A Study on SIP Session Timer for Wireless VoIP

Ai-Chun Pang*, Chih-Hsiao Liu†, Shu Ping Liu‡ and Hui-Nien Hung§
*Graduate Institute of Networking and Multimedia, Dept. of Comp. Sci. & Info. Engr., National Taiwan University
Taipei, Taiwan, R.O.C.; Email: acpang@csie.ntu.edu.tw
†‡Dept. of Comp. Sci. & Info. Engr., National Taiwan University
Taipei, Taiwan, R.O.C.; Email: {r91103, r92065}@csie.ntu.edu.tw
§Institute of Statistics, National Chiao Tung University
Hsinchu, Taiwan, R.O.C.; Email: hhung@stat.nctu.edu.tw

***Abstract*— The *Session Timer* mechanism for Session Initiation Protocol (SIP) has been proposed to track the states of the communicating sessions for proxy servers. Upon the occurrence of the session failure (e.g., radio link disconnection), the proxy server can quickly release the resources allocated for the failed session by using *SIP Session Timer*. Based on *SIP Session Timer*, we propose a dynamic session refreshing approach to adjust the session timer depending on the conditions of radio links for wireless VoIP subscribers. The study indicates that with our dynamic session refreshing approach, the session failure can be efficiently detected without the considerable increase of signaling traffic.**

I. INTRODUCTION

With the explosive growth of Internet subscriber population, supporting Internet telephony services, also known as *Voice over IP* (VoIP), is considered as a promising trend in telecommunication business. Recent introduction of mobile/wireless systems (e.g., 3G/GPRS, IEEE 802.11 WLAN, Bluetooth) has driven the Internet into new markets to support mobile/wireless users. Thus, how to efficiently provide VoIP services over mobile/wireless networks becomes an important research issue, which has been intensively studied [5], [1], [10], [6].

Two major standards are currently used for VoIP products. One is proposed by the ITU-T/H.323, and the other is developed by the IETF/SIP (Internet Engineering Task Force/Session Initiation Protocol). SIP brings simplicity, familiarity, and clarity to Internet telephony that H.323 does not have. Thus a lot of documents about SIP are under development and many SIP-based VoIP products are on the market. Besides, SIP has been promised to be used for future all-IP mobile networks to deliver voice services, and this indicates that the role of SIP becomes more and more important for wireless VoIP industries [8].

SIP [11] is an application-layer signaling protocol for creating, modifying and terminating multimedia sessions or calls. Two major network elements are defined in SIP: the user agent and the network server. The user agent (UA) that contains both a *user agent client* (UAC) and a *user agent server* (UAS) resides in SIP terminals such as hard-phones and soft-phones. The UAC (or calling user agent) is responsible for issuing SIP requests, and the UAS (or called user agent) receives the SIP request and responds to the request. There are three types of SIP network servers: the proxy server, the redirect server and the registrar. The proxy server forwards the SIP requests from a UAC to the destination UAS. Also, the proxy server is responsible for performing user authentication, service logic execution and billing/charging for a SIP-based VoIP network. The redirect server plays a similar role as the proxy server, except that the redirect server responds to a request issuer with the destination address instead of forwarding the request. To support user mobility, a UA informs the network of its current location by explicitly registering with a registrar. The registrar is typically co-located with a proxy or redirect server.

The basic SIP specification described above does not define a mechanism for proxy servers to track the states of the sessions after the sessions are established [11]. In other words, a basic SIP proxy server is not able to determine whether an established session is still alive or dead. For example, when a wireless UA in conversation fails to connect the network (e.g., due to abnormal radio disconnection), the SIP proxy server can not recognize the failure of the session. Then the proxy server will hold the resources reserved for the failed session, which results in the blockings of other new session requests due to the lack of the resources. To resolve this problem, one of SIP Extensions, *SIP Session Timer*, specifies a keep-alive mechanism for established SIP sessions [3]. In this mechanism, the duration of a communicating session is extended by using an UPDATE request sent from one SIP UA to the proxy server (then to the other SIP UA). A session timer (maintained in the proxy server and the UAs) records the duration of the session that the UA requests to extend. When the session timer nearly expires, the UA re-sends an UPDATE request to refresh the session interval. Note that the initial value for the session timer is carried in an INVITE request instead of UPDATE.

The existing approaches to implement the *SIP Session Timer* mechanism are based on static (periodic) session refreshing. That is, the interval between two successive UPDATE requests (i.e., the length of the session timer) is set to a fixed value. The selection of the length for the session timer significantly affects the system performance in the static session refreshing

TABLE I

THE VARIABLES USED IN OUR DYNAMIC SESSION REFRESHING APPROACH

Parameter	Description	Value
Adjusting Window (AW)	The window size for collecting radio link information from lower layers	2
Average FER (aFER)	The average FER value within AW	-
Bad Threshold (BT)	Used to check whether the state of the network is bad or not	28%
Decreasing Ratio (DR)	The ratio used to decrease the session timer	1.15
Good Threshold (GT)	Used to check whether the state of the network is good or not	10%
Increasing Ratio (IR)	The ratio used to increase the session timer	1.30
Lower Bound (LB)	A lower limit of the length of the session timer	$1/20\mu$
Network Change (NC)	Used to check whether the network state changes or not	18%
Query Number (QN)	The number of queries for retrieving the lower-layer radio link information	-
Session Timer (ST)	The session interval	-
Upper Bound (UB)	An upper limit of the length of the session timer	$1/5\mu$

approach. If the interval is small, the proxy server can release the reserved resources for the disconnecting session quickly. However, a small session interval may cause excessive messaging traffic. On the contrary, if the session interval is large, excessive messaging traffic can be avoided while the proxy server will hold the useless resources for a long time. From the above reasons, the length of the session timer should be adaptively changed according to different network situations. Thus based on *SIP Session Timer*, we propose a dynamic session refreshing approach to adjust the session interval based on the network state to optimize the system performance.

II. THE DYNAMIC SESSION REFRESHING APPROACH

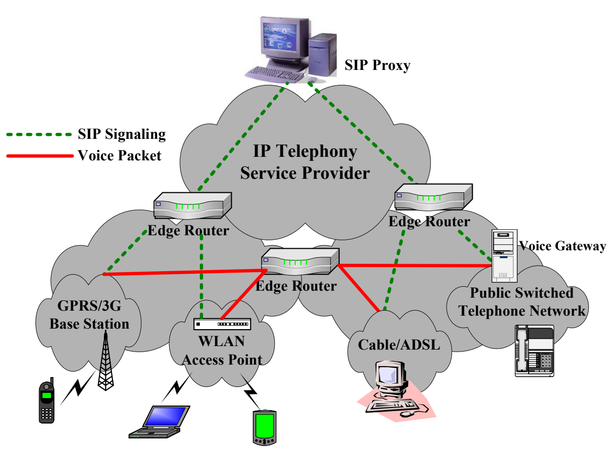


Fig. 1. The SIP-based VoIP network architecture

In *SIP Session Timer*, the UA in conversation sends an UPDATE request to extend the duration of the communicating session. The interval for two consecutive UPDATE requests (i.e., the length of the session timer) is determined through a negotiation between the UAC and the UAS. If an UPDATE request is not received before the session timer expires, the session is considered as abnormal disconnection, and will be force-terminated. Then the proxy server will release the allocated resources for the failed session.

Based on the network architecture shown in Figure 1, our dynamic session refreshing approach is described below. In this figure, SIP UAs can access IP telephony services via heterogeneous networks including the wireless/mobile networks (e.g., IEEE 802.11 WLAN and GPRS/3G) and the wireline networks (e.g., cable, ADSL and PSTN). In Figure 1, the dashed and solid lines respectively represent the SIP signaling and RTP(Real-time Transport Protocol)/RTCP(RTP control protocol) voice paths, where the SIP signaling is carried through the proxy servers, and the voice packets are directly transmitted between two communicating UAs. For an established session, abnormal detaching from the network for one of the participant UAs will result in the session force-termination. By using the SIP Session Timer mechanism, the occurrence of the session force-termination can be detected by the proxy server, and then the proxy server can quickly release the resources allocated for the failed session.

The UA's abnormal detaching mainly results from the following reasons: (1) the crash of the UA , and (2) the radio disconnection for a wireless UA. The crash of the UA is not predictable and seldom occurs while the condition of mobile/wireless networks can be estimated based on the data collected from low layers (e.g., medium access control; MAC). Note that we assume that the probability for the congestion/disconnection of wireline networks is negligible. Thus to efficiently provide the *SIP Session Timer* mechanism, we propose a dynamic session refreshing approach to adaptively adjust the length of the session timer based on the condition of the mobile/wireless network to improve the system performance.

To estimate the state of the radio link for a wireless UA, the data from lower layers (e.g., MAC) should be periodically collected. If the collected data indicate that the *frame error rate* (FER) (or packet loss statistics) has been low for a period of time, the network condition is considered as a good state. The period of time denoted by the **Adjusting Window** (AW) is used as a history reference to determine the point of the next UPDATE request. All FER values collected within an AW are weight-averaged, and its value is denoted by $aFER$.

A low $aFER$ value represents a "GOOD" network state with low probabilities of packet loss and radio disconnection. Whether the network state is identified as good or not depends on the **Good Threshold** (GT). If $aFER$ is equal to or less than GT, the network condition is considered as a good state. In this case, to save the network bandwidth, the session timer is increased based on the **Increase Ratio** (IR) to avoid sending the UPDATE request frequently. If the network state has been good for a long time, the session interval will become extremely large. Suppose that the session disconnection suddenly occurs. With such a large session timer, the session failure will be detected by the proxy server too slowly. Thus, to prevent the session timer from being over-enlarged, an **Upper Bound** (UB) for the session timer is set.

On the contrary, when $aFER$ is high (i.e., equal to or larger than the **Bad Threshold** (BT)), the network condition is considered as a bad state. In this state, the probability of packet loss is high, and the established session will fail due to the radio disconnection very probably. Thus in order to detect the session failure earlier, the UPDATE requests should be sent to the proxy server more frequently by decreasing the session timer based on the **Decrease Ratio** (DR). Similar to UB in the good state, a **Lower Bound** (LB) in the bad state is used to prevent the session timer from being over-reduced, which results in overwhelming signaling traffic and the decrease of the available network bandwidth.

Based on the above descriptions, the session interval can be smoothly increased/decreased with IR/DR according to the estimated state of the radio link. However, when the network condition rapidly switches between "GOOD" and "BAD" states, the session timer may not be immediately changed to a proper value by using IR/DR. To further improve our approach, the situation for the significant change between the network states should be considered. Whether a significant network change occurs depends on the difference between the previous collected FER value ($pFER$) and the current collected FER value ($cFER$) from the lower layer. If $|pFER - cFER| > NC$ (**Network Change**), the session interval is adjusted to the initial value instead of slightly increasing/decreasing the current value by IR/DR.

The flow chart of our dynamic session refreshing algorithm is shown in Figure 2, and its steps are described as follows. The variables used in our dynamic session refreshing algorithm are summarized in Table I. Table I also presents the values set for these variables for our experiments shown in Section III.

III. PEFORMANCE EVALUATION

Based on the scenario shown in Figure 3, this section proposes an analytic model [1], and a simulation models to evaluate the performance of *SIP Session Timer* for wireless VoIP. Our analytic model has been validated against the simulation experiments. The simulation model follows the approach we developed in [9], and the details are omitted.

In Figure 3, the proxy server monitors the state (i.e., dead or alive) of the communicating session between UA1 and

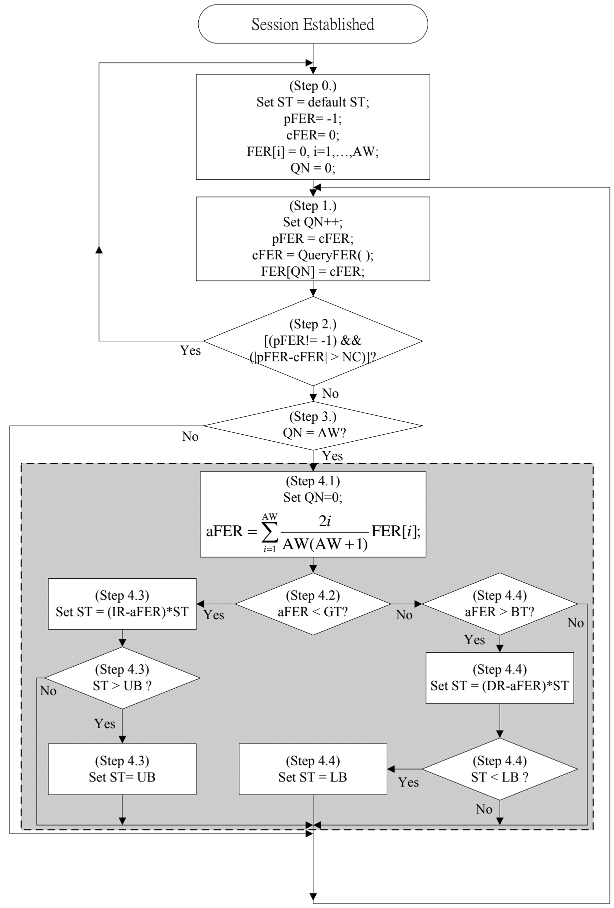


Fig. 2. The flow chart for our dynamic session refreshing approach

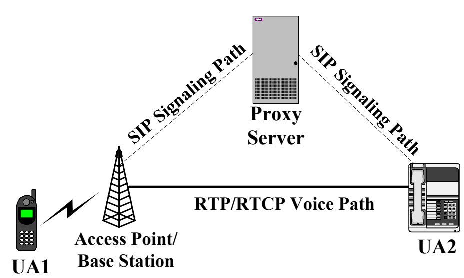


Fig. 3. The scenario for the analytic model

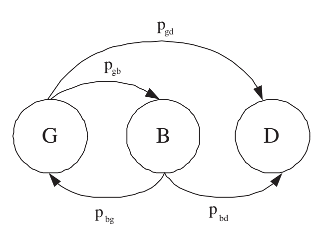


Fig. 4. The transition probabilities between the network conditions for the wireless link (G: GOOD, B: BAD and D: DEAD)

[1]For details of our analytic model, the reader is referred to http://www.csie.ntu.edu.tw/~acpang/TR.htm.

UA2 through the *SIP Session Timer* mechanism. We assume that UA1 accesses the IP telephony service via the wireless link such as IEEE 802.11 WLAN and 3G/GPRS, and UA2 is connected to the Internet through the wireline access (e.g., ADSL and cable). After the session is established, UA1 is responsible for issuing the UPDATE request to the proxy server to refresh the session interval. By using UPDATE from UA1, the proxy server is informed about whether the session is dead or alive. Note that by using the quality feedback information carried in RTCP packets, our model can be easily extended to the case where both UA1 and UA2 are the wireless VoIP users. In the remainder of this paper, the term "call" represents the real-time multimedia/voice session.

To model the condition of the wireless link for UA1, three kinds of network states,"GOOD", "BAD" and "DEAD", are considered. Different kinds of network states represent different *Frame Error Rate* (FER) for wireless links. A large FER leads to a high probability of packet loss. When UA1 (i.e., the call that UA1 involves) resides in "GOOD" state, the FER and packet-loss probability are small, and most voice and signaling packets can be successfully transmitted from UA1 to the proxy server and then to UA2. In "BAD" state, with a large FER, the network condition is unstable, and this results in a large number of lost packets. When the wireless network enters in "DEAD" state, the signaling path (between UA1 and the proxy server) and the voice path (between UA1 and UA2) are force-disconnected, and all packet deliveries from UA1 will fail. Figure 4 shows the transition probabilities between "GOOD", "BAD" and "DEAD" states for the wireless link of UA1, where $P_{bd} + P_{bg} = 1$ and $P_{gd} + P_{gb} = 1$. The time intervals (i.e., t_g and t_b) that UA1 stays in "GOOD" and "BAD" states are assumed to have Exponential distributions with rates λ_g and λ_b, respectively. This assumption will be relaxed to accommodate Gamma distributions for t_g and t_b in our developed simulation model. The Gamma distribution is often used in wireless/mobile communications network modeling [2], [4]. It has been shown that the distribution of any positive random variable can be approximated by a mixture of Gamma distributions (see Lemma 3.9 in reversibility and stochastic networks [7]). Also, we assume that the packet loss probabilities for "GOOD" and "BAD" states are respectively P_{lg} and P_{lb}.

Furthermore, the following input parameters are considered.

> t_c: the call holding time with an Exponential rate μ. We assume that the call termination is performed either by UA1 or by UA2 through issuing the BYE request.
> t_u: the time interval between two consecutive UPDATE packets for UA1. We assume that t_u has an Exponential distribution with rate λ_u.
> P_s: the probability that the packet is successfully transmitted between UA1 and the proxy server (or UA2). Thus $P_s = \frac{(1-P_{lg})\lambda_b+(1-P_{lb})\lambda_g}{\lambda_b+\lambda_g}$.

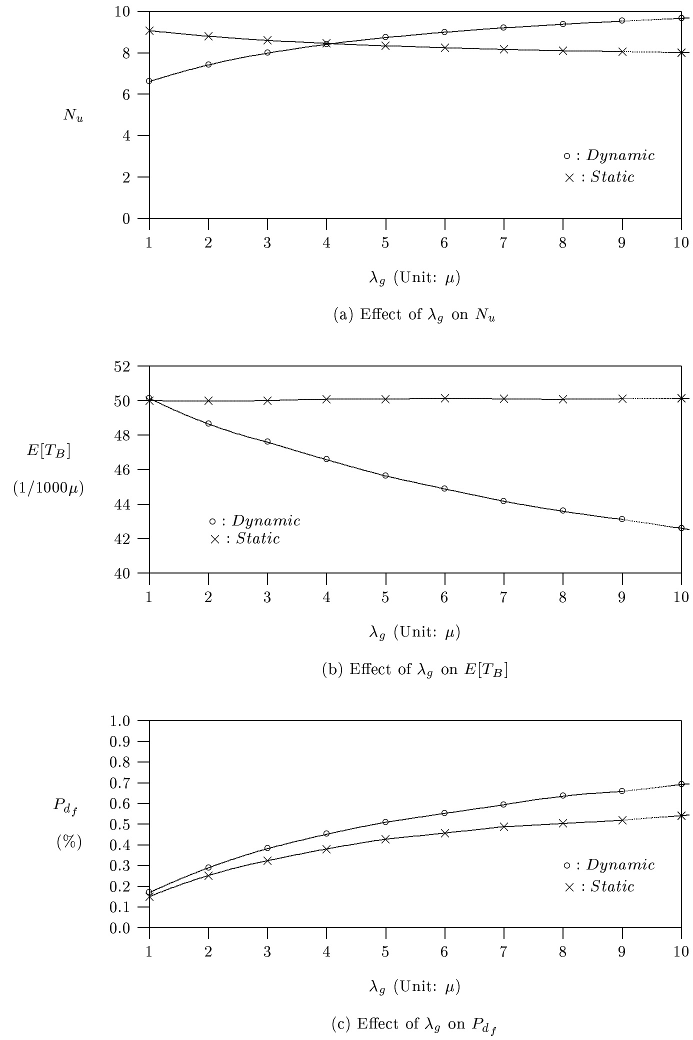


Fig. 5. The Effect of λ_g on N_u, $E[T_B]$ and P_{d_f}

Several output measures are defined in this study, and listed as follows.

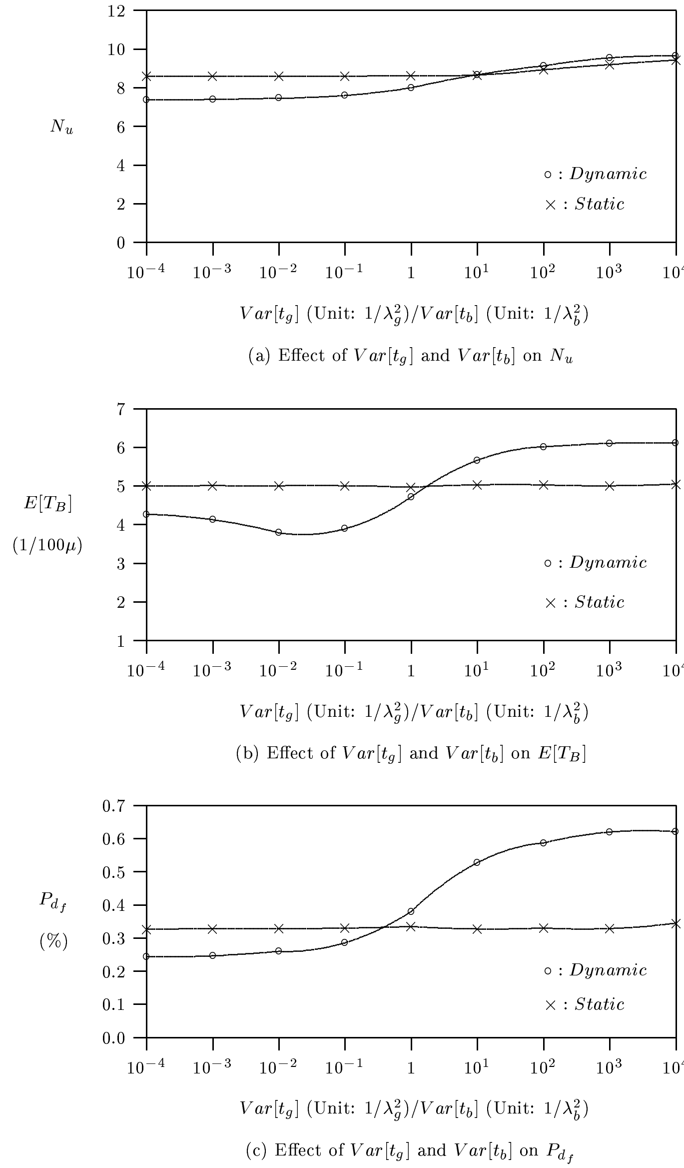
> P_{d_f}: the probability that the detection event (i.e., UPDATE loss) occurs before the call actually fails or completes. This probability is also called the mis-detection probability.
> N_u: the average number of UPDATE requests transmitted between UA1 and UA2 (via the proxy server) for an established call
> $E[T_B]$: the expected number of **Bad Debt**. The **Bad Debt** is defined as the time interval between the time that the failure (i.e., UA1 enters in "DEAD" state) occurs and the time that the proxy server releases the resources for the call.

Based on the developed analytic and simulation models, we use numerical examples to investigate the performance of N_u (i.e., the expected number of UPDATE requests for a call), P_{d_f} (i.e., the mis-detection probability) and $E[T_B]$ (i.e., the expected number of **Bad Debt**) for our dynamic session refreshing approach. In our experiments, the default values for the input parameters are set, i.e., $\lambda_g = 3\mu$, $\lambda_b = 5\mu$, $P_{lg} = 10^{-6}$, $P_{lb} = 10^{-3}$, $P_{gd} = 10^{-6}$ and $P_{bd} = 0.05$. Furthermore, the initial value for the session timer (ST) is set to $\frac{1}{10\mu}$, and the query frequency for radio-link information is 30μ.

Effect of λ_g: Figures 5 (a), (b) and (c) plot the the expected number of UPDATE requests per call (N_u), the expected number of **Bad Debt** ($E[T_B]$), and the mis-detection probability (P_{d_f}) as a function of λ_g, where the input parameters except λ_g are set to the default values. In Figure 5 (a), as λ_g increases (i.e., the reduction of the average time of the good state where a wireless UA resides), the curves for the static and dynamic session refreshing approaches respectively decrease and increase. Even in a very unstable network condition, less than two more UPDATE requests are required on average. For $\lambda_g \leq 4\mu$, the static session refreshing approach has more UPDATE requests than the dynamic one. On the other hand, when λ_g is larger than 4μ, the opposite result is observed. This phenomenon is explained as follows. As λ_g increases, the average time of the bad state for a call relatively increases. Thus the call suffers from the radio disconnection more probably, and the call holding time decreases due to the increasing force-termination probability. For the static session refreshing approach, the UPDATE request is periodically sent regardless of the network state. As the call holding time decreases, N_u for the static approach decreases. On the contrary, the frequency of UPDATE deliveries for our dynamic approach increases when the network state remains bad, and this results in the increase of the session refreshing number N_u.

Figure 5 (b) shows that $E[T_B]$ for the static session refreshing approach is not influenced by λ_g. However, for the dynamic session refreshing approach, $E[T_B]$ significantly decreases as λ_g increases (up to nearly 18% of $E[T_B]$ is saved), and this indicates that our dynamic approach effectively adjusts the session timer especially when the network condition is unstable. Furthermore, as shown in Figure 5 (c), P_{d_f} is an increasing function of λ_g for both the static and dynamic approaches. The mis-detection results from UPDATE loss prior to the actual call failure or completion. As λ_g increases, the UPDATE request gets more probably lost due to the increase of the time of the bad state for a call, and thus the mis-detection probability increases. Specifically, the increasing rate of P_{d_f} is slightly faster for the dynamic approach than for the static one, and this results from more UPDATE deliveries per call for our dynamic session refreshing.

Effect of the Variances of t_g and t_b: Figure 6 plots N_u, $E[T_B]$ and P_{d_f} as a function of $Var[t_g]$ and $Var[t_b]$, where $Var[t_g]$ and $Var[t_b]$ are the variances of t_g and t_b. In this experiment, t_g and t_b are assumed to have Gamma distributions with means $1/\lambda_g = 1/3\mu$ and $1/\lambda_b = 1/5\mu$, respectively. Figure 6 shows that the performance of N_u, $E[T_B]$ and P_{d_f} for the static session refreshing approach is almost un-affected by $Var[t_g]$ and $Var[t_b]$. However, these output measures for the dynamic approach increase as $Var[t_g]$ and $Var[t_b]$ increase. Specifically, when $Var[t_g] > 1/\lambda_g^2$ ($Var[t_b] > 1/\lambda_b^2$), N_u, $E[T_B]$ and P_{d_f} significantly increase as $Var[t_g]$ and $Var[t_b]$ increase. On the other hand, $Var[t_g]$ and $Var[t_b]$ have an insignificant effect on N_u, $E[T_B]$ and P_{d_f} when $Var[t_g]$ and $Var[t_b]$ are small. For our dynamic session refreshing, the length of the session timer is adaptively adjusted according the estimation of the network condition. The estimation of the network condition is performed through the collection and calculation of the history records. The large $Var[t_g]$ and $Var[t_b]$ indicate that the network state is unpredictable, which results in the performance degradation of our dynamic session refreshing approach.

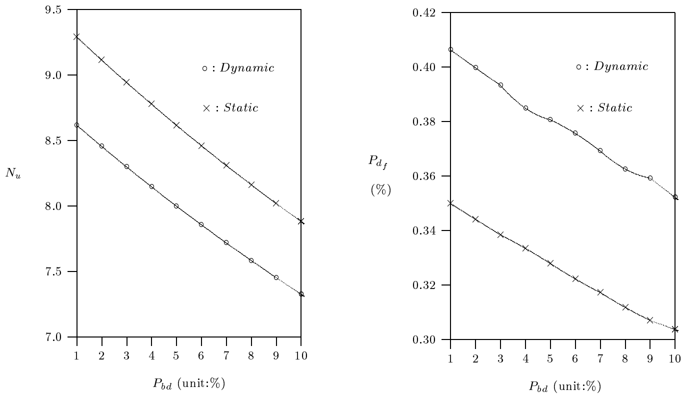


(a) Effect of $Var[t_g]$ and $Var[t_b]$ on N_u

(b) Effect of $Var[t_g]$ and $Var[t_b]$ on $E[T_B]$

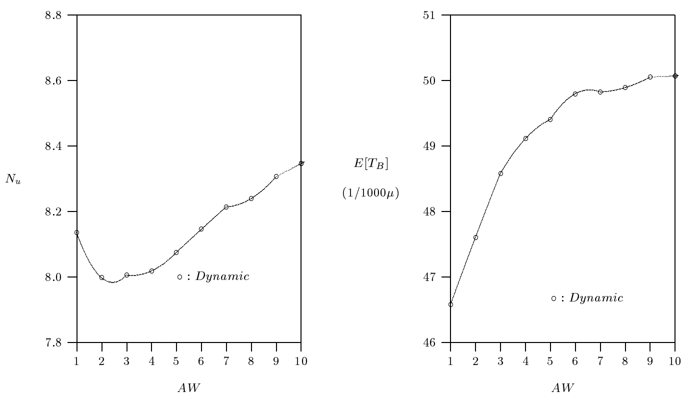
(c) Effect of $Var[t_g]$ and $Var[t_b]$ on P_{d_f}

Fig. 6. The Effect of $Var[t_g]$/$Var[t_b]$ on N_u, $E[T_B]$ and P_{d_f}



(a) Effect of P_{bd} on N_u (b) Effect of P_{bd} on P_{d_f}

Fig. 7. The Effect of P_{bd} on N_u and P_{d_f}



(a) Effect of AW on N_u (b) Effect of AW on $E[T_B]$

Fig. 8. The Effect of AW on N_u and $E[T_B]$

Effect of P_{bd}: Figure 7 plots N_u and P_{d_f} as a function of P_{bd}. The curve for the effect of P_{bd} on $E[T_B]$ is not presented since **Bad Debt** is irrelevant to the transition probability from the bad state to the dead state for an established call. Figure 7 (a) shows that for both the static and dynamic session refreshing approaches, N_u decreases as P_{bd} increases. The increase of P_{bd} results in more call force-terminations due to the session failure, and thus the decrease of the number of UPDATE deliveries. Furthermore, the curve for static session refreshing is steeper than that for dynamic session refreshing. The decreasing rate of N_u for these two approaches depends on the ratio of t_g to t_b where an established call resides. If $\frac{\lambda_b}{\lambda_g} > 1$, the decreasing rate of N_u for the static approach is faster than that for the dynamic one. On the contrary, an opposite result is observed. Figure 7 (b) shows that P_{d_f} decreases as P_{bd} increases for both the static and dynamic session refreshing approaches.

Effect of AW: Figure 8 illustrates the effect of AW on N_u and $E[T_B]$ for the dynamic session refreshing approach. Figure 8 (a) shows an intuitive result that N_u decreases and then increases as AW increases. In other words, a large window size does not benefit our dynamic approach since collecting too much out-of-date information results in the misjudgment of our algorithm. From Figure 8 (a), we observe that the optimum value of AW is 2. Also, Figure 8 (b) shows that for the dynamic session refreshing approach, $E[T_B]$ increases as AW increases.

IV. Conclusions

Session Initiation Protocol (SIP) has been considered as the most promising candidate for call setup signaling for future VoIP (Voice over IP) services. The *SIP Session Timer* mechanism was proposed to track the states of the communicating sessions for proxy servers. Upon the occurrence of the session failure (e.g., radio link disconnection), the proxy server can quickly release the resources allocated for the failed session by using *SIP Session Timer*. Based on *SIP Session Timer*, we proposed a dynamic session refreshing approach to adjust the session timer depending on the conditions of radio links for wireless VoIP subscribers. An analytic and a simulation models were developed to investigate the performance of the static and dynamic session refreshing approaches. Our study indicates that with this new approach, the session failure can be efficiently detected without the considerable increase of signaling traffic (i.e., less than two more UPDATE requests are sent while up to 18% of the **Bad Debt** is reduced as the network condition becomes unstable) and the optimum value of **Adjusting Window** in our approach is two.

Acknowledgement

This work was supported in part by National Science Council under contract NSC 93-2213-E-002-093 and NSC 93-2213-E-002-025, Intel, Microsoft and CCL/ITRI.

References

[1] Chang, M.-F., Lin, Y.-B., and Pang, A.-C. vGPRS: A Mechanism for Voice over GPRS. *ACM Wireless Networks*, 9:157–164, 2003.
[2] Chlamtac, I., Fang, Y., and Zeng, H. Call Blocking Analysis for PCS Networks under General Cell Residence Time. *IEEE WCNC, New Orleans*, September 1999.
[3] Donovan, S. and Rosenberg, J. The SIP Session Timer. Technical Report draft-ietf-sip-session-timer-14, Internet Engineering Task Force, February 2004.
[4] Fang, Y. and Chlamtac, I. Teletraffic Analysis and Mobility Modeling for PCS Network. *IEEE Trans. on Comm.*, 47(7):1062–1072, 1999.
[5] Garg, S. and Kappes, M. An experimental study of throughput for UDP and VoIP traffic in IEEE 802.11b networks. *IEEE WCNC*, April 2003.
[6] Garg, S. and Kappes, M. Can I add a VoIP call? *IEEE ICC*, May 2003.
[7] Kelly, F. P. *Reversibility and Stochastic Networks*. John Wiley & Sons Ltd., 1979.
[8] Lin, Y.-B., Huang, Y.-R., Pang, A.-C. and Chlamtac, Imrich. All-IP Approach for Third Generation Mobile Networks. *IEEE Network*, 16(5):8–19, 2002.
[9] Pang, A.-C., Lin, Y.-B., Tsai, H.-M., Agrawal, P. Serving Radio Network Controller Relocation for UMTS All-IP Network. *IEEE Journal on Selected Aarea in Communications*, 22(4), 2004.
[10] Rao, Herman C.-H., Lin, Y.-B. and Chou, S.-L. iGSM: VoIP Service for Mobile Network. *IEEE Communications Magazine*, April 2000.
[11] Rosenberg, J., et al. SIP: Session Initiation Protocol. IETF RFC 3261, June 2002.