TTL Prediction Schemes and Effects of Inter-Update Time Distribution on Wireless Data Access*

Yuguang Fang[a], Zygmunt Haas[b], Ben Liang[b] and Yi-Bing Lin[c]

[a] Department of Electrical and Computer Engineering
University of Florida
435 Engineering Building, P.O.Box 116130
Gainesville, FL 32611, USA
Tel: +1-352-846-3043, Fax: +1-352-392-0044, Email: *fang@ece.ufl.edu*

[b] Wireless Networks Laboratory
School of Electrical Engineering
Cornell University
323 Frank Rhodes Hall
Ithaca, NY 14853, USA
Tel: +1-607-255-3454, Fax: +1-607-255-9072, Email: {*haas,liang*}@*ece.cornell.edu*

[c] Department of Computer Science & Information Engineering
National Chiao Tung University
Hsinchu, Taiwan
Tel: +886-3-5731842, Fax: +886-3-5724176, Email: *liny@csie.nctu.edu.tw*

Abstract Modern mobile networks such as GPRS and UMTS support wireless data applications. One successful example is the ever popular i-mode in Japan. Wireless data services (wireless Internet) become more important as more and more customers of handheld devices enjoy the convenience of the ubiquitous computing. To improve effective wireless data access, the time-to-live (TTL) management for data entries becomes important due to its use in effective caching design. In this paper, we study three TTL prediction schemes and investigate the effects of the inter-update time distribution on the wireless data access. Performance analysis is carried out via simulations as well as analytical modeling. We expect our results will be useful for the future wireless data access systems in which transmission power for mobile devices is more limited.

Keywords: Time-to-live (TTL), Weakly consistency, Wireless data, Caching.

* The work of Yuguang Fang was supported in part by the National Science Foundation Faculty Early Career Development Award under grant ANI-0093241 and the Office of Naval Research Young Investigator Award under grant N000140210464. The work of Zygmunt Haas and Ben Liang was partially funded by the ONR as part of the Multidisciplinary University Research Initiative (MURI) under the contract number N00014-00-1-0564 and by the NSF grants number ANI-9805094 and ANI-0081357. Ben Liang is now with the Department of Electrical and Computer Engineering, University of Toronto, Ontario, Canada. Yi-Bing Lin's work was sponsored in part by MOE Program for Promoting Academic Excellence of Universities under the grant number 89-E-FA04-1-4, FarEastone, IIS/Academia Sinica, and the Lee and MTI Center for Networking Research, NCTU.

1 Introduction

Modern mobile networks such as GPRS and UMTS ([8]) support wireless data applications. Examples such as the popular i-Mode in Japan have received a great deal of attention due to their success of some Internet services. The standard Wireless Application Protocol (WAP) ([6, 8]) are tailored for web accessing, which represents the first step towards the wireless Internet. In this environment (wireless Internet), a mobile customer may use a wireless handheld device to access data services from the application server through the mobile network. Mobile users have become fastest growing community of web users in the last few years. Many cellular phones are equipped with web browsing capabilities, and it is predicted that the number of wireless Internet devices will outnumber desktop computers by 2003. As another example, a user may access Web using Palm Pilot through a wireless data service such as Omnisky [11]. Omnisky is supported by Cellular Digital Packet Data (CDPD) [4, 8] with rates varying from 5Kbps to 13Kbps. To provide convenient services to mobile customers, web site personalization techniques have been developed [9] to automatically adapt and personalize web sites to mobile customers.

One of the challenging design tasks is how to make data ubiquitously available while minimizing the transmissions from mobile devices (to save battery power). An application running on the wireless handheld device may repeatedly access a data entry received from the application server. If the data entry is not sensitive to time, then the customer may access the data stored in the cache of the wireless handheld device instead of querying the application server, and the expensive wireless transmission overhead is reduced. Effective caching strategies should be used for such applications. If the data entry is sensitive to time, then the current data entry should be provided from the application server. In this case, it is better to push such a data entry to the mobile device before it is queried because the transmission power from mobile devices tend to be higher (more expensive) than the receiving power. Therefore, it is reasonable to handle time-sensitive wireless applications and time-insensitive applications in a different fashion. One way to do so is to use the timers (time-stamps or time-to-live) for data entries.

Some time-sensitive wireless applications can tolerate certain degree of inaccuracy (e.g., most web accesses and location dependent information in wireless applications). For this type of applications, we can set an expiration period t to predict when the data entry will be updated. During the period t, the data entry in the cache of the handheld device is used. When t expires, the next data access results in a query to the mobile network. In this case, the application is *weakly consistent* where the wireless handheld device may occasionally access the stale data. A mechanism is required to predict when a data entry expires. In Apache [1] and Squid [15], a time-to-live (TTL) interval t is defined for the data entry stored in the wireless handheld device. The TTL for a data entry is determined based on whether the data entry is modified due to either a mobile query or a server update, which leads to a simple TTL prediction algorithm.

Note that the TTL prediction mechanism is typically exercised with cache replacement such as LRU (least recently used) and LFU (least frequently used) [3] in a proxy cache for WWW accesses. Since the storage of a handheld device is limited, the wireless application may determine that no cache replacement algorithm is exercised for frequently accessed data (or they are likely to be replaced by infrequently accessed data). That is, when a wireless handheld device runs a particular application, some data used by this application are considered as "frequently accessed", and will always be kept in the handheld device until they expire. This is especially true for some location dependent services provisioned by the mobile operators. The customer may also enable a data entry as "frequently accessed", and the handheld device will not exercise cache replacement for this data entry until the frequently accessed indication is disabled. In Squid [15], the TTL option can be specified by users so that users can control caching for certain applications.

In this paper, we study three TTL prediction schemes for wireless data access. We also propose analytical and simulation models to provide performance analysis for the TTL prediction mechanisms. Our models are flexible enough to accommodate any fudge factor values used to generate the TTL interval. Since GPRS traffic reported by mobile operators indicated that the traditional web access patterns do not apply to the GPRS-based wireless data access, our model consider general distributions for data update and access. These distributions can be used to approximate data obtained from GPRS field operations/trials. Based on our model, we show how the inter-update time distribution affects the accuracy of TTL interval prediction.

2 TTL Prediction Schemes

In this section, we describe three schemes to determine the proper TTL interval when the handheld device queries the server. This series of schemes require increasing record-keeping of the history of the server inter-update intervals.

TTL Scheme #1: This scheme is based on the implementation of the Apache and Squid systems. When the handheld device queries the server, the server data entry either has been modified or remains the same as the cached one. In the former case, it is assumed that the server returns the updated data entry with a timestamp indicating when the data entry was last modified. In the latter case, the server returns a positive acknowledgment of the cache validity. Thus, the handheld always knows the time of the last server update. Let T_b be the difference between the time of the query and the time of the last server update. Then, in this scheme, the TTL interval is given by

$$T_{TTL_1} = c_f T_b \ , \tag{1}$$

where c_f is a system defined fudged factor.

TTL Scheme #2: In this scheme, it is assumed that the server remembers the length of the previous inter-update interval, denoted by T_p. When the handheld device queries the server, the server sends back the value of T_p as part of the reply. Then, the TTL interval for the current query is given by

$$T_{TTL_2} = c_f T_p \ . \tag{2}$$

TTL Scheme #3: As proposed in [14], a running average of the inter-update intervals can be obtained by a handheld device. If the server maintains this average, it can be sent to the handheld device during its query to the server. The exact method of obtaining this average, including the designs of the windowing duration and the weights of averaging, is outside of this paper's scope. Let T_e denote the average inter-update interval. Then, in this scheme, the TTL interval for the current query is given by

$$T_{TTL_3} = c_f T_e \ . \tag{3}$$

One main advantage of using TTL_1 is that, on the server side, it requires no extra equipment or processing overhead above what is already implemented in many of the currently deployed systems. For example, the time of the last update is built into the Hypertext Transfer Protocol (HTTP). However, as shown in the next section, this TTL prediction may not be as accurate as the other two schemes.

The schemes TTL_2 and TTL_3, on the other hand, require the server to remember its past updates and share that information with the handheld device. In particular, a server supporting TTL_3 may need to maintain a record of its long-term updating history. However, since T_e gives a more consistent estimate of the server's next updating time, we expect TTL_3 to outperform TTL_1 and TTL_2 in terms of data access cost.

In what follows, we will study the performance of these schemes and the effect of the server inter-update interval distribution on the cost of wireless data access.

3 Assumptions and Output Measures

In this section, we describe the assumptions used in this paper and the output measures used to evaluate the performance of TTL-interval prediction schemes. Consider the interval between two consecutive queries

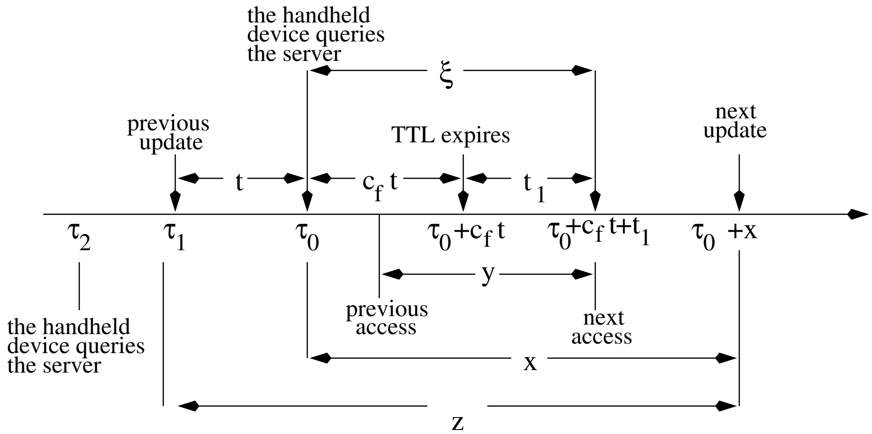

Figure 1: The Timing Diagram

from the wireless handheld device to the server. This interval is referred to as a *cycle*. In Figure 1, $[\tau_2, \tau_0)$ is a cycle. The access at the beginning of a cycle (e.g., τ_2 in Figure 1) results in a query to the server. During (τ_2, τ_0), the handheld device returns the cached copy to all accesses to the data entry. We assume that accesses to a data entry form a point process with general distribution. Furthermore, the inter-update intervals are assumed to be a random variable with a general distribution. Based on these assumptions, we consider the following primary output measures:

- The expected number $E[K_1]$ of *non-stale* accesses in a cycle: For a non-stale access, when the access occurs, the data entry in the cache is the same as that in the server. Note that the non-stale accesses include the one that results in the query to the server at the beginning of a cycle.
- The expected number $E[K]$ of accesses in a cycle: This number includes the stale and the non-stale accesses in the cache plus the access resulting in a query from the handheld device to the server (for the cycle $[\tau_2, \tau_0)$ in Figure 1, this query occurs at τ_2). Thus, $K \geq 1$ always holds.
- The probability β that when the handheld device queries the server, the data entry is valid (i.e., the data entry has not been modified since the last query).

It is clear that the handheld device communicates with the server for every $E[K]$ accesses. Based on $E[K_1]$ and $E[K]$, we can investigate the accuracy of TTL interval prediction through the *staleness ratio* p_s, which is the probability that the handheld device returns a stale data entry for an access. That is,

$$p_s = \frac{E[K] - E[K_1]}{E[K]} \tag{4}$$

Thus, we define the cost due to data staleness as

$$C_{stale} = \gamma p_s \ , \tag{5}$$

where γ represents the penalty of returning a stale data entry to the application.

Another output measure considered in this paper is the server query cost or wireless transmission cost C_{query}. Suppose that the cost of transmitting a data entry is one unit. We further denote δ as the cost for the handheld device to query the server without the data entry being transmitted. It is clear that $0 < \delta < 1$.

When the handheld device queries the server, if the data entry is valid then the server returns a positive acknowledgment with the cost δ. If the data has been modified, then the server returns the updated data entry to the handheld device and the transmission cost is one unit. Note that on average, a query to the server occurs for every $E[K]$ accesses. That is, the transmission costs for the $E[K]$-1 accesses in the cycle are 0. Thus, if we normalize the cost (e.g., wireless transmission delay) for a query with data transmission as one unit, then the server query cost per access can be expressed as

$$C_{query} = \frac{\delta\beta + (1-\beta)}{E[K]} = \frac{1-(1-\delta)\beta}{E[K]} \tag{6}$$

Then, the total cost per access is

$$C_{access} = C_{stale} + C_{query} = \gamma\frac{E[K]-E[K_1]}{E[K]} + \frac{1-(1-\delta)\beta}{E[K]} \ . \tag{7}$$

Obviously, shorter TTL intervals lead to smaller stale data probability p_s, and hence lower C_{stale}. However, shorter TTL intervals also create more queries to the server, which may lead to higher C_{query}. Ideally, if one has the exact information of the future server update times, the TTL should be set to expire at the next server update instant. However, in practice, one can only predict the proper TTL based on the known statistics. In the previous section, we present three TTL prediction schemes. Next, we study the effect of TTL selection on the cost of wireless data access.

4 Performance Study

Analytical models under certain simplified assumptions are possible. For example, for the third TTL prediction scheme, we are able to completely characterize the output measures. For the first and second TTL prediction schemes, we are also able to give some approximate analytical results. All analytical results and their derivations are presented in the appendices. In what follows, we will evaluate the performance of all three TTL prediction schemes and the effects of the inter-update probability distribution via both simulations as well as the analytical results.

4.1 Simulation Setup

Simulations are carried out in Matlab to evaluate the TTL prediction schemes for wireless data access. The following three distributions of the server inter-update intervals are studied:

- exponential
- Rayleigh
- deterministic

These three distributions represent a gradient of increasing memory level, from memoryless, in the case of the exponential distribution, to fully future knowledgeable, in the deterministic case.

The arrival process of accesses to a data entry is assumed to have a general distribution. Previous studies suggest that the arrivals of the Internet dial-up access connections can be described by a Poisson model [10]. However, published results based on the wireline Web access trace indicate that the user requests to a document on the Web do not follow a pure Poisson process [12]. Currently there is no access trace available for wireless data. Furthermore, wireless data access is affected by mobility and will not follow the access patterns observed in the wireline networks. In the following, we first consider the Poisson access stream for mean value analysis and then summarize our simulation results with general access patterns.

In each simulation, the mean of the inter-update intervals is set to have one time unit, the data accesses are assumed to occur five times as frequent as the server updates, and 10000 server updates are simulated.

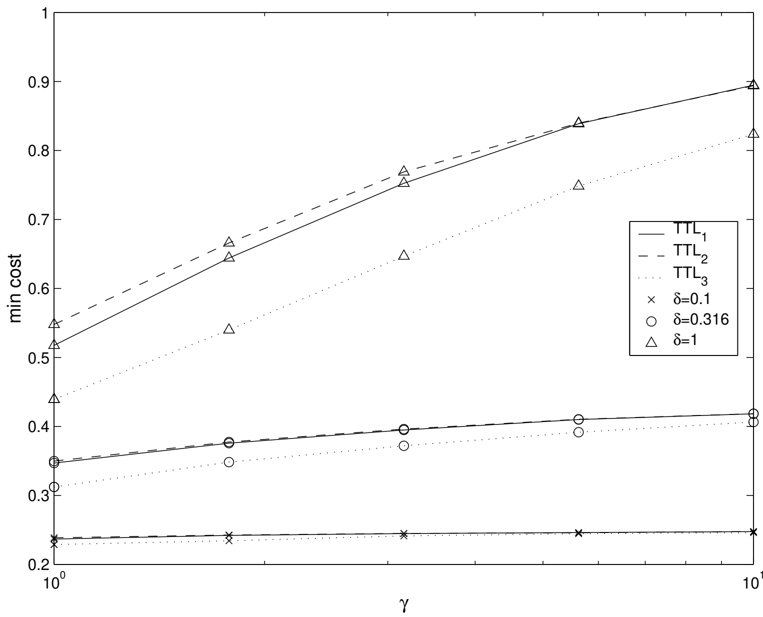

Figure 2(a): Exponential updating, optimal cost per access vs. γ

For each inter-update distribution, the three TTL schemes are applied separately. For each scheme, the fudge factor c_f is allowed to vary from 10^{-3} to 10^3. The optimal c_f is obtained through observations, and the corresponding optimal cost per data access is recorded.

In Figures 2(a)-4(b), we plot the optimal costs over δ and γ, where δ has the range between 0.1 and 1, and γ has the range between 1 and 10. In addition, our experiments have shown that, when δ is below 0.1, since the query cost is very low when the cache is valid, the cost per data access can be trivially minimized by querying the server at almost every data access. Furthermore, when γ is below 1, since the penalty against stale data access is very low, the cost per data access can be trivially minimized by seldom querying the server at all. Finally, when γ is above 10, since the penalty against stale data access is very high, the best scheme is, again, to query the server at almost every data access. All of these extreme cases are independent of the TTL scheme, and therefore, are not of interest in this study.

4.2 Comparison of TTL Schemes

As shown in Figures 2(a)-4(b), for all server inter-update distributions, TTL_3 outperforms TTL_1 by a substantial amount if δ is large. This is reasonable, considering TTL_3 requires the average of the inter-update intervals taken over a relatively long period.

For servers with the exponential inter-update distribution, which has the memoryless property, TTL_1 slightly outperforms TTL_2, especially when δ is large and when γ is small. However, for most other cases,TTL_2 outperforms TTL_1. In particular, for servers with the deterministic inter-update distribution, TTL_2 is equivalent to TTL_3. Thus, we can infer that the relative performance of TTL_1, as used in Apache and Squid, suffers as the memory level of the inter-update interval distribution increases.

4.3 Cost Sensitivity to δ and γ

Figures 2(a)-4(b) also suggest that, for all inter-update distributions and all three TTL schemes, the data access cost is very sensitive to the non-update query cost δ. Adjusting the labeling of the plot axes reveals

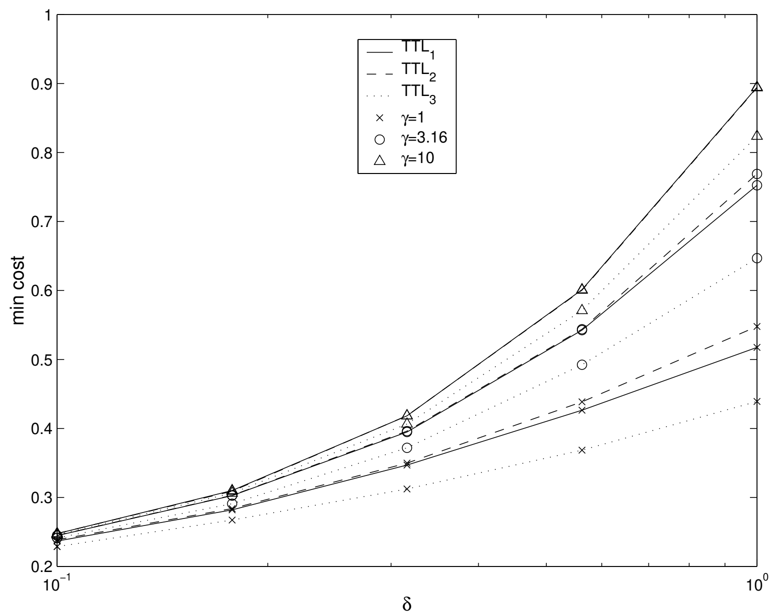

Figure 2(b): Exponential updating, optimal cost per access vs. δ

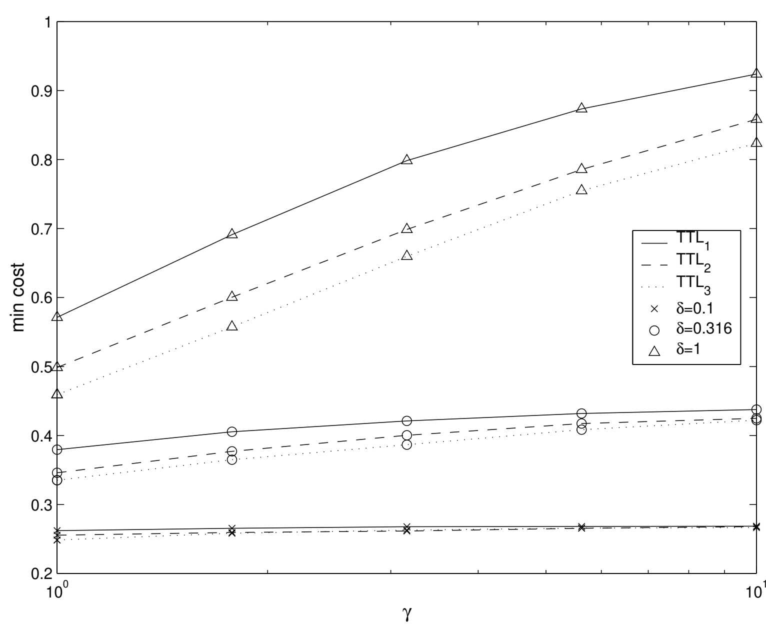

Figure 3(a): Rayleigh updating, optimal cost per access vs. γ

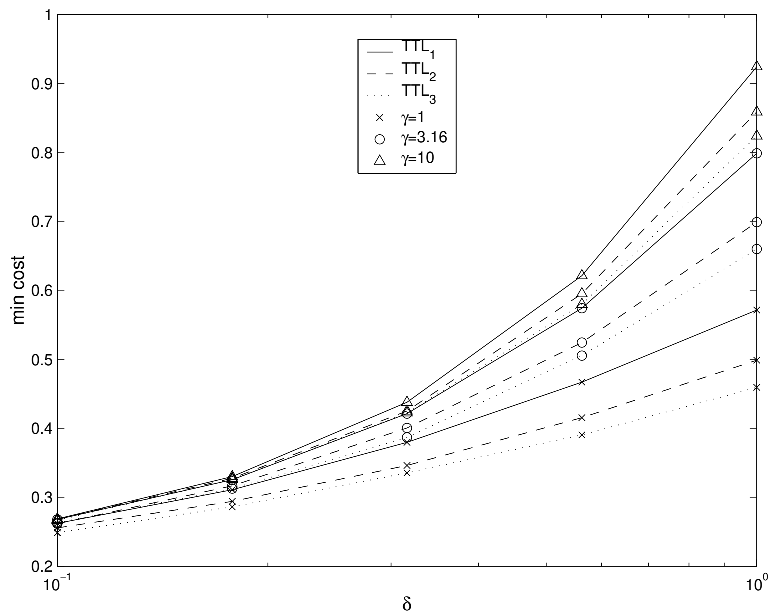

Figure 3(b): Rayleigh updating, optimal cost per access vs. δ

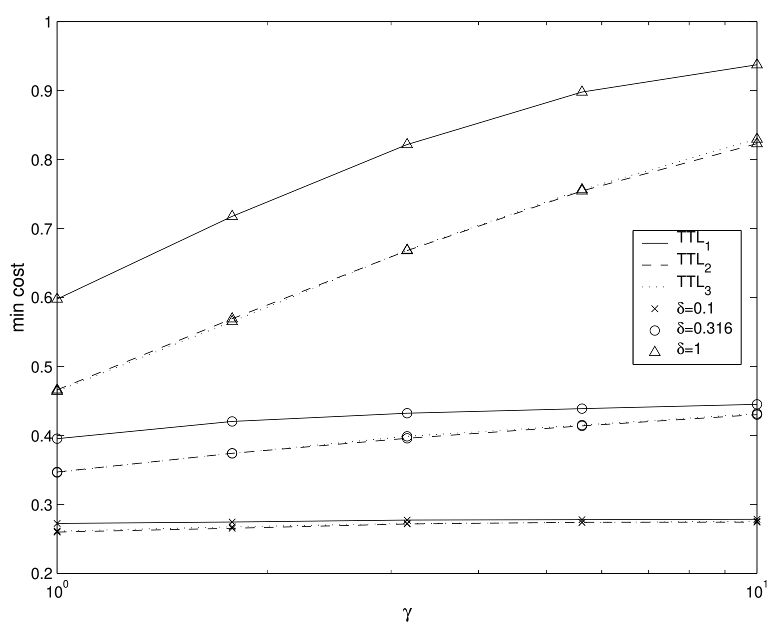

Figure 4(a): Deterministic updating, optimal cost per access vs. γ

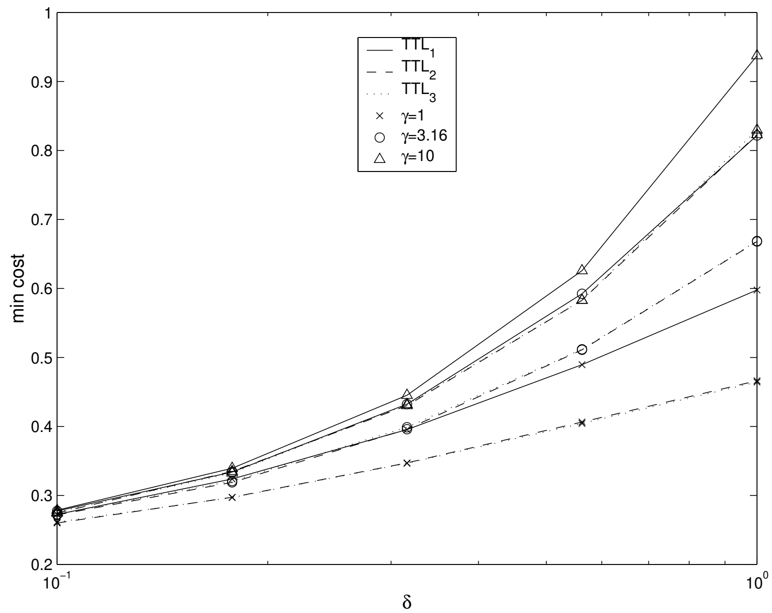

Figure 4(b): Deterministic updating, optimal cost per access vs. δ

that the cost is almost linearly increasing with δ. Therefore, it is important for a system designer to ensure that δ is kept small for cost effective data caching.

On the other hand, the data access cost is less sensitive to the stale data penalty γ. This is due to the optimal adjustment of the fudge factor c_f, such that queries are performed more frequently when γ is large. In systems where δ is small, one can afford to query the server more often, as the average cost per query is reduced in this case.

In fact, when δ is sufficiently small, all TTL schemes perform similarly, regardless of the inter-update distribution. As shown in these figures, when $\delta < 0.3$, the access cost of all three schemes is within 15% of each other. This can lead to the simplified numerical evaluations of some TTL schemes. For example, although TTL_1 is the most often used scheme in practical applications due to its minimal requirement on the server, it is generally hard to precisely analyze the performance of TTL_1. However, the performance of TTL_3 can be accurately analyzed numerically. Therefore, given a cached data access system that employs TTL_1, one can first obtain an accurate cost estimate of TTL_3 and then apply that result as a close approximation of the actual cost.

Towards this end, Appendix I provides an analytical framework for evaluating the cost of TTL_3. In addition, Appendix II gives two more methods that approximately compute the cost of TTL_1 under a set of assumed conditions.

4.4 Non-Poisson Data Accesses

The above simulations were repeated assuming data access streams with Rayleigh and deterministic inter-arrival intervals. In both cases, we observed the same patterns of the cost comparison among the different TTL prediction schemes and of the cost sensitivity to δ and γ. For brevity, the redundant simulation results are not presented.

5 Conclusions

In this paper, we study three prediction schemes for setting up time-to-live (TTL) for data entries in wireless data access. Due to the fact that the field data for current wireless data access is not available for the inter-update time, we use general distribution model for the inter-update time and carry out the performance analysis. The effects of the inter-update time distribution on the performance of wireless data access under the three TTL prediction schemes are investigated via the simulations as well as analytical modeling. The study shows that the TTL prediction scheme outperforms the currently used TTL prediction scheme used in Apache and Squid when the query cost from mobile device is high while most schemes perform similarly when the query cost is low. We expect our results will be useful for wireless data access systems in which mobile transmissions are more costly.

Appendix I: Numerical Analysis for TTL Scheme #3

In this section, we present an analytical framework for evaluating the performance of the TTL Scheme #3, where the TTL interval depends on the mean duration of the previous inter-update intervals.

Assume that the data accesses are a Poisson stream with rate λ. Let random variable Y represent the inter-access time, then Y has an exponential density function $f(y)$ where

$$f(y) = \lambda e^{-\lambda y} \quad \text{and} \quad E[Y] = \frac{1}{\lambda} \tag{8}$$

Let random variable Z represent the inter-update time, which has a general cumulative distribution function $F(z)$, density function $f(z)$, Laplace transform $f^*(s)$ and mean $E[Z] = 1/\mu$. Suppose that Z is a non-lattice random variable and $E[Z^2] < \infty$. Since the queries are independent of the server updates, the residual life X of Z has the cumulative distribution function $R(x)$, density function $r(x)$, and Laplace transform $r^*(s)$, where from [13]

$$r(x) = \mu\Big[1 - F(x)\Big] \tag{9}$$

$$r^*(s) = \left(\frac{\mu}{s}\right)\Big[1 - f^*(s)\Big] . \tag{10}$$

Suppose that T_{TTL} has probability density function $r_f(t_f)$, the cumulative distribution function $R(t_f)$, and the Laplace transform $r_f^*(s)$. Recall from Section 2 that, in this scheme, the TTL interval is

$$T_{TTL} = \frac{c_f}{\mu} , \tag{11}$$

where c_f is the fudge factor. Then,

$$r_f(t_f) = \delta\left(t - \frac{c_f}{\mu}\right) , \quad \text{and} \quad r_f^*(s) = e^{-\frac{c_f}{\mu}s} . \tag{12}$$

Consider the interval t_1 between when the TTL interval expires and when the next data access arrives. Let $f_1(t_1)$ and $f_1^*(s)$ denote the probability density function and Laplace transform of random variable t_1, respectively. From the memoryless property of the exponential distribution, t_1 has the same distribution as Y, thus we have

$$f_1(t_1) = \lambda e^{-\lambda t_1} \quad \text{and} \quad f_1^*(s) = \frac{\lambda}{s + \lambda} . \tag{13}$$

Let $\xi = T_{TTL}+t_1$, and let $f_\xi(\tau)$ and $f_\xi^*(s)$ denote its probability density function and Laplace transform. Then from (12)

$$f_\xi(\tau) = f_1\left(t - \frac{c_f}{\mu}\right) \quad \text{and} \quad f_\xi^*(s) = f_1^*(s)e^{-\frac{c_f}{\mu}s} . \tag{14}$$

The probability β is derived as

$$\begin{aligned}
\beta &= \Pr(T_{TTL} + t_1 \le X) \\
&= \Pr(\xi \le X) \\
&= \int_{x=0}^{\infty} \Pr(\xi \le x) r(x) dx \\
&= \int_{x=0}^{\infty} \left\{ \left(\frac{1}{2\pi i}\right) \int_{s=c-i\infty}^{c+i\infty} \left[\frac{f_\xi^*(s)}{s}\right] e^{sx} ds \right\} r(x) dx \\
&= \left(\frac{1}{2\pi i}\right) \int_{s=c-i\infty}^{c+i\infty} \left[\frac{f_\xi^*(s)}{s}\right] \left[\int_{x=0}^{\infty} r(x) e^{sx} dx\right] ds \\
&= \left(\frac{1}{2\pi i}\right) \int_{s=c-i\infty}^{c+i\infty} \left[\frac{f_\xi^*(s)}{s}\right] r^*(-s) ds \\
&= \left(\frac{1}{2\pi i}\right) \int_{s=c-i\infty}^{c+i\infty} \left[\frac{f_1^*(s)e^{-\frac{c_f}{\mu}s}}{s}\right] r^*(-s) ds \\
&= \left(\frac{\mu}{2\pi i}\right) \int_{s=c-i\infty}^{c+i\infty} \left[\frac{f_1^*(s)e^{-\frac{c_f}{\mu}s}}{-s^2}\right] [1 - f^*(-s)] ds \\
&= \mu \sum_{p \in \sigma_f} \operatorname*{Res}_{s=p} \left[\frac{f_1^*(s)e^{-\frac{c_f}{\mu}s}}{s^2}\right] [1 - f^*(-s)] ,
\end{aligned} \tag{15}$$

where σ_f denotes the set of poles of $f^*(-s)$ and $\mathrm{Res}_{s=p}$ denotes the residue at the pole $s = p$.

If Z, and hence X, is exponentially distributed, we have $f^*(s) = \mu/(s + \mu)$. Then, we have

$$\begin{aligned}
\beta &= \mu \operatorname*{Res}_{s=\mu} \left[\frac{f_1^*(s)e^{-\frac{c_f}{\mu}s}}{s^2}\right] \left(1 - \frac{\mu}{-s+\mu}\right) \\
&= \mu \left[\frac{f_1^*(s)e^{-\frac{c_f}{\mu}s}}{s}\right]\Bigg|_{s=\mu} \\
&= \frac{\lambda e^{-c_f}}{\mu + \lambda} .
\end{aligned}$$

Now we derive $E[K_1]$ as follows. Consider Figure 1. After τ_0, if the TTL interval expires earlier than the next update, then all data accesses occurring in $[\tau_0, \tau_0 + T_{TTL})$ are non-stale. On the other hand, if the TTL interval expires after the next update, then non-stale accesses occur in period $[\tau_0, \tau_0 + x]$. Thus, we conclude that during a cycle, the non-stale accesses occur in the period $T_{min} = E[\min(T_{TTL}, X)]$. Since the accesses are a Poisson stream, from [13],

$$E[K_1] = 1 + \frac{E[T_{min}]}{E[Y]} \tag{16}$$

The probability density function for T_{min} is

$$\begin{aligned} r_{min}(t_m) &= -\frac{d}{dt_m}\Pr(\min\{T_{TTL}, X\} \geq t_m) = -\frac{d}{dt_m}\left[\Pr(T_{TTL} \geq t_m)\Pr(X \geq t_m)\right] \\ &= \int_{x=t_m}^{\infty} r_f(t_m)r(x)dx + \int_{t_f=t_m}^{\infty} r_f(t_f)r(t_m)dt_f \end{aligned}$$

The Laplace transform for T_{min} is

$$\begin{aligned} r^*_{min}(s) &= \int_0^{\infty} r_f(t)\left[\int_t^{\infty} r(\tau)d\tau\right]e^{-st}dt + \int_0^{\infty} r(t)\left[\int_t^{\infty} r_f(\tau)d\tau\right]e^{-st}dt \\ &= \int_0^{\infty} r_f(t)\left[\frac{1}{2\pi i}\int_{\sigma-i\infty}^{\sigma+i\infty}\frac{1-r^*(z)}{z}e^{zt}dz\right]e^{-st}dt + \int_0^{\infty} r(t)\left[\frac{1}{2\pi i}\int_{\sigma-i\infty}^{\sigma+i\infty}\frac{1-r^*_f(z)}{z}e^{zt}dz\right]e^{-st}dt \\ &= \frac{1}{2\pi i}\int_{\sigma-i\infty}^{\sigma+i\infty}\frac{1-r^*(z)}{z}\left[\int_0^{\infty} r_f(t)e^{-(s-z)t}dt\right]dz + \frac{1}{2\pi i}\int_{\sigma-i\infty}^{\sigma+i\infty}\frac{1-r^*_f(z)}{z}\left[\int_0^{\infty} r(t)e^{-(s-z)t}dt\right]dz \\ &= \frac{1}{2\pi i}\int_{\sigma-i\infty}^{\sigma+i\infty}\frac{1-r^*(z)}{z}r^*_f(s-z)dz + \frac{1}{2\pi i}\int_{\sigma-i\infty}^{\sigma+i\infty}\frac{1-r^*_f(z)}{z}r^*(s-z)dz \end{aligned} \tag{17}$$

where σ is a sufficiently small positive number.

Applying the Residue Theorem, we obtain the expected value $E[T_{min}]$

$$\begin{aligned} E[T_{min}] &= -r^{*(1)}_{min}(0) \\ &= -\frac{1}{2\pi i}\int_{\sigma-i\infty}^{\sigma+i\infty}\frac{1-r^*(z)}{z}r^{*(1)}_f(-z)dz - \frac{1}{2\pi i}\int_{\sigma-i\infty}^{\sigma+i\infty}\frac{1-r^*_f(z)}{z}r^{*(1)}(-z)dz \\ &= \sum_{p\in\sigma_{rf}}\operatorname*{Res}_{s=p}\frac{1-r^*(s)}{s}r^{*(1)}_f(-s) + \sum_{p\in\sigma_r}\operatorname*{Res}_{s=p}\frac{1-r^*_f(s)}{s}r^{*(1)}(-s)\ , \end{aligned} \tag{18}$$

where σ_r is the set of poles of $r^*(-s)$ and σ_{rf} is the set of poles of $r^*_f(-s)$ in the strict right half complex plane, and where we also have used the following fact that the derivative $g^{(1)}(s)$ of function $g(s)$ will share the same set of poles except the multiplicities We can express $E[T_{min}]$ in terms of function $f^*(s)$ via the equations (9)–(12), however, we omit such expressions here due to its complexity.

If the inter-update time Z is exponential, then $f^*(s) = r^*(s) = \mu/(s+\mu)$ and $r^{*(1)}(-s) = -\mu/(s-\mu)^2$. From equation (18), we obtain

$$\begin{aligned} E[T_{min}] &= -\frac{c_f}{\mu}\sum_{p\in\sigma_{rf}}\operatorname*{Res}_{s=p}\frac{1}{s+\mu}e^{\frac{c_f}{\mu}s} + \sum_{p\in\sigma_r}\operatorname*{Res}_{s=p}\frac{1-e^{-\frac{c_f}{\mu}s}}{s}r^{*(1)}(-s) \\ &= \frac{c_f}{\mu}\operatorname*{Res}_{s=-\mu}\frac{1}{s+\mu}e^{\frac{c_f}{\mu}s} - \operatorname*{Res}_{s=\mu}\frac{1-e^{-\frac{c_f}{\mu}s}}{s}\cdot\frac{\mu}{(s-\mu)^2} \\ &= \frac{c_f}{\mu}e^{\frac{c_f}{\mu}s}\bigg|_{s=-\mu} - \mu\frac{d}{ds}\left(\frac{1-e^{-\frac{c_f}{\mu}s}}{s}\right)\bigg|_{s=\mu} \\ &= \frac{c_f}{\mu}e^{-c_f} + \frac{1-(1+c_f)e^{-c_f}}{\mu} \\ &= \frac{1-e^{-c_f}}{\mu}\ , \end{aligned} \tag{19}$$

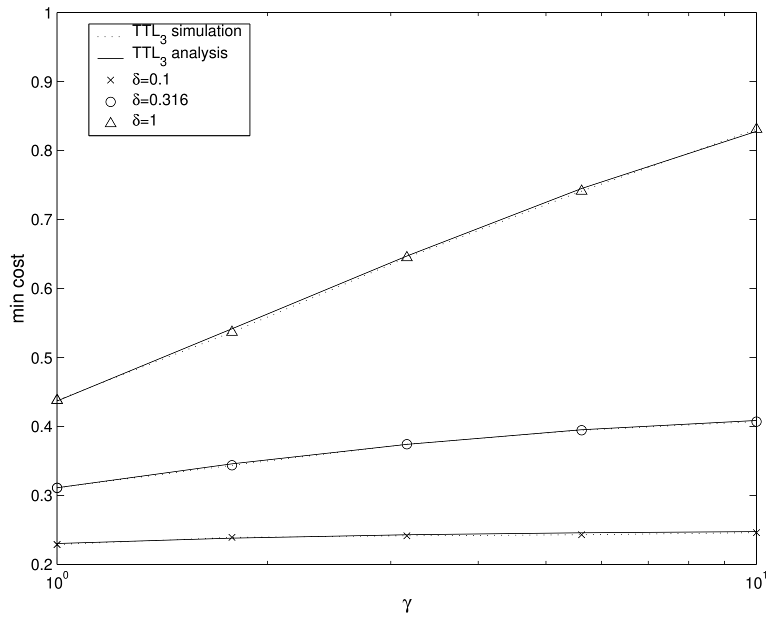

Figure 5(a): TTL_3 analysis verification, optimal cost per access vs. γ

which is consistent with the direct computation.

The cost per wireless data access, in the case of TTL prediction scheme $\#3$, is computed based on the preceding analytical framework and compared with the simulation results. Figures 5(a) and 5(b) illustrate this comparison in scenarios with parameters the same as those in Figures 2(a) and 2(b). These plots validate the simulation model against the analytical approach.

Appendix II: Approximating the Cost of TTL Schemes #1 and #2

This section describes an analysis framework that provides the approximate cost of wireless data access using TTL schemes #1 and #2. Since the analysis of these two schemes are very similar, in what follows, we concentrate on the analysis of TTL scheme #1. For scheme #2, we only provide brief pointers where the analysis differs.

In this approximation, we have made the following two assumptions:

Assumption #1: The handheld queries to the server are sufficiently independent of the server updates, such that the queries can be considered random observers of the inter-update interval;

Assumption #2: At the time of every query, the TTL interval is independent of the forward residual life of the current server inter-update interval.

These assumptions are reasonable in systems with near-exponential server inter-update intervals and relatively infrequent queries.

Consider the timing diagram in Figure 1. Assume that the data accesses are a Poisson stream with rate λ. Let random variable Y represent the inter-access time, then Y has an exponential density function $f(y)$ where

$$f(y) = \lambda e^{-\lambda y} \quad \text{and} \quad E[Y] = \frac{1}{\lambda} \tag{20}$$

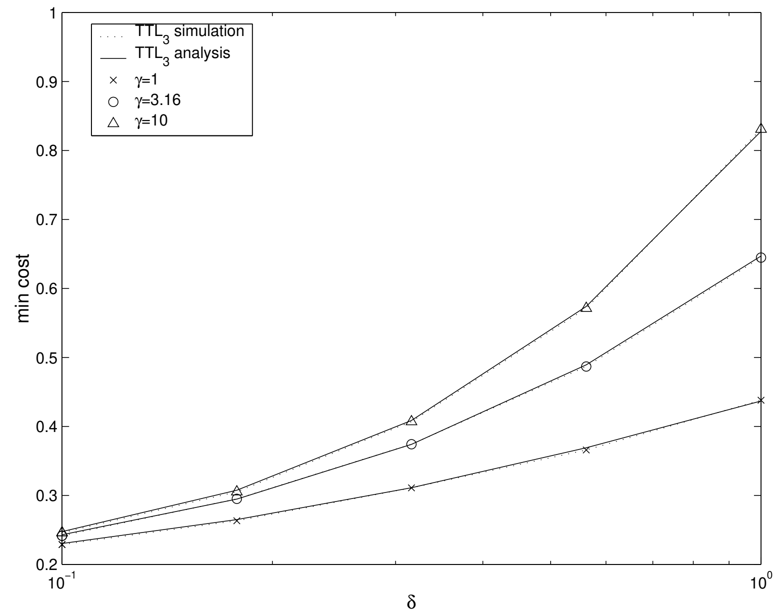

Figure 5(b): TTL_3 analysis verification, optimal cost per access vs. δ

Let random variable Z represent the inter-update time, which has a general cumulative distribution function $F(z)$, density function $f(z)$, Laplace transform $f^*(s)$ and mean $E[Z] = 1/\mu$. Suppose that Z is a non-lattice random variable and $E[Z^2] < \infty$, then the residual life X of Z has the cumulative distribution function $R(x)$, density function $r(x)$, and Laplace transform $r^*(s)$, where from [13]

$$r(x) = \mu\Big[1 - F(x)\Big] \tag{21}$$

$$r^*(s) = \left(\frac{\mu}{s}\right)\Big[1 - f^*(s)\Big] \tag{22}$$

Let random variable T be the interval between the previous update and when the handheld device queries the server. In Figure 1, $T = t = \tau_0 - \tau_1$. Since the queries to the server are random observer of the inter-update interval Z, T is the reverse residual life of Z. From the reversibility property of residual life [13], T has the same distribution as X (the residual life of Z). In Apache [1], the TTL interval is computed as $T_{TTL} = c_f T$, where c_f is the fudge factor. In Figure 1, the TTL interval is $t_f = c_f t$. Suppose that T_{TTL} has probability density function $r_f(t_f)$, the cumulative distribution function $R(t_f)$, and the Laplace transform $r_f^*(s)$. As previously discussed, T has the density function $r(t_f)$ and Laplace transform $r^*(s)$, and[1]

$$r_f(t_f) = \left(\frac{1}{c_f}\right) r\left(\frac{t_f}{c_f}\right) \quad \text{and} \quad r_f^*(s) = r^*(c_f s) \tag{23}$$

Consider the interval t_1 between when the TTL interval expires and when the next data access arrives. Let $f_1(t_1)$ and $f_1^*(s)$ denote the probability density function and Laplace transform of random variable t_1,

[1]For TTL scheme #2, we have

$$r_f(t_f) = f(t_f) \quad \text{and} \quad r_f^*(s) = f^*(s)\,.$$

The rest follows exactly as the analysis of TTL scheme #1 with the above replacing (23).

respectively. From the memoryless property of the exponential distribution, t_1 has the same distribution as Y, thus we have

$$f_1(t_1) = \lambda e^{-\lambda t_1} \quad \text{and} \quad f_1^*(s) = \frac{\lambda}{s+\lambda} \tag{24}$$

Let $\xi = T_{TTL} + t_1$ (in Figure 1, $T_{TTL} = t_f = c_f t$), and let $f_\xi(\tau)$ and $f_\xi^*(s)$ denote the probability density function and Laplace transform. Then from (23) and (24)

$$f_\xi(\tau) = \int_{t_f=0}^{\tau} r_f(t_f) f_1(\tau - t) dt_f \quad \text{and} \quad f_\xi^*(s) = r_f^*(s) f_1^*(s) \tag{25}$$

The probability β is derived as

$$\begin{aligned}
\beta &= \Pr(T_{TTL} + r_1 \le X) \\
&= \Pr(\xi \le X) \\
&= \int_{x=0}^{\infty} \Pr(\xi \le x) r(x) dx \\
&= \int_{x=0}^{\infty} \left\{ \left(\frac{1}{2\pi i}\right) \int_{s=c-i\infty}^{c+i\infty} \left[\frac{f_\xi^*(s)}{s}\right] e^{sx} ds \right\} r(x) dx \\
&= \left(\frac{1}{2\pi i}\right) \int_{s=c-i\infty}^{c+i\infty} \left[\frac{f_\xi^*(s)}{s}\right] \left[\int_{x=0}^{\infty} r(x) e^{sx} dx\right] ds \\
&= \left(\frac{1}{2\pi i}\right) \int_{s=c-i\infty}^{c+i\infty} \left[\frac{f_\xi^*(s)}{s}\right] r^*(-s) ds \\
&= \left(\frac{1}{2\pi i}\right) \int_{s=c-i\infty}^{c+i\infty} \left[\frac{r_f^*(s) f_1^*(s)}{s}\right] r^*(-s) ds \\
&= \left(\frac{(\mu^2/c_f)}{2\pi i}\right) \int_{s=c-i\infty}^{c+i\infty} \left[\frac{(1 - f^*(c_f s)) f_1^*(s)}{-s^3}\right] [1 - f^*(-s)] ds \\
&= \frac{\mu^2}{c_f} \sum_{p \in \sigma_f} \operatorname*{Res}_{s=p} \left[\frac{(1 - f^*(c_f s)) f_1^*(s)}{s^3}\right] [1 - f^*(-s)]
\end{aligned} \tag{26}$$

where σ_f denotes the set of poles of $f^*(-s)$ and $\mathrm{Res}_{s=p}$ denotes the residue at the pole $s = p$. If X is exponentially distributed, then we have $f^*(s) = \mu/(s+\mu)$, hence we have

$$\begin{aligned}
\beta &= \frac{\mu^2}{c_f} \operatorname*{Res}_{s=\mu} \left[\frac{[1 - f^*(c_f s)] f_1^*(s)}{s^3}\right] \left(1 - \frac{\mu}{-s+\mu}\right) \\
&= \frac{\mu^2}{c_f} \left[\frac{[1 - f^*(c_f s)] f_1^*(s)}{s^2}\right]\Bigg|_{s=\mu} \\
&= \left(\frac{1}{1+c_f}\right) \left(\frac{\lambda}{\mu+\lambda}\right)
\end{aligned}$$

Now we derive $E[K_1]$ as follows. Consider Figure 1. After τ_0, if the TTL interval expires earlier than the next update, then all data accesses occurring in $[\tau_0, \tau_0 + c_f t)$ are non-stale. On the other hand, if the TTL interval expires after the next update, then non-stale accesses occur in period $[\tau_0, \tau_0 + x]$. Thus, we conclude that during a cycle, the non-stale accesses occur in the period $T_{min} = E[\min(T_{TTL}, X)]$. Since the accesses are a Poisson stream, from [13],

$$E[K_1] = 1 + \frac{E[T_{min}]}{E[Y]} \tag{27}$$

The probability density function for T_{min} is

$$\begin{aligned} r_{min}(t_m) &= -\frac{d}{dt_m}\Pr(\min\{T_{TTL}, X\} \geq t_m) = -\frac{d}{dt_m}\left[\Pr(T_{TTL} \geq t_m)\Pr(X \geq t_m)\right] \\ &= \int_{t_f=t_m}^{\infty} r_f(t_f) r(t_m) dt + \int_{x=t_m}^{\infty} r_f(t_m) r(x) dx \end{aligned}$$

The Laplace transform for T_{min} is

$$\begin{aligned} r^*_{min}(s) &= \int_0^\infty r_f(t)\left[\int_t^\infty r(\tau)d\tau\right]e^{-st}dt + \int_0^\infty r(t)\left[\int_t^\infty r_f(\tau)d\tau\right]e^{-st}dt \\ &= \int_0^\infty r_f(t)\left[\frac{1}{2\pi i}\int_{\sigma-i\infty}^{\sigma+i\infty}\frac{1-r^*(z)}{z}e^{zt}dz\right]e^{-st}dt + \int_0^\infty r(t)\left[\frac{1}{2\pi i}\int_{\sigma-i\infty}^{\sigma+i\infty}\frac{1-r_f^*(z)}{z}e^{zt}dz\right]e^{-st}dt \\ &= \frac{1}{2\pi i}\int_{\sigma-i\infty}^{\sigma+i\infty}\frac{1-r^*(z)}{z}\left[\int_0^\infty r_f(t)e^{-(s-z)t}dt\right]dz + \frac{1}{2\pi i}\int_{\sigma-i\infty}^{\sigma+i\infty}\frac{1-r_f^*(z)}{z}\left[\int_0^\infty r(t)e^{-(s-z)t}dt\right]dz \\ &= \frac{1}{2\pi i}\int_{\sigma-i\infty}^{\sigma+i\infty}\frac{1-r^*(z)}{z}r_f^*(s-z)dz + \frac{1}{2\pi i}\int_{\sigma-i\infty}^{\sigma+i\infty}\frac{1-r_f^*(z)}{z}r^*(s-z)dz \end{aligned} \tag{28}$$

where σ is a sufficiently small positive number. Applying the Residue Theorem, we obtain the expected value $E[T_{min}]$

$$\begin{aligned} E[T_{min}] &= -r^{*(1)}_{min}(0) \\ &= -\frac{1}{2\pi i}\int_{\sigma-i\infty}^{\sigma+i\infty}\frac{1-r^*(z)}{z}r_f^{*(1)}(-z)dz - \frac{1}{2\pi i}\int_{\sigma-i\infty}^{\sigma+i\infty}\frac{1-r_f^*(z)}{z}r^{*(1)}(-z)dz \\ &= \sum_{p\in\sigma_{rf}}\operatorname*{Res}_{s=p}\frac{1-r^*(s)}{s}r_f^{*(1)}(-s) + \sum_{p\in\sigma_r}\operatorname*{Res}_{s=p}\frac{1-r_f^*(s)}{s}r^{*(1)}(-s) \\ &= \sum_{p\in\sigma_r}\operatorname*{Res}_{s=p/c_f}\frac{1-r^*(s)}{s}r_f^{*(1)}(-s) + \sum_{p\in\sigma_r}\operatorname*{Res}_{s=p}\frac{1-r_f^*(s)}{s}r^{*(1)}(-s) \end{aligned} \tag{29}$$

where σ_r is the set of poles of $r^*(-s)$ and σ_{rf} is the set of poles of $r_f^*(-s)$ in the strict right half complex plane, and where we also have used the following facts that the derivative $g^{(1)}(s)$ of function $g(s)$ will share the same set of poles except the multiplicities and that $r_f^*(-s)$ has poles in the following form: p/c_f for $p \in \sigma_r$. We can express $E[T_{min}]$ in terms of function $f^*(s)$ via the equations (21)–(23), however, we omit such expressions here due to its complexity. We also notice that $r^*(-s)$, $f^*(-s)$ and their derivatives share the same set of poles in the strict right half complex plane except the multiplicities, we can express $E[T_{min}]$ as follows:

$$E[T_{min}] = \sum_{p\in\sigma_f}\operatorname*{Res}_{s=p/c_f}\frac{1-r^*(s)}{s}r_f^{*(1)}(-s) + \sum_{p\in\sigma_f}\operatorname*{Res}_{s=p}\frac{1-r_f^*(s)}{s}r^{*(1)}(-s) \tag{30}$$

If the inter-update time Z is exponential, then $f^*(s) = r^*(s) = \mu/(s+\mu)$ and $r_f^*(s) = \mu/(c_f s + \mu) = \mu_f/(s+\mu_f)$ where $\mu_f = \mu/c_f$. From equation (30), we obtain

$$\begin{aligned}
E[T_{min}] &= \operatorname*{Res}_{s=\mu_f} \frac{1-\mu/(s+\mu)}{s}\left(-\frac{\mu_f}{(-s+\mu_f)^2}\right) + \operatorname*{Res}_{s=\mu} \frac{1-\mu_f/(s+\mu_f)}{s}\left(-\frac{\mu}{(-s+\mu)^2}\right) \\
&= -\operatorname*{Res}_{s=\mu_f} \frac{1}{s+\mu}\cdot\frac{\mu_f}{(s-\mu_f)^2} - \operatorname*{Res}_{s=\mu} \frac{1}{s+\mu_f}\cdot\frac{\mu}{(s-\mu)^2} \\
&= -\mu_f \frac{d}{ds}\left(\frac{1}{s+\mu}\right)\Bigg|_{s=\mu_f} - \mu\frac{d}{ds}\left(\frac{1}{s+\mu_f}\right)\Bigg|_{s=\mu} \\
&= \frac{1}{\mu+\mu_f} = \frac{c_f}{(1+c_f)\mu} \qquad (31)
\end{aligned}$$

which is consistent with the direct computation.

Next, we present another approach to compute the probability β and the expectation $E[T_{min}]$ for scheme #1. It is well-known that ([7]) the phase-type distributions (PH) are dense in the set of all distributions in $[0, \infty)$, i.e., any distribution of a nonnegative random variable can be approximated by Phase-type distributions. The exponential distribution, the Erlang distributions, the hyper-exponential distribution and the hyper-Erlang distribution are all special cases of PH distributions. The advantage of PH distributions is that most computations are reduced to matrix manipulations. A PH distribution is the distribution of the time till absorption into the absorbing state 0 in a finite state Markov chain with states $\{0, 1, 2, \ldots, n\}$ and with initial probability vector (α_0, α) and infinitesimal generator

$$Q = \begin{pmatrix} 0 & \mathbf{0} \\ \mathbf{t} & T \end{pmatrix}$$

where α is row vector of size n and T is an $n \times n$ matrix. It can be shown ([7]) that this distribution can be uniquely determined by the (α, T), so we say a random variable X is $PH(\alpha, T)$ if X is PH distributed with parameter (α, T). In fact, a random variable with $PH(\alpha, T)$ has the following probability density function

$$f(x) = -\alpha \exp(Tx) T \mathbf{1}_T, \;\; x \geq 0$$

where $\mathbf{1}_T$ is the column vector of all 1's with the same dimension as the matrix T (i.e., if T is an $n \times n$ matrix, $\mathbf{1}_T$ will be $n-$dimensional vector). We need the following result ($\otimes$ indicates the Kronecker product and $\oplus$ denotes the Kronecker sum):

Lemma ([7],[2])

(1) Assume that $F(x)$ is the cumulative distribution function of a random variable with $PH(\alpha, T)$ with expectation $1/\mu$, then the distribution $\mu[1 - F(x)]$ is also PH distributed with $PH(\pi, T)$ where $\pi = (\alpha T^{-1}\mathbf{1}_T)^{-1}\alpha T^{-1}$.

(2) Assume that random variables X and Y are independent with $PH(\alpha, T)$ and $PH(\nu, S)$, respectively, then the random variable $\min\{X, Y\}$ is also PH distributed with $PH(\gamma, C)$ where

$$\gamma = \alpha \otimes \nu, \;\; C = T \oplus S$$

(3) Assume that random variables X and Y are independent with $PH(\alpha, T)$ and $PH(\nu, S)$, respectively, then the random variable $X + Y$ is also PH distributed with $PH(\gamma, C)$ where

$$\gamma = [\alpha, \alpha_0 \nu], \;\; C = \begin{pmatrix} T & \mathbf{t}\nu \\ \mathbf{0} & S \end{pmatrix}$$

where $\alpha_0 = 1 - \alpha \mathbf{1}_T$, $\mathbf{t} = -T\mathbf{1}_T$.

(4) Assume that random variables X and Y are independent with $PH(\alpha, T)$ and $PH(\nu, S)$, respectively, then

$$\Pr(X \leq Y) = (\nu \otimes \alpha)(-S \oplus T)^{-1}(\mathbf{1}_S \otimes (-T\mathbf{1}_T)).$$

Now we assume that the inter-update time is $PH(\alpha, T)$, then from Lemma (1), the residual life X is $PH(\pi, T)$ where $\pi = (\alpha T^{-1}\mathbf{1}_T)^{-1}\alpha T^{-1}$, thus we have

$$r(x) = -\pi \exp(Tx)T\mathbf{1}_T$$

and

$$r_f(x) = \frac{1}{c_f} r\left(\frac{x}{c_f}\right) = -\pi \exp((T/c_f)x)(T/c_f)\mathbf{1}_T$$

which is $PH(\pi, T/c_f)$.

We first compute β. Since T_{TTL} is $PH(\pi, T/c_f)$ and r_1 is exponentially distributed with $PH(1, -\lambda)$, from Lemma, we conclude that $T_{TTL} + r_1$ is PH distributed with $PH(\gamma_\beta, C_\beta)$ where

$$\gamma_\beta = (\alpha, 1 - \alpha\mathbf{1}_T),\ C_\beta = \begin{pmatrix} T/c_f & -(T/c_f)\mathbf{1}_T \\ 0 & -\lambda \end{pmatrix}.$$

Applying Lemma (4), we obtain

$$\begin{aligned} \beta &= (\pi \otimes \gamma_\beta)(-T \oplus C_\beta)^{-1}(\mathbf{1}_T \otimes (-C_\beta \mathbf{1}_{C_\beta})) \\ &= (\pi \otimes \gamma_\beta)(-T \oplus C_\beta)^{-1}\left(\mathbf{1}_T \otimes \begin{pmatrix} \mathbf{0} \\ \lambda \end{pmatrix}\right) \end{aligned} \tag{32}$$

If the inter-update time Z is exponentially distributed, we have $\alpha = 1$, $T = -\mu$, $\pi = 1$, $\gamma_\beta = (1, 0)$, $C_\beta = \begin{pmatrix} -\mu/c_f & \mu/c_f \\ 0 & -\lambda \end{pmatrix}$. Applying (32) we obtain (I_A denotes the identity matrix with the same dimension as a matrix A)

$$\begin{aligned} \beta &= (\pi \otimes \gamma_\beta)(-T \otimes I_{C_\beta} - I_T \otimes C_\beta)^{-1}\left(\mathbf{1}_T \otimes \begin{pmatrix} \mathbf{0} \\ \lambda \end{pmatrix}\right) \\ &= (1 \otimes (1, 0))\left(-\begin{pmatrix} -\mu & 0 \\ 0 & \mu \end{pmatrix} - \begin{pmatrix} -\mu/c_f & \mu/c_f \\ 0 & -\lambda \end{pmatrix}\right)^{-1}\left(1 \otimes \begin{pmatrix} \mathbf{0} \\ \lambda \end{pmatrix}\right) \\ &= \begin{pmatrix} 1 & 0 \end{pmatrix}\begin{pmatrix} \frac{1+c_f}{c_f}\mu & -\frac{1}{c_f}\mu \\ 0 & \lambda + \mu \end{pmatrix}^{-1}\begin{pmatrix} 0 \\ \lambda \end{pmatrix} \\ &= \frac{1}{1+c_f}\frac{\lambda}{\lambda+\mu} \end{aligned}$$

which is the same as what was previously obtained.

Next, we compute $E[T_{min}]$. Again since T_{TTL} is $PH(\pi, T/c_f)$ and X is $PH(\pi, T)$, where $\pi = (\alpha T^{-1}\mathbf{1}_T)^{-1}\alpha T^{-1}$. From Lemma, we conclude that T_{min} is also PH distributed with $PH(\gamma_e, C_e)$ where

$$\gamma_e = \pi \otimes \pi,\ \ C_e = T \oplus (T/c_f) = T \otimes I_T + I_T \otimes (T/c_f) \tag{33}$$

Therefore, from the property of PH distribution, we obtain the expectation for T_{min} as follows

$$E[T_{min}] = \gamma_e(-C^{-1})\mathbf{1}_{C_e} = -(\pi \otimes \pi)[T \oplus (T/c_f)]^{-1}\mathbf{1}_{C_e} \tag{34}$$

If the inter-update time Z is Erlang distributed with parameter (m, μ), then we know it has the following representation $PH(\alpha, T)$ where

$$\alpha = (1, 0, \ldots, 0),\ T = -m\mu \begin{pmatrix} 1 & -1 & 0 & \cdots & 0 \\ 0 & 1 & -1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & -1 \\ 0 & 0 & \cdots & 0 & 1 \end{pmatrix} = -m\mu T_0 \tag{35}$$

Then, $r(t)$ will be $PH(\pi, T)$ with

$$\pi = (\alpha T^{-1}\mathbf{1}_T)^{-1}\alpha T^{-1} = \frac{1}{m}\begin{pmatrix} 1 & 1 & \cdots & 1 \end{pmatrix}.$$

Thus, we have (A^T denotes the matrix transpose of A)

$$\begin{aligned} E[T_{min}] &= -\frac{1}{m^2}\mathbf{1}^T(T \oplus (T/c_f))^{-1}\mathbf{1} \\ &= \frac{1}{m^3\mu}\mathbf{1}^T(T_0 \oplus (T_0/c_f))^{-1}\mathbf{1} \\ &= \frac{1}{m^3\mu}\sum_{i,j}\left[(T_0 \oplus (T_0/c_f))^{-1}\right]_{ij} \end{aligned} \tag{36}$$

where $\mathbf{1}$ is a column vector of all 1's with appropriate dimension for the matrix multiplication. When $m = 1$, i.e., Z is exponentially distributed, we have $T_0 = 1$, hence

$$E[T_{min}] = \frac{1}{1^3\mu}(1 + 1/c_f)^{-1} = \frac{c_f}{1 + c_f}\frac{1}{\mu},$$

which is the same as we obtained earlier.

If Z is hyper-Erlang distribution with the following probability density function ([5]):

$$f(t) = \sum_{i=1}^{M} p_i \frac{(m_i\mu_i)^{m_i} t^{m_i - 1}}{(m_i - 1)!} e^{-m_i\mu_i t},\ \ p_i \geq 0,\ \ \sum_{i=1}^{M} p_i = 1,\ \ \frac{1}{\mu} = \sum_{i=1}^{M}\frac{p_i}{\mu_i}\ \ M > 0 \tag{37}$$

From a result in [7], we know that the hyper-Erlang distribution $f(t)$ has the following representation $PH(\alpha_{he}, T_{he})$ where

$$\begin{aligned} \alpha_{he} &= \begin{pmatrix} p_1\alpha_1 & p_2\alpha_2 & \cdots & p_M\alpha_M \end{pmatrix} \\ \alpha_i &= \begin{pmatrix} 1 & 0 & \cdots & 0 \end{pmatrix},\ i = 1, 2, \ldots, M \\ T_{he} &= \begin{pmatrix} T_1 & & & \\ & T_2 & & \\ & & \ddots & \\ & & & T_M \end{pmatrix} \\ T_i &= -m_i\mu_i \begin{pmatrix} 1 & -1 & 0 & \cdots & 0 \\ 0 & 1 & -1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & -1 \\ 0 & 0 & \cdots & 0 & 1 \end{pmatrix}_{m_i \times m_i},\ \ i = 1, 2, \ldots, M \end{aligned}$$

¿From (34) we obtain

$$E[T_{min}] = -(\pi_{he} \otimes \pi_{he})[T_{he} \oplus (T_{he}/c_f)]^{-1}\mathbf{1} \tag{38}$$

where

$$\pi_{he} = [\alpha_{he} T_{he}^{-1} \mathbf{1}_{he}]^{-1} \alpha_{he} T_{he}^{-1} = \left[\sum_{i=1}^{M} p_i \alpha_i T_i^{-1} \mathbf{1}_{T_i} \right]^{-1} \begin{pmatrix} p_1 \alpha_1 T_1^{-1} & p_2 \alpha_2 T_2^{-1} & \cdots & p_M \alpha_M T_M^{-1} \end{pmatrix},$$

and $\mathbf{1}$ is a column vector of all 1's with appropriate dimension for matrix multiplication (the dimension is $\left(\sum_{i=1}^{M} m_i\right)^2$).

As a final remark, we notice that the approach using the Residue theorem may overcome the dimension explosion inherited in the matrix-geometric approach, however, the former does not give explicit formula while the latter does.

References

[1] Apache 1.3. HTTP Server Document; http://www.apache.org, 2000.

[2] Asmussen, S. Matrix-analytic Models and Their Analysis. *Scandinavian Journal of Statistics*, 27(2):193–226, 2000.

[3] Cao, P., and Irani, S. Cost-Aware WWW Proxy Caching Algorithms. *Proc. Usenix Symp. Internet Technologies and Systems*, 1997.

[4] Chuang, Y.-M., Lee, T.-Y., and Lin, Y.-B. Trading CDPD Availability and Voice Blocking Probability in Cellular Networks. *IEEE Network*, 2(12):48–54, 1998.

[5] Fang, Y., and Chlamtac, I. Teletraffic Analysis and Mobility Modeling for PCS Networks. *IEEE Transactions on Communications*, 47(7):1062–1072, July 1999.

[6] Helme, S. The New Generation. *Mobilecommunications Asia*, pages 12–16, January 2000.

[7] Latouche, G., and Ramaswami, V. *Introduction to Matrix Analytic Methods in Stochastic Modeling*. SIAM, Philadelphia, 1999.

[8] Lin, Y.-B., and Chlamtac, I. *Wireless and Mobile Network Architectures*. John Wiley & Sons, 2001.

[9] Maglio, P.P., and Barrett, R. Intermediaries Personalize Information Streams. *Communications of ACM*, 43(8):68–74, 2000.

[10] Naldi, M. Measurement-based modelling of Internet dial-up access connections. *Computer Networks*, 31(22), 1999.

[11] Omnisky. http://www.omnisky.com, 2000.

[12] Pitkow, J.E. Summary of WWW Characterizations. *Computer Networks and ISDN Systems*, 30(1-7), 1998.

[13] Ross, S.M. *Stochastic Processes*. John Wiley & Sons, 1996.

[14] Shim, J., Scheuermann,P. and Vingralek, R. Proxy cache algorithms: design, implementation, and performance. *IEEE Transactions on Knowledge and Data Engineering*, 11(4), 1999.

[15] Squid 2.3. Internet Object Cache Document; http://squid.nlanr.net/Squid, 2000.