Performance of Multiuser Network-Aware Prefetching in Heterogeneous Wireless Systems

Ben Liang, Stephen Drew, and Da Wang

Ben Liang (Corresponding Author)[1]
Department of Electrical and Computer Engineering
University of Toronto
10 King's College Road
Toronto, Ontario, M5S 3G4, Canada
Email: liang@comm.utoronto.ca
Tel: 1-416-946-8614, Fax: 1-416-978-4425

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

Da Wang
Department of Electrical and Computer Engineering
University of Toronto
10 King's College Road
Toronto, Ontario, M5S 3G4, Canada
Email: da.wang@utoronto.ca

[1]A preliminary version of this article was presented in the International Conference on Quality of Service in Heterogeneous Wired/Wireless Networks (QShine) 2006. This work was supported in part by the Natural Sciences and Engineering Research Council of Canada and Bell Canada through its Bell University Laboratories R&D program.

Abstract

We study the performance of multiuser document prefetching in a two-tier heterogeneous wireless system. Mobility-aware prefetching was previously introduced 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 the document access delay and the user prefetching quantity. This phenomenon is particularly acute in the heterogeneous environment. We investigate into alleviating the system traffic load through prefetch thresholding, accounting for server queuing prioritization. We propose a novel analysis framework to evaluate the performance of the thresholding approach. Numerical and simulation results show that the proposed analysis is accurate for a wide variety of access, service, and mobility patterns. We further demonstrate that stability can be maintained even under heavy usage, providing both the same scalability as a non-prefetching system and the performance gain associated with prefetching.

Keywords: Mobile prefetching, heterogeneous wireless networks, performance modelling, queuing analysis

1 Introduction

The future wireless information system will likely 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 different access networks should dynamically adapt and make intelligent choices to balance the trade-offs between various performance factors [21]. 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 a wide-area cellular network and a WLAN as examples for these two networks, as shown in Fig. 1.

Prefetching is a technique in which the client device pro-actively 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 are highly likely to be accessed by the user and hence should be automatically fetched by the browser. Another example is in the Infostation approach [11], where roaming users receive large chunks of information through discontinuous pockets of high-throughput network coverage. 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 delays or down-times between consecutive access requests.

There exists much research on Web document prediction and prefetching [16, 12, 13, 5, 2, 7, 4]. Most of the proposed methods make use of user histories to arrive at

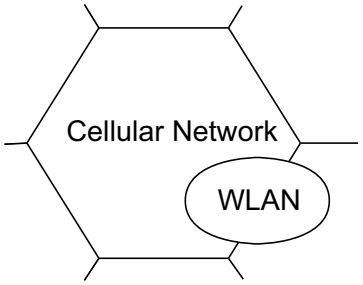

Figure 1: An example two-tier wireless heterogeneous network.

informed predictions. It has been shown that traditional Web caching systems without prefetching can 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% [12, 7]. 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]. 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 successfully prefetched document when the user is about to leave the coverage area of the WLAN potentially displaces more expensive future cellular network access with cheaper WLAN access. However, the major side effect of 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, 20] or storage capacity [19].

In this work, we extend our previous results in [9] and examine the effect of multiple prefetching users on system performance. A unique challenge that arises in the multiuser 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 prefetch requests to the server, further increasing the time to service requests, which in turn may lead to more aggressive prefetching. The increase in traffic delay 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. Therefore, the user must adjust their prefetching aggressiveness based on their current mobility, application characteristics, and the system load. We term this *network-aware* prefetching. Other metrics that are used in the prefetching decision process may include the network bandwidths, data access costs, and the user perceived value of time.

We propose a novel analysis framework to evaluate the performance of network-aware prefetching in a two-tier network. The analysis framework accounts for queuing prioritization and reneging at the document server, allowing service differentiation between regular and prefectch requests. The analysis framework is developed such that the effects of any mobility pattern, network topology, or access pattern could be used as inputs into network-aware prefetching over heterogeneous networks. We further show

the effects of prioritized service and user selfishness by illustrating the performance of alternate queue management and prefetching algorithms.

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 Multiuser Network-Aware Prefetching

In this section, we first describe the system model for multiuser network-aware prefetching and then present a derivation for the individual user's optimal 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, 15]. Users access documents one-by-one from a set $\mathcal{I}$. They are provided with a mechanism to estimate the access probabilities for their next set of possible documents [16, 12, 13, 5, 2, 7, 4]. Denote these probabilities by $p_a(i;j)$, where $i \in \mathcal{I}$, and $j \in \mathbb{Z}$ are the indices of access epochs.

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*, denoted by H, and will prefetch in a single batch request all documents with access probabilities greater than the threshold, i.e., from the set $\{i|p_a(i;j) > H\}$.

The central server on the WLAN side 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 at any time within the inter-request interval, we study a two-priority system where regular document requests are given high priority (HP), and all prefetch requests are given low priority (LP). We choose a preemptive resume system [17], 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 to reduce unnecessary service. 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 reneged.

When the user is outside the WLAN, it is served by the cellular network. Since we do not consider prefetching within the cellular network, all requests outside the WLAN are for regular documents. Furthermore, we assume that the cellular network provides subscription-based guaranteed quality-of-service, and hence any queuing delay is negligible. Let s denote the average document size and b_W and b_C denote the constant data rate provided by the WLAN and the cellular network, respectively. Then the average transmission delay outside the WLAN is s/b_C. Since $b_W \gg b_C$ in general, we assume the transmission delay within the WLAN is small and negligible.

To facilitate the user cost analysis below, we let α_W and α_C denote the price per byte of access to the WLAN and to the cellular network, respectively. Note that these prices may account for both the monetary cost charged by the service provider and the communication energy cost to the mobile device. We further let α_T denote the cost of lost time, where lost time is defined as the duration in which a user is waiting for the server to service an HP document request. It has been demonstrated in [12] that, in general, a suitable value for the user perceived cost of access delay may be the user's income level.

2.2 User Prefetching Strategy

We define the total cost for accessing a document as the sum of access cost and the penalty for access delay. Then, the prefetching threshold for an individual user 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 by c_p and c_{np}, respectively. Here we have omitted the indices in $p_a(i;j)$ for brevity.

If the document is not prefetched and if it is indeed requested by the user at time t_r, then the user's document request will be forwarded to the server with high priority. Let t_w be the user's residual WLAN residence time. Suppose the document is requested when the user has moved out of the WLAN, i.e. $t_w < t_r$, then the document cost is a sum of the cellular access cost and the cellular access delay cost:

$$c_C = \alpha_C s + \alpha_T s/b_C \ . \tag{1}$$

Otherwise, an HP request is sent to the WLAN server. Then the overall document cost depends on whether the HP request is served before the user moves out of the WLAN:

$$\begin{aligned} c_{HP} =& P\{S^0_{HP} < t_w\}(\alpha_W s + \alpha_T E[S^0_{HP} | S^0_{HP} < t_w]) \\ &+ P\{S^0_{HP} > t_w\}(\alpha_T E[t_w | S^0_{HP} > t_w] + c_C) \\ =& P\{S^0_{HP} < t_w\}\alpha_W s + P\{S^0_{HP} > t_w\}c_C \\ &+ \alpha_T E[\min(S^0_{HP}, t_w)] \ , \end{aligned} \tag{2}$$

where S^0_{HP} is the *untouched* sojourn time of an HP request if there were no reneging due to the user moving out of the WLAN. Given the preemptive nature of the server, we may assume that with high probability the HP sojourn time is less than t_w. Then

we can use the following approximation[2] to simplify the computation of c_{HP}:

$$c_{HP} \approx \alpha_W s + \alpha_T E[S_{HP}] \ , \tag{3}$$

where S_{HP} denotes the actual, or *touched*, HP request sojourn time in the server queue, given by

$$S_{HP} = \min(S^0_{HP}, t_w) \ . \tag{4}$$

The expected cost of not prefetching the document is then

$$c_{np} = p_a(P\{t_w < t_r\}c_C + P\{t_r < t_w\}c_{HP}) \ . \tag{5}$$

The expected cost of prefetching the document includes the access cost of prefetching the document, and the access and delay cost of requesting the document, in the case of failed prefetching and the document actually being requested by the user. Let S^0_{LP} be the *untouched* sojourn time of a LP request if there were no reneging due to a new request by the user or the user moving out of the WLAN. Then the cost of prefetching depends on the different ways that S^0_{LP}, t_r, and t_w are ordered. If $S^0_{LP} < t_r < t_w$ or $S^0_{LP} < t_w < t_r$, then the prefetch request is served by the WLAN server in time. In this case, whether or not the user actually requests the document at time t_r, the document cost is given by $\alpha_W s$. If $t_r < S^0_{LP} < t_w$ or $t_r < t_w < S^0_{LP}$, then the user requests a new document before the prefetching is completed, and the user is still in the WLAN. In this case, if the document under consideration is actually the one requested by the user, then a document request will be sent to the WLAN server, and the cost of this is c_{HP}. If $t_w < S^0_{LP} < t_r$ or $t_w < t_r < S^0_{LP}$, then the prefetch request is not served before the user moves out of the WLAN, and the user requests a new document when it is out of the WLAN. In this case, if the document under consideration is actually the one requested by the user, then a document request will be sent to the cellular network, and the cost of this is c_C. Summarizing the above three cases, we have

$$\begin{aligned} c_p =& P\{S^0_{LP} < \min(t_r, t_w)\}\alpha_W s \\ &+ P\{t_r < t_w\}P\{S^0_{LP} > \min(t_r, t_w)\}p_a c_{HP} \\ &+ P\{t_w < t_r\}P\{S^0_{LP} > \min(t_r, t_w)\}p_a c_C \ . \end{aligned} \tag{6}$$

The prefetching strategy is a binary decision on each document in $\mathcal{I}$. In general, the more likely is a document to be accessed next, the more beneficial it is to prefetch it. Then, assuming the prefetching decisions on different documents are independent, the optimal prefetching threshold is given by

$$H = \min\{p_a | c_p \le c_{np}\} \ . \tag{7}$$

Substituting (5) and (6) above and simplifying, we have

$$H = \frac{\alpha_W s}{P\{t_r < t_w\}c_{HP} + P\{t_w < t_r\}c_C} \ . \tag{8}$$

[2]This approximation allows fast computation of the prefetching strategy by mobile devices. It is not required for the analysis in this paper.

Somewhat surprisingly, even though the LP sojourn time, and hence the likelihood that prefetching will be successful, plays a major role in the cost of prefetching, it is not a direct factor in the optimal prefetching decision by a user. An intuitive explanation is as follows. If the prefetch request is not served by the WLAN server in time before either the next user request or the user moving out of the WLAN, then attempting to prefetch the document is the same as not doing so. Hence, the c_{np} and c_p comparison is determined only by the weighted cost contribution from the case where prefetching is served in time. In this case, there is no longer the need to consider the LP sojourn time.

The LP sojourn time affects the prefetching threshold only indirectly through $E[S_{HP}]$, by changing the amount of HP requests. Equation (8) shows also that, in practice, a user can determine its prefetching threshold by simply obtaining the current value of $E[S_{HP}]$. This value can be estimated through the user's own HP request history, or it can be estimated by the server and disseminated to the users.

Note that (8) gives only the optimal prefetching threshold for an individual user, assuming each user is selfish. Given a fixed set of users, it is clear that S^0_{HP} is upper bounded, and hence an equilibrium for $E[S^0_{HP}]$ and H exists. However, the prefetching decision based on this *user optimal* threshold generally is not optimal in terms of the overall welfare of the users in the network. In Section 4, we provide simulation results to show the difference between the *network optimal* prefetching strategy and the user optimal prefetching strategy for different number of users in the network.

3 Performance Analysis Framework

This section provides an analytical framework for multiuser network-aware prefetching. We present a method to compute the distributions of S^0_{HP} and S^0_{LP}, which depend on the prefetching quantity by the other users and hence H. Since H is in turn a function of S^0_{HP}, a recursive procedure can be employed to approach the optimal individual-user prefetching threshold, assuming symmetric user behavior. 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 $(\#LP, \#HP)$, denoting the number of LP and HP requests in the queue. 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 residual 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 simplified analysis provides a close approximation even for non-exponential t_r, t_s, and t_w.

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 by $P_L[k]$, we count the number of documents with access probability exceeding the prefetching threshold:

$$P_L[k] = \sum_{\zeta \in \Delta} P\{p_a = \zeta\} P\Big\{ \sum_{i \in \mathcal{I}} 1(\zeta(i;j) > H) = k \Big\} , \tag{9}$$

where Δ represents the set of all document access probability distributions and $1(\cdot)$ is the indicator function. In applications where the total number of documents is large (e.g., Web browsing), to reduce the size of the above summation, we may replace $\mathcal{I}$ by a subset $\mathcal{I}'$ containing only the documents with non-negligible access probability. Let $x_D = |\mathcal{I}'|$. Then x_D represents the number of all probable 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 hence 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 do not use this assumption in our simulation in Section 4, 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

$$\lambda_{k,0|j,n} = \lambda N p_W P_L[k] \sum_{i=1}^{x_D} P_L[i] P\{C|i\}(1 - P\{D_L|j,n\}) , \tag{10}$$

where N is the number of users in the WLAN, p_W is the probability that the inter-request time is less than the WLAN residence time, i.e.,

$$p_W = P\{t_r < t_w\} = \frac{\lambda}{\lambda + \gamma} , \tag{11}$$

$P\{C|k\}$ 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 were 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.,

$$P\{D|j,n\} = P\{t_r < S^0_{LP}|j,n\} . \tag{12}$$

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

$$\begin{aligned} P\{M|j,n\} = P_L\{0\} + \sum_{i=1}^{x_D} P_L\{i\}(1 - P\{C|i\}) \\ (1 - P\{D_L|j,n\}) \ . \end{aligned} \tag{13}$$

For there to be a net movement of $k \geq 0$ 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 $i \geq k+1$ LP requests inducing $i-k$ dropped LP requests. Hence, for $0 \leq k \leq x_D$, the arrival rate of request batches that cause a queue state net movement of $(k, 1)$, given the current state (j, n), is

$$\begin{aligned} \lambda_{k,1|j,n} = \lambda N P_L[k] \Big(p_W P\{M|j,n\} + 1 - p_W \Big) \\ + \lambda p_W \sum_{i=k+1}^{x_D} P_L[i] P_L[i-k] P\{D_L|j,n\} \ . \end{aligned} \tag{14}$$

For the net movement of LP requests to be less than zero, there must be dropped LP requests. Thus, for $0 < k \leq x_D$, the arrival rate of request batches that cause a queue state net movement of $(-k, 1)$, given the current state (j, n), is

$$\lambda_{-k,1|j,n} = \lambda N p_W \sum_{i=0}^{x_D-k} P_L[i] P_L[i+k] P\{D_L|j,n\} \ . \tag{15}$$

Note that for precise computation of the above arrival rates, one needs to further apply the upper bound j for the number of dropped LP requests. This can be carried out in a straight forward manner, but with awkward notations. Instead, we ignore this effect of j throughout the analysis in this paper. The resulting inaccuracy will be insignificant, since the upper bound j is small and restricting only when the traffic load at the server queue is low, in which case the LP requests are not likely to be dropped.

The arrival rates in (10), (14), and (15) are then combined with service rates for the preemptive-resume priority model and WLAN departure rates, as shown in Fig. 2, to define a continuous-time Markov chain that represents the server queue. Note that there is no need to consider the removal of stale requests, since they are already accounted for in the arrival rates. This Markov chain is clearly ergodic. We use $u_{j,n}$ to denote the steady state distribution of this Markov chain. We further define λ' as the sum of all outgoing rates from a specific state

$$\begin{aligned} \lambda'_{j,n} = \sum_{k=1}^{x_D} \lambda_{k,0|j,n} + \sum_{k=0}^{x_D} \lambda_{k,1|j,n} \\ + \sum_{k=1}^{min(x_D,j)} \lambda_{-k,1|j,n} \ . \end{aligned} \tag{16}$$

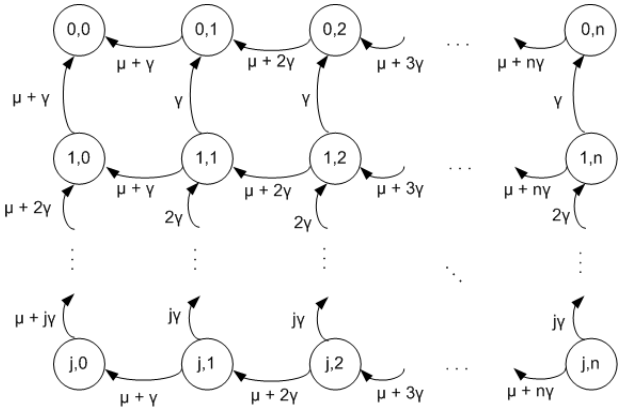

Figure 2: Priority service and WLAN departure rate model.

Then, the following balance equations can be used to numerically compute $u_{j,n}$ [18]:

$$\lambda'_{0,0}u_{0,0} = (\mu+\gamma)u_{1,0} + (\mu+\gamma)u_{0,1}\ , \tag{17}$$

$$\begin{aligned}(\lambda'_{0,n} + \mu + n\gamma)u_{0,n} &= (\mu+(n+1)\gamma)u_{0,n+1} + \gamma u_{1,n} \\ &+ \lambda_{0,1|0,n-1}u_{0,n-1} + \sum_{k=1}^{x_D}\lambda_{-k,1|k,n-1}u_{k,n-1}, \quad n \geq 1\ ,\end{aligned} \tag{18}$$

$$\begin{aligned}(\lambda'_{j,0} + \mu + j\gamma)u_{j,0} &= (\mu+(j+1)\gamma)u_{j+1,0} \\ &+ (\mu+\gamma)u_{j,1} + \sum_{k=1}^{min(x_D,j)}\lambda_{k,0|j-k,0}u_{j-k,0}, \quad j \geq 1\ ,\end{aligned} \tag{19}$$

$$\begin{aligned}(\lambda'_{j,n} + \mu + (j+n)\gamma)u_{j,n} &= (\mu+(n+1)\gamma)u_{j,n+1} \\ &+ (j+1)\gamma u_{j+1,n} + \sum_{k=1}^{min(x_D,j)}\lambda_{k,0|j-k,n}u_{j-k,n} \\ &+ \sum_{k=0}^{min(x_D,j)}\lambda_{k,1|j-k,n-1}u_{j-k,n-1} \\ &+ \sum_{k=1}^{x_D}\lambda_{-k,1|j+k,n-1}u_{j+k,n-1}, \quad j \geq 1, n \geq 1\ .\end{aligned} \tag{20}$$

3.2 HP and LP Sojourn Time Distribution

Since the server queue is preemptive, the touched HP sojourn time 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 S^0_{HP}. It can be shown that, given n HP requests in the queue,

the untouched sojourn time of the $(n+1)$th HP request has distribution [17]

$$f_{S^0_{HP}|n}(x) = \frac{\mu}{\beta_{n+1}(\gamma)} \sum_{i=0}^{n} \frac{(-1)^i}{i!(n-i)!} e^{-(\mu+i\gamma)x} \,, \tag{21}$$

where

$$\begin{aligned} &\beta_1(\gamma) = 1 \,, \\ &\beta_n(\gamma) = \left[\left(\frac{\mu+\gamma}{\gamma} \right) \left(\frac{\mu+2\gamma}{\gamma} \right) ... \left(\frac{\mu+(n-1)\gamma}{\gamma} \right) \right]^{-1} , \quad n \geq 2 \,. \end{aligned} \tag{22}$$

Then, since (21) is a weighted sum of exponential distributions, it is easy to show that the touched sojourn time, S_{HP}, has distribution

$$f_{S_{HP}|n}(x) = \frac{\mu}{\beta_{n+1}(\gamma)} \sum_{i=0}^{n} \frac{(-1)^i e^{-[\mu+(i+1)\gamma]x}}{i!(n-i)!} \frac{\mu+(i+1)\gamma}{\mu+i\gamma} \,. \tag{23}$$

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 [17], the relationship between the busy period of an M/M/1 queue initiated by a workload x is

$$E[e^{-\theta T_{HP}[x]}] = E[e^{-x\eta}] \,, \tag{24}$$

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

$$\eta = \frac{\theta + \lambda_{HP} - \mu + \sqrt{(\theta + \lambda_{HP} + \mu)^2 - 4\lambda_{HP}\mu}}{2} \,, \tag{25}$$

and λ_{HP} is the request rate of HP documents and can be computed by summing over all request rates that include one web document, i.e.,

$$\lambda_{HP} = \sum_{j,n} u_{j,n} \sum_{i=-x_D}^{x_D} \lambda_{i,1|j,n} \,. \tag{26}$$

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

$$\overline{f}_{W_{LP}|n}(\theta) = \left(\frac{\mu}{\mu+\eta} \right)^n \,. \tag{27}$$

Furthermore, similar to (21), we can determine the untouched LP waiting time distribution, due to j existing LP requests in the queue, given by the Laplace transform

$$\overline{f}_{S^0_{LP}|j}(\theta) = \frac{\mu}{\beta_{j+1}(\nu)} \sum_{i=0}^{j} \frac{(-1)^i}{i!(j-i)!} \frac{1}{\mu+i\nu+\eta} \,, \tag{28}$$

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,

$$\overline{f}_{S^0_{LP}|j,n}(\theta) = \overline{f}_{S^0_{LP}|j}(\theta)\overline{f}_{W_{LP}|n}(\theta) \; . \tag{29}$$

To improve the above approximation, we can further consider concurrently the dropping of LP requests and the position of the new LP request in the batch request. We assume that in an 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

$$\begin{aligned} \overline{f}_{\overline{S}^0_{LP}|j,n}(\theta) = & \left(\sum_{k=1}^{x_D} f\{k, 1|j, n\} + \sum_{k=1}^{x_D} f\{-k, 1|j, n\} \right) \\ & \left(\frac{\mu}{\mu + \eta} \right)^{n+1} + \sum_{k=1}^{x_D} f\{k, 0|j, n\} \left(\frac{\mu}{\mu + \eta} \right)^{n} , \end{aligned} \tag{30}$$

where $f\{k, m|j, n\}$, $1 \leq k \leq x_D$, 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

$$\begin{aligned} f\{k, 0|j, n\} = & \, p_W \left(\frac{P_L\{k\}}{k} \sum_{i=1}^{k} \overline{f}_{S^0_{LP}|j+i,n}(\theta) \right) \\ & \sum_{i=1}^{x_D} P_L[i] P\{C|i\} (1 - P\{D_L|j, n\}) \; , \end{aligned} \tag{31}$$

$$\begin{aligned} f\{k, 1|j, n\} = & \left(p_W P\{M|j, n\} + 1 - p_W \right) \\ & \left(\frac{P_L[k]}{k} \sum_{i=1}^{k} \overline{f}_{S^0_{LP}|j+i,n}(\theta) \right) + p_W P\{D_L|j, n\} \\ & \sum_{i=k+1}^{\min(k+j, x_D)} P_L[i-k] \frac{P_L[i]}{i} \sum_{l=1}^{i} \overline{f}_{S^0_{LP}|j+l-i+k,n}(\theta) \; , \end{aligned} \tag{32}$$

$$\begin{aligned} f\{-k, 1|j, n\} = & \, p_W \sum_{i=1}^{\min(x_D, j)-k} P_L[i+k] P\{D_L|j, n\} \\ & \frac{P_L[i]}{i} \sum_{l=1}^{i} \overline{f}_{S^0_{LP}|j-k+l-i,n}(\theta) \; . \end{aligned} \tag{33}$$

For the numerical analysis in Section 4, we use (30) for improved LP sojourn time estimation.

Finally, the probability of LP dropping due to staleness, $P\{D|j,n\}$, which is used throughout the above analysis, can be computed recursively using the LP sojourn time distribution. With t_r exponential with rate λ, we have

$$\begin{aligned} P\{D|j,n\} &= P\{t_r < \overline{S}^0_{LP}|j,n\} \\ &= \int_0^\infty \int_0^s \lambda e^{-\lambda t} f_{\overline{S}^0_{LP}|j,n}(s)dtds \\ &= 1 - \overline{f}_{\overline{S}^0_{LP}|j,n}(\lambda) \, . \end{aligned} \tag{34}$$

3.3 Performance Metrics

The amount of HP and LP traffic received by the users, i.e., the traffic that is successfully serviced by the queue, is an important performance metric. HP requests may be dropped due to mobility, i.e., when $S^0_{HP} > t_w$. Hence, the rate of received HP traffic is

$$\begin{aligned} \rho_{HP} &= \sum_{j,n} u_{j,n} P\{S^0_{HP} < t_w|n\} \sum_{i=-x_D}^{x_D} \lambda_{i,1|j,n} \\ &= \sum_{j,n} u_{j,n} \int_0^\infty \int_s^\infty \gamma e^{-\gamma t} f_{S^0_{HP}|n}(s)dtds \sum_{i=-x_D}^{x_D} \lambda_{i,1|j,n} \\ &= \sum_{j,n} u_{j,n} \overline{f}_{S^0_{HP}|n}(\gamma) \sum_{i=-x_D}^{x_D} \lambda_{i,1|j,n} \, . \end{aligned} \tag{35}$$

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

$$\begin{aligned} \rho_{LP} &= \lambda_{LP} \sum_{j,n} u_{j,n} P\{\overline{S}^0_{LP} < \min(t_r, t_w)|j,n\} \\ &= \lambda_{LP} \sum_{j,n} u_{j,n} \int_0^\infty \int_s^\infty (\lambda+\gamma) e^{-(\lambda+\gamma)t} f_{\overline{S}^0_{LP}|j,n}(s)dtds \\ &= \lambda_{LP} \sum_{j,n} u_{j,n} \overline{f}_{\overline{S}^0_{LP}|j,n}(\lambda+\gamma) \, , \end{aligned} \tag{36}$$

where λ_{LP} is the rate of LP requests:

$$\lambda_{LP} = \lambda N \sum_{k=0}^{|\mathcal{I}|} k P_L[k] \, . \tag{37}$$

We are also interested in the expected cost perceived by a user. Given any prefetching threshold H, the expected cost per document access is

$$\begin{aligned} C_p(H) = &\sum_{\zeta \in \Delta} P\{p_a = \zeta\} \\ &\cdot \sum_{i \in \mathcal{I}} \big(1(\zeta(i;j) > H) c_p(i) + 1(\zeta(i;j) < H) c_{np}(i) \big) \, . \end{aligned} \tag{38}$$

In comparison, the expected cost per document access without prefetching is

$$C_{np} = \sum_{\zeta \in \Delta} P\{p_a = \zeta\} \sum_{i \in \mathcal{I}} c_{np}(i) \; . \tag{39}$$

We define the performance gain of prefetching with threshold H as the ratio between (39) and (38).

4 Numerical and Simulation Results

A C++ based event-driven simulation environment has been developed to validate the proposed analysis model and to obtain further insights into design alternatives. The simulator consists of a server queue connected to the WLAN, a cellular network model, and multiple users accessing the WLAN server in a decentralized manner. To maintain the same number of users in the WLAN, we employ a cyclic model, such that immediately after a user departs the WLAN, it is replaced by another user joining the WLAN. Note that as explained previously, the simplifying assumptions made in Section 3 are not made in the simulation.

4.1 Experimental Setup

To represent the level of user mobility, we use the quantity $p_W = P\{t_r < t_w\}$, the probability that the user remains inside the WLAN at the next document access. To describe the various levels of predictability in a user's future document access, we introduce a predictability parameter b, such that the next-document access probabilities $p_a(i,j)$ follow an exponentially weighted, truncated geometric distribution as follows:

$$p_a(i,j) = \frac{q_j(1-q_j)^{i-1}}{1-(1-q_j)^{x_D}} \; , i = 1, 2, \ldots, x_D \; , \tag{40}$$

where q_j is a random variable with a truncated exponential distribution

$$f_q(x) = \frac{be^{bx}}{e^b - 1} \; , 0 < x < 1 \; , b \in \mathbb{R} \; . \tag{41}$$

Clearly, the larger q_j is, the more concentrated is the probability distribution for the set of probable documents, and hence the more predictable is the next document access. Therefore, the parameter b allows the tuning of $p_a(i,j)$ to represent a wide range of predictability. In particular, the larger b is, the more frequent are the occasions where the next document access is highly predictable.

As an illustrative example, for the numerical analysis and simulation in this section, the default value of b is set to 0, unless otherwise stated. We further assume that $x_D = 10$. We choose nominal values for the other system parameters. We assume that the inter-request time t_r has mean 12 seconds, the residual WLAN residence time t_w has mean 48 seconds, both of which are exponentially distributed in the default case. Hence, the default value of p_W is 0.8. The other default system parameters are

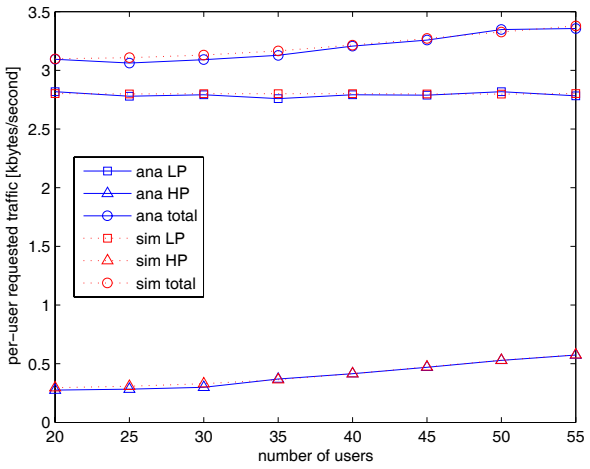

Figure 3: Requested traffic rate per user.

$b_W = 100KB/s$, $b_C = 5KB/s$, $\alpha_W = \$1/MB$, $\alpha_C = \$0.05/KB$, $s = 10KB$, and $\alpha_T = \$20$ per hour.

In the recursive numerical analysis, we use 0.1 as the initial value for H and a zero matrix for $P\{D|j,n\}$. The recursion is stopped when $P\{D|j,n\}$ converges. In each simulation run, 3600 seconds are simulated, from which we allow 900 seconds of warm-up time to eliminate the transient behavior. The server keeps track of the entire history of the HP request sojourn time S_{HP} and continuously updates its time average. Each user updates its H value dynamically based on the average value of S_{HP} using (8).

For each data point in the simulation results, 100 repeated runs are conducted. In general, we observe that the resultant 99% confidence intervals are too small to see in our data plots. Therefore, they are shown only in figures where they are significant.

4.2 Traffic Load and Prefetching Threshold

Fig. 3 illustrates the requested HP and LP traffic rates per user versus the number of users in the WLAN. We observe that the analysis and simulation results nearly overlap each other. Furthermore, the amount of requested LP traffic is much greater than the amount of requested HP traffic, suggesting aggressive prefetching. The HP demand gradually increases, showing that less and less of the prefetch requests are resulting in hits. However, the LP demand is well regulated and remains stable.

Not all requests are served successfully. A comparison of the expected overall HP and LP traffic received by the users in the system is shown in Fig. 4. The analysis and simulation results are again very close, with less than 10% difference in all cases. The benefit of adaptive prefetching thresholding is seen, as the amount of LP traffic peaks at between 40 to 50 users, at the same time as the total traffic at the server

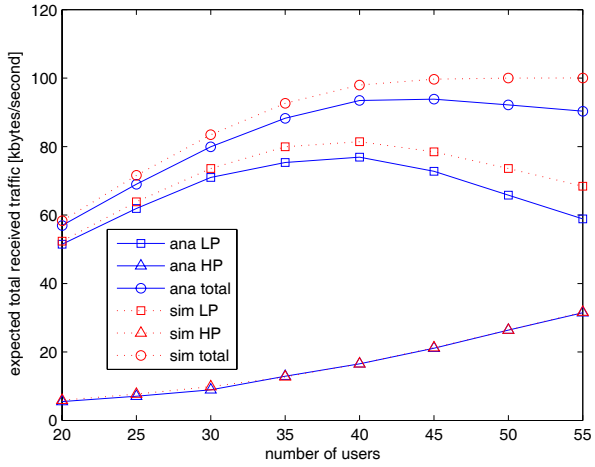

Figure 4: Expected received traffic by all users.

approaches capacity. Indeed, the server hovers near capacity even for a much larger user population.

In Fig. 5, we show the user prefetching threshold H versus the number of users. This figure demonstrate that H remain steady even as the server load becomes heavy, confirming out observation in Fig. 3 that the LP demand is stable. Fig. 5 also compares the prefetching thresholds for different document request arrival patterns. We present further details on this in the next subsection.

4.3 Non-Markovian Parameters

Throughout the analysis in this work, we have assumed that t_r, t_s, and t_w are Markovian. In this subsection, we compare the analytical results against simulation with non-Markovian parameters. For each of t_r, t_s, and t_w, we obtain simulation results with a heavy-tail Pareto distribution of index 10 and a nearly-Gaussian Erlang distribution of order 20, both scaled to have the same mean as in the default case with an exponential distribution.

Fig. 5 and Fig. 6 show the user prefetching threshold when one of the above parameters is non-Markovian while the other two are exponential. They both demonstrate that the analysis provide accurate results for all cases. We further observe from simulation that the effect of non-Markovian t_r, t_s, or t_w on the traffic load is not significant. Fig. 7 illustrates this for the case of t_s. Figures for the other cases are similar and are omitted to avoid redundancy. Hence, we conclude that the prefetching threshold and the system performance is almost insensitive to the different inter-request time, transmission time, and WLAN residence time distributions. Therefore, the proposed analysis is applicable to a wide range of practical systems.

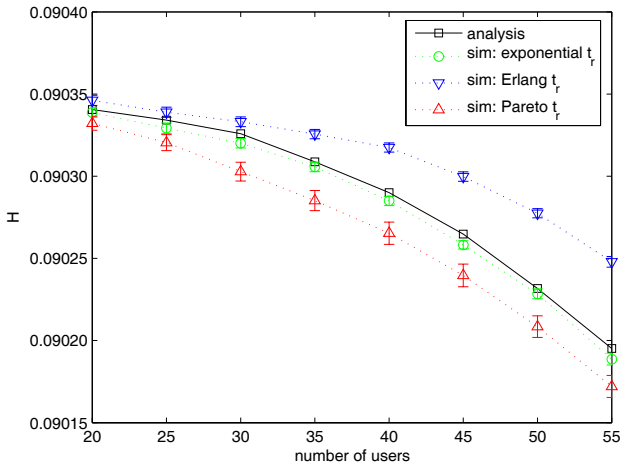

Figure 5: Prefetching threshold for different inter-request time distributions.

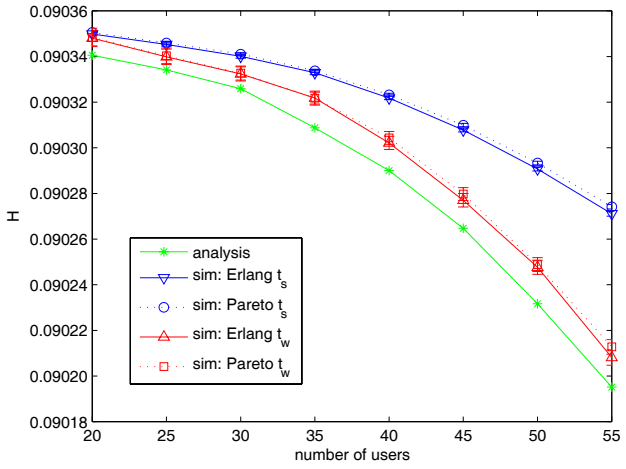

Figure 6: Prefetching threshold for different transmission time and residual WLAN residence time distributions.

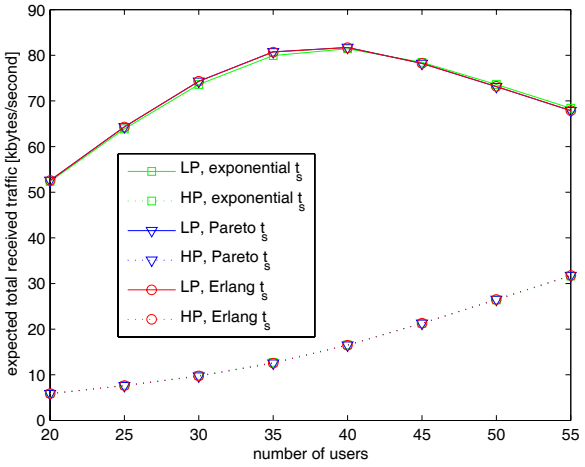

Figure 7: Expected received traffic by all users, for different document transmission time distributions.

4.4 Performance Gain and Effect of Mobility and Access Predictability

We plot the performance gain of prefetching using the individual-user optimal threshold over non-prefetching, for five different levels of mobility, represented by p_W, in Fig. 8, and for five different levels of access predictability, represented by b, in Fig. 9. Both figures show that significant advantage can be achieved by using prefetching. Fig. 8 further suggests 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 40), the more aggressive prefetching by users due to higher mobility remains detrimental to system performance. Fig. 9 quantifies how the performance gain increases rapidly as the user access pattern becomes more predictable. However, it again confirms that the performance gain diminishes as the system load is increased. We show in the next subsection that a prefetching strategy optimized for all users in the network can alleviate the diminishing of performance gain at high system load.

4.5 Prefetching Scheme Alternatives

We consider two alternate prefetching schemes. In the first, less complex, alternative, we note that differentiated service is not universally available. Hence, we assume that the prefetch requests and regular document requests at the server queue are not prioritized. In this *non-priority* scheme, the user still applies the prefetching threshold H derived in (8), with the HP sojourn time now representing simply the regular document sojourn time. In the second, more complex, alternative, we note that the prefetching strategy in Section 2 assumes selfish users whose prefetching decision does not take

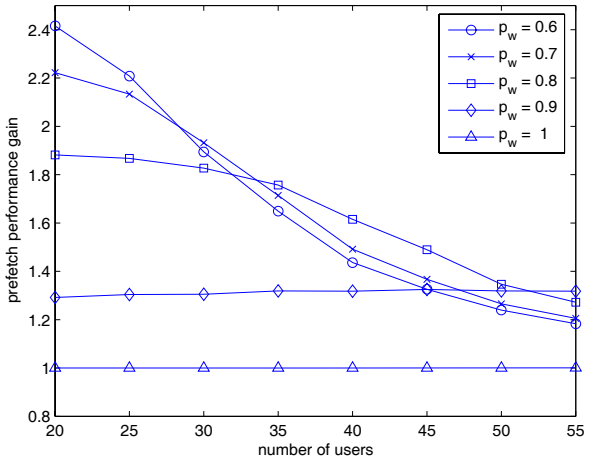

Figure 8: Performance gain for various degrees of mobility.

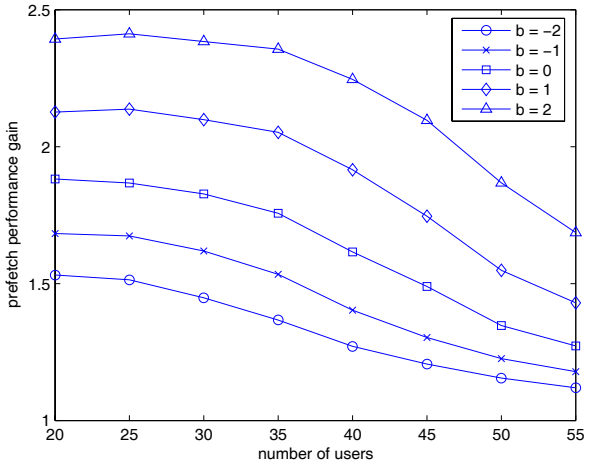

Figure 9: Performance gain for various degrees of access predictability.

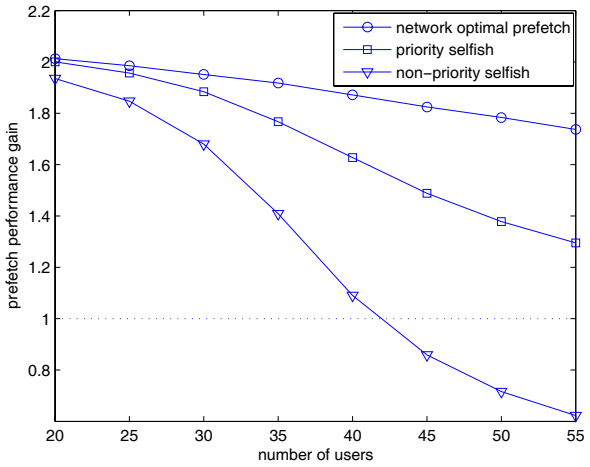

Figure 10: Comparing performance gains with non-priority prefetching and network optimal prefetching.

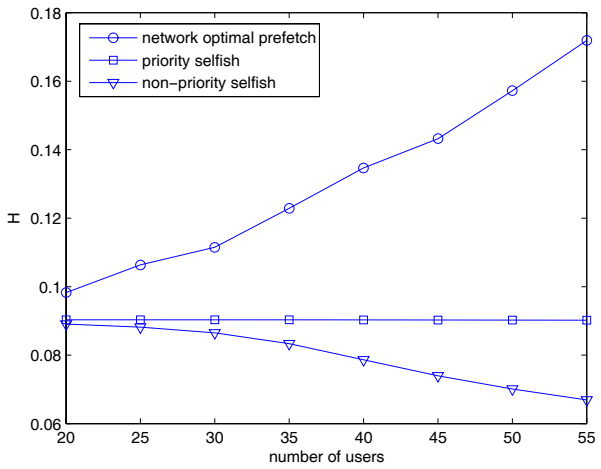

Figure 11: Comparing prefetching thresholds with non-priority prefetching and network optimal prefetching.

into account the welfare of the other users. The overall network efficiency suffers with selfish prefetching, which is reflected by the quickly diminishing performance gain shown in Figures 8 and 9. Hence, we study an idealized *network optimal* strategy, where the optimal H is chosen to minimize the expected total cost of all users. We note that the derivation developed to compute (38) is still applicable to evaluate and optimize the performance of this scheme.

Fig. 10 and Fig. 11 show the performance gain and associated prefetching thresholds, respectively, of all three prefetching schemes. For the non-priority scheme, we observe that the prefetching threshold quickly decreases as the number of users increases, since now the regular document requests take longer time to be served while competing with the prefetch requests. At low system load, prefetching without prioritization still outperforms no prefetching by a large amount. However, severe performance degradation at hight system load results from the flooding of prefetch requests. In particular, when there are more than about 40 users, the equilibrium point of this scheme is worse than a scheme with no prefetching.

For the network optimal scheme, we observe that, at low system load, the prefetching threshold and prefetching performance are similar to those in selfish individual user prefetching. However, the network optimal prefetching threshold increases as the number of users increases, indicating that the users should reduce their prefetch requests at high system load, so as not to overwhelm the server. This is contrary to selfish prefetching, where the prefetching threshold is pegged to the expected HP sojourn time and remains somewhat constant. There is significant value in optimizing the prefetching threshold for all users. Fig. 10 demonstrates that, at high system load, it can reduce the access cost by more than 25% over selfish prefetching. The design of methods to promote adaptive network optimal prefetching in a decentralized network with competing users [14] remains an open problem for future research.

5 Conclusions

Adaptive document access strategies are necessary in future wireless systems where heterogeneous access technologies are seamlessly integrated. Document prefetching can significantly improve the performance of such integrated systems. It gives the users faster response time and the service providers revenue from increased activity without loss of service due to instability. However, 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 dynamic control of the prefetching threshold, multiuser network-aware prefetching can scale well under heavy usage, even with many concurrent selfish users. 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, service, and mobility patterns. Finally, we have explored alternate prefetching schemes without queuing prioritization or with a network optimal prefetching thresh-

old, demonstrating quantitatively the importance of differentiated service and user cooperation.

References

[1] http://www.mozilla.org.

[2] M. Angermann. Analysis of speculative prefetching. *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. Gitzenis and N. Bambos. Power-controlled data prefetching/caching in wireless packet networks. In *Proc. of IEEE INFOCOM*, pages 1405–1414, June 2002.

[11] D. Goodman et al. Infostations: A new system model for data and messaging services. *IEEE 47th Vehicular Technology Conference*, 2:969–973, May 1997.

[12] Z. Jiang and L. Kleinrock. An adaptive network prefetching scheme. *IEEE Journal on Selected Areas in Communications*, 16(3):358–368, Apr. 1998.

[13] Z. Jiang and L. Kleinrock. Web prefetching in a mobile environment. *IEEE Personal Communications*, 5:25–34, Oct. 1998.

[14] J. Lau and B. Liang. Optimal pricing for selfish users and prefetching in heterogeneous wireless networks. Submitted for peer review, 2006.

[15] 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.

[16] V. N. Padmanabhan and J. C. Mogul. Using predictive prefetching to improve world wide web latency. *Comput. Commun. Rev.*, 26:22–36, 1996.

[17] N. U. Prabhu. *Foundations of Queueing Theory*. Kluwer Academic Publishers, 1997.

[18] S. M. Ross. *Stochastic Processes, 2nd Edition*. John Wiley & Sons, Inc., 1996.

[19] N. Tuah, M. Kumar, and S. Venkatesh. Resource-aware speculative prefetching in wireless networks. *ACM/Kluwer Wireless Networks*, 9:61–72, 2003.

[20] L. Yin and G. Cao. Adaptive power-aware prefetch in wireless networks. *IEEE Transactions on Wireless Communications*, (5):1648–1658, Sep 2004.

[21] A. H. Zahran, B. Liang, and A. Saleh. Signal threshold adaptation for vertical handoff in heterogeneous wireless networks. *ACM/Springer Mobile Networks and Applications (MONET), Special Issue on Soft Radio Enabled Heterogeneous Networks*, 11(4):625 – 640, Aug. 2006.