untouched sojourn time of the $(n+1)$ th HP request has distribution [16]

$$f_{S_{HP}^0|n}(x) = \frac{\mu}{\beta_{n+1}(\gamma)} \sum_{i=0}^n \frac{(-1)^i}{i!(n-i)!} e^{-(\mu+i\gamma)x}, \tag{16}$$

where

$$\beta_n(\gamma) = \left[\left(\frac{\mu + \gamma}{\gamma} \right) \left(\frac{\mu + 2\gamma}{\gamma} \right) \dots \left(\frac{\mu + (n-1)\gamma}{\gamma} \right) \right]^{-1}. \tag{17}$$

Then, the touched sojourn time, $S_{HP} = \min(S_{HP}^0, t_w)$, has distribution

$$f_{S_{HP}|n}(x) = \frac{\mu}{\beta_{n+1}(\gamma)} \sum_{i=0}^n \frac{(-1)^i e^{-[\mu+(i+1)\gamma]x}}{i!(n-i)!} \frac{\mu + (i+1)\gamma}{\mu + i\gamma}. \tag{18}$$

The sojourn time of an LP request is the HP busy period with an initial workload of x equal to the total service time

of all HP and LP requests ahead of it in the queue. We label the HP busy period $T_{HP}[x]$. From [16], the relationship between the busy period of an M/M/1 queue initiated by a workload x is

$$E[e^{-\theta T_{HP}[x]}] = E[e^{-x\eta}], \tag{19}$$

where $\eta \equiv \eta(\theta)$ is given by

$$\eta = \frac{\theta + \lambda_{HP} - \mu + \sqrt{(\theta + \lambda_{HP} + \mu)^2 - 4\lambda_{HP}\mu}}{2}, \tag{20}$$

where λ_{HP} is the request rate of HP documents and can be computed by summing over all request rates that include one web document. Neglecting HP reneging due to mobility, the HP requests are served with rate μ . Hence, we have an upper bound LP sojourn time probability distribution, due to n HP requests, in the Laplace domain

$$\bar{f}_{W_{LP}|n}(\theta) = \left(\frac{\mu}{\mu + \eta} \right)^n. \tag{21}$$

Furthermore, similar to (16), we can determine the untouched LP waiting time distribution, due to j existing LP requests in the queue, given by the Laplace transform

$$\bar{f}_{S_{LP}^0|j}(\theta) = \frac{\mu}{\beta_{j+1}(\nu)} \sum_{i=0}^j \frac{(-1)^i}{i!(j-i)!} \frac{1}{\mu + i\nu + \eta}, \tag{22}$$

where $\nu = \lambda + \gamma$ is the rate of LP reneging.

Hence, assuming that the sojourn times of different LP requests in the same batch are the same and that no LP dropping is induced by the new LP requests, we have an approximation to the untouched LP sojourn time distribution given queue state (j, n) , in the Laplace domain,

$$\bar{f}_{S_{LP}^0|j,n}(\theta) = \bar{f}_{S_{LP}^0|j}(\theta) \bar{f}_{W_{LP}|n}(\theta). \tag{23}$$

To improve the above approximation, we can further consider the concurrent dropping of LP requests and the position of the new LP request in the batch request. We assume that in a LP batch of size k , any given LP request will be uniformly distributed among all k possible positions. Then, from an initial state (j, n) , the Laplace domain average distribution for the untouched LP sojourn time is

$$\begin{aligned}
\bar{f}_{\bar{S}_{LP}^0|j,n}(\theta) &= \left(\sum_{k=0}^{|x_D|} f\{k, 1|j, n\} + \sum_{k=1}^{|x_D|-1} f\{-k, 1|j, n\} \right) \\
&\quad \left(\frac{\mu}{\mu + \eta} \right)^{n+1} + \sum_{k=1}^{|x_D|} f\{k, 0|j, n\} \left(\frac{\mu}{\mu + \eta} \right)^n,
\end{aligned} \tag{24}$$

where $f\{k, m|j, n\}$ denotes the weighted Laplace domain average sojourn distribution of a batch request causing net

movement (k, m) given queue state (j, n) , and is computed by

$$f\{k, 0|j, n\} = p_W \left(\frac{P_L\{k\}}{k} \sum_{i=1}^k \bar{f}_{S_{LP}^0|j+i, n}(\theta) \right) + \sum_{i=1}^{|x_D|} P_L[i] P\{C|i\} (1 - P\{D_L|j, n\}), \quad (25)$$

$$f\{k, 1|j, n\} = \left(p_W P\{M|j, n\} + 1 - p_W \right) \left(\frac{P_L[k]}{k} \sum_{i=1}^k \bar{f}_{S_{LP}^0|j+i, n}(\theta) \right) + p_W P\{D|j, n\} + \sum_{i=k+1}^{\min(k+j, |x_D|)} P_L[i-k] \frac{P_L[i]}{i} \sum_{l=1}^i \bar{f}_{S_{LP}^0|j+l-i+k, n}(\theta), \quad (26)$$

$$f\{-k, 1|j, n\} = p_W \sum_{i=1}^{|x_D|-k} P_L[i+k] P\{D|j, n\} + \frac{P_L[i]}{i} \sum_{l=1}^i \bar{f}_{S_{LP}^0|j-k+l-i, n}(\theta). \quad (27)$$

Finally, the distribution of the touched LP sojourn time, $S_{LP} = \min(S_{LP}^0, \min(t_r, t_w))$, can be derived similarly to (18).

3.3 LP Dropping Probability and User Received Traffic

The probability of LP dropping due to staleness, $P\{D|j, n\}$, is a parameter used throughout Section 3.1. It can be computed recursively using the LP sojourn time distribution. With t_r exponential with rate λ , we have

$$P\{D|j, n\} = P\{t_r < \bar{S}_{LP}^0|j, n\} = \int_0^\infty \int_0^s \lambda e^{-\lambda t} f_{\bar{S}_{LP}^0|n}(s) dt ds = 1 - \bar{f}_{\bar{S}_{LP}^0|j, n}(\lambda). \quad (28)$$

The amount of traffic received by users, that is the traffic that successfully gets serviced by the queue, is another important performance metric. HP requests may be dropped due to mobility, i.e., when $S_{HP}^0 > t_w$. Hence, the amount of traffic received per HP request is

$$\rho_{HP} = \sum_{j, n} u_{j, n} P\{S_{HP}^0 > t_w|n\} = \sum_{j, n} u_{j, n} \int_0^\infty \int_0^s \gamma e^{-\gamma t} f_{S_{HP}^0|n}(s) dt ds = \sum_{j, n} u_{j, n} (1 - \bar{f}_{S_{HP}^0|n}(\gamma)). \quad (29)$$

To obtain the total rate of HP traffic received, we multiply ρ_{HP} by the sum of rates for all request that include one HP request.

Similarly, LP requests may be dropped due to either mobility or for staleness, i.e., when $S_{LP} > \min(t_r, t_w)$. Hence, we have the amount of traffic received per LP request

$$\rho_{LP} = \sum_{j, n} u_{j, n} P\{\bar{S}_{LP}^0 > \min(t_r, t_w)|n\} = \sum_{j, n} u_{j, n} \int_0^\infty \int_0^s (\lambda + \gamma) e^{-(\lambda + \gamma)t} f_{\bar{S}_{LP}^0|n}(s) dt ds = \sum_{j, n} u_{j, n} (1 - \bar{f}_{\bar{S}_{LP}^0|j, n}(\lambda + \gamma)). \quad (30)$$

To obtain the total rate of LP traffic received, we multiply ρ_{LP} by the rate of LP requests.

4 Numerical and Simulation Results

A multi-threaded Java based simulation environment has been developed to validate the proposed analysis model. Note that as explained previously, the simplifying assumptions made in Section 3 are not made in simulation.

To represent the various levels of predictability on a user's future document access, we pick the document access probabilities, p_a , uniformly from a set of nine truncated geometric distributions of length 10 and parameter varying from 0.1 to 0.9 with step size 0.1. The intra-request time is has mean 12 seconds and is exponential in the default case. The other default system parameters are $b_W = 100KB/s$, $b_C = 5KB/s$, $\alpha_{B_W} = \$1/MB$, $\alpha_{B_C} = \$0.05/KB$, $s = 10KB$, and $\alpha_T = \$20$ per hour. It was demonstrated in [11] that, in general, a suitable value for the user perceived cost of access delay may be the user's income level.

The simulations were run for various number of users and different values of $p_W = P\{t_r < t_w\}$, the probability that the user remains inside the WLAN at the next document access, which represents the level of user mobility. All data points are taken after the system has reached an equilibrium state.

4.1 Traffic Load

A comparison of the expected HP and LP traffic generated is shown in Fig. 3, with $p_W = 0.8$. The analysis and simulation results are very close, with less than 10% difference in most cases. Furthermore, the amount of LP traffic is much greater than the amount of HP traffic, suggesting aggressive prefetching. The benefit of adaptive optimal prefetching thresholding is seen, as the amount of LP

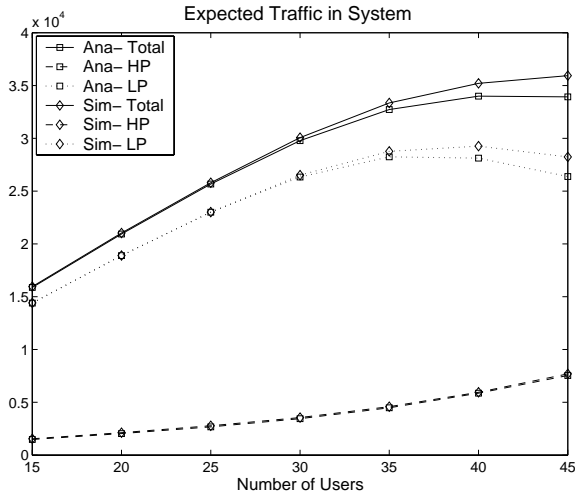


Figure 3. Expected traffic generated in server queue.

traffic peaks at between 40 to 50 users, at the same time as the total traffic at the server approaches capacity. Indeed, the server hovers near capacity even for a much larger user population.

We next present the amount of traffic received per user in Fig. 4. We see that the success ratio shrinks as more users enter the system. The HP traffic gradually increases, showing that less and less of the prefetching requests are resulting in hits.

4.2 Performance Gain and Effect of Mobility

We plot the performance gain of prefetching using the optimal threshold over non-prefetching in Fig. 5, for five different degrees of mobility. The performance gain is defined as the percentage cost reduction from the non-prefetching system. These results suggest that, when the server utilization is below capacity, higher degrees of mobility lead to higher gains. However, when the server is near capacity (e.g., when the number of users is greater than 50), the more aggressive prefetching by users due to higher mobility remains detrimental to system performance.

4.3 Non-Markovian Inter-requests

In Fig. 6, we compare the optimal user prefetching threshold for both Markovian and non-Markovian inter-request durations. Simulation results are obtained with a heavy-tail Pareto inter-request time of index 10 and also with a nearly Gaussian Erlang inter-request time of order 20, both scaled to have mean 12 seconds. They are compared with the analysis and simulation results obtained

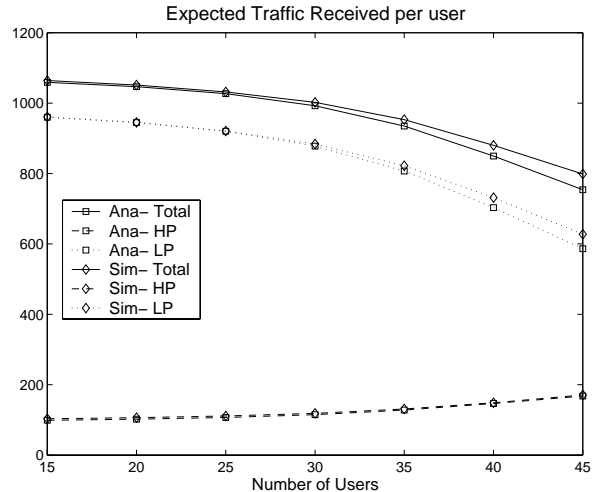


Figure 4. Expected traffic per user.

from exponential inter-request time as in Section 4.1, with $p_W = 0.8$. We observe that the prefetching threshold is almost insensitive to the different inter-request distributions, especially for moderate to low total number of users (e.g., less than 70). Hence, the proposed analysis can be applicable to a wide range of practical systems.

5 Conclusions

Adaptive document access strategies are necessary in future wireless systems where heterogeneous access technologies will be seamlessly integrated. Document prefetching can significantly improve the performance of such integrated systems, but it needs to be carefully designed, taking into consideration its effect on the system traffic load when multiple users are present. In this paper, we have proposed a novel analysis framework toward optimal document prefetching over a two-tier network with priority queuing.

Through numerical and simulation studies using typical parameter values, we demonstrate that, with optimal control of the prefetching threshold, multi-user prefetching can be scalable and operate well under heavy usage with many concurrent users. The proposed network-aware prefetching gives the users faster response time, and the service providers the revenue of increased activity without loss of service due to instability. Our experimental results further demonstrate that the proposed analysis can be used to evaluate the performance and provide optimization guidelines for systems with non-Markovian access patterns.

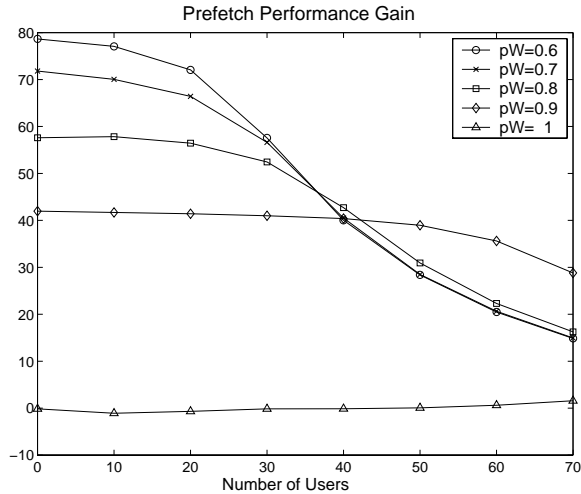


Figure 5. Performance gain for various degrees of mobility.

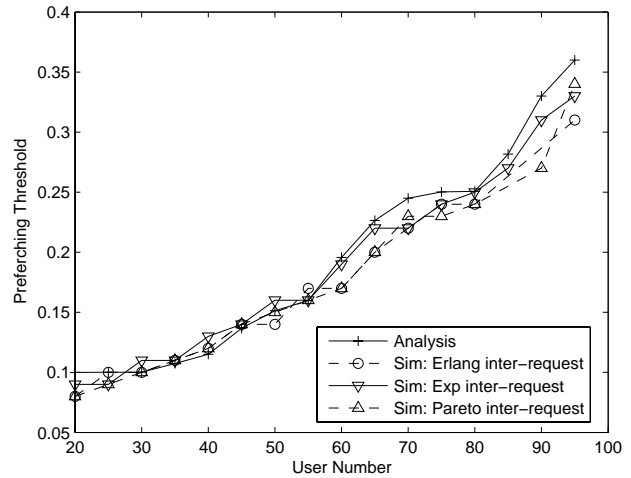


Figure 6. Prefetching threshold for different inter-request distributions.

Acknowledgments

The authors would like to thank the anonymous reviewers for their helpful comments and Mr. Da Wang for his contribution to part of the computer simulation and technical discussion.

References

- [1] <http://www.mozilla.org>.
- [2] M. Angermann. Analysis of speculative prefetching. *SIGMOBILE Mob. Comput. Commun. Rev.*, 6(2):13–17, 2002.
- [3] R. Berezdivin, R. Breinig, and R. Topp. Next-generation wireless communications concepts and technologies. *IEEE Communications Magazine*, 40(3):108–116, March 2002.
- [4] D. Bonino, F. Corno, and G. Squillero. A real-time evolutionary algorithm for web prediction. In *Proc. of IEEE/WIC Int. Conf. on Web Intelligence*, pages 139–145, Oct 2003.
- [5] E. Cohen, B. Krishnamurthy, and J. Rexford. Efficient algorithms for predicting requests to web servers. In *Proc. of IEEE INFOCOM*, pages 284–293, March 1999.
- [6] M. Crovella and P. Barford. The network effects of prefetching. In *Proc. of IEEE INFOCOM*, pages 1232–1239, 1998.
- [7] B. D. Davison. Predicting web actions from html content. In *Proc. of ACM HYPERTEXT*, pages 159–168, Jun 2002.
- [8] S. Drew. Multiuser network-aware web prefetching in heterogeneous wireless network. Master's thesis, University of Toronto, May 2005.
- [9] S. Drew and B. Liang. Mobility-aware web prefetching over heterogeneous wireless networks. In *Proc. of the 15th IEEE PIMRC*, pages 687–691, Sept 2004.
- [10] S. Gitenis and N. Bambos. Power-controlled data prefetching/caching in wireless packet networks. In *Proc. of IEEE INFOCOM*, pages 1405–1414, June 2002.
- [11] Z. Jiang and L. Kleinrock. An adaptive network prefetching scheme. *IEEE Journal on Selected Areas in Communications*, 16(3):358–368, Apr. 1998.
- [12] Z. Jiang and L. Kleinrock. Web prefetching in a mobile environment. *IEEE Personal Communications*, 5:25–34, Oct. 1998.
- [13] B. Liang and Z. J. Haas. Predictive distance-based mobility management for multi-dimensional PCS networks. *IEEE/ACM Transactions on Networking*, 11(5):718–732, Oct. 2003.
- [14] B. Liang, A. H. Zahran, and A. Saleh. Application signal threshold adaptation for vertical handoff in heterogeneous wireless networks. In *Proc. of IFIP Networking*, May 2005. Lecture Notes in Computer Science, vol. 3462, pp. 1193–1205.
- [15] V. N. Padmanabhan and J. C. Mogul. Using predictive prefetching to improve world wide web latency. *SIGCOMM Comput. Commun. Rev.*, 26:22–36, 1996.
- [16] N. U. Prabhu. *Foundations of Queueing Theory*. Kluwer Academic Publishers, 1997.
- [17] S. M. Ross. *Stochastic Processes, 2nd Edition*. John Wiley & Sons, Inc., 1996.
- [18] N. Tuah, M. Kumar, and S. Venkatesh. Resource-aware speculative prefetching in wireless networks. *Wireless Networks*, 9:61–72, 2003.
- [19] L. Yin and G. Cao. Adaptive power-aware prefetch in wireless networks. *IEEE Transactions on Wireless Communications*, (5):1648–1658, Sep 2004.